

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
AUTOMAÇÃO INDUSTRIAL

ENGENHARIA IMUNOLÓGICA: DESENVOLVIMENTO E
APLICAÇÃO DE FERRAMENTAS COMPUTACIONAIS
INSPIRADAS EM SISTEMAS IMUNOLÓGICOS ARTIFICIAIS

Autor: Leandro Nunes de Castro Silva
Orientador: Prof. Dr. Fernando José Von Zuben

Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos pré-requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica

Área de Concentração: Engenharia de Computação

Campinas – SP – Brasil

Maio de 2001

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

Si38e

Silva, Leandro Nunes de Castro
Engenharia Imunológica: Desenvolvimento e Aplicação de
Ferramentas Computacionais Inspiradas em Sistemas Imunológicos
Artificiais / Leandro Nunes de Castro Silva. --Campinas, SP: [s.n.],
2001.

Orientador: Fernando José Von Zuben.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de
Engenharia Elétrica e de Computação.

1. Engenharia. 2. Sistema imune. 3. Computação. 4. Inteligência
Artificial. I. Zuben, Fernando José Von II. Universidade Estadual de
Campinas. Faculdade de Engenharia Elétrica e de Computação. III.
Título.

Banca Examinadora

Prof. Dr. Fernando José Von Zuben
(DCA/FEEC/Unicamp)

Prof. Dr. Felipe M. G. França
(COPPE/UFRJ)

Profa. Dra. Dagmar Ruth Stach Machado
(DMI/IB/Unicamp)

Prof. Dr. Christiano Lyra Filho
(DENSIS/FEEC/Unicamp)

Prof. Dr. Márcio Luiz de Andrade Netto
(DCA/FEEC/Unicamp)

Prof. Dr. Fernando Antônio Campos Gomide
(DCA/FEEC/Unicamp)

Prof. Dr. Ricardo Ribeiro Gudwin
(DCA/FEEC/Unicamp)

Novos paradigmas são propostos e aceitos constantemente não necessariamente por serem fiéis à sua fonte inspiradora ou por representarem verdades absolutas, mas por serem úteis e factíveis.

Agradecimentos

Ao Prof. Dr. Fernando José Von Zuben pela orientação e amizade durante todos estes anos de pós-graduação na Unicamp.

À Profa. Dr. Dagmar R. Stach-Machado do DMI/IB pela orientação indispensável nos tópicos referentes à imunologia.

Aos membros da banca pelas contribuições na geração da versão final da tese.

Aos meus pais pelo apoio e carinho que dedicam à nossa família.

Aos meus amigos que compartilharam comigo, direta ou indiretamente, este período de muito trabalho, alegrias e tristezas.

A todas aquelas pessoas que estiveram comigo e que me motivaram e apoiaram no desenvolvimento da tese.

À Faculdade de Engenharia Elétrica e de Computação e ao Departamento de Engenharia de Computação e Automação Industrial pelo fornecimento de instalações e condições de trabalho apropriadas.

À FAPESP pela concessão de uma bolsa de doutoramento (processo n. 98/11333-9) no país.

Ao Prof. Dr. Juan Manuel Corchado pela viabilização de uma visita ao Instituto de Linguagens e Sistemas de Informática da Universidade de Vigo, Ourense/Espanha em janeiro/fevereiro de 1999.

Ao Instituto Santa Fé, pela concessão de auxílio para participação no workshop intitulado “Design Principles for the Immune System and Other Distributed Autonomous Systems”, realizado no Instituto Santa Fé, Novo México/U.S.A., em julho de 1999.

Ao IEEE e à AAAI pela concessão de auxílios financeiros para participação em diversas conferências internacionais nos anos de 1998, 1999 e 2000.

À força suprema responsável pela criação e destino da humanidade.

É com muito orgulho que dedico este trabalho à meus pais, irmãos e sobrinhos. Tenho a certeza de que mesmo distantes estamos sempre unidos pela força inexplicável que denominamos de amor.

Sumário

BANCA EXAMINADORA.....	i
AGRADECIMENTOS.....	v
SUMÁRIO.....	ix
RESUMO.....	xiii
CAPÍTULO 1: INTRODUÇÃO.....	1
1.1. INTRODUÇÃO E MOTIVAÇÃO.....	1
1.2. OBJETIVOS E CONTRIBUIÇÕES.....	2
1.3. NOTAÇÃO.....	3
1.4. FORMALIZAÇÃO DA TESE.....	4
1.4.1. Estrutura da Tese.....	4
1.4.2. Trabalhos Complementares.....	5
CAPÍTULO 2: O SISTEMA IMUNOLÓGICO.....	7
2.1. INTRODUÇÃO.....	7
2.2. BREVE HISTÓRICO DA PESQUISA EM IMUNOLOGIA.....	8
2.3. PRINCÍPIOS FUNDAMENTAIS E ELEMENTOS CONSTITUINTES.....	12
2.4. MECANISMOS BÁSICOS DE DEFESA DO SISTEMA IMUNOLÓGICO.....	13
2.5. ANATOMIA DO SISTEMA IMUNOLÓGICO.....	14
2.6. SISTEMA IMUNE ADAPTATIVO.....	17
2.7. RECONHECIMENTO DE PADRÕES.....	18
2.7.1. A Molécula de Anticorpo e a Diversidade Imunológica.....	20
2.7.2. O Receptor de Célula T (TCR) e a Diversidade Imunológica.....	22
2.8. PRINCÍPIO DA SELEÇÃO CLONAL.....	23
2.8.1. Aprendizagem por Reforço e Memória Imunológica.....	25
2.8.2. Maturação de Afinidade.....	27
2.9. DISTINÇÃO PRÓPRIO/NÃO-PRÓPRIO.....	30
2.9.1. Seleção Positiva.....	32
2.9.2. Seleção Negativa.....	33
2.10. TEORIA DA REDE IMUNOLÓGICA.....	35
CAPÍTULO 3: ENGENHARIA IMUNOLÓGICA.....	39
3.1. INTRODUÇÃO.....	39
3.2. SISTEMAS IMUNOLÓGICOS ARTIFICIAIS E ENGENHARIA IMUNOLÓGICA.....	41
3.3. ESTADO DA ARTE EM SISTEMAS IMUNOLÓGICOS ARTIFICIAIS.....	45
3.3.1. Alguns Trabalhos Pioneiros.....	45
3.3.2. Reconhecimento de Padrões.....	46
3.3.3. Segurança Computacional.....	47
3.3.4. Robótica.....	48
3.3.5. Otimização.....	48
3.3.6. Controle.....	49
3.3.7. Abordagens Conexionistas.....	49
3.3.8. Detecção de Falhas e Anomalias.....	50
3.3.9. Abordagens Baseadas em Agentes.....	51
3.3.10. Aprendizagem de Máquina.....	51
3.3.11. Autômatos Celulares.....	52
3.3.12. Outras Aplicações.....	52
3.4. UMA ESTRUTURA FORMAL PARA A ENGENHARIA IMUNOLÓGICA.....	53
3.4.1. O Espaço de Formas (Shape-Space).....	54

3.4.2. Representações Celulares, Moleculares e Suas Afinidades	56
3.4.3. Algoritmos e Processos.....	64
3.5. EXEMPLOS DE APLICAÇÕES	73
3.5.1. Um Sistema Imunológico Artificial.....	74
3.5.2. Aplicações de Engenharia Imunológica.....	77
CAPÍTULO 4: SISTEMAS INTELIGENTES E O SISTEMA IMUNOLÓGICO.....	83
4.1. INTRODUÇÃO	83
4.2. REDES NEURAIS ARTIFICIAIS	84
4.2.1. Características Principais	85
4.2.2. Modelos Neurais e Algoritmos de Treinamento	91
4.3. O SISTEMA IMUNOLÓGICO E O SISTEMA NERVOSO CENTRAL	99
4.3.1. Aspectos Cognitivos	100
4.3.2. Similaridades e Diferenças	104
4.3.3. Engenharia Imunológica e Redes Neurais Artificiais	107
4.4. COMPUTAÇÃO EVOLUTIVA	108
4.4.1. Evolução Natural	109
4.4.2. Algoritmo Evolutivo Básico	111
4.4.3. Algoritmos Genéticos	113
4.4.4. Programação Genética	117
4.4.5. Nichos e Espécies	118
4.5. MICROEVOLUÇÃO: UMA EVOLUÇÃO DENTRO DO INDIVÍDUO.....	120
4.6. OUTRAS ABORDAGENS DE SISTEMAS INTELIGENTES	122
4.6.1. Computação Molecular.....	122
4.6.2. Sistemas Nebulosos	124
CAPÍTULO 5: FERRAMENTAS DE ENGENHARIA IMUNOLÓGICA.....	127
5.1. INTRODUÇÃO	127
5.1.1. Características Gerais	128
5.1.2. Complexidade Computacional.....	129
5.1.3. Funções Para Pseudocódigos.....	131
5.2. SAND: SIMULATED ANNEALING APLICADO AO PROBLEMA DE DIVERSIDADE POPULACIONAL	132
5.2.1. Motivação e Escopo de Aplicações	133
5.2.2. Simulated Annealing	133
5.2.3. Descrição e Análise do SAND.....	134
5.2.4. Caracterização e Comparação.....	139
5.2.5. Complexidade Computacional.....	140
5.3. CLONALG: IMPLEMENTAÇÕES COMPUTACIONAIS DO PRINCÍPIO DE SELEÇÃO CLONAL.....	141
5.3.1. Motivação e Escopo de Aplicações	141
5.3.2. Descrição e Análise do CLONALG	142
5.3.3. Caracterização e Comparação.....	147
5.3.4. Complexidade Computacional.....	149
5.4. ABNET: UMA REDE DE ANTICORPOS	150
5.4.1. Motivação e Escopo de Aplicações	150
5.4.2. Descrição e Análise da ABNET	150
5.4.3. Pseudocódigo.....	156
5.4.4. Caracterização e Comparação.....	157
5.4.5. Complexidade Computacional.....	158
5.5. aiNET: UM MODELO DE REDE IMUNOLÓGICA ARTIFICIAL	159
5.5.1. Motivação e Escopo de Aplicações	160
5.5.2. Descrição	160
5.5.3. Pseudocódigo.....	165
5.5.4. Extração de Conhecimento e Estrutura da aiNet Treinada	166
5.5.5. Caracterização e Comparação.....	174

5.5.6. Complexidade Computacional.....	176
CAPÍTULO 6: APLICAÇÕES EM ENGENHARIA	179
6.1. INTRODUÇÃO	179
6.2. CLONALG.....	180
6.2.1. Reconhecimento de Padrões	180
6.2.2. Otimização	181
6.2.3. Análise de Sensibilidade.....	188
6.2.4. Discussão	190
6.3. SAND.....	191
6.3.1. Espaço de Formas Binário de Hamming	192
6.3.2. Espaço de Formas Euclidianos	197
6.3.3. Análise de Sensibilidade.....	202
6.3.4. Discussão	203
6.4. ABNET	203
6.4.1. Decisão de Lógica Majoritária.....	203
6.4.2. Classificação de Lentes de Contato	207
6.4.3. Reconhecimento de Caracteres Binários	208
6.4.4. Análise de Sensibilidade.....	211
6.4.5. Discussão	213
6.5. AI.NET	214
6.5.1. Análise de Dados	214
6.5.2. Projeto Automático de Redes Neurais de Funções de Base Radial	224
6.5.3. Análise de Sensibilidade.....	231
6.5.4. Discussão	234
CAPÍTULO 7: CONCLUSÕES E PERSPECTIVAS FUTURAS.....	237
7.1. INTRODUÇÃO	237
7.2. CONTRIBUIÇÕES.....	238
7.3. PERSPECTIVAS	239
7.3.1. Novos Algoritmos Inspirados em Mecanismos Imunológicos	239
7.3.2. Variações dos Algoritmos Propostos.....	242
7.3.3. Especulações Sobre o Futuro da Área	244
BIBLIOGRAFIA	245
APÊNDICE: GLOSSÁRIO BIOLÓGICO.....	267
ÍNDICE REMISSIVO	277

Resumo

DE CASTRO, L. N., **Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais**. Campinas: FEEC/UNICAMP, Maio de 2001. Tese de Doutorado – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, p. 286.

Esta tese propõe a *engenharia imunológica* como um novo paradigma de computação a ser empregado na análise, síntese e aplicação de ferramentas computacionais para a solução de problemas complexos de engenharia. A engenharia imunológica (EI) irá definir a ferramenta de solução de um determinado problema baseado nas características do próprio problema, e depois utilizar a ferramenta desenvolvida na obtenção da solução. Ao invés de buscar a modelagem parcial ou total do sistema imunológico, a EI procura desenvolver e implementar modelos computacionais que preservem alguns dos principais mecanismos dos sistemas naturais. A concatenação de grande parte da literatura disponível sobre sistemas imunológicos artificiais, a comparação teórica e empírica com outras abordagens de sistemas inteligentes, a proposição de quatro ferramentas de EI, e a discussão sobre aspectos cognitivos e evolutivos dos sistemas imunológicos fazem parte das contribuições desta tese.

Palavras-chave: representação e extração de conhecimento, problemas complexos de engenharia, sistemas inteligentes, sistema imunológico.

Abstract

DE CASTRO, L. N., **Immune Engineering: Development and Application of Computational Tools Inspired by Artificial Immune Systems**. Campinas: FEEC/UNICAMP, May 2001. Dr. E. E. Thesis – School of Electrical and Computer Engineering, State University of Campinas, p. 286.

This thesis proposes the *immune engineering* (IE) as a new computational paradigm to be employed in the analysis, synthesis and application of computational tools to the solution of complex engineering tasks. The immune engineering is going to define the solution tool for a certain problem based on the characteristics of the problem itself. The resulting tool is then applied to obtain the solution. Instead of partially or fully modeling the immune system, the IE seeks to develop and implement computational models that preserve some of its main mechanisms. The concatenation of most of the works on artificial immune systems, the theoretical and empirical comparisons to other computational intelligence strategies, the proposal of four IE tools, and a discussion concerning cognitive and evolutionary aspects of the immune system are part of the contributions of this thesis.

Keywords: knowledge extraction and representation, complex engineering problems, intelligent systems, immune system.

CAPÍTULO 1

INTRODUÇÃO

Este capítulo visa introduzir e discutir os principais aspectos que levaram ao desenvolvimento e formalização da engenharia imunológica como uma nova abordagem para o tratamento de problemas complexos de engenharia. O enfoque do trabalho, as motivações, objetivos, contribuições e estrutura da tese fazem parte deste capítulo.

“A construção do conhecimento constitui uma conjugação de intelecto e emoção, de razão e vontade; a episteme é fruto de inteligência e de amor.” – Platão

1.1. Introdução e Motivação

Atualmente, existe um grande interesse pelo estudo dos sistemas imunológicos biológicos. Imunologistas e profissionais da área médica estão tentando compreender melhor o sistema imunológico para que eles possam combater doenças infecciosas como a AIDS, doenças auto-imunes como esclerose múltipla, e outros problemas de saúde. Biólogos e imunologistas teóricos estão interessados em modelar o sistema imunológico visando reproduzir fenômenos observados em laboratório e/ou prever comportamentos ainda não obtidos experimentalmente. Outros pesquisadores que começam a se interessar pela imunologia são os engenheiros e cientistas da computação, que tentam simular mecanismos imunológicos particulares com o objetivo de criar sistemas artificiais para a solução de problemas de engenharia. Por exemplo, podem ser traçados fortes paralelos entre o reconhecimento de padrões do sistema imunológico e o reconhecimento de padrões em computação. Uma metáfora ainda mais abrangente, poderia relacionar as características de *imunovigilância* e *resposta imune* com os procedimentos de segurança computacional, que podem incluir detecção e eliminação de vírus, intrusos de rede, e assim por diante.

Esta tese interpreta o sistema imunológico (SI) particularmente como um mecanismo biológico capaz de reconhecer e eliminar elementos causadores de patologias. Deve-se mencionar, entretanto, que esta visão “armamentista” do SI é algumas vezes criticada pelos imunologistas, que preferem abordar o sistema imunológico como um mecanismo de identificação do próprio indivíduo, responsável pela manutenção da integridade física e homeostase do organismo. Trata-se, então, de um ponto de vista particular e intencional sobre os objetivos do SI. A ênfase é dada aos problemas de reconhecimento de padrões pelo sistema imune adaptativo, embora sejam apresentadas discussões sucintas sobre a relação entre o sistema imunológico, o sistema nervoso central e o sistema endócrino. São levantados aspectos cognitivos do SI e alguns de seus mecanismos são caracterizados como

evolutivos, sob o ponto de vista da teoria originalmente proposta por Charles Darwin (1859).

Embora restrito a tópicos específicos, um aspecto peculiar desta tese está na abordagem de alguns conceitos em imunologia que não estão mais em voga ou que são contestados por alguns pesquisadores, como por exemplo a teoria da rede imunológica e seus aspectos cognitivos. No entanto, sob o ponto de vista de engenharia, estes conceitos mostraram-se adequados e úteis por razões que serão apresentadas e discutidas ao longo do texto.

Nas áreas de engenharia e computação, tem surgido um forte interesse pelo estudo dos sistemas imunológicos devido, principalmente, à sua capacidade de processamento de informação. Sob uma perspectiva de engenharia, existem diversas características do SI que podem ser destacadas:

- Unicidade: cada animal possui seu próprio sistema imunológico, com suas capacidades e vulnerabilidades particulares;
- Reconhecimento de padrões internos e externos ao sistema: as células e moléculas que não pertencem ao organismo são reconhecidas e eliminadas pelo SI;
- Detecção de anomalia: o SI pode detectar e reagir a agentes patogênicos (causadores de anomalias) a que o organismo nunca havia sido exposto anteriormente;
- Detecção imperfeita (tolerância a ruídos): um reconhecimento perfeito não é necessário para que o SI reaja contra um elemento causador de patologia (*patógeno*);
- Diversidade: existe uma quantidade limitada de células e moléculas no SI que são utilizadas para se obter o reconhecimento de um número praticamente infinito de elementos, incluindo aqueles sintetizados em laboratório;
- Aprendizagem por reforço: a cada encontro com o mesmo patógeno, o sistema imunológico melhora a qualidade de sua resposta; e
- Memória: os componentes do SI bem sucedidos no reconhecimento e combate às patologias são armazenados para uma resposta futura mais intensa e efetiva.

1.2. Objetivos e Contribuições

Os *Sistemas Imunológicos Artificiais* (SIA) (Dasgupta, 1998a), que surgiram a partir de tentativas de modelar e aplicar princípios imunológicos no desenvolvimento de novas ferramentas computacionais, já vêm sendo utilizados em diversas áreas, como reconhecimento de padrões, detecção de faltas e anomalias, segurança computacional, otimização, controle, robótica, scheduling, análise de dados, aprendizagem de máquina, dentre outras.

Esta tese representa uma iniciativa pioneira no sentido de apresentar uma estrutura formal para o desenvolvimento de sistemas imunológicos artificiais. Mais especificamente, ela propõe a *Engenharia Imunológica* (EI). Partindo da proposição da engenharia imunológica como um novo paradigma de computação e uma nova área de atuação dentro dos SIA, é ilustrado como diversos princípios e mecanismos do sistema imunológico natural podem ser empregados no desenvolvimento de novas ferramentas computacionais para a solução

de problemas em diversas áreas do conhecimento e suas relações com aprendizado de máquina, processamento de informação e resolução de problemas.

Enquanto as técnicas convencionais de engenharia usualmente requerem a especificação detalhada do comportamento preciso de cada componente do sistema, a engenharia imunológica, requer apenas uma especificação geral, ou aproximada, de alguns aspectos que caracterizam o comportamento global do sistema, como uma medida (ou função) de desempenho, afinidade ou adaptabilidade.

Além disso, também é feito um esforço no sentido de relacionar os sistemas imunológicos artificiais com outros paradigmas de inteligência computacional como, *redes neurais artificiais* (RNA), *computação evolutiva* (CE), *lógica nebulosa* (FL – *fuzzy logic*) e *computação de DNA*, ou *computação molecular* (*DNA computing*).

São propostas quatro ferramentas de computação, implementadas sob a forma de algoritmos, ou programas, de computador:

1. SAND (Simulated ANnealing for Diversity): algoritmo baseado na técnica clássica de *Simulated Annealing* (Kirkpatrick *et al.*, 1987) cujo objetivo principal é aumentar a diversidade de uma população de indivíduos (repertório de anticorpos) representado sob a forma de cadeias binárias ou vetores com elementos pertencentes ao conjunto dos números reais;
2. CLONALG (CLONal selection ALGORITHM): implementação computacional do algoritmo de seleção clonal e maturação de afinidade;
3. ABNET (AntiBody NETwork): rede neural artificial Booleana com processos de crescimento (inserção de unidades), poda (remoção de unidades) e atualização de pesos, também baseados nos princípios imunológicos da seleção clonal e maturação de afinidade;
4. aiNet (Artificial Immune Network): modelo de rede imunológica artificial com aplicações a problemas de análise de dados, como compressão de informação e clusterização.

Estes algoritmos são provas concretas de como idéias extraídas de sistemas naturais podem ser empregadas no desenvolvimento de novas ferramentas computacionais para a solução de problemas de engenharia e para a implementação de mecanismos de aprendizagem de máquina. Eles foram aplicados a diversas classes de problemas, destacando otimização, reconhecimento de padrões e análise de dados. Os desempenhos apresentados foram comparados com outras abordagens de inteligência computacional já consolidadas, como as baseadas em RNAs e nos algoritmos de computação evolutiva.

1.3. Notação

Para facilitar a leitura e interpretação do conteúdo proposto e apresentado no texto desta tese, foi adotada a seguinte notação genérica:

- Expressões (ou caracteres) em **negrito** representam matrizes: ex. **Ab**, **Ag**;
- Expressões (ou caracteres) em **negrito e itálico** correspondem a vetores: ex. ***Ab_i***, ***Ag_j***, ***f***;

- Expressões (ou caracteres) em *itálico* são variáveis: ex. n, m, d, x, y ;
- Subíndices entre chaves indicam cardinalidade: ex. $\mathbf{Ab}_{\{m\}}$; e
- Expressões (ou caracteres) em *itálico com subíndice(s)* correspondem a elementos de um vetor ou matriz: ex. $Ab_{i,j}, m_{i,j}, f_j$.

1.4. Formalização da Tese

Esta tese está formalizada em um texto composto por sete capítulos e um apêndice, complementados por dois relatórios técnicos e uma Home Page (web site), como descrito a seguir.

1.4.1. Estrutura da Tese

Capítulo 1: Introdução

Introduz e discute os principais aspectos que levaram ao desenvolvimento e formalização da engenharia imunológica como uma nova abordagem para o tratamento de problemas complexos de engenharia. O enfoque do trabalho, as motivações, os objetivos, e a estrutura da tese fazem parte desta introdução.

Capítulo 2: Sistemas Imunológicos

Neste capítulo é feita uma revisão sobre o sistema imunológico dos animais vertebrados visando fundamentar os aspectos biológicos necessários ao desenvolvimento e formalização da engenharia imunológica como um novo paradigma de computação. Um maior enfoque é dado aos tópicos imprescindíveis ao desenvolvimento das ferramentas propostas nesta tese embora o conteúdo apresentado ofereça margens ao desenvolvimento de novos algoritmos inspirados no sistema imunológico.

Capítulo 3: Engenharia Imunológica

Este capítulo apresenta uma estrutura formal para o desenvolvimento da engenharia imunológica (EI) como uma nova linha de pesquisa a ser empregada na análise e síntese de ferramentas computacionais para a solução de problemas complexos de engenharia. É discutida a relação entre a engenharia imunológica e os sistemas imunológicos artificiais (SIA) e é feita uma revisão bibliográfica de grande parte da literatura atual em SIA.

Capítulo 4: Sistemas Inteligentes e o Sistema Imunológico

Este capítulo tem por objetivo revisar alguns tópicos específicos da teoria de sistemas inteligentes, destacando as redes neurais artificiais e os algoritmos evolutivos. O objetivo é o de salientar similaridades e diferenças entre estas teorias, o sistema imunológico (Capítulo 2) e a engenharia imunológica (Capítulo 3).

Capítulo 5: Ferramentas de Engenharia Imunológica

Este capítulo concretiza a proposta de engenharia imunológica através da proposição de quatro ferramentas computacionais desenvolvidas com base nos princípios imunológicos introduzidos no Capítulo 2 e no paradigma de engenharia imunológica proposto no Capítulo 3. Dentre os princípios fundamentais do sistema imunológico biológico utilizados, destacam-se o princípio da seleção clonal e a teoria da rede imunológica. A formalização teórica e o desenvolvimento dos algoritmos SAND, CLONALG, ABNET e aiNet são apresentados. Algumas das ferramentas propostas constituem estratégias híbridas que utilizam além de conceitos imunológicos, noções de outras técnicas computacionais como as redes neurais artificiais competitivas e os algoritmos evolutivos apresentados no Capítulo 4.

Capítulo 6: Aplicações em Engenharia

Este capítulo apresenta os resultados computacionais da aplicação das quatro ferramentas de engenharia imunológica propostas no Capítulo 5, SAND, CLONALG, ABNET e aiNet, a diversos problemas de engenharia, enfocando as áreas de aprendizado de máquina, reconhecimento de padrões, aproximação de funções, clusterização e otimização. O desempenho das ferramentas propostas é comparado com o de diversas outras estratégias de sistemas inteligentes, como aquelas baseadas em RNAs e algoritmos evolutivos.

Capítulo 7: Conclusões e Perspectivas Futuras

As diversas implicações resultantes da proposição da engenharia imunológica como um novo paradigma computacional são discutidas neste capítulo. Como complemento, uma discussão sobre a tendência da área de sistemas imunológicos artificiais e a proposta de possíveis extensões deste trabalho e novas direções a serem tomadas fazem parte deste capítulo.

Apêndice: Glossário de Biologia

Este apêndice contém uma vasta lista com o significado de grande parte da terminologia biológica utilizada nesta tese visando ajudar o leitor no processo de compreensão dos conceitos empregados.

1.4.2. Trabalhos Complementares

Relatório Técnico – Parte I

RT DCA 01/99 – Departamento de Engenharia de Computação e Automação Industrial.

“Artificial Immune Systems and Their Applications – Part I: Basic Theory and Applications”, p. 95.

<ftp://ftp.dca.fee.unicamp.br/pub/docs/techrep/1999/DCA99-001.pdf>

<ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/lnunes/rtdca0199.pdf>

Neste relatório, após uma breve introdução sobre o sistema imunológico biológico dos animais vertebrados, são discutidas as principais estratégias empregadas pelo sistema

imunológico na solução de problemas, e é introduzido o conceito de engenharia imunológica. A engenharia imunológica faz uso dos respectivos conceitos biológicos objetivando criar ferramentas computacionais para a determinação de soluções para problemas complexos de engenharia, partindo de informações contidas nos próprios problemas. O texto é concluído com o desenvolvimento de diversos algoritmos de engenharia imunológica. Estes algoritmos são amplamente discutidos e exemplos de suas aplicações potenciais à diversos tipos de problemas são apresentados.

Relatório Técnico – Parte II

RT DCA 02/00 – Departamento de Engenharia de Computação e Automação Industrial.
“Artificial Immune Systems and Their Applications – Part II: A Survey of Applications”, p. 65.

<ftp://ftp.dca.fee.unicamp.br/pub/docs/techrep/2000/DCA00-002.pdf>

<ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/lnunes/rtdca0100.pdf>

Nesta segunda parte do relatório técnico, é feita uma breve descrição de diversas aplicações que utilizam metáforas ou mecanismos imunológicos para resolver problemas que são usualmente abordados por técnicas de sistemas inteligentes, como redes neurais artificiais, inteligência artificial, lógica nebulosa e computação evolutiva. São reunidos diversos trabalhos envolvendo aplicações em engenharia dos chamados sistemas imunológicos artificiais. Além disso são listadas e destacadas as principais características e aplicações em potencial destas ferramentas. Os métodos estão distribuídos por assunto e/ou aplicação, respeitando a terminologia original e o enfoque adotado pelos autores.

Web Site Sobre Engenharia Imunológica

<http://www.dca.fee.unicamp.br/~lnunes/immune.html>

Este site contém as principais contribuições do autor para a área de sistemas imunológicos artificiais (SIA) e para a engenharia imunológica. Estão também disponíveis links para web sites de grupos de pesquisa em SIA e diversas páginas pessoais de pesquisadores com expressiva contribuição para a área. Uma versão de um tutorial contendo o código fonte das ferramentas de engenharia imunológica propostas pelo autor durante o desenvolvimento desta tese também está disponível para download, assim como links para outros materiais introdutórios à área de SIA.

CAPÍTULO 2

O SISTEMA IMUNOLÓGICO

Neste capítulo será feita uma revisão sobre o sistema imunológico dos animais vertebrados visando fundamentar os aspectos biológicos necessários ao desenvolvimento e formalização da engenharia imunológica como um novo paradigma de computação. A ênfase é dada a uma visão sistêmica da imunologia, tentando apresentar uma perspectiva de processamento de informação, como arquitetura, mecanismos e princípios imunológicos incorporados. Particular atenção será dada ao princípio da seleção clonal e aos mecanismos de maturação de afinidade, além da teoria da rede imunológica.

“Mente e saúde . . . O homem é o único ser vivo capaz de ser seu próprio agente agressor, através de seus pensamentos, angústias e inseguranças diante da vida” – L. V. Bonamin

2.1. Introdução

A palavra *imunologia* é derivada do Latim *immunis* ou *immunitas* cujo significado é “isento de carga”, sendo que a carga pode referir-se a uma taxa monetária imposta ao cidadão, uma regra ou lei de restrição de direitos e liberdade, ou uma enfermidade. Indivíduos que não sucumbem a uma doença quando infectados são ditos *imunes* e o status de uma resistência específica a uma determinada doença é chamado de *imunidade*.

Definição 2.1: A *imunologia* é o ramo da biologia responsável pelo estudo das reações de defesa que conferem resistência às doenças (Klein, 1990).

Definição 2.2: O sistema que defende o animal contra o ataque constante de *microorganismos* é chamado de *sistema imunológico* (Tizard, 1995).

O sistema imunológico é fundamental para a sobrevivência do animal e, por isso, precisa atuar de forma eficiente. Existe uma grande quantidade de componentes e mecanismos distintos atuando no sistema imunológico. Alguns destes elementos são otimizados para defender contra um único invasor enquanto outros são direcionados contra uma grande variedade de *agentes infecciosos*.

Existe uma redundância considerável no sistema imunológico, de forma que vários mecanismos de defesa sejam ativados contra um único invasor. Sob o ponto de vista tanto biológico quanto de engenharia, a presença de mecanismos de *aprendizagem* e *memória* são características fundamentais do sistema imunológico. Ele possui a capacidade de extrair

informações dos agentes infecciosos e disponibilizá-las para uso futuro em casos de novas infecções pelos mesmos agentes ou agentes similares.

Este capítulo visa introduzir não apenas aqueles conceitos do sistema imunológico utilizados no desenvolvimento da engenharia imunológica, mas também apresentar uma visão genérica dos principais elementos e processos envolvidos em uma resposta imune, de forma que o leitor adquira subsídios para compreender outras abordagens de sistemas imunológicos artificiais.

O capítulo inicia com um breve histórico da pesquisa em imunologia, seguido de uma visão geral dos princípios e mecanismos de defesa do sistema imune, enfatizando o sistema adaptativo. O capítulo é concluído com um estudo sobre a teoria da rede imunológica.

2.2. Breve Histórico da Pesquisa em Imunologia

A imunologia é uma ciência relativamente nova. Sua origem é atribuída à Edward Jenner, que descobriu, há aproximadamente 200 anos, em 1796, que a vacínia (ou *cowpox*), induzia proteção contra a varíola, uma doença freqüentemente fatal. Jenner batizou seu processo de *vacinação*, uma expressão ainda utilizada para descrever a inoculação de indivíduos sãos, com amostras atenuadas ou mortas de agentes causadores de doenças, objetivando a proteção futura contra a enfermidade (Janeway *et al.*, 2000).

Quando Jenner introduziu a vacinação, ele nada sabia sobre os agentes infecciosos que causam as doenças. No século XIX, Robert Koch provou que as doenças infecciosas eram causadas por *microorganismos patogênicos*, cada qual responsável por uma determinada *enfermidade* ou *patologia*. Atualmente, existem quatro grandes categorias de microorganismos causadores de doença ou *patógenos*: os *vírus*, as *bactérias*, os *fungos* e os *parasitas*.

As descobertas de Koch e outros pesquisadores do século XIX possibilitaram o desenvolvimento da imunologia, estendendo a vacinação para outras doenças. Por volta de 1880, Louis Pasteur projetou com sucesso uma vacina contra a cólera aviária e desenvolveu uma vacina anti-rábica também bem sucedida na inoculação de uma criança mordida por um cão raivoso. Tantos triunfos práticos resultaram na busca pelos mecanismos de proteção imunológica.

Pasteur, embora bem sucedido no desenvolvimento de vacinas, possuía muito pouco conhecimento sobre os mecanismos envolvidos no processo de *imunização*. Ele sugeriu que organismos na vacina eram capazes de remover nutrientes essenciais do corpo e, assim, evitar o crescimento e proliferação dos agentes causadores de doença. Aproximadamente dez anos mais tarde, em 1890, Emil von Behring e Shibasaburo Kitasato demonstraram que a proteção induzida pelos processos de vacinação não se devia a remoção de nutrientes, mas estavam associadas ao surgimento de fatores de proteção no soro dos indivíduos vacinados. Estas substâncias foram denominadas de *anticorpos*, as quais se ligavam especificamente e eram capazes de neutralizar os agentes infecciosos. Emil von Behring recebeu, em 1901, o primeiro prêmio Nobel em medicina pelo seu trabalho sobre a produção de anticorpos (Tizard, 1995).

A primeira grande controvérsia em imunologia surgiu quando Elie Metchnikoff demonstrou em 1882, primeiro em animais invertebrados e depois nos mamíferos, que algumas células eram capazes de “comer” microorganismos. Estas células foram denominadas de *fagócitos*, e ele propôs que elas compunham o principal mecanismo de defesa contra estes microorganismos. Ele sugeriu que os anticorpos apresentavam pouca importância no sistema imunológico. O conflito quanto à relevância dos fagócitos e anticorpos foi resolvido em 1904 quando Almroth Wright e Joseph Denys demonstraram que os anticorpos eram capazes de se ligar a bactérias e promover a sua destruição pelos fagócitos (Tizard, 1995).

Ainda na última década do século XIX, Paul Ehrlich formulou uma teoria denominada de *teoria da cadeia lateral* (*side-chain theory*). A principal premissa desta teoria era a de que a superfície dos *glóbulos brancos* ou *leucócitos* (células mediadoras do sistema imunológico) está coberta com diversas *cadeias laterais*, ou *receptores*, que formam ligações químicas com os *antígenos* encontrados (Cziko, 1995). De forma ampla, *antígenos* correspondem a quaisquer moléculas capazes de serem reconhecidas pelo sistema imune (URL 1; Dreher, 1995; Timmis, 2000; Janeway *et al.*, 2001). Dado qualquer antígeno, pelo menos um destes receptores seria capaz de reconhecer e se ligar a ele. A informação essencial para a produção de todos os anticorpos necessários seria *providenciada* pelos genes do animal. Por esta razão, esta teoria também é conhecida como *germinal* (ou *providencial*), referindo-se ao conjunto total de genes (*genoma*) que é transmitido de um organismo ou par de organismos para seus descendentes. Ele também verificou um crescimento explosivo na produção de anticorpos após a exposição a um dado antígeno e desenvolveu uma técnica para estimar a quantidade de anticorpos no sangue. Em sua teoria, o contato com um dado antígeno seria responsável por *selecionar* e *estimular* uma célula a sintetizar aqueles receptores particulares, que seriam posteriormente secretados para a corrente sanguínea sob a forma de anticorpos. Estas características de seleção e estimulação celulares também permitem a caracterização da teoria de Ehrlich como *seletivista*, uma vez que os antígenos seriam os responsáveis pela seleção de células pré-existentes cujos receptores apresentam uma alta *afinidade* ao estímulo antigênico (Silverstein, 1985). A *afinidade* corresponde à força de ligação entre moléculas como, por exemplo, um antígeno e um anticorpo. O prêmio Nobel de 1908 foi dividido por Ehrlich e Metchnikoff.

No período entre 1910 e 1930, experimentos desenvolvidos por Obermayer, Pick e principalmente por Jules Bordet e Karl Landsteiner com *haptens artificiais* (antígenos inexistentes na natureza, ou seja, artificialmente sintetizados), levaram ao abandono, por mais de meio século, da *teoria seletivista* de Ehrlich. Jules Bordet recebeu o prêmio Nobel em 1919 pelo descobrimento de um conjunto de proteínas que atuam juntas no ataque a formas extracelulares de agentes patogênicos. Este conjunto de proteínas foi denominado de *sistema complemento*, ou simplesmente *complemento*. Em 1930, Karl Landsteiner recebeu o prêmio Nobel pela identificação dos diferentes tipos sanguíneos, resultando no sucesso dos procedimentos de transfusão de sangue.

Entre 1914 e 1955 predominava o ponto de vista de que era inconcebível que qualquer teoria seletivista sobre a formação de anticorpos estivesse correta (Piattelli-Palmarini, 1986). As propostas teóricas originadas no período de 1930 a 1950 foram principalmente de

caráter sub-celular. As atenções se concentraram na biosíntese de moléculas de anticorpos produzidas pelas células. A conclusão foi de que o antígeno deveria trazer para a célula as informações referentes à estrutura complementar da molécula de anticorpo, introduzindo uma teoria chamada de *instrucionista (template instruction theory)*. Os primeiros trabalhos conhecidos na linha instrucionista foram executados por Breinl e Haurowitz, e posteriormente desenvolvidos e defendidos pelo ganhador do prêmio Nobel Linus Pauling (Jerne, 1974a; Cziko, 1995). Pauling postulou que todos os anticorpos possuem a mesma seqüência de *aminoácidos*, mas que sua conformação tridimensional seria determinada durante a síntese por contato direto com o antígeno, que serviria como um *padrão* ou *molde (template)*.

As teorias seletivistas da formação de anticorpos foram reavivadas por Niels K. Jerne logo em seguida, nos anos 50. Jerne assumiu que uma população diversa de anticorpos naturais surge durante o desenvolvimento, mesmo na ausência de antígenos. O antígeno se combinaria através da seleção de anticorpos circulantes contendo estruturas complementares a este antígeno. A qualidade de uma resposta a um dado antígeno dependeria da concentração dos anticorpos circulantes e poderia ser melhorada pela exposição prévia ao antígeno.

Restou a McFarlane Burnet (e também a Talmage) assumir que cada célula produz e expressa em sua superfície um único tipo de molécula de anticorpo, e que o *evento seletivo* é o estímulo dado pelo antígeno, sendo que aquelas células que produzem anticorpos complementares a ele irão se proliferar (*expansão clonal*) e secretar anticorpos. Nesta *teoria da seleção* (ou *expansão clonal*, Burnet (1959) assumiu que a diversidade dos anticorpos era gerada por processos aleatórios, como *mutação somática*, durante o período pré-natal, de forma que logo após o nascimento, o animal teria um *repertório fixo de anticorpos*. Além disso, ele postulou a morte de qualquer célula portadora de anticorpos capazes de reconhecer *antígenos próprios*, denominadas *células auto-reativas*, durante este período de geração de diversidade (Bell & Perelson, 1978). Peter Medawar confirmou experimentalmente a teoria da seleção clonal proposta por Burnet. Estes estudos sobre como o organismo reage aos agentes externos e apresentam *tolerância* (ausência de reação) às células do próprio organismo, resultaram em mais um prêmio Nobel na imunologia para Medawar e Burnet.

Em 1971, Jerne argumentou que a eliminação de células auto-reativas fornecia um mecanismo poderoso de *seleção negativa* favorecendo a diversidade celular frente à possibilidade de reconhecer antígenos muito parecidos com o próprio. Considerações sobre como os antígenos próprios, particularmente aqueles das moléculas de anticorpo, denominados de *idiotopos*, poderiam afetar a geração de diversidade e a regulação das respostas imunes, levaram Jerne a propor sua *teoria da rede imunológica* (Jerne, 1973, 1974a,b, 1984, 1985), que lhe rendeu um prêmio Nobel em 1984.

Mais recentemente, Susumo Tonegawa (1983) formalizou seu estudo sobre *estrutura e geração da diversidade das moléculas de anticorpo*, propondo que no genoma de uma célula germinal está contida, em múltiplos segmentos gênicos ao longo de um cromossomo, a informação genética para codificar uma molécula de anticorpo. Ele demonstrou como os anticorpos são gerados e como eles se combinam a uma grande variedade de moléculas.

Dessa forma, ele contribuiu para resolver um importante dilema da imunologia: ‘Como, partindo de um genoma finito, é possível sintetizar uma diversidade de receptores capaz de reconhecer uma variedade praticamente infinita de agentes patogênicos?’. Este trabalho garantiu mais um prêmio Nobel para a imunologia no ano de 1987.

Nos últimos anos, grande parte dos estudos em imunologia tem se concentrado nos problemas da *apoptose celular*, *apresentação de antígenos*, *citocinas*, *regulação e maturação da resposta imune*, *memória imunológica*, *doenças auto-imunes*, *vacinas de DNA* e *sinalização intra- e intercelular*.

Dentre os principais desafios da imunologia para o século XXI, Abbas & Janeway (2000) destacam o aumento da compreensão dos mecanismos de controle da *resposta imune adaptativa*, de forma que seja possível convertê-la de um estado agressivo para um estado benigno em situações como resposta a *alérgenos*, antígenos próprios (doenças auto-imunes) e *tecidos transplantados*. Além disso, um aumento na eficácia das respostas a certos vírus como do HIV, malária, tuberculose e a alguns tumores também poderia ser alcançado através da manipulação da *resposta imune adaptativa*.

A Tabela 2.1 resume as principais tendências e seus respectivos pesquisadores no campo da imunologia até o início dos anos 90.

Tabela 2.1. Períodos da história da imunologia (adaptado de Jerne, 1974a).

Tendências	Período	Pioneiros	Noções
Aplicação	1796-1870	Jenner E Koch R	Imunização Patologia
	1870-1890	Pasteur L Metchnikoff E	Imunização Fagocitose
Descrição	1890-1910	von Behring E & Kitasato S Ehrlich P	Anticorpos Receptores celulares
	1910-1930	Bordet J Landsteiner K	Especificidade/Complemento Haptenos/Tipos sanguíneos
Mecanismos (Sistema)	1930-1950	Breml & Haurowitz Pauling L	Síntese de anticorpos Instrucionismo
	1950-1980	Burnet J & Talmage Jerne N	Seleção clonal Rede e interação celular
Molecular	1980-1990	Tonegawa S	Estrutura e diversidade de receptores de antígenos

2.3. Princípios Fundamentais e Elementos Constituintes

O sistema imunológico representa a principal barreira do hospedeiro contra as infecções, e tem a capacidade de realizar uma resposta rápida e efetiva contra os patógenos invasores. Além disso, pode elaborar um outro tipo de resposta igualmente eficaz, porém mais lenta e duradoura. Estes dois tipos de respostas são efetuadas pelos *sistemas imune inato e adaptativo*, respectivamente.

Ambos os sistemas (inato e adaptativo) dependem da atividade das *células brancas*, ou *leucócitos*. A imunidade inata é mediada principalmente pelos *macrófagos* e *granulócitos*, enquanto a imunidade adaptativa é mediada pelos *linfócitos*, como ilustrado na Figura 2.1.

As células do *sistema imune inato* estão imediatamente disponíveis para o combate contra uma ampla variedade de patógenos, sem exigir prévia exposição aos mesmos, e atuam do mesmo modo em todos os indivíduos normais. Os *macrófagos* e *neutrófilos* possuem a capacidade de ingerir e digerir vários microorganismos e partículas antigênicas. O macrófago também possui a habilidade de apresentar antígenos a outras células, sendo portanto denominado de *célula apresentadora de antígeno* (APC – *antigen presenting cells*). Os granulócitos, ou *leucócitos polimorfonucleares*, constituem um grupo de células com núcleos multilobulados contendo *grânulos* citoplasmáticos preenchidos com elementos químicos (*enzimas*), como ilustrado na Figura 2.2. Os neutrófilos são os elementos celulares mais numerosos e importantes da resposta imune inata, e também têm a capacidade de ingerir patógenos. Os *eosinófilos* são importantes principalmente na defesa contra infecções por parasitas, e a função dos *basófilos* ainda não é bem conhecida (Janeway *et al.*, 2000).

Uma *resposta imune específica*, como a produção de anticorpos a um determinado agente infeccioso, é conhecida como uma *resposta imune adaptativa*. Os anticorpos são produzidos pelos linfócitos B (ou células B) em resposta a infecções, e sua presença em um indivíduo reflete as infecções às quais o mesmo já foi exposto. Os linfócitos são capazes de desenvolver uma memória imunológica, ou seja, reconhecer o mesmo estímulo antigênico caso ele entre novamente em contato com o organismo, evitando assim o re-estabelecimento da doença (Sprenst, 1994; Ahmed & Sprenst, 1999). Assim, a resposta imune adaptativa aperfeiçoa-se a cada encontro com um antígeno.

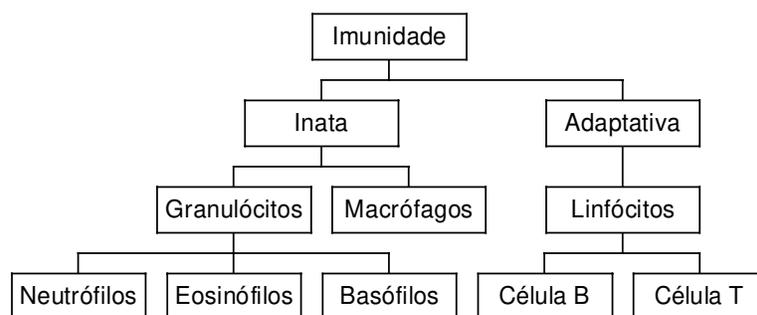


Figura 2.1. Mecanismos de defesa e seus principais mediadores.



Figura 2.2. Grânulos preenchendo os granulócitos ou leucócitos polimorfonucleares.

Os *linfócitos* que medeiam uma resposta imune adaptativa são responsáveis por *reconhecer* e *eliminar* os agentes patogênicos, proporcionando a imunidade duradoura, a qual pode ocorrer após a exposição a uma doença ou vacinação. A grande maioria dos linfócitos encontra-se em estado inativo, e possuirão atividade quando houver algum tipo de interação com um estímulo antigênico, necessário para a ativação e proliferação linfocitária. Existem dois tipos principais de linfócitos: *linfócitos* B (ou *células* B) e *linfócitos* T (ou *células* T), como ilustrado na Figura 2.1. As células B e T expressam, em suas superfícies, receptores de antígeno altamente específicos para um dado determinante antigênico (Seção 2.7).

Enquanto a resposta imune adaptativa resulta na imunidade contra a re-infecção ao mesmo agente infectante, a resposta imune inata permanece constante ao longo da vida de um indivíduo, independente da exposição ao antígeno (Scroferneker & Pohlmann, 1998). Esta é uma importante diferença entre a resposta adaptativa e a resposta inata. Em conjunto, os sistemas inato e adaptativo contribuem para uma defesa notavelmente eficaz, garantindo que, embora passemos nossas vidas cercados por germes potencialmente patogênicos, apresentemos resistência às enfermidades. Devido à grande importância do sistema imune adaptativo, ele será estudado em maiores detalhes na Seção 2.6.

2.4. Mecanismos Básicos de Defesa do Sistema Imunológico

Nosso corpo é protegido por uma grande variedade de células e moléculas que operam em harmonia, sendo que o alvo principal de uma resposta imunológica é o antígeno (Ag). A Figura 2.3 apresenta um esquema simplificado dos principais mecanismos de reconhecimento e ativação do sistema imunológico.

Células apresentadoras de antígeno (APCs) especializadas, como macrófagos, circulam pelo corpo ingerindo e digerindo os patógenos encontrados, fragmentando-os em *peptídeos* antigênicos (Nossal, 1993) (I). Partes destes peptídeos se ligam a moléculas do *complexo de histocompatibilidade principal* (MHC – *major histocompatibility complex*) e são apresentados na superfície celular (II) sob a forma de um *complexo MHC/peptídeo* (Seção 2.7.2). As *células* T possuem receptores de superfície (Figura 2.6(b)) que têm a função de reconhecer diferentes complexos MHC/peptídeo (III). Uma vez *ativados* pelo reconhecimento MHC/peptídeo, as células T se dividem e secretam *linfocinas* (sinais químicos) que mobilizam outros componentes do sistema imunológico (IV). Diferente dos receptores das células T, entretanto, os receptores das células B são capazes de reconhecer partes livres solúveis dos antígenos, sem as moléculas do MHC (V). As *células* B, que também possuem moléculas receptoras de especificidade única em suas superfícies, respondem a estes sinais.

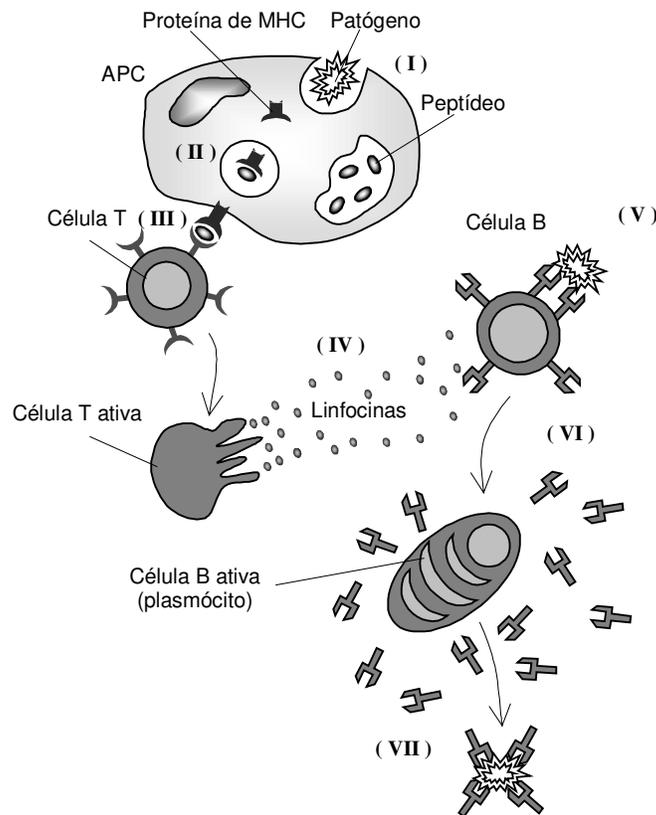


Figura 2.3. Esquema simplificado dos mecanismos de reconhecimento e ativação do sistema imunológico.

Quando ativadas, as células B se dividem e se diferenciam em *plasmócitos*, secretando anticorpos em altas taxas, que são formas solúveis dos seus receptores (VI). A ligação dos anticorpos aos antígenos encontrados faz com que o patógeno seja neutralizado (VII), levando à sua destruição pelas enzimas do *sistema complemento* ou por fagócitos. Algumas células B e T se transformam em *células de memória*, as quais permanecem na circulação garantindo uma resposta rápida e eficaz contra uma futura exposição àquele antígeno.

2.5. Anatomia do Sistema Imunológico

Os tecidos e órgãos que compõem o sistema imunológico estão distribuídos por todo o nosso corpo. São conhecidos como *órgãos linfóides*, uma vez que estão relacionados com a produção, crescimento e desenvolvimento dos *linfócitos*. Nos órgãos linfóides, os linfócitos interagem com diversos tipos de células, seja durante seu processo de maturação, seja durante o início de uma resposta imune adaptativa. Os órgãos linfóides podem ser divididos em *primários* (ou *centrais*), responsáveis pela produção e maturação de linfócitos, e *secundários* (ou *periféricos*) nos quais os linfócitos encontram os estímulos antigênicos, iniciando as respostas adaptativas.

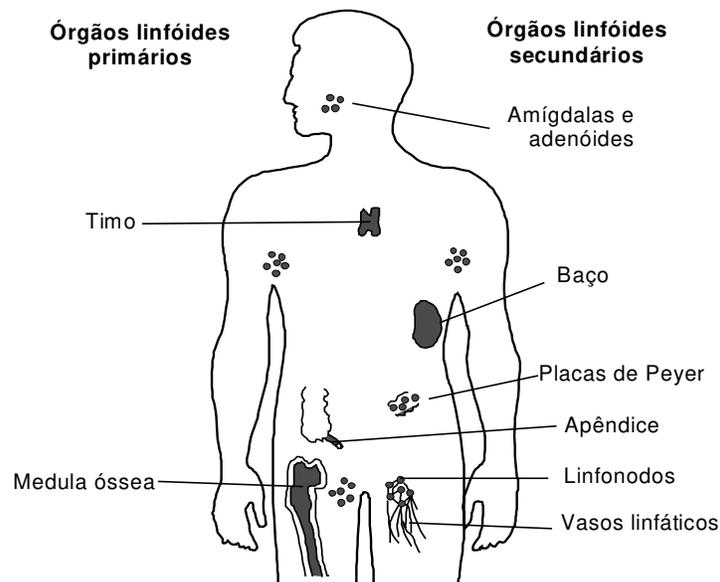


Figura 2.4. Anatomia do sistema imunológico (órgãos linfóides).

Os órgãos linfóides (Figura 2.4) e suas principais funções incluem:

1. Órgãos linfóides primários:

- *Medula óssea*: é o local da *hematopoese*, ou seja, a geração dos elementos celulares do sangue, incluindo as *hemácias*, os monócitos, os leucócitos polimorfonucleares (granulócitos), os linfócitos B e as plaquetas. Nos mamíferos, a medula óssea é também o sítio de desenvolvimento das células B e a fonte de células-tronco que dão origem aos linfócitos T após a migração para o timo;
- *Timo*: órgão localizado na porção superior do tórax onde ocorre o desenvolvimento das células T. Algumas células migram para o timo a partir da medula óssea, e lá se multiplicam e amadurecem, transformando-se em células T.

2. Órgãos linfóides secundários:

- *Amígdalas e Adenóides*: constituem grandes agregados de células linfóides organizadas como parte do sistema imune associado a mucosas ou ao intestino;
- *Linfonodos*: atuam como regiões de convergência de um extenso sistema de vasos que coletam o fluido extracelular dos tecidos, fazendo-o retornar para o sangue. Este fluido celular é produzido continuamente por filtração do sangue e é denominado de *linfa*. É também o ambiente onde ocorre a resposta imune adaptativa;
- *Apêndice e Placas de Peyer*: linfonodos especializados contendo células imunológicas destinadas a proteção do sistema gastrointestinal;
- *Baço*: é o maior órgão linfóide secundário. É também o único órgão linfóide entreposto na corrente sanguínea constituindo-se, portanto, no local onde os linfócitos combatem os organismos que invadem a corrente sanguínea. Contém uma *polpa vermelha* responsável pela remoção de células sanguíneas envelhecidas, e uma *polpa branca* de células linfóides que responde aos antígenos levados ao baço pelo sangue;

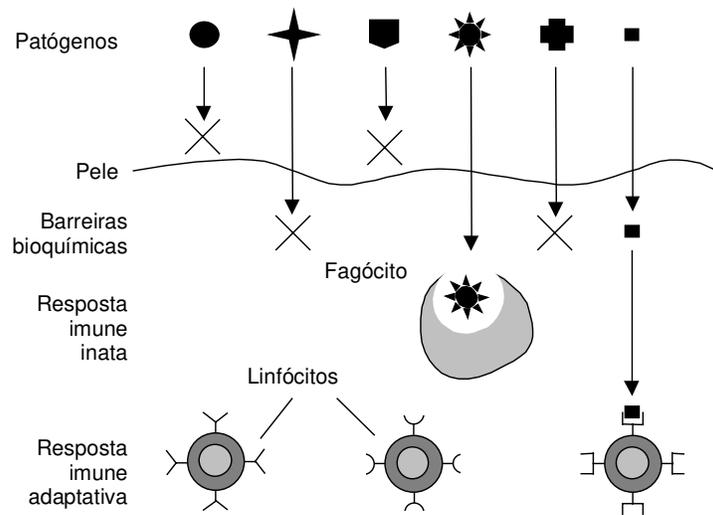


Figura 2.5. Estrutura multicamadas do sistema imunológico.

- *Vasos linfáticos*: rede de canais que transporta a linfa para o sangue e órgãos linfóides. Os vasos aferentes drenam o líquido dos tecidos e carregam as células portadoras dos antígenos dos locais de infecção para os órgãos linfáticos (linfonodos). Nos linfonodos, as células apresentam o antígeno aos linfócitos que estão recirculando, os quais elas ajudam a ativar. Uma vez que estes linfócitos específicos passaram por um processo de proliferação e diferenciação, eles deixam os linfonodos como células efetoras através dos vasos linfáticos eferentes.

O sistema imunológico possui uma arquitetura de múltiplas camadas, com mecanismos de regulação e defesa espalhados em vários níveis (Figura 2.5). As camadas de proteção podem ser divididas como a seguir (Janeway *et al.*, 2000; URL 1; Rensberger, 1996; Hofmeyr, 1997, 2000):

- *Barreiras físicas*: a pele funciona como uma espécie de escudo protetor contra os invasores, sejam estes maléficos ou não. O sistema respiratório também ajuda na manutenção dos antígenos distantes. Seus mecanismos de defesa incluem a apreensão de pequenas partículas nos pêlos e mucosas nasais e a remoção de elementos via tosse e espirros. A pele e as membranas que fazem parte do sistema respiratório e digestivo também contém macrófagos e anticorpos;
- *Barreiras bioquímicas*: fluidos como a saliva, o suor e as lágrimas contém enzimas como a *lisozima*. Os ácidos estomacais eliminam grande parte dos microorganismos ingeridos junto com a comida e a água. O pH e a temperatura corporais podem apresentar condições de vida desfavoráveis para alguns microorganismos invasores;
- *Sistema imune inato*: é a primeira linha de defesa contra muitos microorganismos comuns. Ele é formado por células fagocitárias, como os macrófagos e os neutrófilos (Figura 2.1), além de fatores solúveis como o complemento e algumas enzimas. As células do sistema imune inato desempenham um papel crucial na iniciação e posterior direcionamento das respostas imunes adaptativas, principalmente devido ao fato de que as respostas adaptativas demoram um certo

período de tempo (da ordem de dias) para exercer seus efeitos. Portanto, a resposta inata apresenta um papel muito importante no controle das infecções durante esse tempo;

- *Sistema imune adaptativo*: Devido à grande importância deste sistema, ele será discutido separadamente a seguir.

2.6. Sistema Imune Adaptativo

Todos os organismos vivos são capazes de apresentar algum tipo de resistência a patógenos, mas a natureza desta resistência difere basicamente de acordo com o tipo de organismo. Tradicionalmente, a imunologia aborda, quase exclusivamente, as reações de defesa dos *vertebrados* (animais contendo ossos) e, em particular, dos mamíferos exemplificados pelos camundongos e humanos (Klein, 1990). Os animais vertebrados desenvolveram um sistema de defesa com a característica principal de ser *preventivo*, ou seja, o sistema imune dos vertebrados é capaz de se prevenir contra qualquer tipo de antígeno que pode ser encontrado (ou sintetizado).

Os linfócitos são as principais células do sistema imune adaptativo, presentes apenas nos animais vertebrados, evoluíram para proporcionar meios de defesa mais versáteis e um maior nível de proteção face às novas infecções pelo mesmo agente, do que os apresentados pelo sistema imune inato. Entretanto, as células do sistema imune inato desempenham um papel crucial no desencadeamento e posterior regulação das respostas imunes adaptativas.

Cada *linfócito virgem* que penetra na corrente circulatória é portador de receptores de antígeno com uma única especificidade. A especificidade destes receptores, contudo, é determinada por um mecanismo de rearranjo gênico especial que atua durante o desenvolvimento linfocitário na medula óssea e no timo, a fim de gerar centenas de diferentes variantes dos genes codificadores das moléculas receptoras. Assim, embora um linfócito individual seja portador de um receptor de especificidade única, a especificidade de cada linfócito é diferente, e os milhões de linfócitos do organismo podem apresentar milhões de especificidades distintas. Os linfócitos sofrem, então, um processo parecido com a *seleção natural* durante a vida do indivíduo (Lederberg, 1988; Klein, 1995): somente aqueles que encontram um antígeno com o qual seu receptor pode interagir serão ativados para proliferar e se diferenciar em células efetoras.

No Capítulo 4, discutiremos brevemente a *teoria da evolução natural* que deu origem a *computação evolutiva*, relacionando a evolução das células imunológicas com a evolução do indivíduo. Este aspecto é particularmente importante para o desenvolvimento e caracterização da engenharia imunológica, a ser proposta no próximo capítulo.

O mecanismo seletivo atualmente aceito foi proposto por McFarlane Burnet (1959) para explicar por que os anticorpos, que podem ser produzidos em resposta a virtualmente qualquer antígeno, são produzidos em cada indivíduo apenas contra aqueles antígenos aos quais ele foi exposto. O autor postulou a existência, no hospedeiro, de muitas células potencialmente produtoras de anticorpos diferentes, cada uma tendo a capacidade de sintetizar um anticorpo de especificidade distinta e exibindo, em sua superfície, o mesmo

tipo de anticorpo ligado à membrana e atuando como receptor de antígeno. Após a ligação do anticorpo de superfície ao antígeno, a célula é ativada para proliferar e produzir uma numerosa prole, conhecida como *clone*. Essas células secretam anticorpos com uma especificidade idêntica à do receptor de superfície. Este princípio recebeu o nome de *teoria da seleção clonal*, e constitui a parte central da imunidade adaptativa. Suas implicações para o processo de aprendizagem e memória imunológica são muitas, razão pela qual a seleção clonal será estudada separadamente na Seção 2.8.

2.7. Reconhecimento de Padrões

Do ponto de vista de *reconhecimento de padrões* no sistema imunológico, a característica mais importante das células B e T é que ambas possuem *moléculas receptoras* (*reconhecedoras*) em suas superfícies capazes de reconhecer antígenos. Os receptores das células B e T reconhecem antígenos com características distintas. O receptor da célula B pode interagir com moléculas antigênicas livres em solução, enquanto o receptor das células T reconhece antígenos processados e ligados à uma molécula de superfície denominada de complexo de histocompatibilidade principal (ver Figura 2.3).

O *receptor de antígeno da célula B* (BCR – *B cell receptor*) é o anticorpo ligado à membrana, e que será secretado quando a célula for ativada. As principais funções da célula B, cujo nome provem do fato de que sua maturação ocorre na medula óssea (*bone marrow*), incluem a produção e secreção de anticorpos como resposta aos agentes patogênicos. Cada célula B produz um tipo específico de anticorpo, capaz de reconhecer e se ligar a uma determinada proteína. A secreção e ligação de anticorpos a antígenos constituem formas de sinalizar outras células para que estas façam a ingestão, processamento e/ou remoção da substância ligada. A Figura 2.6(a) ilustra uma célula B com destaque para a molécula de anticorpo. O reconhecimento imunológico ocorre em nível molecular e é baseado na *complementaridade* entre a região de ligação do receptor e uma porção do antígeno chamada *epítipo*. Enquanto os anticorpos possuem um único tipo de receptor, os antígenos podem possuir múltiplos epítipos, significando que um único antígeno pode ser reconhecido por diferentes moléculas de anticorpo, como ilustrado na Figura 2.6(b). Devido à sua grande importância tanto para o sistema biológico quanto para os sistemas artificiais, a Seção 2.7.1 será dedicada exclusivamente à molécula de anticorpo.

A célula T é assim chamada devido ao fato de que sua maturação ocorre no *timo* (Dreher, 1995). Suas funções incluem a regulação das ações de outras células e o ataque direto às células infectadas do organismo hospedeiro. Os linfócitos T podem ser divididos em dois grandes subgrupos: células T auxiliares (T_H – *T helper*) e células T citotóxicas (T_K – *T killer* ou *T citotóxic*). O *receptor de antígeno da célula T* (TCR – *T cell receptor*) possui algumas diferenças estruturais em relação aos receptores das células B (Figura 2.7(a)), como será discutido na Seção 2.7.2. Enquanto o BCR é capaz de reconhecer antígenos livres, o TCR reconhece apenas antígenos processados (fragmentados sob a forma de peptídeos) e ligados à uma molécula de superfície chamada complexo de histocompatibilidade principal, ou MHC. A Figura 2.7(b) ilustra o TCR e sua ligação ao complexo MHC/peptídeo.

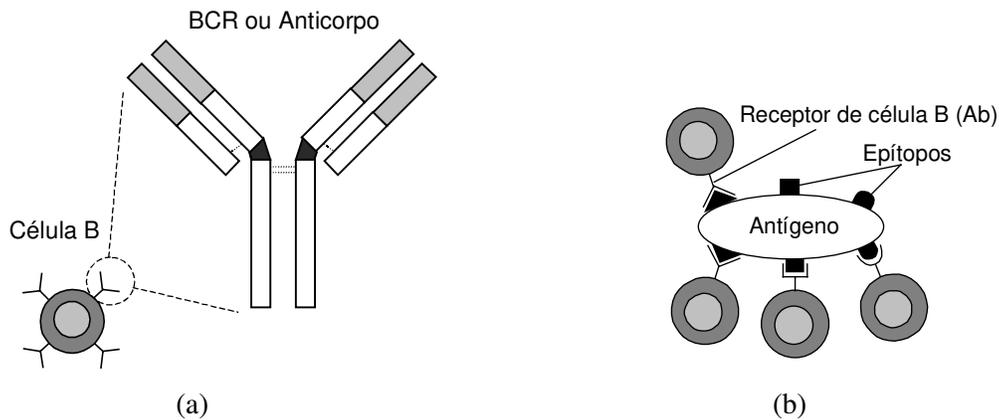


Figura 2.6. Linfócito B. (a) Célula B com destaque para a molécula de anticorpo em sua superfície (Seção 2.7.1). (b) A porção de um antígeno que é reconhecida por um anticorpo é chamada epítipo. Enquanto os anticorpos são mono-específicos, os antígenos podem apresentar vários epítipos distintos.

Existem duas grandes classes de moléculas de MHC (Germain, 1994, 1995), chamados de MHC classe I (MHC-I) e MHC classe II (MHC-II). As moléculas de classe I são encontradas em todas as células, enquanto as moléculas de classe II são encontradas em um conjunto de células chamadas células apresentadoras de antígeno (APC), como por exemplo as células B, os macrófagos e as *células dendríticas* (Banchereau & Steinman, 1998). As células T citotóxicas reconhecem antígenos ligados a moléculas de MHC-I, permitindo a detecção de células infectadas por vírus. As células T auxiliares interagem com antígenos ligados ao MHC-II. As células apresentadoras de antígeno capturam uma proteína antigênica do ambiente e a processam (ingestão e digestão) de forma a cortá-la em pequenos fragmentos chamados *peptídeos*. Alguns destes peptídeos ou fragmentos peptídicos ligam-se a uma molécula de MHC-II e o complexo MHC/peptídeo é transportado para a superfície da APC, onde ele pode interagir com uma célula T auxiliar (T_H). Este processo está ilustrado na Figura 2.7.

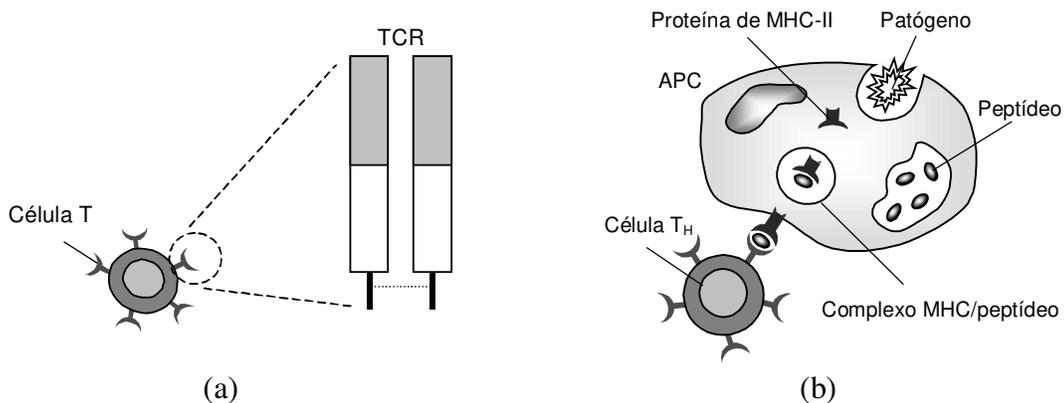


Figura 2.7. Linfócito T. (a) Célula T com destaque para o TCR (Seção 2.7.2). (b) O receptor de célula T (TCR) se liga ao complexo formado por uma molécula de MHC próprio mais um peptídeo antigênico.

Ambos os sistemas apresentam peptídeos de moléculas *próprias* e também de moléculas exógenas ou *não-próprias*. As células T, portanto, precisam discriminar entre o que é *próprio* e o que é *não-próprio*, assunto que será abordado na Seção 2.9.

2.7.1. A Molécula de Anticorpo e a Diversidade Imunológica

O anticorpo, ou imunoglobulina, é uma glicoproteína composta por quatro *cadeias polipeptídicas*: duas *cadeias leves* (L) idênticas e duas *cadeias pesadas* (H) também idênticas (Tonegawa, 1983, 1985; Janeway *et al.*, 2000; Perelson & Weisbuch, 1997), como ilustrado na Figura 2.8. Extensivas análises de cadeias polipeptídicas das moléculas de imunoglobulina revelaram que elas são compostas por uma região aminoterminal altamente variável (*região variável*) e uma região carboxiterminal (*região constante*) com poucos tipos distintos. A região variável, ou região-V, é responsável pelo reconhecimento antigênico. Estas subregiões são usualmente chamadas de *regiões determinadas por complementaridade* (CDR – *complementarity-determining regions*). A região constante, ou região-C, é responsável por uma variedade de funções efetoras, como fixação do complemento e ligação a outros receptores celulares do sistema imune. Portanto, a molécula de anticorpo é bifuncional, podendo ligar-se a um antígeno e exercer uma atividade biológica (efetora).

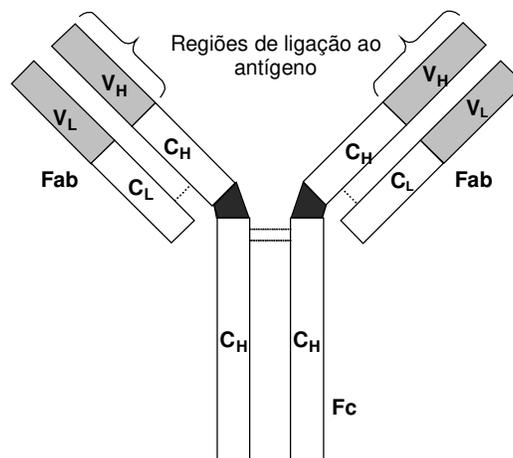


Figura 2.8. Esquema da molécula de anticorpo. A região variável (região-V) é responsável pelo reconhecimento e ligação ao antígeno, e a região constante (região-C) é responsável por uma variedade de funções efetoras, tais como a fixação do complemento. A molécula é simétrica possuindo dois sítios idênticos de ligação ao antígeno (Fab) e um sítio (Fc) que se liga aos receptores em células efetoras, ou interagem com as proteínas do complemento. Os subíndices L e H referem-se as cadeias leves e pesadas, respectivamente.

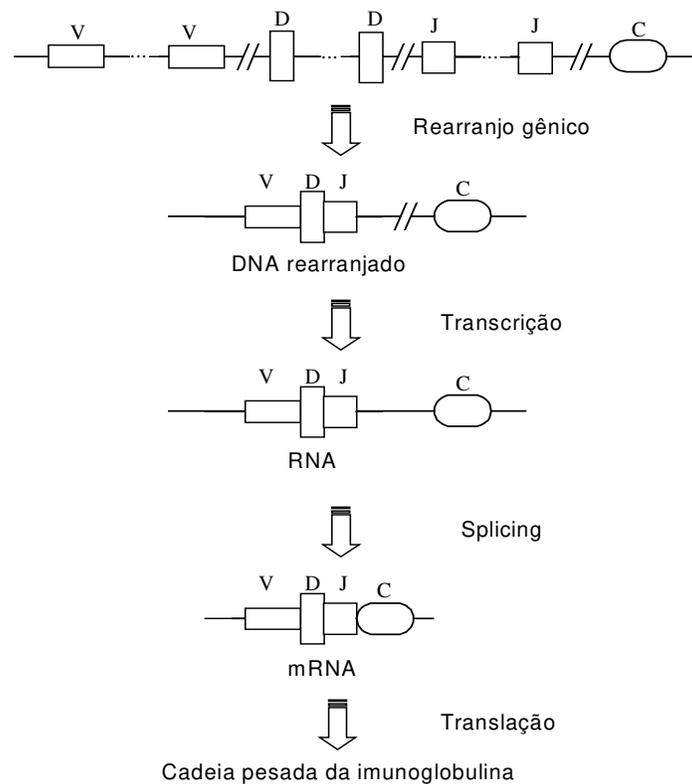


Figura 2.9. O processo de rearranjo gênico que leva à formação da cadeia pesada de uma molécula de anticorpo. Fragmentos gênicos (exatamente um de cada biblioteca) são concatenados de forma ordenada. Cada cadeia recebe em sua extremidade um fragmento constante C. Um processo de *transcrição* une C a V(D)J e o *splicing* do RNA junta as seqüências reunidas da região-V às seqüências que codificam a região-C, resultando na cadeia pesada da molécula de anticorpo (adaptado de Oprea, 1999).

Uma característica importante da molécula de anticorpo (e também do TCR a ser estudado posteriormente) é o fato de que no genoma de cada indivíduo existem múltiplos segmentos gênicos, com seqüências relativamente distintas, que codificam uma parte do receptor, ou seja, existem bibliotecas de fragmentos gênicos. Estes segmentos gênicos devem ser colocados juntos para formar uma molécula completa de imunoglobulina ativa nos linfócitos B. Além disso, mutações somáticas com taxas elevadas (*hipermutação*) podem ser introduzidas em um gene de imunoglobulina. Tanto a recombinação genética quanto a mutação somática contribuem para o crescimento da diversidade da informação genética codificada no genoma.

De forma simplificada, os rearranjos de DNA que reúnem o material genético que codifica as regiões V e C de uma molécula de imunoglobulina unem segmentos separados do DNA da região-V, um dos quais é adjacente ao DNA que codifica a região-C. Na cadeia leve, estes segmentos recebem a denominação de V (variável) e J (junção). A junção de um segmento V com um segmento J cria um segmento contínuo de DNA que codifica toda a região-V da cadeia leve. A região-V da cadeia pesada é codificada por 3 segmentos

gênicos. Além dos segmentos gênicos V e J, um terceiro segmento denominado D (diversidade) é encontrado entre os segmentos V e J da cadeia pesada. A Figura 2.9 ilustra o processo de síntese da cadeia pesada de uma molécula de imunoglobulina.

A presença dos mecanismos de recombinação genética e mutação somática como formas de geração da diversidade populacional dos anticorpos é intrigante (Tonegawa, 1985). Por que dois sistemas distintos teriam evoluído para exercer a mesma função? Os dois mecanismos estão sob um controle estrito durante o desenvolvimento das células B. A recombinação dos segmentos gênicos das imunoglobulinas ocorre primeiro, e está completa no momento em que as células são expostas pela primeira vez ao estímulo antigênico. Isto gera uma população celular com ampla variedade em sua especificidade, permitindo que algumas delas sejam compatíveis com um dado estímulo antigênico. O mecanismo de mutação somática é, então, chamado para operar durante a proliferação (ou *clonagem*) das células selecionadas, onde um *clone* é uma célula ou conjunto de células descendentes de uma mesma célula mãe. Através da troca das bases *nucleotídicas*, a hipermutação somática permite refinar a resposta imunológica ao antígeno reconhecido, criando genes de moléculas de imunoglobulina cujos produtos são capazes de reconhecer com maior afinidade aquele antígeno. A atuação conjunta destes mecanismos de geração e diversificação dos anticorpos faz com que o sistema imunológico seja capaz de produzir uma quantidade quase infinita de receptores celulares distintos partindo de um genoma finito.

A Seção 2.8.2 discute a maturação de afinidade dos anticorpos, ou seja, os processos de hipermutação somática e *edição de receptores* (ou recombinação V(D)J), responsáveis por um aumento da diversidade populacional e pela melhoria na capacidade de reconhecimento antigênico pelos receptores linfocitários.

2.7.2. O Receptor de Célula T (TCR) e a Diversidade Imunológica

Existem quatro cadeias polipeptídicas que podem formar dois tipos de TCR, cada qual contendo duas destas quatro cadeias. Duas cadeias polipeptídicas estão ligadas por uma ponte dissulfídrica, numa estrutura semelhante a um fragmento Fab de imunoglobulina, como ilustrado na Figura 2.10. A diversidade da região-V de um TCR é gerada de forma similar à diversidade das moléculas de imunoglobulina, porém com algumas características distintas (Tizard, 1995).

As quatro cadeias polipeptídicas de um TCR estão codificadas por quatro bibliotecas distintas de genes (α , β , γ e δ) que se assemelham aos segmentos gênicos que codificam as cadeias leve e pesada de uma molécula de imunoglobulina. Todas as quatro bibliotecas contêm segmentos gênicos V, J e C, e as bibliotecas β e δ também contêm segmentos D.

Outra diferença das bibliotecas gênicas das moléculas de imunoglobulina é que cada biblioteca gênica do TCR apresenta dois genes de região-C. Nas bibliotecas α/δ estes dois genes da região-C são funcionalmente e estruturalmente diferentes, de forma que um codifica a região constante α ($C\alpha$) e outro codifica a região constante δ ($C\delta$). Todas as células T_H e T_K rearranjam e expressam genes α e β , enquanto uma pequena subpopulação utiliza genes γ e δ (Tizard, 1995).

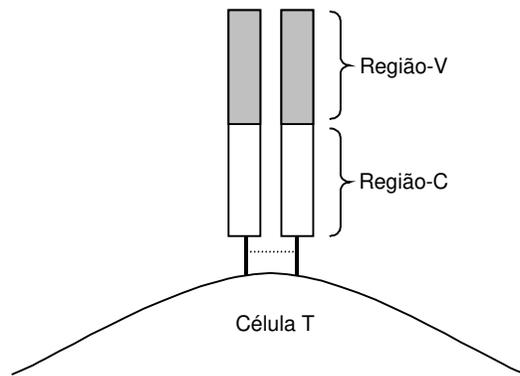


Figura 2.10. Um TCR é formado por duas das quatro cadeias polipeptídicas α , β , γ e δ , sendo que cada cadeia contém uma porção semelhante ao domínio constante e uma porção semelhante ao domínio variável da molécula de imunoglobulina.

Além de apresentarem formas distintas de reconhecimento antigênico, a molécula de imunoglobulina possui dois sítios idênticos de ligação ao antígeno (Figura 2.6) e pode ser secretada, enquanto o TCR possui um único sítio e é sempre uma molécula de superfície celular (Figura 2.7).

A ativação das células T necessita mais do que a ligação do TCR ao complexo MHC/peptídeo. São necessários sinais co-estimulatórios. A ausência ou presença dos sinais co-estimulatórios também é importante para determinar se a interação do TCR com o complexo MHC/peptídeo irá levar à ativação ou à indução de tolerância (Seção 2.9).

2.8. Princípio da Seleção Clonal

Uma vez que cada célula apresenta um padrão (forma) distinto de receptor antigênico, o número de linfócitos que pode se ligar a um determinado antígeno é restrito. A fim de produzir células efetoras específicas em quantidade suficiente para combater uma infecção, um linfócito ativado deve se proliferar antes que sua prole se diferencie em células efetoras.

O *princípio* (ou *teoria*) da *seleção* (ou *expansão*) *clonal* está associado às características básicas de uma resposta imune adaptativa a um estímulo antigênico. Ele estabelece que apenas aquela célula capaz de reconhecer um determinado estímulo antigênico irá se proliferar, sendo, portanto, selecionada em detrimento das outras.

Quando um animal é exposto a um antígeno, uma subpopulação de linfócitos (células B) responde através da produção de anticorpos. Cada célula secreta um único tipo de anticorpo, que é relativamente específico para o antígeno. Através da ligação do antígeno com o receptor da célula B e, dado um *segundo sinal* (ou *sinal co-estimulatório*) de células acessórias como a célula T_H , um antígeno estimula a célula B a se proliferar (dividir) e transformar-se em uma célula terminal capaz de secretar anticorpos em altas taxas. Estas células são chamadas de *plasmócitos*.

As células B, além de se proliferar e diferenciar em plasmócitos, também podem se diferenciar em *células B de memória*, caracterizadas por longos períodos de vida. As

células de memória circulam pelo sangue, vasos linfáticos e tecidos, provavelmente não produzem anticorpos (Perelson *et al.*, 1978), mas quando re-expostas ao mesmo estímulo antigênico começam a se diferenciar em plasmócitos capazes de produzir anticorpos pré-selecionados pelo antígeno específico que estimulou a *resposta primária*. Além disso, linfócitos em desenvolvimento e que se apresentam como potencialmente auto-reativos são removidos do repertório antes de seu amadurecimento (Seção 2.9.2). A Figura 2.11 ilustra o princípio da seleção clonal.

A seleção clonal ocorre tanto no caso dos linfócitos B quanto no caso das células T. As células T não secretam anticorpos, mas são muito importantes na *regulação* da resposta das células B, sendo preeminentes nas respostas imunes mediadas por células (Figura 2.3, Seção 2.4).

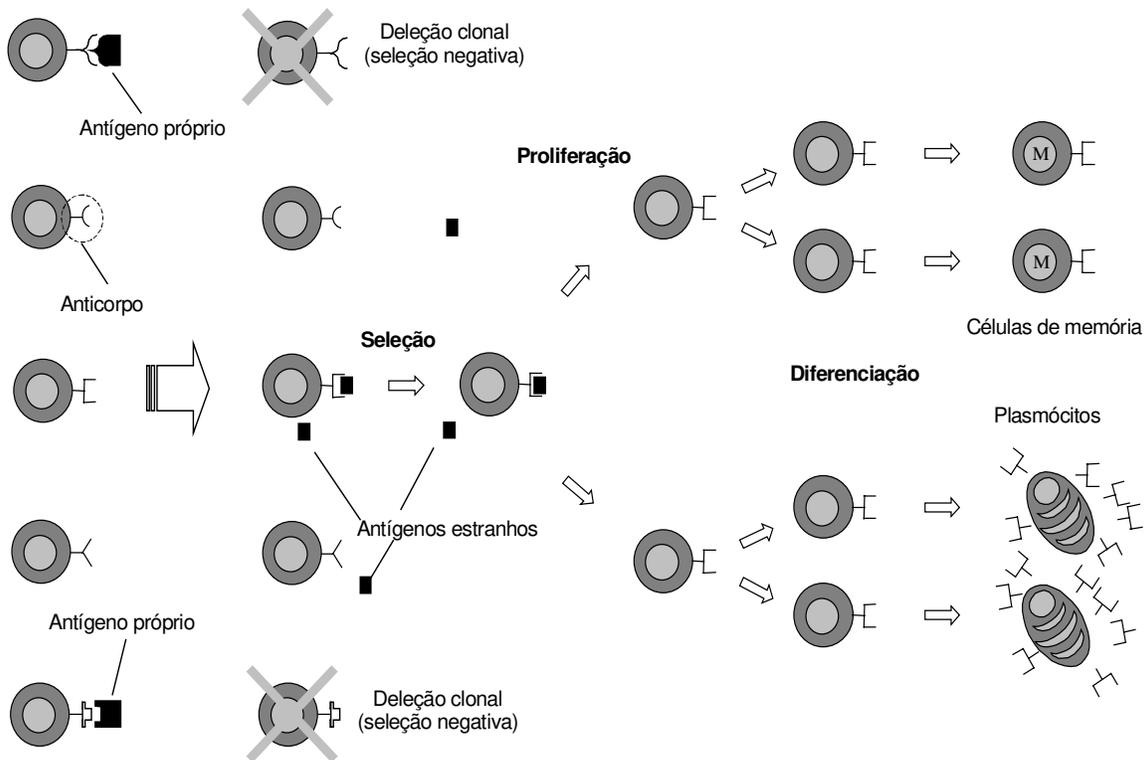


Figura 2.11. Esquema do princípio da seleção clonal. Os formatos distintos dos receptores celulares (anticorpos) correspondem a representações pictóricas distintas. Durante o processo de geração do repertório linfocitário, algumas células são portadoras de receptores que se ligam a antígenos próprios sendo, portanto, eliminadas precocemente no desenvolvimento, antes que sejam capazes de responder, assegurando uma tolerância aos antígenos próprios. Quando o antígeno interage com o receptor num linfócito maduro, tal célula é ativada e começa a se dividir. Origina-se um clone de progênie idêntica, que irá se diferenciar em células efetoras (plasmócitos) e de memória.

Em resumo, as principais características da teoria da seleção clonal são (Burnet, 1978):

- Eliminação ou inativação dos novos linfócitos diferenciados capazes de reagir com padrões antigênicos expressos por elementos do próprio organismo, denominados *antígenos próprios* (Seção 2.9.2). Esta característica assegura uma *tolerância*, ou ausência de resposta, ao próprio;
- A interação de uma molécula estranha com um receptor de linfócito capaz de ligar-se a essa molécula leva à ativação linfocitária;
- Restrição fenotípica de um padrão para cada célula diferenciada e retenção deste padrão pelos descendentes clonais; e
- Geração de variações genéticas aleatórias, através de um mecanismo de hipermutação somática, expressas sob a forma de diversos tipos de anticorpos.

2.8.1. Aprendizagem por Reforço e Memória Imunológica

Para que o sistema imunológico seja capaz de proteger nosso organismo, o reconhecimento antigênico não é suficiente: também é preciso que haja recursos suficientes para montar uma resposta imunológica efetiva contra os agentes patogênicos. Como em situações *presa-predador* típicas, o tamanho da subpopulação de linfócitos, ou seja, o *tamanho do clone* específico para o antígeno, em relação ao tamanho da população de antígenos, é crucial na determinação do resultado da infecção. A *aprendizagem imunológica* envolve o aumento do tamanho da população e afinidade Ag-Ab (antígeno-anticorpo) de linfócitos que reconheceram determinado antígeno. Como o número total de linfócitos do sistema imunológico é regulado, um aumento no tamanho de alguns clones específicos resulta na redução do tamanho de outros clones. Entretanto, o número total de linfócitos não permanece absolutamente constante. Se o sistema imunológico aprendesse apenas pelo crescimento populacional de alguns clones específicos, ele deveria esquecer alguns antígenos aprendidos previamente, aumentar o tamanho global do sistema ou constantemente reduzir a quantidade de células geradas aleatoriamente e responsáveis pela introdução e manutenção da diversidade populacional (Perelson & Weisbuch, 1997).

Durante a evolução do sistema imunológico, um organismo encontra um dado antígeno repetidas vezes. Uma resposta imune adaptativa à exposição inicial de um dado antígeno é composta por um conjunto pequeno de clones de células B, cada um produzindo anticorpos de diferentes especificidades (*afinidades*). A eficiência da resposta adaptativa a encontros secundários poderia ser consideravelmente aumentada através do armazenamento de células produtoras de anticorpos com alta afinidade àquele antígeno, denominadas de *células de memória*, de forma que se tenha um grande clone inicial nos encontros subsequentes (Ada & Nossal, 1987). Ao invés de “partir do começo” toda vez que um dado estímulo antigênico é apresentado, essa estratégia garante que a velocidade e eficácia da resposta imunológica se aperfeiçoe após cada infecção (Perelson *et al.*, 1978, Farmer *et al.*, 1986). Este esquema é característico de uma estratégia de *aprendizagem por reforço* (Sutton & Barto, 1998), onde o sistema está continuamente melhorando a capacidade de executar sua tarefa (Seção 4.2.1.2.3).

Para ilustrar a resposta (memória) imunológica, considere que um antígeno Ag_1 é introduzido em um animal em um tempo 0. Poucos anticorpos específicos a Ag_1 estarão

presentes no soro e, após uma fase de *latência*, os anticorpos contra o antígeno Ag_1 começam a aumentar em concentração e afinidade até um certo nível e, assim que a infecção é eliminada sua concentração começa a cair (*resposta primária*). Quando outro antígeno Ag_2 (diferente de Ag_1) é introduzido, o mesmo padrão de resposta é apresentado, mas para um tipo de anticorpo de especificidade distinta daquela apresentada pelos anticorpos que reconheceram Ag_1 , demonstrando a especificidade da resposta imune adaptativa. Por outro lado, uma característica importante da memória imunológica é sua associatividade: células B adaptadas a um certo tipo de antígeno Ag_1 apresentam uma *resposta secundária* mais rápida e eficiente não somente a Ag_1 , mas também a um antígeno estruturalmente relacionado como, por exemplo, Ag_1' . Este fenômeno é chamado de *reação imunológica cruzada*, ou *resposta reativa cruzada* (*cross-reactive response*) (Hoffmann, 1986; Ada & Nossal, 1987; Sprent, 1994; Smith *et al.*, 1997; Hodgkin, 1998; Mason, 1998). Esta memória associativa é chamada de *capacidade de generalização*, ou simplesmente *generalização*, na literatura de *redes neurais artificiais* (Haykin, 1999). A Figura 2.12 ilustra as respostas primária, secundária e reativa cruzada.

Comparando-se a resposta primária com a secundária, esta última é caracterizada por uma fase de latência menor, e uma maior e mais prolongada taxa de produção de anticorpos. Além disso, uma dose de antígenos substancialmente menor do que a requerida para iniciar uma resposta primária pode desencadear uma resposta secundária.

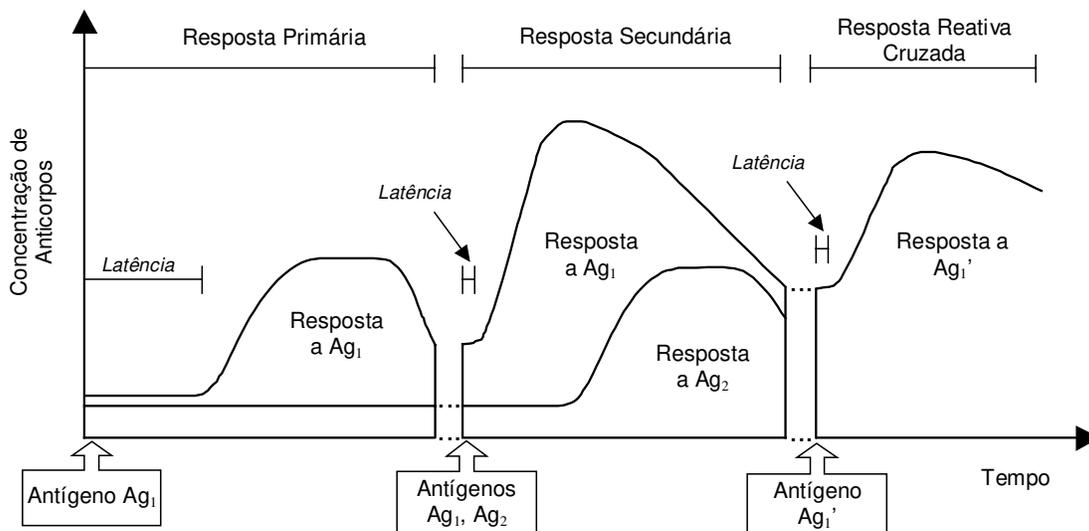


Figura 2.12. Resposta imunológica primária, secundária e reativa cruzada. Depois que um antígeno Ag_1 é visto uma vez (*resposta primária*), encontros subsequentes com o mesmo antígeno, ou um antígeno similar Ag_1' (*reação cruzada*), provocará uma resposta mais rápida e eficiente (*resposta secundária*), não apenas a Ag_1 , mas também a Ag_1' . As respostas primárias a antígenos distintos (Ag_1 e Ag_2) são similares.

Alguns autores sugerem que as células de memória permanecem em um estado de repouso antes de atuar numa resposta secundária (Allen *et al.*, 1987; Coutinho, 1989). Estas células de memória seriam funcionalmente independentes das outras células, sendo a memória uma propriedade do clone, pelo menos no contexto de respostas secundárias. Sprent (1994) sugeriu que células de memória típicas estão semi-ativadas, agindo em respostas pequenas a estímulos persistentes.

Essa discussão sobre o estado funcional das células de memória é importante, pois em duas das ferramentas computacionais a serem propostas nesta tese um conjunto de células de memória é considerado funcionalmente independente dos outros membros do repertório imunológico. Isso se deve ao fato de que, sob uma perspectiva de engenharia e baseado em evidências estatísticas, células com maiores afinidades devem ser preservadas como candidatas de alta qualidade ao reconhecimento de antígenos, e não devem ser substituídas por candidatas de qualidade inferior.

Em resumo, a aprendizagem e memória imunológica podem ser adquiridas através dos seguintes mecanismos (Ahmed & Sprent, 1999; Janeway *et al.*, 2000):

Aprendizagem:

- Exposição repetida ao estímulo antigênico;
- Aumento do tamanho de clones específicos; e
- Maturação de afinidade do receptor.

Memória:

- Existência de linfócitos com períodos prolongados de vida que persistem em um estado de repouso até um segundo encontro antigênico;
- Exposição repetida ao antígeno mesmo na ausência de infecção, ou em infecções crônicas; e
- Reatividade cruzada.

Além da teoria da seleção clonal, a *teoria da rede idiotípica* (ou *teoria da rede imunológica*), a ser discutida na Seção 2.10, apresenta uma abordagem conceitualmente diferente para explicar fenômenos como aprendizagem e memória imunológica.

2.8.2. Maturação de Afinidade

Em uma resposta imunológica auxiliada pelas células T, o repertório de células B ativadas por antígenos é diversificado basicamente por dois mecanismos: *hipermutação somática e edição de receptores* (Tonegawa, 1983, 1985; Berek & Ziegner, 1993; Nussenzweig, 1998; George & Gray, 1999). Somente os descendentes com alta afinidade antigênica são selecionados para fazerem parte do conjunto de células de memória.

Os anticorpos presentes em uma resposta de memória (resposta secundária) possuem, em média, maiores afinidades do que aqueles das respostas primárias. Este fenômeno, que é restrito às respostas dependentes das células T, é chamado de *maturação de afinidade*. Esta maturação, ou amadurecimento, requer que a região de ligação ao antígeno, região-V (Seção 2.7), na resposta secundária, seja estruturalmente diferente (contenha diferentes

seqüências gênicas) daquelas apresentadas nas respostas primárias. Três mecanismos distintos de mutação podem ser observados na região-V dos anticorpos (Allen *et al.*, 1987):

- Mutações pontuais;
- Pequenas deleções; e
- Troca não recíproca de seqüências, seguindo uma conversão genética.

Mutações aleatórias são introduzidas nos genes da região-V durante a expansão clonal dos linfócitos B e ocasionalmente uma dessas mudanças irá provocar um aumento da afinidade do anticorpo ao estímulo antigênico que o selecionou. Estes descendentes com alta afinidade antigênica são então selecionados para fazerem parte do conjunto de memória. Não somente um mecanismo de mutação é utilizado para diversificar o repertório de anticorpos, mas também algum mecanismo deve existir tal que células B raras com receptores mutantes de alta afinidade sejam selecionadas e predominem nas respostas futuras. Devido à natureza aleatória do processo de mutação somática, uma grande parcela dos genes mutantes torna-se não funcional, ou desenvolve receptores *auto-reativos* (Storb, 1998). Estas células com receptores de baixa afinidade, ou as células auto-reativas, devem ser eficientemente eliminadas (ou tornarem-se *anérgicas*) de forma que elas não contribuam significativamente para o conjunto de memória (Berek & Ziegner, 1993; Adams, 1996; Nussenzweig, 1998; George & Gray, 1999).

A forma com que as células B contendo receptores de baixa afinidade, ou auto-reativos, são eliminadas ainda não é muito bem compreendida, mas provavelmente ocorre *apoptose* no centro germinativo (Coutinho, 1989; Nossal, 1992). A apoptose é um processo de morte celular programada, onde uma cascata de eventos intracelulares resultam na condensação e fragmentação do DNA, morte e fagocitose dos resíduos celulares (McConkey *et al.*, 1990; Cohen, 1993; Schwartz & Banchemau, 1996).

A análise do desenvolvimento do repertório de anticorpos expresso pelas células B nos centros germinativos demonstra claramente o papel que a mutação seguida de seleção exerce na maturação da resposta imunológica. Ambos os processos são vitais na maturação da resposta imune. O aumento da afinidade dos anticorpos da resposta primária para a secundária, e assim sucessivamente, demonstra que a maturação da resposta imunológica é um processo contínuo, como ilustrado na Figura 2.13.

Existem três aspectos essenciais de uma resposta imune adaptativa: *diversidade* suficiente para combater um universo antigênico, *distinção próprio/não-próprio*, e *memória imunológica*. Na teoria da seleção clonal originalmente proposta por Burnet (1959), a memória seria fornecida pela expansão do tamanho de um determinado clone específico ao estímulo antigênico, e a mutação aleatória seguida de seleção permitiria o aumento da afinidade deste clone. Além disso, células auto-reativas seriam deletadas durante seu desenvolvimento. Resultados recentes sugerem que o sistema imunológico pratica não somente a seleção clonal de linfócitos, mas também uma *seleção molecular de receptores* (Nussenzweig, 1998). Ao invés da deleção clonal esperada de todas as células auto-reativas, ocasionalmente alguns linfócitos B sofrem uma *edição de receptores*: estas células B deletam seus receptores auto-reativos e desenvolvem receptores completamente novos através da recombinação genética dos elementos componentes das bibliotecas V(D)J.

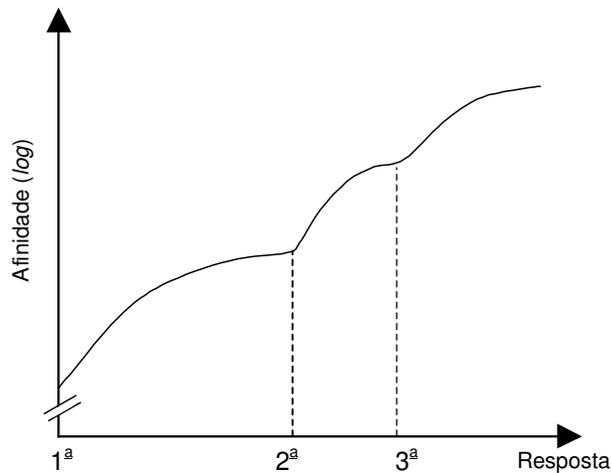


Figura 2.13. Esboço do processo de maturação de afinidade durante sucessivas respostas imunes adaptativas.

Embora a edição e seleção de receptores não façam parte do modelo proposto por Burnet, a teoria de seleção clonal pode certamente acomodar a edição de receptores caso a seleção de receptores ocorra antes da seleção celular. Qualquer clone com alta afinidade desenvolvido por hipermutação somática ou edição de receptores seria preferencialmente expandido, mas algumas células de baixa afinidade também poderiam fazer parte do repertório mantendo, assim, a diversidade populacional.

George & Gray (1999) defenderam a existência do mecanismo de edição de receptores durante a maturação de afinidade como uma estratégia capaz de permitir que um receptor saia de uma região de *ótimo local* em uma *superfície de afinidade*. A Figura 2.14 ilustra esta idéia considerando todas as possíveis regiões de ligação ao antígeno no eixo x , onde as regiões adjacentes estão mais próximas entre si. A afinidade antígeno-anticorpo (Ag-Ab) é mostrada no eixo y . Se um dado anticorpo Ab_1 é selecionado durante uma resposta primária, então pequenas mutações seguidas de seleção permitem que o sistema imunológico explore regiões locais em torno de Ab_1 tomando pequenos passos na direção de um anticorpo com maior afinidade, levando à um *ótimo local* Ab_1^* . Como os mutantes de baixa afinidade são descartados, os anticorpos não podem ‘descer a montanha’. A edição de receptores permite que um anticorpo execute grandes passos através da superfície, caindo em posições onde a afinidade pode ser inferior à anterior (Ab_2). Entretanto, ocasionalmente o salto resultará em um anticorpo numa posição da montanha onde a região de escalada é mais promissora (Ab_3). Partindo deste ponto, mutações somáticas seguidas de seleção permitem a determinação do *ótimo global* das regiões de ligação, Ab_3^* .

Em resumo, pequenas mutações são úteis para a exploração local do espaço de regiões de ligação, enquanto a edição de receptores permite que a resposta imunológica saia de ótimos locais insatisfatórios. Assim, os dois processos executam funções complementares no processo de maturação de afinidade. Somando-se aos mecanismos de mutação somática e edição de receptores, uma fração de novas células é constantemente gerada pela medula óssea e adicionada ao repertório de linfócitos, mantendo a diversidade populacional.

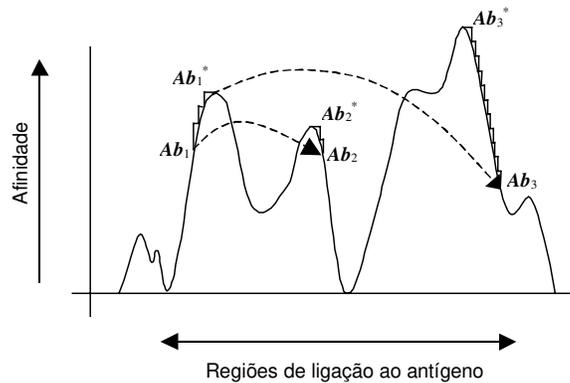


Figura 2.14. Representação esquemática de um espaço de formas para as regiões de ligação ao antígeno. Mutações somáticas seguidas de seleção permitem a determinação de ótimos locais, enquanto a edição de receptores pode introduzir modificações mais expressivas, permitindo que novos candidatos sejam posicionados em regiões do espaço que levarão à obtenção do ótimo global.

2.8.2.1. Regulação do Mecanismo de Hipermutação

O mecanismo de hipermutação opera seletivamente durante a proliferação celular a uma taxa próxima a 1×10^{-3} por par-base das regiões variáveis. Uma vez que o comprimento combinado destas regiões variáveis é de aproximadamente 700 pares-base, em média uma mutação a cada divisão celular é introduzida (Allen *et al.*, 1987; Berek & Ziegner, 1993; Perelson & Weigel, 1998).

Um rápido acúmulo de mutações é necessário para uma rápida maturação da resposta imunológica, mas a maioria das mudanças introduzidas resultarão em anticorpos auto-reactivos ou de baixa afinidade. Se uma célula que acabou de sofrer uma mutação capaz de aumentar sua afinidade antigênica continuar sendo mutada à mesma taxa durante as respostas imunológicas seguintes, então o acúmulo de variações indesejadas pode causar a perda das variações que levaram ao aumento da afinidade. Dessa forma, um curto pico de hipermutação somática, seguido de um intervalo para seleção e expansão clonal devem formar a base do processo de maturação. O mecanismo de seleção deve fornecer uma regulação do processo de hipermutação, que é dependente da afinidade do receptor. Células com receptores de baixa afinidade permanecem sendo mutadas, enquanto células com altas afinidades devem ter suas taxas de mutação controladas, e até inativadas (Kepler & Perelson, 1993a,b).

2.9. Distinção Próprio/Não-Próprio

Para cada um dos dois tipos principais de linfócitos, B e T, é possível considerar três classes de repertórios celulares (Jerne, 1974; Coutinho *et al.*, 1984; de Boer & Perelson, 1991; Perelson & Weisbuch, 1997; Storb, 1998):

- *Repertório potencial*: determinado pela quantidade, estrutura e mecanismos de expressão dos genes que codificam os BCRs ou TCRs, mais os possíveis variantes somáticos derivados deles;
- *Repertório disponível* (ou *expresso*): definido como o conjunto de moléculas que está disponível para ser utilizado como receptores celulares, ou seja, todos aqueles receptores que podem, mas não necessariamente estão sendo utilizados;
- *Repertório ativo*: em uso pelas células ativas e que participam diretamente na resposta imunológica.

O sistema imunológico em sua capacidade de reconhecer antígenos é *completo*. As moléculas de anticorpo e os receptores de linfócitos T podem, em essência, reconhecer qualquer molécula própria ou não-própria, até mesmo aquelas artificialmente sintetizadas. As moléculas de anticorpo possuem *idiotopos imunogênicos*, ou seja, elas possuem epítomos que podem ser reconhecidos por outros anticorpos. O *axioma da completude imunológica* propõe que estes anticorpos serão reconhecidos por outros anticorpos. Este argumento leva ao conceito da *rede imunológica*, que será discutido na próxima seção. Os principais fatores que resultam na completude imunológica dos repertórios linfocitários são: a *diversidade* obtida por mutação, edição e rearranjo genético, a *reatividade cruzada* e a *multi-especificidade* (Inman, 1978; Perelson & Oster, 1979; Tonegawa, 1983; Coutinho *et al.*, 1984; Jerne, 1985; Varela *et al.*, 1988). A reatividade cruzada e a multi-especificidade são algumas das principais razões pelas quais um repertório linfocitário menor que um possível conjunto de antígenos pode ser capaz de reconhecer todos os elementos deste conjunto antigênico (Inman, 1978; Hodgkin, 1998; Mason, 1998). A idéia por trás da multi-especificidade é a de que um anticorpo específico é capaz de reconhecer antígenos com formas relativamente distintas, desde que uma quantidade mínima de interações complementares ocorram entre um epítomo deste antígeno e a região-V da molécula de anticorpo (Inman, 1978; Perelson & Weisbuch, 1997).

A completude do repertório linfocitário representa um paradoxo, pois qualquer molécula pode ser reconhecida, inclusive as próprias, que também passam a ser vistas como antígenos, ou *antígenos próprios*. Para que o sistema imunológico funcione apropriadamente, é preciso que ele seja capaz de distinguir as células e moléculas do próprio organismo, chamadas de *próprio*, e moléculas estranhas, chamadas *não-próprio*, que são, em princípio, indistinguíveis (Perelson & Weisbuch, 1997). Se o sistema imunológico não for capaz de fazer esta distinção, então uma resposta imunológica será desencadeada contra os antígenos próprios, causando *doenças auto-imunes*. A ausência de resposta contra um antígeno próprio é chamada de *tolerância ao próprio* (Tizard, 1995; Kruisbeek, 1995; Schwartz & Banchereau, 1996).

Além da característica aleatória na produção dos receptores linfocitários, um encontro entre um receptor e um antígeno não resulta inevitavelmente na ativação do linfócito, mas pode casualmente provocar a morte ou inativação (*anergia*) celular. Assim, existe um mecanismo de *seleção negativa* que evita que os linfócitos auto-específicos (auto-reativos) se tornem auto-agressivos. Por outro lado, uma pequena porcentagem de células sofre uma *seleção positiva* tornando-se células capazes de montar uma resposta imunológica, chamadas de *células imunocompetentes*, para constituir o *repertório linfocitário disponível*.

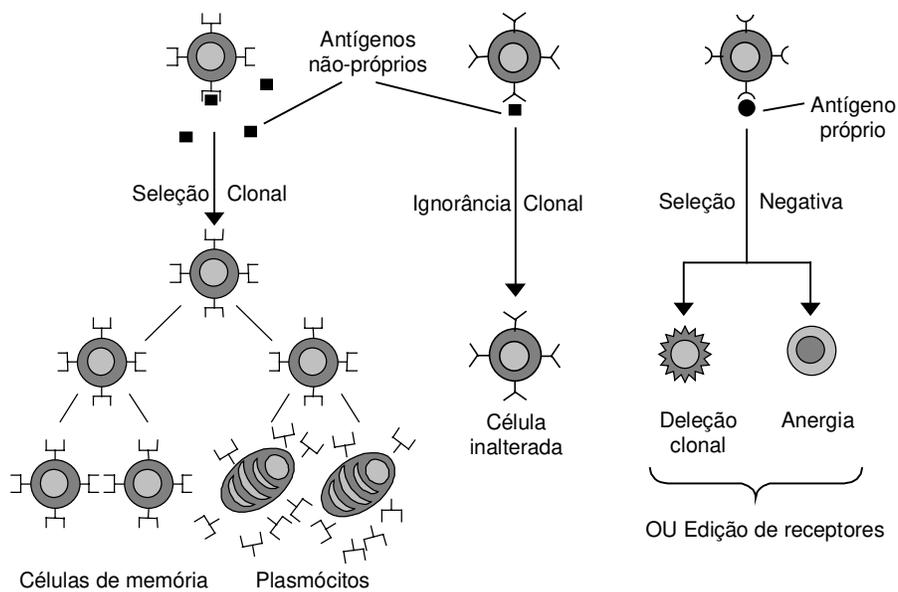


Figura 2.15. Esquema das principais interações entre antígenos e linfócitos.

Alguns dos principais resultados do encontro de um antígeno com um receptor linfocitário estão ilustrados na Figura 2.15, e podem ser listados juntamente com suas principais causas (Parijs & Abbas, 1998; Mani, 1999):

- *Expansão Clonal* (Seção 2.8): reconhecimento antigênico na presença de sinais co-estimulatórios;
- *Seleção Positiva* (Seção 2.9.1): reconhecimento de um complexo MHC-próprio/peptídeo;
- *Seleção Negativa* (Seção 2.9.2): reconhecimento de antígenos próprios nos órgãos linfóides centrais, ou reconhecimento de antígenos próprios na periferia na ausência de sinais co-estimulatórios; e
- *Ignorância Clonal*: todas as circunstâncias nas quais uma célula imunocompetente falha na resposta a um antígeno como, por exemplo, se a concentração antigênica for muito baixa.

2.9.1. Seleção Positiva

A *seleção positiva* dos linfócitos B e T tem como objetivo principal selecionar aqueles linfócitos capazes de operar como células imunocompetentes, ou seja, atuar em uma resposta imune adaptativa. Para isso, é preciso que os linfócitos sejam capazes de reconhecer os seus ligantes (von Boehmer, 1994).

2.9.1.1. Seleção Positiva de Células T

Todas as células T devem reconhecer antígenos associados a moléculas de MHC-próprio que formam os complexos MHC/peptídeo, ou MHC-próprio/peptídeo. Sendo assim, é primeiramente necessário selecionar aquelas células T cujos receptores são capazes de

reconhecer e se ligar às moléculas de MHC-próprio. Este processo é denominado *seleção positiva*. Figurativamente, diz-se que a seleção positiva dos linfócitos T permite que o sistema imunológico seja capaz de reconhecer o próprio.

A seleção positiva visa assegurar que as células T maduras sejam capazes de reconhecer, preferencialmente, antígenos estranhos (não-próprios) no contexto de moléculas do MHC-próprio (Anderson *et al.*, 1999; von Boehmer, 1994). Dentre as principais conseqüências da seleção positiva das células T, que ocorre no timo, destacam-se:

- as células T são selecionadas para posterior desenvolvimento em células imunocompetentes;
- término do processo de rearranjo genético do TCR; e
- alteração do período de vida da célula.

2.9.1.2. Seleção Positiva das Células B

A seleção positiva das células B maduras envolve o resgate da morte celular. Como resultado do reconhecimento e ligação ao antígeno, e auxílio da célula T_H, os linfócitos B em proliferação sofrem hipermutações. Aquelas células filhas mutantes que se ligam mais eficientemente ao antígeno são selecionadas para expansão e, portanto, resgatadas da morte celular.

Comparando-se com as células T, a seleção positiva das células B maduras se assemelha a seleção positiva das células T imaturas. No caso das células T imaturas, elas são resgatadas da morte celular devido ao reconhecimento de uma molécula de MHC-próprio, enquanto as células B maduras são resgatadas da morte celular devido ao reconhecimento de um antígeno não-próprio.

2.9.2. Seleção Negativa

O conceito de seleção negativa, seguindo o reconhecimento de um ligante por um receptor celular, permite o controle dos linfócitos que possuem receptores anti-próprios. A seleção negativa descreve o processo no qual a interação do linfócito com o antígeno resulta na morte (*deleção clonal*) ou anergia (*anergia clonal*) deste linfócito, como ilustrado na Figura 2.15. A célula B ou T é simplesmente eliminada do repertório ou inativada (Nossal, 1994). A seleção negativa pode ser central ou periférica. Os órgãos linfóides primários (centrais) são projetados para não permitir a entrada de antígenos externos e preservar os antígenos próprios, enquanto os órgãos linfóides secundários (periféricos) são projetados para filtrar e concentrar os elementos não-próprios, promovendo reações co-estimulatórias a uma resposta imunológica (Zinkernagel & Kelly, 1997).

2.9.2.1. Seleção Negativa das Células T

O processo de *seleção tímica* (dentro do timo) é baseado nas seguintes considerações. O timo é composto por uma grande quantidade de células apresentadoras de antígenos, incluindo macrófagos, células dendríticas e células epiteliais especializadas. O timo é protegido por uma barreira sangüínea que faz com que estas APCs primariamente apresentem complexos MHC-próprio/peptídeo ao repertório de células T que está sendo

formado. A seleção tímica resulta da interação das células T imaturas (*timócitos*) com os ligantes MHC-próprio em células APCs dentro do timo, resultando na morte (deleção clonal) daquelas células T que forem auto-reativas. O tempo e a extensão deste processo de deleção depende da afinidade da ligação do TCR ao antígeno próprio. Células T que se ligam aos antígenos próprios com alta afinidade são deletadas precocemente e de forma mais efetiva do que aquelas que se ligam com baixas afinidades.

A seleção tímica não é perfeita, e algumas células T auto-reativas podem escapar para a periferia como células imunocompetentes, causando o risco de uma doença auto-imune. O sinal para a ativação de uma célula T requer mais do que a simples ligação do complexo MHC/peptídeo ao TCR: vários processos adjuntos, como a ligação de moléculas de adesão, são necessários para a ativação celular. Na ausência de atividade co-estimulatória, a união do TCR ao complexo MHC/peptídeo pode levar à inativação (anergia) deste linfócito. O sistema imune inato é responsável pela liberação de uma grande quantidade de sinais co-estimulatórios para a resposta imune adaptativa.

2.9.2.2. Seleção Negativa das Células B

A tolerância promovida pelas células T seria insuficiente para a proteção contra doenças auto-imunes. Células B imaturas dentro da medula óssea também são sensíveis a uma indução de tolerância por seleção negativa, caso elas encontrem um antígeno na ausência dos sinais co-estimulatórios liberados principalmente pelas células T. Da mesma forma que as células T, linfócitos B auto-reativos também podem escapar da seleção negativa centralizada. Neste caso, a ativação ou tolerância da célula B será resultado da quantidade, avidéz, tempo e de quais sinais co-estimulatórios estarão presentes. Uma ligação brusca e repentina (característica de antígenos estranhos) do receptor ao antígeno geralmente induz uma resposta clonal, enquanto uma estimulação constante e relativamente fraca leva à inibição e posterior apoptose (Schwartz & Banchemreau, 1996). Este mecanismo de tolerância periférica serve tanto para as células B quanto para as células T e está ilustrado na Figura 2.16.

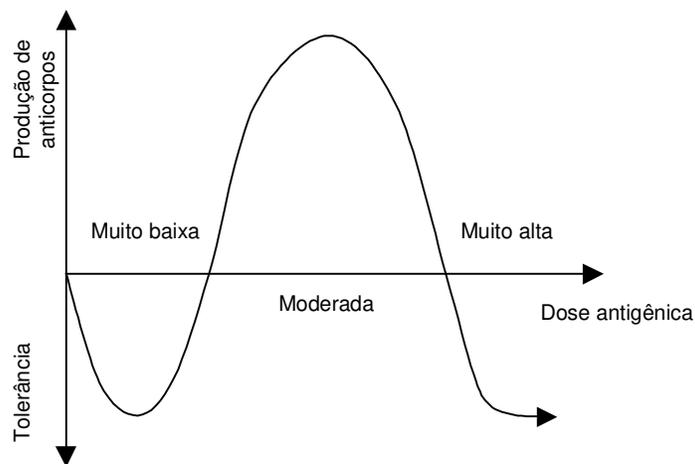


Figura 2.16. Altas e baixas doses antigênicas induzem tolerância, enquanto uma dose moderada resulta em uma produção de anticorpos (adaptado de Tizard, 1995).

2.10. Teoria da Rede Imunológica

Nesta seção, objetivamos apresentar os conceitos básicos da *teoria da rede imunológica*, originalmente proposta por Niels Kaj Jerne (1974a). Alguns modelos específicos de rede imunológica serão apresentados e discutidos na Seção 3.4.3.5 e um novo modelo será proposto na Seção 5.5. Referências sobre conceitos e modelos de rede imunológica podem ser encontrados em Hoffmann (1975); Richter (1975, 1978); Bonna & Kohler (1983); Coutinho *et al.* (1984); Jerne (1984); Langman & Cohn (1986); Farmer *et al.* (1986); Segel & Perelson (1988); Perelson (1988, 1989); Coutinho (1989, 1995); Varela & Stewart (1990a,b); Stewart & Varela (1991); de Boer & Perelson (1991); Calenbuhr *et al.* (1995); Detours *et al.* (1996); Bernardes & dos Santos (1997); Perelson & Weisbuch (1997).

Ao invés de explicar os processos de sinalização celular e a interação de anticorpos, células e seus mecanismos efetores, a teoria de rede inicialmente proposta (Jerne, 1974a) apresentava um novo ponto de vista sobre a atividade linfocitária, a produção de anticorpos, a seleção do repertório pré-imune, a distinção próprio/não-próprio e a tolerância imunológica, a memória e a evolução do sistema imunológico (Varela & Coutinho, 1991). Foi sugerido que *o sistema imunológico é composto por uma rede regulada de células e moléculas que se reconhecem mesmo na ausência de antígenos*. Este ponto de vista estava em conflito com a teoria da seleção clonal (ver Seção 2.8) já existente naquela época, que assumia que o sistema imunológico era composto por um conjunto discreto de clones celulares originalmente em repouso, sendo que a atividade apenas existiria quando um estímulo externo se apresentasse ao organismo.

As Seções 2.7 e 2.8.2 discutiram a aleatoriedade nos processo de geração e maturação das moléculas de anticorpo. Esta aleatoriedade leva à idéia de que novas moléculas (anticorpos) podem ser vistos como elementos invasores ao organismo, sendo tratados como antígenos. Na nova proposta da rede imunológica, o termo região de ligação do anticorpo (região-V) foi mudado para *paratopo*, e determinante antigênico substituído por *epítopo*.

Os epítopos e os paratopos são considerados então como as duas características essenciais para o reconhecimento imunológico. Foi demonstrado experimentalmente que as moléculas de anticorpo também apresentavam epítopos, que poderiam exercer alguma funcionalidade. Um *idiotipo* foi definido como o conjunto de epítopos apresentados pela região-V de um conjunto de moléculas de anticorpo, e um *idiotopo* era cada epítopo idiotípico. Os padrões de idiotopos eram determinados pelas mesmas regiões variáveis das cadeias polipeptídicas dos anticorpos que determinavam os paratopos (ver Figura 2.8). Foi proposta uma definição formal para o sistema imunológico:

Definição 2.3: O sistema imunológico é constituído por uma rede enorme e complexa de paratopos que reconhecem conjuntos de idiotopos, e de idiotopos que são reconhecidos por conjuntos de paratopos, assim, cada elemento pode reconhecer e ser reconhecido (Jerne, 1974a).

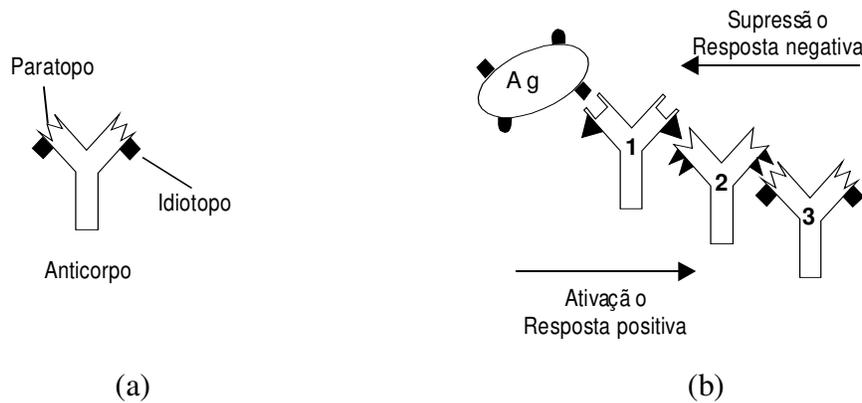


Figura 2.17. Teoria da rede imunológica. (a) Molécula de anticorpo destacando o paratopo e idioto. (b) Respostas positiva e negativa resultantes da interação de um paratopo com um idioto ou um epítipo.

Os linfócitos poderiam responder positivamente ou negativamente ao sinal de reconhecimento. Uma resposta positiva levaria à proliferação e ativação celular, resultando na secreção de anticorpos, enquanto uma resposta negativa levaria à tolerância e supressão. A Figura 2.17 ilustra a molécula de anticorpo com destaque para o idioto, o paratopo, e as respostas positiva e negativa.

O comportamento da rede imunológica, ou *rede idiotípica*, ilustrado na Figura 2.18, pode ser explicado como a seguir (Jerne, 1974a). Quando um dado antígeno invade o nosso organismo, seu epítipo é reconhecido (com vários graus de especificidade) por um conjunto de diferentes paratopos, chamado p_1 . Estes paratopos do conjunto p_1 estão associados a um conjunto de idiotopos i_1 . O símbolo p_1i_1 denota o conjunto total de moléculas de anticorpo e linfócitos B capazes de reconhecer este antígeno. Dentro da teoria da rede imunológica, cada paratopo do conjunto p_1 reconhece um conjunto de idiotopos, e todo o conjunto p_1 reconhece um conjunto ainda maior de idiotopos. Este conjunto i_2 de idiotopos é chamado de *imagem interna* do epítipo (ou antígeno), pois ele é reconhecido pelo mesmo conjunto p_1 que reconhece o antígeno. O conjunto i_2 está associado a um conjunto p_2 de paratopos expresso por moléculas e receptores celulares do conjunto p_2i_2 .

Além disso, cada idioto do conjunto p_1i_1 é reconhecido por um conjunto de paratopos, de forma que todo o conjunto i_1 é reconhecido por um conjunto p_3 ainda maior de paratopos que estão associados aos idiotopos i_3 de anticorpos e linfócitos pertencentes a um conjunto p_3i_3 chamado de conjunto anti-idiotípico. Seguindo este esquema, é possível chegar a conjuntos cada vez maiores de receptores que reconhecem e são reconhecidos por conjuntos previamente definidos na rede. Além do conjunto p_1i_1 , existe um conjunto paralelo p_xi_1 de moléculas e receptores que apresentam idiotopos do conjunto i_1 associados a paratopos que não reconhecem o antígeno dado. As setas indicam um efeito estimulatório quando os idiotopos são reconhecidos por paratopos em receptores celulares e um efeito supressivo quando os paratopos reconhecem os idiotopos em receptores celulares.

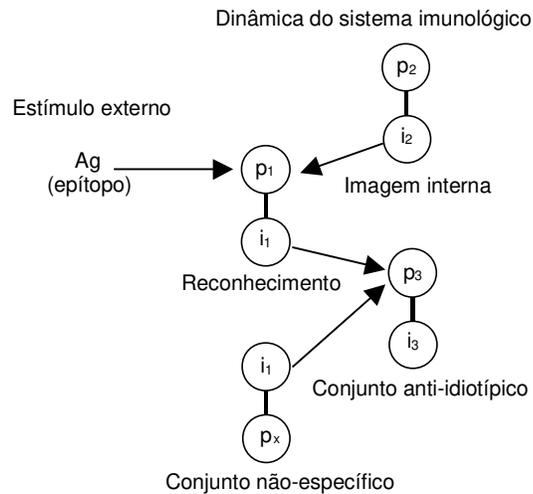


Figura 2.18. Visão detalhada da rede idiotípica (adaptado de Jerne, 1974a).

A teoria de rede é particularmente interessante para o desenvolvimento de ferramentas computacionais, pois ela fornece uma medida aproximada de propriedades emergentes como aprendizagem e memória, tolerância ao próprio, tamanho e diversidade de populações celulares. Estas propriedades não podem ser compreendidas partindo-se de uma análise de componentes isolados e serão exploradas no Capítulo 5 para o desenvolvimento de um modelo de rede imunológica artificial a ser aplicada a problemas de compressão de dados e clusterização.

É possível destacar três características das redes imunológicas (Varela *et al.*, 1988; Bersini & Varela, 1990; Varela & Coutinho, 1991):

- *Estrutura*: responsável pela descrição dos padrões de interconexão entre seus componentes celulares e moleculares, desconsiderando as conseqüências destas interações. Os eventos importantes são os próprios elementos do sistema e não suas interações;
- *Dinâmica*: a dinâmica da rede imunológica trata as interações entre os diversos componentes do sistema: as variações das regiões-V como resultado das interações mútuas e ações recíprocas entre as células e moléculas da rede;
- *Metadinâmica*: uma propriedade única do sistema imunológico, que vai além da dinâmica de rede, é a contínua produção de novos anticorpos. Como mencionado anteriormente, qualquer novo elemento possível, mesmo que sintetizado artificialmente, pode interagir com o sistema imunológico. Existe uma constante renovação do repertório linfocitário e, conseqüentemente, da estrutura da rede via o recrutamento destes novos linfócitos e a morte de elementos não estimulados ou auto-reativos. A metadinâmica representa o potencial que o sistema imunológico possui de introduzir diversidade, garantindo sua capacidade de combater novos antígenos.

Em resumo, a característica central da teoria da rede imunológica é a definição da identidade molecular do indivíduo, pois a tolerância é uma propriedade global que não pode ser reduzida à existência ou à atividade de um clone específico. Ela surge de uma estrutura em forma de rede que se expressa no início da evolução do sistema imunológico e é seguida pela aprendizagem ontogênica da composição molecular do ambiente no qual o sistema imunológico se desenvolve. A organização em rede impõe um padrão de dinâmica para os anticorpos que é distinto das respostas imunológicas a antígenos externos. Estes padrões de dinâmica são perfeitamente compatíveis com a manutenção da memória que não está localizada em células de memória, mas distribuída pela rede.

CAPÍTULO 3

ENGENHARIA IMUNOLÓGICA

Este capítulo propõe uma definição formal para a engenharia imunológica (EI) como uma nova linha de pesquisa derivada dos sistemas imunológicos artificiais, a qual tem o objetivo de analisar e sintetizar ferramentas computacionais para a solução de problemas complexos. É discutida a relação entre a engenharia imunológica e os sistemas imunológicos artificiais (SIA), é feita uma revisão bibliográfica da literatura atual em SIA e é proposta uma estrutura formal para a engenharia imunológica. O capítulo é concluído com alguns exemplos de sistemas imunológicos artificiais e de engenharia imunológica.

“Nossas melhores idéias são geralmente aquelas que unem dois mundos distintos” – M. Minsky

3.1. Introdução

Atualmente, grande parte das soluções tecnológicas existentes são baseadas em um conjunto estrito de regras, ou planos, que especificam uma seqüência detalhada de passos a serem seguidos até se chegar a uma solução. O comportamento global do sistema é usualmente simplificado para que ele possa ser previsto e estudado. Em outras palavras, um dos problemas de *engenharia* consiste em *projetar* um dispositivo básico para executar uma tarefa específica, sendo que as abordagens tradicionais de projeto oferecem procedimentos sistemáticos, seqüenciais e iterativos. Com o incremento gradual de complexidade que vem sendo verificado junto aos problemas de engenharia, estas abordagens não são capazes de fornecer o tratamento mais eficiente, demandando o surgimento de novos paradigmas.

A capacidade de certos sistemas naturais, ou biológicos, vai muito além de qualquer estratégia tecnológica existente na atualidade. Sendo assim, elucidar e aplicar um conjunto geral de princípios que governam o comportamento destes sistemas naturais pode levar ao desenvolvimento de *novas formas de engenharia* (Wolfram, 1986). A natureza fornece muitos exemplos de sistemas compostos por elementos simples, nos quais uma *interação cooperativa e competitiva* destes elementos resulta num comportamento global complexo como, por exemplo, colônias de insetos, o próprio sistema imunológico e o sistema nervoso central (ver ilustração na Figura 3.1). Uma das principais características destes sistemas é a *robustez*, expressa sob a forma de uma tolerância a pequenas perturbações em componentes individuais. Esta robustez engloba o *princípio da distributividade*, em que componentes (*agentes*) individuais do sistema contribuem de forma infinitesimal para o desempenho global.

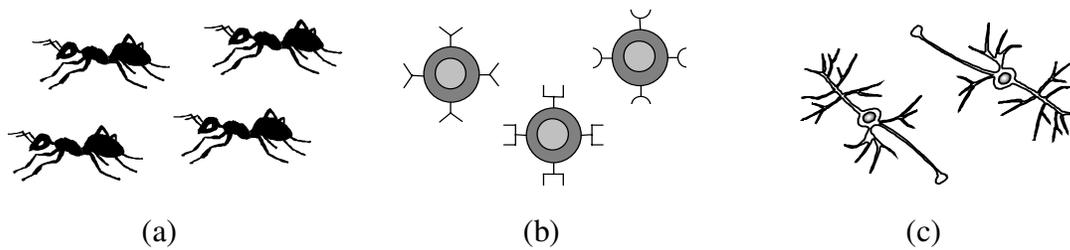


Figura 3.1. Exemplos de sistemas naturais compostos por diversos elementos distintos com capacidade limitada, mas cuja coletividade permite a emergência de comportamentos complexos. (a) Formigas (colônia de insetos). (b) Linfócitos (sistema imunológico). (c) Neurônios (sistema nervoso central).

Quando os diversos agentes operam em conjunto, comportamentos complexos podem emergir. Os sistemas vivos são, sem sombra de dúvidas, *sistemas complexos*.

Definição 3.1: Um *sistema complexo* pode ser visto como um conjunto de objetos, agentes, elementos, ou processos, que exibem um comportamento dinâmico e agregado, em que a ação de um objeto possivelmente afeta as ações subsequentes dos outros objetos (Bonabeau & Theraulaz, 1995), englobando o comportamento coletivo e sua interação com o ambiente.

Idéias extraídas de sistemas naturais já vêm sendo utilizadas com muito sucesso para o desenvolvimento de ferramentas tecnológicas capazes de resolver problemas de complexidade elevada, cujas soluções eram, até então, desconhecidas ou inatingíveis. Devido ao seu grau de complexidade e capacidade de processamento de informação, aquele, dentre os sistemas naturais, que tem recebido maior atenção é o cérebro humano. A *inteligência artificial* se desenvolveu em um estágio inicial de investigação acerca da inteligência humana, visando compreender como um cérebro de dimensões físicas reduzidas e consumo de energia limitado, seja ele biológico ou eletrônico, poderia perceber, compreender, prever e manipular um mundo extremamente diversificado (Russel & Norvig, 1995). A inteligência artificial levou ao desenvolvimento dos computadores como são conhecidos atualmente, e vem sendo aplicada a problemas com representação simbólica de dados, e problemas em diversas áreas, como, por exemplo, busca e teoria de jogos. Na tentativa de criar sistemas de processamento a partir de uma abordagem conexionista, foram desenvolvidas as *redes neurais artificiais*, que representam uma tecnologia fundamentada em várias disciplinas como neurociência, matemática, estatística, física, computação e engenharia (Haykin, 1999). Suas aplicações vão desde o reconhecimento de padrões até a otimização de funções e aprendizagem de máquina. Os *sistemas nebulosos*, ou *sistemas fuzzy*, surgiram como uma nova formalização matemática para a inferência e raciocínio lógicos, permitindo o tratamento formal de grandezas envolvendo incertezas. Suas aplicações em problemas de engenharia, principalmente na área de controle, têm sido muitas, como controle de sinais de trânsito, condicionamento de ar, injeção eletrônica de combustíveis, freios ABS, etc. (Pedrycz & Gomide, 1998). Por outro lado, a *simulação de processos evolutivos naturais* objetivando resolver problemas de explosão combinatória e multimodais demonstrou ser uma estratégia eficaz e robusta. A otimização do comportamento de um sistema obtida através da simulação de processos evolutivos

representa uma abordagem poderosa para aprendizagem de máquina e estudo de fenômenos auto-organizados. A implementação computacional destes métodos de simulação da evolução, chamada *computação evolutiva*, possibilita a determinação de soluções para várias classes de problemas (Bäck *et al.*, 2000a,b). Além dessas, outras linhas de pesquisa ainda mais recentes têm surgido como novos paradigmas de computação. É possível destacar a *vida artificial* (Langton, 1995) que engloba todos aqueles sistemas desenvolvidos pelo homem e que possuem alguma propriedade fundamental da vida natural (Taylor & Jefferson, 1995), e a *computação de DNA*, ou *computação molecular* (Adleman, 1994), que propõe a utilização de ferramentas da *biologia molecular* para a codificação de cadeias de DNA e manipulação de operações computacionais seguindo *protocolos* básicos e utilizando processos enzimáticos, como será melhor discutido na Seção 4.6.1. Mais recentemente, a *Semiótica Computacional* (SC) surgiu como uma tentativa de emular, em um computador digital, o ciclo *semiótico* (Gudwin, 1996). O principal objetivo da SC é o de construir sistemas autônomos que apresentem comportamentos inteligentes, incluindo percepção, modelagem do ambiente, capacidade de julgamento e comportamento emergente. Grande parte deste comportamento inteligente deve ser resultado do processamento semiótico dentro do sistema autônomo, no sentido de que o sistema inteligente resultante deve ser comparável a um sistema semiótico. O aspecto central da SC está na descoberta de unidades elementares de inteligência e suas relações semióticas.

3.2. Sistemas Imunológicos Artificiais e Engenharia Imunológica

Além das abordagens citadas na seção anterior, um novo ramo da teoria de sistemas inteligentes, denominado *sistemas imunológicos artificiais* (SIA), tem surgido ao longo dos últimos dez anos e vem se consolidando através de workshops em diversas conferências desde o ano de 1996. Naquele ano, foi realizado no Japão o primeiro workshop sobre sistemas baseados em imunologia, IMBS (*Immunity-Based Systems*). É certo que o estudo e a modelagem de aspectos do sistema imunológico já vinham sendo realizados há muito mais tempo, porém sem a iniciativa de integração e caracterização de uma linha de pesquisa, que passou a existir a partir de 1996. Desde 1997, vêm sendo organizadas anualmente sessões especiais (*special tracks*) na conferência System, Man and Cybernetics (SMC) do IEEE (Instituto dos Engenheiros Eletricistas e Eletrônicos). No ano de 2000, além da sessão especial no SMC, houve um workshop e uma palestra intitulada “Por que um Computador Precisa de um Sistema Imunológico” na conferência chamada GECCO (*Genetic and Evolutionary Computation Conference*), realizada em Las Vegas/EUA. Um tutorial sobre *computação imunológica* também foi apresentado no CEC em julho de 2000 (*Congress on Evolutionary Computation*). No Brasil, o primeiro tutorial sobre “Sistemas Imunológicos Artificiais e Suas Aplicações” foi proferido pelo autor desta tese no VI Simpósio Brasileiro de Redes Neurais (SBRN), realizado no Rio de Janeiro em novembro de 2000. No ano de 2001, a revista IEEE Transactions on Evolutionary Computation trará uma edição especial (*special issue*) sobre os sistemas imunológicos artificiais. Para o ano de 2001, estão agendados. Todos estes eventos demonstram o crescente interesse por esta nova linha de pesquisa e estão resumidos na Tabela 3.1.

Existem diversas teorias, algumas contraditórias, e modelos computacionais para simular diferentes componentes e mecanismos do sistema imunológico. Também existe um crescente número de sistemas desenvolvidos utilizando-se um ou mais princípios imunológicos, como os descritos no Capítulo 2, com os objetivos primordiais de modelar o sistema biológico e resolver problemas complexos de engenharia. A partir de 1996, estes sistemas passaram a ser denominados *sistemas imunológicos artificiais*.

Tabela 3.1. Histórico dos principais eventos e acontecimentos em Sistemas Imunológicos Artificiais. IMBS: *Immunity-Based Systems Conference*; SMC: *Systems, Man And Cybernetics*; GECCO: *Genetic And Evolutionary Computation Congress*; SBRN: *Simpósio Brasileiro de Redes Neurais*; CEC: *Congress on Evolutionary Computation*; ICANNGA: *International Conference on Artificial Neural Networks and Genetic Algorithms*; TEC: *Transactions on Evolutionary Computation*.

Ano	Evento	Atividade	Organizador/ Palestrante
1996	IMBS	Workshop	Y. Ishida
1997	SMC	Special Track / Tutorial	D. Dasgupta
1998	Livro	Primeiro livro editado	D. Dasgupta (ed.)
	SMC	Special Track	D. Dasgupta
1999	SMC	Special Track	D. Dasgupta
2000	GECCO	Workshop	D. Dasgupta
		Palestra: “Por que um Computador Precisa de um Sistema Imunológico”	S. Forrest
	SMC	Tutorial: “Computação Imunológica”	D. Dasgupta
		Special Track	
SBRN	Tutorial: “Sistemas Imunológicos Artificiais e Suas Aplicações”	L. N. de Castro	
2001	ICANNGA	Tutorial: “Uma Introdução aos Sistemas Imunológicos Artificiais”	L. N. de Castro
	CEC	Tutorial: “Sistemas Imunológicos Artificiais: Uma Tecnologia Emergente”	J. Timmis
	GECCO	Palestra: “Computação Baseada no Sistema Imunológico”	S. Forrest
	Revista	Primeiro Special Issue sobre SIA na revista IEEE TEC	D. Dasgupta (ed.)

Embora alguns trabalhos propuseram apresentar um modelo genérico de sistemas imunológicos artificiais (Hunt & Cooke, 1996; Hofmeyr & Forrest, 1999, 2000), poucos foram aqueles que apresentaram uma definição formal para eles, e nenhum contribuiu com uma linguagem genérica o bastante para permitir um tratamento unificado de conceitos e ferramentas. Como exemplo, podemos citar as seguintes definições:

Definição 3.2: Os *sistemas imunológicos artificiais* são metodologias de manipulação de dados, classificação, representação e raciocínio que seguem um paradigma biológico plausível: o sistema imunológico humano (WWW Starlab).

Definição 3.3: Um *sistema imunológico artificial* é um sistema computacional baseado em metáforas do sistema imunológico natural (Timmis, 2000a).

O primeiro e único livro até o momento sobre sistemas imunológicos artificiais foi editado no ano de 1998 e traz como título “Sistemas Imunológicos Artificiais e Suas Aplicações” (Dasgupta, 1998). O editor apresenta a seguinte definição para os sistemas imunológicos artificiais:

Definição 3.4: Os *sistemas imunológicos artificiais* são compostos por metodologias inteligentes, inspiradas no sistema imunológico biológico, para a solução de problemas de mundo real (Dasgupta, 1998).

O prefácio do livro fornece o seguinte escopo desta nova linha de pesquisa:

- Métodos computacionais inspirados em princípios imunológicos;
- Sistemas imunológicos artificiais aplicados ao reconhecimento de padrões;
- Sistemas baseados em imunologia para a detecção de falhas e anomalias;
- Modelos de redes imunológicas e suas aplicações;
- Sistemas multi-agentes baseados em imunologia;
- Abordagens multi-agentes para a modelagem e simulação de sistemas imunológicos;
- Sistemas auto-organizados baseados em imunologia;
- Sistemas baseados em imunologia para o desenvolvimento de inteligência coletiva;
- Métodos de busca e otimização baseados em imunologia;
- Sistema imunológico como protótipo para sistemas autônomos descentralizados;
- Abordagens imunológicas para vida artificial;
- Abordagens imunológicas para segurança de sistemas de informação;
- Abordagens imunológicas para proteção contra vírus e vermes computacionais;
- Metáforas imunológicas para aprendizagem de máquina; e
- Computação imunológica para mineração de dados (*data mining*).

É possível verificar, pelas definições e pelo escopo apresentado, que os sistemas imunológicos artificiais fundamentam linhas de atuação bem amplas, com aplicações em diversas áreas. Nosso objetivo, ao apresentar um modelo formal de *engenharia imunológica* (EI), não é apenas o de introduzir um conceito formal genérico, mas também o de fazer uma coletânea dos modelos existentes na literatura de SIA, definir suas potencialidades e explorá-las de forma que seja possível desenvolver ferramentas de

engenharia imunológica partindo da síntese dos SIAs e da análise dos mecanismos imunológicos envolvidos.

A engenharia imunológica pode então ser definida da seguinte forma:

Definição 3.5: A *engenharia imunológica* é um processo de meta-síntese, o qual vai definir a ferramenta de solução de um determinado problema baseado nas características do próprio problema, e depois aplicá-la na obtenção da solução. Ao invés de buscar a reconstrução parcial ou total do sistema imunológico tão fielmente quanto possível, a engenharia imunológica deve procurar desenvolver e implementar modelos pragmáticos inspirados no sistema imunológico que preservem algumas de suas propriedades essenciais e que se mostrem passíveis de implementação computacional e eficazes no desenvolvimento de ferramentas de engenharia.

É importante salientar que não é nosso objetivo estabelecer uma fronteira rígida entre os conceitos e nomenclaturas já existentes (SIA, computação imunológica, etc.) e a engenharia imunológica, mas apresentá-la como um novo paradigma computacional que, não apenas utiliza os já existentes, mas também propõe um modelo genérico de construção de ferramentas de solução de problemas. Para tanto, sempre que conveniente, a EI irá restringir arbitrariamente a diversidade de células e moléculas dos sistemas imunológicos artificiais, e os tipos de funções que elas podem exercer. Será apresentado, na Seção 3.4, um mecanismo genérico para modelagem matemática dos linfócitos B e T e dos antígenos, e serão descritas expressões que permitem avaliar a afinidade entre eles. Serão propostas representações pictóricas, similares às existentes nas redes neurais artificiais, permitindo uma visualização gráfica das interações entre componentes dos SIA, e entre eles e os antígenos. Embora a EI proponha um modelo para os dois principais linfócitos do sistema imunológico biológico, veremos, nos Capítulos 5 e 6, que ela opera com um único tipo celular, no caso o linfócito B. Devido a monoespecificidade dos receptores antigênicos deste tipo de linfócito, não é feita nenhuma distinção entre uma célula B e seus receptores antigênicos de superfície, sendo os componentes celulares e moleculares da EI denominados genericamente de anticorpos.

Assim, partindo dos conceitos apresentados para os sistemas imunológicos artificiais e dos propósitos vinculados à engenharia imunológica, é possível propor uma visão mais abrangente para o elenco total de componentes dos sistemas imunológicos artificiais, sendo possível acrescentar ao seu escopo de atuação os seguintes itens:

- Estruturas e algoritmos híbridos baseados em mecanismos ou metáforas imunológicas;
- Algoritmos computacionais inspirados em princípios imunológicos; processo também denominado de computação imunológica; e
- Engenharia imunológica e suas aplicações.

Sob esta nova taxonomia, segue abaixo uma lista de aplicações potenciais dos sistemas imunológicos artificiais. Destas possíveis aplicações, as 8 primeiras são amplamente exploradas pela engenharia imunológica, como será visto nos Capítulos 5 e 6.

1. Reconhecimento de padrões;
2. Aproximação de funções;
3. Otimização de processos;
4. Análise de dados e clusterização;
5. Aprendizagem de máquina;
6. Memórias associativas;
7. Geração e manutenção de diversidade;
8. Programação e computação evolutiva;
9. Detecção de falhas e anomalias;
10. Controle e scheduling;
11. Segurança computacional e de rede; e
12. Geração de comportamentos emergentes.

Na próxima seção, discutiremos as diversas ferramentas de sistemas imunológicos artificiais existentes na literatura. Estas ferramentas estarão categorizadas de acordo com suas características e/ou aplicações, sendo possível anteciper que nem todas elas serão de interesse direto para a engenharia imunológica.

3.3. Estado da Arte em Sistemas Imunológicos Artificiais

Devido, principalmente, à insipiência desta linha de pesquisa, é possível listar a grande maioria dos trabalhos já publicados pela comunidade científica no campo dos sistemas imunológicos artificiais. O relatório técnico RT DCA – 02/00 do Departamento de Engenharia de Computação e Automação Industrial, Faculdade de Engenharia Elétrica e de Computação da Unicamp, publicado em fevereiro de 2000 (de Castro & Von Zuben, 2000a), faz uma revisão de parte desta literatura enfocando os esquemas de codificação, funções de avaliação e como a metáfora imunológica foi utilizada para o desenvolvimento destas ferramentas. Atualmente, um ano depois, a quantidade de trabalhos publicados subiu de aproximadamente 100 para algo em torno de 130, evidenciando ainda mais o crescimento desta nova linha de pesquisa.

3.3.1. Alguns Trabalhos Pioneiros

O artigo publicado por Farmer *et al.* (1986) é conhecido como sendo um dos primeiros trabalhos a relacionar o sistema imunológico com os *sistemas classificadores* propostos por J. Holland (1998). Entretanto, na primeira edição de seu livro, datada de 1974, J. Holland já propunha algumas similaridades de representação entre a teoria da rede imunológica e os algoritmos genéticos.

Farmer *et al.* (1986) propuseram um modelo teórico contínuo baseado em equações diferenciais ordinárias, como será brevemente descrito na Seção 3.4.3.5, para modelar a teoria da rede imunológica discutida na Seção 2.10. Foi demonstrado que as equações dinâmicas do sistema imunológico e de um sistema classificador assumem a mesma forma, sugerindo que os sistemas imunológicos, assim como os sistemas classificadores, podem ter grande aplicabilidade na solução de problemas de engenharia. Este foi o primeiro artigo a

aplicar cadeias de atributos binárias na representação das moléculas e células do sistema imunológico.

Além deste, os trabalhos de Hoffmann (1986) e Hoffmann *et al.* (1986) estão entre os primeiros encontrados na literatura a propor um modelo formal de rede neural artificial inspirada no sistema imunológico (Seção 3.3.7).

É importante mencionar que anteriormente a estes já existiam diversos modelos teóricos do sistema imunológico (*imunologia teórica*), com o enfoque principal na compreensão e modelagem dos fenômenos imunológicos envolvidos, e não o desenvolvimento de sistemas imunológicos artificiais como um novo paradigma da teoria de sistemas inteligentes. Os artigos de Mohler *et al.* (1980) e Perelson & Weisbusch (1997) apresentam amplas revisões sobre a imunologia teórica.

No Capítulo 4, discutiremos algumas relações entre os sistemas imunológicos artificiais e biológicos, o sistema nervoso central e outras técnicas de sistemas inteligentes, enfocando as redes neurais artificiais e a computação evolutiva.

3.3.2. Reconhecimento de Padrões

Hunt & Cooke (1996a) propuseram um sistema imunológico artificial baseado no modelo de Farmer *et al.* (1986) discutido anteriormente. Este modelo possui características similares ao sistema biológico e propõe uma rede adaptativa não-linear com um mecanismo evolutivo de aprendizagem e memória endereçável por conteúdo. Os autores compararam este modelo com as *redes neurais artificiais*, os *sistemas classificadores* e os *sistemas de raciocínio baseado em casos* (CBR – *Case-Based Reasoning*) e apresentaram aplicações a problemas de reconhecimento de padrões. A implementação em Java deste sistema foi proposta posteriormente (Hunt *et al.*, 1999).

Um modelo binário do sistema imunológico foi utilizado por Forrest & Perelson (1992) e Forrest *et al.* (1993) com o objetivo de estudar a aprendizagem e o reconhecimento de padrões que ocorrem em nível individual e de espécies no sistema biológico. Um *algoritmo genético* (ver Capítulo 4) foi utilizado para estudar a manutenção da diversidade e capacidade de generalização deste modelo binário, onde generalização corresponde à detecção de esquemas comuns compartilhados por diversos antígenos. Os resultados referentes à formulação matemática foram apresentados em Smith *et al.* (1993).

Em uma *reação química*, um conjunto de *reagentes* é mapeado em um conjunto de *produtos*. Ainda utilizando uma representação binária, Dasgupta *et al.* (1999) descreveram um conjunto de reagentes e produtos (antígenos) para o reconhecimento de espectros. Cada cadeia binária (antígeno ou anticorpo) corresponde à ocorrência de um pico dentro de um mesmo comprimento de onda. Um algoritmo genético clássico foi utilizado para criar uma biblioteca de anticorpos especialistas capazes de executar o reconhecimento do espectro.

3.3.3. Segurança Computacional

O problema de segurança computacional envolve principalmente a detecção de intrusos (usuários não-autorizados) em uma rede de computadores e a detecção e eliminação de vírus e vermes computacionais.

Forrest *et al.* (1994) compararam a proteção de sistemas de computadores com o problema de distinção próprio/não-próprio (ver Seção 2.9) tratado naturalmente pelo sistema imunológico. Foi descrita uma estratégia de detecção de anomalias baseada na seleção negativa de linfócitos T dentro do timo, ou seleção tímica (ver Seção 2.9.2). Foi desenvolvido um algoritmo chamado de *algoritmo de seleção negativa*, que será descrito na Seção 3.4 como um dos componentes formais dos sistemas imunológicos artificiais. Objetivando superar algumas deficiências da estratégia proposta por Forrest *et al.* (1994), D'haeseleer *et al.* (1996) introduziram dois novos algoritmos para geração de detectores. Wierzhón (2000a,b) estudou os métodos de geração do repertório linfocitário propostos por Forrest *et al.* (1994) e aperfeiçoados por D'haeseleer *et al.* (1996) e propôs um novo algoritmo com baixa complexidade computacional. Ele também apresentou receitas de como contar “buracos” no repertório, ou seja, como avaliar a quantidade de antígenos (cadeias) que podem passar despercebidos pelo SIA.

No sistema desenvolvido por Kephart (1994), um conjunto de anticorpos específicos para reconhecer vírus ou vermes computacionais desconhecidos é gerado de forma a promover uma resposta mais rápida e eficiente a infecções futuras. Ele também estava preocupado em minimizar o risco de doenças auto-imunes, de modo que o computador poderia identificar erroneamente softwares legítimos como ilegítimos. Outras referências sobre o mesmo trabalho podem ser encontradas em Kephart *et al.* (1997a,b, 1999).

Lamont *et al.* (1999) e Harmer & Lamont (2000) propuseram uma arquitetura distribuída para um sistema imunológico de detecção e eliminação de vírus computacionais. Este projeto faz uso extensivo de outras abordagens, como algoritmos evolutivos e agentes inteligentes, para desenvolver um sistema autônomo adaptativo capaz de reconhecer e eliminar novos tipos de vírus.

Dasgupta (1999d, 2000) propôs um sistema baseado em agentes para detecção e resposta a anomalias e/ou intrusos em redes de computadores. Nesta abordagem, os agentes percorrem os nós e roteadores da rede monitorando sua situação. Os agentes imunológicos possuem características como mobilidade, adaptabilidade e colaboratividade, sendo capazes de interagir dinamicamente com o ambiente e outros agentes.

Okamoto & Ishida (1999a,b) apresentaram uma abordagem distribuída para o problema de detecção e neutralização de vírus computacionais via agentes autônomos heterogêneos. O sistema detecta os vírus fazendo uma comparação dos elementos próprios (como os primeiros bytes do cabeçalho de um arquivo, o tamanho do arquivo, seu caminho, etc.) com os arquivos ativos do *host*. A neutralização do vírus é feita sobrescrevendo-se a informação do próprio, contida em outros computadores da rede, nos arquivos infectados.

Kim & Bentley (1999a) revisaram a analogia entre o sistema imunológico humano e os sistemas de detecção de intrusos de rede. O objetivo principal deste artigo era desvendar as

características significativas do sistema biológico que se mostravam úteis para o desenvolvimento de sistemas de proteção de redes de computadores. Posteriormente (Kim & Bentley, 1999b), eles propuseram um algoritmo de seleção negativa com *niching* para tratar o mesmo problema.

Hofmeyr & Forrest (1999, 2000) têm trabalhado no desenvolvimento de um sistema imunológico artificial para o problema de segurança de redes de computadores. Estes trabalhos utilizam o algoritmo de seleção negativa (Forrest *et al.*, 1997), que será discutido na Seção 3.4, e tentam desenvolver um sistema imunológico artificial para computadores.

Gu *et al.* (2000) propuseram uma *camada de anticorpos (antibody layer)*, inspirada em mecanismos imunológicos biológicos, que detectam dinamicamente novos antígenos da Internet e também simplificam os procedimentos de segurança de redes de computadores.

3.3.4. Robótica

Os principais trabalhos de Ishiguro *et al.* (1996, 1997) e Watanabe *et al.* (1998, 1999) focalizaram o desenvolvimento de um mecanismo dinâmico e descentralizado de tomada de decisões (*consensus-making*) baseado na teoria da rede imunológica. Nesta abordagem, esperava-se que a “inteligência” emergisse de interações mútuas entre os agentes, e entre eles e seu ambiente. Esta estratégia foi avaliada utilizando-se um robô para coleta de lixo.

O objetivo do sistema proposto por Jun *et al.* (1999) e Lee *et al.* (1999) era desenvolver um conjunto de robôs capaz de encontrar e realizar determinadas tarefas espalhadas em um ambiente, ou seja, robôs que apresentassem um comportamento coletivo emergente. Para isso, eles utilizaram o princípio da seleção clonal e a teoria da rede imunológica.

3.3.5. Otimização

Hajela & Lee (1996), Hajela *et al.* (1997) e Hajela & Yoo (1999), descreveram a implementação de uma busca genética (ver Seção 4.4) restrita para simular mecanismos do sistema imunológico biológico. Os autores estudaram a capacidade de adaptação e reconhecimento de padrões (esquemas) do sistema biológico vislumbrando melhorar o desempenho de algoritmos genéticos aplicados a problemas de otimização estrutural restrita.

Endo *et al.* (1998) e Toma *et al.* (1999) propuseram um algoritmo de otimização adaptativa baseado na teoria da rede imunológica e no processo de apresentação peptídica por moléculas de MHC (ver Seção 2.7.2).

Fukuda *et al.* (1999) desenvolveram um algoritmo imunológico através da modelagem da diversidade, teoria da rede e seleção clonal para solucionar problemas de otimização de funções multimodais.

De Castro & Von Zuben (1999b, 2000b, 2001a) apresentaram um algoritmo de seleção clonal, que inclui o processo de maturação de afinidade da resposta imunológica, para resolver problemas complexos como aprendizagem de máquina, reconhecimento de padrões e otimização multimodal. Este algoritmo será descrito em detalhes na Seção 5.3.

Gaspar & Collard (1999, 2000) exploraram o problema de otimização dependente do tempo, que consiste em localizar ótimos sucessivos de funções não-estacionárias, como uma medida de adaptabilidade de sistemas artificiais. O processo de otimização dependente do tempo associa, no tempo t , uma função de fitness a cada indivíduo, levando à definição de uma função de fitness dependente do tempo. Utilizando modelos de redes imunológicas, um sistema imunológico artificial é proposto e comparado com os métodos evolutivos de otimização dependente do tempo.

3.3.6. Controle

Bersini (1991) descreveu o desenvolvimento de uma metodologia de controle adaptativo cujos elementos básicos são inspirados na compreensão do sistema imunológico biológico. Entre eles destacam-se a noção de um domínio de viabilidade, a utilização de aprendizagem por reforço, a distributividade do controle, a transposição da metadinâmica para uma técnica de otimização, a adaptabilidade e a memória. Similaridades entre o SI, o método Q-learning e os sistemas classificadores também foram descritas.

Krishnakumar *et al.* (1995) e Krishnakumar & Neidhoefer (1997a,b, 1999) definiram um sistema computacional imunizado que utiliza a metáfora imunológica juntamente com técnicas computacionais de hardware e software para reproduzir a adaptabilidade e robustez do sistema imunológico biológico. Esta abordagem foi testada no problema de controle autônomo de um avião.

Ootsuki & Sekiguchi (1999) propuseram um método inspirado no sistema imunológico para a determinação de seqüências de controle de uma planta industrial. Foram utilizadas redes de Petri e a teoria da rede imunológica.

3.3.7. Abordagens Conexionistas

Por se tratar de sistemas com grande número de componentes e atributos, existem várias semelhanças e diferenças entre o sistema nervoso central e o sistema imunológico. Jerne (1974a), Hoffmann (1986), Hoffmann *et al.* (1986), Vertosick & Kelly (1989, 1991), Dasgupta (1997, 1999a) e De Castro & Von Zuben (2001f) ilustram a relação entre estes sistemas e também levam em consideração as redes neurais artificiais. Estes aspectos serão discutidos na Seção 4.3.

Além da relação entre os dois sistemas, alguns algoritmos de treinamento de redes neurais artificiais têm surgido, ou sido otimizados, utilizando-se idéias inspiradas na imunologia. Hoffmann (1986) e Hoffmann *et al.* (1986) utilizaram a analogia entre a rede imunológica e as redes neurais para desenvolver um modelo denominado de rede neural artificial não-ortodoxo.

Os trabalhos de Vertosick & Kelly (1989, 1991) conjecturaram a possibilidade do sistema imunológico oferecer uma representação alternativa na busca por arquiteturas de redes neurais. Baseados nos sistemas de processamento paralelo e distribuído (Rumelhart *et al.*, 1986), eles procuraram mapear o sistema imunológico em uma rede neural.

Abbattista *et al.* (1996) apresentaram um modelo de memória associativa, baseado na teoria da rede imunológica, capaz de armazenar e recuperar padrões contendo ruído. Eles tentaram aumentar a capacidade de memória de uma variação da rede de Hopfield discreta (Hopfield, 1984) utilizando a metadinâmica (Varela *et al.*, 1988) da teoria da rede imunológica. Para uma descrição da rede de Hopfield discreta, ver Seção 4.2.2.3.

De Castro & Von Zuben (2001f) destacaram os aspectos cognitivos do sistema imunológico, resultando no desenvolvimento de uma rede neural Booleana competitiva com um processo de crescimento simulando a expansão clonal dos linfócitos B mais estimulados e a morte das células menos estimuladas. As principais características deste algoritmo são: aprendizagem competitiva, geração automática da arquitetura de rede e representação binária do conjunto de pesos. Este modelo será detalhado na Seção 5.4.

Como será discutido na Seção 4.2.2.1, para que uma rede neural artificial com função de ativação de base radial (RBF – *radial basis function*) opere adequadamente, é necessário que os parâmetros das RBFs sejam determinados de forma apropriada. De Castro & Von Zuben (2001b,e) propuseram um algoritmo inspirado em um modelo de rede imunológica artificial, a ser apresentado na Seção 5.5, capaz de determinar a localização e quantidade dos centros das RBFs, baseado apenas no conjunto de dados de entrada da rede.

A inicialização do conjunto de pesos a ser utilizado no treinamento de uma rede neural artificial do tipo perceptron de múltiplas camadas (ver Seção 4.2.1.1.2) possui uma grande influência na velocidade de aprendizagem e qualidade da solução obtida após a convergência da rede (De Castro & Von Zuben, 1998a,b; De Castro *et al.*, 1998a). De Castro & Von Zuben (2001c) propuseram uma abordagem imunológica para a inicialização de pesos deste tipo de rede. Este método será descrito na Seção 5.2, e seus resultados serão apresentados na Seção 6.3.

3.3.8. Detecção de Falhas e Anomalias

O comportamento normal de um sistema é usualmente caracterizado por uma série de observações ao longo do tempo. O problema de detectar anomalias, ou falhas, pode ser visto como a determinação de desvios em relação a um padrão de comportamento característico do sistema. A identificação de vírus computacionais e intrusos de rede são considerados problemas de detecção de anomalias, mas como pertencem a uma classe muito particular de aplicações, foram discutidos separadamente na Seção 3.3.3.

Aisu & Mazutani (1996) propuseram um sistema de aprendizagem imunológica com o objetivo de integrar um mecanismo de busca distribuída a um processo de relaxamento de restrições baseado no modelo de rede imunológica introduzido por Farmer *et al.* (1986). Este sistema foi aplicado ao problema de detecção automática de anomalias em imagens.

O algoritmo de seleção negativa proposto por Forrest *et al.* (1994) foi adaptado para detecção de anomalias em séries temporais (Dasgupta & Forrest, 1996). Em um trabalho posterior (Dasgupta & Forrest, 1999), os autores apresentaram uma discussão mais detalhada sobre este sistema, destacando suas aplicações industriais como, por exemplo, a detecção de ruptura de ferramentas.

Deaton *et al.* (1997) apresentaram uma implementação baseada em cadeias de DNA do algoritmo de seleção negativa proposto por Forrest *et al.* (1994) e discutido na Seção 3.3.3, para estudar a detecção de variações em cadeias genéticas.

Por último, McCoy & Devarajan (1997) estudaram as relações entre o reconhecimento de padrões efetuado pelo sistema imunológico e o problema de segmentação de imagens aéreas. O algoritmo de seleção negativa foi utilizado para definir um conjunto de detectores capaz de identificar todos os elementos não pertencentes à classe desejada.

3.3.9. Abordagens Baseadas em Agentes

Como no caso dos algoritmos de detecção de anomalias, as abordagens baseadas em agentes também têm sido muito aplicadas a diversos problemas, com destaque para a segurança computacional. Sendo assim, esta seção irá apresentar apenas aqueles modelos cujas aplicações ainda não foram apresentadas na Seção 3.3.3.

Dilger (1996, 1997) propôs um sistema de segurança para prédios inteligentes baseado na teoria de agentes e no sistema imunológico. Seu objetivo inclui aumentar as facilidades técnicas de uma casa, ou prédio, como, por exemplo, o controle da temperatura ambiente. O sistema de segurança foi modelado de forma similar ao sistema imunológico biológico em dois aspectos: ele era capaz de se auto-reconhecer e produzir uma reação flexível de acordo com o grau de perigo.

Explorando propriedades imunológicas como diversidade, tolerância e memória, Ishida (1996c) desenvolveu um algoritmo imunológico baseado em agentes com aplicações aos problemas de neutralização de ruídos, ajuste de modelos e tomada de decisão.

3.3.10. Aprendizagem de Máquina

É importante mencionar que quase todos os trabalhos revisados nesta tese abordam, de certa forma, o problema de aprendizagem de máquina. Entretanto, nesta seção daremos ênfase apenas àqueles artigos que tratam a aprendizagem como aplicação principal.

O efeito Baldwin propõe que a aprendizagem de características úteis durante a vida de um indivíduo pode aumentar a probabilidade de que gerações futuras adquiram estas mesmas características, mesmo que elas não sejam geneticamente propagáveis. Hightower *et al.* (1996) utilizaram um modelo binário do sistema imunológico, juntamente com o algoritmo de seleção clonal e maturação de afinidade, para estudar a influência da aprendizagem na evolução da representação genética de um repertório de anticorpos.

Potter & De Jong (1998) utilizaram um mecanismo de distinção próprio/não-próprio como fonte de inspiração para o desenvolvimento de um algoritmo genético co-evolutivo que diferencia exemplos de contra-exemplos de um dado conceito. Este modelo foi aplicado à aprendizagem de conceitos partindo de exemplos positivos e negativos pré-classificados.

Objetivando estudar os processos complexos apresentados pelos organismos vivos, incluindo aprendizagem e evolução, Nagano & Yonezawa (1999) propuseram um modelo baseado em sistemas complexos e mecanismos primitivos de defesa existentes no SI.

3.3.11. Autômatos Celulares

No início dos anos 90 surgiram alguns modelos do sistema imunológico baseados em autômatos celulares (Celada & Seiden, 1992, 1996; Seiden & Celada, 1992), porém o objetivo destes modelos era o mesmo daqueles que utilizavam equações diferenciais: servir como mais uma ferramenta de análise do próprio sistema imunológico biológico. Estes modelos inicialmente só consideravam um grupo restrito de linfócitos, e foram posteriormente estendidos para incluir respostas a vírus e às células T citotóxicas (Bezzi *et al.*, 1997; Kleinstein & Seiden, 2000). Uma implementação em paralelo destes autômatos celulares foi proposta por Bernaschi *et al.* (1998).

3.3.12. Outras Aplicações

Ballet *et al.* (1997) simularam uma resposta secundária utilizando um sistema multi-agentes com o objetivo de estudar a cinética da proliferação de anticorpos em resposta a vários tipos de substâncias antigênicas. Dentre eles, antígenos letais fracamente reconhecíveis e proliferativos.

Utilizando um modelo relativamente mais simples do que o empregado em Ballet *et al.* (1997), Epstein & Axtell (1996) estudaram como a dinâmica das infecções intra- e inter-agentes poderia afetar outros processos sociais, como migração e comércio.

Slavov & Nikolaev (1998) propuseram um algoritmo evolutivo de busca baseado numa versão discreta de um modelo de rede imunológica. O objetivo era resolver problemas de busca NP-completos através da indução de um autômato com número finito de estados.

Nikolaev *et al.* (1999) introduziram uma versão imunológica da programação genética, onde uma busca progressiva era controlada por uma função de adaptabilidade dinâmica, baseada na analogia com um modelo de rede imunológica.

Suzuki & Yamamoto (2000a,b) estão trabalhando no desenvolvimento de uma estrutura baseada em Java para a construção de uma rede imunológica artificial a ser aplicada na coordenação de servidores Web abertos.

Hart & Ross (1998, 1999a,b) propuseram um sistema imunológico artificial para a produção de schedules robustos a serem aplicados ao problema de alocação de tarefas, no qual as tarefas chegam continuamente e o ambiente está sujeito a variações. Este modelo inclui a evolução de bibliotecas genéticas, a maturação de afinidade e o princípio da seleção clonal.

Hunt *et al.* (1995) e Hunt & Fellows (1996) ilustraram como o sistema imunológico é inerentemente baseado em casos, e como ele está apoiado em sua memória endereçável por conteúdo para realizar a tarefa de identificar novas situações similares a casos anteriores. Timmis *et al.* (1999, 2000) aplicaram este sistema imunológico artificial organizado sob uma estrutura em rede ao problema de análise de dados e clusterização.

Forrest & Hofmeyr (1999), Hofmeyr & Forrest (1999, 2000) e Farmer *et al.* (1986) apresentaram similaridades e diferenças entre o sistema imunológico biológico e os sistemas classificadores, focalizando seus modelos e aplicações específicas.

Ishida (1993) propôs um modelo de rede imunológica para o reconhecimento mútuo entre um antígeno e um anticorpo aplicado ao diagnóstico de processos. Posteriormente, Ishida (1996a) utilizou este mesmo modelo dinâmico para desenvolver uma rede contendo sensores, e capaz de diagnosticar falhas nos dados recebidos pelos sensores.

Suzuki & Yamamoto (1998) sugeriram que os mecanismos de reconhecimento e eliminação antigênica constituem exemplos da capacidade de reflexão do sistema imunológico: o problema da distinção próprio/não-próprio foi reduzido à comparação da reflexão de uma imagem em um espelho com um elemento externo.

Hightower *et al.* (1995) e Perelson *et al.* (1996) utilizaram um algoritmo genético e um modelo binário do sistema imunológico para estudar os efeitos da evolução na codificação genética de moléculas de anticorpo. Uma característica importante desta codificação é o fato de que nem todos os genes contidos no genótipo estão expressos no fenótipo. A Seção 3.4.3.1, que propõe um modelo formal para a medula óssea, irá descrever em maiores detalhes este modelo de geração de moléculas.

Oprea & Forrest (1998, 1999) utilizaram uma representação similar à discutida anteriormente para explorar como a probabilidade de sobrevivência de um indivíduo está relacionada ao tamanho do repertório de anticorpos, e estenderam esta análise para medidas de afinidade (regras de ligação ou *matching*) antígeno-anticorpo mais realistas. A Seção 3.4.2 apresenta uma série de algoritmos de determinação de afinidades Ag-Ab.

Ishida (1996b) apresentou um breve resumo de aplicações de algoritmos inspirados na imunologia e propôs que o sistema imunológico constitui um processo de auto-identificação.

Uma abordagem baseada no algoritmo clássico de *simulated annealing* (Kirkpatrick *et al.*, 1986) com o objetivo de gerar um conjunto inicial de candidatos à solução de um determinado problema foi apresentada por de Castro & Von Zuben (2000d). Este conjunto de candidatos a solução possui a característica de apresentar uma boa cobertura do espaço de busca. Este algoritmo foi aplicado com sucesso ao problema de inicialização de redes neurais com múltiplas camadas (de Castro & Von Zuben, 2001c).

Finalmente, de Castro & Von Zuben (2000c) desenvolveram um modelo de rede imunológica artificial para a solução de problemas de compressão de informação. Quando utilizada em conjunto com algumas ferramentas de clusterização hierárquica de dados e um método de representação de grafos, esta ferramenta demonstrou capacidade de separar clusters mesmo quando os dados de treinamento são não-linearmente separáveis e para configurações arbitrárias dos clusters (formato e distribuição dos dados).

3.4. Uma Estrutura Formal para a Engenharia Imunológica

Esta seção tem como objetivo principal apresentar um modelo genérico para o desenvolvimento de ferramentas de engenharia imunológica. Para isso, serão descritas formas de modelar matematicamente os diversos componentes do sistema imunológico biológico, os elementos com os quais eles interagem e como medir estas interações. Em seguida, serão apresentados algoritmos para simular diversos mecanismos e princípios

imunológicos biológicos com escopo genérico de aplicações. Por último, serão apresentados exemplos de como estes modelos, mecanismos e algoritmos estão sendo aplicados aos sistemas imunológicos artificiais.

O enfoque da engenharia imunológica está voltado para um único tipo celular, em particular as células B, e mecanismos de reconhecimento, desconsiderando as potencialidades efetoras do sistema imunológico.

3.4.1. O Espaço de Formas (Shape-Space)

Para descrever quantitativamente as interações entre as células e moléculas do sistema imunológico e os antígenos, Perelson & Oster (1979) introduziram o conceito de *espaço de formas*, S (*Shape-Space*). Baseado na idéia do espaço de formas (S), é possível apresentar um argumento teórico demonstrando que um repertório imunológico completo pode ser determinado (Segel & Perelson, 1988; Perelson & Weisbuch, 1997). A completude do repertório imunológico foi discutida na Seção 2.9.

A idéia por trás do espaço de formas é que o grau de *ligação* (*matching*, ou *reconhecimento*) entre um receptor celular e seu *ligante* (molécula que se liga ao receptor celular), geralmente envolve interações não covalentes curtas baseadas em cargas eletrostáticas, ligações de hidrogênio, interações de van der Waals, etc. As moléculas devem possuir uma *afinidade* mútua mínima, que será representada na forma de nível de acoplamento entre superfícies que possuam uma determinada quantidade de *regiões complementares*, como ilustrado na Figura 3.2. A forma, a distribuição da carga eletrostática e a existência de grupos químicos em posições complementares apropriadas são propriedades importantes dos antígenos e anticorpos para a determinação das interações destas moléculas. Este conjunto de características foi denominado de *forma generalizada* de uma molécula (Perelson, 1989). Suponhamos que seja possível descrever a forma generalizada de uma molécula de anticorpo, mais especificamente de sua região de ligação (região-V), por um conjunto de L parâmetros: comprimento, largura, profundidade, carga eletrostática, etc., de qualquer padrão da região de ligação. A quantidade exata de parâmetros (L) e seus respectivos valores podem ser tomados como arbitrários no desenvolvimento de ferramentas de sistemas imunológicos artificiais. Assim, um ponto em um espaço L -dimensional, denominado espaço de formas S , especifica a forma generalizada da região de ligação ao antígeno.

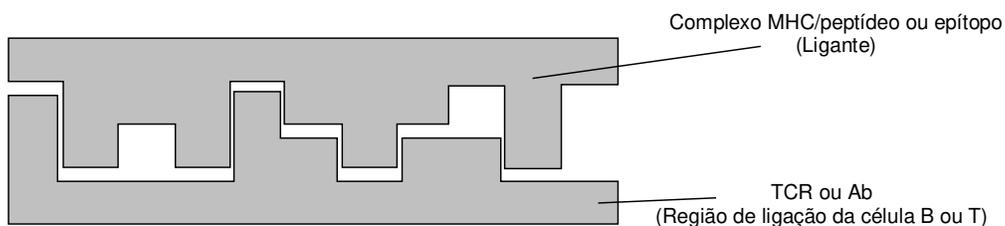


Figura 3.2. Reconhecimento via porções de regiões complementares.

Se o repertório de moléculas de anticorpo é de tamanho N , então o espaço de formas para este repertório contém N pontos. É esperado que estes pontos estejam contidos em uma região finita V do espaço, uma vez que existe uma quantidade finita de larguras, comprimentos, cargas eletrostáticas, etc., que uma região de ligação pode assumir. Como as interações Ag-Ab são medidas via regiões de complementaridade, os determinantes antigênicos (*epítomos* ou *idiotopos*) também são caracterizados por formas generalizadas cujos complementos devem estar contidos dentro da mesma região V . Se as formas do paratopo e do epítopo (ou idiotopo) não forem exatamente complementares, estas moléculas ainda assim podem se ligar, porém com menor afinidade. Assume-se, então, que cada paratopo interage especificamente com todos os epítomos (idiotopos) que, no espaço de formas, têm seus complementos contidos em uma região V_ϵ , chamada de *região de reconhecimento*, ao redor do paratopo, caracterizada pelo parâmetro ϵ denominado de *limiar de afinidade*. Como cada anticorpo pode reconhecer todos os epítomos dentro de uma vizinhança de reconhecimento, um número reduzido de anticorpos pode reconhecer uma quantidade bem maior de pontos na região V_ϵ . Este princípio está relacionado à *reatividade cruzada*, ou *resposta reativa cruzada*, discutida na Seção 2.8.1, onde o complemento de padrões similares ocupam regiões vizinhas no espaço de formas, e podem ser reconhecidos pelo mesmo padrão de anticorpos, desde que um limiar de afinidade ϵ adequado seja escolhido. A Figura 3.3 ilustra o espaço de formas S , destacando os paratopos, epítomos (ou idiotopos) e o limiar de afinidade.

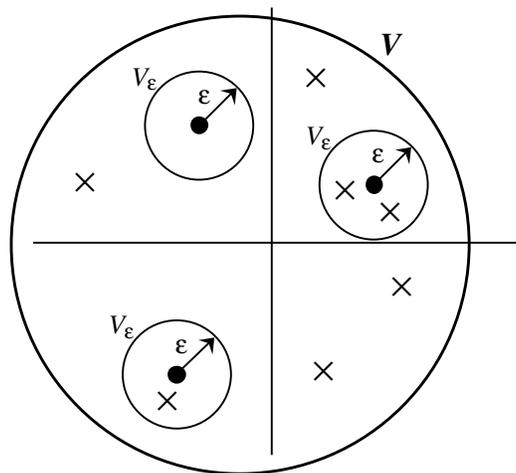


Figura 3.3. No espaço de formas S , existe uma região V na qual a forma do paratopo (\bullet) e do complemento do epítopo (\times) estão localizadas. Um anticorpo é capaz de reconhecer qualquer epítopo (ou idiotopo) cujo complemento esteja situado em uma região V_ϵ em torno do paratopo (adaptado de Perelson, 1989).

3.4.2. Representações Celulares, Moleculares e Suas Afinidades

A representação utilizada para modelar um antígeno e um anticorpo (epítipo, paratopo e idioto) irá contribuir para a definição do tipo de distância a ser empregado na avaliação da *afinidade*, ou *grau de ligação*, entre as moléculas. Matematicamente, a forma generalizada de uma molécula m , seja ela um anticorpo (Ab) ou um antígeno (Ag), pode ser representada por um conjunto de coordenadas com valores reais $m = \langle m_1, m_2, \dots, m_L \rangle$, que é interpretado como um ponto m em um espaço real S^L ($m \in S^L \subseteq \mathcal{R}^L$, onde S representa o espaço de formas e L sua dimensão). Sem perda de generalidade, estudaremos as interações entre um antígeno (próprio ou não-próprio) e um anticorpo.

A afinidade entre um antígeno e um anticorpo (Ag-Ab) está relacionada à distância entre eles, que pode ser estimada via qualquer medida de distância (ou similaridade) entre dois vetores, ou cadeias de atributos, como, por exemplo, a distância Euclidiana ou de Manhattan. Estas medidas de distância podem ser utilizadas para avaliar o grau de ligação, ou *match score*, entre as moléculas. No caso da distância Euclidiana, se as coordenadas de um anticorpo forem dadas por $\mathbf{Ab} = \langle Ab_1, Ab_2, \dots, Ab_L \rangle$ e as coordenadas do complemento de um antígeno dadas por $\mathbf{Ag} = \langle Ag_1, Ag_2, \dots, Ag_L \rangle$, então a distância D entre estas moléculas pode ser medida pela Equação (3.1). A Equação (3.2) apresenta o caso da distância de Manhattan.

$$D = \sqrt{\sum_{i=1}^L (Ab_i - Ag_i)^2}. \quad (3.1)$$

$$D = \sum_{i=1}^L |Ab_i - Ag_i|. \quad (3.2)$$

O espaço de formas S que utiliza coordenadas reais e cuja medida de distância é descrita pela Equação (3.1) é chamado de *espaço de formas Euclidiano* (Segel & Perelson, 1988; De Boer *et al.*, 1992; Smith *et al.*, 1997). Espaços de forma que utilizam coordenadas reais e a distância de Manhattan, ao invés da Euclidiana, são denominados *espaços de formas de Manhattan*. Embora não seja conhecida nenhuma referência sobre os espaços de Manhattan, a distância de Manhattan constitui uma alternativa interessante para a distância Euclidiana, principalmente no caso de implementações em paralelo (ou em hardware) de algoritmos baseados no espaço de formas. Entretanto, é importante mencionar que embora a distância de Manhattan permita uma implementação computacional mais eficiente para implementações em hardware do que a norma Euclidiana (Ienne & Kuhn, 1995), ela altera a topologia do espaço.

Uma alternativa aos espaços de valores reais é o chamado *espaço de formas de Hamming*, no qual os antígenos e os anticorpos são representados por seqüências de símbolos, denominadas *cadeias de atributos*, pertencentes a um alfabeto finito de tamanho k (Farmer *et al.*, 1986; de Boer & Perelson, 1991; Seiden & Celada, 1992a,b; Hightower *et al.*, 1995, 1996; Perelson *et al.*, 1996; Detours *et al.*, 1996; Smith *et al.*, 1997; Oprea & Forrest, 1998, 1999). Tais seqüências podem ser interpretadas como peptídeos e os diferentes símbolos como propriedades dos aminoácidos ou de classes equivalentes de aminoácidos

que serão concatenados para formar um receptor celular ou um ligante antigênico (ver Seção 2.7 que trata os mecanismos de geração de receptores de células B). A Equação (3.3) representa a medida de distância de Hamming utilizada para avaliar a afinidade entre um antígeno e um anticorpo no espaço de formas de Hamming:

$$D = \sum_{i=1}^L \delta_i, \text{ onde } \delta_i = \begin{cases} 1 & \text{se } Ab_i \neq Ag_i \\ 0 & \text{outros casos} \end{cases} \quad (3.3)$$

Dependendo do problema a ser abordado, um *espaço de formas Inteiro* também pode ser empregado. Neste tipo de espaço, os atributos correspondem a variáveis que assumem valores inteiros. Estes espaços são muito utilizados em problemas como o caixeiro viajante (Endo *et al.*, 1998; Toma *et al.*, 1999; de Castro & Von Zuben, 2000a) ou problemas de *scheduling* (Hart *et al.*, 1998; Hart & Ross, 1999).

As Equações de (3.1) a (3.3) apresentam as propostas mais diretas de determinação da afinidade molecular (cálculo do *match score*) no espaço Euclidiano, de Manhattan e de Hamming, respectivamente. Se a afinidade *D* entre as moléculas for maior ou igual a um determinado *limiar de afinidade* ϵ , então assume-se que ocorreu uma ligação entre as moléculas. Note que este limiar de afinidade é diretamente proporcional à quantidade de antígenos que um anticorpo pode reconhecer e pode assumir valores diferentes para cada um dos anticorpos.

No espaço de formas, as coordenadas de um antígeno estão representadas como o complemento de seus atributos. Sendo assim, quando a distância entre um anticorpo e o complemento de um antígeno é mínima, as moléculas apresentam uma afinidade máxima. Se a afinidade entre as moléculas não é máxima, torna-se necessário recorrer às normas definidas em cada espaço, para que seja possível medir as interações Ag-Ab. Por outro lado, se adotarmos as coordenadas do próprio antígeno no espaço de formas, e não mais de seu complemento, então a afinidade será maior para distâncias maiores, pelo menos no caso da distância de Hamming e alfabeto binário. Para outros tipos de normas e alfabetos, deve-se adotar o conceito de padrões de reflexão, introduzidos por Suzuki & Yamamoto (1998), ou seja, deve-se refletir a imagem do antígeno em um espelho antes de medir sua afinidade a um dado anticorpo. Um exemplo está ilustrado na Figura 3.4.

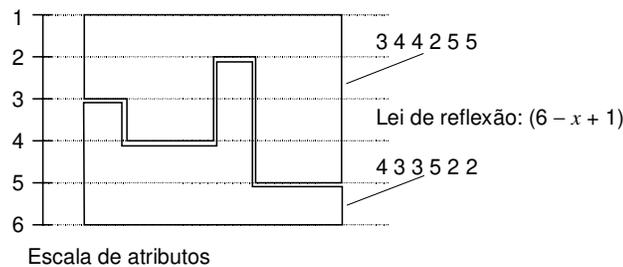


Figura 3.4. Padrões de reflexão para o caso de cadeias de atributos complementares.

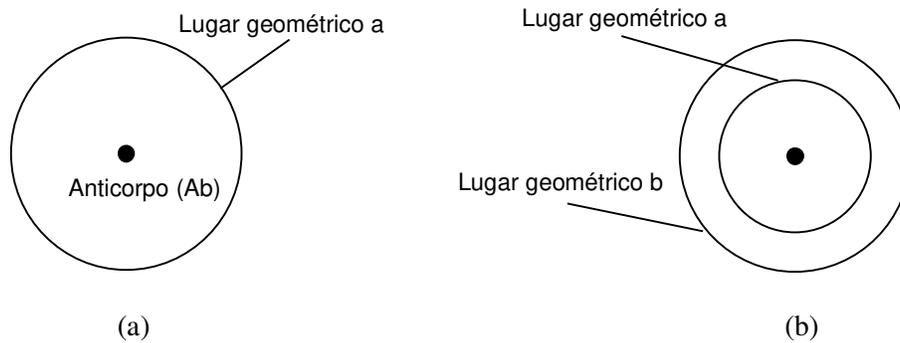


Figura 3.5. Afinidade por similaridade de formas em espaços Euclidianos. (a) Lugar geométrico de antígenos com mesma afinidade ao anticorpo. (b) Os antígenos pertencentes ao lugar geométrico b têm maior afinidade ao anticorpo que os antígenos pertencentes ao lugar geométrico a.

Embora não esteja totalmente de acordo com o conceito de complementaridade de formas, a afinidade em espaços de formas onde as variáveis assumem valores inteiros ou reais (o intervalo de valores deve ser finito) e em que a distância é medida pela norma Euclidiana pode ser definida como sendo proporcional à distância entre um anticorpo e um antígeno (sem complementá-lo ou refleti-lo). Neste caso, vai haver um lugar geométrico de pontos no espaço de formas tal que todo antígeno que lá se encontra apresenta a mesma afinidade a um dado anticorpo, como ilustrado na Figura 3.5 para atributos assumindo valores reais.

3.4.2.1. Medidas de Afinidade para Espaços de Formas de Hamming

No espaço de Hamming, assuma que cadeias binárias representam as moléculas. Assim, a interpretação gráfica dos antígenos e dos anticorpos (Ag-Ab) é direta, como ilustrado na Figura 3.6.

Ainda no caso dos espaços de formas de Hamming, existem diversas outras maneiras, descritas na literatura, de se calcular a afinidade entre duas moléculas, lembrando que a ligação entre um anticorpo e um antígeno ocorre quando suas cadeias são, de alguma forma, complementares (ver Figura 3.2).

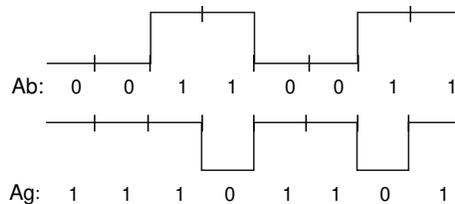


Figura 3.6. Interpretação gráfica da interação de duas cadeias binárias de comprimento $L = 8$.

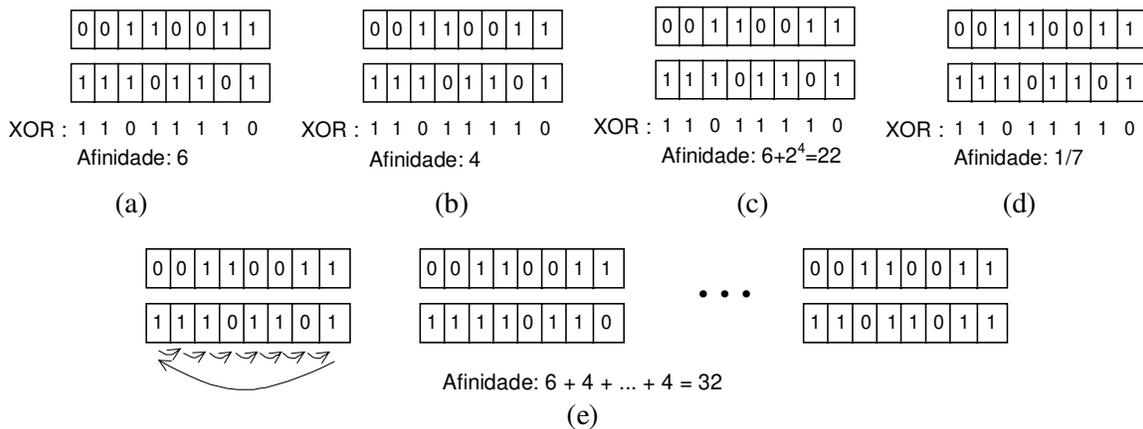


Figura 3.7. Diferentes algoritmos de cálculo de afinidade entre moléculas representadas no espaço de formas de Hamming. (a) Quantidade total de bits complementares (Equação (3.3)). (b) Quantidade de r -bits complementares consecutivos. (c) Algoritmo de Hunt (Equação (3.4)). (d) Medida de similaridade de Rogers & Tanimoto. (e) Deslocamento de bits da segunda cadeia e soma da quantidade total de bits complementares para cada alinhamento.

A *afinidade* entre duas moléculas pode corresponder ao número de bits complementares (Figura 3.7(a)) (Forrest & Perelson, 1992; Hajela & Lee, 1996; Hightower *et al.*, 1996; De Castro & Von Zuben, 2000d), ou à quantidade de r -bits complementares consecutivos, ilustrado na Figura 3.7(b) (Forrest *et al.*, 1994; Dasgupta & Forrest, 1996; Hofmeyr & Forrest, 1999, 2000). Como pode ser visto nesta figura, a afinidade pode ser calculada pela aplicação do operador OU-exclusivo (XOR) entre os bits que estão alinhados. Para o caso (a), a afinidade esperada entre duas moléculas escolhidas aleatoriamente é igual à metade de seus comprimentos, assumindo que este seja o mesmo para as duas cadeias. Espaços de formas que medem r -bits complementares consecutivos são mais plausíveis sob o ponto de vista biológico.

Hunt *et al.* (1995, 1996) propuseram que regiões com grande quantidade de bits complementares podem ser interessantes para a detecção de características similares em regiões simétricas das moléculas, o que poderia ser útil para o reconhecimento de padrões, por exemplo. Denominamos de *medida de afinidade de Hunt* a seguinte regra:

$$D = D_H + \sum_i 2^{l_i} , \tag{3.4}$$

onde D_H é a distância de Hamming total dada pela Equação (3.3), e l_i é o comprimento l de cada parte i da região de ligação que possui mais do que 2 bits complementares, como ilustrado na Figura 3.7(c).

Harmer & Lamont (2000) utilizaram a medida de similaridade de Rogers & Tanimoto para avaliar a afinidade entre duas moléculas representadas por cadeias binárias:

$$\begin{aligned}
 D &= \frac{a+d}{a+d+2(b+c)}, \\
 a &= \sum_{i=1}^L \zeta_i, \quad \zeta_i = \begin{cases} 1 & Ab_i = Ag_i = 1 \\ 0 & \text{outros casos} \end{cases} \\
 b &= \sum_{i=1}^L \xi_i, \quad \xi_i = \begin{cases} 1 & Ab_i = 1, Ag_i = 0 \\ 0 & \text{outros casos} \end{cases} \\
 c &= \sum_{i=1}^L \gamma_i, \quad \gamma_i = \begin{cases} 1 & Ab_i = 0, Ag_i = 1 \\ 0 & \text{outros casos} \end{cases} \\
 d &= \sum_{i=1}^L \varphi_i, \quad \varphi_i = \begin{cases} 1 & Ab_i = Ag_i = 0 \\ 0 & \text{outros casos} \end{cases}
 \end{aligned} \tag{3.5}$$

Os autores sugerem que esta medida é mais seletiva do que a distância de Hamming e menos do que o critério de r -bits complementares consecutivos.

Uma outra forma de medir a interação de duas moléculas pode ser feita fixando-se uma delas e deslizando-se o último bit da segunda para a primeira posição desta mesma cadeia e aplicando-se o XOR entre as cadeias resultantes. Este processo é repetido até que a cadeia volte à sua configuração original, como ilustrado na Figura 3.7(e). Esta estratégia pode ser empregada para cadeias com comprimentos distintos.

A definição de um parâmetro ε correspondente ao limiar de afinidade Ag-Ab determina se ocorre ou não a ligação entre estas moléculas. Entretanto, a ocorrência da ligação, e portanto do reconhecimento antigênico, pode assumir valores graduais, ou seja, pode existir um *valor de ligação* (vl) entre as moléculas pertencente ao intervalo $[0,1]$, onde $vl = 0$ indica a não existência de ligação (não reconhecimento) e $vl = 1$ corresponde à afinidade máxima Ag-Ab (reconhecimento perfeito).

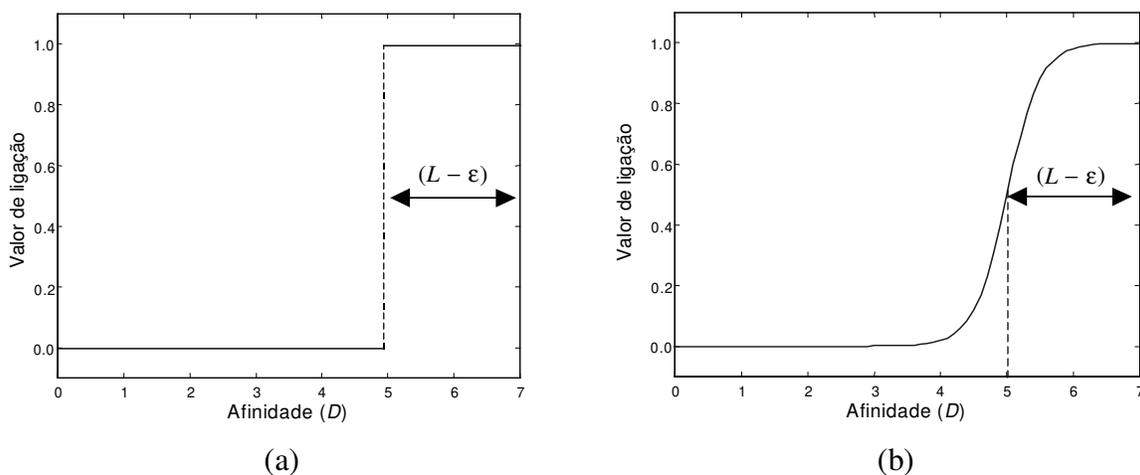


Figura 3.8. Relação entre o valor de ligação e a afinidade D para cadeias de comprimento $L = 7$ e limiar de afinidade $\varepsilon = 2$. (a) Função de ativação degrau. (b) Função de ativação sigmoidal.

Para definir o valor de ligação (vl) entre duas moléculas, proporcionalmente à afinidade D entre elas, várias *funções de ativação* podem ser empregadas como, por exemplo, uma função do tipo *degrau* ou *sigmoidal* (Hightower *et al.*, 1996; de Castro & Von Zuben, 1999, 2000d), ilustradas na Figura 3.8. No caso da ativação tipo degrau (Figura 3.8(a)), uma ligação é estabelecida quando o valor de afinidade é superior a $L - \epsilon$. No caso de ativação sigmoidal (Figura 3.8(b)), ϵ define o ponto de inflexão da curva, implicando que uma afinidade superior a $L - \epsilon$ produzirá um alto valor de ligação. A inclinação da função sigmoidal pode ser variada de acordo com o problema a ser tratado. Uma função de ativação do tipo degrau modela a fronteira *crisp* de ligação (reconhecimento) discutido acima, onde o antígeno é ou não reconhecido, enquanto uma função do tipo sigmoidal permite a existência de valores graduais de ligação.

No espaço de formas de Hamming, a quantidade total N de moléculas distintas que podem ser geradas é dada por

$$N = k^L, \quad (3.6)$$

onde k é o tamanho do alfabeto (número de elementos distintos) e L é o comprimento das cadeias. Uma dada molécula de anticorpo reconhece um conjunto de antígenos e, portanto, cobre uma certa porção do espaço de formas (ver Figura 3.3). O limiar de afinidade ϵ determina a *cobertura* de cada anticorpo. Se $\epsilon = L$, ou seja, uma ligação perfeita é desejada, então um anticorpo só é capaz de reconhecer o antígeno representado pelo seu complemento exato. No espaço de formas binário de Hamming, a quantidade de antígenos coberta por um único anticorpo com limiar de afinidade ϵ é dada por

$$C = \sum_{i=0}^{\epsilon} \binom{L}{i} = \sum_{i=0}^{\epsilon} \frac{L!}{i!(L-i)!}, \quad (3.7)$$

onde C é a cobertura de cada anticorpo e L o comprimento das cadeias.

Baseado na Equação (3.7), dados uma única cadeia de comprimento L e um limiar de afinidade ϵ , a quantidade mínima N_m de moléculas de anticorpo necessárias para cobrir completamente o espaço de formas de Hamming é

$$N_m = \text{ceil} \left(\frac{N}{C} \right), \quad (3.8)$$

onde N é dado pela Equação (3.6), C é dado pela Equação (3.7) e *ceil* é o operador que arredonda o valor entre parênteses para o inteiro superior mais próximo. A Tabela 3.2 exemplifica a cobertura do espaço de antígenos para cadeias binárias de diferentes comprimentos e limiares de afinidade.

Embora a abordagem de espaço de formas tenha sido proposta para avaliar as interações de antígenos e anticorpos, ela pode ser aplicada a contextos mais complexos, incluindo a modelagem de receptores de células T (TCRs) e moléculas do complexo de MHC (Seiden & Celada, 1992; Forrest *et al.*, 1993; Hightower *et al.*, 1996; Perelson *et al.*, 1996).

Tabela 3.2. Cobertura antigênica C de cada anticorpo e tamanho mínimo N_m do repertório para cobertura completa, considerando cadeias binárias ($k = 2$) de diferentes comprimentos L e limiares de afinidade ε .

L	2^L	$\varepsilon = 0$		$\varepsilon = 1$		$\varepsilon = 2$		$\varepsilon = 3$	
		C	N_m	C	N_m	C	N_m	C	N_m
2	4	1	4	3	2	4	1	-----	-----
3	8	1	8	4	2	7	2	8	1
4	16	1	16	5	4	11	2	15	2
6	64	1	64	7	10	22	3	42	2
8	256	1	256	9	29	37	7	93	3
16	65536	1	65536	17	3856	137	479	697	95
32	4.30×10^9	1	4.30×10^9	33	1.30×10^8	529	8.12×10^6	5489	7.82×10^5
64	1.84×10^{19}	1	1.84×10^{19}	61	2.84×10^{17}	2081	8.86×10^{16}	43745	4.22×10^{14}

3.4.2.2. Representações Pictóricas

A principal característica da linguagem proposta para a engenharia imunológica é tentar formalizar quantitativamente as interações das células e moléculas do sistema imunológico e seus ligantes, incluindo elementos do próprio sistema, assim como descrever mecanismos genéricos de interação. O enfoque é dado a um único tipo de célula, especificamente as células B, aos antígenos (próprios e não-próprios) e como medir a afinidade entre eles. Sendo assim, é possível introduzir uma representação pictórica, similar à adotada para as redes neurais artificiais (ver Capítulo 4), para o modelo genérico das células (ou unidades) que compõem a EI, como descrito na Figura 3.9. É importante salientar que este modelo assume um único tipo de célula para o sistema imunológico, não sendo suficientemente genérico para ser aplicável a todos os componentes dos sistemas imunológicos artificiais, que podem incluir diversos elementos distintos, como macrófagos, células B e T, sinalizadores (citocinas), etc. (ver Seção 3.3). Por outro lado, a grande maioria dos modelos de SIAs propostos atualmente opera com um único tipo celular e poderia, portanto, ser formalizada utilizando este modelo.

Um aspecto interessante do modelo proposto é que uma célula m_i , representada por uma cadeia (ou vetor) de L atributos, pode interagir com um elemento, que pode ser outra célula ou um estímulo antigênico, m_j , com um número distinto l de atributos. Além disso, o grau de interação entre eles é quantizado por um limiar ε , que irá determinar a afinidade mínima necessária entre eles para que haja uma ligação (reconhecimento). A célula genérica m_i do sistema imunológico é composta pelo seu paratopo (de comprimento l_1) e seu idiotopo (de comprimento $L - l_1$), como proposto na teoria da rede imunológica (Seção 2.10).

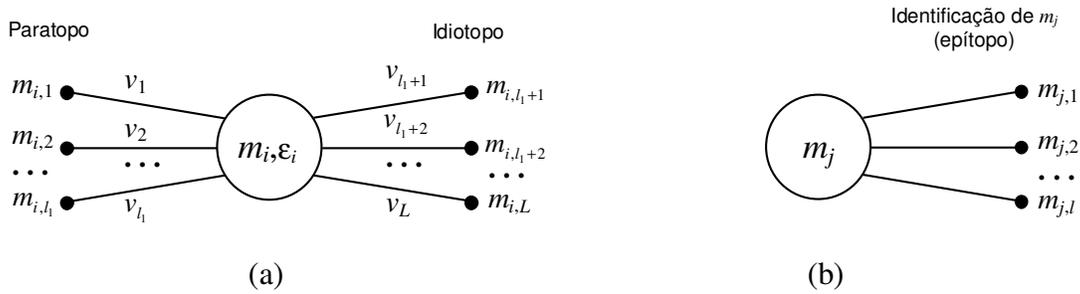


Figura 3.9. Representações pictóricas de um anticorpo e um antígeno no contexto da engenharia imunológica. (a) Modelo genérico do anticorpo $m_i = \langle m_{i,1}, \dots, m_{i,l_i}, m_{i,l_i+1}, \dots, m_{i,L} \rangle$ com seu respectivo limiar de afinidade ϵ_i . (b) Antígeno $m_j = \langle m_{j,1}, m_{j,2}, \dots, m_{j,l} \rangle$.

Na Figura 3.9(a), cada célula possui um conjunto de receptores (conexões) correspondentes ao paratopo, e responsáveis pela interação com os outros elementos do ambiente, sejam eles células ou moléculas do próprio organismo (outros anticorpos) ou elementos invasores (antígenos), ilustrados na Figura 3.9(b). Também podem existir pesos associados a cada uma destas conexões, para ponderar o grau de interação de cada componente da cadeia que representa os elementos individuais. Geralmente são utilizados pesos associados quando os modelos abordam a teoria da rede imunológica e, mesmo assim, todos os modelos apresentados na literatura até a atualidade, incluindo o que será proposto na Seção 5.5, utilizam um único valor de peso associado a cada célula, diferentemente da representação proposta, que permite um peso específico $v_k, k = 1, \dots, L$, associado a cada atributo do vetor m . É importante notar que estes pesos possuem uma conotação diferente, em sistemas imunológicos artificiais, daquela empregada em alguns tipos de redes neurais artificiais. No primeiro caso, o peso representa o grau de interação das células e no segundo, cada peso corresponde a uma parcela da informação armazenada na rede. Na Seção 4.3.2, veremos com mais detalhes as similaridades e diferenças entre estes modelos.

A Figura 3.10 apresenta as respectivas representações compactas dos elementos apresentados na Figura 3.9. Esta representação justifica-se pela facilidade de visualização de modelos como os propostos para a teoria da rede imunológica, em que a representação apresentada na Figura 3.9 implicaria em um excesso de conexões na figura dificultando sua interpretação.

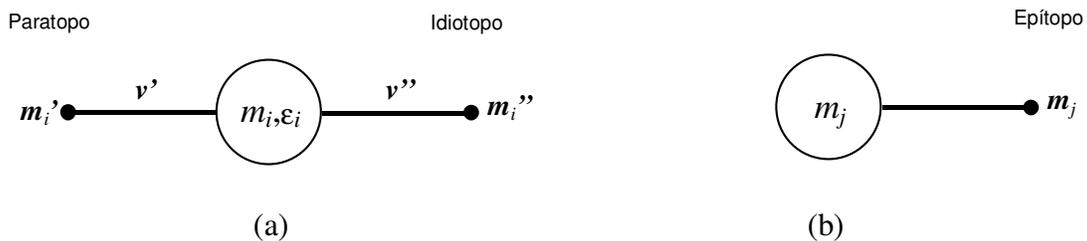


Figura 3.10. Representações compactas. (a) Célula da engenharia imunológica correspondente à Figura 3.9(a) $m_i = [m_i', m_i'']$. (b) Elemento invasor (antígeno não-próprio), correspondente à Figura 3.9(b).

As representações pictóricas propostas nas Figuras 3.9 e 3.10 para os elementos da engenharia imunológica determinam apenas suas características internas. Para medir a afinidade entre estes elementos, pode ser utilizado qualquer um dos algoritmos propostos, dado o espaço de formas apropriado e a representação molecular adotada.

3.4.3. Algoritmos e Processos

Esta seção visa ilustrar os principais algoritmos computacionais que modelam diferentes mecanismos e teorias imunológicas, e que podem ser aplicados genericamente aos sistemas imunológicos artificiais. Dentre estes, destacam-se *modelos de medula óssea* (MMO), o *algoritmo de seleção negativa* (ASN), o *algoritmo de seleção clonal* (CLONALG – *clonal selection algorithm*) e os *modelos contínuos da rede imunológica*. Os modelos de medula óssea são utilizados para a geração de vetores e/ou cadeias de atributos que representam os repertórios imunológicos; o ASN possui aplicações diversas em detecção de anomalias, como segurança computacional e previsão de séries temporais; o algoritmo CLONALG tem várias aplicações no reconhecimento de padrões e otimização multimodal e combinatorial; e os modelos contínuos de rede imunológica são normalmente empregados em processos de robótica e controle.

3.4.3.1. Medula Óssea

Como discutido na Seção 2.5, a medula óssea é responsável pela geração das células sanguíneas, incluindo os linfócitos. Foi visto também que as células B possuem anticorpos de especificidade única, ou seja, cada célula B produz um único tipo de anticorpo. Devido à monoespecificidade das células B, muitos modelos, como o de engenharia imunológica proposto nesta tese, não fazem distinção entre a célula B e sua molécula de superfície, ou anticorpo.

O material genético de uma molécula de anticorpo, e portanto de uma célula B, está armazenado em cinco bibliotecas separadas e distintas, sendo que duas delas são utilizadas para gerar a região variável da cadeia leve (V_L) e três para gerar a região variável da cadeia pesada (V_H). A produção de uma molécula de anticorpo se dá através da concatenação de componentes selecionados aleatoriamente a partir de cada uma das bibliotecas gênicas.

O modelo computacional mais simples de medula óssea é aquele que gera cadeias, ou vetores, de comprimento L utilizando um gerador de números pseudo-aleatórios. Para o caso de espaços reais, basta determinar o intervalo de pertinência do vetor m como, por exemplo, $m \in [0,1]^L$. No caso de espaços de Hamming, o vetor que representa a molécula m deve ser composto por elementos pertencentes a um alfabeto finito pré-definido. Para espaços inteiros, um algoritmo de permutação de L pode ser empregado.

Os modelos mais complexos, e biologicamente mais plausíveis, de construção de repertórios linfocitários exigem a utilização de bibliotecas gênicas a partir das quais as células e/ou moléculas serão evoluídas ou simplesmente geradas.

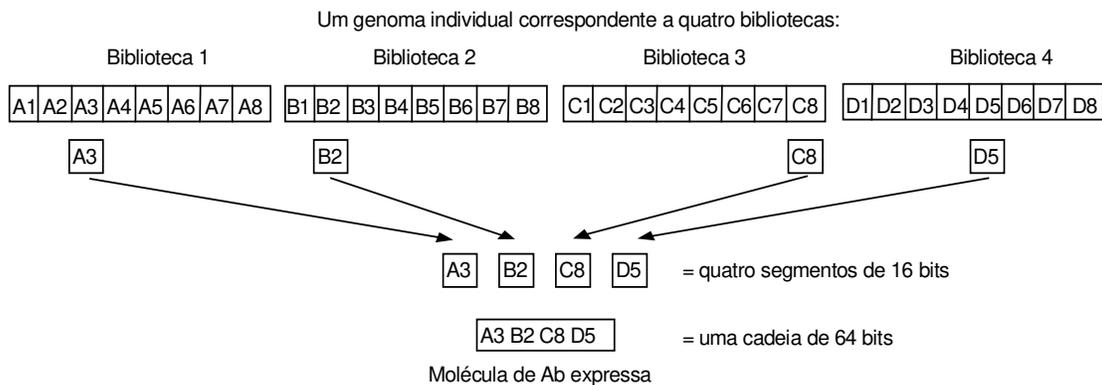


Figura 3.11. Processo de construção/expressão de um anticorpo a partir de bibliotecas genéticas.

Hightower *et al.* (1995) e Perelson *et al.* (1996) utilizaram um algoritmo genético para estudar o efeito da evolução na codificação genética das moléculas de anticorpo. Uma característica desta codificação é que nem todos os genes existentes no *genótipo* (coleção total de genes) estão expressos no *fenótipo* (moléculas de anticorpo expressas). Neste modelo, as cadeias binárias representando o genótipo de um indivíduo foram divididas em quatro bibliotecas de segmentos gênicos para a geração de moléculas de anticorpo, como ilustrado na Figura 3.11. Cada biblioteca continha oito elementos, representados por cadeias binárias de comprimento 16, de forma que cada genoma individual possuía um total de 128 bits. Os anticorpos expressos tinham comprimento total igual a 64 ($4 \times 16 = 64$).

Modelos semelhantes podem ser empregados para a simulação da medula óssea no processo de geração de receptores celulares. A quantidade de bibliotecas, o tamanho dos segmentos gênicos e o comprimento L das moléculas serão definidos pelo projetista de acordo com o problema a ser tratado. É importante ressaltar que a utilização de bibliotecas gênicas para a geração do repertório linfocitário impõe inerentemente uma determinada estrutura ao repertório, como o tipo de dado a ser utilizado (numérico, simbólico, etc.) e o intervalo de varredura (p.ex., 1-10, Segunda, Terça, ..., etc.).

Um sistema imunológico contendo l bibliotecas, cada uma com c componentes, pode produzir c^l moléculas de anticorpo distintas, ou seja, o repertório potencial de anticorpos é composto por c^l moléculas.

3.4.3.2. Algoritmo de Seleção Negativa

Forrest *et al.* (1994) desenvolveram um algoritmo para detecção de anomalias, com aplicações em proteção computacional, baseado na seleção negativa de linfócitos T dentro do timo. Este tópico já foi abordado na Seção 2.9, que discute os processos de distinção próprio/não-próprio. O algoritmo foi denominado de *algoritmo de seleção negativa* (ASN).

O ASN é executado em duas fases, como a seguir:

1. *Censoriamento*, Figura 3.12(a):

- Defina o conjunto de cadeias próprias (S) que se deseja proteger;

- Gere cadeias aleatoriamente e avalie a afinidade (*match*) entre cada uma delas e as cadeias próprias. Caso a afinidade seja superior a um determinado limiar, rejeite a cadeia, caso contrário armazene-a em um conjunto de detectores (R).

2. *Monitoramento*, Figura 3.12(b):

- Dado o conjunto de cadeias que se deseja proteger (cadeias protegidas), avalie a afinidade entre cada uma delas e o conjunto de detectores. Se a afinidade for superior a um limiar pré-definido, então um elemento não-próprio foi identificado.

Os autores utilizaram uma representação no espaço binário de Hamming, e a regra dos r -bits consecutivos como medida de afinidade. Também foram apresentadas equações estimando a probabilidade de ocorrência de uma ligação em pelo menos r -bits consecutivos entre duas cadeias binárias aleatórias.

Note que este algoritmo é capaz de executar tarefas como reconhecimento de padrões armazenando informações sobre o conjunto complementar (não-próprio) ao conjunto dos padrões que se deseja proteger (próprio).

Algumas observações podem ser feitas em relação ao algoritmo de seleção negativa:

1. Ele é ajustável proporcionalmente à probabilidade de detecção de uma anomalia para a regra dos r -bits consecutivos;
2. O tamanho do conjunto de detectores não cresce necessariamente com o número de cadeias a serem protegidas;
3. A probabilidade de detecção de anomalias aumenta exponencialmente com a quantidade de algoritmos de detecção independentes;
4. A detecção é simétrica; e
5. Existe um custo exponencial de geração de detectores em relação ao tamanho do conjunto de cadeias a serem protegidas (próprio), pois a geração aleatória das cadeias do conjunto R_0 leva à geração repetida de diversas cadeias.

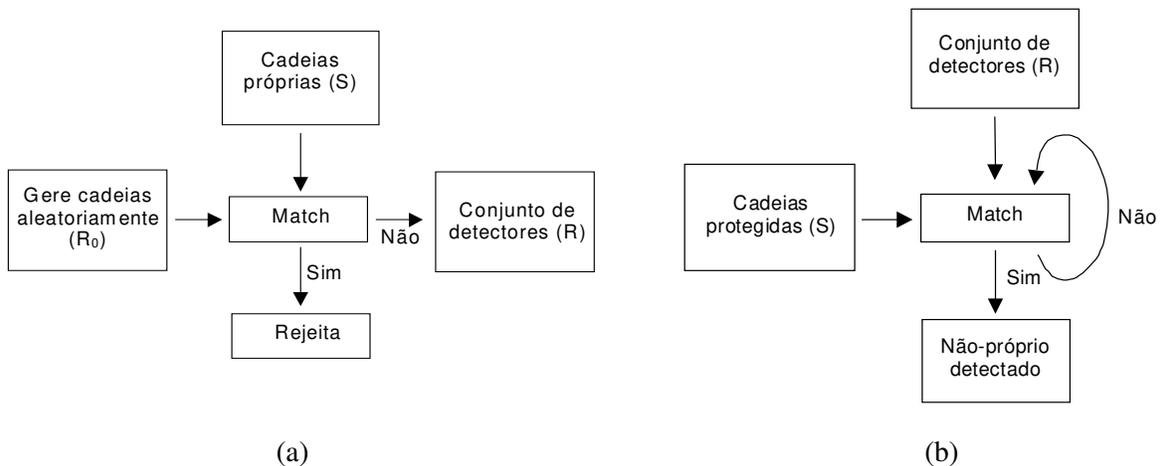


Figura 3.12. Fluxograma do algoritmo de seleção negativa. (a) Geração de um conjunto válido de detectores (censuriamento). (b) Monitoramento das cadeias protegidas, que podem ser as cadeias próprias adicionadas de outros dados que se deseja proteger.

A principal dificuldade deste algoritmo de seleção negativa está no item 5 acima. Algumas abordagens foram propostas para superar esta deficiência (D'haeseleer *et al.*, 1996; Wierzbón, 2000a,b).

3.4.3.3. Algoritmo de Seleção Clonal

O princípio da seleção clonal e o processo de maturação de afinidade constituem a base de uma resposta imune adaptativa (Seção 2.8). Os principais aspectos considerados para o desenvolvimento do algoritmo de seleção clonal (CLONALG – *clonal selection algorithm*) foram (de Castro & Von Zuben, 2000a): manutenção das células de memória funcionalmente independentes do repertório, seleção e reprodução (clonagem) das células mais estimuladas, morte das células menos estimuladas, maturação de afinidade e re-seleção dos clones com maiores afinidades antigênicas, geração e manutenção de diversidade, hipermutação proporcional à afinidade celular. Este algoritmo foi desenvolvido em duas versões: uma para o reconhecimento de padrões e aprendizagem de máquina e a outra para otimização. As duas versões serão apresentadas em detalhes nas Seções 5.3.2.1 e 5.3.2.2, respectivamente. Nesta seção, o algoritmo CLONALG será apresentado de forma mais genérica e simplificada.

O funcionamento do algoritmo pode ser descrito como segue:

1. Gere um conjunto (P) de candidatos a solução, composto pelo subconjunto de células de memória (M) mais o restante ($P_{\{r\}}$) da população ($P = P_{\{r\}} \cup M$);
2. Determine (processo de seleção) os n melhores indivíduos ($P_{\{n\}}$) da população (P), baseado em uma medida de afinidade;
3. Reproduza (processo de clonagem) estes n melhores indivíduos, gerando uma população temporária de clones (C). A quantidade de filhos de cada indivíduo é diretamente proporcional à sua afinidade;
4. Submeta a população de clones a um esquema de hipermutação (ver próxima seção), em que a taxa de mutação é proporcional à afinidade do anticorpo. Uma população de anticorpos maduros é gerada (C^*);
5. Re-selecione os melhores indivíduos de C^* para compor o conjunto de memória M;
6. Substitua d anticorpos por novos indivíduos (diversidade). Os anticorpos com menores afinidades possuem maiores probabilidades de serem substituídos.

Neste algoritmo, se tomarmos $n = N$, ou seja, se selecionarmos todos os indivíduos da população para reprodução, cada candidato à solução será visto localmente, resultando em um algoritmo capaz de executar uma busca multimodal dentro do espaço de formas S .

As etapas 2 e 5 do CLONALG podem ser feitas de forma probabilística, ou seja, aqueles indivíduos com maiores afinidades terão maiores probabilidades de serem selecionados.

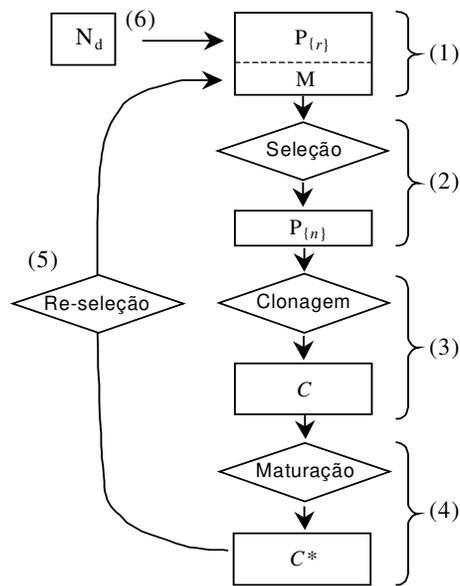


Figura 3.13. Diagrama de blocos do algoritmo de seleção clonal, CLONALG. A numeração aponta as etapas de funcionamento do algoritmo.

3.4.3.4. Hipermutação Somática

A hipermutação somática é utilizada pelo sistema imunológico para inserir e manter a diversidade do repertório linfocitário e também para melhorar a afinidade (capacidade de reconhecimento) dos anticorpos em relação aos estímulos aplicados.

Como o espaço de formas permite representar computacionalmente qualquer componente do sistema imunológico e seus ligantes através de vetores e/ou cadeias de atributos, é possível utilizar diversos algoritmos para a inserção de variações na codificação destes elementos. Estes algoritmos podem ser idênticos àqueles empregados na computação evolutiva (Capítulo 4), desde que seja respeitado o tipo de espaço de formas empregado na codificação das cadeias.

No caso dos espaços de Hamming, uma posição da cadeia é escolhida aleatoriamente e seu elemento trocado por um outro elemento pertencente ao alfabeto k . A Figura 3.14 ilustra uma mutação de ponto único e de múltiplos pontos para o caso de cadeias binárias e de cadeias quaternárias, ou seja, aquelas com um alfabeto de comprimento $k = 4$.

Um espaço de formas Inteiro pode ser tratado como um espaço de Hamming quando a quantidade de valores que podem ser assumidos pelos elementos do vetor de atributos é reduzida. Por outro lado, se esta quantidade for elevada, o espaço Inteiro pode ser tratado como um espaço de reais, como descrito logo a seguir.

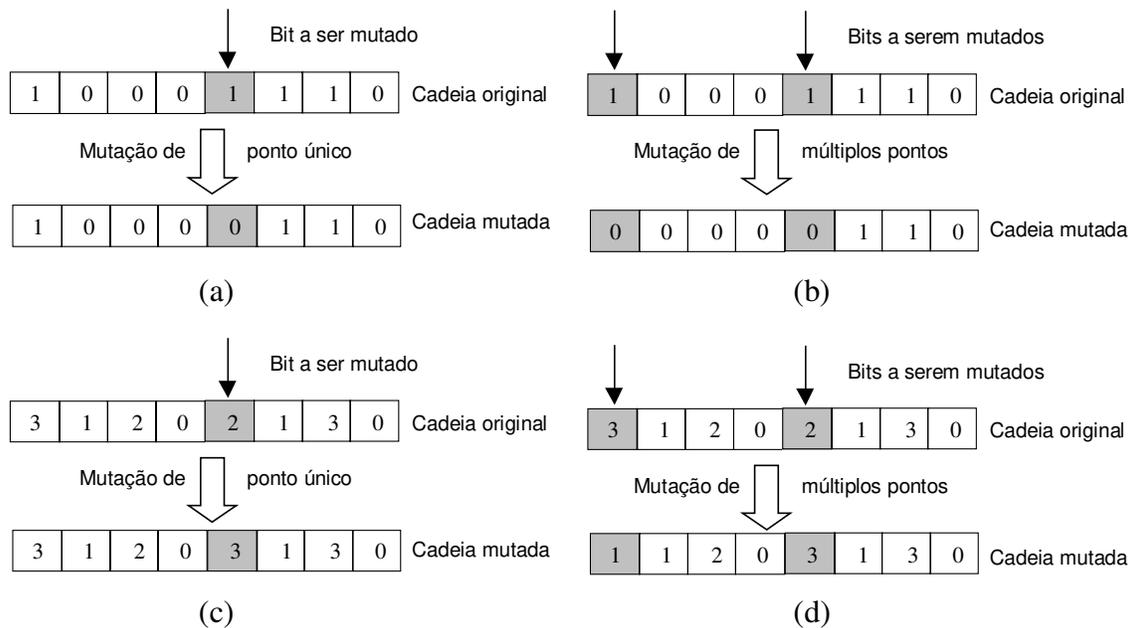


Figura 3.14. Mutações específicas para o espaço de Hamming. (a) Mutações de ponto único para cadeias binárias. (b) Mutações de múltiplos pontos para cadeias binárias. (c) Mutações de ponto único para $k = 4$. (d) Mutações de múltiplos pontos para $k = 4$.

Quando os elementos do vetor de atributos devem obedecer a certas restrições, é possível propor operadores de mutação específicos. Por exemplo, quando o que importa é a ordem, e os atributos existentes não podem ser modificados, basta escolhermos pares de pontos da cadeia e trocar os elementos destas posições, como ilustrado na Figura 3.15.

Por último, tem-se os casos dos espaços de valores reais, espaço Euclidiano e de Manhattan. A forma mais simples de efetuar uma mutação em um vetor de elementos com valores reais, é introduzindo uma pequena perturbação em um elemento específico deste vetor, respeitando os limites superior e inferior que cada coordenada deste vetor pode assumir. Esta perturbação pode ser determinada adicionando-se ou multiplicando-se um valor aleatório à coordenada atual do vetor (ver mutação indutiva, Seção 4.4.3).

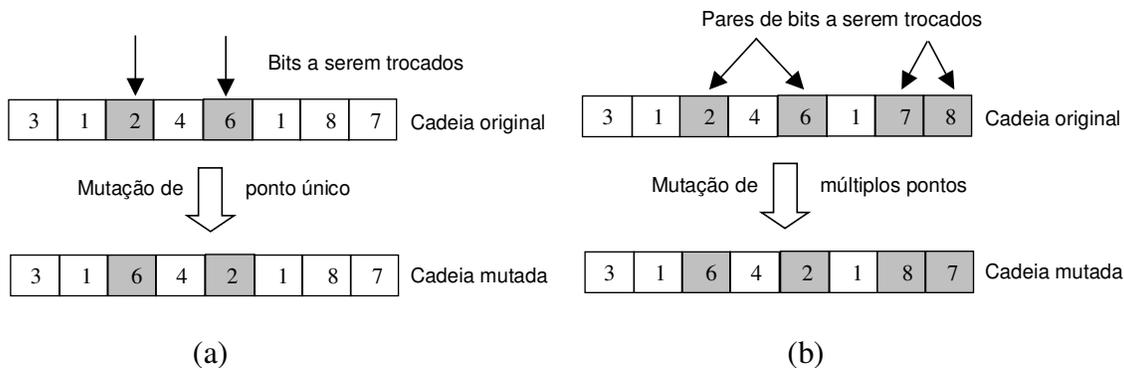


Figura 3.15. Mutações inversivas para o espaço Inteiro. (a) Mutações de ponto único. (b) Mutações de múltiplos pontos.

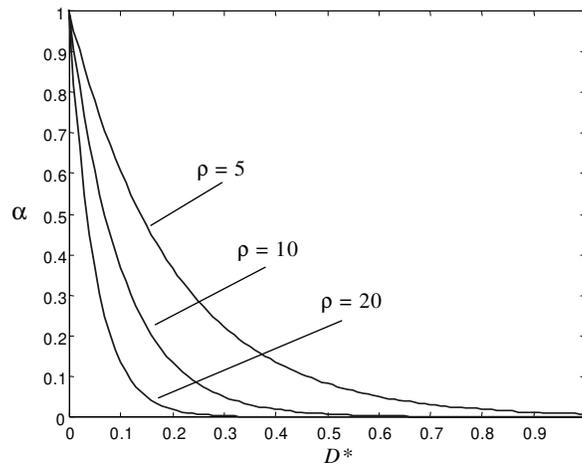


Figura 3.16. Relação entre a afinidade normalizada D^* do anticorpo e sua taxa de mutação α de acordo com a Equação (3.9), para diferentes valores do parâmetro ρ .

Em todos os casos, uma probabilidade de mutação p_m define a taxa com que cada posição da cadeia pode ser mutada. Sabendo que essa probabilidade é proporcional à afinidade, quanto maior a afinidade Ag-Ab, menor o valor de p_m e vice-versa. A idéia é preservar os indivíduos com alta afinidade ao mesmo tempo em que tentamos aumentar a afinidade daqueles indivíduos cuja afinidade é baixa. Trata-se de uma abordagem puramente imunológica, como visto na Seção 2.8.2.1. Esta característica pode ser simulada da seguinte forma: se a afinidade máxima normalizada de um anticorpo Ab_i do repertório Ab é $D^* = 1$, então sua taxa de mutação deve ser nula para que seu clone mantenha esta máxima afinidade. Caso contrário, quanto menor a afinidade, maior a taxa α de mutação. O inverso da função exponencial pode ser utilizado para estabelecer a relação entre α e o valor normalizado da afinidade D^* , como descrito pela Equação (3.9) e ilustrado na Figura 3.16.

$$\alpha = \exp(-\rho D^*), \quad (3.9)$$

onde ρ é um parâmetro de controle do amortecimento da função exponencial e D^* é o valor normalizado da afinidade D , que pode ser dado por $D^* = D/D_{\max}$.

3.4.3.5. Modelos Contínuos de Rede Imunológica

Niels K. Jerne (1974a) não foi apenas responsável pela introdução da teoria da rede imunológica, como discutido na Seção 2.10, ele foi também o primeiro pesquisador a apresentar uma formalização matemática para a sua teoria (Jerne, 1974b). Uma equação diferencial foi construída para descrever a dinâmica de um conjunto de linfócitos idênticos. Estes linfócitos idênticos foram denominados de linfócitos do tipo i , e N_i denotava a quantidade de linfócitos do tipo i . Os linfócitos do tipo i interagem com linfócitos de outros tipos como, por exemplo, linfócitos do tipo j e também anticorpos do tipo j via idiotopos e regiões de ligação. As interações podiam ser estimulatórias ou inibitórias. As interações de diferentes tipos de linfócitos levariam, naturalmente, à geração de uma rede de linfócitos.

Jerne sugeriu que a taxa de variação de um determinado tipo de linfócito seria dada por:

$$\frac{dL_i}{dt} = \alpha - \beta L_i + L_i \sum_{j=1}^m \varphi(E_j, K_j, t) - L_i \sum_{j=1}^m \psi(I_j, K_j, t), \quad (3.10)$$

onde α é a taxa com a qual os linfócitos vindos de outros compartimentos entram no conjunto i e β é a taxa (por linfócito) com a qual os linfócitos morrem ou saem do conjunto. As funções φ e ψ representam os sinais excitatório e inibitório da rede, respectivamente. A primeira soma varre todos os sinais excitatórios gerados por idiotopos nos conjuntos E_j , que são reconhecidos com constantes de associação K_j pelas regiões de ligação dos linfócitos do tipo i . A segunda soma representa todas as interações inibitórias geradas pelos linfócitos nos conjuntos I_j , cujas regiões de ligação reconhecem idiotopos nas células de L_i . Uma equação diferencial deste tipo é necessária para cada elemento da rede imunológica, e seria responsável por governar a quantidade de linfócitos excitatórios e inibitórios em cada conjunto. Um aspecto importante deste modelo é que um termo adicional é necessário para representar um estímulo externo, pois ele considera apenas os anticorpos.

Como discutido na Seção 3.3.1, Farmer *et al.* (1986, 1987) propuseram modelos binários para a teoria da rede imunológica. Como no modelo proposto por Jerne (1974b), e brevemente discutido acima, eles também consideraram um conjunto de equações diferenciais para representar as células e moléculas do sistema imunológico, juntamente com um limiar de afinidade, para remover anticorpos pouco estimulados, e operadores genéticos de recombinação (Holland, 1998), para simular as variações genéticas.

Neste modelo, as células e moléculas do sistema imunológico eram constituídas de cadeias binárias com comprimentos variáveis, como ilustrado na Figura 3.17. Uma molécula de anticorpo estava representada pelo seu epítopo (e , ou idiotopo i) e seu paratopo (p), concatenados em uma única cadeia binária. As cadeias podiam se ligar por regiões de complementaridade em qualquer alinhamento possível, modelando o fato de que duas moléculas podem reagir de várias formas distintas. A Equação (3.11) corresponde a uma matriz de especificidades de ligação, ou simplesmente matriz de especificidades, $m_{i,j}$

$$m_{i,j} = \sum_k G \left(\sum_n e_i(n+k) \wedge p_j(n) - \varepsilon + 1 \right), \quad (3.11)$$

onde $G(x) = x$ para $x > 0$ e $G(x) = 0$ nos outros casos, $e_i(n)$ é o n -ésimo bit do i -ésimo epítopo, $p_j(n)$ é o n -ésimo bit do j -ésimo paratopo, \wedge corresponde à medida de afinidade (ou regra de ligação) por complementaridade, Equação (3.3), e ε corresponde ao limiar de afinidade. Se a ligação ocorre em mais de uma alinhamento, seus pesos são somados, como no caso da Figura 3.7(e) para cadeias de comprimento distinto.

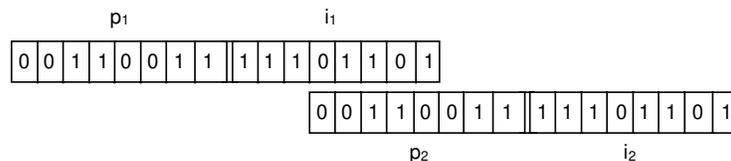


Figura 3.17. Cadeias binárias representando o epítopo (ou idiotopo) e paratopo dos anticorpos.

Para modelar a dinâmica da rede imunológica, foram assumidos N tipos de anticorpos com concentrações $\{x_1, \dots, x_N\}$ e M antígenos com concentrações $\{y_1, \dots, y_M\}$. A taxa de variação da concentração de anticorpos é dada por:

$$\dot{x}_i = c \left[\sum_{j=1}^N m_{j,i} x_i x_j - k_1 \sum_{j=1}^N m_{i,j} x_i x_j + \sum_{j=1}^M m_{j,i} x_i y_j \right] - k_2 x_i, \quad (3.12)$$

onde o primeiro termo representa a estimulação do paratopo de um anticorpo do tipo i por um idiotopo de um anticorpo j . O segundo termo representa a supressão de um anticorpo do tipo i quando seu idiotopo é reconhecido pelo paratopo do tipo j . O parâmetro c é uma taxa constante que depende da quantidade de colisões por unidade de tempo e da taxa de produção de anticorpos estimulados por uma colisão. A constante k_1 representa uma possível desigualdade entre estimulação e supressão. O terceiro termo modela as concentrações antigênicas e o último termo modela a tendência das células morrerem (taxa de mortalidade natural, k_2). Uma equação para medir a variação na concentração antigênica também foi apresentada

$$\dot{y}_i = -k_3 \sum_{j=1}^M m_{j,i} x_j y_i. \quad (3.13)$$

onde k_3 é uma constante arbitrária.

As Equações de (3.10) a (3.13) representam sistemas adaptativos, no sentido de que anticorpos que reconhecem antígenos ou outros anticorpos possuem seus clones amplificados, ou seja, se reproduzem. Por outro lado, aqueles anticorpos pouco ou não estimulados são eliminados do repertório. A fonte de novos anticorpos fornece ao sistema a capacidade de reconhecer antígenos diferentes daqueles para os quais o sistema está preparado. Como pode ser visto nestas equações, não é feita distinção alguma entre as células B, suas moléculas de superfície e os anticorpos livres.

Como último modelo baseado em equações diferenciais a ser apresentado nesta tese, destacamos a *segunda geração de redes imunológicas*, proposta por Varela & Coutinho (1991). De acordo com este modelo, é possível destacar três características importantes das redes imunológicas: estrutura, dinâmica e metadinâmica (Varela *et al.*, 1988; Varela & Coutinho, 1991), como discutido na Seção 2.10.

A formulação matemática de um modelo da segunda geração de redes imunológicas é descrita como a seguir. Cada clone de um conjunto de N elementos é indexado por i . Cada idiotopo existe ligado a uma superfície em quantidades $b_i(t)$ ou livre em quantidades $f_i(t)$. A afinidade entre dois idiotopos distintos é denotada por $m_{i,j}$, e a sensibilidade $\sigma_i(t)$ da rede ao i -ésimo idiotopo é dada por

$$\sigma_i(t) = \sum_{j=1}^N m_{j,i} f_j. \quad (3.14)$$

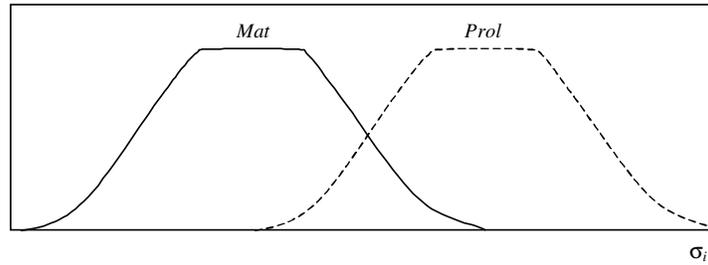


Figura 3.18. Funções de maturação (*Mat*) e proliferação (*Prol*).

A dinâmica das moléculas deste sistema de equações varia de acordo com a probabilidade de expansão das respectivas células B, que por sua vez é função do grau de conectividade de um idiótipo com a configuração atual da rede, ou seja, ela depende da sua sensibilidade. Assim, uma equação diferencial é proposta para descrever a dinâmica das moléculas livres:

$$\frac{df_i}{dt} = -f_i(k_1 + k_2\sigma_i) + k_3b_iMat(\sigma_i) \tag{3.15}$$

onde k_1 , k_2 e k_3 são constantes arbitrárias, b_i são as moléculas ligadas à superfície celular e $Mat(\cdot)$ é a função de maturação linfocitária como descrita na Figura 3.18.

Um anticorpo é produzido como resultado da maturação de uma célula B deste tipo de clone. De forma correspondente, a dinâmica das moléculas ligadas, ou seja a quantidade de células que transportam o idiótipo i em suas superfícies, decai a uma dada taxa e prolifera proporcionalmente ao seu grau de conectividade de rede. Um último termo $Meta[i]$ é adicionado para incluir células que serão recrutadas do repertório de células em repouso:

$$\frac{db_i}{dt} = b_i(k_4Prol(\sigma_i) - k_5) + Meta[i] \tag{3.16}$$

$Prol(\cdot)$ é a função de proliferação linfocitária, como descrita na Figura 3.18.

É possível apresentar uma estrutura geral para os modelos de rede imunológica descritos na forma (Perelson, 1989):

Taxa de variação da população	=	Entrada de novas células	-	Morte de células pouco estimuladas	+	Reprodução de células estimuladas	(3.17)
-------------------------------------	---	--------------------------------	---	--	---	---	---------------

onde o último termo inclui o reconhecimento de um idiótipo por um paratopo e a estimulação Ag-Ab.

3.5. Exemplos de Aplicações

O objetivo desta seção é apresentar algumas aplicações de sistemas imunológicos artificiais existentes na literatura e que podem ser caracterizados como engenharia imunológica, embora não tivessem recebido esta conotação por parte dos respectivos autores. As aplicações destacam como a metáfora imunológica foi utilizada, qual o tipo de codificação

(espaço de formas) adotado e quais foram os principais mecanismos imunológicos empregados, como o algoritmo de seleção negativa e os modelos dinâmicos de rede imunológica. O algoritmo de seleção clonal (CLONALG) é uma das contribuições desta tese e, portanto, suas aplicações serão ilustradas apenas no Capítulo 6.

Na próxima seção, será descrito um exemplo de SIA aplicado ao problema de segurança computacional, e nas seções seguintes serão apresentados três SIAs que podem ser classificados como exemplos de engenharia imunológica, pois envolvem basicamente o reconhecimento Ag-Ab e podem ser estudados e modelados empregando-se a estrutura formal para a engenharia imunológica, desenvolvida neste capítulo.

3.5.1. Um Sistema Imunológico Artificial

O SIA a ser descrito a seguir engloba diversos tipos celulares, o algoritmo de seleção negativa, sinais co-estimulatórios, linfocinas, etc. Além disso, são enfocados tanto o processo de reconhecimento antigênico (detecção de usuários não cadastrados, ou intrusos), quanto os processos de eliminação do intruso.

3.5.1.1. Segurança Computacional

Provavelmente, um dos domínios mais evidentes para se empregar mecanismos artificiais inspirados no sistema imunológico seja a área de segurança computacional, onde a analogia entre a proteção do organismo e a proteção de um ou mais computadores contra invasores externos (vírus, vermes, intrusos de rede, etc.) é intuitiva.

Tabela 3.3. Relação entre a segurança de redes de computadores e o sistema imunológico.

Sistema Imunológico	Segurança de Redes de Computadores
Próprio	Conjunto de pares de conexões TCP/IP que ocorrerem normalmente entre computadores
Não-próprio	Conjunto de conexões que não são normalmente observadas em uma rede LAN (rede local de computadores)
Células imunológicas	Detectores
Localidade no organismo	Máquina, ou <i>host</i> , de uma LAN na qual um conjunto de detectores é colocado em execução
Timócitos	Novos detectores gerados em cada conjunto de detectores
Linfócitos T maduros	Detectores que sobrevivem ao processo de seleção negativa
Nível de citocina	Limiar adaptativo
Competição pelo reconhecimento antigênico	Competição entre detectores por pacotes externos
Sinal co-estimulatório	Mensagem enviada via e-mail por um operador
Molécula de MHC	Máscaras de permutação

Para ilustrar a aplicação de um SIA ao problema de segurança computacional, escolhemos os trabalhos desenvolvidos por Hofmeyr & Forrest (1999, 2000), cujo objetivo principal é o de proteger uma rede de computadores de intrusos ilegais. A Tabela 3.3 ilustra a metáfora imunológica empregada no problema de segurança de redes de computadores. A interação de cada um destes elementos será explicada a seguir.

Uma conexão foi definida como sendo uma tripla de dados: o endereço da fonte IP, o endereço do destino IP e o serviço (ou porta) de comunicação entre os computadores. A Figura 3.19 ilustra a arquitetura do sistema imunológico artificial proposto. Foi definido um tipo básico de *célula detectora*, ou simplesmente *detector*, combinando propriedades de diversas células imunológicas. Este detector pode assumir vários estados, como descrito na 0. Foi utilizado o espaço de formas de Hamming para representar as moléculas (células) do sistema, onde cada detector é representado por uma cadeia binária de comprimento $L = 49$ bits, e uma pequena quantidade de estados (ver Figura 3.19). Para avaliar a afinidade entre as moléculas, os autores empregaram a regra dos r -bits consecutivos, ilustrada na Figura 3.7(b), Seção 3.4.2.

Os detectores foram agrupados em um conjunto por máquina (*host*). Dentro de cada conjunto de detectores, novos detectores, ou *timócitos*, são criados continuamente. Estes novos detectores permanecem imaturos por um determinado período de tempo, durante o qual eles têm a oportunidade de se ligar a qualquer conexão de rede atual. Este período é denominado de *tolerância*. Se um detector se liga enquanto ele está imaturo, então ele é eliminado. Esta etapa é feita de acordo com o algoritmo de seleção negativa, apresentado na Seção 3.4.3.2. Os detectores que sobrevivem a esta primeira fase de testes, tornam-se maduros. Cada detector maduro é válido e pode atuar de forma independente. O tempo que um detector leva para amadurecer pode ser visto como uma fase de aprendizagem. Ao final deste período de aprendizagem, se o detector não se ligar a um pacote ele será eliminado, porém se ele se ligar a uma quantidade suficiente de pacotes não-próprios, ele se tornará um *detector de memória*, com período de vida prolongado e limiar de ativação reduzido.

Como o processo de seleção negativa é incompleto, ou seja, nem todos os componentes próprios estão presentes durante a maturação dos linfócitos (detectores), alguns linfócitos podem detectar elementos próprios, causando um fenômeno equivalente a uma doença auto-imune. Para resolver este problema, foram criados dois mecanismos: um limiar de ativação e um limiar adaptativo.

Tabela 3.4. Estados que podem ser assumidos pelos detectores.

Estado do Detector	Característica
Timócito	Linfócito T imaturo que irá sofrer uma seleção negativa
Linfócito B virgem	Célula B que ainda não encontrou nenhum antígeno
Linfócito B de memória	Célula com vida longa e facilmente estimulável

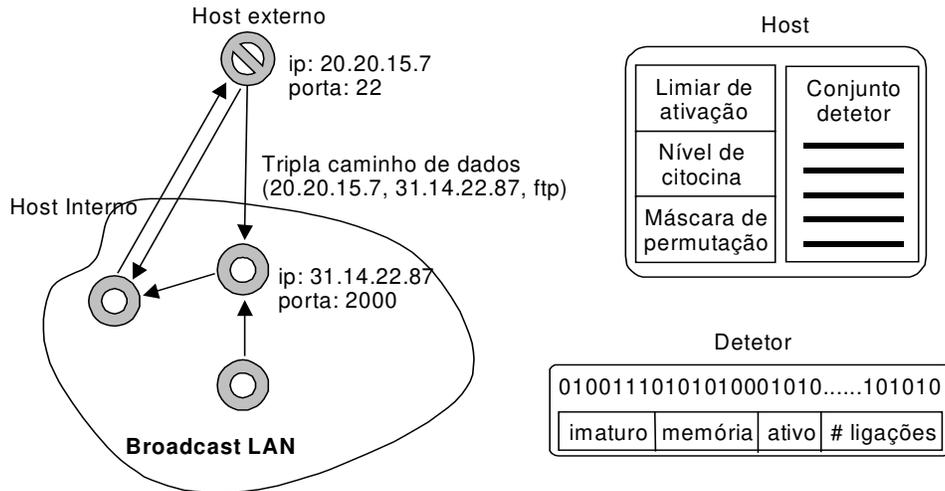


Figura 3.19. Arquitetura do sistema imunológico artificial proposto por Hofmeyr & Forrest (1999, 2000) para segurança computacional.

No caso do limiar de ativação, cada detetor armazena a quantidade de vezes que ele faz uma ligação, e caso essa ultrapasse um limiar pré-definido, um alarme é disparado e seu contador zerado. Sempre que um detetor se liga pela primeira vez, ele sensibiliza o sistema de detecção, de forma que todos os detectores naquela máquina se ativem mais facilmente no futuro. Este processo é chamado de limiar adaptativo e simula o nível de citocina do sistema.

Além da seleção negativa e da maturação das células virgens em células de memória, a maturação de afinidade também pode ser utilizada como mecanismo de aprendizagem neste sistema. Os autores propõem, embora ainda não tenha sido implementado, que detectores bem sucedidos podem se reproduzir (e migrar para outros computadores) e sofrerem uma mutação somática. Estes processos poderiam ser realizados pelo algoritmo de seleção clonal, proposto na Seção 3.4.3.3 e a ser detalhado no Capítulo 5.

Quando um detetor soa um alarme, existe uma chance de que este alarme seja falso (reação auto-imune). Por isso, antes de tomar uma decisão, o SIA espera um sinal co-estimulatório sob a forma de uma mensagem enviada por e-mail. A Figura 3.20 resume o ciclo de vida de um detetor.

Por último, são utilizados mecanismos de permutação para controlar como o pacote da rede é apresentado para o sistema de detecção, analogamente à apresentação peptídica pelas moléculas de MHC.

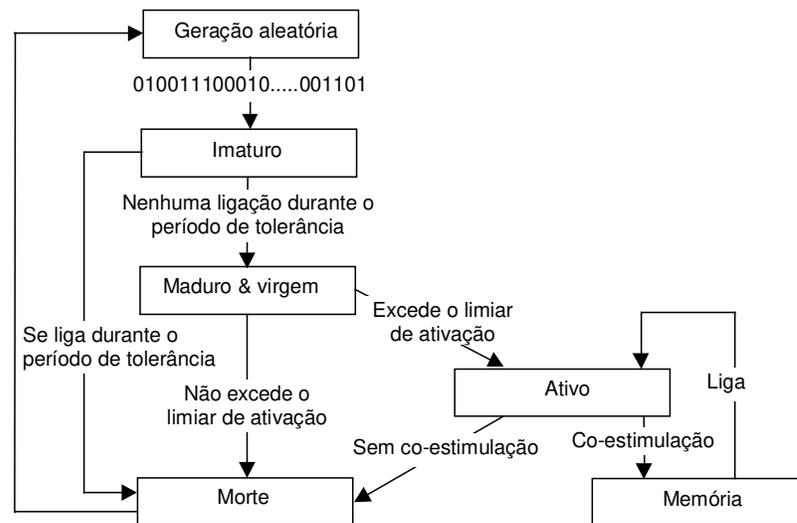


Figura 3.20. Ciclo de vida de um detector.

3.5.2. Aplicações de Engenharia Imunológica

Nos exemplos a serem descritos a seguir, é possível identificar claramente suas características de engenharia imunológica, onde o objetivo principal é efetuar um reconhecimento Ag-Ab (que pode ser um evento seletivo e/ou adaptativo). Existe apenas um tipo de elemento (anticorpo) no SIA, cuja interação com o antígeno é avaliada por alguma função (ou medida) de afinidade.

3.5.2.1. Scheduling

Hart *et al.* (1998) e Hart & Ross (1999, 2000) propuseram um sistema imunológico artificial capaz de produzir arranjos seqüenciais (planejamentos ou *schedules*) de forma rápida e eficiente para tratar conjuntos particulares de circunstâncias encontradas, por exemplo, em uma linha de montagem de uma fábrica.

Os autores fizeram a seguinte analogia entre um ambiente de planejamento de uma fábrica e o sistema imunológico biológico:

- Independentemente do corpo humano possuir uma quantidade limitada de material genético, e existir uma variedade quase infinita de possíveis patógenos, o sistema imunológico é capaz de reagir de forma rápida e eficiente aos antígenos previamente encontrados e a novos antígenos também.
- Enquanto uma fábrica está em operação, podem ocorrer variações em seu ambiente resultando na necessidade de se alterar os planejamentos previamente definidos. A natureza destas variações é ampla, podendo cobrir uma vasta gama de possibilidades. Algumas destas variações, ou eventos, ocorrem freqüentemente e são previsíveis, enquanto outras são imprevisíveis.

Tabela 3.5. Relação entre o problema de scheduling e o sistema imunológico.

Sistema Imunológico	Scheduling
Antígeno	Seqüência de tarefas em uma máquina particular, dado um determinado cenário
Universo antigênico	Conjunto de antígenos que representa os possíveis cenários/contingências que podem ocorrer
Anticorpo	Partes de uma seqüência de tarefas que são comuns a mais de uma máquina
Ligação (matching)	A ligação Ag-Ab ocorre se os planejamentos produzidos utilizando as instruções dadas por um Ab e a informação fornecida por um Ag resultarem na realização da tarefa antes do prazo final especificado pelo antígeno
Valor de ligação	Representa o número máximo de unidades de tempo que uma tarefa pode estar atrasada em um planejamento produzido pela interação Ag-Ab. Um planejamento perfeito possui um valor de ligação 0 e, quanto maior este valor, pior o planejamentos
Resposta Imunológica	Planejamento evoluído por um algoritmo genético

A Tabela 3.5 ilustra a relação entre a nomenclatura imunológica e o problema de planejamento abordado.

O modelo do sistema imunológico utilizado foi baseado nos trabalhos de Hightower *et al.* (1995, 1996), onde o material genético a ser empregado na produção das moléculas de anticorpo está armazenado em um conjunto de bibliotecas, seguindo o padrão descrito na Seção 3.4.3.1, Figura 3.11. Um algoritmo genético (GA) foi utilizado com sucesso para evoluir os componentes das bibliotecas imunológicas.

O algoritmo para a geração dos planejamentos é:

1. Selecione aleatoriamente um segmento de cada uma das bibliotecas genéticas para formar um anticorpo;
2. Aplique uma seqüência de mutações aleatórias ao anticorpo; e
3. Avalie o planejamento produzido aplicando as instruções presentes no anticorpo aos dados do antígeno.

O modelo proposto utiliza um espaço de formas do tipo Inteiro, onde as moléculas são representadas por permutações dos L elementos componentes do vetor. Como os anticorpos representam partes de uma seqüência de tarefas, ou seja, componentes de uma das bibliotecas, seu comprimento L_2 é menor ou igual ao comprimento L_1 dos antígenos, ou seqüência de tarefas de uma máquina ($L_2 \leq L_1$). Elementos do tipo ‘*don’t care*’ podem existir nos anticorpos objetivando facilitar a ligação de cadeias incompletas. Um ‘*don’t care*’ pode se ligar a qualquer elemento (tarefa).

Antígeno	1 2 3 4 5 6 7 8 9	Valor de ligação	
	3 4 6 7 8 *		1
	3 4 6 7 8 *		1
Anticorpo	3 4 6 7 8 *	10 + 1 = 11	
	3 4 6 7 8 *	15 + 1 = 16	Afinidade

Figura 3.21. Codificação inteira para as moléculas, juntamente com a função de matching das cadeias. Uma correspondência entre os números equivalem a um valor de ligação 5, enquanto um ‘don’t care’ adiciona 1 ao valor total da afinidade.

A afinidade entre duas moléculas é calculada utilizando-se uma função de afinidade, ou matching, que alinha as duas cadeias e verifica a correspondência elemento a elemento. Caso as cadeias possuam comprimentos distintos, então um valor de ligação é calculado para cada alinhamento possível, como ilustrado na Figura 3.21.

Os autores focalizaram o controle da evolução da população variando a diversidade dos anticorpos produzidos e a cobertura do universo antigênico. Eles demonstraram que os anticorpos evoluídos poderiam ser utilizados para reconstruir o conjunto original de planejamentos ao qual o sistema imunológico artificial foi exposto, e também construir planejamentos que não foram vistos anteriormente.

3.5.2.2. Mineração de Dados (Data Mining)

Mineração de dados, ou extração de conhecimentos de bancos de dados, refere-se ao processo de identificação de informação útil em um conjunto de dados geralmente de dimensão elevada e não-estruturado. Hunt *et al.* (1995) e Hunt & Fellows (1996) ilustraram como o sistema imunológico é inerentemente baseado em casos e como ele está apoiado em uma memória endereçável por conteúdo. Eles construíram uma memória de casos inspirada no sistema imunológico e na teoria da rede imunológica e mostraram como seu reconhecimento de padrões, aprendizagem e memória podem gerar um sistema de *raciocínio baseado em casos*, do tipo apresentado por Aamodt & Plaza (1994).

Estes trabalhos deixaram claro que, além da representação numérica das moléculas e células do sistema imunológico, qualquer outra forma de representação pode ser empregada, incluindo representação simbólica. Hunt *et al.* (1995) utilizaram uma *representação baseada em casos* para os antígenos e os anticorpos. A Figura 3.22 ilustra um possível modelo de representação baseada em casos: a afinidade é determinada fazendo-se a concordância (matching) de cada um dos casos do antígeno com o respectivo anticorpo. A medida de afinidade é dada pela Equação (3.4).

	Descrição	Data	Vôo	País	Meio	Origem	Destino	Valor R\$
Anticorpo:	Viagem	1996	212	Brasil	Avião	Porto Alegre	São Luís	1.000,00
Antígeno:	Viagem	2000	312	Brasil	Avião	Porto Alegre	São Luís	2.000,00
Avaliação:	1	0	0	1	1	1	1	0

Afinidade: $5 + 2^4 = 21$

Figura 3.22. Representação baseada em casos para as moléculas dos anticorpos e dos antígeno.

3.5.2.3. Robótica

Os principais trabalhos de Ishiguro *et al.* (1996, 1997) e Watanabe *et al.* (1998, 1999) enfocaram o problema de desenvolver um mecanismo dinâmico e descentralizado de geração de comportamentos baseado na teoria da rede imunológica. Nestas abordagens, a “inteligência” deveria emergir a partir de interações mútuas entre agentes (regras condição/ação ou comportamento simples), chamados de módulos competentes, e entre um robô e seu ambiente. As questões em aberto que eles tentaram responder foram referentes à construção de um mecanismo capaz de tomar a decisão mais apropriada dentre vários módulos e como definir estes módulos. O método foi avaliado em um problema de coleta de lixo utilizando um robô simulado, considerando a auto-suficiência, ou autonomia, do robô, como ilustrado na Figura 3.23(a). A Tabela 3.6 apresenta a relação entre o problema de tomada de decisão do robô e o sistema imunológico biológico.

O objetivo principal deste sistema imunológico artificial é de que o robô selecione adequadamente um módulo competente (anticorpo) de acordo com a situação detectada (antígeno apresentado). O robô, chamado de *imunóide*, consome uma energia E_m a cada iteração, perde uma energia E_l quando está carregando lixo e E_c quando colide com um obstáculo. O nível atual de energia é dado por:

$$E(t) = E(t - 1) - E_m - E_l - E_c, \tag{3.18}$$

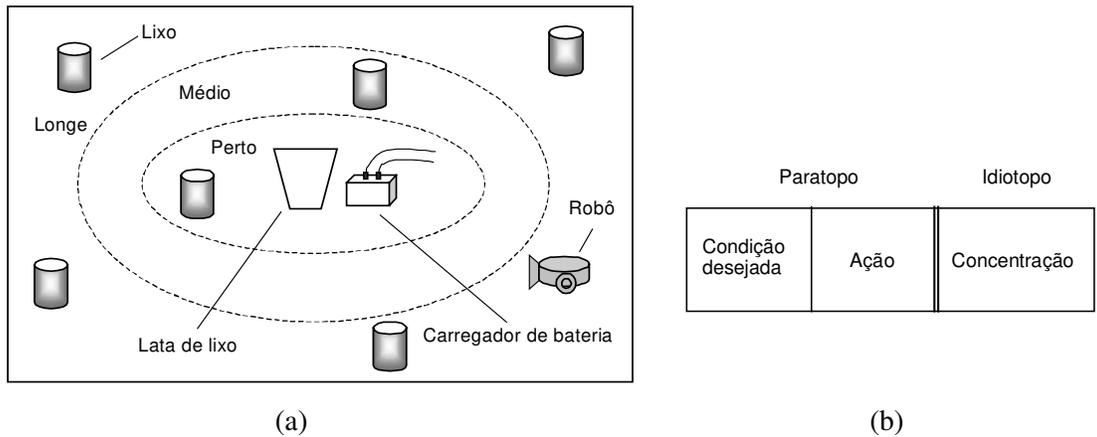


Figura 3.23. Problema de tomada de decisão. (a) Ambiente. (b) Estrutura de um anticorpo.

Tabela 3.6. Relação entre o problema de tomada de decisão e o sistema imunológico.

Sistema Imunológico	Tomada de Decisões
Antígeno	Situação atual do robô (como distância e direção relativas ao obstáculo, nível interno de energia, etc.), detectada por sensores
Anticorpo	Módulo competente (regra condição/ação)
Estimulação/Supressão da rede imunológica	Interações entre módulos competentes

A identidade de cada anticorpo é dada pela sua estrutura, representada pelo seu paratopo e idiotopo, como demonstrado na Figura 3.23(b). O nível de concentração de um anticorpo i , a_i , determina, via *roulette wheel* (ver Seção 4.4.3.2), a sua probabilidade de ser selecionado e varia de acordo com a seguinte equação dinâmica:

$$\frac{dA_i(t)}{dt} = \left[\alpha \sum_{j=1}^N m_{ji} a_j(t) - \alpha \sum_{k=1}^N m_{ik} a_k + \beta m_i - k_i \right] a_i(t) , \quad (3.19)$$

$$a_i(t+1) = \frac{1}{1 + \exp(0.5 - A_i(t+1))} ,$$

onde N é a quantidade de anticorpos, $m_{i,j}$ e m_i representam as afinidades Ab_j-Ab_i e Ab_i-Ag , respectivamente. O primeiro e o segundo termos do lado direito da Equação (3.19) denotam a estimulação e supressão entre anticorpos, respectivamente, o terceiro termo representa o reconhecimento antigênico, e o quarto termo a taxa de mortalidade natural. A expressão escolhida para $a_i(t)$ visa manter uma estabilidade (caráter assintótico) na concentração de anticorpos.

3.5.2.4. Contribuições da Tese em Aplicações Práticas

As ferramentas de engenharia imunológica propostas nesta tese foram aplicadas a diversos problemas de engenharia, principalmente:

- Geração de condições iniciais ótimas para algoritmos de computação inteligente, como redes neurais artificiais de múltiplas camadas e algoritmos genéticos;
- Aprendizagem de máquina;
- Estudo dos processos de auto-organização;
- Reconhecimento e classificação de padrões;
- Otimização de funções, enfocando os casos multimodal e combinatorial;
- Projeto automático de redes neurais artificiais;
- Análise de dados, enfocando o problema de clusterização; e
- Compressão de informações.

A descrição detalhada dos algoritmos será feita no Capítulo 5, enquanto as aplicações em engenharia serão devidamente apresentadas no Capítulo 6.

CAPÍTULO 4

SISTEMAS INTELIGENTES E O SISTEMA IMUNOLÓGICO

Este capítulo faz uma revisão de tópicos específicos da teoria de sistemas inteligentes, destacando as redes neurais artificiais e os algoritmos evolutivos. O objetivo é não apenas o de salientar similaridades e diferenças entre estas teorias, o sistema imunológico biológico (Capítulo 2) e os sistemas imunológicos artificiais (Capítulo 3), mas também o de apresentar alguns dos algoritmos que serão utilizados na comparação de desempenho com as ferramentas a serem propostas no Capítulo 5. Uma breve discussão sobre computação molecular e sistemas nebulosos também é feita. Serão apresentados pontos de vista cognitivos do sistema imunológico biológico, e a resposta imune adaptativa guiada pelo princípio da seleção clonal será caracterizada como uma microevolução segundo a teoria Darwiniana.

“A inteligência é uma faculdade especial, própria de certas classes de seres orgânicos, aos quais dá, com o pensamento, a sua vontade de agir, a consciência de sua existência e de sua individualidade, assim como os meios de estabelecer relações com o mundo exterior e de prover as suas necessidades” – A. Kardec

4.1. Introdução

Ao introduzir a proposta de engenharia imunológica na Seção 3.2, foram discutidos diversos componentes da teoria de sistemas inteligentes, como *inteligência artificial clássica*, *redes neurais artificiais*, *computação evolutiva*, *vida artificial*, *sistemas nebulosos* e *computação molecular* ou *computação de DNA*. Aprofundando-se no estudo de cada uma destas abordagens, seria possível perceber a existência de diversas características comuns entre elas, os sistemas imunológicos biológicos e os sistemas imunológicos artificiais. Entretanto, devido ao enfoque adotado nesta tese, daremos prioridade ao estudo das redes neurais artificiais (Seção 4.2) e da computação evolutiva (Seção 4.4). Alguns comentários serão feitos, na Seção 4.6, tentando relacionar o SI com os sistemas nebulosos e a computação molecular. Além disso, muitos dos algoritmos de treinamento de redes neurais e de computação evolutiva revisados neste capítulo serão empregados para comparações com as ferramentas a serem propostas no Capítulo 5.

Os leitores que detiverem conhecimentos sobre redes neurais artificiais e computação evolutiva podem ir direto às Seções 4.3, 4.5 e 4.6, sem perda da seqüência do trabalho.

4.2. Redes Neurais Artificiais

Definição 4.1: Uma *rede neural artificial* (RNA) é um sistema de processamento massivamente paralelo, composto por unidades simples com capacidade natural de armazenar conhecimento e disponibilizá-lo para uso futuro (Haykin, 1999).

As RNAs assemelham-se ao cérebro em dois aspectos:

- Elas extraem conhecimento do ambiente através de um processo de *aprendizagem* ou *treinamento*; e
- Os pesos das conexões entre os neurônios, conhecidos como *pesos sinápticos*, são utilizados para armazenar o conhecimento adquirido.

A Figura 4.1 apresenta um modelo de neurônio biológico com a seqüência de propagação dos sinais pela célula. Os neurônios artificiais também são chamados de *nós*, ou *unidades*. A natureza das RNAs faz com que seu estudo seja multidisciplinar, envolvendo pesquisadores de diversas áreas, como neurofisiologia, psicologia, física, computação e engenharia.

Neurofisiologistas e psicólogos estão particularmente interessados em compreender o funcionamento do sistema neural humano. As características de resposta a estímulos apresentada por neurônios individuais, bem como redes de neurônios, são alvo de estudo dos neurofisiologistas, enquanto os psicólogos estudam funções do cérebro tratando aspectos cognitivos e estão interessados na utilização de técnicas baseadas em redes neurais para criar modelos detalhados do comportamento humano.

Cientistas da área de computação têm em vista a construção de computadores dotados de processamento paralelo e distribuído, buscando superar as limitações impostas pelos computadores atuais, que realizam processamento serial simbólico.

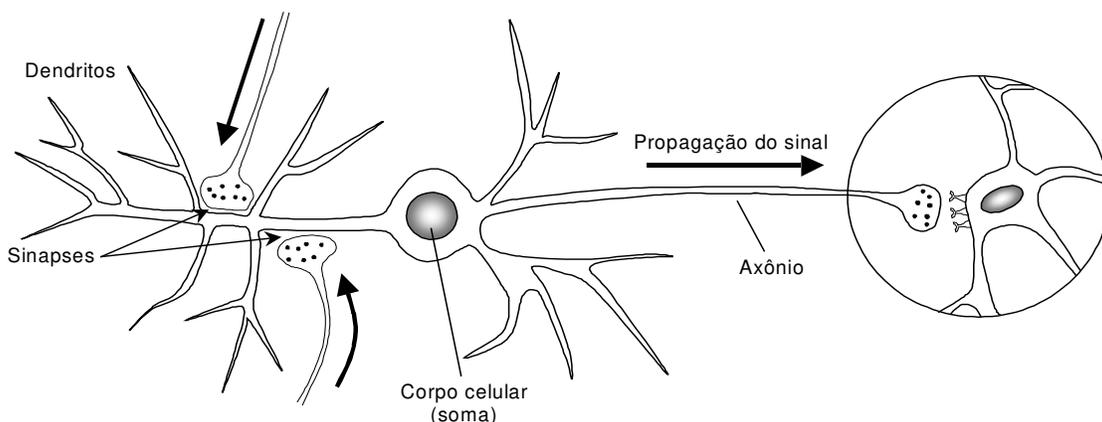


Figura 4.1. Célula neural biológica (as setas largas indicam a seqüência de propagação de sinais pelos neurônios).

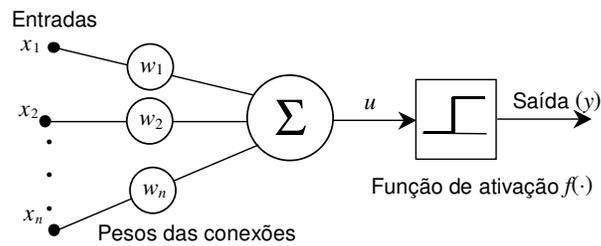


Figura 4.2. Representação funcional de um neurônio artificial.

Inspirados na habilidade apresentada pelos seres humanos e outros animais no desempenho de funções como o processamento de informação sensorial e a capacidade de interação com ambientes pouco definidos, os engenheiros estão preocupados em desenvolver sistemas artificiais capazes de desempenhar tarefas semelhantes. Habilidades como capacidade de processamento de informação incompleta ou imprecisa e generalização são propriedades desejadas em tais sistemas.

McCulloch & Pitts (1943) projetaram a estrutura que é conhecida como a unidade básica de uma rede neural. Estes pesquisadores propuseram um modelo de neurônio como uma unidade de processamento binária (Figura 4.2) e provaram que estas unidades são capazes de executar várias operações lógicas (OU, AND, etc.). Este modelo, apesar de muito simples, trouxe uma grande contribuição para as discussões sobre a construção dos primeiros computadores digitais, permitindo a criação dos primeiros modelos matemáticos de dispositivos artificiais que buscavam analogias biológicas. Matematicamente, o neurônio da Figura 4.2 pode ser expresso por:

$$y = f(u) = f(x_1w_1 + x_2w_2 + \dots + x_nw_n) = f(\mathbf{w}^T \mathbf{x}), \quad (4.1)$$

onde y é a saída do neurônio, u é a ativação do neurônio, $f(\cdot)$ sua função de ativação, x_i ($i = 1, \dots, n$) é o i -ésimo componente do vetor \mathbf{x} de entradas, e w_i ($i = 1, \dots, n$) é o i -ésimo componente do vetor \mathbf{w} de pesos do neurônio.

4.2.1. Características Principais

As redes neurais artificiais têm sido desenvolvidas como generalizações de modelos matemáticos de cognição humana ou neurobiologia, assumindo que:

- O processamento da informação ocorre com o auxílio de vários elementos chamados *neurônios*;
- Os sinais são propagados de um elemento a outro através de *conexões*;
- Cada conexão possui um *peso* associado, que, em uma rede neural típica, pondera o sinal transmitido; e
- Cada neurônio (ou unidade) possui uma *função de ativação* (geralmente não-linear), que tem como argumento a soma ponderada dos sinais de entrada, para determinar sua saída.

Uma rede neural pode ser caracterizada por três aspectos principais: (1) padrão de conexões entre as unidades (*arquitetura* ou *estrutura*), (2) método de determinação dos pesos das conexões (*algoritmo de treinamento* ou *aprendizagem*) e (3) *função de ativação*.

4.2.1.1. Arquitetura

A forma pela qual os neurônios de uma RNA estão estruturados (interconectados) está intimamente relacionada ao algoritmo de aprendizagem a ser utilizado para treiná-la. A classificação dos algoritmos de aprendizagem será apresentada na Seção 4.2.1.2. Nesta seção, será feita uma breve descrição das principais arquiteturas de redes neurais artificiais.

Em geral é possível distinguir três classes fundamentais de arquiteturas: *redes feedforward de uma única camada*, *redes feedforward de múltiplas camadas* e *redes recorrentes*.

4.2.1.1.1. Redes Feedforward de Uma Única Camada

No caso mais simples de redes em camadas (*layers*), tem-se uma camada de entrada com neurônios cujas saídas alimentam a última camada da rede. Geralmente, os neurônios de entrada são propagadores puros, ou seja, eles simplesmente repetem o sinal de entrada em sua saída distribuída. Por outro lado, as unidades de saída costumam ser unidades processadoras, como apresentado na Figura 4.2. A propagação de sinais nesta rede é puramente unidirecional (*feedforward*): os sinais são propagados apenas da entrada para a saída, e nunca vice-versa. Esta arquitetura está ilustrada na Figura 4.3(a) e a direção de propagação dos sinais na Figura 4.3(b).

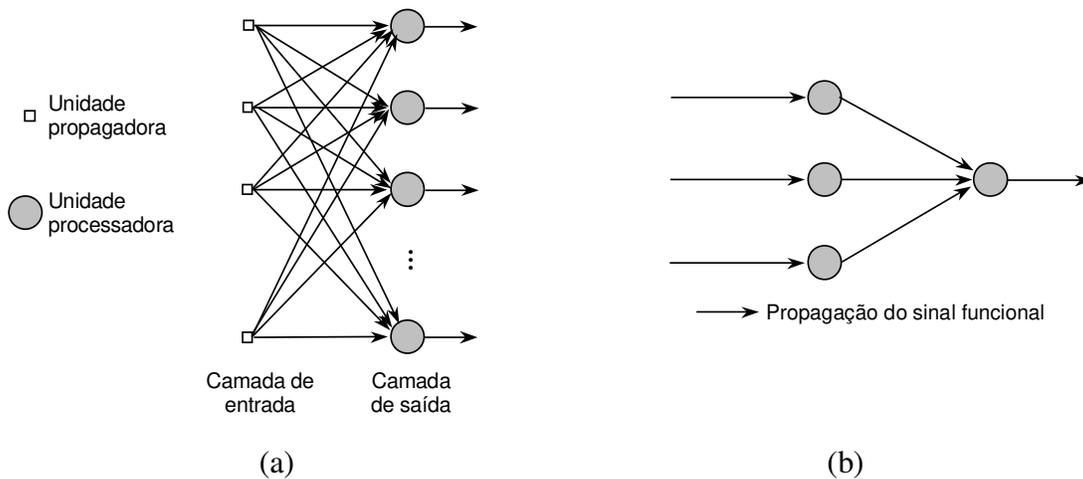


Figura 4.3. Redes neurais tipo *feedforward* com uma única camada de unidades processadoras. (a) Arquitetura. (b) Sentido de propagação do sinal funcional.

4.2.1.1.2. Redes Feedforward de Múltiplas Camadas

A segunda classe de rede feedforward se distingue pela presença de uma ou mais camadas intermediárias ou escondidas (camadas em que os neurônios são efetivamente unidades processadoras, mas não correspondem à camada de saída). Adicionando-se uma ou mais camadas intermediárias, aumenta-se o poder computacional de processamento não-linear e armazenagem da rede. O conjunto de saídas dos neurônios de cada camada da rede é utilizada como entrada para a camada seguinte. A Figura 4.4(a) ilustra uma rede feedforward de múltiplas (duas) camadas intermediárias.

As redes feedforward de múltiplas camadas, são geralmente treinadas usando o algoritmo de retro-propagação do erro (*error backpropagation*), embora existam outros algoritmos de treinamento. Este algoritmo requer a propagação direta (*feedforward*) do sinal de entrada através da rede, e a retro-propagação (propagação reversa, ou *backpropagation*) do sinal de erro, como ilustrado na Figura 4.4(b).

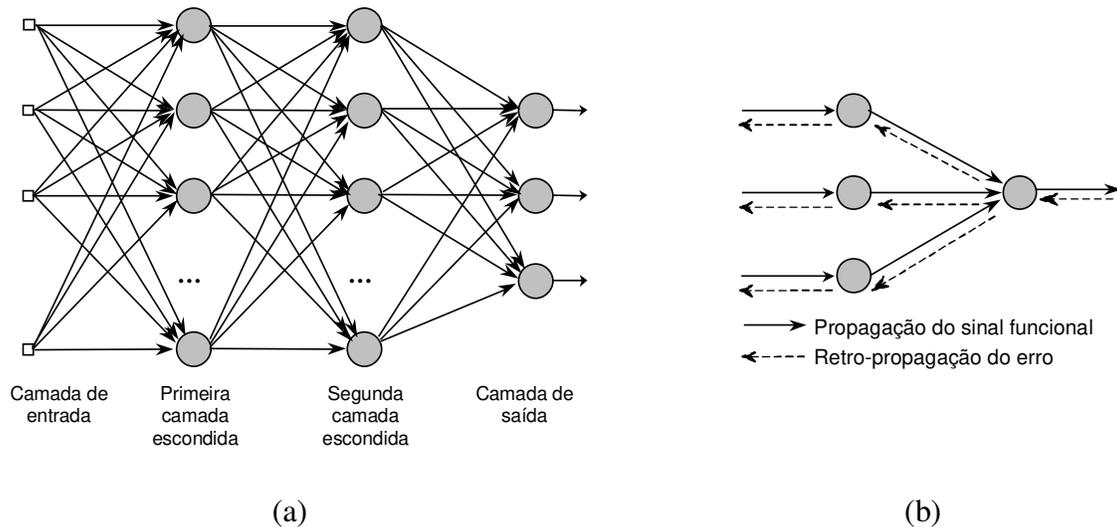


Figura 4.4. Redes neurais tipo *feedforward* com múltiplas camadas. (a) Arquitetura. (b) Sentido de propagação do sinal funcional e do sinal de erro.

4.2.1.1.3. Redes Recorrentes

As redes recorrentes distinguem-se das redes feedforward pela existência de pelo menos um laço (*loop*) de recorrência (*feedback*). Por exemplo, uma rede recorrente pode consistir de uma única camada de neurônios com cada neurônio alimentando seu sinal de saída de volta para a entrada de todos os outros neurônios, como ilustrado na Figura 4.5. O laço de recorrência possui um grande impacto na capacidade de aprendizagem e no desempenho da rede (dos Santos & Von Zuben, 2000). Além disso, este laço envolve a utilização de ramos particulares compostos de unidades de retardo (z^{-1}), resultando em um comportamento dinâmico não-linear, assumindo que a rede possui componentes não-lineares.

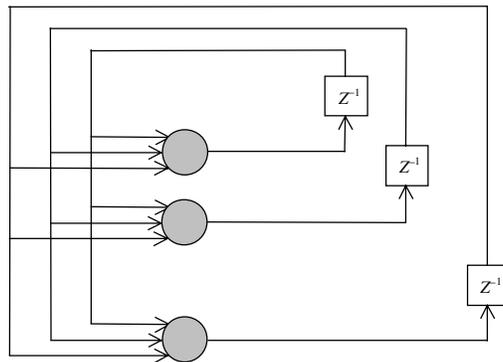


Figura 4.5. Arquitetura recorrente de rede sem nenhuma camada intermediária.

4.2.1.2. Métodos de Aprendizagem

A capacidade de *aprendizagem* é uma das características marcantes das RNAs. Uma rede neural aprende, basicamente, através de um processo iterativo de ajuste de pesos e limiares (bias). Atualmente, existem processos mais sofisticados de aprendizagem (ou *treinamento*), que são capazes de ajustar não apenas os pesos da rede, mas também sua arquitetura e as funções de ativação dos neurônios (Von Zuben, 1996, Kwok & Yeung, 1997, de Castro *et al.*, 1999a,b; de Castro & Von Zuben, 1999c).

Definição 4.2: *Aprendizagem* (ou *treinamento*) é o processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um mecanismo de apresentação de estímulos fornecidos pelo ambiente no qual a rede está inserida. O tipo de treinamento é definido pela forma na qual os parâmetros são modificados (Haykin, 1999).

Esta definição de aprendizagem implica na seguinte seqüência de eventos:

- Apresentação de estímulos à rede neural;
- Alteração dos parâmetros livres da rede; e
- Novo padrão de resposta ao ambiente.

Os principais paradigmas de aprendizagem são: (1) supervisionada, (2) não-supervisionada, e (3) por reforço.

4.2.1.2.1. Aprendizagem Supervisionada

Trata-se de um paradigma de aprendizagem, no qual um *supervisor* possui conhecimento sobre o ambiente em que a rede está inserida. Este conhecimento está representado sob a forma de um conjunto de amostras de *entrada-saída*. O ambiente, por sua vez, é desconhecido. A Figura 4.6 ilustra esta abordagem. Os parâmetros da rede são ajustados pela combinação do sinal de entrada com um sinal de erro, que é a diferença entre a saída desejada e a fornecida pela rede.

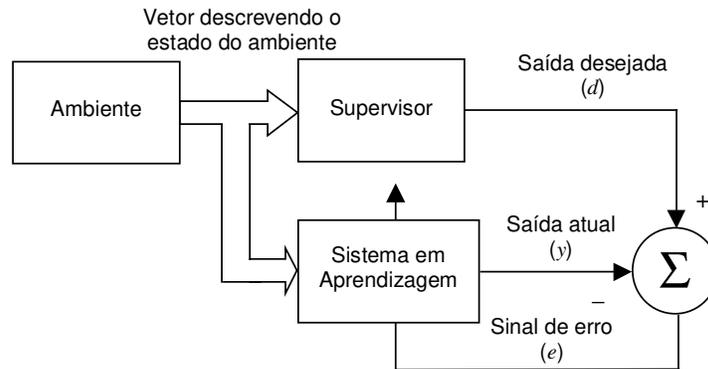


Figura 4.6. Diagrama de blocos do processo de aprendizagem supervisionada.

Seja t o índice que denota tempo discreto ou, mais precisamente, o intervalo de tempo do processo iterativo responsável pelo ajuste de pesos do neurônio k . O único sinal de saída $y_k(t)$, do neurônio k , é comparado com uma *saída desejada*, denominada $d_k(t)$. Conseqüentemente, um sinal de erro $e_k(t)$ é produzido:

$$e_k(t) = d_k(t) - y_k(t). \quad (4.2)$$

4.2.1.2.2. Aprendizagem Não-Supervisionada

No processo de *aprendizagem não-supervisionada* ou *auto-organizada*, não existe um supervisor para avaliar o desempenho da rede em relação ao conjunto de treinamento (Figura 4.7), ou seja, os dados são não-rotulados. A rede se adapta a regularidades estatísticas dos dados de entrada, desenvolvendo a habilidade de criar representações internas para codificar características da entrada e, assim, gerar novas classes automaticamente. Geralmente os algoritmos auto-organizados utilizam aprendizagem competitiva.

Na *aprendizagem competitiva*, os neurônios de saída da rede competem entre si para se tornarem ativos. Um único neurônio de saída é ativado a cada iteração. Esta característica torna o algoritmo apropriado para descobrir características estatísticas salientes, que podem ser utilizadas para classificar um conjunto de padrões de entrada.

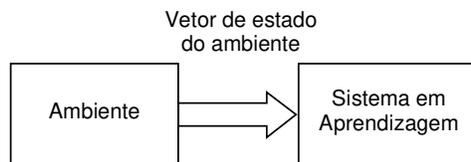


Figura 4.7. Diagrama de blocos dos processos auto-organizados.

Existem três elementos básicos para uma regra de aprendizagem competitiva:

- Um conjunto de neurônios iguais, exceto pelos pesos das conexões;
- Um limite imposto ao peso de cada neurônio; e
- Um mecanismo de competição entre os neurônios. Aquele que vencer a competição é chamado de vencedor (*winner-takes-all*).

Neurônios individuais aprendem a se especializar em grupos (ou clusters) de padrões similares, tornando-se *detectores de características* para diferentes classes de padrões de entrada.

Em sua forma mais simples, uma rede competitiva possui uma única camada de neurônios de saída, totalmente interconectados. Também existem conexões laterais entre os neurônios, como indicado na Figura 4.8, capazes de efetuar uma inibição lateral entre os neurônios vizinhos.

Para um neurônio k ser o vencedor, seu campo induzido local v_k em relação a um determinado padrão x deve ser o maior de toda a rede. O sinal de saída y_k do neurônio vencedor k é setado em 1, e o sinal de saída de todos os outros neurônios que perderam a competição é setado em 0

$$y_k = \begin{cases} 1 & \text{se } v_k > v_j \quad \forall j, j \neq k \\ 0 & \text{demais casos} \end{cases}, \quad (4.3)$$

onde o campo induzido local v_k representa a ação combinada das entradas positivas e laterais do neurônio.

Se um neurônio não responde a um determinado padrão de entrada, nenhuma aprendizagem ocorre. Por outro lado, se um neurônio ganha a competição, um ajuste $\Delta w_{k,j}$ é aplicado ao vetor de pesos $w_{k,j}$ deste neurônio vencedor

$$\Delta w_{k,j} = \begin{cases} \alpha(x_j - w_{k,j}) & \text{se } k \text{ vence a competição} \\ 0 & \text{se } k \text{ perde a competição} \end{cases}, \quad (4.4)$$

onde α é a taxa de aprendizagem. Esta regra possui o efeito geral de mover o vetor de pesos $w_{k,j}$ do neurônio vencedor k na direção do correspondente padrão de entrada x .

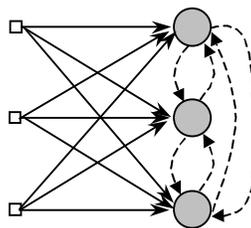


Figura 4.8. Arquitetura de uma rede competitiva simples com conexões diretas (feedforward) excitatórias da entrada para a saída e conexões laterais inibitórias (setas tracejadas).

4.2.1.2.3. Aprendizagem por Reforço

A *aprendizagem por reforço* enfatiza a aprendizagem do indivíduo através da interação direta com o ambiente, sem se basear em uma supervisão ou um modelo completo deste ambiente, visando minimizar um índice escalar de desempenho. A Figura 4.9 ilustra um tipo de aprendizagem por reforço (Haykin, 1999) baseado em um crítico que converte um *signal primário de reforço* recebido do ambiente em um sinal de reforço de maior qualidade chamado *signal de reforço heurístico*. O objetivo da aprendizagem é *minimizar uma esperança do erro acumulado*, ao invés de simplesmente tratar o erro atual.

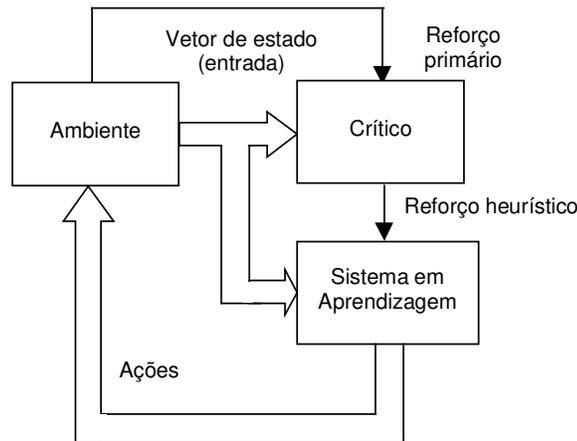


Figura 4.9. Diagrama de blocos do processo de aprendizagem por reforço.

4.2.2. Modelos Neurais e Algoritmos de Treinamento

Existem diversas arquiteturas e algoritmos de treinamento distintos para as redes neurais artificiais. Dentre elas destacam-se o perceptron de uma ou mais camadas, as redes neurais de função de base radial (RBF – *radial basis function*), as redes de Kohonen (1982) e as redes de Hopfield. Esta seção tem por objetivo descrever apenas as redes do tipo RBF, as de Kohonen (auto-organizadas) e as redes de Hopfield discretas, que foram utilizadas nesta tese principalmente para efeito de comparação de desempenho em aplicações em que foram utilizados os algoritmos de engenharia imunológica a serem propostos.

4.2.2.1. Redes Neurais de Base Radial (RBF)

Uma rede neural com função de ativação de base radial (RBF – *radial basis function*) pode ser vista como uma rede feedforward de múltiplas camadas (Seção 4.2.1.1.2), contendo três camadas com papéis completamente distintos (Broomhead & Lowe, 1988). A camada de entrada é composta por unidades sensoriais que conectam a rede a seu ambiente. A segunda camada (única camada intermediária) aplica uma transformação não-linear do espaço de entrada para o espaço intermediário. As unidades intermediárias não-lineares têm ativação local e suas respostas correspondem às ativações das funções de base radial.

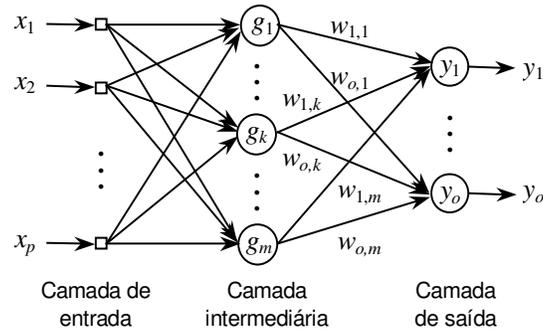


Figura 4.10. Rede neural com funções de ativação de base radial (RBF).

A camada de saída é linear, fornecendo cada saída da rede como uma combinação linear das respostas das unidades intermediárias (Haykin, 1999; Karayiannis & Mi, 1997). A Figura 4.10 ilustra uma rede RBF com múltiplas saídas (note a presença de um único conjunto de pesos ajustáveis, \mathbf{w} , entre as unidades intermediárias e de saída).

Para um vetor de entrada p -dimensional $\mathbf{x} = (x_1, \dots, x_p)$, onde $\mathbf{x} \in \mathbf{X} \subset \mathfrak{R}^p$, a saída da rede RBF pode ser determinada da seguinte forma

$$y_i = \mathbf{w}_i^T \mathbf{g} = \sum_{j=1}^m w_{i,j} g_j, \quad i = 1, \dots, o \quad (4.5)$$

onde $\mathbf{w}_i = [w_{i,1}, \dots, w_{i,m}]^T$, $i = 1, \dots, o$, são os vetores de peso para cada neurônio i de saída, $\mathbf{g} = [g_1, g_2, \dots, g_m]^T$ é o vetor com os valores de ativação das funções base, e o é a quantidade de neurônios de saída da rede. Dado um conjunto de vetores protótipos $\mathbf{c}_j \in \mathfrak{R}^p$, $j = 1, \dots, m$, a saída de cada RBF é

$$g_j = h(\|\mathbf{x} - \mathbf{c}_j\|, \rho_j), \quad j = 1, \dots, m \quad (4.6)$$

onde $h(\cdot)$ é a função base e $\|\cdot\|$ é uma norma, geralmente Euclidiana, definida no espaço de entrada. As escolhas de h consideradas na literatura incluem $h(r) = r$ (linear), $h(r) = r^3$ (cúbica), $h(r) = r^2 \log r$ (thin-plate spline), $h(r) = \exp(-r^2/2)$ (Gaussiana), $h(r) = \text{sqrt}(r^2+1)$ (multiquádrica), and $h(r) = 1/\text{sqrt}(r^2+1)$ (multiquádrica inversa), onde em todos os casos r é o raio escalonado $\|\mathbf{x} - \mathbf{c}_j\|/\rho_j$ (Broomhead & Lowe, 1988; Poggio & Girosi, 1990; Chen *et al.*, 1991; Elanayar & Shin, 1994; Whitehead & Choate, 1996).

As funções de base radial foram originalmente propostas na implementação de métodos de interpolação, e suas propriedades como interpolantes têm sido extensivamente estudadas (Poggio & Girosi, 1990; Freeman & Saad, 1995). Neste contexto, uma função de mapeamento $y: \mathfrak{R}^N \rightarrow \mathfrak{R}$ satisfazendo a Equação (4.5), para $o = 1$, deve ser determinada. Considere um conjunto de N dados amostrais $\{\mathbf{x}_i \in \mathfrak{R}^p | i = 1, \dots, N\}$. Se os valores desejados de y são conhecidos para todas as N amostras, ou seja $\{d_i \in \mathfrak{R} | i = 1, \dots, N\}$, então cada função base $h(\cdot)$ pode ser centralizada em uma destas amostras. Assim, existem tantos centros (vetores protótipos) \mathbf{c}_j quanto dados de treinamento $m = N$ (Powell, 1987). Em notação matricial

$$\mathbf{H} \mathbf{w} = \mathbf{d}, \quad (4.7)$$

onde os vetores \mathbf{d} e \mathbf{w} de dimensão $N \times 1$ representam o vetor de saídas desejadas e o vetor de pesos de saída, respectivamente, e \mathbf{H} é uma matriz $N \times N$, denominada matriz de interpolação, que assume a forma:

$$\mathbf{H} = \begin{bmatrix} h(\|x_1 - c_1\|, \rho_1) & \cdots & h(\|x_1 - c_m\|, \rho_m) \\ \vdots & \ddots & \vdots \\ h(\|x_N - c_1\|, \rho_1) & \cdots & h(\|x_N - c_m\|, \rho_m) \end{bmatrix}. \quad (4.8)$$

A solução para o problema de interpolação da Equação (4.7) é dada por

$$\mathbf{w} = \mathbf{H}^{-1} \mathbf{d} \quad (4.9)$$

Resta-nos garantir que a matriz \mathbf{H} seja não-singular, tal que a inversa de \mathbf{H} exista.

O teorema de Michelli (1986) estabelece como único pré-requisito para a não-singularidade de \mathbf{H} (existência de \mathbf{H}^{-1}) que as N amostras sejam diferentes, independentemente dos valores de N e p . Este teorema é válido para as funções multiquádrica e sua inversa e para as demais funções de base radial (RBF).

De acordo com Broomhead & Lowe (1988), o processo de interpolação descrito acima pode não ser uma boa estratégia para o treinamento de redes do tipo RBF para algumas classes de problemas, devido a uma má capacidade de generalização da rede resultante. Além disso, se N é muito grande, e/ou existem muitos dados redundantes, as chances de se obter uma matriz \mathbf{H} não-inversível também são grandes. A restrição de ter tantas funções de base radial quanto amostras de treinamento torna o problema sobre-determinado. Para superar estas dificuldades computacionais, a complexidade da rede deve ser reduzida, requerendo uma aproximação para uma solução regularizada (Poggio & Girosi, 1990). A abordagem empregada envolve a busca de uma solução sub-ótima em um espaço de dimensão reduzida. Um novo conjunto de funções base $\{g_j, j = 1, \dots, m_1, m_1 < N\}$, assumidas linearmente independentes, deve ser definido. O novo conjunto de centros $\{c_j | j = 1, \dots, m_1\}$ deve ser determinado e o vetor \mathbf{w}^* , assumindo $\sigma = 1$, deve ser tal que resolva o seguinte problema de otimização $\min_{\mathbf{w}} \|\mathbf{H}\mathbf{w} - \mathbf{d}\|^2$. Desconsiderando o uso do parâmetro de regularização, a solução \mathbf{w}^* do problema para $m_1 < N$, é simplesmente dada por

$$\mathbf{w}^* = \mathbf{H}_1^+ \mathbf{d} = (\mathbf{H}_1^T \mathbf{H}_1)^{-1} \mathbf{H}_1^T \mathbf{d}, \quad (4.10)$$

onde \mathbf{H}_1 é dada pela Equação (4.11), e \mathbf{H}_1^+ é a pseudo-inversa da matriz \mathbf{H}_1 (Broomhead & Lowe, 1988).

$$\mathbf{H}_1 = \begin{bmatrix} h(\|x_1 - c_1\|, \rho_1) & \cdots & h(\|x_1 - c_{m_1}\|, \rho_{m_1}) \\ \vdots & \ddots & \vdots \\ h(\|x_N - c_1\|, \rho_1) & \cdots & h(\|x_N - c_{m_1}\|, \rho_{m_1}) \end{bmatrix}. \quad (4.11)$$

Haykin (1999) sugere que este método é relativamente insensível à utilização do parâmetro de regularização, desde que haja uma escolha adequada dos centros das funções de base radial.

4.2.2.1.1. Centros Fixos

A abordagem mais simples para o treinamento de uma rede RBF é assumir fixos os parâmetros \mathbf{c}_j e ρ_j , $j = 1, \dots, m_1$, das funções de base radial. As localizações dos centros devem ser escolhidas de alguma forma, geralmente baseada nos dados de treinamento. Assumindo uma função Gaussiana como função de base radial e tomando $\rho_j = \rho$, $j = 1, \dots, m_1$, onde ρ é uma medida de dispersão associada ao desvio padrão dos centros, resulta

$$h(\|\mathbf{x} - \mathbf{c}_j\|^2, \rho) = \exp\left(-\frac{1}{2} \frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{\rho^2}\right), \quad j = 1, \dots, m_1, \quad (4.12)$$

onde \mathbf{x} é o vetor de entradas, \mathbf{c}_j a localização do j -ésimo centro e ρ é o desvio padrão das RBFs Gaussianas, dado pela seguinte equação

$$\rho = \frac{d_{\max}}{\sqrt{2m_1}} \quad (4.13)$$

onde m_1 é a quantidade de funções de base radial e d_{\max} é a distância máxima entre os centros.

A Equação (4.13) garante que as RBFs individuais não serão muito achatadas ou alargadas (Haykin, 1999). Assim, os únicos parâmetros que ainda precisam ser definidos são os pesos da camada de saída da rede. Um procedimento direto para isso é empregar o método da pseudo-inversa apresentado na Equação (4.10).

4.2.2.1.2. Seleção Auto-Organizada de Centros

Uma outra abordagem para estimar a localização apropriada dos centros das RBFs na camada intermediária é utilizando uma etapa de auto-organização antes de estimar os pesos da camada de saída da rede. O algoritmo de clusterização k -means tende a posicionar os centros das RBFs nas regiões do espaço de entrada onde dados significativos estão presentes. A determinação de um valor adequado para m_1 , a quantidade de centros, pode requerer experimentação. Sejam $\{\mathbf{c}_k(t)\}_{k=1}^{m_1}$ os centros das RBFs na iteração t . Assim, o algoritmo k -means opera como a seguir:

1. *Inicialização*: Escolha diferentes valores aleatórios para os centros $\mathbf{c}_k(t)$, $k = 1, \dots, m_1$.
2. *Amostragem*: Escolha aleatoriamente uma amostra $\mathbf{x}(t)$ a partir dos dados de entrada.
3. *Seleção*: Determine o índice k do centro mais próximo ao vetor $\mathbf{x}(t)$ de acordo com

$$k(\mathbf{x}(t)) = \arg \min_k \|\mathbf{x}(t) - \mathbf{c}_k(t)\|, \quad k = 1, \dots, m_1, \quad (4.14)$$

onde $\mathbf{c}_k(t)$ é o centro da k -ésima RBF na iteração t .

4. *Atualização*: Ajuste os centros das RBFs de acordo com a seguinte regra

$$\mathbf{c}_k(t+1) = \begin{cases} \mathbf{c}_k(t) + \alpha[\mathbf{x}(t) - \mathbf{c}_k(t)], & k = k(\mathbf{x}) \\ \mathbf{c}_k(t), & \text{outros casos} \end{cases} \quad (4.15)$$

onde $\alpha \in (0,1)$ é a taxa de ajuste.

5. *Ciclo*: Repita os Passos de 2 a 5 até que nenhuma variação significativa em c_k , $k = 1, \dots, m_1$, seja observada.

Uma limitação do algoritmo k -means é que ele só é capaz de atingir soluções ótimas locais, dependendo da escolha inicial dos centros. Assim, uma escolha inicial inapropriada dos centros pode resultar em centros posicionados em regiões do espaço com baixa densidade de dados.

4.2.2.2. Redes de Kohonen

As redes que possuem aprendizado competitivo ou auto-organizado (Seção 4.2.1.2.2) são chamadas *redes competitivas* ou *auto-organizadas*, onde os neurônios de saída competem entre si para estarem ativos ou não. Apenas um neurônio de saída, ou um por grupo, é ativado a cada iteração. As redes auto-organizadas de Kohonen (1982) são caracterizadas pela formação de um mapa topográfico dos padrões de entrada e, por isso, são denominadas de *mapas auto-organizáveis de Kohonen* (SOM – *self-organizing maps*).

No SOM, a localização espacial dos neurônios auto-organizados vai indicar características intrínsecas aos padrões de entrada. Seu principal objetivo é a transformação adaptativa e ordenada de um conjunto de dados de entrada em um mapa (*grid*) uni- ou bidimensional de saída. Os dados de entrada que apresentam semelhanças entre si são agrupados em regiões do mapa de saída, formando classes ou agrupamentos denominados *clusters*.

Durante o processo de auto-organização do mapa, a unidade do cluster cujo vetor de pesos mais se aproxima do vetor dos padrões de entrada é escolhida como sendo a *vencedora*. A *unidade vencedora e suas unidades vizinhas têm seus pesos atualizados*. Além disso, estas redes possuem parâmetros variáveis com o tempo, como a taxa de aprendizagem α , e um raio de vizinhança N_R que indica quais unidades serão atualizadas simultaneamente.

A Figura 4.11 apresenta arquiteturas típicas de um SOM, considerando configurações de vizinhança unidimensional e bidimensional, embora dimensões mais elevadas possam ser consideradas (Costa, 1999). Além disso, dada a dimensão, a quantidade de unidades ou neurônios de saída pode ser arbitrada e mantida fixa, ou então definida automaticamente pelo algoritmo de treinamento (Fritzke, 1993; Cho, 1997; de Castro & Von Zuben, 1999b).

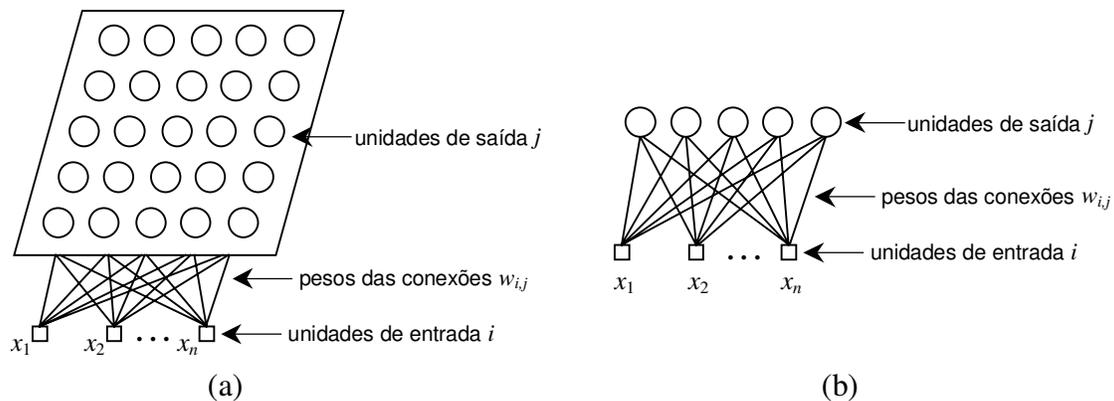


Figura 4.11. Arquiteturas típicas de um SOM. (a) Bidimensional. (b) Unidimensional.

O algoritmo de treinamento padrão do SOM é apresentado abaixo (Kaski, 1997):

1. Inicialização e definições de parâmetros
 - Inicialize os pesos $w_{i,j}$;
 - Defina N_R e α .
2. Enquanto a condição de parada é falsa, faça:
 - 2.1. Para cada j determine:
 - 2.1.1. $J = \arg \min_j \{ \|w_j - x_l\| \}$
 - 2.1.2. $\forall j \in N_R$ de J , e $\forall k$:

$$w_{j,k}(t+1) = w_{j,k}(t) + \alpha [x_{l,k} - w_{j,k}(t)]$$
 - 2.2. Atualize a taxa de aprendizagem
 - 2.3. Reduza o raio de vizinhança
3. Teste a condição de parada.

A taxa de aprendizagem decresce lentamente com o tempo. A formação de um mapa ocorre em duas fases: 1) formação inicial da ordem correta do mapa, e 2) convergência final.

No processo de treinamento, N_R é decrementado de 1 após um determinado número de iterações até que N_R seja igual a zero. Aqui, diz-se que ocorreu uma iteração quando o vetor de padrões x_1, x_2, \dots, x_N tiver sido apresentado uma vez. Os pesos das conexões para o nó J e todos os nós em sua vizinhança definidos por N_R , como mostrado na Figura 4.12, são atualizados. Outros critérios de vizinhança podem ser definidos.

Um método de visualização de um SOM treinado, denominado matriz de distâncias unificadas, ou matriz-U, foi desenvolvido por Ultsch (1993), com o objetivo de permitir a detecção visual das relações topológicas dos neurônios.

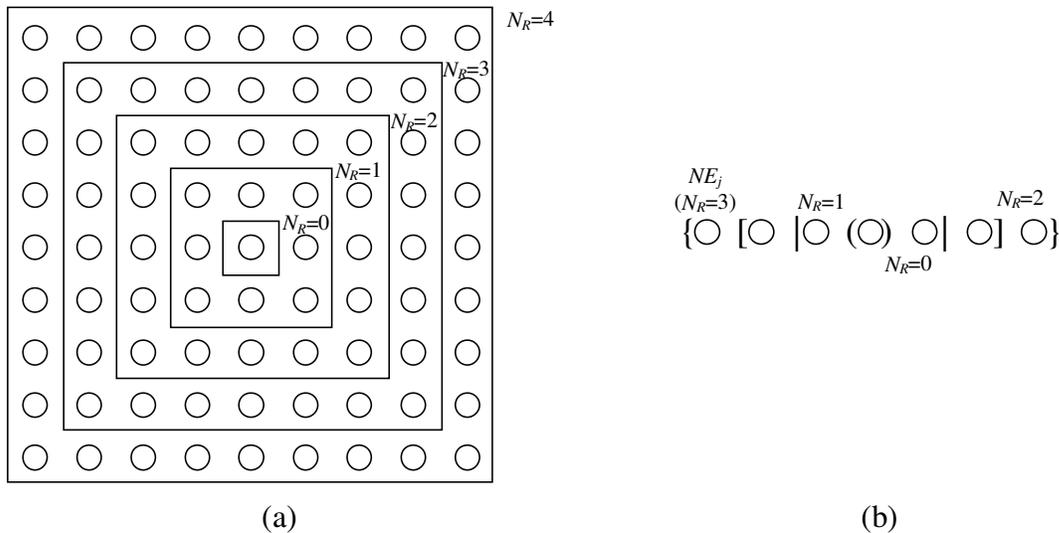


Figura 4.12. Vizinhos do nó j , $NE_j(N_R)$ para arquitetura bidimensional (a), e unidimensional (b).

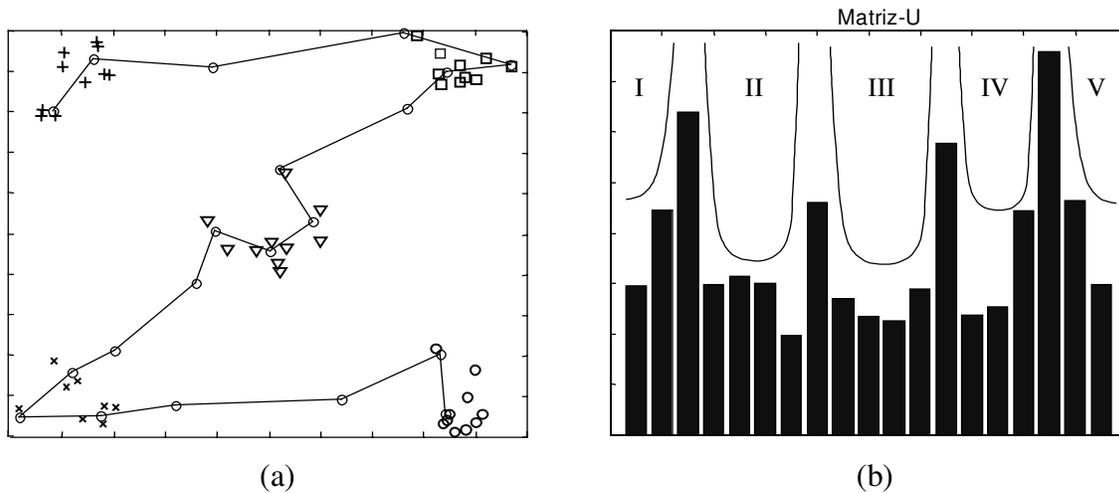


Figura 4.13. Representação da matriz-U. (a) Dados de treinamento e topologia final de uma rede de Kohonen (SOM) para vizinhança unidimensional. (b) Matriz-U.

A idéia básica da matriz-U é usar a mesma métrica utilizada no treinamento da rede para calcular as distâncias entre os pesos dos neurônios adjacentes. Para o caso unidimensional, o resultado é um *histograma* no qual a altura de cada barra corresponde à distância do neurônio a seu vizinho mais próximo. Desta forma, é possível visualizar similaridades entre neurônios e portanto determinar a quantidade de clusters existente no conjunto amostral. A Figura 4.13 ilustra a matriz-U para um problema simples de classificação, onde o conjunto de treinamento possui cinco clusters linearmente separáveis. Cada vale da Figura 4.13(b) corresponde a um cluster distinto.

4.2.2.3. Redes de Hopfield Discretas

As redes de Hopfield (1982, 1984) possuem suas raízes na Física Estatística, mais do que em observações biológicas. As redes de Hopfield são totalmente recorrentes e podem ser descritas como um *sistema dinâmico cujo espaço de estados contém um conjunto de pontos fixos (estáveis) que representam as memórias fundamentais do sistema*. Consequentemente, estas redes possuem uma propriedade emergente que ajuda a recuperar informações e lidar com ruídos. Devem ser vistas como uma *memória associativa* ou *memória endereçável por conteúdo*.

Memórias associativas fornecem mecanismos para armazenar e recuperar dados baseados em seu conteúdo, ao invés de baseado em um endereço de armazenamento. A informação está distribuída pelo sistema (rede neural).

A rede de Hopfield originalmente proposta utiliza uma matriz simétrica de pesos, de modo que as realimentações existentes geram um comportamento dinâmico não-linear caracterizado apenas pela presença de pontos fixos. Após um transitório inicial, qualquer estado inicial em que a rede se encontre será conduzido até um dos pontos fixos da dinâmica, geralmente associados a memórias fundamentais. A validade deste argumento

pode ser mostrada através de uma *função de energia* e da garantia de que a dinâmica é sempre dissipativa.

Nesta arquitetura não há neurônios intermediários (Figura 4.5). O estado completo da rede é visível em todos os momentos. A restauração de um padrão é feita colocando o padrão desconhecido na rede como um estado inicial e permitindo à dinâmica estabilizar. Iterações sucessivas do algoritmo mudam o estado da rede de forma a colocá-la próxima a uma das memórias fundamentais. O processo de restauração não é perfeito. Em alguns casos, a rede pode estabilizar em um padrão que não é uma das memórias fundamentais, conhecido como *estado espúrio*. Estas memórias são caracterizadas, por exemplo, por combinações lineares de um número ímpar de memórias fundamentais.

Função de Energia

Considere uma rede de Hopfield com pesos simétricos $w_{i,j} = w_{j,i}$ e $w_{i,i} = 0$, e seja s_i o estado do neurônio i , onde $i = 1, \dots, N$. A *função de energia* da versão discreta da rede de Hopfield considerada aqui é definida por

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{j,i} s_i s_j. \quad (4.16)$$

A variação da energia ΔE devido a uma variação Δs_j do estado do neurônio j é dada por

$$\Delta E = -\Delta s_j \sum_{i=1}^N w_{ji} s_i. \quad (4.17)$$

Os mínimos locais da superfície de energia correspondem a *atratores* do espaço de estados, que são as memórias da rede. Para garantir a emergência da memória associativa, duas condições devem ser satisfeitas:

1. As memórias fundamentais armazenadas pela rede serem estáveis; e
2. Os padrões estáveis possuem uma base de atração (região de influência).

Algoritmo de Treinamento

O algoritmo apresentado aqui é colocado para padrões bipolares $\{-1, +1\}$, mas a função de ativação pode ser modificada facilmente para aceitar padrões binários $\{0, 1\}$.

1. Utilize a *regra de Hebb* (Equação (4.18)) ou a regra da projeção (Equação (4.19)) para inicializar os pesos (Haykin, 1999):

$$\mathbf{W} = \frac{1}{N} \sum_{i=1}^p \mathbf{v}_i \mathbf{v}_i^T - \frac{p}{N} \mathbf{I}. \quad (4.18)$$

$$\mathbf{W} = \mathbf{P}(\mathbf{P}\mathbf{P}^+)^{-1} \mathbf{P}^T. \quad (4.19)$$

2. Para cada padrão de entrada \mathbf{x} , faça
 - 2.1. Coloque a ativação inicial igual ao vetor de entrada externo \mathbf{x}

$$y = x$$
 - 2.2. Para cada saída y_i
 - 2.2.1. Calcule a entrada de rede

$$y_{in_i} = \sum_j y_j w_{j,i}$$

2.2.2. Determine a nova ativação (sinal de saída)

$$y_i = \begin{cases} 1 & \text{se } y_{in_i} > 0 \\ y_i & \text{se } y_{in_i} = 0 \\ -1 & \text{se } y_{in_i} < 0 \end{cases}$$

2.2.3. Propague o valor de y_i para todas as outras unidades.

2.3. Teste a convergência

4.3. O Sistema Imunológico e o Sistema Nervoso Central

A interação do sistema imunológico (SI), o sistema nervoso central (SNC) e o sistema endócrino (SE), os dois últimos algumas vezes denominados conjuntamente de sistema *neuroendócrino* (SNE), é importante para a garantia da integridade e homeostase do organismo. Nesta seção, pretendemos discutir brevemente sobre a relação entre estes sistemas, mais especificamente entre o SI e o SNC. Para isso, serão discutidos aspectos cognitivos do sistema imunológico e serão apresentadas similaridades e diferenças entre o SI e o SNC, e entre a engenharia imunológica e as RNAs apresentadas nas Seções 3.2 e 4.2, respectivamente.

A influência do SNC no SE pode ser claramente observada em situações como, por exemplo, a necessidade de se correr ao banheiro em um momento de estresse psicológico, como uma prova ou a defesa de uma tese. Sensações físicas de desconforto são comuns quando as pessoas estão submetidas a crises de ansiedade. Estudos tentando relacionar anomalias como câncer e outras doenças com o temperamento e condições psicológicas do indivíduo levaram ao surgimento da *psiconeuroimunologia* (Ader *et al.*, 1991), que aborda de forma integrada as diversas funções dos sistemas nervoso e imunológico, tentando esclarecer as doenças conhecidas como *psicossomáticas* (Bonamin, 1994).

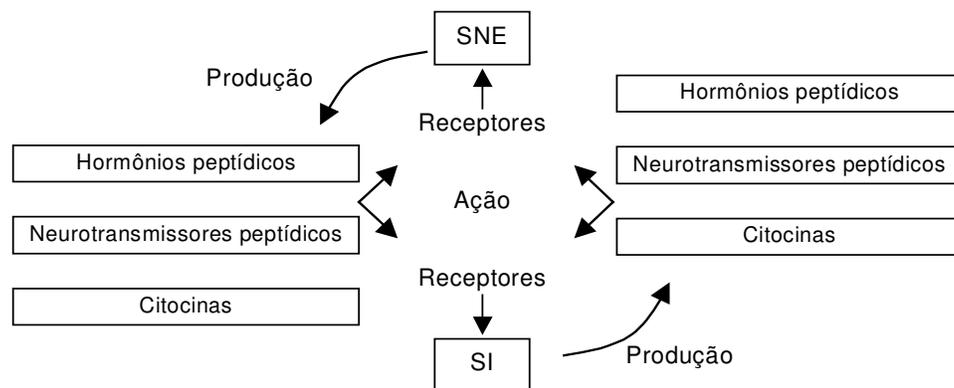


Figura 4.14. Um mecanismo de comunicação entre o SI e o SNE através do compartilhamento de moléculas sinalizadoras e seus receptores (adaptado de Blalock, 1994).

Uma análise em sentido oposto, ou seja, de como o sistema imunológico influencia o sistema neuroendócrino, também tem sido de fundamental importância para a compreensão da relação entre eles. Várias observações, como a alteração de padrões de atividade neural e hormonais seguindo a inoculação antigênica em indivíduos saudáveis, permitiram a apresentação de um modelo molecular bidirecional de comunicação entre o SNE e o SI, como apresentado na Figura 4.14 (Blalock, 1994). Sem nos prendermos aos muitos detalhes desta interação, objetivamos salientar que os mesmos sinais moleculares e seus receptores podem ser utilizados tanto para a regulação do próprio sistema em questão quanto para a comunicação entre eles.

4.3.1. Aspectos Cognitivos

Existe uma grande dificuldade por parte da comunidade científica na obtenção de um consenso sobre o significado de palavras e expressões como *cognição*, *intencionalidade* e *mente*. Os conceitos variam para diferentes ramos de pesquisa, como psicologia, filosofia e neurociência. Esta seção tem por objetivo discutir sucintamente algumas abordagens sobre o que se entende por estes temas, enfocando o problema da cognição, e também apresentar argumentos da literatura sugerindo que o sistema imunológico possui propriedades suficientes para ser caracterizado como um sistema cognitivo. Além disso, o leitor será posicionado acerca de minha opinião pessoal quanto à visão cognitiva do sistema imunológico, a qual também é defendida por diversos pesquisadores, principalmente imunologistas. É indiscutível que os pontos de vista a serem apresentados, assim como a teoria cognitiva do sistema imunológico a ser defendida nesta seção, também não são de consenso geral e muito menos completas. Apesar disso elas visam fornecer subsídios para um estudo mais amplo da intitulada *cognição imunológica*.

O *cognitivismo* surgiu no ano de 1956 a partir de dois encontros realizados em Cambridge e Dartmouth, nos quais as principais idéias que forneceriam as bases da *ciência cognitiva* moderna foram propostas (Gardner, 1995). A ciência cognitiva é o resultado de um esforço multidisciplinar que traz para a psicologia, ciência da computação e inteligência artificial vários aspectos de neurobiologia, lingüística e filosofia (Edelman, 1992). Ela se baseia no conceito de *representações mentais* e em um conjunto de suposições coletivamente denominadas de *posição funcional*. Partindo deste ponto de vista, as pessoas se comportam de acordo com o conhecimento gerado por representações mentais simbólicas. Assim, a *cognição* consiste na manipulação destes símbolos. Os fenômenos psicológicos são descritos por processos funcionais. A eficácia de tais processos reside na possibilidade de interpretar itens como símbolos, de formas abstratas e bem definidas, de acordo com um conjunto de regras. Este conjunto de regras constitui o que é conhecido como *sintaxe*. O exercício das regras sintáticas é uma forma de *computação*, no sentido amplo de manipulação simbólica, ou operação com símbolos, de acordo com um procedimento definido. Tais processos bem definidos constituem representações *semânticas*, que indicam formas precisas de especificação do que os símbolos representam para o mundo.

Sob esta perspectiva computacional, a proposta central da cognição é de que a inteligência, incluindo a humana, pode realmente ser definida como uma forma de computação com representações simbólicas (Varela *et al.*, 1991). A noção central, neste caso, é a de

representação ou *intencionalidade*. *Sistemas intencionais* têm como função legitimar predicados mentais, assim como uma máquina de Turing deve estabelecer regras para a interpretação de objetos (Dennett, 1978). O argumento cognitivo defende que o comportamento inteligente pressupõe a habilidade de representar o mundo de alguma forma. Provavelmente, um dos aspectos mais controversos da cognição esteja na sua premissa de que a única forma de se obter um comportamento inteligente ou uma intencionalidade é através da suposição de que as ações ocorrem através da manipulação de representações que são fisicamente executáveis por um código simbólico no cérebro, máquina ou outro sistema. Assim, um dos problemas que precisam ser resolvidos é como relacionar a intencionalidade, ou os estados representativos, com as mudanças físicas ocorridas durante uma ação.

A expressão *cognitivo* chegou à imunologia vinda da psicologia, na qual ela se refere às funções superiores da mente, incluindo o reconhecimento de objetos, identificação do organismo e *intencionalidade* (Mitchison, 1994). O objetivo inicial era enfatizar que o sistema imunológico sabe o que ele está procurando quando encontra um antígeno, ou seja, sua *representação* (ou *organização*) *interna* lhe fornece uma certa intencionalidade. Alguns imunologistas como I. Cohen, F. Varela, A. Coutinho e N. Jerne tratam a *cognição imunológica* como sendo baseada principalmente no conceito de próprio/não-próprio. Outros autores mais conservadores preferem definir a imunologia como uma *ciência de reconhecimento* (Edelman, 1975, 1992 – Cap. 8), ou visualizar o sistema imunológico como um *sistema sensorial* (Blalock, 1994) e não um sistema cognitivo. Tada (1997) apresentou um paradigma diferente, propondo que sistemas como o imunológico e o nervoso central podem ser denominados de “supersistemas”.

A idéia de *reconhecimento* surgiu na imunologia desde seu início. A identificação dos elementos estranhos ao organismo implicitamente requer que algum componente esteja efetuando a identificação, ou reconhecimento, destes elementos. O reconhecimento é um evento *perceptivo* e, portanto, deve se apoiar em algum aparato cognitivo (Tauber, 1994, 1997). Esta visão reflete a riqueza escondida em termos como reconhecimento, aprendizagem e memória, características pertinentes ao sistema imunológico.

Cohen (1992a) define um sistema cognitivo como a seguir:

Definição 4.3: Um *sistema cognitivo* é capaz de extrair informações e experiências de dados de entrada através da manipulação de informações já contidas no próprio sistema, ou seja, ele age através de um senso de direção ou intencionalidade. Sendo assim, os sistemas cognitivos não são processadores passivos ou armazenadores de informação, e sim projetados para manipular informações particulares do domínio no qual eles operam.

Niels K. Jerne (1974a,b, 1984, 1985) é considerado o verdadeiro autor do modelo cognitivo do sistema imunológico (Tauber, 1997). A visão cognitiva da teoria da rede imunológica apresentada na Seção 2.10, é fundamentada em duas premissas:

- O sistema imunológico é constituído por um universo de imagens que só são reconhecidas por estarem expressas em uma linguagem conhecida pelo sistema;

- O sistema imunológico se auto define, ou seja, ele é projetado para conhecer a si próprio.

Dessa forma, os elementos próprios provocam um determinado tipo de resposta, enquanto os não-próprios induzem outra resposta, baseada não na natureza intrínseca do não-próprio, mas no fato de que o sistema imunológico enxerga o antígeno externo no *contexto* de invasão ou degeneração. O elemento chave é o anticorpo, que atua como um antígeno através de domínios idiotípicos. Existe portanto, uma *imagem interna* do universo antigênico. O reconhecimento mútuo entre os elementos do sistema imunológico (linfócitos B e anticorpos) forma uma vasta rede interconectada de elementos que se comunicam entre si, a chamada *rede idiotípica* ou *rede imunológica*.

Quando o sistema imunológico é visto como uma entidade cognitiva, ele representa uma complementação do sistema nervoso central (ou simplesmente sistema neural). Certamente, para que o SI reconheça um determinado elemento, ele, primeiramente, precisa percebê-lo e em seguida decidir que atitude tomar. Trata-se, portanto, de um tipo de modelo cognitivo, onde os sistemas imunológico e neural podem ser vistos de forma análoga. Cada um possui características perceptivas: a capacidade de discernir entre o universo interno e o externo. O processamento de informação é fundamental para a operação de ambos e as respectivas propriedades perceptivas estão ligadas a mecanismos efetores. Além das analogias funcionais, a crescente evidência da interdependência deles, através de moléculas mensageiras, neurotransmissores e hormônios, já foi brevemente discutida no início da Seção 4.3. Adotando-se o modelo em rede proposto por Jerne (1974a), as similaridades entre estes sistemas aumentam ainda mais.

Cohen (1992a,b) também apresentou uma *abordagem contextualista* para a cognição imunológica. Considerando os sistemas cognitivos como aqueles capazes de extrair informações e experiências de dados de entrada através da manipulação de informações já contidas no sistema, Cohen (1992b) sugeriu o chamado *homúnculo imunológico*:

Definição 4.4: O *homúnculo imunológico* é a imagem interna do próprio adquirida pelo reconhecimento primário dos antígenos próprios, tanto no timo, como na periferia (Cohen, 1992b).

Esta definição está fundamentada na idéia de que o sistema imunológico será capaz de realizar sua tarefa mais eficientemente através da coleta e processamento de informações caso ele seja dotado de uma representação interna de seu ambiente, definindo o foco de atenção junto ao qual ele deve operar. Os antígenos serão reconhecidos como não-próprios pois eles são apresentados em um *contexto* que denuncia sua patologia. A autoimunidade é tratada como uma característica normal do sistema imunológico, que constantemente tenta identificar e monitorar os elementos do hospedeiro. Se estes antígenos próprios forem alterados de forma contextual, seus significados mudam e uma resposta imunológica é iniciada. Assim, o próprio não é mais uma entidade, e sim emerge dinamicamente em um processo de auto-identificação, que muda continuamente ao longo da vida do indivíduo.

De forma similar, uma teoria baseada na definição do próprio e da rede imunológica foi desenvolvida por Coutinho e seus colaboradores (Coutinho *et al.*, 1984; Varela *et al.*, 1988; Coutinho, 1989; Varela & Coutinho, 1991). A postura adotada sugere que as propriedades

globais do sistema imunológico, como tolerância e distinção próprio/não-próprio, não podem ser compreendidas através da análise de componentes individuais (imunologia celular e molecular). Foi proposto que características essenciais das redes imunológicas, como estrutura (conectividade) e dinâmica, juntamente com a teoria da seleção clonal, fornecem uma abordagem poderosa para o estudo de alguns aspectos cognitivos do sistema imunológico, como a memória por exemplo. A memória imunológica é tida como uma característica clonal, pelo menos no contexto de respostas secundárias, e o reconhecimento antigênico (diretamente relacionado à memória) é provavelmente a propriedade imunológica de maior apelo cognitivo.

Baseado em um ponto de vista conceitualmente diferente, Blalock (1994) aborda o sistema imunológico como um mecanismo sensorial, assim como o sistema neural, porém ele assume que a cognição só pode ser atribuída a estímulos do tipo fisiológicos, emocionais, etc. Foi proposto que o sistema imunológico é capaz de reconhecer e responder a estímulos que não podem ser detectados pelo sistema neural. Estes estímulos foram denominados de não cognitivos, e incluem elementos como bactérias, vírus, tumores, antígenos, etc., que passariam despercebidos não fosse pelo sistema imunológico. Um vírus não pode ser visto a olho nu, nem cheirado, não possui gosto ou odor e não faz nenhum barulho, mas pode ser percebido pelos sintomas que ele causa. Isto ocorre através do reconhecimento deste estímulo por células do sistema imunológico, que o converte em informações químicas sob a forma de hormônios, neurotransmissores e citocinas (Figura 4.14). Estes sinais são recebidos pelo SNE resultando em mudanças fisiológicas e psicológicas. Aparentemente a função sensorial do SI imita o SNE no sentido de que um determinado estímulo promove uma resposta particular resultando em uma resposta fisiológica.

É interessante notar que o termo cognitivo nas abordagens imunológicas discutidas implica em consciência: a propriedade da intencionalidade aparece em conjunto com a da personalidade. Ambas são dependentes da história e experiência individuais de cada um e levam em consideração uma profunda noção de identidade (próprio). O objetivo é o de enfatizar que princípios cognitivos englobando a idéia de imagem interna podem ser aplicados de forma genérica ao sistema imunológico, que trata:

- *A busca por um contexto*: quando agir;
- *A extração de sinais a partir de ruídos*: como focalizar o reconhecimento; e
- *O problema da resposta*: que decisão tomar.

É possível afirmar que o SI não apenas pode ser visto como um sistema cognitivo, mas também é capaz de complementar e/ou regular as capacidades neurais de reconhecimento e tomada de decisão através da percepção de estímulos que não podem ser detectados fisiologicamente de forma direta. É evidente, como foi brevemente discutido nas seções anteriores, que existe uma reciprocidade (e comunicação direta) de ações entre estes e outros sistemas.

Finalmente, Tada (1997) cunhou o termo “supersistema” para designar sistemas vitais altamente integrados como o sistema imunológico, o sistema nervoso central e a embriogênese. Os diversos elementos de um supersistema se relacionam através de adaptação e co-adaptação mútuas entre seus componentes, criando um sistema dinâmico

auto-regulado através de auto-organização. Trata-se de um sistema auto-contido, porém aberto a estímulos ambientais que podem ser traduzidos em mensagens internas para os processos de auto-regulação e expansão. Um supersistema é caracterizado pela sua auto-regulação, pela geração de seus diversos tipos de componentes através de processos estocásticos seguidos de seleção e adaptação (conseqüências da auto-organização), individualidade e tomada de decisão em resposta a estímulos internos e externos.

4.3.2. Similaridades e Diferenças

Durante a discussão sobre os aspectos cognitivos do sistema imunológico já foi possível perceber algumas similaridades e diferenças entre o SI e o SNC. Parte delas pode ser diretamente confrontada com os respectivos modelos artificiais, tanto na engenharia imunológica quanto nas redes neurais artificiais.

Dentre a vasta gama de similaridades e diferenças entre o SI e o SNC destacam-se (Jerne, 1974; Hoffmann, 1986; Vertosick & Kelly, 1989, 1991; Ishida, 1993, 1996; Dasgupta, 1997, 1999):

- Ambos os sistemas são constituídos por uma grande quantidade e diversidade de células. No homem o SI possui aproximadamente 10^{12} linfócitos e o sistema neural consiste de aproximadamente 10^{10} neurônios. Em ambos os casos células individuais são altamente específicas na execução de suas tarefas;
- A diversidade celular em cada sistema permite que respostas apropriadas sejam geradas para uma grande variedade de estímulos. O sistema imunológico reconhece e responde a formas diversas de moléculas. O SNC reconhece e responde a estímulos recebidos de nossos cinco sentidos: visão, audição, paladar, tato e olfato. Dessa forma, o SI pode ser visto como nosso “sexto sentido”, como proposto por Blalock (1994) e discutido na seção anterior;
- O SNC é geralmente decomposto em suas partes sensorial e motora. Uma divisão análoga em função de reconhecimento e efetora do sistema imunológico pode ser feita, onde os mecanismos efetores são responsáveis pela eliminação do antígeno;
- Nos sistemas neurais, os processos de aprendizagem e aquisição de memória são conseqüências da alteração dos pesos das conexões entre os neurônios, também chamados de sinapses (Figura 4.1), e não da alteração dos próprios neurônios. Além disso, o cérebro permite que as memórias sejam endereçadas por conteúdo, de forma que a morte freqüente de neurônios individuais não afeta drasticamente o desempenho do cérebro como um todo. No caso do sistema imunológico, o conhecimento está armazenado nas próprias células, ou em clones específicos de células e a aprendizagem ocorre através de variações nos padrões de receptores celulares;
- Sob o ponto de vista da teoria da rede imunológica proposta por Jerne (1974a), e apresentada na Seção 2.10, em ambos os sistemas as células estão funcionalmente conectadas entre si sob a forma de uma rede, apresentando funções estimulatórias ou supressivas (inibitórias) entre células vizinhas. No SNC as conexões entre as células são mediadas por axônios, sinapses e dendritos, podendo ser visualizadas em

um microscópio. No caso da teoria da rede imunológica, os componentes do sistema são capazes de reconhecer e serem reconhecidos, oferecendo uma rede de comunicação com características menos óbvias do que as redes neurais;

- As redes neurais utilizam sinais eletroquímicos de comunicação, enquanto o SI emprega ingredientes químicos;
 - As redes neurais apresentam hierarquias, enquanto o sistema imunológico é descentralizado;
 - Ambos utilizam processamento paralelo, onde os estímulos são processados simultaneamente em diferentes partes dos sistemas;
 - A capacidade de apresentar memória distingue estes dois sistemas de todos os outros. A memória de um evento pode permanecer por muitos e muitos anos;
 - Em ambos os casos, o conhecimento armazenado durante a vida não é automaticamente passado para as gerações subsequentes, mesmo que essa herança forneça uma aparente vantagem seletiva para o indivíduo. Este aspecto particular de como a evolução imunológica influencia na evolução da espécie será estudado na Seção 4.5;
 - Ambos os sistemas possuem uma profunda noção de si mesmos, ou seja, do próprio. Uma das tarefas do sistema imunológico é distinguir entre o próprio e o não-próprio, e eliminar tudo aquilo que for categorizado como não-próprio (Seção 2.9). A distinção entre próprio e não-próprio é, de forma similar, uma das funções primitivas do sistema nervoso, e é fundamental para o enigma da consciência;
 - As células dos dois sistemas possuem receptores que permitem a comunicação entre si e/ou com elementos externos (Figura 4.15);
 - Algumas células do sistema imunológico aprendem a reconhecer e responder a formas particulares (antígenos) somente em contextos específicos. Este fenômeno é similar ao reconhecimento dependente do contexto dos sinais recebidos pelo SNC;
 - O SI compartilha com o SNC a propriedade de que um reconhecimento definitivo pode ser obtido aplicando-se um estímulo “fuzzy”. Uma memória pode ser recuperada por um estímulo similar, mas não necessariamente idêntico, ao estímulo original. Em imunologia este fenômeno é chamado de memória reativa cruzada (Ada & Nossal, 1987; Smith *et al.*, 1997), como discutido na Seção 2.8.
 - Ambos os sistemas penetram a maioria dos tecidos do corpo, mas não entram em contato direto devido à *barreira cérebro-sangue (blood-brain barrier)*;
 - O funcionamento ideal de ambos os sistemas é aparentemente insensível a pequenos detalhes em seus elementos ou conexões. Os sistemas devem ser projetados de tal forma que, independente do padrão de conexões ou de interação celular, um repertório (população) de elementos deve existir e ser diverso o suficiente para descrever o universo a ser representado;
 - O corolário do aspecto anterior indica que ambos os sistemas são altamente tolerantes a falhas, devido principalmente a redundância e diversidade de seus componentes.
-

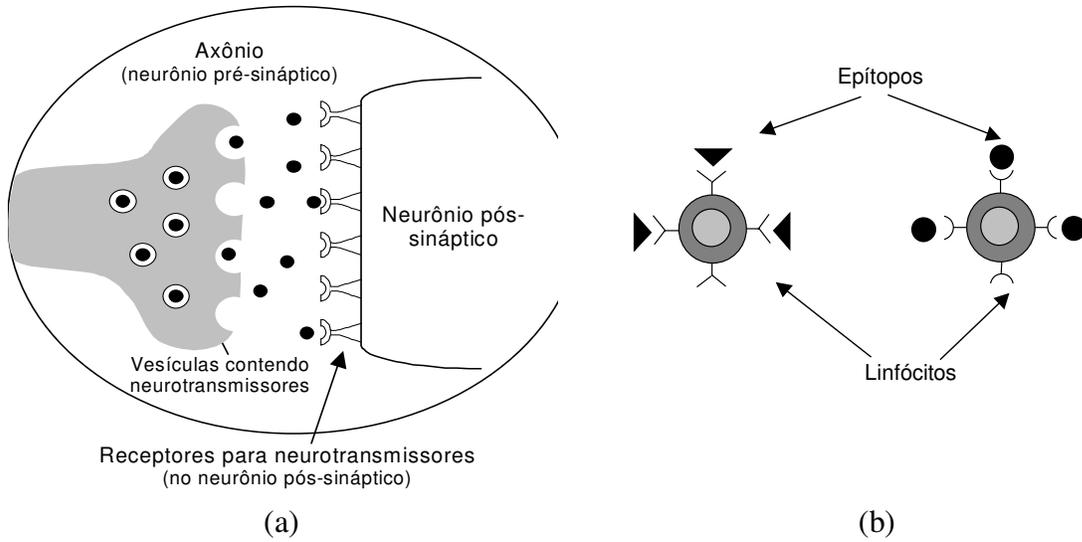


Figura 4.15. Receptores celulares. (a) Os neurônios liberam neurotransmissores quando a atividade elétrica alcança outro neurônio. (b) Os linfócitos possuem receptores celulares capazes de reconhecer padrões de epítomos ou complexos MHC/peptídeo.

4.3.3. Engenharia Imunológica e Redes Neurais Artificiais

Baseados na discussão acima sobre as similaridades e diferenças entre o SI e o SNC, considerando a estrutura formal, apresentada na Seção 3.4, para a engenharia imunológica e os modelos de redes neurais artificiais discutidos no início deste capítulo, é possível fazer um paralelo entre a EI e as RNAs, como descrito na Tabela 4.1.

Tabela 4.1. EI × RNAs.

Característica	EI	RNA
Unidade básica	Célula composta por uma cadeia de atributos, conexões associadas e um limiar de afinidade	Neurônio composto por uma função de ativação, conexões associadas e um limiar de ativação
Interações com outras unidades	As células possuem conexões que a identificam (receptores) e permitem reconhecer outros elementos. Estas conexões podem ser ponderadas indicando o grau de interação com outros elementos	Os pesos sinápticos das conexões podem assumir valores positivos ou negativos indicando uma ação excitatória ou inibitória dos outros neurônios
Atividade	A célula possui uma imagem interna do ambiente que é comparada com a informação recebida	O neurônio processa a informação recebida
Conhecimento	Armazenado nos pesos das conexões e nas cadeias de atributos de cada célula	Armazenado nos pesos das conexões e na forma das funções de ativação
Aprendizagem	Ocorre através da modificação das cadeias de atributos das células e de seus pesos associados	Ocorre através da modificação dos valores e quantidade de pesos sinápticos, juntamente com a alteração das formas das respectivas funções de ativação
Limiar	Determina a ligação (reconhecimento) entre uma célula e o estímulo apresentado	Determina a ativação do neurônio
Robustez	Escalonável, auto-tolerante, flexível e tolerante a falhas	Flexível e tolerante a falhas
Localização	As células podem se deslocar	Os neurônios residem em posições fixas na rede
Comunicação	Ocorre através do contato celular, também representado por um conjunto de conexões	Ocorre através das ligações sinápticas
Estado	Concentração e/ou afinidade entre uma célula e seu estímulo	Nível de ativação dos neurônios de saída
Controle	Uma reação adaptativa determina o tipo de interação entre os componentes do sistema	Um algoritmo de aprendizagem determina o tipo de ajuste dos parâmetros livres da rede

4.4. Computação Evolutiva

Simular processos evolutivos naturais objetivando resolver problemas de otimização tem demonstrado ser um método rápido e robusto. A otimização do comportamento de um sistema por processos evolutivos representa uma abordagem poderosa para a aprendizagem de máquina (*machine-learning*) e fenômenos auto-organizados. A implementação computacional de métodos de simulação da evolução, chamada *computação evolutiva*, possibilita a determinação de ótimos globais para várias classes de problemas cujas soluções eram, até então, desconhecidas ou insatisfatórias. A computação evolutiva pode ser dividida em três grandes grupos (Bäck & Schwefel, 1993; Bäck *et al.*, 2000a):

- Programação evolutiva (PE) introduzida por Fogel *et al.* (1966);
- Estratégias evolutivas (EE) inicialmente propostas por Ingo Rechenberg (1973); e
- Algoritmos genéticos (GAs – *genetic algorithms*), propostos por John Holland (1975).

Todos estes ramos da computação evolutiva seguem o mesmo *algoritmo evolutivo básico*, que será discutido na Seção 4.4.2. Porém, na forma como foram inicialmente propostos, eles possuem algumas diferenças entre si, como resumidas na Tabela 4.2.

Atualmente, o GA constitui o método evolutivo mais amplamente utilizado e, por isso, será discutido em maiores detalhes na Seção 4.4.3. Sua relevância nesta tese se dá sob a forma de aspectos comparativos com algumas das ferramentas propostas. Além disso, serão apresentados comentários gerais sobre a programação genética (PG), um outro ramo da computação evolutiva, devido a sua importância para uma das possíveis extensões desta tese. Por último, será apresentado um modelo de *espécies* em computação evolutiva, também para efeitos comparativos mais adiante com uma das ferramentas a serem propostas.

Tabela 4.2. Principais características dos algoritmos evolutivos (Bäck & Schwefel, 1993).

	EE	PE	GA
Representação	Valores reais	Valores reais	Cadeias binárias
Auto-adaptação	Desvio padrão e covariância	Variâncias (na meta-PE)	Nenhuma
Fitness	Valor da função objetivo	Valor escalonado da função objetivo	Valor escalonado da função objetivo
Mutação	Principal operador	Único operador	Operador secundário
Recombinação	Diversos tipos. Importante para a auto-adaptação	Nenhuma	Principal operador
Seleção	Determinística	Probabilística	Probabilística

4.4.1. Evolução Natural

A *evolução* não é um processo imediato, pois ocorre através do acúmulo de comportamentos cada vez mais apropriados dentro de uma determinada espécie, ou de um determinado ambiente.

C. Darwin (1859) apresentou as seguintes hipóteses para explicar a origem das espécies:

1. Os filhos tendem a ser em maior número que os pais;
2. O número de indivíduos de uma espécie permanece aproximadamente constante;
3. De (1) e (2) conclui-se que haverá luta pela sobrevivência; e
4. Dentro de uma mesma espécie os indivíduos apresentam pequenas diferenças.

Este *princípio da seleção natural* indica que os indivíduos cujas variações se adaptam melhor ao ambiente terão maior probabilidade de sobreviver e reproduzir. Como complemento a estas hipóteses propostas por Darwin, três hipóteses adicionais foram incorporadas, conduzindo ao chamado *neodarwinismo*:

5. Algum processo de variação continuada deve ser responsável pela introdução de novas informações junto ao material genético dos organismos;
6. Não há limite para a sucessão de variações que podem ocorrer; e
7. A seleção natural é o mecanismo para a preservação das novas informações correspondentes a uma maior adaptação.

A *seleção* (competição e exclusão) existe para estatisticamente eliminar (ou reduzir o número de) indivíduos cujo comportamento é pouco apropriado, pois nem todos os indivíduos podem sobreviver na presença de recursos limitados. É importante notar que a evolução otimiza o comportamento, e não o material genético (Atmar, 1994). Sob o ponto de vista da computação, é possível definir evolução da seguinte forma:

Definição 4.5: *Evolução* é um método de busca (em paralelo) de indivíduos com um alto índice de adaptação a um determinado critério, recorrendo-se a um número finito de candidatos, sujeitos a condições de adaptação variáveis.

4.4.1.1. Terminologia Biológica

A terminologia biológica a ser empregada nos algoritmos evolutivos representa uma analogia às entidades biológicas reais, sendo que as entidades computacionais corresponderão invariavelmente a estruturas bem mais simples que seus equivalentes biológicos.

A Figura 4.16 apresenta a hierarquia dos sistemas biológicos. Os algoritmos evolutivos geralmente operam com uma população de indivíduos, de uma única espécie, evoluindo dentro de um ecossistema (ou ambiente) pré-definido.

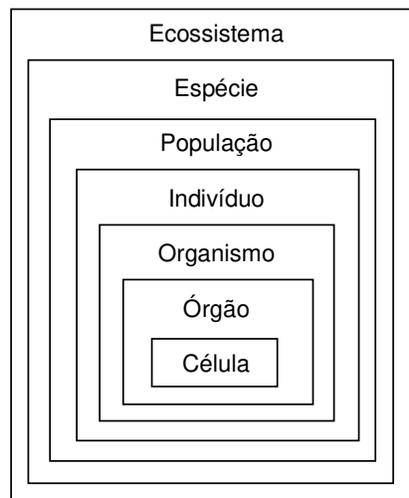


Figura 4.16. Hierarquia dentro dos sistemas biológicos.

Como apresentado na Figura 4.16, todos os organismos vivos são compostos por células, e cada célula contém o mesmo conjunto de um ou mais *cromossomos* – cadeias de DNA – que servem como identificadores do organismo. Um cromossomo pode ser conceitualmente dividido em *genes* (blocos funcionais de DNA), cada qual codificando uma proteína particular. Os tipos diferentes de características são chamados *alelos*. Cada gene está localizado em uma posição particular (*locus*) no cromossomo (Mitchel, 1998).

Muitos organismos possuem múltiplos cromossomos em cada célula. A coleção completa do material genético é chamada de *genoma* do organismo. O termo *genótipo* refere-se ao conjunto particular de genes contido em um genoma. Dois indivíduos que possuem o mesmo genoma têm o mesmo genótipo. O genótipo leva ao *fenótipo* do organismo, ou seja, suas características físicas e mentais, como a cor dos olhos, pele, etc.

Organismos cujos cromossomos aparecem em pares são chamados *diploides*, enquanto organismos com cromossomos que aparecem individualmente são ditos *haploides*. Geralmente, espécies com reprodução *sexuada* (entre dois indivíduos) são diploides, enquanto espécies com reprodução *assexuada* (clonagem de um único indivíduo, como por exemplo a reprodução dos linfócitos) são *haploides*. Durante a reprodução sexuada, ocorre a *recombinação genética* (ou *crossover*): em cada um dos pais, genes são trocados entre cada par de cromossomos para a formação de um *gameta* (cromossomo único), e os gametas dos dois pais formam pares constituindo um conjunto completo de cromossomos diploides. Na reprodução haploide, genes são trocados entre os cromossomos de cada um dos pais. Os *descendentes* (*offsprings*) estão sujeitos à *mutação*, na qual *nucleotídeos* (unidades elementares de DNA) são mudados de pai para filho. A *aptidão* (*adaptabilidade* ou *fitness*) de um organismo é tipicamente considerada equivalente à probabilidade que este indivíduo possui de se reproduzir (*viabilidade*), ou como uma função da quantidade de descendentes que ele pode ter (*fertilidade*).

Tabela 4.3. Terminologia biológica × terminologia computacional (assumindo representação binária).

Biologia	Computação
Cromossomo haplóide	Cadeia de bits que representa um candidato a solução
Gene	Um único bit, ou um bloco de bits
Alelo	Valores que podem ser assumidos para cada gene do cromossomo
Crossover	Troca de material genético entre cromossomos
Mutação	Troca aleatória de um determinado bit do cromossomo
Genótipo	Configuração de bits no cromossomo de um indivíduo
Fenótipo	Decodificação de um ou mais cromossomos

A Tabela 4.3 faz um paralelo entre a terminologia biológica apresentada e a terminologia computacional a ser utilizada nos algoritmos evolutivos. A grande maioria das aplicações de algoritmos evolutivos emprega codificação haplóide baseada em um único cromossomo.

4.4.2. Algoritmo Evolutivo Básico

Os algoritmos evolutivos básicos são geralmente implementados da seguinte forma (Atmar, 1994):

1. Uma população inicial de indivíduos é escolhida aleatoriamente. O tamanho N da população é geralmente grande.
2. A população é reproduzida, com erro, de forma que cada pai gere n descendentes. O mecanismo de introdução de erro durante a reprodução deve ser estocástico e variável.
3. A qualidade do comportamento do sistema é avaliada para todos os indivíduos da população, pais e filhos, baseada em uma medida de adaptabilidade ou fitness de cada indivíduo. Uma de duas condições é geralmente implementada: 1) os N melhores indivíduos são selecionados para reproduzir e compor a próxima geração, 2) N dos melhores indivíduos são selecionados probabilisticamente. O tamanho da população é geralmente restrito.
4. O processo se repete, voltando ao passo 2. A convergência é assumida quando uma solução pré-definida é atingida, ou um número fixo de gerações foi executado.

Definição 4.6: O processo de busca de soluções dentro de um conjunto de candidatos a solução é chamado de *busca em um espaço de buscas*.

Definição 4.7: *Superfície de adaptação* é a representação de um mapeamento do espaço de todos os possíveis genótipos para as respectivas medidas de adaptabilidade (fitness) (Mitchel, 1995).

4.4.2.1. Formalização do Processo

Dois espaços de estados são inerentes aos processos evolutivos: um espaço de estado genotípico, \mathbf{G} , (codificação), e um espaço fenotípico, \mathbf{P} , (comportamental ou de expressão do genótipo). Analogamente, dois alfabetos também são definidos; um alfabeto de entrada do ambiente, \mathbf{I} , e um alfabeto de saída, \mathbf{Z} . A evolução da população dentro de uma única geração está esquematizada na Figura 4.17.

Considere uma população de genótipos, $g_i \in \mathbf{G}$. Existe uma função de mapeamento f_1 tal que cada g_i é transformado em um fenótipo p_i . O mapeamento é inerentemente de muitos para um, pois existem infinitos genótipos que podem resultar em um mesmo fenótipo.

Definição 4.8: *Epigênese* corresponde ao desenvolvimento celular sensível às condições locais. Dito de outra forma, epigênese é a teoria segunda a qual a constituição dos seres vivos se inicia a partir de uma célula sem estrutura e se faz mediante sucessiva formação e adição de novas partes que, previamente, não existiam no ovo fecundado.

O primeiro mapeamento da Figura 4.17, *epigênese*, incorpora regras de expressão genética, crescimento e desenvolvimento, em um contexto local:

$$f_1 : \mathbf{I} \times \mathbf{G} \rightarrow \mathbf{P} \quad (4.20)$$

O segundo mapeamento, *seleção*, descreve os processos de seleção, imigração e emigração de indivíduos dentro de uma população local:

$$f_2 : \mathbf{P} \rightarrow \mathbf{P} \quad (4.21)$$

Os efeitos da seleção ocorrem somente em \mathbf{P} e não afetam diretamente as estruturas codificadas em \mathbf{G} .

O terceiro mapeamento, *representação*, descreve a representação fenotípica dentro de uma população antes da reprodução:

$$f_3 : \mathbf{P} \rightarrow \mathbf{G} \quad (4.22)$$

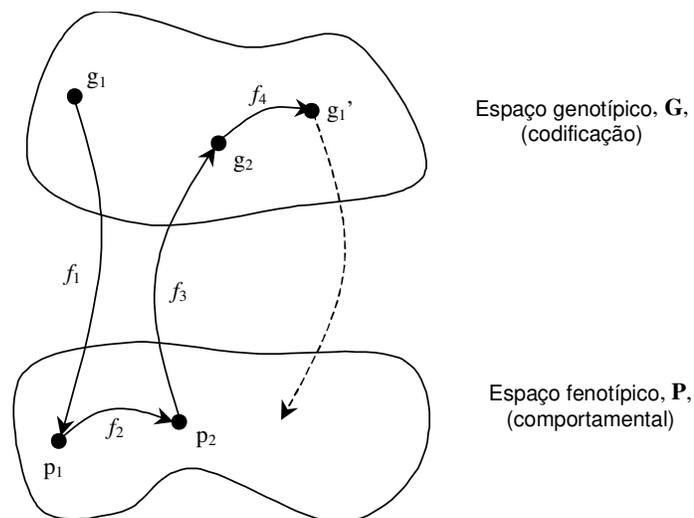


Figura 4.17. Evolução de uma população a cada geração.

O quarto mapeamento, *mutação*, incorpora variação aleatória e direcionada na codificação (incluindo reparo e recombinação):

$$f_4 : \mathbf{G} \rightarrow \mathbf{G} \quad (4.23)$$

Somente através da seleção repetida de comportamentos adequados em \mathbf{P} , geração após geração, uma aprendizagem apropriada pode ser acumulada em \mathbf{G} . A evolução ocorre através da repetição indefinida destes quatro mapeamentos.

4.4.3. Algoritmos Genéticos

Definição 4.9: Os *Algoritmos genéticos* (GAs) são procedimentos computacionais estocásticos cujos métodos de busca modelam os fenômenos biológicos de herança genética e seleção natural (Michalewicz, 1996).

Os algoritmos genéticos simples constituem modelos abstratos da evolução natural e operam com uma população de tamanho fixo e indivíduos representados por “cadeias genéticas” de comprimento fixo. Novas populações evoluem através da seleção probabilística proporcional ao fitness dos indivíduos, produzindo, via crossover e mutação, descendentes semelhantes aos pais.

4.4.3.1. Características Principais e Escopo de Aplicações

Dentre as principais características dos algoritmos genéticos podemos citar (Mitchel, 1995):

- Busca sobre uma população de pontos, e não sobre um único ponto;
- Utilização de funções de custo (funções objetivo, de fitness, ou de adaptabilidade), ao invés de derivadas ou outro tipo de conhecimento auxiliar; e
- Utilização de regras de transição probabilísticas, e não determinísticas.

Dado um problema, como saber se um algoritmo genético irá resolvê-lo de forma eficiente? Não existe nenhuma resposta rigorosa para esta pergunta, mas vários aspectos intuitivos podem ser levantados:

- Se o espaço de buscas é grande, não é perfeitamente suave e nem unimodal, não é conhecido, ou se a função de fitness é ruidosa, o GA se apresenta como uma ferramenta competitiva;
- Se o espaço é suave, ou unimodal, então métodos de *gradiente* ou de *hill climbing* serão muito superiores;
- Se o espaço de buscas é conhecido (como no caso do *caixeiro viajante*), heurísticas podem ser introduzidas em métodos específicos, de forma que eles apresentem bom desempenho.

4.4.3.2. Algoritmo Genético Clássico e Modificado

Nesta seção, introduziremos o algoritmo genético clássico e suas modificações de forma a garantir um melhor desempenho em termos de qualidade da solução (busca pelo ótimo global) e velocidade de convergência.

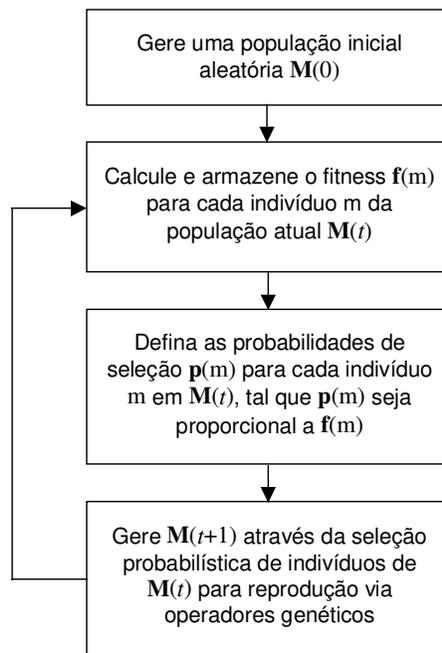


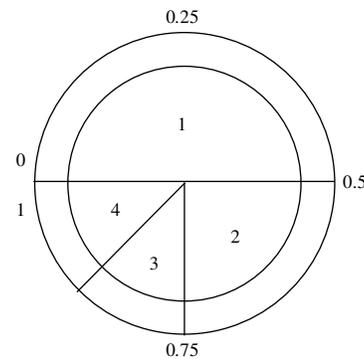
Figura 4.18. Algoritmo genético clássico.

Definida uma representação genética apropriada ao problema, o *algoritmo genético básico* pode ser descrito pelo fluxograma da Figura 4.18 (De Jong, 1994). Este algoritmo possui as seguintes características: 1) codificação binária, 2) reprodução e seleção natural via *Roulette Wheel*, 3) crossover simples, e 4) mutação.

Na seleção via *Roulette Wheel* (RW), a probabilidade de seleção de um cromossomo (indivíduo) é diretamente proporcional a seu valor da função de fitness. A Figura 4.19 ilustra o RW para uma população composta por quatro indivíduos. Rodar uma roleta significa obter um valor de um gerador de números aleatórios com distribuição uniforme no intervalo $[0,1]$. O valor obtido vai definir o cromossomo escolhido de acordo com as marcações na Figura 4.19(b).

N	Cromossomo	Fitness	Graus
1	0001100101010	6.0	180
2	0101001010101	3.0	90
3	1011110100101	1.5	45
4	1010010101001	1.5	45

(a)



(b)

Figura 4.19. Exemplo de Roulette Wheel para uma população com quatro indivíduos.

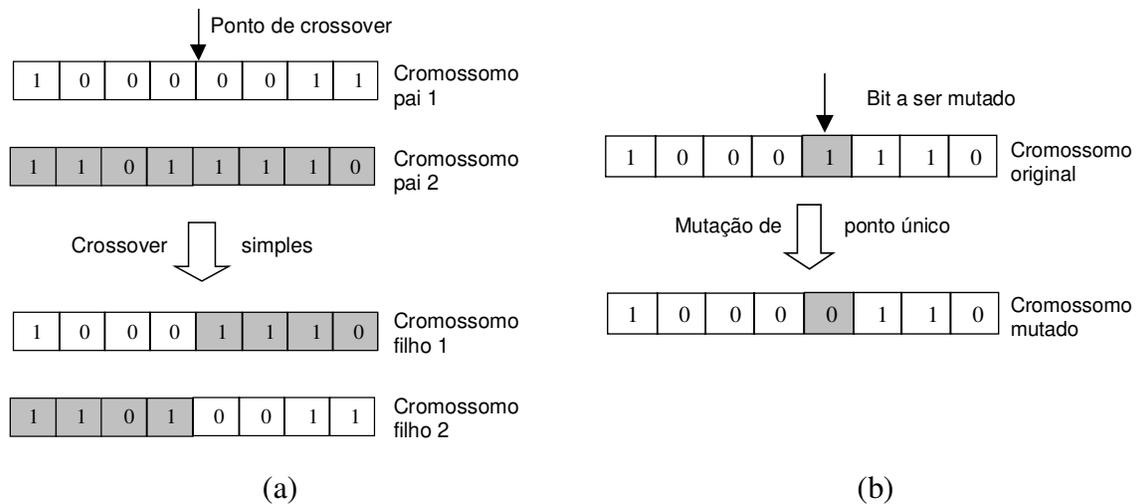


Figura 4.20. Operadores básicos de recombinação. (a) Crossover simples. (b) Mutação de ponto único.

O *crossover simples* é realizado da seguinte forma: dados dois cromossomos (pais), escolhe-se uma posição (locus) aleatória destas cadeias e efetua-se a troca de material genético, conforme indicado na Figura 4.20(a). Cada indivíduo tem uma probabilidade p_c de sofrer crossover. No processo de *mutação*, uma posição (locus) do cromossomo é escolhida aleatoriamente e seu bit trocado de $0 \rightarrow 1$, ou de $1 \rightarrow 0$, conforme indicado na Figura 4.20(b). A probabilidade de mutação p_m define a taxa com que cada locus pode ser mutado.

Dentre os principais problemas com o algoritmo genético clássico, destacam-se:

- A política de reprodução e seleção permite a perda do melhor indivíduo;
- A posição do gene no cromossomo influi na probabilidade de realizar crossover; e
- Dificuldade de codificação quando os parâmetros são números reais.

Para solucionar estes problemas, algumas estratégias foram propostas:

- Utilização de mecanismos alternativos de seleção;
- Crossover uniforme; e
- Codificação em strings (cadeias) de números reais.

Várias modificações podem ser introduzidas no algoritmo genético clássico visando amenizar os problemas citados acima. Estas modificações englobam basicamente outros mecanismos de (1) recombinação (crossover e mutação) e (2) seleção.

4.4.3.2.1. Outros Operadores de Recombinação

1. *Crossover* simples entre indivíduo aleatório e o melhor indivíduo.
O objetivo deste operador é preservar seqüências genéticas (partes da cadeia) do melhor indivíduo na população durante todo o processo evolutivo.
2. *Crossover* uniforme entre indivíduos aleatórios.

Este tipo de crossover tem a característica de não ser influenciado pela posição do gene na probabilidade de realizar crossover. São escolhidos dois indivíduos da população, determina-se a percentagem de genes que serão trocados, procede-se a um sorteio de alelos na quantidade determinada, efetuando o crossover somente dos alelos envolvidos, como ilustrado na Figura 4.21.

3. *Crossover* uniforme entre indivíduos aleatórios e o melhor indivíduo.
4. *Mutação* indutiva (somente para codificação real).
Semelhante à mutação aleatória, só que ao invés de sortear um novo valor para o alelo, sorteia-se um valor a ser somado ao valor atual do alelo.
5. *Mutação* do melhor indivíduo.
Pode ser aleatória ou indutiva.

4.4.3.2.2. Outros Mecanismos de Seleção

1. *Elitista*: os n melhores indivíduos da população são selecionados.
2. *Aleatória*: são selecionados aleatoriamente n indivíduos da população intermediária.
 - 2.1. *Salvacionista*: seleciona-se o melhor indivíduo e os outros aleatoriamente.
 - 2.2. *Não-salvacionista*: todos os indivíduos são selecionados aleatoriamente.
3. *Por diversidade*: os indivíduos mais diversos são selecionados.
4. *Bi-classista*: são selecionados os $p_b\%$ melhores e os $p_w\%$ piores indivíduos.
5. *Por torneio binário*: dois indivíduos são escolhidos aleatoriamente. Um número aleatório $r \in [0,1]$ é gerado. Caso $r < k$ (k é um parâmetro, como por exemplo 0.75), o indivíduo com maior fitness é selecionado, senão o outro indivíduo é escolhido.

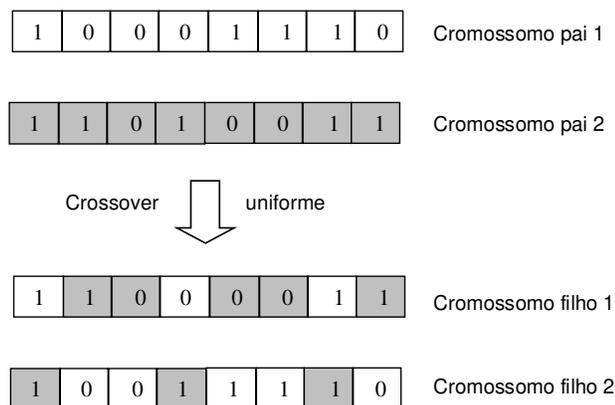


Figura 4.21. Crossover uniforme.

4.4.4. Programação Genética

A principal questão que levou ao surgimento da programação genética foi: “Como os computadores podem aprender a resolver problemas sem serem explicitamente programados para tal?”.

A maior barreira para se atingir este objetivo está na característica determinística da grande maioria dos ramos de pesquisa em inteligência artificial. Nestes casos, as propriedades principais que um programa deve apresentar para ser um candidato à solução de problemas de interesse prático são: correção, consistência, motivação lógica, precisão, ordenação, parcimônia, e definibilidade.

A *programação genética* (PG) pode gerar programas que falham em todos estes quesitos. Ela manipula soluções corretas e incorretas, encoraja inconsistências e abordagens contraditórias, pode não apresentar um seqüenciamento lógico, é predominantemente probabilística, produz soluções não-parcimoniosas e não apresenta um critério de terminação claramente definido. Pode ser definida como a seguir:

Definição 4.10: Programação Genética (PG) constitui um algoritmo evolutivo no qual as estruturas de dados que sofrem adaptação são programas executáveis de computador. A avaliação do fitness na programação genética envolve a execução dos programas evoluídos. Assim, a PG envolve uma busca baseada na evolução junto aos candidatos pertencentes ao espaço de possíveis programas computacionais (Bäck *et al.*, 2000a – Cap. 11).

A PG pode se constituir em uma opção especialmente interessante para o caso de programação de computadores com processamento paralelo. No contexto originalmente adotado para a PG, um programa computacional é basicamente uma seqüência de aplicações de funções a argumentos: *paradigma funcional*.

A implementação da PG é imediata quando associada a linguagens de programação que permitem a manipulação de um programa computacional na forma de estrutura de dados, inclusive por possibilitar que novos dados do mesmo tipo e recém-criados sejam imediatamente executados sob a forma de programas computacionais. Entretanto, qualquer linguagem computacional capaz de implementar (mesmo que indiretamente) a mesma estrutura de dados pode ser potencialmente empregada. Originalmente, sua linguagem é a LISP (Koza, 1992).

Para um sistema de programação genética, um problema é definido pela *representação* e pela função de *fitness*, tornando necessário um conhecimento profundo do problema em questão.

A *representação* de um problema consiste de *funções* e *terminais*, elementos básicos de um programa estruturado no paradigma funcional. Os programas computacionais na linguagem definida são os indivíduos, passíveis de representação em estruturas de dados do tipo árvore. Estes programas precisam ser executados para se obter o correspondente candidato à solução (Kinnear Jr., 1994). A Figura 4.22 ilustra uma estrutura em árvore representativa do programa computacional responsável pela avaliação da função $f = (x - y) - (a + b)$, sendo que x , y , a e b são terminais, e $+$ e $-$ são funções.

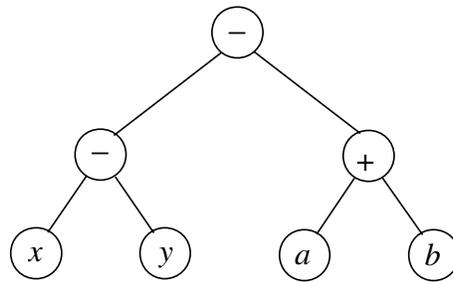


Figura 4.22. Árvore representativa de um programa em PG para avaliar a expressão $f = (x - y) - (a + b)$.

4.4.5. Nichos e Espécies

Como discutido anteriormente (Definição 4.7), os algoritmos genéticos baseiam-se na teoria da seleção natural e operam de forma análoga à evolução natural. Entretanto, enquanto os processos evolutivos naturais mantêm uma variedade de *espécies*, cada uma ocupando um *nicho ecológico* diferente, os algoritmos genéticos tradicionais (ou clássicos) convergem rapidamente para um único ponto. Isso significa que todos os indivíduos da população tornam-se praticamente iguais. Mesmo quando existem múltiplas soluções para um determinado problema, o GA clássico localiza apenas uma delas (Mahfoud, 1995). No Capítulo 6, onde apresentaremos as aplicações das ferramentas propostas de engenharia imunológica, serão feitas comparações de desempenho com os algoritmos genéticos clássicos, um método de niching (mais especificamente o método de fitness sharing apresentado a seguir), e seus aspectos de convergência para um ou mais ótimos da função serão ilustrados e discutidos.

Intuitivamente, podemos visualizar uma espécie como sendo uma classe de organismos que apresentam características comuns, e um nicho como o local ocupado pelo elementos de uma mesma espécie.

Definição 4.11: Um *método de niching* é aquele que faz com que os algoritmos genéticos consigam manter uma população diversa de indivíduos, tornando-os capazes de localizar múltiplas soluções ótimas (subpopulações) dentro de uma única população (Goldberg, 1989).

O estudo de nichos e espécies em computação evolutiva, mais especificamente em algoritmos genéticos, está diretamente relacionado ao problema da *diversidade populacional*. Uma das principais motivações para o desenvolvimento dos métodos de niching foi a busca pela diversidade, que possui duas funções dentro dos GAs: retardar a convergência objetivando aumentar a exploração da superfície de adaptação, e permitir a determinação de múltiplas soluções.

Os mecanismos básicos de seleção dos algoritmos genéticos reproduzem os indivíduos de maior fitness e descartam aqueles com valores baixos de fitness, resultando em uma redução de diversidade da população. Três fatores fazem com que o GA clássico elimine indivíduos caracterizados por valores reduzidos de fitness e privilegie aqueles com valores

de fitness mais elevados: *pressão seletiva*, *ruído de seleção* e *quebra pelo operador*. A pressão seletiva é o resultado do valor esperado do processo de seleção: indivíduos com fitness baixos devem desaparecer de uma população com tamanho finito. O ruído de seleção é resultado da variância do processo de seleção: em uma população finita, escolhas aleatórias entre indivíduos com fitness idênticos adiciona ruído ao fitness esperado de cada indivíduo, forçando o surgimento de boas soluções. A quebra pelo operador é resultado da aplicação dos operadores de recombinação, como crossover e mutação, que são potencialmente capazes de eliminar bons candidatos à solução. As técnicas para diversificar populações geralmente reduzem um destes três fatores, ou uma combinação deles (Mahfoud, 1995).

Um método de niching deve ser capaz de formar e manter múltiplas e diversas soluções finais com valores iguais ou diferentes de fitness. Além disso, estas soluções devem ser mantidas por um longo período de tempo quando comparado ao tamanho da população. Mahfoud (1995) classificou os métodos de niching em duas categorias de comportamento: 1) espacial/temporal e de 2) único ambiente/múltiplos ambientes. Dentro da primeira categoria, um método espacial de niching forma e mantém subpopulações dentro de uma única população, enquanto um método temporal desenvolve nichos seqüencialmente ao longo do tempo. Na segunda categoria, um ambiente pode ser visto como uma função de fitness e, portanto, o algoritmo pode ser classificado como capaz de resolver problemas multi-objetivo no caso de múltiplos ambientes. A Tabela 4.4 ilustra estas categorias e como alguns algoritmos ou sistemas podem ser classificados dentro delas. Os *GAs ecológicos* induzem a formação de nichos através da utilização ou criação de múltiplos ambientes, onde cada ambiente pode ser entendido como sendo uma função de fitness (Mahfoud, 1995).

Por ser o método de niching mais empregado na literatura, descreveremos a seguir o algoritmo de *fitness sharing* para que possamos posteriormente discutir sua relação com alguns modelos imunológicos existentes, e compará-lo empiricamente com um dos algoritmos propostos nesta tese.

Tabela 4.4. Os métodos de niching podem ser classificados em duas categorias de comportamento: 1) espacial/temporal e de 2) único/múltiplos ambientes.

	Único ambiente	Múltiplos ambientes
Temporal	Localização seqüencial	Superespecificação GAs ecológicos
Espacial	Crowding Competição restrita Fitness sharing	GAs ecológicos Sistemas Imunológicos

4.4.5.1. Fitness Sharing

Goldberg & Richardson (1987) propuseram um esquema prático que utiliza uma metáfora de *compartilhamento de fitness* (*fitness sharing*) para solucionar o problema de otimização de funções multimodais. Neste esquema, uma função de compartilhamento é definida para determinar a vizinhança e grau de compartilhamento para cada indivíduo da população. Para um dado indivíduo, o grau de compartilhamento é determinado somando-se o valor da função de compartilhamento de todos os outros indivíduos da população. Indivíduos próximos a outros requerem um alto grau de compartilhamento, enquanto indivíduos distantes entre si requerem valores pequenos de compartilhamento. O compartilhamento reduz o fitness de um indivíduo proporcionalmente à quantidade de indivíduos similares a ele dentro da população. Especificamente, o fitness compartilhado $f_s(x_i)$ de um indivíduo é igual a seu fitness $f(x_i)$ dividido pelo *contador de nichos*, que por sua vez é a soma dos valores das funções de compartilhamento entre este indivíduo e o restante da população:

$$f_s(x_i) = \frac{f(x_i)}{\sum_{j=1}^N sh(d(x_i, x_j))}, \quad (4.24)$$

onde N é a quantidade de indivíduos da população e $sh(\cdot)$ é uma função de compartilhamento proporcional à distância $d(\cdot, \cdot)$ entre dois elementos da população. A função de compartilhamento considera um *limiar de compartilhamento* (similaridade) σ_{share} tal que, se a distância entre dois indivíduos da população é maior ou igual σ_{share} , eles não afetam seus respectivos fitness. A função de compartilhamento mais empregada é (Bäck *et al.*, 2000b):

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}} \right)^\gamma, & \text{se } d < \sigma_{share} \\ 0, & \text{outros casos} \end{cases}, \quad (4.25)$$

onde γ é uma constante (geralmente $\gamma = 1$) capaz de regular a forma da função de compartilhamento. Medidas de distância genotípica ou fenotípica podem ser empregadas, de acordo com o problema a ser tratado.

4.5. Microevolução: Uma Evolução Dentro do Indivíduo

Na Seção 4.4, que descreve a computação evolutiva, discutimos como a evolução natural serviu de fonte inspiradora para o desenvolvimento de algoritmos computacionais capazes de simular processos evolutivos.

Os processos de *aprendizagem* ou *adaptação* podem se manifestar de três formas distintas na seleção natural:

- *Adaptação filogenética*: acúmulo de conhecimentos com reflexos nas células germinais de uma espécie;
- *Adaptação sociogenética*: comportamentos adaptativos são acumulados durante a vida dentro de um grupo (sociedade); e

- *Adaptação ontogenética*: comportamentos apropriados são aprendidos através de tentativa e erro durante a vida de um indivíduo.

Organismos individuais sofrem adaptação ontogenética de várias formas: os músculos ficam mais fortes quanto mais são utilizados, o comportamento varia ao longo do tempo, etc. Estas mudanças são rapidamente observáveis, porém um dos tipos de adaptação ontogenética mais estudado envolve o sistema imunológico dos mamíferos (Cziko, 1995).

Para que possamos compreender a evolução dentro do sistema imunológico, faremos um breve resumo da teoria da seleção clonal e maturação de afinidade dos linfócitos B (Seção 2.8). O princípio da seleção clonal pressupõe que uma grande quantidade de linfócitos B contendo receptores antigênicos, chamados de anticorpos, estão circulando constantemente pelo nosso organismo. A grande diversidade deste repertório é resultado da recombinação aleatória de fragmentos genéticos contidos em diversas bibliotecas, somada à inserção aleatória de sequências genéticas conforme a célula se desenvolve. Esta grande diversidade do repertório linfocitário virtualmente garante que pelo menos uma célula irá produzir um anticorpo capaz de reconhecer, e portanto se ligar a qualquer antígeno que invada o organismo (completude do repertório). A ligação Ag-Ab estimula a produção de clones desta célula (divisão celular), de modo que sucessivas gerações resultam no crescimento exponencial da quantidade de anticorpos circulantes do tipo selecionado. Alguns destes anticorpos permanecem circulando mesmo após o final da resposta, constituindo a memória imunológica. Outras células se diferenciam em plasmócitos produzindo anticorpos em grandes escalas. Finalmente, durante a reprodução, alguns clones sofrem o processo de maturação de afinidade, onde mutações somáticas são inseridas com altas taxas e, combinadas a uma forte pressão seletiva, melhoram a capacidade destes anticorpos em reconhecerem os respectivos antígenos (afinidade Ag-Ab).

O funcionamento da resposta imune adaptativa baseado no princípio da seleção clonal, brevemente discutido acima, revela que o sistema imunológico pode ser visto como um microcosmo da teoria da evolução proposta por Charles Darwin (Seção 4.4.1) com as três características fundamentais de *diversidade da população* (repertório), *variação genética* e *seleção natural* (Perelson *et al.*, 1978; Cziko, 1995; Adams, 1996). A diversidade do repertório é evidente, pois o sistema imunológico é capaz de produzir uma quantidade bem maior de anticorpos (repertório potencial) do que é utilizado instantaneamente (repertório ativo). Aparentemente, grande parte dos anticorpos produzidos não executam nenhuma tarefa durante a resposta imunológica. A variação genética ocorre durante a produção de novos anticorpos através da recombinação de fragmentos genéticos das diversas bibliotecas do nosso genoma. Por último, a seleção opera de forma que anticorpos capazes de reconhecerem e se ligarem aos antígenos terão maiores probabilidades de se reproduzir e serem mantidos como células de memória.

A similaridade entre a evolução biológica adaptativa e a produção de anticorpos ainda é mais forte quando consideramos que os dois processos centrais na produção de anticorpos, recombinação e mutação genética, são os mesmos responsáveis pela evolução biológica das espécies sexuadas. A recombinação dos genes que compõem as moléculas de imunoglobulina é responsável pela grande diversidade da população, enquanto a mutação serve como um mecanismo de ajuste da afinidade do anticorpo em relação ao antígeno. Nas

espécies com reprodução sexuada, estes mesmos processos são responsáveis por fornecer as variações sob as quais a seleção natural irá atuar para tornar o organismo melhor adaptado ao seu ambiente (Holland, 1975). Assim, uma variação cega cumulativa juntamente com a seleção natural, que durante milhões de anos resultaram na emergência dos mamíferos, permanece crucial na batalha diária pela sobrevivência desta espécie. Deve ser notado, entretanto, que a recombinação genética das moléculas de imunoglobulina envolvidas na produção de anticorpos difere consideravelmente da recombinação do material genético proveniente dos pais na reprodução sexuada. No primeiro caso, nucleotídeos podem freqüentemente ser inseridos e deletados aleatoriamente dos segmentos gênicos recombinados, enquanto a reprodução sexuada envolve quase sempre apenas a troca (crossover) de material genético dos pais, gerando filhos constituídos por uma mistura genética dos cromossomos dos pais.

Enquanto a evolução biológica adaptativa ocorreu através da seleção natural *entre* organismos (ou entre organismos e genes (Dawkins, 1989)), a pesquisa em imunologia revelou a primeira evidência clara de que uma mudança adaptativa ontogenética pode ser atingida através de variação e seleção cumulativas *dentro* dos organismos. A seleção natural pode ser vista como atuando no sistema imunológico em dois níveis. Primeiro, permitindo a multiplicação e variação das características dos linfócitos que apresentam maiores afinidades ao antígeno e, segundo permitindo a multiplicação dos indivíduos cujo material genético apresenta maior potencialidade de fornecer máxima defesa contra doenças infecciosas associadas ao menor risco de apresentação de doenças autoimunes.

Uma última similaridade entre o funcionamento do sistema imunológico e a evolução biológica pode ser vista como o nosso próprio conhecimento sobre como cada um destes mecanismos operam. A nossa compreensão da adaptabilidade de um organismo a seu ambiente evoluiu de uma explicação instrucionista (Seção 2.1), do tipo proposto por Lamarck, para um modelo puramente seletivo, do tipo neodarwinista discutido na Seção 4.4.1. Estágios semelhantes ocorreram no desenvolvimento da teoria da evolução atualmente aceita.

4.6. Outras Abordagens de Sistemas Inteligentes

Esta seção visa discutir brevemente a relação entre outras abordagens de sistemas inteligentes, em particular a computação molecular e os sistemas nebulosos, o sistema imunológico biológico e a engenharia imunológica.

4.6.1. Computação Molecular

O trabalho pioneiro apresentado por Adleman (1994) introduziu um experimento realizado com *moléculas de DNA* para resolver o problema do caminho Hamiltoniano (HPP – *Hamiltonian path problem*). Este problema pertence à classe NP-completo, é de natureza combinatorial e se assemelha bastante ao problema do caixeiro viajante (TSP – *travelling salesman problem*) a ser apresentado, discutido e resolvido por uma das ferramentas de engenharia imunológica que será proposta no Capítulo 5. O experimento de Adleman

(1994) levou às seguintes conclusões sobre a computação molecular: 1) é possível desenvolver *protocolos* capazes de coordenar moléculas individuais de forma que elas realizem tarefas específicas, e 2) existe uma grande potencialidade das moléculas de DNA na solução de problemas complexos de engenharia. De forma resumida, este experimento consistiu em arranjar cuidadosamente um conjunto de moléculas de DNA de forma que o comportamento químico por elas seguidas (*protocolo molecular*) naturalmente pudesse realizar o procedimento computacional desejado. As operações principais desta química são mecanismos de ligação que permitem que os nucleotídeos básicos dos ácidos nucleicos das moléculas de DNA formem estruturas maiores através de processos de *ligação* (*ligation* ou *matching*) e *hibridização* (*hybridization*). Estes processos de ligação se dão entre pares Watson-Crick complementares, A-T (Adenina-Timina) ou C-G (Citosina-Guanina), portanto, uma única cadeia irá se ligar a outra através da operação de complementaridade de Watson-Crick.

De forma genérica, os procedimentos mais importantes para a computação molecular podem ser caracterizados como a seguir (Garzon & Deaton, 1999):

1. *Eletroforese de gel*: funciona como um microscópio, permitindo a visualização das moléculas a olho nu;
2. *Enzimas de restrição*: funcionam como “tesouras” em operações de corte e ligação; e
3. *Reação em cadeia de polimerase* (PCR – *polymerase chain reaction*): opera como uma copiadora para a computação molecular, permitindo a duplicação de moléculas.

Estes procedimentos permitem executar alguns dos três passos principais da metodologia de computação molecular: a 1) *codificação* que mapeia o problema em cadeias de DNA, a 2) *hibridização/ligação* que realiza o processamento principal e a 3) *extração* que permite visualizar os resultados a olho nu.

Baseado na discussão acima sobre computação molecular e no Capítulo 3 que apresenta os conceitos de engenharia imunológica, é possível traçar um paralelo entre elas, de acordo com a Tabela 4.5.

O único trabalho conhecido na literatura até o presente momento, que desenvolve um sistema imunológico artificial baseado em moléculas de DNA foi apresentado por Deaton *et al.* (1997). Neste trabalho, os autores propuseram uma versão molecular do algoritmo de seleção negativa (ASN) introduzido por Forrest *et al.* (1994) e discutido na Seção 3.4.3.2. No artigo apresentado por Nagano & Yonezawa (1999), foram empregadas seqüências de DNA para representar os elementos próprios e não-próprios de um sistema de auto-reconhecimento, entretanto, a dinâmica deste sistema foi coordenada por um algoritmo genético e não por protocolos de computação molecular.

Tabela 4.5. Paralelo entre a engenharia imunológica e a computação molecular.

	Engenharia Imunológica	Computação Molecular
Codificação	Cadeias de atributos, em S , que representam as células e moléculas do sistema imunológico	Cadeias de DNA
Recombinação	Mutação e edição de receptores	Enzimas de restrição
Processamento	Algoritmos (ex. ASN, CLONALG) resultantes da síntese de princípios imunológicos	Protocolos específicos de hibridização/ligação
Ligação	Complementaridade em S	Pares complementares Watson-Crick, A-T e C-G
Reprodução	Expansão clonal (mitose celular)	PCR
Decodificação	Interpretação das cadeias de atributos resultantes	Extração (via eletroforese de gel)

4.6.2. Sistemas Nebulosos

O conceito de *conjunto nebuloso* foi introduzido, em 1965, por Lotfi A. Zadeh. A ele é atribuído o reconhecimento como grande colaborador da Teoria de Controle Moderno. Em meados dos anos 60, Zadeh observou que os recursos tecnológicos disponíveis eram incapazes de automatizar as atividades relacionadas a problemas de natureza industrial, biológica ou química, que compreendessem situações ambíguas, não passíveis de processamento através da lógica computacional fundamentada na álgebra (lógica) Booleana. Procurando solucionar estes problemas, Zadeh (1965) publicou um artigo resumindo os conceitos sobre *conjuntos nebulosos*, revolucionando o assunto com a criação dos *sistemas de inferência nebulosa*.

A inferência nebulosa envolve a representação e operação com *variáveis lingüísticas* e *conjuntos nebulosos*. Conjuntos nebulosos surgiram como uma nova forma de *representação de incertezas*, e têm sido bastante utilizados em problemas de mundo real e, principalmente, em conjunto com outras metodologias como redes neurais artificiais, computação evolutiva e, mais recentemente, sistemas imunológicos artificiais.

A *lógica* refere-se aos princípios formais que sustentam as leis do pensamento e espelham os fundamentos de várias áreas da ciência e engenharia. Sistemas com lógica de dois valores (lógica binária ou bi-valorada) tratam sentenças cujos valores são *verdadeiro* ou *falso*. Em lógica *multi-valorada*, uma sentença pode ser *verdadeira*, *falsa*, ou pode assumir *valores verdade intermediários*. A *lógica nebulosa* vai muito além da lógica multi-valorada, e admite valores verdade que são conjuntos nebulosos do intervalo unitário. Os valores verdade podem ser vistos como caracterizações lingüísticas de valores verdade

numéricos. Assim, a lógica nebulosa trata os princípios do raciocínio aproximado (Pedrycz & Gomide, 1998).

Krishnakumar *et al.* (1995) propuseram um *sistema computacional imunizado* empregando metáforas imunológicas e técnicas computacionais com o objetivo de reproduzir a robustez e adaptabilidade do sistema imunológico biológico. A estrutura do sistema híbrido proposto, simulando blocos de processamento do sistema imunológico, era composta por redes neurais artificiais, sistemas nebulosos e algoritmos evolutivos, de acordo com o problema a ser tratado. O desempenho desta abordagem foi avaliado em um problema de controle autônomo de aviões.

Nos trabalhos apresentados por Lee *et al.* (1999) e Jun *et al.* (1999), foram utilizados diferentes tipos de anticorpos, cada qual com uma tarefa específica. Neste caso, o nível de estimulação de um dado anticorpo era função do seu percentual de sucesso na execução de uma tarefa, de acordo com uma função semelhante à apresentada na Figura 4.23. Esta função é típica em sistemas nebulosos, onde o universo de discurso de uma variável (no caso o percentual de sucesso) é particionado em vários intervalos, permitindo uma representação aproximada do fenômeno tratado.

Além destes exemplos de aplicações, na Seção 3.4.2.1 foi definido um valor de ligação vl entre duas moléculas proporcionalmente à afinidade entre elas. Este vl é dependente de uma função de ativação capaz de permitir valores graduais de ligação. No sistema imunológico biológico, o reconhecimento antigênico acontece de forma aproximada, ou seja, uma resposta imune adaptativa pode ser promovida mesmo quando não ocorre uma ligação perfeita entre o antígeno e o anticorpo. Juntamente com a reatividade cruzada, estas características ressaltam a existência de processamentos nebulosos (aproximados) dentro do sistema imunológico, sugerindo que a lógica nebulosa pode ser bastante apropriada para modelar diversos mecanismos e aspectos do sistema imunológico biológico.

Finalmente, na Seção 5.5 apresentaremos um modelo de rede imunológica artificial para resolver problemas de clusterização e compressão de dados, e discutiremos uma forma de clusterização nebulosa (gradual) que pode ser desempenhada por este modelo.

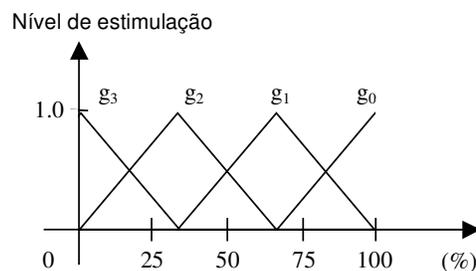


Figura 4.23. Função de determinação do nível de estimulação de um dado anticorpo.

CAPÍTULO 5

FERRAMENTAS DE ENGENHARIA IMUNOLÓGICA

Este capítulo concretiza a engenharia imunológica através da proposição de quatro ferramentas computacionais desenvolvidas baseadas nos princípios imunológicos biológicos, introduzidos no Capítulo 2, e no paradigma de engenharia imunológica, proposto no Capítulo 3. Dentre os princípios fundamentais do sistema imunológico biológico utilizados, destacam-se a questão da diversidade populacional, o princípio da seleção clonal e a teoria da rede imunológica. A formalização teórica e o desenvolvimento dos algoritmos SAND, CLONALG, ABNET e aiNet são apresentados. Algumas das ferramentas propostas constituem estratégias híbridas que utilizam, além de conceitos imunológicos, noções de outras técnicas computacionais como as redes neurais artificiais competitivas e os algoritmos evolutivos revisados e discutidos no Capítulo 4.

“O acaso favorece a mente preparada” – Autor desconhecido

5.1. Introdução

Neste capítulo são propostas quatro ferramentas de engenharia imunológica:

1. SAND: algoritmo baseado na técnica clássica de *Simulated Annealing* (Kirkpatrick *et al.*, 1987) cujo objetivo principal é aumentar a diversidade de um repertório de anticorpos representado sob a forma de cadeias binárias (espaço de formas de Hamming) e vetores reais (espaço de formas Euclidiano). É utilizado o processo de mutação somática proporcional à afinidade;
2. CLONALG: implementação computacional do algoritmo de seleção clonal, cujas aplicações incluem aprendizagem de máquina e reconhecimento de padrões, otimização multimodal e combinatorial. Engloba o princípio de seleção clonal e maturação de afinidade;
3. ABNET: rede neural artificial Booleana com processos de crescimento (inserção de unidades), poda (remoção de unidades) e atualização de pesos, baseados nos princípios imunológicos da seleção clonal e maturação de afinidade;
4. aiNet: modelo de rede imunológica artificial com aplicações a problemas de análise de dados, como compressão de informações e clusterização. Utiliza o algoritmo CLONALG como parte do seu processamento e a teoria da rede imunológica para controlar a dimensão da rede.

A apresentação de todos os algoritmos propostos inclui a descrição e motivação para o seu desenvolvimento, o escopo de aplicações, a listagem de pseudocódigos, um estudo sobre o custo computacional envolvido no processamento de cada iteração e uma discussão caracterizando e relacionando as ferramentas propostas com outras abordagens da literatura. O estudo e a apresentação teórica dos algoritmos fazem parte deste capítulo, enquanto as aplicações e comparações de desempenho com outras abordagens serão apresentadas no Capítulo 6.

5.1.1. Características Gerais

Todos os quatro algoritmos propostos nesta tese e que serão descritos neste capítulo foram desenvolvidos baseados em algum princípio imunológico e serão analisados utilizando a estrutura formal para a engenharia imunológica proposta no Capítulo 3.

Como discutido na proposta de engenharia imunológica (Cap. 3), devido à característica monoclonal dos anticorpos, não é feita distinção entre uma célula B e seu receptor e, por convenção, todos os N indivíduos da população são denominados genericamente de anticorpos e representados por \mathbf{Ab}_i , $i = 1, \dots, N$. Considerando o problema genérico de reconhecimento de padrões, existe uma população \mathbf{Ag} de antígenos a ser reconhecida por um repertório \mathbf{Ab} de anticorpos. Cada anticorpo e cada antígeno é representado por um vetor (cadeia de atributos) $\mathbf{Ab}_i = \langle Ab_{i,1}, \dots, Ab_{i,l}, Ab_{i,l+1}, \dots, Ab_{i,L} \rangle$ e $\mathbf{Ag}_j = \langle Ag_{j,1}, Ag_{j,2}, \dots, Ag_{j,L} \rangle$, $j = 1, \dots, M$, respectivamente, considerados de mesmo comprimento L . Não é feita distinção entre o paratopo e o idiotopo dos anticorpos. Portanto, a cadeia de atributos que representa esta molécula pode ser resumida na forma $\mathbf{Ab}_i = \langle Ab_{i,1}, Ab_{i,2}, \dots, Ab_{i,L} \rangle$ e a representação pictórica dos anticorpos componentes da EI, ilustrada na Figura 3.9(a), pode ser sintetizada como na Figura 5.1(a), que constitui um caso particular da Figura 3.9(a). Esta representação pictórica é mais interessante para a visualização da aiNet, a ser apresentada na Seção 5.5.

Existem diversas características em comum entre as quatro ferramentas de engenharia imunológica a serem propostas, destacando-se:

- São baseadas em um repertório, ou população, de anticorpos;
- Foram genericamente desenvolvidas para resolver problemas de aprendizagem de máquina e reconhecimento de padrões, mas muitas serão aplicadas a outros contextos, como otimização e classificação;
- Não fazem diferença entre o paratopo e o idiotopo dos anticorpos, ou seja, a mesma cadeia de atributos que faz o reconhecimento antigênico é também responsável pelo reconhecimento dos elementos próprios (este aspecto é relevante apenas no modelo de rede imunológica artificial proposto, aiNet);
- Todas as células possuem o mesmo limiar de afinidade ε (quando ele é considerado); e
- Células ou anticorpos individuais podem sofrer variações genéticas, mais especificamente mutação (maturação de afinidade), permitindo que eles se adaptem melhor ao ambiente.

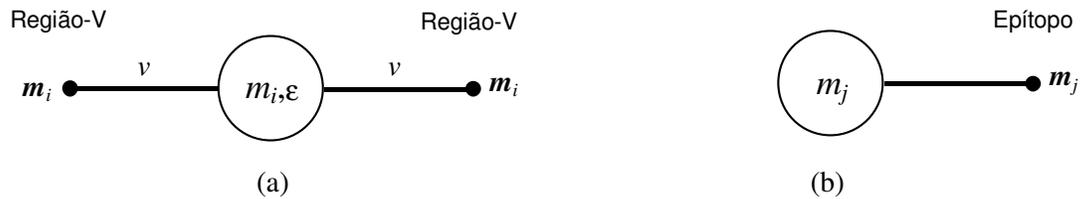


Figura 5.1. Representações pictóricas das moléculas utilizadas nos algoritmos propostos. (a) Anticorpo. A região-V é a região de ligação ao antígeno. (b) Antígeno.

Outro aspecto importante de todos os modelos propostos, é a utilização de uma taxa de mutação proporcional à afinidade, ou fitness, de cada anticorpo. O que o processo de maturação de afinidade, descrito na Seção 2.8.2, estabelece, é que quanto maior a afinidade antigênica de um receptor celular, menor a taxa de mutação a que ele estará sujeito durante o processo de reprodução e vice-versa. Esta característica pode ser simulada como proposto na Seção 3.4.3.4, Figura 3.16.

5.1.2. Complexidade Computacional

Um programa computacional, embora derivado de um algoritmo correto, pode ser inútil para algumas aplicações (dados de entrada), devido ao tempo de processamento ou à quantidade de memória necessária para armazenar os dados e variáveis envolvidas. A *análise de um algoritmo* refere-se ao processo de derivação de estimativas de tempo e espaço (memória) requeridos para executá-lo. A *complexidade de um algoritmo* refere-se à quantidade de tempo e espaço necessários para executá-lo (Johnsonbaugh, 1997). A determinação de parâmetros de desempenho de um programa computacional não é uma tarefa fácil e depende de uma série de fatores, tais como o computador sendo utilizado, a forma de representação dos dados, e como e com qual linguagem de programação o código foi implementado. Embora estimativas precisas do tempo de processamento de um algoritmo devam considerar estes aspectos, informações úteis podem ser obtidas pela análise da complexidade computacional do algoritmo.

O tempo necessário para executar um algoritmo é função dos dados de entrada. Ao invés de tratar os dados de entrada diretamente, utiliza-se parâmetros que caracterizam sua dimensão como, por exemplo, a quantidade N de dados. Neste capítulo, todos os algoritmos a serem propostos acompanharão uma análise assintótica de complexidade envolvendo o tempo de processamento e a quantidade de memória necessários para executá-los.

Sendo f e g funções reais com domínio no conjunto dos inteiros positivos, a notação usual para análise de custo computacional é (Horowitz & Sahni, 1978):

- $f(n) = O(g(n))$ se existem constantes $c, n_0 > 0$ tais que $|f(n)| \leq c |g(n)|, \forall n > n_0$. Diz-se que $f(n)$ é de *ordem no máximo* $g(n)$;
- $f(n) = \Omega(g(n))$ se existem constantes $c, n_0 > 0$ tais que $|g(n)| \leq c |f(n)|, \forall n > n_0$. Diz-se que $f(n)$ é de *ordem pelo menos* $g(n)$;
- $f(n) = \Theta(g(n))$ se $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$. Diz-se que $f(n)$ é de *ordem* $g(n)$;
- $f(n) = o(g(n))$ se $f(n) / g(n) \rightarrow 0$ quando $n \rightarrow \infty$.

Com exceção do primeiro algoritmo a ser apresentado, denominado SAND, todas as outras três ferramentas de engenharia imunológica a serem propostas (CLONALG, ABNET e aiNet) possuem duas etapas em comum: um mecanismo de seleção do(s) melhor(es) indivíduo(s) (competição), e um mecanismo de variação genética (mutação) dos indivíduos da população. Dada uma população \mathbf{Ab} composta por N anticorpos, n membros desta população são selecionados, produzindo uma matriz $\mathbf{Ab}_{\{n\}}$, e reproduzidos gerando um conjunto \mathbf{C} de clones. O conjunto \mathbf{C} de clones irá sofrer um processo de mutação com taxas elevadas, gerando um conjunto \mathbf{C}^* . Para simplificar o estudo de cada algoritmo individualmente, apresentaremos uma análise genérica dos operadores de seleção e mutação empregados.

5.1.2.1. Complexidade Computacional dos Operadores de Seleção

Suponha que N_c clones são gerados a partir da seleção dos n melhores indivíduos de \mathbf{Ab} ($\mathbf{Ab} \in S^{N \times L}$), e que seus valores de afinidade, ou fitness, estão armazenados em um vetor \mathbf{f} . A forma mais comum de selecionar os n melhores indivíduos é ordenando o vetor de afinidades de forma que estes n melhores indivíduos possam ser extraídos facilmente. Este processo de ordenação pode ser executado em um tempo $O(N_c \cdot \log N_c)$ utilizando-se por exemplo o algoritmo *heapsort* (Sedgewick, 1988). Note que não é necessário ordenar todo o vetor, é possível criar uma pilha com custo $O(N_c)$, selecionar o melhor elemento e aplicar um mecanismo de reparação da pilha ao restante dos elementos. Como este passo deve ser executado n vezes e o processo de reparação da pilha requer um tempo $O(\log N_c)$, toda a rotina demanda um tempo limitado a $O(N_c + n \cdot \log N_c)$. Outros algoritmos de ordenação (*sort*) podem ser empregados, onde Fischetti & Martello (1988) apresentam um método de *quicksort* sofisticado capaz de extrair os n melhores elementos em um tempo $O(N_c)$. A quantidade de memória requerida por qualquer um destes algoritmos é $O(N_c)$.

Se desejarmos obter apenas o melhor indivíduo de uma população de tamanho N_c , ao invés dos n melhores, o custo computacional do algoritmo fica reduzido a $O(\log N_c)$.

5.1.2.2. Complexidade Computacional dos Operadores de Mutação

Primeiramente, é suposto que o operador de mutação é aplicado a uma cadeia de atributos de comprimento L (L -upla) e que esta operação é executada localmente para cada cadeia.

Seja $p_i \in (0,1]$ a probabilidade de que um componente $i = 1, \dots, L$ sofra uma mutação elementar. A mutação ocorre da seguinte forma: para cada componente i , sorteie um número aleatório uniformemente distribuído, $u \in (0,1)$, e mute o componente i caso $u \leq p_i$, do contrário mantenha-o inalterado. Esta etapa requer um tempo $\Theta(L)$. Entretanto, considerar que mutações elementares requerem um tempo constante nem sempre é apropriado. Por exemplo, seja $\mathbf{x} \in \mathbf{X} = \mathcal{R}^L$ e seja $\mathbf{z} \sim N(0, \mathbf{C})$ um vetor aleatório com distribuição normal de média zero e matriz de covariância \mathbf{C} simétrica e definida positiva. A menos que \mathbf{C} seja diagonal, os componentes do vetor aleatório \mathbf{z} estão correlacionados. Como são necessários $O(L)$ números aleatórios com distribuição normal e $O(L^2)$ operações elementares para construir o vetor aleatório \mathbf{z} , a operação de mutação completa $\mathbf{x} + \mathbf{z}$ requer um tempo $O(L^2)$. Consequentemente, uma operação elementar de mutação requer um tempo $O(L)$ (Bäck *et al.*, 2000b). A execução da operação de mutação sobre todos os N indivíduos

da população requer um tempo $O(N.L)$. De forma similar, a quantidade de memória necessária para realizar a mutação de uma população de N indivíduos de comprimento L é $O(N.L)$.

É importante mencionar que em alguns dos algoritmos a serem propostos (como ABNET e aiNet), o processo de mutação é determinístico, ou seja, a mutação é direcionada de forma a aumentar a afinidade Ag-Ab. Esse procedimento elimina o custo computacional necessário para sortear um número aleatório $u \in (0,1)$, mas requer um mecanismo de determinação das posições i apropriadas para direcionar a mutação.

5.1.3. Funções Para Pseudocódigos

Para apresentar os algoritmos de forma didática e tal que eles possam ser reproduzidos com facilidade, serão apresentados *pseudocódigos* das rotinas implementadas para cada uma das ferramentas propostas. Os algoritmos serão introduzidos sob a forma de funções genéricas, onde um conjunto de parâmetros de entrada/saída deverão ser atribuídos. Um conjunto de funções básicas e suas respectivas descrições estão apresentadas a seguir (entre parênteses encontram-se os nomes dos algoritmos que utilizam estas funções):

1. **Ab** := gera (N, L) ;
Gera uma população **Ab** contendo N anticorpos de comprimento L cada um. É válida para qualquer espaço de formas. Pode ser implementada por qualquer um dos modelos de medula óssea propostos na Seção 3.4.3.1. (Todos)
2. **f** := afinidade (**Ab**, **Ag**($j, :$)) ;
Calcula a afinidade **f** da população **Ab** em relação ao antígeno **Ag**. Pode ser diretamente ou inversamente proporcional à distância entre eles, dependendo do algoritmo e/ou aplicação, seguindo as regras apresentadas na Seção 3.4.2. (CLONALG, ABNET, aiNet)
3. **f** := decodifica (**Ab**) ;
Decodifica e calcula a afinidade **f** da população **Ab** (cada elemento do vetor **f** representa a afinidade de um único indivíduo da população **Ab**) em relação a uma função objetivo $g(\cdot)$ a ser otimizada. (CLONALG)
4. **Ab**_{ n } := select (**Ab**, **f**, n) ;
Seleciona n anticorpos da população **Ab** proporcionalmente à afinidade **f**, gerando uma subpopulação **Ab**_{ n }. (CLONALG, ABNET, aiNet)
5. **E** := energy (**Ab**) ;
Calcula a energia da população **Ab** de acordo com as Equações (5.5) e (5.9). (SAND)
6. **C** := clona (**Ab**_{ n }, β , **f**) ;
Clona (reproduz) os elementos da população **Ab**_{ n } proporcionalmente à afinidade **f** e a um fator multiplicativo β , gerando uma população **C**. (CLONALG, aiNet)

7. $\mathbf{C}^* := \text{hypermut}(\mathbf{C}, \mathbf{f})$;
Muta os elementos da população \mathbf{C} proporcionalmente à afinidade \mathbf{f} , gerando uma população \mathbf{C}^* , de acordo com os mecanismos apresentados na Seção 3.4.3.4. (SAND, CLONALG)
8. $\mathbf{C}^* := \text{dmut}(\mathbf{C}, \mathbf{Ag}, \mathbf{f})$;
Mutação direcionada. Os elementos da população \mathbf{C} são mutados proporcionalmente à afinidade \mathbf{f} visando aumentar a afinidade ao(s) antígeno(s) \mathbf{Ag} , gerando uma população \mathbf{C}^* . (ABNET, aiNet)
9. $\mathbf{Ab}_{\{m\}} := \text{insere}(\mathbf{Ab}_{\{m\}}, \mathbf{Ab}_{\{d\}})$;
Insere a matriz $\mathbf{Ab}_{\{d\}}$ na população $\mathbf{Ab}_{\{m\}}$ ou, dito de outra forma, concatena a matriz $\mathbf{Ab}_{\{m\}}$ com a matriz $\mathbf{Ab}_{\{d\}}$. (CLONALG, aiNet)
10. $\mathbf{Ab}_{\{r\}} := \text{replace}(\mathbf{Ab}_{\{r\}}, \mathbf{Ab}_{\{d\}}, \mathbf{f})$;
Substitui d anticorpos de $\mathbf{Ab}_{\{r\}}$ pelos d elementos da população $\mathbf{Ab}_{\{d\}}$ proporcionalmente à afinidade \mathbf{f} . (CLONALG, aiNet)
11. $[\mathbf{Ab}, \alpha] := \text{split}(\mathbf{Ab}, \mathbf{v}, \tau, \varepsilon, \alpha_0)$;
Clona (reproduz) um elemento de \mathbf{Ab} proporcionalmente aos parâmetros \mathbf{v} , τ , e ε (Seção 5.4), gerando um único clone. O parâmetro α_0 é passado de forma a permitir o retorno de $\alpha := \alpha_0$. (ABNET)
12. $[\mathbf{Ab}, \alpha] := \text{poda}(\mathbf{Ab}, \tau, t, \beta, \alpha_0)$;
Elimina um elemento de \mathbf{Ab} se $\tau = 0$ durante β iterações consecutivas. O parâmetro α_0 é passado de forma a permitir o retorno de $\alpha := \alpha_0$. (ABNET)
13. $[\mathbf{Ab}_{\{m\}}, \mathbf{S}] := \text{suppress}(\mathbf{Ab}_{\{m\}}, \mathbf{S}, \sigma_s)$;
Elimina elementos de $\mathbf{Ab}_{\{m\}}$ cuja afinidade $s_{i,j}$ é inferior ao parâmetro σ_s , gerando uma nova população $\mathbf{Ab}_{\{m\}}$ e uma nova matriz de afinidades \mathbf{S} . (aiNet)
14. $\mathbf{S} := \text{afinidade}(\mathbf{Ab}, \mathbf{Ab})$;
Calcula a afinidade \mathbf{S} da população \mathbf{Ab} em relação a si própria. Neste caso, a afinidade é inversamente proporcional à distância entre os anticorpos. (aiNet)

5.2. SAND: Simulated Annealing Aplicado ao Problema de Diversidade Populacional

Nesta seção, apresentamos uma abordagem baseada no algoritmo de *Simulated Annealing* (Kirkpatrick *et al.*, 1987) cujo objetivo principal é gerar uma população de candidatos à solução que apresenta uma ampla cobertura do espaço de buscas. A estratégia não emprega conhecimento prévio do problema a ser tratado e produz uma população de candidatos à solução com cardinalidade fixa e conhecida. Este algoritmo foi inicialmente proposto por de Castro & Von Zuben (1999a). O algoritmo, denominado SAND (*Simulated ANnealing approach for Diversity*), possui duas versões: uma primeira para espaços de formas binários de Hamming e outra para espaços Euclidianos.

5.2.1. Motivação e Escopo de Aplicações

O problema de geração de condições iniciais ótimas, ou seja, de geração de candidatos à solução, possui aplicabilidade em diversas áreas, desde os métodos de otimização mais clássicos, como métodos de gradiente, até as abordagens de sistemas inteligentes, particularmente as redes neurais artificiais e os algoritmos evolutivos discutidos no Capítulo 4.

Uma das principais características do sistema imunológico é sua capacidade de gerar receptores antigênicos altamente diversificados sendo, portanto, capazes de reconhecer uma enorme quantidade de moléculas. Este mecanismo é *antecipatório*, no sentido de que nenhum conhecimento prévio sobre os antígenos precisa ser considerado para a geração dos receptores (anticorpos). Entretanto, em qualquer momento da vida de um indivíduo, o sistema imunológico mantém uma quantidade finita de elementos (células e moléculas) circulantes e, portanto, uma variedade limitada de receptores. Sendo assim, é crucial que o sistema imunológico otimize o uso (e aplicação) destes recursos limitados, mantendo receptores em regiões “estratégicas” do espaço de possíveis antígenos.

O algoritmo SAND induz diversidade em uma população de tamanho fixo através da maximização de funções de energia que consideram a afinidade entre os elementos que compõem um repertório de anticorpos. A população de antígenos a ser posteriormente reconhecida é considerada desconhecida. Portanto o método opera de forma antecipatória como no caso do sistema imunológico biológico. Para a maximização da diversidade do repertório de anticorpos, são propostas duas *funções de energia* a serem utilizadas no processo iterativo de busca do algoritmo de Simulated Annealing. O algoritmo pode ser aplicado aos processos de geração de condições iniciais para ferramentas de engenharia imunológica e de outras áreas, como conjuntos de pesos iniciais para o treinamento de RNAs e até populações de cromossomos a serem utilizadas em algoritmos de computação evolutiva.

5.2.2. Simulated Annealing

O algoritmo de *Simulated Annealing* (Recozimento Simulado) faz uma conexão entre a mecânica estatística e a otimização combinatorial (Kirkpatrick *et al.*, 1987; Haykin, 1999). A origem do método está associada a propriedades agregadas de uma grande quantidade de átomos encontrados em amostras de líquidos ou em elementos sólidos. O comportamento do sistema em equilíbrio térmico, a uma dada temperatura, pode ser caracterizado experimentalmente por pequenas flutuações em torno de um comportamento médio.

A temperatura é simplesmente um parâmetro de controle que produz efeitos na função de energia (ou função de custo). O processo de simulated annealing consiste em inicialmente “aquecer” o sistema sendo otimizado a uma temperatura bastante elevada, e depois reduzir gradativamente a temperatura até que o sistema “congele” e nenhuma variação possa ser observada. Etapas de elevação de temperatura também podem ser incorporadas. A cada temperatura, a simulação deve proceder por tempo suficiente para que o sistema atinja um estado estacionário.

A cada iteração deste algoritmo, uma pequena perturbação aleatória é dada em um átomo, e a variação ΔE resultante na energia do sistema é calculada. Se $\Delta E \leq 0$, a perturbação é aceita, e a nova configuração é utilizada como condição inicial da próxima iteração. O caso $\Delta E > 0$ é tratado probabilisticamente: a probabilidade de que a configuração seja aceita é dada pela Equação (5.1):

$$P(\Delta E) = \exp(-\Delta E/T), \quad (5.1)$$

onde E é a energia da configuração, T a temperatura e ΔE uma pequena perturbação na energia medida.

Assim, transições para fora de ótimos locais são sempre possíveis a temperaturas não nulas. A seqüência de temperaturas e a dimensão das variações ΔE necessárias para atingir o equilíbrio a cada temperatura são denominadas de *annealing schedule*.

5.2.3. Descrição e Análise do SAND

Seja a seguinte notação:

- T : temperatura da configuração;
- \mathbf{Ab} : repertório total de N anticorpos ($\mathbf{Ab} \in S^{N \times L}$);
- \mathbf{Ab}_T : repertório temporário gerado pela perturbação de \mathbf{Ab} ($\mathbf{Ab}_T \in S^{N \times L}$);
- E : energia da configuração \mathbf{Ab} ;
- E_T : energia da configuração temporária \mathbf{Ab}_T ;
- α : taxa de hipermutação a ser aplicada na geração de \mathbf{Ab}_T ; e
- β : decrescimento geométrico da temperatura (*cooling schedule*).

Dentro do escopo de engenharia imunológica, o algoritmo SAND pode ser descrito como a seguir:

1. Gere um repertório \mathbf{Ab} inicial de N anticorpos;
2. Calcule a energia E deste repertório (baseado nas funções a serem descritas posteriormente);
3. Provoque uma pequena perturbação no repertório \mathbf{Ab} , proporcional a α , gerando um repertório temporário \mathbf{Ab}_T e reduza a taxa α de mutação;
4. Calcule a energia E_T deste repertório \mathbf{Ab}_T e avalie ΔE ;
 - 4.1. Caso $\Delta E < 0$, aceite a perturbação e retorne ao Passo 3;
 - 4.2. Caso $\Delta E > 0$, trate probabilisticamente, de acordo com a Equação (5.1), e retorne ao Passo 3;
 - 4.3. Caso $\Delta E = 0$, avalie a quantidade de iterações (ct) na qual o sistema permanece em estado estacionário. Caso este número de iterações ultrapasse um limiar pré-definido δ , reduza a temperatura T do sistema, restaure a taxa α de mutação e retorne ao Passo 3; caso contrário, retorne ao Passo 3, mas mantenha temperatura e mutação constantes.

O fluxograma da Figura 5.2 descreve o comportamento do algoritmo.

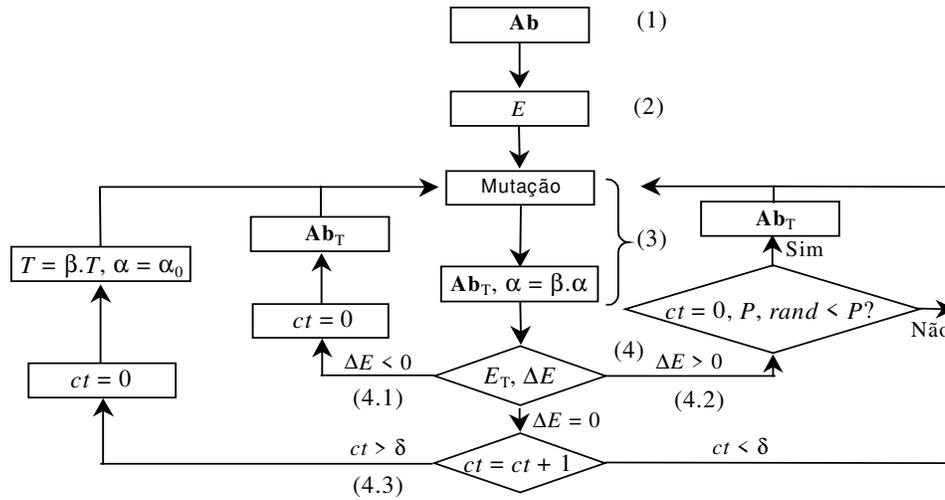


Figura 5.2. Diagrama de blocos do algoritmo SAND (etapas de elevação da temperatura T não foram incorporadas).

5.2.3.1. Pseudocódigo

Considere a implementação de uma função computacional para simular o algoritmo SAND. Esta função pode ter como argumentos os seguintes parâmetros:

Entrada:

- Comprimento L dos anticorpos;
- O tamanho N da população de anticorpos a ser gerada;
- Quantidade gen de iterações a serem executadas (possível critério de parada);
- A taxa β de decrescimento geométrico da temperatura;
- Limiar δ de iterações de estado estacionário para a redução da temperatura; e
- O valor máximo mE da energia do sistema (geralmente muito próximo de 100%, a ser descrito pelas Equações (5.5) e (5.9)).

Saída:

- Repertório \mathbf{Ab} de anticorpos otimizados (maximamente diversificados).

Dados os parâmetros de entrada e saída, é possível descrever um pseudocódigo para a implementação do SAND como a seguir:

```
function [Ab] = sand(N, L, gen, β, δ, mE);
t := 0;
ct := 0;
T := 1;
α0 := 1;
Ab := gera(N, L); % Passo 1
% A cada iteração, faça
while t < gen & E < mE,
    E := energy(Ab); % Passo 2
    AbT := hypermut(Ab, α); % Passo 3
    ET := energy(AbT); % Passo 4
```

```

dE := E - ET; % Passo 4
if dE < 0, % Passo 4.1
    Ab := AbT;
    E := ET;
    ct := 0;
elseif dE > 0, % Passo 4.2
    if rand < exp(-dE/T),
        Ab := AbT;
        E := ET;
        ct := 0;
    end,
else, % Passo 4.3
    ct := ct + 1;
    if ct mod δ = 0,
        T := β * T;
        ct := 0;
        α := α0;
    end;
    α := β * α;
end;
end;

```

5.2.3.2. Espaço de Formas de Hamming

Para implementação do algoritmo, é preciso propor as equações para o cálculo da energia do sistema nos espaços de formas de Hamming e Euclidiano.

No primeiro caso, ou seja, em um espaço de formas binário de Hamming, dado um repertório de N anticorpos de comprimento L , considere as seguintes equações:

$$s(i) = \begin{cases} 1, & \mathbf{Ab}_i \neq \mathbf{Ab}_j, \forall j \\ 0, & \text{outros casos} \end{cases}, \quad (5.2)$$

$$D(\%) = 100 \cdot \frac{1}{N} \times \sum_{i=1}^N s(i), \quad (5.3)$$

$$F(\%) = 100 \cdot \frac{4}{L \cdot N^2} \times \sum_{i=1}^N \sum_{j=i+1}^N HD(i, j), \quad (5.4)$$

$$E(\%) = \frac{F(\%) + D(\%)}{2}. \quad (5.5)$$

Lembrando que o objetivo é gerar e manter a diversidade, cada uma destas equações contribui para a determinação da energia do sistema da seguinte forma:

- O vetor $s(i)$, $i = 1, \dots, N$, na Equação (5.2), apresenta valor 1 somente quando cada anticorpo \mathbf{Ab}_i for diferente de todos os outros anticorpos do repertório, ou seja, cada anticorpo possui uma única especificidade;
- $D(\%)$, na Equação (5.3), é o percentual de indivíduos distintos em relação a todo o repertório;

- Na Equação (5.4), $F(\%)$ é a distância de Hamming (dada pela Equação 3.3) percentual entre todos os anticorpos do repertório. F será máximo ($F = 100\%$) quando, para $N \leq 2^L$, a distância de Hamming entre todos os anticorpos do repertório for máxima. Para $N > 2^L$, F nunca atingirá seu valor máximo, pois haverá pelo menos $N - 2^L$ anticorpos repetidos no repertório. Oprea (1999) demonstrou que a distância de Hamming máxima entre os elementos de um repertório binário de anticorpos é dada por $L.N/(2.(N-1))$, critério que pode ser empregado como parte da função de energia do SAND para espaços de formas de Hamming; e
- $E(\%)$, dado pela Equação (5.5), é a energia da configuração que mede a diversidade do repertório \mathbf{Ab} .

Note que se o número de indivíduos na população é menor do que a quantidade total de indivíduos diferentes que podem ser gerados para uma cadeia binária de atributos de comprimento L ($N = 2^L$), então diferentes populações podem surgir para a mesma energia máxima (100%), e o repertório ótimo não é único. Esta população de tamanho limitado pode nunca alcançar uma máxima cobertura do espaço de formas, para $\varepsilon = 0$ (limiar de afinidade nulo), mas certamente irá atingir algum estado próximo da máxima cobertura possível, dado seu tamanho restrito e valor de ε pré-definidos.

5.2.3.3. Espaço de Formas Euclidiano

Para que possamos aplicar o algoritmo SAND a espaços de formas com valores reais, uma outra função de energia deve ser proposta. Neste caso, medidas de distância, ou afinidade, Euclidiana ou de Manhattan podem ser empregadas.

Considerando um espaço de formas Euclidiano, a nova medida de energia a ser otimizada pode ser simplesmente definida como a soma das distâncias Euclidianas entre todos os vetores que representam os anticorpos do repertório imunológico:

$$E = \sum_{i=1}^N \sum_{j=i+1}^N ED(i, j), \quad (5.6)$$

onde ED é uma matriz quadrada e simétrica que contém a distância Euclidiana entre todos os anticorpos do repertório, de acordo com a Equação 3.1.

No caso dos espaços de Hamming, o critério de parada adotado para o algoritmo SAND pode ser $E = 100\%$. No espaço Euclidiano, a medida de energia não é um valor percentual e, portanto, outro critério de parada considerando a diversidade populacional deve ser apresentado.

A abordagem proposta, entre outras possibilidades, envolve a análise de dados direcionais. Dado um conjunto de anticorpos (vetores) \mathbf{Ab}_i , $i = 1, \dots, N$, inicialmente é necessário transformá-los em vetores unitários, ou normalizá-los, resultando em um conjunto \mathbf{I} de N vetores unitários de comprimento L ($\mathbf{I}_i \in \mathfrak{R}^L$, $i = 1, \dots, N$). O vetor direcional médio é

$$\bar{\mathbf{I}} = \frac{1}{N} \sum_{i=1}^N \mathbf{I}_i. \quad (5.7)$$

Supondo uma medida de distância Euclidiana, uma métrica para determinar a diversidade máxima dentro de um conjunto de vetores unitários pode ser simplesmente dada por

$$\bar{R} = (\bar{\mathbf{I}}^T \cdot \bar{\mathbf{I}})^{1/2}, \quad (5.8)$$

onde T representa a operação de transposição matricial e \bar{R} corresponde à distância do vetor médio à origem do sistema de coordenadas. A Equação (5.8) representa a amplitude do vetor resultante, porém nada pode ser inferido sobre a posição relativa entre os anticorpos da população (vetores individuais).

O critério de parada SC a ser empregado está baseado no índice I_{SC}

$$I_{SC}(\%) = 100 \cdot (1 - \bar{R}). \quad (5.9)$$

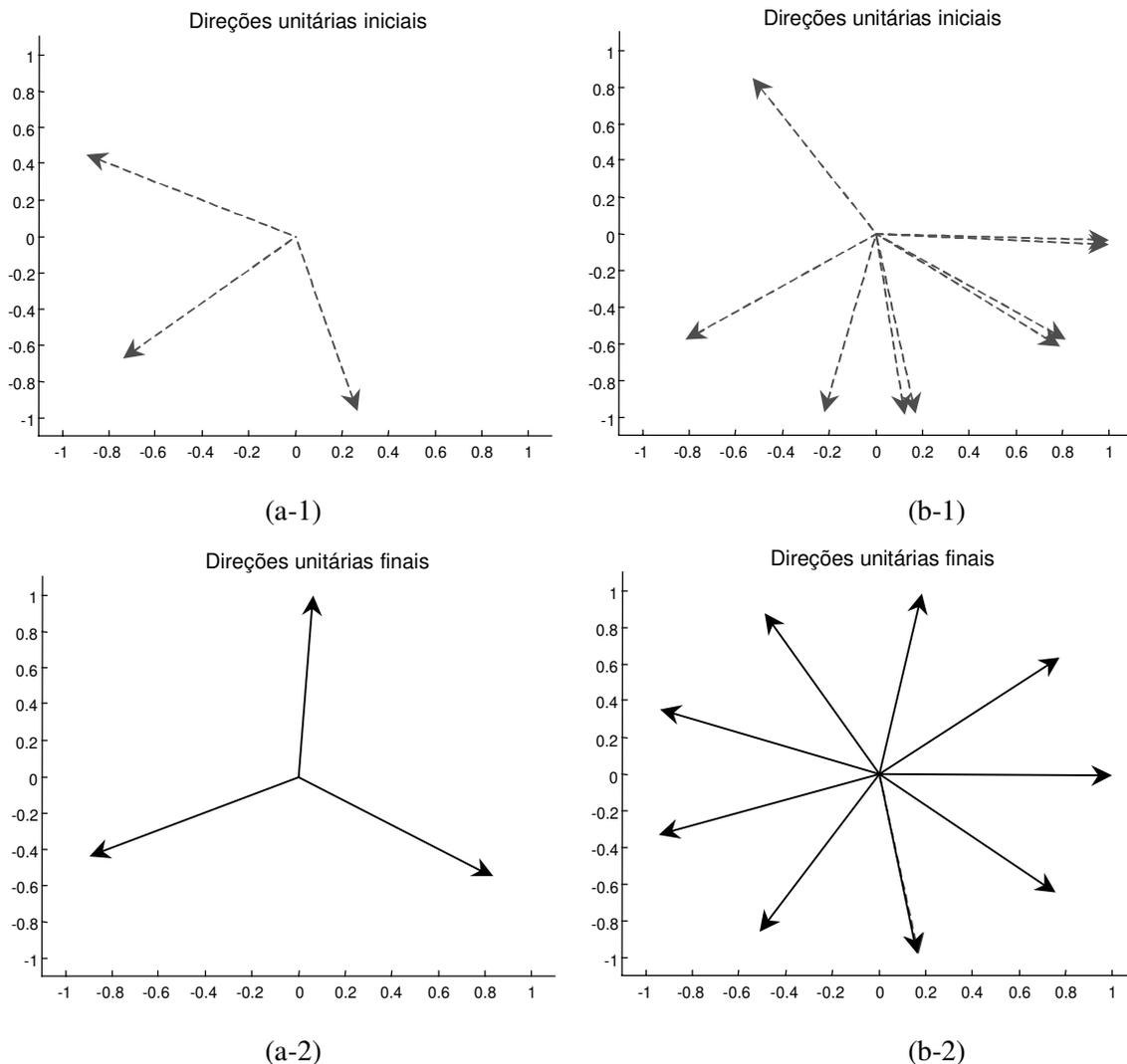


Figura 5.3. Geração de direções unitárias utilizando o algoritmo SAND para os espaços de formas Euclidianos. (a-1, a-2) Três vetores ($N = 3$). (b-1, b-2) Nove vetores ($N = 9$). As setas tracejadas indicam o conjunto inicial de vetores (direções) (a-1,b-1), e as setas sólidas representam os vetores (direções) otimizados (a-2, b-2).

A Equação (5.9) corresponde à medida percentual de diversidade e será igual a 100% quando \overline{R} for zero. Em termos práticos, um valor de I_{SC} próximo de 100% para o critério de parada é uma escolha razoável.

A geração de um repertório diverso de anticorpos em \mathfrak{R}^L corresponde a produzir um conjunto de direções que esteja bem distribuídas em uma hiperesfera. Para propósitos de visualização, considere o caso de gerar um conjunto de vetores em \mathfrak{R}^2 que apresente a melhor cobertura deste espaço. A Figura 5.3(a) e (b) apresenta os vetores evoluídos (otimizados) para $N = 3$ e $N = 9$, respectivamente. Note que o algoritmo é capaz de gerar direções com cobertura máxima do espaço \mathfrak{R}^2 . O algoritmo SAND permite a obtenção de resultados similares para quaisquer valores de N e L , o que não pode ser obtido por nenhum outro algoritmo conhecido na literatura, principalmente para valores elevados de L .

5.2.4. Caracterização e Comparação

É preciso ressaltar algumas características fundamentais deste algoritmo. Primeiramente, foi discutido nos Capítulos 2 e 3 que o processo de reconhecimento antigênico efetuado pelo sistema imune adaptativo dá-se pela complementaridade de formas entre um antígeno e um anticorpo, ou seja, quanto mais complementares as moléculas, maiores suas afinidades. Sem perda de generalidade, o algoritmo SAND considera a afinidade entre anticorpos como sendo inversamente proporcional à distância entre eles, ou seja, o SAND tem por objetivo gerar anticorpos cujas formas (representadas por cadeias de atributos ou vetores) sejam as mais complementares possíveis. Neste caso, a afinidade entre eles é mínima, evitando que um anticorpo i reconheça um anticorpo j do repertório cujos atributos são complementares a ele. Entretanto, como o algoritmo SAND gera um repertório de anticorpos com máxima distância (complementaridade) entre seus elementos, o conceito de reconhecimento antigênico por complementaridade de formas pode ser diretamente empregado para uma dada população antigênica e repertório resultante. De forma recíproca, a afinidade Ag-Ab também pode ser avaliada por uma medida de similaridade, permitindo concluir que a estratégia adotada pelo SAND é aplicável aos dois casos. A avaliação de afinidade empregando medida de similaridade, ao invés de complementaridade, tem sido bastante utilizada em sistemas imunológicos artificiais, como em Hajela & Lee (1996), Oprea (1999) e Harmer & Lamont (2000), dentre vários outros.

A determinação da taxa de mutação α é um parâmetro crítico neste algoritmo e pode seguir os padrões propostos para os procedimentos de hipermutação somática durante uma resposta imune adaptativa, onde indivíduos com alta afinidade antigênica sofrem baixas taxas de mutação e vice-versa. No caso do SAND, a população como um todo determina uma única afinidade (energia).

De acordo com a descrição da Seção 5.2.3, a taxa de mutação α segue um padrão similar ao proposto por Kepler & Perelson (1993a,b), onde gerações com baixas taxas de mutação são intercaladas por picos de mutações, e mutações benéficas (capazes de aumentar a afinidade antigênica) são adquiridas seqüencialmente ao invés de simultaneamente. A Figura 5.4 ilustra o comportamento da taxa de mutação para a versão de Hamming do algoritmo SAND, dada a seguinte linha de comando: `[Ab] = sand(128, 7, 200, 0.9, 3, 100)`.

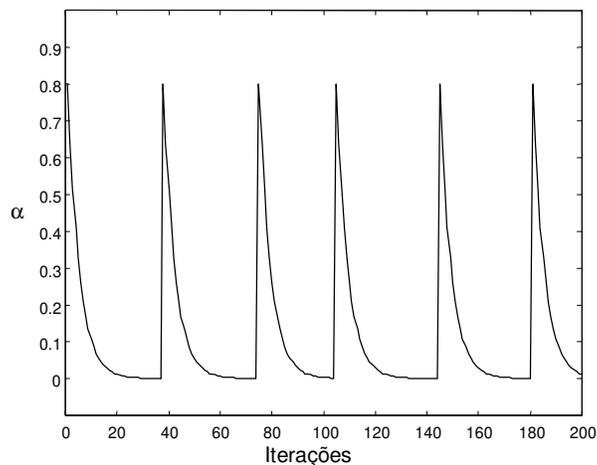


Figura 5.4. Comportamento típico da taxa de mutação (α) para o algoritmo SAND.

Em aplicações típicas de algoritmos genéticos (Seção 4.4), a estratégia evolutiva produz indivíduos cada vez mais aptos (com maior fitness) dentre uma vasta população de candidatos à solução. O algoritmo é utilizado para evoluir uma população de candidatos à solução onde um único indivíduo irá especificar a melhor solução, ou seja, um pico de fitness. Os operadores de seleção e recombinação (crossover, mutação e inversão) empregados no GA guiam a população em direção ao seu melhor indivíduo (Mitchell, 1998). Nas aplicações do SAND, busca-se uma população com caráter cooperativo, que, sob a forma de um conjunto de indivíduos, seja capaz de realizar uma tarefa específica: maximizar a cobertura do espaço de buscas, baseado em uma população de tamanho fixo. Sob este ponto de vista, não existe um único indivíduo da população que possa ser considerado o melhor, mas toda a população irá representar a solução. Este objetivo sugere que o algoritmo genético clássico não será capaz de produzir o resultado desejado e, ao invés de desenvolver um GA capaz de manter populações diversas (Forrest *et al.*, 1993; Smith *et al.*, 1993; Mahfoud, 1995; Matsui, 1999), optamos pelo algoritmo de simulated annealing, que também possui a vantagem de ser menos custoso computacionalmente para algumas aplicações.

A abordagem do SAND possui uma essência similar à abordagem Michigan empregada nos sistemas classificadores, propostos por Holland (1975). A abordagem Michigan pode ser vista como um modelo computacional da cognição, onde o conhecimento da entidade é expresso como um coleção de regras que interagem com o ambiente e sofre modificações. Assim como no nosso método de simulated annealing, todas as regras (ou seja, toda a população) representam um único candidato à solução.

5.2.5. Complexidade Computacional

O algoritmo SAND proposto para geração de diversidade possui duas etapas principais de processamento: 1) a determinação da energia da configuração (Passos 2 e 4), e 2) a mutação do repertório **Ab** de anticorpos (Passo 3).

Tabela 5.1. Complexidade computacional por geração para execução do algoritmo SAND, onde N é o tamanho da população de anticorpos de comprimento L .

	<i>Tempo de Processamento</i>	<i>Memória</i>
Espaço de Hamming	$O(N^2)$	$\propto N [2L + L(N - 1)/2 + 1]$
Espaço Euclidiano	$O(N^2)$	$\propto L [2N + N(N - 1)/2 + 1]$

O custo computacional do algoritmo é quadrático em relação à quantidade N de anticorpos, pois demanda a comparação de cada anticorpo do repertório com todos os outros (Equações (5.4) e (5.6)), possuindo um custo computacional associado da ordem de $O(N^2)$. O processo de mutação de toda a população tem custo computacional $O(N.L)$, como discutido na Seção 5.1.2. Assim, somando-se a complexidade das duas etapas e considerando a característica assintótica desta análise, é possível determinar a complexidade computacional total do algoritmo, como apresentado na Tabela 5.1.

Para a determinação da quantidade de memória necessária para executar o algoritmo, pode ser feita uma análise distinta para os espaços de Hamming e Euclidiano. Em ambos os casos, são armazenadas duas matrizes, \mathbf{Ab} e \mathbf{Ab}_T , contendo os repertórios de anticorpos atual e temporário, ambas de dimensão $S^{N \times L}$. Além disso, as matrizes que determinam a distância entre os anticorpos são simétricas e, portanto, é possível armazenar apenas suas partes triangulares superiores, de dimensão $S^{(N(N-1)/2) \times L}$. No caso dos espaços binários de Hamming, uma cadeia binária de dimensão L , também é armazenada e, para os espaços Euclidianos, um vetor resultante de comprimento L deve existir. A quantidade de memória associada a estes algoritmos é apresentada na Tabela 5.1.

5.3. CLONALG: Implementações Computacionais do Princípio de Seleção Clonal

Nesta seção, são descritas duas versões para a implementação computacional do princípio de seleção clonal, denominadas genericamente de CLONALG (*CLONal selection ALGORITHM*), considerando explicitamente o processo de maturação de afinidade. Este algoritmo foi originalmente proposto por de Castro & Von Zuben (2000a) e brevemente apresentado na Seção 3.4.3.3. Duas versões do algoritmo estão disponíveis: a primeira para resolver problemas de aprendizagem de máquina e reconhecimento de padrões e a segunda para resolver problemas de otimização (de Castro & Von Zuben, 2001a).

5.3.1. Motivação e Escopo de Aplicações

Nas Seções 2.6 e 2.8 verificou-se que o princípio de seleção clonal é utilizado para descrever as características básicas de uma resposta imune adaptativa a estímulos antigênicos. Ele estabelece a idéia de que somente as células capazes de reconhecer antígenos irão se proliferar, sendo assim selecionadas em detrimento das outras. Durante a reprodução celular, estas células (clones) estarão sujeitas a um processo de mutação

somática com taxas elevadas que, juntamente com uma forte pressão seletiva, resultará no aumento da afinidade entre o anticorpo e o antígeno que o selecionou. Dessa forma, é possível concluir que estes processos são parcialmente responsáveis pela aprendizagem e memória imunológica, características particularmente interessantes sob o ponto de vista de engenharia. Além da capacidade de aprendizagem e memória, foi argumentado, na Seção 4.5, que a evolução celular efetuada pelo sistema imunológico constitui uma espécie de microevolução, onde variações ontogenéticas (dentro do indivíduo) podem de alguma forma contribuir para uma maior adaptabilidade do indivíduo ao seu ambiente. Dadas essas características do processo biológico, é suposto que a implementação computacional do algoritmo de seleção clonal pode levar ao desenvolvimento de ferramentas adaptativas capazes de resolver problemas de aprendizagem de máquina, reconhecimento de padrões e otimização, em que os processos de seleção e maturação são responsáveis pela geração de indivíduos cada vez melhores adaptados ao ambiente no qual estão inseridos.

5.3.2. Descrição e Análise do CLONALG

O algoritmo foi inicialmente proposto para resolver problemas de aprendizagem de máquina e reconhecimento de padrões (antígenos), onde uma população aleatória de anticorpos está presente e tem por objetivo aprender a reconhecer um conjunto de antígenos. Dadas suas características adaptativas, o algoritmo foi estendido para aplicações a problemas de otimização.

5.3.2.1. Reconhecimento de Padrões

Para facilitar a descrição do algoritmo, considere o diagrama de blocos da Figura 5.5. Para o problema de reconhecimento de padrões, é suposta a existência de um repertório inicial aleatório de anticorpos com a finalidade de reconhecer uma população conhecida de antígenos. Como proposto por alguns imunologistas, e discutido na Seção 2.8.1, existe um conjunto de memória, composto por indivíduos específicos do repertório de anticorpos apresentando altas afinidades antigênicas, que estará funcionalmente desconectado do restante da população. Seja a seguinte notação:

- **Ab**: repertório de anticorpos disponíveis ($\mathbf{Ab} \in S^{N \times L}$, $\mathbf{Ab} = \mathbf{Ab}_{\{r\}} \cup \mathbf{Ab}_{\{m\}}$);
- $\mathbf{Ab}_{\{m\}}$: repertório de anticorpos de memória ($\mathbf{Ab}_{\{m\}} \in S^{m \times L}$, $m \leq N$);
- $\mathbf{Ab}_{\{r\}}$: restante do repertório de anticorpos ($\mathbf{Ab}_{\{r\}} \in S^{r \times L}$, $r = N - m$);
- **Ag**: população de M antígenos a serem reconhecidos ($\mathbf{Ag} \in S^{M \times L}$);
- f_j : vetor contendo a afinidade de todos os anticorpos em relação ao antígeno \mathbf{Ag}_j ;
- $\mathbf{Ab}_{\{n\}}^j$: subconjunto de **Ab** composto pelos n anticorpos de maior afinidade a \mathbf{Ag}_j ($\mathbf{Ab}_{\{n\}}^j \in S^{n \times L}$, $n \leq N$);
- \mathbf{C}^j : população de N_c clones gerada a partir de $\mathbf{Ab}_{\{n\}}^j$ ($\mathbf{C}^j \in S^{N_c \times L}$);
- \mathbf{C}^{j*} : população \mathbf{C}^j após o processo de maturação de afinidade;
- $\mathbf{Ab}_{\{d\}}$: d anticorpos com baixa afinidade antigênica de $\mathbf{Ab}_{\{r\}}$ serão substituídos por $\mathbf{Ab}_{\{d\}}$ novos anticorpos ($\mathbf{Ab}_{\{d\}} \in S^{d \times L}$, $d \leq r$); e
- $\mathbf{Ab}_{i^{j*}}$: candidato de \mathbf{C}^{j*} a fazer parte do conjunto de memória.

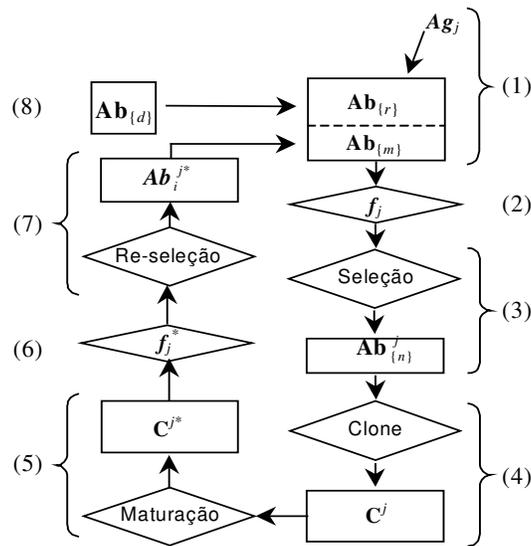


Figura 5.5. Diagrama de blocos do algoritmo de seleção clonal, CLONALG, para o problema de reconhecimento de padrões.

O algoritmo pode ser descrito como a seguir (Figura 5.5). Dada a existência de um repertório \mathbf{Ab} de anticorpos composto por um conjunto de memória $\mathbf{Ab}_{\{m\}}$ mais o restante $\mathbf{Ab}_{\{r\}}$ da população, e supondo, sem perda de generalidade, que a quantidade m de anticorpos de memória é igual à quantidade M de antígenos a serem reconhecidos.

1. Em um instante de tempo t , um antígeno \mathbf{Ag}_j ($\mathbf{Ag}_j \in \mathbf{Ag}$) é apresentado a todos os anticorpos da população \mathbf{Ab} ;
2. Um vetor de afinidades f_j em relação aos anticorpos da população \mathbf{Ab} é determinado;
3. Do conjunto \mathbf{Ab} , um subconjunto $\mathbf{Ab}_{\{n\}}^j$, composto pelos n anticorpos com maiores afinidades a \mathbf{Ag}_j , é selecionado baseado na afinidade $f_{i,j}$ de cada anticorpo \mathbf{Ab}_i em relação ao antígeno \mathbf{Ag}_j ;
4. Os n indivíduos selecionados irão se proliferar (clonagem) proporcionalmente às suas afinidades ao antígeno, gerando uma população \mathbf{C}^j de clones: quanto maior a afinidade, maior o número de clones de cada um dos n anticorpos;
5. Em seguida, a população \mathbf{C}^j de clones é submetida ao processo de maturação de afinidade gerando uma nova população \mathbf{C}^{j*} , onde cada anticorpo irá sofrer uma mutação com taxa inversamente proporcional a sua afinidade: quanto maior a afinidade, menor a taxa de mutação;
6. Determine a afinidade f_j^* entre o conjunto \mathbf{C}^{j*} de clones mutados e o antígeno \mathbf{Ag}_j ;
7. Desta população madura \mathbf{C}^{j*} , re-seleccione o melhor anticorpo para ser um candidato a entrar no conjunto de memória $\mathbf{Ab}_{\{m\}}$. Se a afinidade deste anticorpo \mathbf{Ab}_i^{j*} em relação a \mathbf{Ag}_j for maior do que a afinidade do respectivo anticorpo de memória, então \mathbf{Ab}_i^{j*} substitui este anticorpo de memória;
8. Finalmente, substitua d anticorpos de $\mathbf{Ab}_{\{r\}}$ por $\mathbf{Ab}_{\{d\}}$ novos indivíduos, induzindo diversidade no repertório (alguns membros de $\mathbf{Ab}_{\{d\}}$ podem ser escolhidos a partir

de \mathbf{C}^{j*} ou gerados utilizando um dos modelos de medula óssea propostos na Seção 3.4.3.1). Os anticorpos com menores afinidades são escolhidos para serem substituídos.

Após a apresentação de todos os antígenos ao repertório de anticorpos, diz-se que ocorreu uma *geração*.

Nesta implementação, a quantidade N_c de clones gerada no Passo 4 para todos os n anticorpos selecionados no Passo 3 é dada pela Equação (5.10):

$$N_c = \sum_{i=1}^n \text{round}(\beta.N / i), \quad (5.10)$$

onde β é um fator multiplicativo, N é a quantidade total de anticorpos do repertório \mathbf{Ab} e $\text{round}(\cdot)$ é o operador que arredonda o valor entre parênteses para o inteiro mais próximo. Neste caso, os anticorpos estão ordenados da maior para a menor afinidade, portanto quanto maior i na Equação (5.10), menor a afinidade e conseqüentemente a quantidade de clones gerada para \mathbf{Ab}_i .

5.3.2.1.1. Pseudocódigo

Considere a implementação de uma função computacional para simular o algoritmo CLONALG para o problema de aprendizagem de máquina e reconhecimento de padrões. Esta função pode ter como argumentos os seguintes parâmetros:

Entrada:

- Comprimento L dos antígenos e anticorpos;
- População \mathbf{Ab} composta por N anticorpos de comprimento L ;
- População \mathbf{Ag} composta por M antígenos de comprimento L ;
- Quantidade gen de gerações a serem executadas (critério de parada);
- Número n de anticorpos a serem selecionados para clonagem;
- Fator multiplicativo β usado na definição da quantidade de clones; e
- Quantidade d de anticorpos com baixa afinidade f , que serão substituídos.

Saída:

- Matriz $\mathbf{Ab}_{\{m\}}$ de anticorpos de memória; e
- Afinidade total \mathbf{F} entre os anticorpos de memória ($\mathbf{Ab}_{\{m\}}$) e a população de antígenos \mathbf{Ag} .

Dados os parâmetros de entrada e saída, é possível descrever um pseudocódigo para a implementação do CLONALG como a seguir:

```
function [ $\mathbf{Ab}_{\{m\}}$ ,  $\mathbf{F}$ ] = clonalg( $\mathbf{Ab}$ ,  $\mathbf{Ag}$ ,  $L$ ,  $gen$ ,  $n$ ,  $\beta$ ,  $d$ );
t := 0;
F := 0;
% A cada geração, faça:
while t < gen & F < (M*L),
    for j = 1 to M,
        f(j, :) := afinidade( $\mathbf{Ab}$ ,  $\mathbf{Ag}(j, :)$ ); % Passo 1
         $\mathbf{Ab}_{\{n\}}(j, :) := \text{select}(\mathbf{Ab}, \mathbf{f}, n)$ ; % Passo 2
         $\mathbf{Ab}_{\{n\}}(j, :) := \text{select}(\mathbf{Ab}, \mathbf{f}, n)$ ; % Passo 3
```

```

C(j, :) := clona(Ab{n},  $\beta$ , f); % Passo 4
C*(j, :) := hypermut(C, f); % Passo 5
f(j, :) := afinidade(C*, Ag(j, :)); % Passo 6
Ab* := select(C*, f, 1); % Passo 7
Ab{m}(j, :) := insere(Ab{m}(j, :), Ab*); % Abm ← Ab se f(Ab*) > f(Abm(j, :))
Ab{d} := gera(d, L); % Gera Abd aleatoriamente
Ab{x} := replace(Ab{x}, Ab{d}, f); % Passo 8
end;
f := afinidade(Ab{m}, Ag);
F := soma(f);
t := t + 1;
end;

```

5.3.2.2. Otimização

Para aplicar o algoritmo de seleção clonal proposto acima, CLONALG, a problemas de otimização, algumas poucas modificações devem ser efetuadas:

- No Passo (1) não existe uma população **Ag** de antígenos a serem reconhecidos, e sim uma função objetivo $g(\cdot)$ a ser otimizada. Dessa forma, a afinidade de um anticorpo (que pode ser interpretado como seu fitness, ou grau de adaptabilidade) passa a corresponder ao valor da função objetivo $g(\cdot)$ avaliada para um dado anticorpo: cada anticorpo Ab_i representa o valor codificado de um argumento da função $g(\cdot)$. Além disso, como não existe uma população específica de antígenos a serem reconhecidos, toda a população **Ab** corresponderá a anticorpos de memória ao final do processo, portanto não é necessário definir um subconjunto específico de memória **Ab**_{m};
- No Passo (7), n anticorpos serão selecionados para compor o conjunto **Ab**.

Se o objetivo do processo de otimização é determinar múltiplos ótimos dentro da população de anticorpos, então dois parâmetros podem apresentar valores previamente definidos para garantir um bom desempenho do algoritmo:

1. Supor $n = N$, ou seja, selecionar todos os anticorpos da população **Ab** para reprodução (clonagem) nos Passos (3) e (4), sendo cada anticorpo analisado localmente;
2. A reprodução proporcional à afinidade não necessariamente se aplica, significando que a quantidade de clones gerada para cada um dos N anticorpos é considerada a mesma, não privilegiando anticorpo algum por sua afinidade. A Equação (5.10), torna-se:

$$N_c = \sum_{i=1}^n \text{round}(\beta.N). \quad (5.11)$$

O diagrama de blocos resultante do algoritmo CLONALG para aplicação a problemas de otimização está resumido na Figura 5.6.

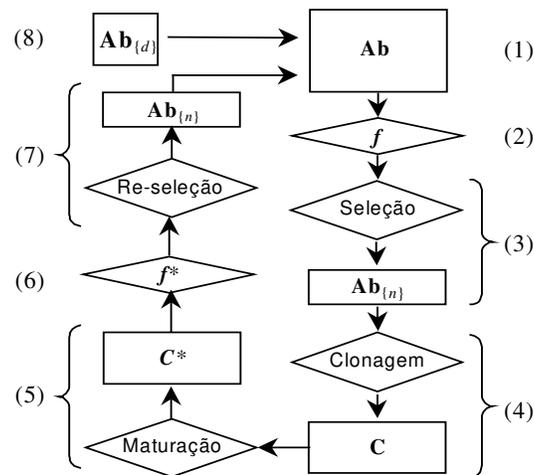


Figura 5.6. Diagrama de blocos do algoritmo de seleção clonal, CLONALG, para o problema de otimização.

5.3.2.2.1. Pseudocódigo

A implementação computacional do algoritmo CLONALG para otimização de funções pode ser feita utilizando o pseudocódigo abaixo, dados os parâmetros de entrada e saída:

Entrada:

- Comprimento L dos anticorpos;
- População \mathbf{Ab} composta por N anticorpos;
- Quantidade gen de gerações a serem executadas (critério de parada);
- Número n de anticorpos a serem selecionados para clonagem ($n = N$ para aplicações a problemas de otimização multimodal);
- Fator multiplicativo β usado na definição da quantidade de clones; e
- Quantidade d de anticorpos com baixa afinidade que serão substituídos.

Saída:

- Matriz \mathbf{Ab} , correspondente aos anticorpos de memória; e
- Afinidade (fitness) f de cada anticorpo da matriz \mathbf{Ab} .

```
function [Ab, f] = clonalg(Ab, L, gen, n,  $\beta$ , d)
% A cada geração, faça:
for t = 1 to gen,
    f      := decodifica(Ab) ;           % Passo 2
    Ab{n} := select(Ab, f, n) ;       % Passo 3
    C     := clona(Ab{n},  $\beta$ , f) ;    % Passo 4
    C*    := hypermut(C, f) ;        % Passo 5
    f     := decodifica(C*) ;          % Passo 6
    Ab{n} := select(C*, f, n) ;      % Passo 7
    Ab    := insere(Ab, Ab{n}) ;      % Ab ← Ab{n}
    Ab{d} := gera(d, L) ;              % Gera Ab{d} aleatoriamente
    Ab    := replace(Ab, Ab{d}, f) ; % Passo 8
end;
f := decodifica(Ab) ;
```

5.3.3. Caracterização e Comparação

O algoritmo CLONALG, como proposto acima, representa uma implementação computacional simplificada do princípio da seleção clonal dos linfócitos B durante uma resposta imune adaptativa (Seção 2.8). Tanto no caso de reconhecimento de padrões quanto para otimização, supõe-se que o organismo possui uma população \mathbf{Ab} de anticorpos que serão estimulados por um antígeno (que pode ser um elemento \mathbf{Ag}_i explícito a ser reconhecido, ou um valor da função objetivo a ser otimizada), e aqueles com maiores afinidades serão selecionados gerando uma população de clones. Durante o processo de clonagem, reprodução por mitose celular, alguns anticorpos sofrerão mutação genética com taxas elevadas e inversamente proporcionais a suas afinidades. Os melhores indivíduos dentre os mutados serão re-selecionados para compor um conjunto de memória. Indivíduos com baixa afinidade são substituídos, simulando o processo de apoptose e/ou edição de receptores.

Na Seção 4.4.2 apresentou-se os passos principais para a implementação de algoritmos evolutivos, resumidos sob a forma do diagrama de blocos da Figura 5.7. Enquanto os algoritmos evolutivos empregam um vocabulário inspirado na genética natural e são baseados na teoria da evolução neo-Darwiniana (Seção 4.4), o algoritmo de seleção clonal, CLONALG, utiliza o espaço de formas e a terminologia imunológica para descrever as interações de antígenos e anticorpos, no caso da aprendizagem de máquina e reconhecimento de padrões, e estudar a evolução celular no caso do procedimento para otimização. É possível verificar, comparando-se a Figura 5.7 com a Figura 5.5 ou a Figura 5.6, que o algoritmo CLONALG apresenta, dentre outras, as mesmas etapas de processamento dos algoritmos evolutivos, permitindo sua caracterização como uma estratégia evolutiva. Além disso, foi discutido na Seção 4.5 que a evolução dentro do sistema imunológico pode ser vista como uma microevolução da teoria Darwiniana.

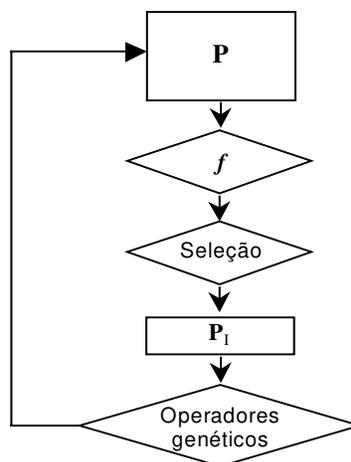


Figura 5.7. Diagrama de blocos de um algoritmo evolutivo básico. \mathbf{P} corresponde a população de indivíduos, f é o fitness da população e \mathbf{P}_1 a população intermediária. Os operadores genéticos englobam a reprodução e variação genética dos descendentes, via crossover, mutação, etc.

Assim como no processo de reprodução celular envolvido no princípio da seleção clonal (mitose), no algoritmo CLONALG, o único operador responsável pela introdução direta de variação genética no repertório celular é uma mutação cuja taxa é elevada e inversamente proporcional à afinidade Ag-Ab. Dessa forma, verifica-se que existe uma correlação entre a taxa de variação da população e a afinidade, ou fitness, dos indivíduos, resultando num processo de auto-adaptação da taxa de variação genética a ser introduzida. Outro aspecto importante deste modelo é que cada indivíduo da população é analisado localmente: um conjunto de clones é gerado com cardinalidade proporcional à sua afinidade (Equações (5.10) e (5.11)), e o processo de seleção é efetuado na população de clones após sua maturação. Este procedimento sugere que o algoritmo é apropriado para solucionar problemas de otimização multimodal, onde cada anticorpo irá explorar sua vizinhança.

As *estratégias evolutivas* (EEs) também empregam uma abordagem de auto-adaptação dos mecanismos de recombinação e mutação (Tabela 4.2). Rechenberg (1973, 1994) propôs uma “janela de evolução” (*evolution window*), na qual variações genéticas em um indivíduo (perturbações causadas pela mutação) resultam em um progresso evolutivo apenas quando eles estão contidos em uma faixa estreita, confinada e determinável. Passos de mutação e recombinação fora da janela de evolução são ineficientes. Sendo assim, as estratégias evolutivas permitem o ajuste dinâmico das taxas de mutação durante a evolução; processo chamado de *meta-evolução* ou *otimização de segundo tipo*. As estratégias evolutivas concentram na mutação o principal operador do processo de variação genética, e portanto de busca, e a recombinação é necessária principalmente para o controle da mutação. As EEs utilizam mutações Gaussianas e regras determinísticas para a variação do desvio padrão das mutações, que devem estar linearmente correlacionadas (Bäck & Schwefel, 1993). Além disso, elas também não fazem a codificação da população. Por outro lado, a abordagem proposta para o algoritmo CLONALG é plausível sob o ponto de vista biológico e segue o padrão proposto na Seção 3.4.3.4, Figura 3.16. O que deve ser verificado é que os valores a serem adotados para as taxas de recombinação genética de algoritmos evolutivos, incluindo recombinação e mutação, devem ser proporcionais ao fitness do indivíduo e adaptados através de algum mecanismo ou regra específica. Além da similaridade quanto à utilização de taxas adaptativas de mutação, o algoritmo CLONALG possui um mecanismo de seleção inspirado no princípio de seleção clonal do sistema imunológico biológico que se assemelha à seleção (μ, λ) das estratégias evolutivas. Na seleção (μ, λ) , λ descendentes são gerados para cada pai da população atual através de mutação e possivelmente recombinação e os μ melhores descendentes são selecionados. Outra diferença entre o algoritmo CLONALG e as EEs, é que o CLONALG trabalha com uma população de indivíduos codificados. Finalmente, as EEs foram originalmente propostas para solucionar problemas de otimização, enquanto o algoritmo CLONALG foi desenvolvido com o objetivo de implementar computacionalmente o princípio imunológico da seleção clonal para resolver problemas de reconhecimento de padrões e, posteriormente, adaptado a problemas de otimização, enfocando o caso multimodal.

Por fim, é possível caracterizar o algoritmo de aprendizagem CLONALG como uma abordagem competitiva (Seção 4.2.1.2.2), uma vez que os anticorpos estarão competindo pelo reconhecimento antigênico ou por uma maior afinidade.

5.3.4. Complexidade Computacional

O algoritmo de seleção clonal proposto, CLONALG, possui três etapas principais de processamento: 1) a determinação da afinidade (ou fitness) dos anticorpos da população (Passos 2 e 6), 2) a seleção (e re-seleção) dos n melhores indivíduos (Passos 3 e 7), e 3) a mutação da população \mathbf{C} de clones (Passo 5).

O custo computacional para a determinação da afinidade da população depende do problema abordado, e nenhuma medida genérica pode ser apresentada. Por outro lado, é possível calcular estimativas de complexidade computacional para os processos de seleção e mutação, como descrito na Seção 5.1.2. Assim, o tempo computacional envolvido no primeiro processo de seleção, Passo 3, é $O(N)$. No segundo processo de seleção, Passo 7, o custo computacional é da ordem $O(N_c)$, onde N_c corresponde à quantidade de clones gerados a partir de $\mathbf{Ab}_{\{n\}}$. Durante a mutação dos N_c clones, é necessário um tempo de processamento da ordem de $O(N_c.L)$, onde L é o comprimento da cadeia de atributos utilizada para representar os anticorpos. Somando-se a complexidade de cada etapa é possível determinar a complexidade computacional do algoritmo, como apresentado na Tabela 5.2.

Na versão desenvolvida para resolver problemas de reconhecimento de padrões, a complexidade computacional por geração do algoritmo aparece multiplicada pelo tamanho M da população de antígenos. Por outro lado, se o algoritmo está sendo aplicado a problemas de otimização multimodal com $n = N$, os processos de seleção podem ser suprimidos do algoritmo, reduzindo o custo computacional total do algoritmo para $O(N_c.L)$.

A quantidade de memória necessária para executar o algoritmo é proporcional à dimensão do vetor de afinidades \mathbf{f} ($\mathbf{f} \in \mathbb{R}^N$), somada à dimensão da matriz $\mathbf{Ab}_{\{n\}}$ ($\mathbf{Ab}_{\{n\}} \in S^{n \times L}$), mais a dimensão da matriz \mathbf{C} de clones ($\mathbf{C} \in S^{N_c \times L}$), mais a dimensão da matriz total de anticorpos \mathbf{Ab} ($\mathbf{Ab} \in S^{M \times L}$). No caso de reconhecimento de padrões, a memória requerida para executar o algoritmo aparece multiplicada pelos M antígenos. O resultado desta soma está apresentado na Tabela 5.2.

Tabela 5.2. Complexidade computacional por geração para execução do algoritmo CLONALG, onde N é o tamanho da população de anticorpos, N_c a quantidade de clones gerada a partir da seleção dos n melhores indivíduos de \mathbf{Ab} , L é o comprimento das cadeias de atributos, e M a quantidade de antígenos a serem reconhecidos.

	<i>Tempo de Processamento</i>	<i>Memória</i>
Reconhecimento de padrões	$O(M(N + N_c.L))$	$\propto M(N + L(n + N_c + N))$
Otimização	$O(N + N_c.L)$	$\propto N + L(n + N_c + N)$
Otimização multimodal	$O(N_c.L)$	

5.4. ABNET: Uma Rede de Anticorpos

Nesta seção será proposta uma rede neural Booleana, denominada ABNET (*AntiBody NETWORK*), ilustrando como diversos princípios imunológicos podem ser empregados no desenvolvimento de arquiteturas e algoritmos de treinamento para redes neurais artificiais. As principais características da ABNET englobam: aprendizagem competitiva, geração automática da estrutura da rede e representação binária (Booleana) do conjunto de pesos. Este modelo foi originalmente proposto por de Castro & Von Zuben (2001f) destacando os aspectos cognitivos do sistema imunológico como discutido na Seção 4.3.1, ou seja, os problemas de aprendizagem, reconhecimento de padrões e memória imunológica.

5.4.1. Motivação e Escopo de Aplicações

Até recentemente, a determinação adequada da arquitetura de uma rede neural artificial para um dado problema envolvia a experiência do projetista na implementação de procedimentos de tentativa e erro (de Castro *et al.*, 1999a). Atualmente, o projeto automático de arquiteturas de RNAs começa a fazer parte do processo de treinamento da rede, através de uma melhor exploração dos padrões de entrada.

O objetivo da ABNET é explorar qualitativamente o algoritmo de seleção clonal através da definição automática de um repertório binário de anticorpos, representado em um espaço de formas de Hamming, e modelado utilizando uma arquitetura de rede neural artificial. A arquitetura de rede empregada é do tipo feedforward com uma única camada (Seção 4.2.1.1.1), onde os neurônios de saída da rede e seus respectivos vetores de pesos associados representam os anticorpos do sistema imunológico, e o reconhecimento dos antígenos é feito por toda a rede, e não por anticorpos individuais. O princípio da seleção clonal será empregado para a determinação da arquitetura da rede, controlando quais unidades deverão ser expandidas e quais eliminadas. O processo de maturação de afinidade será responsável pela adaptação, ou aprendizagem, da rede. Este modelo, assim como as redes neurais auto-organizadas, foi desenvolvido para aplicações genéricas a problemas de reconhecimento de padrões (particularmente binários) e clusterização.

5.4.2. Descrição e Análise da ABNET

Assim como no caso do algoritmo CLONALG para o problema de reconhecimento de padrões, será suposta a existência de uma população **Ag** de antígenos a ser reconhecida por um repertório **Ab** de anticorpos. A população antigênica corresponde a um grupo de padrões a serem reconhecidos e clusterizados pela rede. É suposto também que o repertório de anticorpos possui um único anticorpo no início do processo de aprendizagem, de forma que o repertório tem que ser construído enquanto submetido à população **Ag** (entretanto, a rede pode ser inicializada com qualquer tamanho não-nulo pré-definido). O repertório de anticorpos será modelado sob a forma de uma rede neural artificial Booleana, chamada ABNET. As principais características da ABNET são:

- Arquitetura construtiva baseada no princípio da seleção clonal;
- Poda da rede, representando a morte dos anticorpos pouco estimulados;

- Pesos Booleanos das conexões (espaço de formas binário de Hamming); e
- Aprendizagem competitiva e não-supervisionada simulando o processo de hipermutação somática direcionada.

O objetivo é construir um repertório de anticorpos capaz de apresentar máxima cobertura de um espaço conhecido de antígenos. Como a rede possui pesos binários, assumindo apenas os valores $\{0,1\}$, o espaço de formas binário de Hamming é o mais adequado para a derivação da ABNET.

Seja a seguinte notação:

- \mathbf{Ab}_i : i -ésimo anticorpo do repertório \mathbf{Ab} . Cada anticorpo corresponde ao vetor de pesos \mathbf{w}_i ligando a unidade de saída i a todas as entradas da rede;
- \mathbf{Ag}_j : j -ésimo antígeno da população \mathbf{Ag} ($\mathbf{Ag} \in S^{M \times L}$);
- τ_i : nível de concentração de cada anticorpo, ou seja, a quantidade de antígenos sendo reconhecidos por cada anticorpo \mathbf{Ab}_i ;
- f : vetor contendo as afinidades de todos os anticorpos da rede em relação ao antígeno \mathbf{Ag}_j apresentado; e
- \mathbf{v} : vetor que rotula qual anticorpo i reconhece cada antígeno j .

Sob o ponto de vista das redes neurais artificiais, os padrões binários a serem reconhecidos são geralmente denominados de *padrões de treinamento* (ou de *entrada*), e as unidades que compõem a rede são chamadas *neurônios* (Seção 4.2). Na engenharia imunológica, os padrões de entrada correspondem aos antígenos, e os neurônios, juntamente com seus pesos associados, são denominados de anticorpos. Um anticorpo i (\mathbf{Ab}_i) será representado pelo vetor de pesos \mathbf{w}_i conectando as entradas da rede à unidade de saída i . A única função das unidades de saída da rede é efetuar a soma ponderada dos sinais de entrada pelos respectivos vetores de pesos, o que corresponde a uma combinação linear entre as entradas e seus pesos associados. Uma característica particular da ABNET é a representação binária dos pesos das conexões no espaço de formas de Hamming. Sem perda de generalidade, a arquitetura inicial é composta por uma única unidade de saída, cujo vetor de pesos representa um único anticorpo hipotético na rede de anticorpos. Outro aspecto importante deste modelo, que não é considerado pelo algoritmo CLONALG, é a existência de um nível de concentração τ_i , que determina a concentração de antígenos para cada anticorpo i da rede. O nível de concentração τ_i , juntamente com a afinidade antigênica $f_{i,j}$, irá determinar qual anticorpo será selecionado para se reproduzir.

Como os antígenos a serem reconhecidos e os anticorpos da rede estão representados em um espaço de formas binário de Hamming, utilizou-se a Equação 3.3 para medir a afinidade entre estas moléculas. Por conveniência, esta equação será repetida a seguir:

$$f_{i,j} = f(\mathbf{Ab}_i, \mathbf{Ag}_j) = \sum_{m=1}^L \delta_m, \text{ onde } \delta_m = \begin{cases} 1 & \text{se } Ab_{i,m} \neq Ag_{j,m} \\ 0 & \text{outros casos} \end{cases} \quad (5.12)$$

Um anticorpo k do repertório \mathbf{Ab} , com maior afinidade ao antígeno \mathbf{Ag}_j dado, é aquele neurônio da rede cujo vetor de pesos \mathbf{w}_k apresenta a maior distância de Hamming (Equação (5.12)) a este antígeno (reconhecimento por complementaridade):

$$k = \arg \max_i \| \mathbf{A}g_j - \mathbf{A}b_i \|. \quad (5.13)$$

A construção deste modelo requer um último parâmetro a ser definido: v que representa um vetor para rotular os anticorpos de maior afinidade. Por exemplo, se $\mathbf{A}b_7$ é o anticorpo de maior afinidade a $\mathbf{A}g_9$, então $v_9 = 7$.

O laço principal do algoritmo de treinamento da ABNET pode ser resumido como a seguir (Figura 5.8):

1. Escolha aleatoriamente um antígeno (padrão de entrada) $\mathbf{A}g_j$, de acordo com sua densidade de probabilidade na população de antígenos;
2. Determine a afinidade f_j de todos os anticorpos da rede ao antígeno $\mathbf{A}g_j$, de acordo com a Equação (5.12);
3. Determine o anticorpo $\mathbf{A}b_k^j$ de maior afinidade a este antígeno $\mathbf{A}g_j$ de acordo com a Equação (5.13);
4. Submeta o vetor de pesos w_k , correspondente ao anticorpo $\mathbf{A}b_k^j$, ao processo de maturação gerando $\mathbf{A}b_k^{j*}$. Este processo é equivalente à atualização de pesos das redes neurais artificiais e será descrito separadamente na Seção 5.4.2.3;
5. Incremente o nível de concentração τ_i do anticorpo selecionado ($i = k$);
6. Atribua $v_j = k$; e
7. Identifique o anticorpo mais estimulado para ser clonado (Equação (5.14), Seção 5.4.2.1) e o menos estimulado para ser um candidato a ser eliminado.

O processo de maturação de afinidade na resposta imune adaptativa e a atualização de pesos das redes neurais artificiais servem ao mesmo propósito: aumentar a qualidade da resposta do sistema ao padrão a ser reconhecido. Nosso objetivo é construir uma rede de anticorpos com dimensão mínima que seja função da distribuição de probabilidade $P(\mathbf{A}g)$ da população de antígenos e de um limiar de afinidade ε a ser descrito posteriormente. Como $P(\mathbf{A}g)$ não é conhecida a priori, o nível de concentração de antígenos τ_j é utilizado para estimar seu valor.

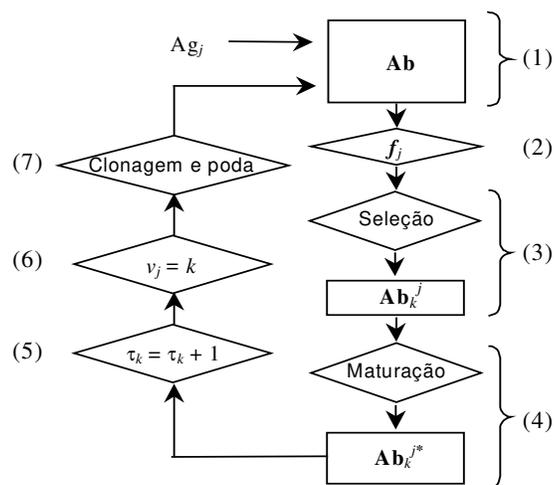


Figura 5.8. Principais passos do algoritmo de treinamento da ABNET.

5.4.2.1. Construção da Rede (Clonagem)

Este processo simula os mecanismos de seleção e reprodução dos anticorpos mais estimulados, de acordo com o princípio da seleção clonal. No nosso modelo, o anticorpo mais estimulado é aquele que possui a maior afinidade $f_{k,j}$ ao antígeno \mathbf{Ag}_j e cujo nível de concentração antigênica τ_k é o mais alto. Como o objetivo é gerar uma rede parcimoniosa, cada anticorpo selecionado irá sofrer uma única mitose, e os dois anticorpos gerados poderão fazer parte da memória imunológica após suas respectivas maturações.

Assim, o processo de construção (ou crescimento) da rede é baseado em dois parâmetros: o nível de concentração τ_i e o limiar de afinidade ϵ . Todos os anticorpos \mathbf{Ab}_i cujo nível de concentração for maior do que 1, $\tau_i > 1$, constituem candidatos em potencial a serem clonados, e o anticorpo s que estiver submetido à maior concentração antigênica será escolhido como único candidato. Este processo é descrito na Equação (5.14), e se nenhum anticorpo i satisfizer este critério, então a arquitetura da rede permanece inalterada:

$$s = \arg \max_{i \in O} \mathbf{Ab}_i, \text{ onde } O = \{i \mid \tau_i > 1\}. \quad (5.14)$$

Se a afinidade do anticorpo s em relação ao antígeno que possui maior afinidade a ele for maior do que o limiar de afinidade ϵ , então s é considerado o anticorpo mais estimulado e é clonado; senão a arquitetura da rede permanece inalterada. Quanto mais antígenos um anticorpo reconhece, mais estimulado este anticorpo estará. A rede promove o reforço dos anticorpos que são muito estimulados, favorecendo sua clonagem. O processo de crescimento é verificado a cada β iterações. A idéia de efetuar o crescimento a cada β iterações é estabelecer um *intervalo de amadurecimento* (aprendizagem) para que o repertório de anticorpos tenha tempo de aumentar sua afinidade aos antígenos através do processo de maturação que ocorre a cada iteração do algoritmo, como será descrito na próxima seção.

O vetor de pesos do novo clone, ou seja, o novo anticorpo, é o complemento exato do antígeno que possui a menor afinidade ao anticorpo clonado. Este aspecto não está em perfeito acordo com o princípio da seleção clonal, mas ele permite a geração de uma rede (ou repertório) de anticorpos que apresente uma maior afinidade a todos os antígenos dados, e o processo de crescimento descrito guia a ABNET direto a esta configuração, reduzindo a quantidade de iterações necessárias para convergência.

Para que clusters de anticorpos similares sejam gerados automaticamente, o novo clone deve ser inserido na vizinhança do anticorpo clonado. Esta idéia simula a preservação de vizinhança, considerada pelos mapas auto-organizáveis de Kohonen descritos na Seção 4.2.

Como ilustração da especificação dos parâmetros τ e ν e do funcionamento do mecanismo de crescimento da ABNET, considere a existência da seguinte população de antígenos a serem reconhecidos: $\mathbf{Ag}_1=[0,0,0]$, $\mathbf{Ag}_2=[1,1,1]$, $\mathbf{Ag}_3=[1,0,0]$. A configuração inicial da rede contém um único anticorpo representado pelo vetor de pesos $\mathbf{w}=[1,1,1]$. Como existe um único anticorpo na rede, os elementos ν_j , $j = 1, \dots, 3$, do vetor de rótulos ν , contêm o mesmo valor 1, ou seja, o anticorpo \mathbf{Ab}_1 reconhece todos os antígenos (Figura 5.9(a)). Isto ocorre porque, sendo o único neurônio, ele é aquele que possui a maior afinidade a todos os antígenos.

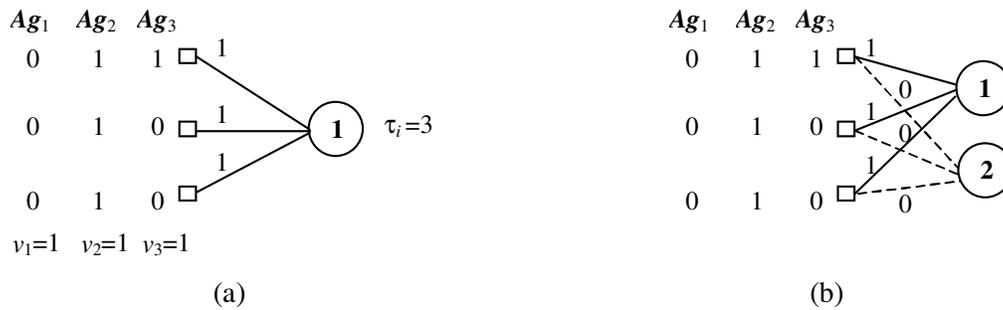


Figura 5.9. Processo de construção da rede. (a) Especificação dos parâmetros τ_i e v_j , onde τ_i define o nível de concentração de cada anticorpo e v_j rotula os anticorpos com maiores afinidade a cada um dos antígenos dados. $Ag_j, j = 1, \dots, 3$, são os antígenos (padrões de entrada ou treinamento); $w_1 = [1,1,1]$ representa Ab_1 . (b) Clonagem e definição do vetor de pesos correspondente ao novo anticorpo, $w_2 = [0,0,0]$ representa Ab_2 .

O antígeno com maior afinidade (maior distância de Hamming) em relação a este anticorpo é Ag_1 , e o antígeno de menor afinidade é Ag_2 (Equação (5.12)). Este anticorpo Ab_1 será selecionado para se reproduzir e seu clone terá pesos complementares a Ag_2 , $w_2 = [0,0,0]$. A Figura 5.9(b) ilustra a geração de um novo anticorpo e a definição de seu vetor de pesos associados. Este procedimento garante uma maior e mais rápida cobertura do espaço de antígenos.

5.4.2.2. Poda da Rede (Morte Celular Programada)

A política de poda, simulando a apoptose (morte programada dos anticorpos pouco estimulados), dá-se como a seguir: se um anticorpo p possui seu nível de concentração igual a zero ($\tau_p = 0$) por uma quantidade de iterações maior do que um limiar pré-definido, então este anticorpo pode ser removido da rede. Como proposto por de Castro & Von Zuben (1999b) em um outro contexto, após executar uma poda em uma rede competitiva, o valor da taxa de aprendizagem, correspondente, neste caso, à taxa de hipermutação a ser descrita na próxima seção, deve ser reinicializada. No treinamento da ABNET, a reinicialização da taxa de hipermutação foi empregada durante os processos de poda e clonagem da rede.

Para ilustrar como ocorre a poda, considere o mesmo exemplo anterior: uma população de antígenos composta por $Ag_1 = [0,0,0]$, $Ag_2 = [1,1,1]$ e $Ag_3 = [1,0,0]$. Suponha que uma ABNET com três anticorpos ($w_1 = [1,1,1]$, $w_2 = [0,0,0]$, $w_3 = [1,0,0]$) foi construída de acordo com a Figura 5.10(a). Neste caso, os vetores $v = [1,2,1]$ e $\tau = [2,1,0]$, indicam que o anticorpo Ab_3 , não estimulado por nenhum antígeno, será removido (podado) da rede após uma quantidade de gerações celulares pré-especificada. Após remover Ab_3 , a rede resultante está apresentada na Figura 5.10(b). Esta política de poda objetiva manter a especificidade dos anticorpos, devido ao fato de que somente aqueles anticorpos que não reconhecem nenhum antígeno serão candidatos a serem removidos.

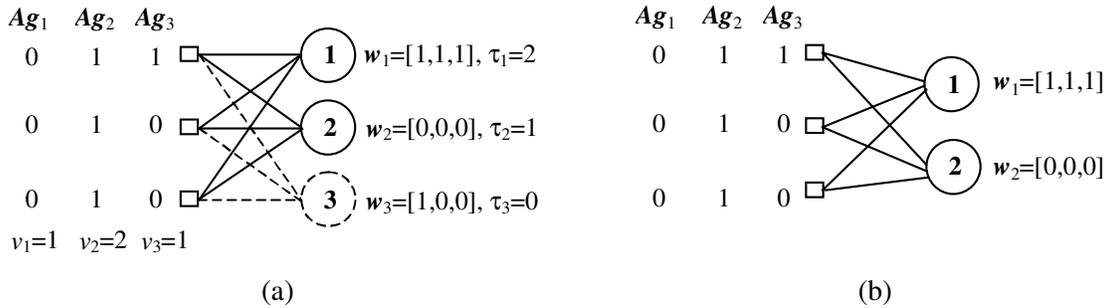


Figura 5.10. Poda de anticorpos não estimulados. (a) O anticorpo Ab_3 , não está sendo estimulado por nenhum antígeno ($\tau_3 = 0$). $Ag_j, j = 1, \dots, 3$, são os antígenos (padrões de entrada); $w_1 = [1, 1, 1]$, $w_2 = [0, 0, 0]$ e $w_3 = [0, 0, 0]$ representam Ab_1, Ab_2 e Ab_3 , respectivamente, e, $\varepsilon = 1$. (b) Rede resultante com apenas dois anticorpos, após um período de tempo maior do que um limiar pré-definido.

A Figura 5.11 ilustra um processo hipotético de construção da ABNET. As circunferências sólidas representam os anticorpos restantes após crescimento e poda, as circunferências em branco representam os anticorpos clonados e as circunferências sombreadas correspondem aos anticorpos que foram removidos durante o processo de aprendizagem/crescimento da rede. As unidades de entrada e suas conexões são representadas apenas na rede inicial (a) e na final (c).

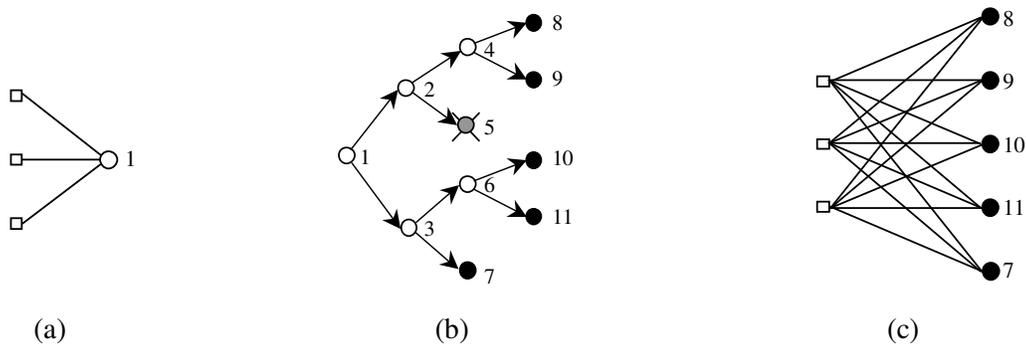


Figura 5.11. Construção da rede. Quadrados: unidades de entrada; circunferências sólidas: anticorpos restantes; circunferências em branco: anticorpos clonados; circunferências sombreadas: anticorpos eliminados. (a) ABNET inicial. (b) Processos de crescimento e poda da rede (as conexões são omitidas e as setas indicam os processos de clonagem). (c) Arquitetura resultante da rede com os anticorpos restantes rotulados.

5.4.2.3. Atualização de Pesos (Maturação de Afinidade)

O processo de atualização de pesos da ABNET é semelhante a uma mutação de múltiplos pontos com taxas variáveis (Seções 3.4.3.4 e 4.4.3). A taxa de hipermutação α irá determinar quantas posições da cadeia de atributos, que representa os anticorpos, serão mudadas.

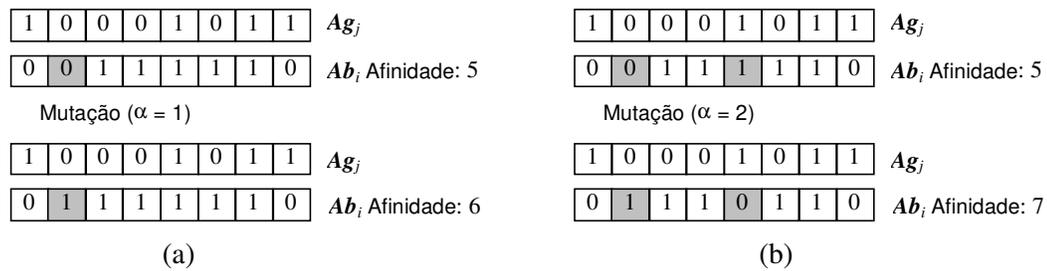


Figura 5.12. Processo de atualização de pesos para antígenos de comprimento $L = 8$, onde as posições de mutação são escolhidas aleatoriamente dentre as complementares (a) $\alpha = 1$. (b) $\alpha = 2$.

Como o espaço de formas utilizado é o binário de Hamming, as posições do anticorpo que não forem complementares ao antígeno reconhecido serão candidatas a sofrerem mutação, resultando em um processo direcionado de busca no espaço de formas. O processo de atualização de pesos, correspondente à maturação de afinidade, é direcionado para que os anticorpos se tornem o complemento do conjunto de antígenos mais rapidamente (Figura 5.12). Quanto maior a distância de Hamming (complementaridade) entre um antígeno e um anticorpo, menor a taxa de mutação e vice-versa. No caso do espaço de formas binário de Hamming, a taxa de mutação $\alpha \in Z_{\{0\}}^+$, onde $Z_{\{0\}}^+$ é o conjunto de inteiros não-negativos, possui um esquema de decrescimento em unidades, e após uma quantidade pré-definida de iterações, o valor de α é reduzido de 1, até que ele atinja $\alpha = 0$, representando a ausência de mutação. Isso geralmente ocorre quando a afinidade $Ag-Ab$ é máxima.

5.4.3. Pseudocódigo

Considere a implementação de uma função computacional para simular a ABNET. Esta função pode ter como argumentos os seguintes parâmetros:

Entrada:

- Comprimento L das cadeias de atributos;
- População \mathbf{Ag} composta por M antígenos de comprimento L ;
- Quantidade gen de gerações a serem executadas (critério de parada);
- Limiar de afinidade ε ;
- Taxa inicial α_0 de hipermutação; e
- A cada β iterações a taxa α de hipermutação é decrescida de uma unidade e um anticorpo é escolhido para ser um candidato à clonagem. Se um anticorpo p não é estimulado por nenhum antígeno durante β iterações, então ele é eliminado da rede.

Saída:

- Matriz \mathbf{Ab} de anticorpos de memória; e
- Matriz \mathbf{F} contendo a afinidade de cada anticorpo Ab_i , $i = 1, \dots, N$ (o valor de N é automaticamente definido ao longo da execução do algoritmo), em relação a cada antígeno Ag_j , $j = 1, \dots, M$.

Dados os parâmetros de entrada e saída, é possível descrever um pseudocódigo para a implementação da ABNET como a seguir:

```
function [Ab, F] = abnet (Ag, L, gen,  $\epsilon$ ,  $\alpha_0$ ,  $\beta$ );
Ab := gera(1, L); % Gera Ab inicial
% A cada geração, faça:
for t = 1 to gen,
    for j = 1 to M, % Passo 1
        f(j, :) := afinidade(Ab, Ag(j, :)); % Passo 2
        Abk := select(Ab, f, 1); % Passo 3
        Abk* := dmut(Abk, Ag(j, :),  $\alpha$ ); % Passo 4
         $\tau_k$  :=  $\tau_k$  + 1; % Passo 5
         $v_j$  := k; % Passo 6
    end;
    if (t mod  $\beta$ ) = 0,
        [Ab,  $\alpha$ ] := split(Ab,  $v$ ,  $\tau$ ,  $\epsilon$ ,  $\alpha_0$ ); % Passo 7
         $\alpha$  :=  $\alpha$  - 1;
    end;
    [Ab,  $\alpha$ ] := poda(Ab,  $\tau$ , t,  $\beta$ ,  $\alpha_0$ ); % Passo 7
    F := afinidade(Ab, Ag);
end;
```

5.4.4. Caracterização e Comparação

Nesta seção, foi proposto um algoritmo construtivo para a geração de uma rede neural Booleana baseado nas propriedades do princípio de seleção clonal e maturação de afinidade do sistema imunológico. O repertório de anticorpos é modelado utilizando-se uma abordagem conexionista (rede neural artificial), com uma rede de anticorpos sendo gerada, onde os vetores de pesos da rede representam os anticorpos. Várias características da resposta imune adaptativa foram consideradas, cada uma com uma finalidade distinta:

- Expansão clonal dos anticorpos mais estimulados: crescimento da rede;
- Morte programada dos anticorpos pouco estimulados: poda da rede; e
- Maturação de afinidade: atualização dos pesos (aprendizagem).

O emprego de um processo de restauração e controle da taxa de hipermutação, juntamente com a definição do intervalo de amadurecimento, devem simular os mecanismos propostos por Kepler & Perelson (1993a,b), onde intervalos de expansão clonal e ausência de mutação são intercalados com a ocorrência de altas taxas de mutação e nenhuma expansão. Estes procedimentos devem conduzir a uma estratégia que visa a otimização do controle dos mecanismos de expansão clonal e maturação de afinidade (Kepler & Perelson, 1993a).

As arquiteturas de rede modeladas não possuem um tamanho pré-definido. Os antígenos a serem reconhecidos serão responsáveis pela seleção e expansão dos anticorpos disponíveis. Durante a evolução da rede, ela desenvolve uma série de características auto-regulatórias como a determinação automática de um tamanho específico. Uma vez que a ABNET atinge um equilíbrio em relação à quantidade de unidades na rede, os anticorpos clonados devem competir com os já estabelecidos para que eles possam entrar na rede, ou seja, fazer parte do repertório.

Sob o ponto de vista de imunologia, a tarefa desempenhada pela ABNET é equivalente à geração de um repertório de anticorpos com tamanho mínimo para reconhecer qualquer antígeno de uma população específica e conhecida. Sob o ponto de vista de sistemas inteligentes, o algoritmo apresenta uma estratégia competitiva (não supervisionada) para a definição de arquitetura e treinamento de uma rede neural artificial Booleana. Como no caso das RNAs, a utilização de um limiar de afinidade ϵ permite o estudo da capacidade de generalização, ou resposta reativa cruzada, da ABNET.

O mecanismo de seleção utilizado pela ABNET pode ser visto como o dual do esquema de competição aplicado a redes neurais competitivas, onde, no primeiro caso, o neurônio (anticorpo) escolhido é o que possui a maior afinidade (maior distância de Hamming) ao antígeno e, no segundo caso, o neurônio escolhido é o que possui menor distância ao padrão de entrada. Além disso, a ABNET é capaz de controlar a dimensão de sua arquitetura como algumas classes de redes neurais auto-organizadas (Fritzke, 1994; Cho, 1997; de Castro & Von Zuben, 1999b), e também possui uma taxa de aprendizagem variável, como no caso dos mapas auto-organizáveis de Kohonen (Seção 4.2.2.2).

A rede de Hamming (Lippman, 1987) se assemelha à ABNET no sentido de que ambas são competitivas e calculam a distância de Hamming entre um conjunto de pesos e os padrões de entrada, mas diferenciam-se completamente quanto ao algoritmo de treinamento. Na rede de Hamming, a arquitetura é fixa e contém duas camadas, os pesos são especificados (e não treinados) de acordo com o conjunto de treinamento e as unidades da camada de saída possuem um bias. Por outro lado, a ABNET é construída durante a aprendizagem, possui uma única camada e não apresenta bias, resultando em uma arquitetura construtiva e parcimoniosa. Ambas as redes são projetadas para tratar conjuntos de dados binários de forma eficiente e são interessantes para a implementação em hardware, principalmente devido às suas operações intrinsecamente binárias. Exemplos de implementação em hardware de redes de Hamming que talvez possam servir de base para o caso da ABNET podem ser encontrados nos trabalhos de Robinson *et al.* (1992), Çilingiroglu (1993) e Schimid *et al.* (1998).

A classe de redes neurais em que os vetores de entrada e saída são strings binárias (0 ou 1) é geralmente denominada de Booleana. Neste grupo de redes neurais, a representação obtida pelo algoritmo de treinamento torna-se um circuito lógico implementável por *portas lógicas* simulando unidades intermediárias e de saída. Exemplos de algoritmos derivados de princípios de álgebra Booleana podem ser encontrados em Biswas & Kumar (1990) e Gray & Michel (1992). Neste caso, o algoritmo de aprendizagem é completamente diferente do empregado pela ABNET, e utiliza operações de álgebra Booleana ao invés da inspiração biológica.

5.4.5. Complexidade Computacional

O algoritmo de treinamento da ABNET, possui três etapas principais de processamento: 1) a determinação da afinidade (ou fitness) dos anticorpos da população (Passo 2); 2) a seleção do melhor indivíduo (Passo 3); e 3) a mutação do anticorpo selecionado (Passo 4).

Tabela 5.3. Complexidade computacional por geração para o treinamento da ABNET, onde N é o tamanho atual da população de anticorpos (dimensão da ABNET), L é o comprimento das cadeias de atributos que representam as moléculas, e M é a quantidade de antígenos a serem reconhecidos.

	<i>Tempo de Processamento</i>	<i>Memória</i>
ABNET	$O(M(L + \log N))$	$\propto N(L + M)$

O custo computacional para a determinação da afinidade da população depende do problema abordado e nenhuma medida genérica pode ser apresentada. Por outro lado, é possível calcular estimativas de complexidade computacional para o processo de seleção e mutação, como descrito na Seção 5.1.2.

Uma particularidade do algoritmo de treinamento da ABNET é o seu processo construtivo. A quantidade de anticorpos do repertório varia dinamicamente: o processo é iniciado com apenas uma unidade na rede e vai aumentando até que a rede atinja uma dimensão de equilíbrio. Portanto, o custo computacional do algoritmo também varia dinamicamente até que a arquitetura da rede atinja este estado de equilíbrio, começando com um baixo custo computacional e aumentando à medida que novas unidades vão sendo inseridas. Na situação final da rede, em que uma quantidade de anticorpos foi inserida e ela atingiu uma arquitetura com dimensão estável, considere que a rede possui N anticorpos (unidades de saída). Assim, o custo computacional envolvido no único processo de seleção, Passo 3, é $O(\log N)$, pois deseja-se encontrar apenas o anticorpo de maior afinidade. Durante a mutação direcionada deste anticorpo, é necessário um tempo de processamento da ordem de $O(L)$, onde L é o comprimento da cadeia de atributos utilizada para representar os anticorpos. Estes passos são executados para os M antígenos apresentados. Somando-se a complexidade de cada etapa, é possível determinar a complexidade computacional do algoritmo, como apresentado na Tabela 5.3.

A cada iteração, a quantidade de memória requerida para executar o algoritmo é proporcional ao número atual N de anticorpos na rede. A matriz \mathbf{A} de anticorpos tem dimensão $S^{N \times L}$, e a matriz \mathbf{F} de afinidades é de dimensão $S^{N \times M}$, assim a memória requerida para executar este algoritmo é proporcional a $N(L + M)$, como apresentado na Tabela 5.3.

5.5. aiNet: Um Modelo de Rede Imunológica Artificial

Nesta seção, é apresentado um modelo de rede imunológica artificial, chamada aiNet (*Artificial Immune NETWORK*), inspirado na teoria da rede imunológica proposta por Jerne (1974a) e discutida na Seção 2.10. Este algoritmo foi originalmente proposto por de Castro & Von Zuben (2000c, 2001g) e a rede desenvolvida apresenta características importantes, como a capacidade de descrever a estrutura interna dos antígenos (dados de treinamento), seu perfil de distribuição de probabilidade e relações de vizinhança (clusters). A aiNet incorpora o algoritmo de seleção clonal, CLONALG (Seção 5.3), como parte do processo de treinamento da rede.

5.5.1. Motivação e Escopo de Aplicações

Na Seção 2.10, foi argumentado que a teoria da rede imunológica é particularmente interessante para o desenvolvimento de ferramentas computacionais, pois ela fornece uma medida aproximada de propriedades emergentes como aprendizagem e memória, tolerância ao próprio, dimensão e diversidade de populações celulares. Características da rede imunológica como estrutura, dinâmica e metadinâmica podem ser diretamente empregadas na construção de um modelo de rede imunológica artificial, denominada aiNet, para resolver problemas de engenharia. No nosso modelo de rede imunológica, o princípio da seleção clonal irá controlar a quantidade e forma dos anticorpos da rede (sua dinâmica e metadinâmica), enquanto técnicas de clusterização hierárquica e teoria de grafos serão utilizadas para definir e interpretar a estrutura final da aiNet. Assim, como no caso da ABNET, o algoritmo de treinamento é genérico, porém a rede resultante irá depender do problema tratado, ou seja, o conjunto de antígenos a serem reconhecidos guiará a busca pela estrutura final da rede e forma dos anticorpos. As principais aplicações da aiNet incluem aprendizagem de máquina, reconhecimento de padrões, compressão e clusterização de dados.

5.5.2. Descrição

Dado um conjunto \mathbf{Ag} de antígenos, onde cada antígeno (padrão ou amostra de treinamento) \mathbf{Ag}_j , $j = 1, \dots, M$, é descrito por L variáveis (atributos ou características) em um espaço de formas Euclidiano, uma rede imunológica artificial deve ser construída para responder às seguintes questões: (1) Existe algum grupo ou subgrupo intrínseco aos antígenos? (2) Se existir(em), quantos são? (3) Quais são as propriedades relevantes destes grupos de antígenos? (4) Como podemos gerar regras de decisão para classificar novos antígenos? (5) Qual é a conformação dos grupos no espaço de formas?

O modelo de rede imunológica artificial, aiNet, pode ser formalmente definido como abaixo:

Definição 5.1: A rede imunológica artificial, chamada aiNet, é um grafo com conexões ponderadas, não necessariamente totalmente interconectado, composto por um conjunto de nós, denominados *anticorpos*, e conjuntos de pares de nós chamados *conexões*, com um valor característico associado, chamado de *peso da conexão* ou simplesmente *peso*.

Como a estrutura da rede poderá ser determinada por alguns métodos de clusterização, iremos definir primeiramente o que significa um cluster e clusterização de dados:

Definição 5.2: Clusterização é um processo de agrupamento de dados que apresentam um elenco de propriedades similares e distintas daquelas presentes em outros dados não pertencentes ao mesmo agrupamento.

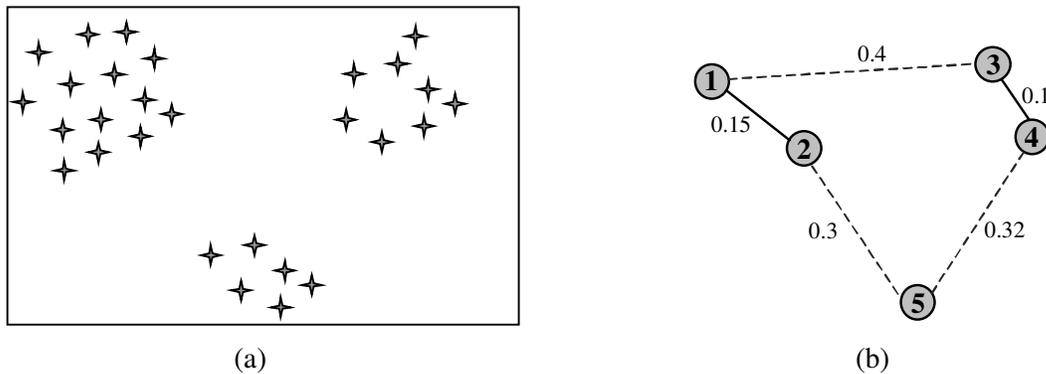


Figura 5.13. Ilustração da aiNet. (a) Conjunto de antígenos a serem reconhecidos e clusterizados, contendo três regiões de alta concentração antigênica. (b) aiNet com os anticorpos rotulados e os respectivos pesos das conexões. As linhas tracejadas indicam conexões que deverão ser detectadas e removidas com o objetivo de gerar subgrafos desconexos, cada um caracterizando um cluster diferente da rede.

Os clusters da rede servirão como *imagens internas* que mapeiam clusters existentes no conjunto de antígenos em clusters existentes nos anticorpos que formam a aiNet. Como ilustração, suponha que existe um conjunto de antígenos composto por três regiões de alta concentração antigênica (densidade de dados), de acordo com a Figura 5.13(a). Uma arquitetura hipotética de rede gerada pelo algoritmo de treinamento da aiNet é apresentada na Figura 5.13(b). O número dentro das unidades indicam seus rótulos, os números ao lado das conexões representam seus pesos associados, e as linhas tracejadas sugerem conexões a serem eliminadas objetivando detectar clusters e definir a arquitetura final da rede. Note a presença de três clusters distintos de anticorpos, cada qual com um número diferente de moléculas, conexões e pesos associados. Estes clusters constituem imagens internas dos clusters correspondentes do conjunto de antígenos. Note também que a quantidade de anticorpos na rede é bem menor do que a quantidade de antígenos a serem reconhecidos, indicando uma redução na cardinalidade da informação a ser armazenada.

O algoritmo de treinamento da aiNet possui duas etapas distintas. Na primeira delas, é feita a interação dos anticorpos da rede e um antígeno a ser reconhecido, e na segunda etapa é quantificada a interação apenas dos anticorpos da rede. As afinidades Ag-Ab e Ab-Ab são avaliadas utilizando-se medidas de proximidade (ou similaridade). O objetivo é utilizar uma métrica de distância para gerar um conjunto de anticorpos que constituam imagens internas dos antígenos a serem reconhecidos, e avaliar o grau de similaridade entre os anticorpos da rede, de forma a permitir um controle da cardinalidade do repertório. Sendo assim, a afinidade Ag-Ab é inversamente proporcional à distância entre eles: quanto menor a distância, maior a afinidade e vice-versa. Enquanto uma alta afinidade Ag-Ab resulta numa resposta imune adaptativa seguindo o algoritmo CLONALG (princípio da seleção clonal), uma alta afinidade entre anticorpos promove uma supressão na rede (eliminação de anticorpos).

É importante salientar que, na estrutura formal para a engenharia imunológica proposta no Capítulo 3, assim como nos sistemas imunológicos biológicos, o reconhecimento

antigênico ocorre por complementaridade de formas, e não por similaridade. Entretanto, em diversas aplicações de sistemas imunológicos artificiais (por exemplo, Oprea, 1999; Hart & Ross, 1999; Hajela & Yoo, 1999), e para os objetivos propostos para a aiNet, a geração de um conjunto de anticorpos com características similares, e não complementares, aos antígenos é mais apropriada, pois permite uma interpretação direta da rede resultante. A mesma abordagem foi empregada para o algoritmo SAND, descrito na Seção 5.2.

Como proposto na teoria original da rede imunológica (Seção 2.10) e nos modelos contínuos apresentados (Seção 3.4.3.5), os anticorpos presentes na rede irão competir pelo reconhecimento antigênico e aqueles que forem bem sucedidos promoverão a ativação da rede e proliferação celular seguindo o princípio da seleção clonal, enquanto aqueles que falharem no reconhecimento serão eliminados. Além disso, quando um anticorpo reconhecer um antígeno próprio, ou seja, outro anticorpo, isso resultará em uma supressão da rede. Na aiNet, a supressão é efetuada eliminando-se um entre dois anticorpos que se reconhecem (auto-reativos), dado um limiar de supressão σ_s . A interação de cada par Ag-Ab é avaliada no espaço de formas Euclidiano através de uma medida de afinidade (similaridade) f , que reflete a probabilidade de se iniciar uma resposta clonal. De forma semelhante, uma afinidade $s_{i,j}$ será especificada a cada par Ab-Ab, refletindo seu grau de interação.

Supondo que na iteração t a aiNet possui N anticorpos, a seguinte notação será empregada:

- **Ab**: repertório de anticorpos disponíveis ($\mathbf{Ab} \in S^{N \times L}$, $\mathbf{Ab} = \mathbf{Ab}_{\{d\}} \cup \mathbf{Ab}_{\{m\}}$);
- $\mathbf{Ab}_{\{m\}}$: repertório total de anticorpos de memória ($\mathbf{Ab}_{\{m\}} \in S^{m \times L}$, $m \leq N$);
- **Ag**: população de M antígenos a serem reconhecidos ($\mathbf{Ag} \in S^{M \times L}$);
- f_j : vetor contendo a afinidade de todos os anticorpos em relação ao antígeno \mathbf{Ag}_j . A afinidade é inversamente proporcional à distância Ag-Ab;
- **S**: matriz de similaridade entre os anticorpos de memória com elementos $s_{i,j}$;
- **C**: população de N_c clones gerada a partir de **Ab** ($\mathbf{C} \in S^{N_c \times L}$);
- **C***: população **C** após o processo de maturação de afinidade;
- d_j : vetor contendo a afinidade de todos os clones de **C*** em relação ao antígeno \mathbf{Ag}_j ;
- ζ : percentual de anticorpos maduros a serem selecionados;
- **M**: clones de memória para o antígeno \mathbf{Ag}_j . Restantes do processo de supressão clonal e seleção dos ζ anticorpos maduros;
- $\mathbf{Ab}_{\{d\}}$: d novos anticorpos a serem inseridos na população **Ab** ($\mathbf{Ab}_{\{d\}} \in S^{d \times L}$);
- σ_d : limiar de morte natural; e
- σ_s : limiar de supressão.

O algoritmo de treinamento da aiNet tem por objetivo construir um conjunto de memórias que melhor represente a estrutura do conjunto de antígenos a serem reconhecidos. Quanto mais específicos forem os anticorpos da rede, menos parcimoniosa será a arquitetura (baixa compressão de dados), e quanto mais generalistas os anticorpos, mais parcimoniosa a arquitetura da rede em relação à quantidade de anticorpos (alta compressão de dados). As conexões entre os componentes da rede serão definidas, utilizando métodos estatísticos ou de teoria de grafos, após a aplicação do algoritmo de treinamento descrito a seguir:

A cada iteração, faça:

1.1. Para cada antígeno \mathbf{Ag}_j , $j = 1, \dots, M$, ($\mathbf{Ag}_j \in \mathbf{Ag}$), faça:

1.1.1. Determine sua afinidade $f_{i,j}$, $i = 1, \dots, N$, em relação a todo o repertório \mathbf{Ab} .

$$f_{i,j} = 1/D_{i,j}, \quad i = 1, \dots, N:$$

$$D_{i,j} = \|\mathbf{Ab}_i - \mathbf{Ag}_j\|, \quad i = 1, \dots, N \quad (5.15)$$

1.1.2. Do conjunto \mathbf{Ab} , um subconjunto $\mathbf{Ab}_{\{n\}}$ composto pelos n anticorpos com maiores afinidades é selecionado;

1.1.3. Os n indivíduos selecionados irão se proliferar (clonagem) proporcionalmente à sua afinidade $f_{i,j}$, gerando uma população \mathbf{C} de clones: quanto maior a afinidade, maior o tamanho do clone para cada um dos n anticorpos selecionados (Equação (5.23));

1.1.4. Em seguida, a população \mathbf{C} de clones é submetida ao processo de maturação de afinidade gerando uma nova população \mathbf{C}^* , onde cada anticorpo k de \mathbf{C}^* irá sofrer uma mutação com taxa α_k inversamente proporcional à afinidade $f_{i,j}$ de seu anticorpo pai: quanto maior a afinidade, menor a taxa de mutação:

$$\mathbf{C}_k^* = \mathbf{C}_k + \alpha_k (\mathbf{Ag}_j - \mathbf{C}_k); \quad \alpha_k \propto 1/f_{k,j}; \quad k = 1, \dots, N_c; \quad N \leq N_c. \quad (5.16)$$

1.1.5. Determine a afinidade $d_{k,j} = 1/D_{k,j}$ entre os elementos de \mathbf{C}^* e o antígeno \mathbf{Ag}_j :

$$D_{k,j} = \|\mathbf{C}_k^* - \mathbf{Ag}_j\|, \quad k = 1, \dots, N_c. \quad (5.17)$$

1.1.6. Da população \mathbf{C}^* , re-seleccione $\zeta\%$ dos anticorpos que apresentam maior $d_{k,j}$ e os coloque em uma matriz \mathbf{M} de clones de memória (memória clonal);

1.1.7. *Morte programada*: Elimine todos os clones de memória cuja afinidade $d_{k,j}$ ao antígeno \mathbf{Ag}_j é superior a um limiar σ_d :

$$\text{Se } D_{k,j} > \sigma_d, \text{ então exclua o anticorpo } k \text{ do conjunto } \mathbf{M}. \quad (5.18)$$

1.1.8. Determine a afinidade $s_{i,j}$ entre os clones de memória:

$$s_{i,j} = \|\mathbf{M}_i - \mathbf{M}_j\|, \quad \forall i, j. \quad (5.19)$$

1.1.9. *Supressão clonal*: elimine aqueles clones de memória cuja afinidade $s_{i,j}$ for inferior ao limiar de supressão σ_s :

$$\text{Se } s_{i,j} < \sigma_s, \text{ então exclua o anticorpo } i \text{ do conjunto } \mathbf{M} \quad (5.20)$$

1.1.10. Concatene a matriz total de anticorpos de memória com a matriz de clones de memória \mathbf{M} para o antígeno \mathbf{Ag}_j : $\mathbf{Ab}_{\{m\}} \leftarrow [\mathbf{Ab}_{\{m\}}; \mathbf{M}]$;

1.2. Calcule a afinidade entre todos os anticorpos de memória:

$$s_{i,j} = \|\mathbf{Ab}_{\{m\}}^i - \mathbf{Ab}_{\{m\}}^j\|, \quad \forall i, j. \quad (5.21)$$

1.3. *Supressão da rede*: elimine os anticorpos de memória tais que $s_{i,j} < \sigma_s$:

$$\text{Se } s_{i,j} < \sigma_s, \text{ então exclua } \mathbf{Ab}_{\{m\}}^i \text{ da rede} \quad (5.22)$$

1.4. A matriz total de anticorpos corresponde à matriz de memória $\mathbf{Ab}_{\{m\}}$ concatenada com d novos anticorpos: $\mathbf{Ab} \leftarrow [\mathbf{Ab}_{\{m\}}; \mathbf{Ab}_{\{d\}}]$

2. Teste o critério de parada.

O limiar de supressão σ_s controla o nível de especificidade dos anticorpos, a acurácia da clusterização e a plasticidade da rede. É sugerido um valor inicial pequeno para este limiar (p.ex., $\sigma_s \leq 10^{-3}$) seguido por um ajuste fino de acordo com o desempenho da rede. A Seção 6.5.3 apresenta a análise de sensibilidade da aiNet em relação aos parâmetros de treinamento definidos pelo usuário, incluindo σ_s .

Para a determinação do tamanho N_c do conjunto de clones gerado para cada um dos j , $j = 1, \dots, M$, antígenos, poderíamos ter empregado a Equação (5.10), como proposto no caso do algoritmo CLONALG para o problema de reconhecimento de padrões. No entanto, optamos por uma outra forma de determinação de N_c , como descrito pela Equação (5.23), com a vantagem de eliminar o parâmetro β (embora ele possa ser facilmente introduzido na equação) definido pelo usuário e a desvantagem de reduzir o controle do usuário sobre N_c .

$$N_c = \sum_{i=1}^n \text{round}(N - D_{i,j} \cdot N), \quad (5.23)$$

onde N é a quantidade total de anticorpos do repertório \mathbf{Ab} , $\text{round}(\cdot)$ é o operador que arredonda o valor entre parênteses para o inteiro mais próximo e $D_{i,j}$ é a distância entre o anticorpo selecionado i e o antígeno \mathbf{Ag}_j , dada pela Equação (5.15). É fácil verificar por esta equação que quanto maior a afinidade $\mathbf{Ag}_j - \mathbf{Ab}_i$, ou seja, quanto menor $D_{i,j}$, maior a quantidade de clones gerada para o anticorpo \mathbf{Ab}_i .

Devido à quantidade de operações efetuadas por este algoritmo, não será apresentado um diagrama de blocos específico para ele, mas note que do Passo 1.1 ao Passo 1.1.7 o algoritmo da aiNet corresponde ao algoritmo CLONALG para o problema de reconhecimento de padrões (Figura 5.5). Além da seleção clonal via o algoritmo CLONALG, os Passos de 1.1.8 a 1.1.10 e 1.2 a 1.3 simulam parte da atividade da rede imunológica. A afinidade entre os anticorpos e o antígeno j apresentado é aumentada de acordo com a Equação (5.16), onde o parâmetro α_k corresponde à taxa de hipermutação individual de cada anticorpo pertencente ao conjunto \mathbf{C} . Esta equação executa uma busca direcionada, onde a distância entre o antígeno \mathbf{Ag}_j e os anticorpos é diminuída proporcionalmente a α_k . Desta forma, o processo de busca otimiza localmente os anticorpos \mathbf{C}_k , $k = 1, \dots, N_c$, com o objetivo de aumentar suas capacidades de reconhecimento antigênico ao longo das iterações. Este processo é característico de uma busca em que cada elemento é otimizado localmente, para que se alcance um melhor comportamento global do sistema.

De acordo com este algoritmo, uma resposta imune adaptativa (expansão clonal) ocorre após a apresentação de cada antígeno à rede. Existem também dois passos supressivos no algoritmo (Passos 1.1.9 e 1.3) denominados, respectivamente, de *supressão clonal* e *supressão da rede*. A supressão clonal é responsável por eliminar anticorpos com alta afinidade entre si dentro de cada clone de memória (anticorpos de \mathbf{M}), enquanto a supressão da rede elimina anticorpos com alta afinidade considerando toda a rede ($\mathbf{Ab}_{\{m\}}$).

Após a fase de aprendizagem, os anticorpos devem representar imagens internas dos antígenos (ou grupos de antígenos) apresentados. Este algoritmo segue a estrutura genérica de rede imunológica dada pela Equação 3.17, em que a taxa de variação populacional é

diretamente proporcional à entrada de novos anticorpos (Passo 1.4), menos a morte dos anticorpos pouco estimulados (Passo 1.1.7) e auto-reativos (Passos 1.1.9 e 1.3), mais a reprodução dos anticorpos estimulados (Passo 1.1.3).

Para avaliar a convergência da aiNet, é possível propor diferentes critérios:

- Interrompa o processo iterativo após uma quantidade pré-definida de iterações;
- Interrompa o processo iterativo quando a rede atingir uma dimensão pré-definida;
- Avalie a afinidade média (correspondente ao erro médio) entre a população de anticorpos e os antígenos apresentados. Este critério é bastante preciso quando o número de anticorpos da rede é aproximadamente igual ao número de antígenos a serem reconhecidos; e
- A aiNet é dita ter convergido se o erro médio aumenta durante k iterações consecutivas.

5.5.3. Pseudocódigo

Considere a implementação de uma função computacional para treinar a aiNet de acordo com o algoritmo proposto. Esta função pode ter como argumentos os seguintes parâmetros:

Entrada:

- Comprimento L das cadeias de atributos;
- População \mathbf{Ag} composta por M antígenos;
- Quantidade gen de gerações a serem executadas (critério de parada);
- Número n de anticorpos a serem selecionados para clonagem;
- Percentual $\zeta\%$ de anticorpos maduros a serem selecionados;
- Limiar σ_d de morte natural;
- Limiar σ_s de supressão; e
- Quantidade d de novos anticorpos a serem inseridos na rede.

Saída:

- Matriz $\mathbf{Ab}_{\{m\}}$ de anticorpos de memória; e
- Matrix \mathbf{S} de similaridade entre os anticorpos de memória da rede, composta por elementos $s_{i,j}$.

Dados os parâmetros de entrada e saída, é possível descrever um pseudocódigo para a implementação da aiNet como a seguir:

```
function [ $\mathbf{Ab}_{\{m\}}$ ,  $\mathbf{S}$ ] = aiNet ( $\mathbf{Ag}$ ,  $L$ ,  $gen$ ,  $n$ ,  $\zeta$ ,  $\sigma_d$ ,  $\sigma_s$ ,  $d$ );
 $\mathbf{Ab}$  := gera ( $N_0$ ,  $L$ ); % Gera  $\mathbf{Ab}$  inicial ( $N_0$  qualquer)
% A cada iteração, faça:
for t = 1 to  $gen$ , % Passo 1
    for j = 1 to  $M$ , % Passo 1.1
         $\mathbf{f}(j, :)$  := afinidade ( $\mathbf{Ab}$ ,  $\mathbf{Ag}(j, :)$ ); % Passo 1.1.1
         $\mathbf{Ab}_{\{n\}}(j, :)$  := select ( $\mathbf{Ab}$ ,  $\mathbf{f}(j, :)$ ,  $n$ ); % Passo 1.1.2
         $\mathbf{C}$  := clona ( $\mathbf{Ab}_{\{n\}}$ , 1,  $\mathbf{f}(j, :)$ ); % Passo 1.1.3
         $\mathbf{C}^*$  := dmut ( $\mathbf{C}$ ,  $\mathbf{Ag}(j, :)$ ,  $\mathbf{f}(j, :)$ ); % Passo 1.1.4
         $\mathbf{f}(j, :)$  := afinidade ( $\mathbf{C}^*$ ,  $\mathbf{Ag}(j, :)$ ); % Passo 1.1.5
         $\mathbf{M}$  := select ( $\mathbf{C}^*$ ,  $\mathbf{f}(j, :)$ ,  $\zeta$ ); % Passo 1.1.6
```

```

[M, f(j, :)] := suppress(M, 1/f(j, :),  $\sigma_d$ ); % Passo 1.1.7
S           := afinidade(M, M);           % Passo 1.1.8
[M, S]      := suppress(M, S,  $\sigma_s$ );    % Passo 1.1.9
Ab_{m}      := insere(Ab_{m}, M);         % Passo 1.1.8
end;
S           := afinidade(Ab_{m}, Ab_{m});   % Passo 1.2
[Ab_{m}, S]  := suppress(Ab_{m}, S,  $\sigma_s$ ); % Passo 1.3
Ab_{d}      := gera(d, L);                % Gera Ab_d aleatoriamente
Ab         := insere(Ab_{m}, Ab_{d});     % Passo 1.4
end;

```

5.5.4. Extração de Conhecimento e Estrutura da aiNet Treinada

Como os anticorpos de memória representam imagens internas dos antígenos, ou grupos de antígenos, aos quais a rede está submetida, eles são representados por cadeias de atributos de mesma dimensão dos dados de entrada (antígenos). Esta propriedade torna o processo de visualização dos anticorpos representados uma tarefa difícil caso a dimensão das moléculas seja superior a três ($L > 3$), o que geralmente ocorre em problemas do mundo real. Para amenizar esta dificuldade, é sugerida a utilização de técnicas de clusterização hierárquica para interpretar a rede resultante, permitindo a identificação e separação automática de clusters sem que seja necessária uma visualização direta da aiNet em um espaço de formas Euclidiano S^L , com L qualquer.

A estrutura da aiNet poderia simplesmente ser determinada interconectando-se todos os anticorpos da rede de acordo com a matriz **S**. Porém, esta estratégia não permite inferir nada sobre a estrutura intrínseca da aiNet e, conseqüentemente, dos antígenos. Uma forma simples de reduzir a complexidade da arquitetura de uma rede totalmente interconectada é suprimir as conexões cujos pesos extrapolam um limiar pré-definido. Esta idéia, embora simples, desconsidera qualquer informação contida na rede (indiretamente nos dados de treinamento), podendo levar a interpretações incorretas da aiNet resultante. O objetivo principal desta seção é o de fornecer ferramentas formais e sofisticadas de interpretação da aiNet gerada pelo algoritmo de treinamento apresentado na Seção 5.5.2. Explicitamente, objetivamos determinar: 1) a quantidade de anticorpos pertencentes a cada cluster, ou classe (sempre que um cluster corresponder a uma classe), 2) a forma de cada cluster e 3) a pertinência de anticorpos da rede a cada um dos clusters identificados. Para isso, utilizamos os parâmetros de saída da função para o treinamento da aiNet: **Ab**_{m}, matriz contendo os anticorpos de memória, e a matriz triangular superior **S** (Seção 5.5.6), contendo a afinidade entre estes anticorpos de memória, mais alguns princípios de análise de clusters.

O problema pode ser colocado da seguinte forma:

*Dada uma aiNet treinada composta por m anticorpos de memória (matriz **Ab**_{m} $\in S^{m \times L}$), e suas respectivas afinidades $s_{i,j}$, derive um esquema de clusterização capaz de detectar separações inerentes entre subconjuntos (clusters) de **Ab**_{m} em um espaço de formas governado por uma medida de afinidade D .*

No caso particular do algoritmo implementado, a medida de afinidade D é dada pela distância Euclidiana (Equação 3.1).

Os algoritmos a serem apresentados são conhecidos da literatura de análise de dados, mas serão adaptados e interpretados para o paradigma da engenharia imunológica, mais especificamente da aiNet. Sob esta perspectiva, a aiNet torna-se responsável pela extração de conhecimento e compressão da informação contida no conjunto de antígenos (dados de entrada), enquanto as técnicas de análise de clusters serão utilizadas para a detecção de clusters na rede, ou seja, interpretação da aiNet. A aiNet pode ser vista como um sistema de pré-processamento para as técnicas de análise de clusters, tornando-se uma ferramenta poderosa para a filtragem e pré-processamento de dados de um conjunto amostral.

5.5.4.1. Técnicas de Clusterização Hierárquica

Para ilustrar os métodos a serem empregados e os que serão propostos, considere o problema de clusterizar os dados apresentados na Figura 5.14(a). Como pode ser visto nesta figura, existem 40 amostras subdivididas em quatro clusters distintos e linearmente separáveis com 10 amostras cada. A Figura 5.14(b) representa os anticorpos resultantes de um treinamento da aiNet para os seguintes parâmetros de entrada: $gen = 10$, $n = 4$, $\zeta = 0.2$, $\sigma_d = 1.0$, $\sigma_s = 0.1$ e $d = 10$. A população inicial de anticorpos continha $N_0 = 10$ moléculas. As técnicas de clusterização hierárquica serão utilizadas para determinar quais anticorpos da rede estarão conectados entre si. Como pode ser visto, a rede resultante contém apenas 9 anticorpos, reduzindo a complexidade do problema (quantidade de dados) em 77,5%.

As técnicas de clusterização hierárquica podem ser subdivididas em métodos *aglomerativos* que fazem sucessivas fusões de m dados em grupos, e métodos *divisivos* que particionam o conjunto de m dados sucessivamente em partições menores. Os resultados de ambos os métodos, aglomerativos e divisivos, podem ser representados sob a forma de um *dendrograma*, que é um diagrama bidimensional que ilustra as fusões ou partições feitas em cada nível (Everitt, 1993). Para análise da aiNet, enfocaremos os métodos aglomerativos, mais especificamente o método do *vizinho mais próximo* (*nearest neighbor* ou *single link*), e a análise de clusters baseada no *centróide*.

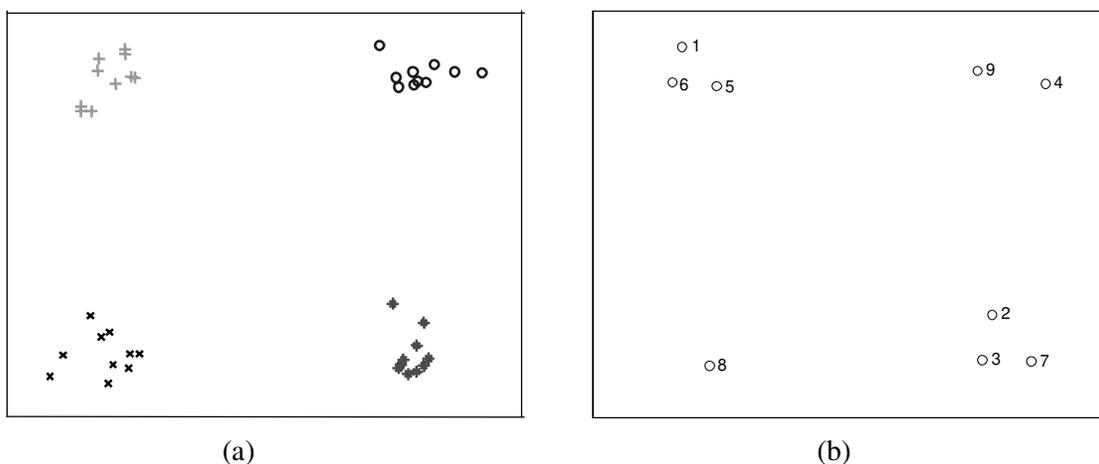


Figura 5.14. Problema didático ilustrativo. (a) Antígenos. (b) Anticorpos resultantes da aplicação do algoritmo de treinamento da aiNet ao problema apresentado em (a). Parâmetros de treinamento: $gen = 10$, $n = 4$, $\zeta = 0.2$, $\sigma_d = 1.0$, $\sigma_s = 0.1$ e $d = 10$.

Dada a conceituação da aiNet sob a forma de um grafo interconectado de nós, ou anticorpos (Definição 5.1), algumas estratégias da teoria de grafos para detecção e descrição de clusters serão empregadas, em particular a *árvore geradora mínima* (*minimal spanning tree*).

Um dos parâmetros de saída da função que implementa a aiNet (Seção 5.5.3) é a matriz **S** de dimensão $N \cdot (N - 1) / 2$. A matriz **S** de afinidades entre os anticorpos de memória da rede e sua matriz de coordenadas **Ab**_{m} estão disponíveis, portanto a aplicação de técnicas hierárquicas de clusterização como o vizinho mais próximo, o vizinho mais distante, e o centróide, para a construção do dendrograma, é direta. De forma resumida, estes algoritmos operam como a seguir:

1. *Vizinho mais próximo* (*nearest neighbor-NN* ou *single link*): grupos compostos inicialmente por um único indivíduo são fundidos de acordo com a distância entre seus membros mais próximos, os grupos com menores distâncias sendo fundidos primeiro. Cada fusão reduz em uma unidade a quantidade de grupos.
2. *Vizinho mais distante* (*furthest neighbor* ou *complete link*): este método opera de forma exatamente oposta ao anterior, no sentido de que a distância entre grupos é tomada entre os pares de indivíduos mais distantes.
3. *Centróide*: os grupos estão contidos em um espaço Euclidiano e são substituídos pelas coordenadas de seus centróides. A distância entre grupos é definida como a distância entre seus centróides. O procedimento funde grupos de acordo com a distância entre seus centróides. Os grupos com menores distâncias sendo fundidos primeiro.

A partir da matriz de similaridades **S** e empregando os métodos descritos acima, deseja-se construir uma árvore, ou conjunto aninhado de clusters dos elementos, de forma a produzir um apelo visual da similaridade entre os grupos gerados pelos anticorpos da rede. Antes de definir o que é um dendrograma, considere os seguintes conceitos da teoria de grafos (Zahn, 1971):

- *Grafo G ponderado*: conjunto de pontos chamados *nós* (ou anticorpos, no caso da aiNet) e um conjunto de pares de nós chamados de *arcos* (ou conexões, no caso da aiNet) com um *peso* associado a cada arco;
- Um *caminho* em um grafo é uma seqüência de arcos que têm um nó comum;
- Um *circuito* é um caminho fechado;
- Um *grafo conexo* possui caminhos entre quaisquer pares de nós; e
- Uma *árvore* é um grafo conexo sem nenhum circuito.

Definição 5.3: Um *dendrograma* é definido como uma árvore com raiz e ponderada, onde todos os nós terminais estão a uma mesma distância (comprimento do caminho) da raiz (Lapointe & Legendre, 1991).

Detalhes sobre a construção de um dendrograma a partir de uma matriz de similaridade (ou dissimilaridade) podem ser encontrados em Hartigan (1967) e Hubert *et al.* (1998). Para a interpretação da aiNet, três características podem descrever adequadamente um dendrograma: 1) topologia, 2) rótulos e 3) altura dos clusters (Lapointe & Legendre, 1995):

1. A *topologia* de um dendrograma representa sua estrutura ou forma. Ela é influenciada pelo algoritmo de clusterização hierárquica empregado;
2. Os *rótulos* identificam os m elementos sendo clusterizados; e
3. A *altura dos clusters* permite que a distância entre os clusters seja descrita por uma escala de alturas associada aos $m - 1$ nós do dendrograma, partindo da raiz.

A Figura 5.15 ilustra dois dendrogramas para os anticorpos apresentados na Figura 5.14(b). Observe a topologia, rótulos e alturas dos clusters. A Figura 5.15(a) ilustra o dendrograma para o método do vizinho mais próximo e a Figura 5.15(b) corresponde ao método do centróide.

Virtualmente nenhum procedimento de clusterização fornece informações quanto ao número de clusters presentes nos dados. Procedimentos não-hierárquicos usualmente requerem a especificação deste parâmetro antes de que a clusterização seja efetuada (razão pela qual escolhemos os métodos hierárquicos ao invés dos não-hierárquicos) e os métodos hierárquicos produzem uma série de soluções variando de m clusters até uma solução com um único cluster. Como pode ser visto na Figura 5.15, o dendrograma pode ser quebrado em diferentes níveis resultando em clusterizações distintas dos dados. Neste caso, as grandes variações nas alturas dos clusters permitem-nos distinguir quatro clusters entre os anticorpos da rede, estando de acordo com a Figura 5.14(b). Este procedimento é chamado de *tamanho do passo (stepsize)* e envolve a análise da diferença entre os *valores de fusão* entre os níveis hierárquicos. Uma ampla revisão sobre diferentes técnicas de determinação do número de clusters em um conjunto de dados pode ser encontrada em Milligan *et al.* (1985).

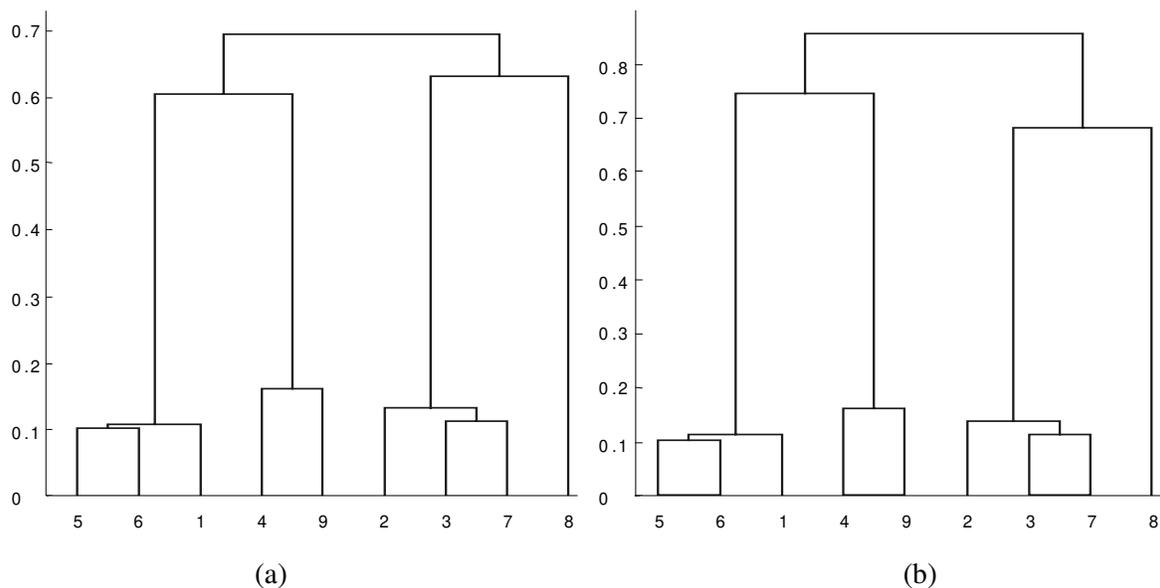


Figura 5.15. Dendrogramas para os anticorpos apresentados na Figura 5.14, o eixo x contém os rótulos dos anticorpos da rede e o eixo y apresenta a altura dos clusters. (a) Vizinho mais próximo. (b) Centróide.

A *árvore geradora mínima* (MST – *minimal spanning tree*) de um grafo constitui um mecanismo poderoso para determinação de uma estratégia adaptativa de detecção do esquema de conectividade do grafo (Zahn, 1971; Leclerc, 1995). Sendo assim, a MST funcionará como uma alternativa para a determinação da arquitetura da aiNet.

Definição 5.4: Uma árvore é uma *árvore geradora* (*spanning tree*) se ela é um subgrafo contendo todos os nós do grafo.

Definição 5.5: Uma *árvore geradora mínima* de um grafo é uma árvore geradora com peso mínimo, onde o peso de uma árvore é definido como sendo a soma dos pesos de seus arcos.

A Figura 5.16(a) ilustra a árvore geradora mínima para os anticorpos definidos pela aiNet. A visualização desta árvore só é factível caso a dimensão dos anticorpos seja inferior a 3 ($L < 3$). Utilizando-se o algoritmo conhecido como *algoritmo de Prim* (Prim, 1957) para construir a MST, é possível desenhar um histograma como na Figura 5.16(b) representando as distâncias entre anticorpos vizinhos.

Definição 5.6: Um *caminho minimax* entre um par de nós é aquele que minimiza o custo (peso máximo do caminho), sobre todos os caminhos.

Esta definição é importante no contexto da aiNet, pois a MST percorre os caminhos minimax, forçando a conexão entre dois nós i e j mais próximos antes de sair em busca de outro nó. Se a MST de um grafo G é única, então o conjunto de caminhos minimax de G define a MST, senão ele define a união de todas as MSTs de G (Carroll, 1995).

A MST pode ser utilizada para determinar a quantidade de clusters da aiNet, que será igual ao número de picos de seu *histograma* mais 1, como ilustrado na Figura 5.16(b). Quando o algoritmo de treinamento da aiNet gera mais do que um anticorpo para cada cluster, a quantidade de clusters também pode ser medida pelo número de vales do respectivo histograma, porém, sem que haja uma visualização dos anticorpos resultantes, é difícil avaliar a quantidade de anticorpos gerados para cada cluster antes de fazer a análise do histograma da MST. Estes picos (vales) no histograma correspondem a grandes variações nas distâncias minimax entre os anticorpos gerados, permitindo a identificação de regiões com alta densidade de anticorpos intercaladas por regiões de baixa densidade no espaço de formas. Por outro lado, os dendrogramas nos permitem não apenas definir a quantidade de clusters, mas também os elementos pertencentes a cada um deles. Para que seja possível definir automaticamente a quantidade de clusters e seus elementos a partir de uma MST, podemos utilizar algumas das técnicas propostas por Zahn (1971).

O problema agora resume-se a remover conexões de uma MST tal que as subárvores resultantes correspondam aos clusters observáveis da aiNet. No exemplo da Figura 5.14, é preciso um algoritmo capaz de detectar e remover as conexões 5-9, 2-4 e 3-8, destacadas na Figura 5.17.

O seguinte critério é empregado:

Uma conexão (i,j) da MST cujo peso associado $s_{i,j}$ for significativamente maior do que a média dos pesos vizinhos, em ambos os lados da conexão (i,j) , deverá ser removida. Esta conexão é chamada inconsistente.

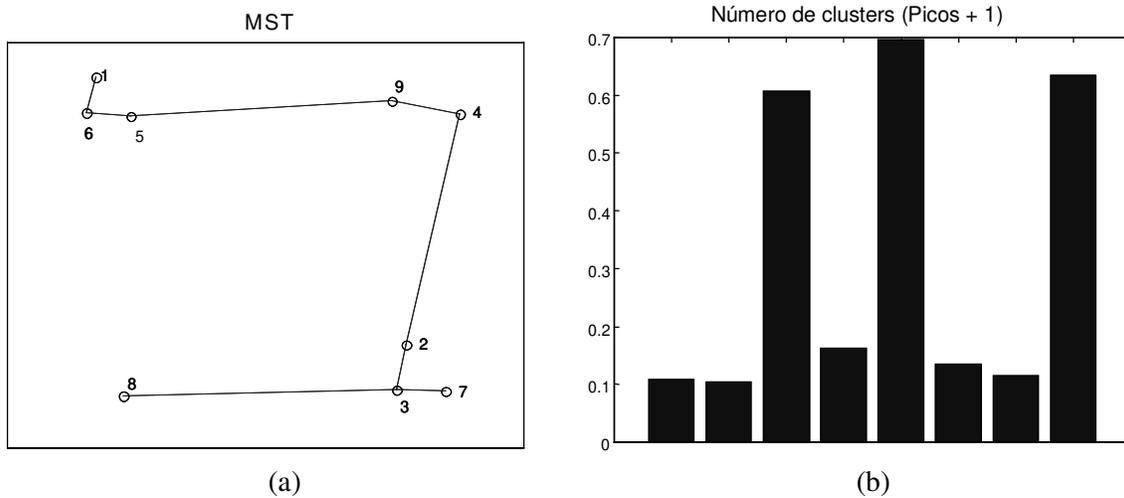


Figura 5.16. Determinação da quantidade de clusters da aiNet. (a) MST. (b) Histograma da MST apresentada em (a).

Existem duas formas diretas de medir a significância ou inconsistência mencionada acima:

1. Verificar quantos desvios padrões separam $s_{i,j}$ do peso médio dos vizinhos em cada lado da conexão (i,j) ; e
2. Calcular o *fator*, ou *razão*, r entre $s_{i,j}$ e as respectivas médias. Neste caso, uma desigualdade triangular não se faz necessária e, portanto, não precisamos restringir a análise a um espaço métrico.

Para ilustrar este critério, suponha que todas as conexões da MST cujo $s_{i,j}$ é maior do que a média dos pesos das conexões vizinhas mais 2 desvios padrões serão removidas, ou seja, um fator $r = 2$ é adotado. As conexões 5-9, 2-4 e 3-8 serão selecionadas para remoção (linhas tracejadas na Figura 5.17).

Após determinar quais conexões serão removidas, é possível detectar a quantidade de clusters existentes na aiNet e seus respectivos anticorpos. Se c_i , $i = 1, \dots, n_c$ (n_c = quantidade de clusters), representa o i -ésimo cluster da rede, então $c_1 = [1,6,5]$, $c_2 = [9,4]$, $c_3 = [2,3,7]$ e $c_4 = [8]$.

O critério proposto falharia na determinação correta dos clusters nas seguintes ocasiões:

- Quando o algoritmo de treinamento da aiNet for capaz de determinar a arquitetura mínima para um certo conjunto de dados, ou seja, um único anticorpo representando cada cluster. Este problema pode ser aliviado através de uma redução no valor do parâmetro σ_s , resultando em uma rede com anticorpos mais específicos. Além disso, geralmente a quantidade de clusters dos dados não é conhecida, e caso seja, σ_s pode ser ajustado para que a aiNet atinja uma dimensão satisfatória; e
- Quando não existir uma distância relativa entre os clusters que seja maior do que a distância entre elementos do mesmo cluster. Embora essa pareça uma limitação do algoritmo, qualquer outra estratégia não-supervisionada de clusterização, baseada em critérios de distância, falharia nesta situação.

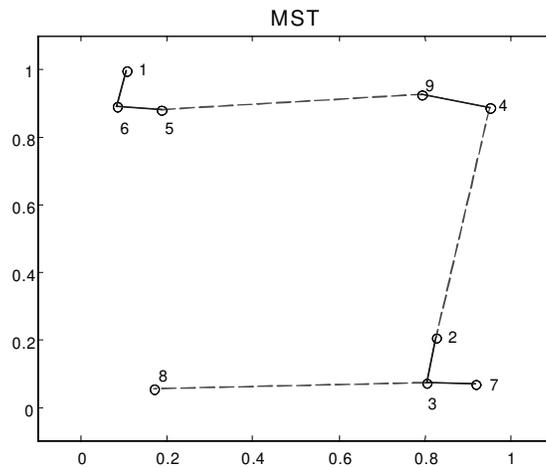


Figura 5.17. Conexões a serem removidas (linhas tracejadas) baseadas no fator $r = 2$.

Como último aspecto de clusterização a ser considerado, temos o problema da *representação dos clusters*. Suponha que cada cluster possa ser unicamente representado pelo seu *centro de massa*, ou *centróide*, v_i , $i = 1, \dots, n_c$, e que a distância entre os clusters seja definida pela distância entre os centróides. A Figura 5.18 ilustra os anticorpos resultantes do treinamento da aiNet e as conexões automaticamente determinadas pelo método proposto para a MST considerando um fator $r = 2$. As estrelas representam os centróides de cada cluster.

A representação dos clusters pelo seu centróide de massa é geralmente útil quando os clusters são compactos e isotrópicos (com formas semelhantes). Nos casos em que os clusters são não-compactos e/ou não-isotrópicos, este esquema não é apropriado para representá-los. Estes casos ficarão mais claros no próximo capítulo, onde apresentaremos simulações computacionais do algoritmo. A representação dos clusters pelos seus centros de massa é interessante pois ela permite especificar *níveis de pertinência* de cada anticorpo da rede a cada um dos clusters detectados, resultando em um esquema de *clusterização gradual* (ou *nebulosa*).

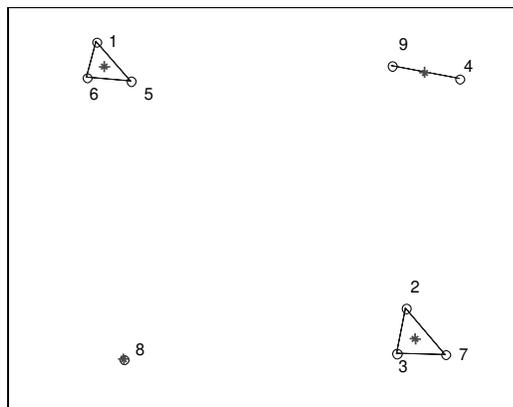


Figura 5.18. aiNet resultante. A rede é composta por quatro clusters (subgrafos) distintos. As estrelas representam o centróide de cada cluster.

5.5.4.2. Clusterização Gradual da aiNet

Os métodos de clusterização discutidos na seção anterior geram *partições* no conjunto de dados. Em uma partição, cada elemento pertence a apenas um cluster. Assim, os clusters são disjuntos neste esquema de clusterização. A *clusterização fuzzy*, ou *clusterização nebulosa*, estende esta noção de forma a associar cada padrão a todos os clusters através de uma função de pertinência (Bezdek & Pal, 1992). As técnicas de clusterização nebulosa mais conhecidas são os algoritmos de *fuzzy k-means* e *fuzzy c-means*, que iterativamente atualizam os centros dos clusters de acordo com uma matriz de proximidade \mathbf{U} atual, até que a variação em \mathbf{U} seja desprezível. Uma breve exposição de espaços de partições nebulosa é apresentada por Bezdek & Pal (1992):

Seja n_c um inteiro, $1 < n_c < M$, e seja $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M\}$ um conjunto de M dados não rotulados em \mathcal{R}^L . Dado \mathbf{X} , dizemos que n_c subconjuntos nebulosos $\{u_i: \mathbf{X} \rightarrow [0,1]\}$ constituem uma *c-partição nebulosa* de \mathbf{X} caso os $(n_c \cdot M)$ valores $\{u_{i,k} = u_i(\mathbf{x}_k), 1 \leq k \leq M, 1 \leq i \leq n_c\}$ satisfaçam as três condições abaixo:

$$0 \leq u_{i,k} \leq 1 \quad \text{para todo } i, k; \quad (5.24)$$

$$\sum u_{i,k} = 1 \quad \text{para todo } k; \quad (5.25)$$

$$0 < \sum u_{i,k} < 1 \quad \text{para todo } i. \quad (5.26)$$

Cada conjunto de $(n_c \cdot M)$ valores satisfazendo as condições de (5.24) a (5.26) pode ser colocado em uma matriz $\mathbf{U} = [u_{i,k}]$, $\mathbf{U} \in \mathcal{R}^{n_c \times M}$. O conjunto de todas essas matrizes é chamado de *c-partição nebulosa não degenerada* de \mathbf{X} .

Após determinar a quantidade e os membros de cada cluster e representá-los pelos seus respectivos centróides, é possível aplicar o conceito de clusterização nebulosa (ou gradual), em que cada anticorpo terá um valor de pertinência em relação a cada um dos centróides dos clusters. No contexto da engenharia imunológica, a clusterização nebulosa relaxa a pertinência dos anticorpos da rede em relação aos centróides dos clusters, $\mathbf{U} = [u_{i,k}]$, que neste caso, pode assumir qualquer valor no intervalo $[0,1]$, satisfazendo a condição (5.24). As condições (5.25) e (5.26) são relaxadas, de forma que a soma das pertinências não seja necessariamente igual a 1. Esta é uma forma natural de estender as *c-partições nebulosas não-degeneradas* de \mathbf{X} a um espaço de soluções nebulosas ainda maior.

A matriz \mathbf{U} para a aiNet pode ser determinada calculando-se a distância \mathbf{U}^* entre todas os m anticorpos de memória resultantes da rede e os n_c centróides dos clusters v_i , $i = 1, \dots, n_c$, normalizando suas linhas no intervalo $[0,1]$ e passando-as por uma função não-linear, de forma que quanto menor a distância entre um anticorpo e seu respectivo centróide, mais perto da unidade estará seu valor de pertinência. Este último passo pode ser efetuado aplicando-se uma função do tipo sigmoideal a $\mathbf{U} = 1./\mathbf{U}^*$, onde o operador $./$ significa que cada valor da matriz \mathbf{U} será determinado pela inversão do respectivo elemento de \mathbf{U}^* .

5.5.5. Caracterização e Comparação

Baseado nas propriedades das redes neurais artificiais apresentadas na Seção 4.2, é possível classificar a aiNet como uma abordagem *conexionista*. Sob esta perspectiva, os anticorpos da rede correspondem a seus nós, a posição dos anticorpos no espaço de formas é o *estado do nó*, os valores de concentração e afinidade determinam o perfil de conectividade da rede, e os antígenos correspondem aos *padrões de treinamento*. As conexões entre os nós, representadas pela matriz **S** (ilustradas genericamente na Figura 5.1(a) e para o caso particular do exemplo tratado nesta seção na Figura 5.18), permitem avaliar a afinidade existente entre os nós da rede, quantificando o reconhecimento idiotípico. A *representação em grafo* da aiNet descreve sua *arquitetura* final, definindo a quantidade de clusters e seus elementos constituintes. A *dinâmica* da rede governa a aprendizagem e plasticidade do sistema, enquanto a *metadinâmica* é responsável por uma melhor exploração do espaço de formas e manutenção da diversidade populacional.

O algoritmo de treinamento pode ser classificado como *competitivo* (Seção 4.2.1.2.2), pois um conjunto de anticorpos está competindo entre si pelo reconhecimento antigênico e conseqüente sobrevivência. A estrutura geral da aiNet é diferente das redes neurais artificiais quanto à funcionalidade das unidades e conexões. No caso das RNAs, os nós são unidades de processamento e as conexões armazenam as informações que definem a função de transferência da rede (conhecimento). Na aiNet, as unidades armazenam as informações (imagens internas do universo antigênico), as conexões quantificam a afinidade, ou grau de interação dos anticorpos da rede, e o processamento é efetuado pelo algoritmo de seleção clonal (CLONALG para reconhecimento de padrões) somado a mecanismos de interação e supressão da teoria da rede imunológica. A Tabela 4.1 apresenta genericamente as principais similaridades entre a engenharia imunológica e as redes neurais artificiais.

Além da visão conexionista da aiNet, é possível categorizar seu algoritmo de treinamento como *evolutivo*, pois ele segue os principais passos do algoritmo CLONALG, proposto na Seção 5.3 e caracterizado na Seção 5.3.3. Uma vantagem da evolução da aiNet sobre a evolução do CLONALG é a utilização da teoria da rede imunológica para o controle da quantidade de anticorpos na rede. Entretanto, a introdução de um mecanismo para avaliar a afinidade entre os anticorpos da rede traz um elevado custo computacional associado, como será apresentado na próxima seção.

A essência deste modelo de rede imunológica artificial é diferente da maioria dos modelos existentes na literatura em dois aspectos. Primeiro, e mais importante, ele é discreto no tempo (iterativo) e não contínuo (baseado em equações diferenciais ordinárias). Segundo, o objetivo principal da aiNet é o de resolver problemas de engenharia e não reproduzir fenômenos biológicos objetivando uma melhor compreensão dos mesmos.

Existem dois modelos na literatura que podem ser diretamente comparáveis à aiNet: uma rede imunológica artificial proposta por Hunt & Cooke (1996) e outra por Timmis (2000).

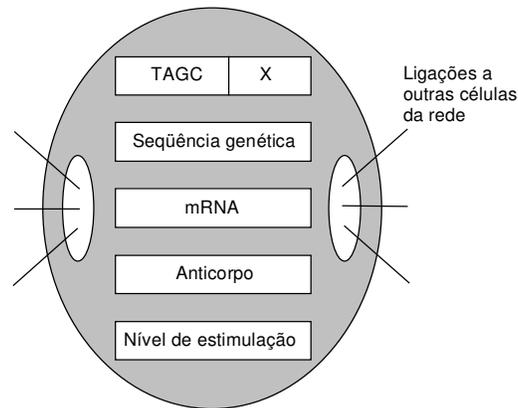


Figura 5.19. Estrutura de uma célula B (Hunt & Cooke, 1996).

O sistema imunológico artificial desenvolvido por Hunt & Cooke (1996) apresenta um modelo de rede imunológica artificial contendo os seguintes elementos:

1. Medula óssea: responsável pela geração das células do SIA e definição do ponto de inserção do antígeno na rede;
2. Célula B: composta por diversos elementos, como ilustrado na Figura 5.19, onde as bibliotecas genéticas, a seqüência de genes e o mRNA fazem parte do procedimento de geração das moléculas de anticorpo. Este modelo também contém o nível de estimulação da célula;
3. Anticorpos: possuindo um paratopo, a medida de afinidade e o processo de ligação;
4. Antígeno: composto por um único epítipo.

A Tabela 5.4 faz uma comparação entre a aiNet e o modelo de Hunt & Cooke (1996).

Tabela 5.4. Relação entre a aiNet e o SIA proposto por Hunt & Cooke (1996).

	aiNet	SIA
Nós	Cadeia de atributos (anticorpos)	Célula B (Figura 5.19)
Inicialização da rede	Aleatória, com pouca influência no comportamento final da aiNet	Crítica para o tempo de processamento
Apresentação antigênica	O antígeno é apresentado para toda a rede	O antígeno é apresentado para uma região aleatória da rede
Medida de afinidade	Equação 3.3	Equação 3.4
Morte celular	Eliminação de anticorpos com baixa afinidade antigênica e alta afinidade em relação a outro anticorpo	Eliminação de células com baixo nível de estimulação (considera interações Ag-Ab e Ab-Ab, Equação (5.27))
Hipermutação	Importante para a aprendizagem	Promove diversidade

O nível de estimulação de cada célula, dado pela Equação (5.27), e a hipermutação somática deste SIA são baseadas em uma variação do modelo de rede proposto por Farmer *et al.* (1986) e brevemente apresentado na Seção 3.4.3.5.

$$estimulação = c \left[\sum_{j=1}^N m(a, xe_j) - k_1 \sum_{j=1}^N m(a, xp_j) + k_2 \sum_{j=1}^n m(a, y) \right] - k_3 \quad (5.27)$$

onde o primeiro termo representa a afinidade dos anticorpos em relação a seus vizinhos, o segundo termo representa a não-afinidade dos anticorpos em relação a seus vizinhos, o terceiro termo corresponde à ligação Ag-Ab e o último termo modela a tendência das células pouco estimuladas morrerem. Os coeficientes k_i , $i = 1, \dots, 3$, são constantes.

Este sistema imunológico artificial foi posteriormente aperfeiçoado (Hunt *et al.*, 1999) para eliminar algumas de suas limitações, como inicialização eficiente da rede e aplicação de medidas alternativas de afinidade. Timmis (2000) consolidou esta proposta de SIA através do desenvolvimento de uma técnica para o controle da dimensão da arquitetura resultante da rede, e de diversas aplicações e extensivas comparações com outras abordagens na área de análise de dados.

5.5.6. Complexidade Computacional

O algoritmo proposto para o treinamento da aiNet possui um laço interno que percorre todos os antígenos da população e um laço externo que varre as gerações de anticorpos. No laço interno (dos antígenos), seis etapas principais de processamento podem ser identificadas, juntamente com seus respectivos custos computacionais associados:

1. A determinação da afinidade entre o antígeno dado e todos os anticorpos da rede (Passo 1.1.1): custo $O(N)$;
2. A seleção dos n melhores indivíduos (Passo 1.1.2): custo $O(N)$;
3. A mutação direcionada da população C de clones gerados (Passo 1.1.4): custo $O(N_c \cdot L)$;
4. A determinação da afinidade entre o antígeno dado e todos os N_c clones gerados (Passo 1.1.5): custo $O(N_c)$;
5. A re-seleção dos ζ melhores indivíduos maduros (Passo 1.1.6): custo $O(N_c)$; e
6. O cálculo da afinidade entre os c clones de memória (Passo 1.1.8): custo $O(c^2)$, onde c corresponde a $\zeta \cdot N$ anticorpos, menos aqueles eliminados no Passo 1.1.7.

No laço externo (rede), a principal etapa de processamento é a determinação da afinidade entre os m anticorpos de memória da rede, com custo associado $O(m^2)$, semelhante ao algoritmo SAND (Seção 5.2). Devido à característica assintótica da complexidade e assumindo que $m > c$, o termo de ordem $O(m^2)$ predomina sobre todos os outros, portanto a complexidade da aiNet está resumida na Tabela 5.5. Note que a quantidade m de anticorpos de memória na rede varia dinamicamente, sendo maior nas primeiras iterações, como será visto na Seção 6.5.3, que faz uma análise de sensibilidade dos parâmetros da aiNet definidos pelo usuário.

Tabela 5.5. Complexidade computacional por geração para execução do algoritmo de treinamento da aiNet, onde N é o tamanho atual da população de anticorpos, N_c a quantidade de clones gerada a partir da seleção dos n melhores indivíduos de \mathbf{Ab} , L é o comprimento das cadeias de atributos que representam as moléculas e m é a quantidade de anticorpos de memória.

	<i>Tempo de Processamento</i>	<i>Memória</i>
aiNet	$O(m^2)$	$\propto N + L(n + N_c + N) + m(m - 1)/2$

A quantidade de memória necessária para executar o algoritmo é igual à do algoritmo CLONALG proposto na Seção 5.3, mais o custo de se armazenar a matriz \mathbf{S} contendo as afinidades entre os m anticorpos de memória da rede. Lembrando que esta matriz é simétrica e tem diagonal principal nula, permitindo armazenar apenas $m(m-1)/2$ elementos, a quantidade de memória requerida para o algoritmo de treinamento da aiNet está apresentada na Tabela 5.5.

CAPÍTULO 6

APLICAÇÕES EM ENGENHARIA

Este capítulo apresenta os resultados computacionais provenientes da aplicação das quatro ferramentas de engenharia imunológica propostas no Capítulo 5 (SAND, CLONALG, ABNET e aiNet) a diversos problemas de engenharia, enfocando as áreas de aprendizagem de máquina, reconhecimento de padrões, aproximação de funções, análise de dados e otimização. Juntamente com as aplicações das ferramentas, são feitas comparações de desempenho com outros algoritmos conhecidos na literatura, onde a grande maioria deles foi revisada no Capítulo 4.

“As aplicações práticas de uma ciência geralmente antecedem o desenvolvimento da própria ciência” – N. K. Jerne

6.1. Introdução

Existe uma grande quantidade de problemas em engenharia cujas soluções, inclusive aquelas obtidas por estratégias de inteligência computacional, ainda são insatisfatórias. Destas aplicações, podemos destacar os problemas de busca (ou otimização) multimodal, análise de dados e classificação de padrões. Além disso, muitas das estratégias de sistemas inteligentes como as redes neurais artificiais e os algoritmos evolutivos possuem deficiências que vão da inicialização à estruturação, chegando até suas estratégias de aprendizagem/evolução. Este capítulo visa demonstrar como as ferramentas propostas nesta tese podem ser utilizadas na solução de problemas de engenharia como os citados acima e, também, como elas podem ser aplicadas na otimização e desenvolvimento de outras abordagens de sistemas inteligentes, com destaque para as redes neurais artificiais e os algoritmos evolutivos.

O modelo proposto para geração de diversidade populacional, SAND, foi o primeiro a ser desenvolvido nesta tese e, por isso, o primeiro a ser apresentado no Capítulo 5. Por outro lado, o algoritmo SAND possui aplicações em diversas áreas, inclusive em outras estratégias propostas nesta tese como o algoritmo CLONALG. Sendo assim, as aplicações em engenharia do algoritmo de seleção clonal (CLONALG) serão apresentadas antes das aplicações do algoritmo SAND.

A Seção 6.3.1 que aplica a ABNET ao problema de decisão de lógica majoritária (MLD – *major logic decision*) foi desenvolvida em cooperação com Getúlio A. de Deus Júnior, em fase de doutoramento pelo Departamento de Comunicações (DECOM) da Faculdade de Engenharia Elétrica e de Computação (FEEC) da Unicamp.

6.2. CLONALG

O algoritmo de seleção clonal (descrito na Seção 5.3), CLONALG, foi desenvolvido em duas versões: uma para resolver problemas de reconhecimento de padrões e aprendizagem de máquina e outra para solucionar problemas de otimização, com destaque para otimização multimodal. Com o objetivo de avaliar o desempenho da versão para o reconhecimento de padrões, ele será aplicado ao reconhecimento de caracteres binários. No caso de problemas de otimização, o algoritmo será testado em diversas funções uni- e multi-dimensionais, e seus resultados comparados com o algoritmo genético clássico e o método de fitness sharing discutido na Seção 4.4.5.

6.2.1. Reconhecimento de Padrões

A capacidade de aprendizagem e aquisição de memória do algoritmo CLONALG foi verificada através de sua aplicação a um problema de reconhecimento de caracteres binários.

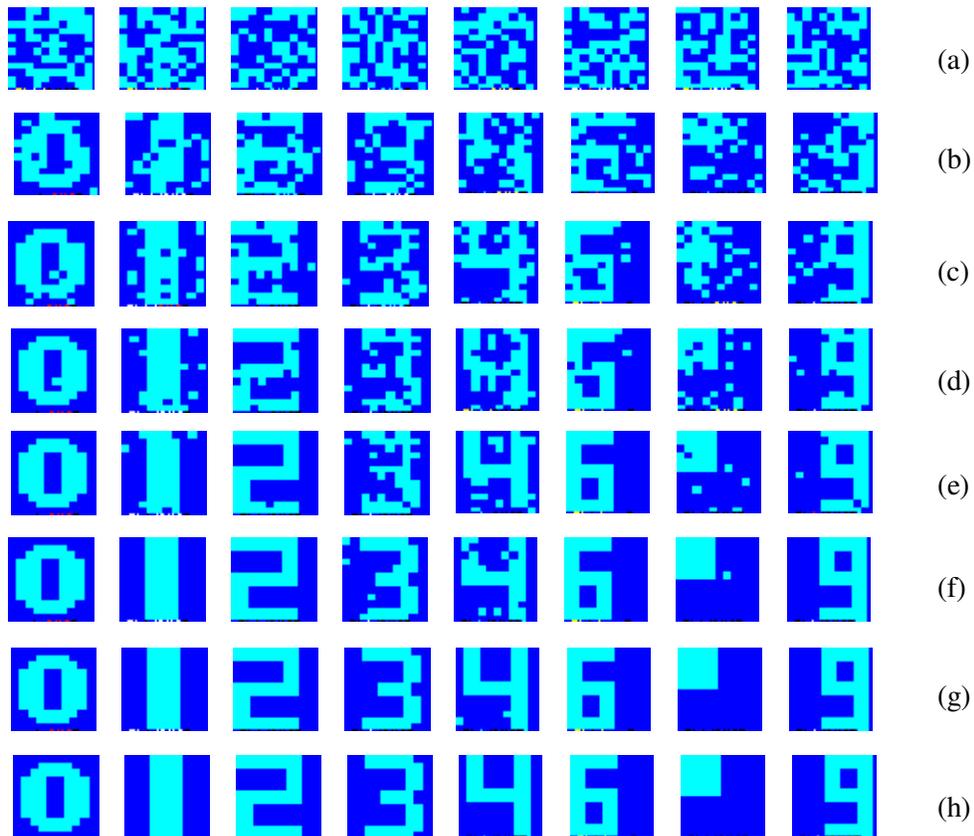


Figura 6.1. Aplicação do algoritmo CLONALG ao problema de reconhecimento de caracteres binários, onde (h) representa o conjunto de antígenos (padrões) a serem aprendidos. Conjunto de memória $\mathbf{Ab}_{\{m\}}$ inicial (a), após 20 (b), 50 (c), 75 (d), 100 (e), 150 (f), 200 (g), e 250 gerações (g). O algoritmo convergiu após 300 gerações.

O objetivo desta seção, é demonstrar que uma variação genética aleatória e cumulativa, juntamente com uma forte pressão seletiva, é capaz de produzir indivíduos com afinidades antigênicas cada vez maiores, simulando o processo de maturação de afinidade da resposta imune adaptativa. Neste caso, assumimos a existência de uma população de antígenos, a serem reconhecidos, representada em um espaço de formas binário de Hamming composto por oito cadeias de atributos ($M = 8$) de comprimento $L = 120$. Estes caracteres já foram utilizados por Lippmann (1987) para avaliar a capacidade de armazenamento e recuperação de padrões de redes neurais discretas de Hopfield (Seção 4.2.2.3). O repertório total \mathbf{Ab} de anticorpos é composto por $N = 10$ indivíduos, onde $m = 8$ deles compõem o conjunto de memória $\mathbf{Ab}_{\{m\}}$. O conjunto de antígenos a serem reconhecidos está ilustrado na Figura 6.1(h). A Figura 6.1(a) apresenta o conjunto de memória $\mathbf{Ab}_{\{m\}}$ inicial, e a Figura 6.1 de (b) a (h) representa o processo de maturação do conjunto de memória ao longo das gerações celulares. A medida de afinidade empregada foi a distância de Hamming entre os antígenos e os anticorpos, dada pela Equação 3.3.

Considerando o algoritmo de treinamento apresentado na Seção 5.3.2.1.1, a seguinte linha de comando foi utilizada: $[\mathbf{Ab}_m, F] = \text{clonalg}(\mathbf{Ab}, \mathbf{Ag}, 120, 300, 5, 10, 0)$.

6.2.2. Otimização

O algoritmo CLONALG reproduz aqueles indivíduos com altas afinidades e seleciona seus clones maturados de maior afinidade. Esta estratégia sugere um algoritmo capaz de realizar uma busca local em torno de cada indivíduo da população. Para avaliar a aplicabilidade deste método a problemas de otimização, considere inicialmente as seguintes funções multimodais unidimensionais ilustradas na Figura 6.2:

- $g_1(x) = \text{sen}^6(5\pi x)$; e
- $g_2(x) = 2^{(-2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$.

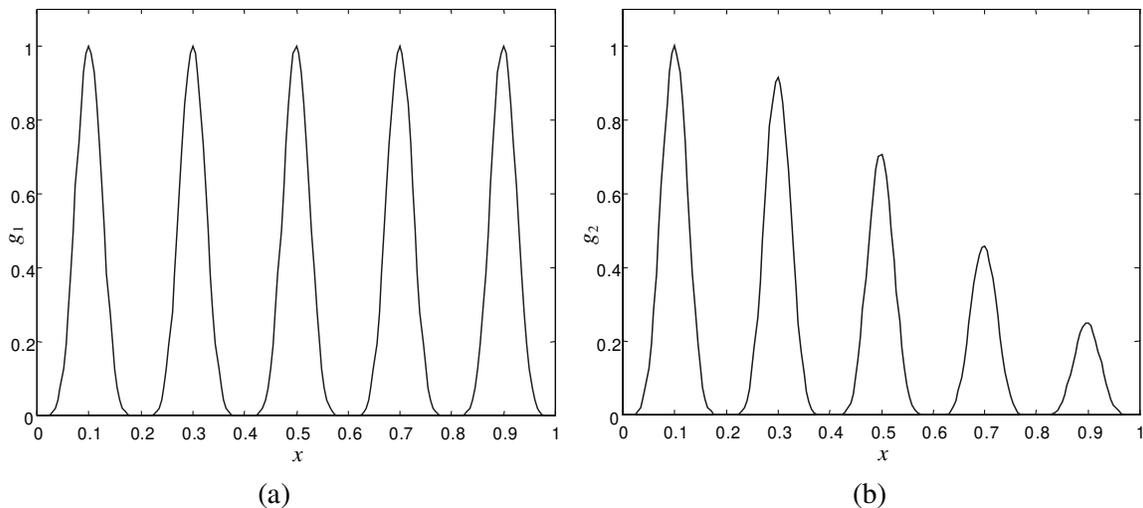


Figura 6.2. Funções a serem maximizadas. (a) Múltiplos ótimos globais, $g_1(x) = \text{sen}^6(5\pi x)$. (b) Um único ótimo global e vários ótimos locais, $g_2(x) = 2^{(-2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$.

Estas funções foram utilizadas por Goldberg & Richardson (1987) para estudar a capacidade de otimização multimodal do método de compartilhamento de fitness (*fitness sharing*) apresentado na Seção 4.4.5.

O objetivo é encontrar a maior quantidade de valores máximos destas funções, sejam eles locais ou globais. Considerando o algoritmo de treinamento apresentado na Seção 5.3.2.2.1, a linha de comando `[Ab, f]=clonalg(Ab, 22, 50, 50, 10, 0)` foi utilizada. O repertório **Ab** possui $N = 50$ anticorpos, codificados por cadeias binárias de comprimento $L = 22$ em um espaço de formas binário de Hamming. O comprimento $L = 22$, corresponde a uma precisão de seis casas decimais. A variável x está definida no intervalo $[0,1]$, e o mapeamento de uma cadeia binária $m = \langle m_L, \dots, m_2, m_1 \rangle$ em um número real x é executado em dois passos:

- Converta a cadeia binária $m = \langle m_L, \dots, m_2, m_1 \rangle$ da base 2 para a base 10:

$$\langle m_L, \dots, m_2, m_1 \rangle_2 = \left(\sum_{i=0}^{21} m_i \cdot 2^i \right)_{10} = x'$$

- Encontre o correspondente valor real para x : $x = x_{\min} + x' \cdot \frac{x_{\max} - x_{\min}}{2^{22} - 1}$, onde $x_{\max} = 1$ e $x_{\min} = 0$.

A medida de afinidade corresponde à avaliação da função $g_i(x)$, $i = 1, 2$, após decodificarmos x como descrito acima. A Figura 6.3 ilustra as populações iniciais e a Figura 6.4 apresenta as populações otimizadas pelo algoritmo CLONALG.

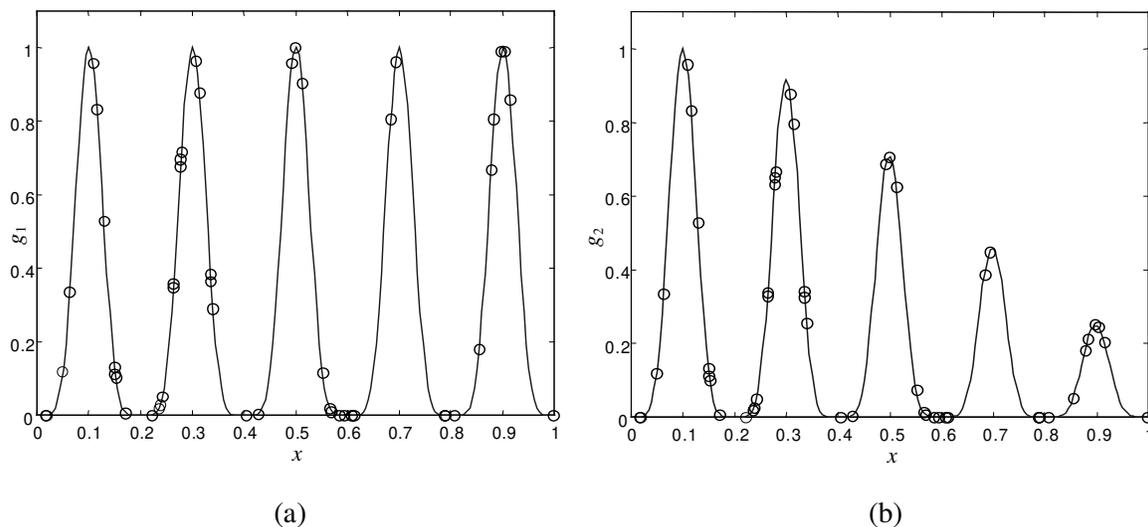


Figura 6.3. Populações iniciais compostas por $N = 50$ indivíduos para maximização das funções: (a) $g_1(x) = \text{sen}^6(5\pi x)$ e (b) $g_2(x) = 2^{-(2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$.

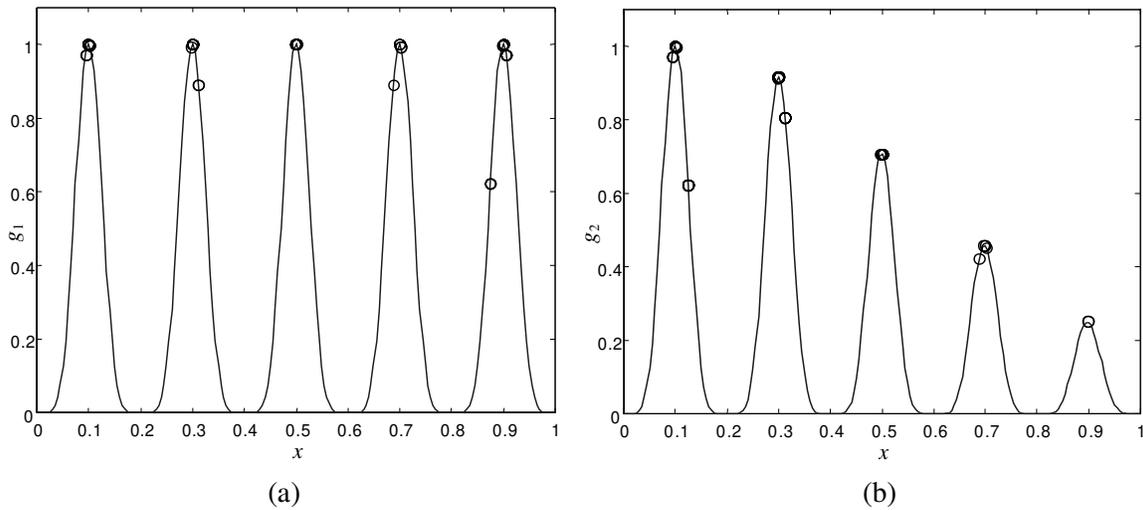


Figura 6.4. Algoritmo CLONALG aplicado ao problema de maximização das funções: (a) $g_1(x) = \text{sen}^6(5\pi x)$ e (b) $g_2(x) = 2^{-(2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$.

Para comparar o desempenho do algoritmo CLONALG, aplicamos um GA (Seção 4.4.3) com seleção por torneio binário, crossover simples e mutação de múltiplos pontos. Os parâmetros escolhidos para a evolução do GA foram: $p_c = 0.6$, $p_m = 0.01$ e $k = 0.6$. A quantidade de gerações e a população inicial foram as mesmas utilizadas para executar o CLONALG, $gen = 50$, $N = 50$. A Figura 6.5 apresenta as populações finais geradas pelo GA quando aplicado às funções $g_1(x)$ e $g_2(x)$.

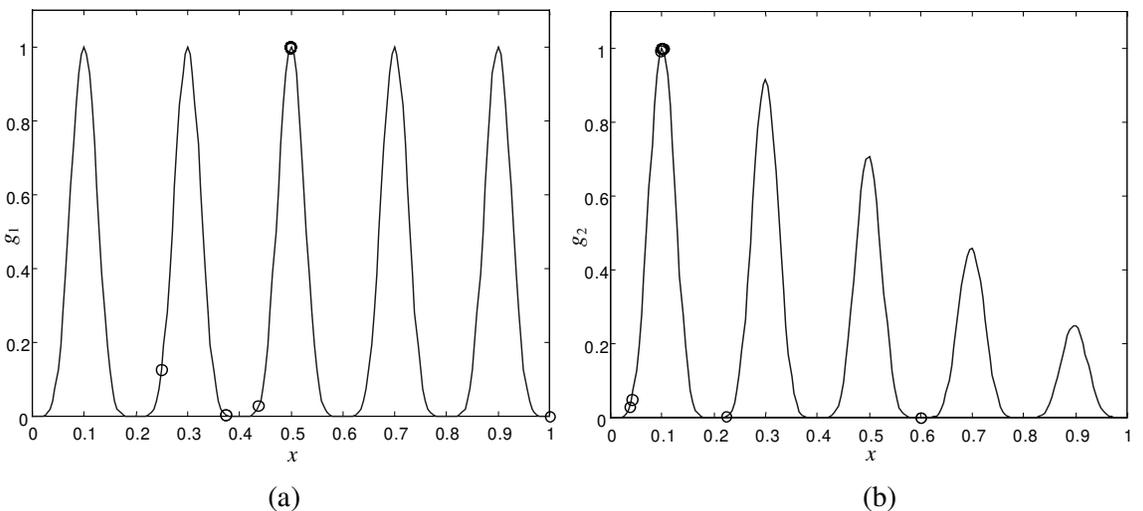


Figura 6.5. Algoritmo genético (GA) aplicado ao problema de maximização das funções: (a) $g_1(x) = \text{sen}^6(5\pi x)$ e (b) $g_2(x) = 2^{-(2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$.

Devido a capacidade do algoritmo CLONALG desempenhar uma busca multimodal, ele também foi comparado ao método de fitness sharing (GA_{sh}) apresentado na Seção 4.4.5. Utilizou-se uma medida genotípica de sharing, ou seja, para avaliar o compartilhamento no fitness de cada indivíduo da população, calculou-se a distância de Hamming entre eles (Equação 3.3) e aplicou-se a função de compartilhamento dada pelas Equações 4.16 e 4.17. Os parâmetros escolhidos para o GA com fitness sharing e seleção por torneio binário foram: $p_c = 0.6$, $p_m = 0.0$ e $\sigma_{share} = 4$. O desempenho do algoritmo está ilustrado na Figura 6.6. Com o objetivo de manter estável a população evoluída pelo algoritmo de fitness sharing quando combinado com seleção por torneio, utilizamos o método de fitness sharing contínuo, como proposto por Oei *et al.* (1991).

Comparando-se a Figura 6.4 com a Figura 6.6, verifica-se que o algoritmo CLONALG apresenta um menor número de elementos sub-ótimos do que o GA_{sh} , ou seja, a subpopulação de indivíduos ocupando cada nicho da função (ótimo local ou global) é mais uniforme, com a grande maioria dos indivíduos posicionados nos picos da curva.

Estes algoritmos também foram aplicados ao problema de maximização da função de duas variáveis $g_3(x,y) = x \cdot \text{sen}(4\pi x) - y \cdot \text{sen}(4\pi y + \pi) + 1$, ilustrada na Figura 6.7.

As variáveis x e y estão definidas no intervalo $[-1,2]$ e podem ser decodificadas como no caso de $g_1(x)$ e $g_2(x)$ adotando-se $x_{\min} = y_{\min} = -1$ e $x_{\max} = y_{\max} = 2$. Os parâmetros comuns utilizados para todos os algoritmos foram $N = 100$, $L = 22$, e os outros parâmetros adotados foram:

- CLONALG: $[\mathbf{Ab}, \mathbf{f}] = \text{clonalg}(\mathbf{Ab}, 22, 100, 100, 10, 10)$;
- GA: $p_c = 0.6$, $p_m = 0.01$ e $k = 0.6$; e
- GA_{sh} : $p_c = 0.6$, $p_m = 0.0$ e $\sigma_{share} = 4$.

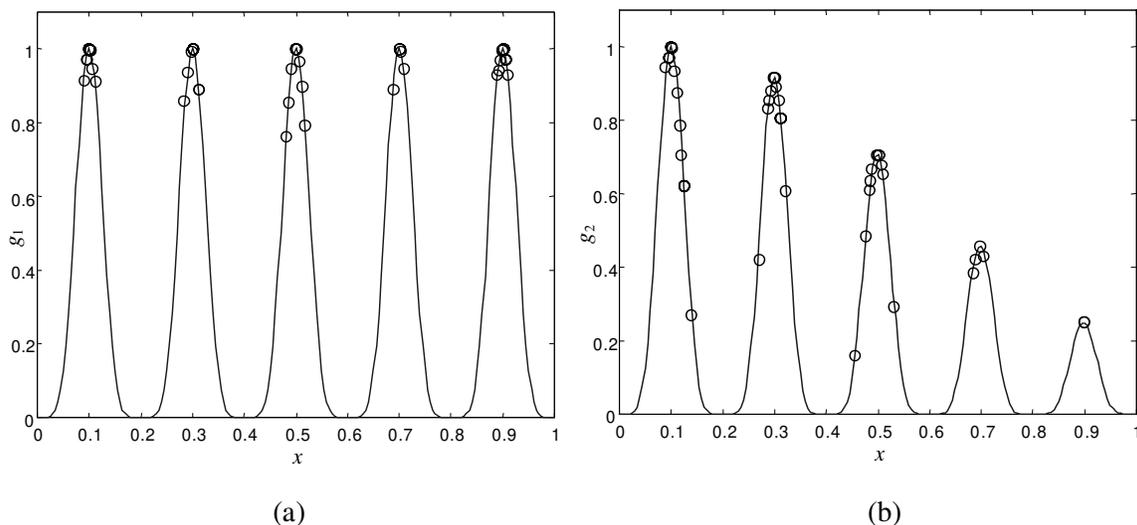


Figura 6.6. Algoritmo genético com fitness sharing (GA_{sh}) aplicado à maximização das funções: (a) $g_1(x) = \text{sen}^6(5\pi x)$ e (b) $g_2(x) = 2^{-(2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$.

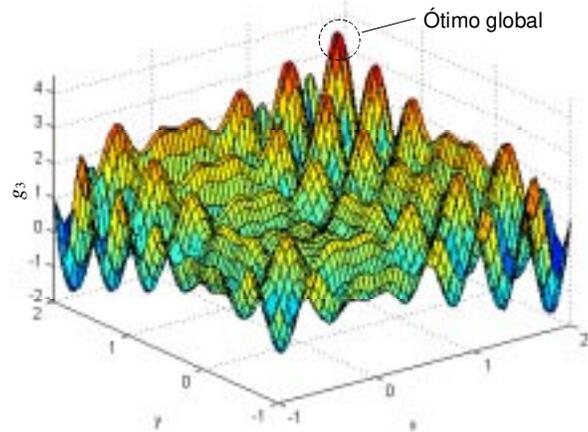


Figura 6.7. Função bidimensional $g_3(x,y) = x \cdot \text{sen}(4\pi x) - y \cdot \text{sen}(4\pi y + \pi) + 1$ a ser maximizada.

Analisando a Figura 6.8, verifica-se que o algoritmo CLONALG também é mais eficiente na localização dos ótimos locais e globais da função g_3 do que os outros métodos.

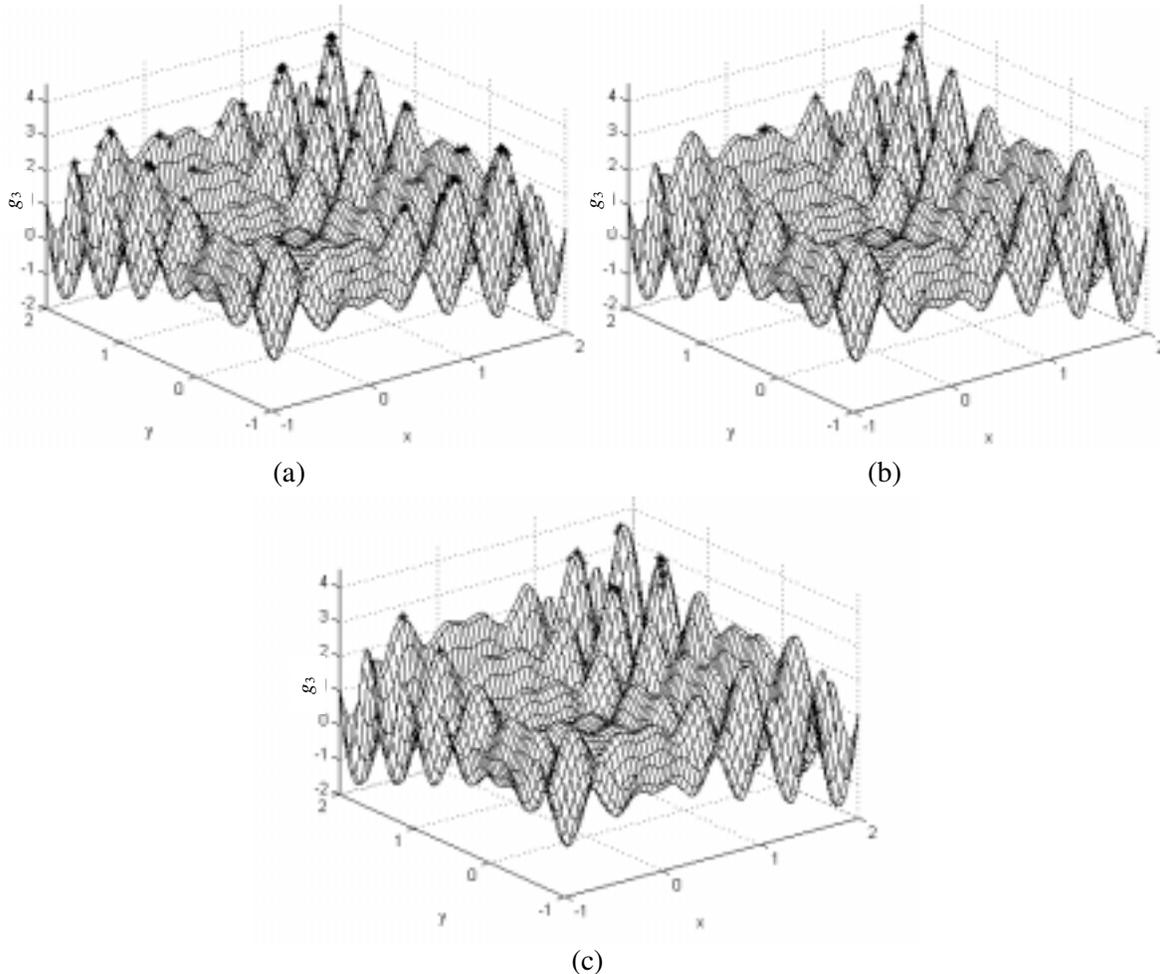


Figura 6.8. Desempenho dos algoritmos aplicados a função $g_3(x,y) = x \cdot \text{sen}(4\pi x) - y \cdot \text{sen}(4\pi y + \pi) + 1$. (a) CLONALG. (b) GA. (c) GA_{sh}.

A Figura 6.9 apresenta a evolução do fitness do melhor indivíduo (traço contínuo) e o fitness médio da população (linha tracejada) dos três algoritmos (CLONALG, GA e GA_{sh}) aplicados à função $g_1(x)$. É possível verificar pela Figura 6.9(a) que o algoritmo CLONALG possui uma seleção elitista que preserva o melhor indivíduo na população, e uma evolução estável com característica assintótica em relação aos ótimos da função. O algoritmo genético simples (Figura 6.9(b)) com seleção por torneio salvacionista apresentou uma evolução assintótica em relação a apenas um ótimo da função, como também pode ser visto na Figura 6.5. A seleção por torneio aplicada ao método de fitness sharing contínuo demonstrou instabilidade da população média e do melhor indivíduo, como pode ser observado pela Figura 6.9(c).

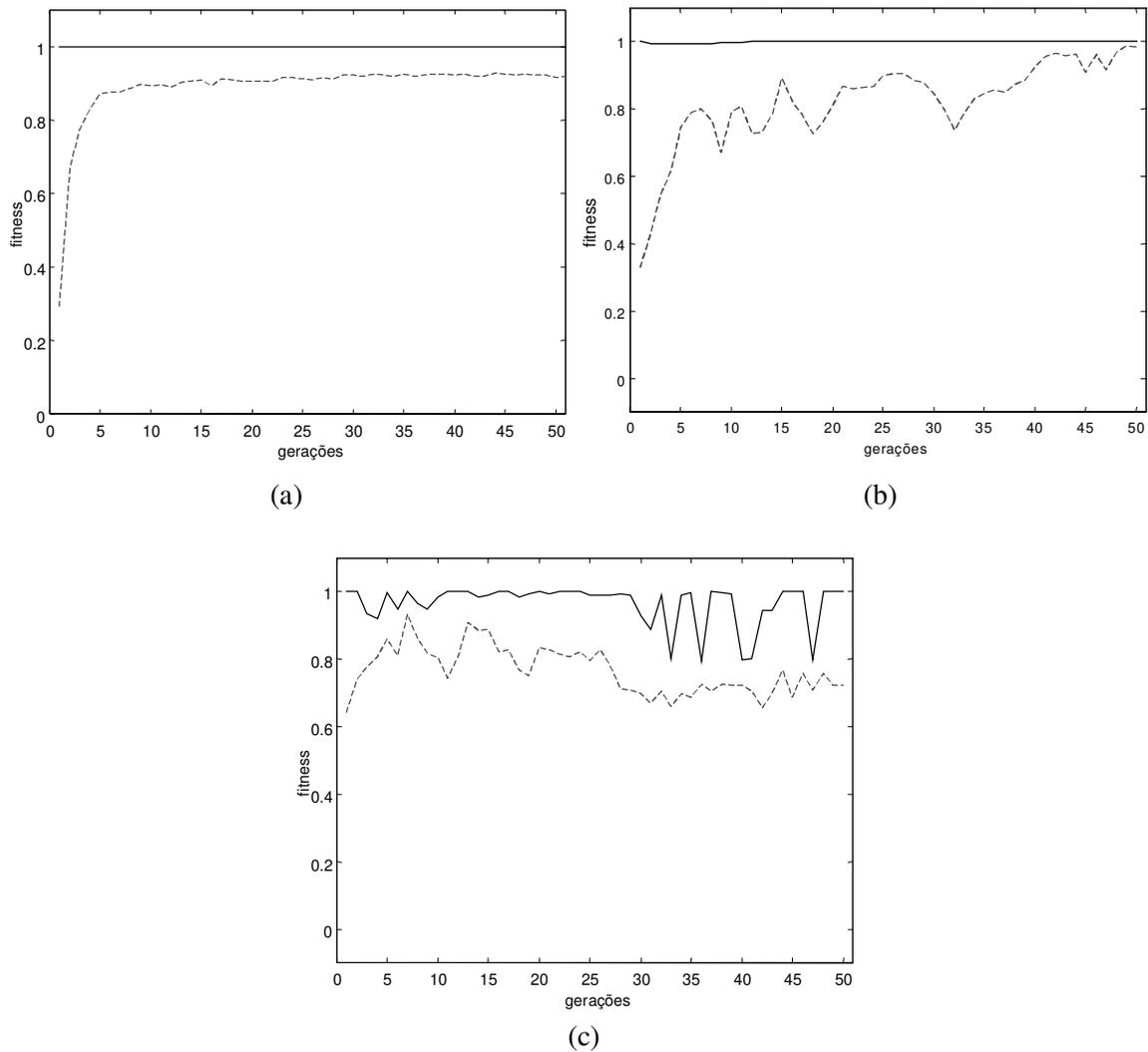


Figura 6.9. Evolução dos algoritmos CLONALG, GA e GA_{sh} aplicados ao problema de maximização da função $g_1(x) = \text{sen}^6(5\pi x)$. A curva sólida representa o fitness do melhor indivíduo da população e a curva tracejada o fitness médio da população ao longo das 50 gerações. (a) CLONALG. (b) GA. (c) GA_{sh}.

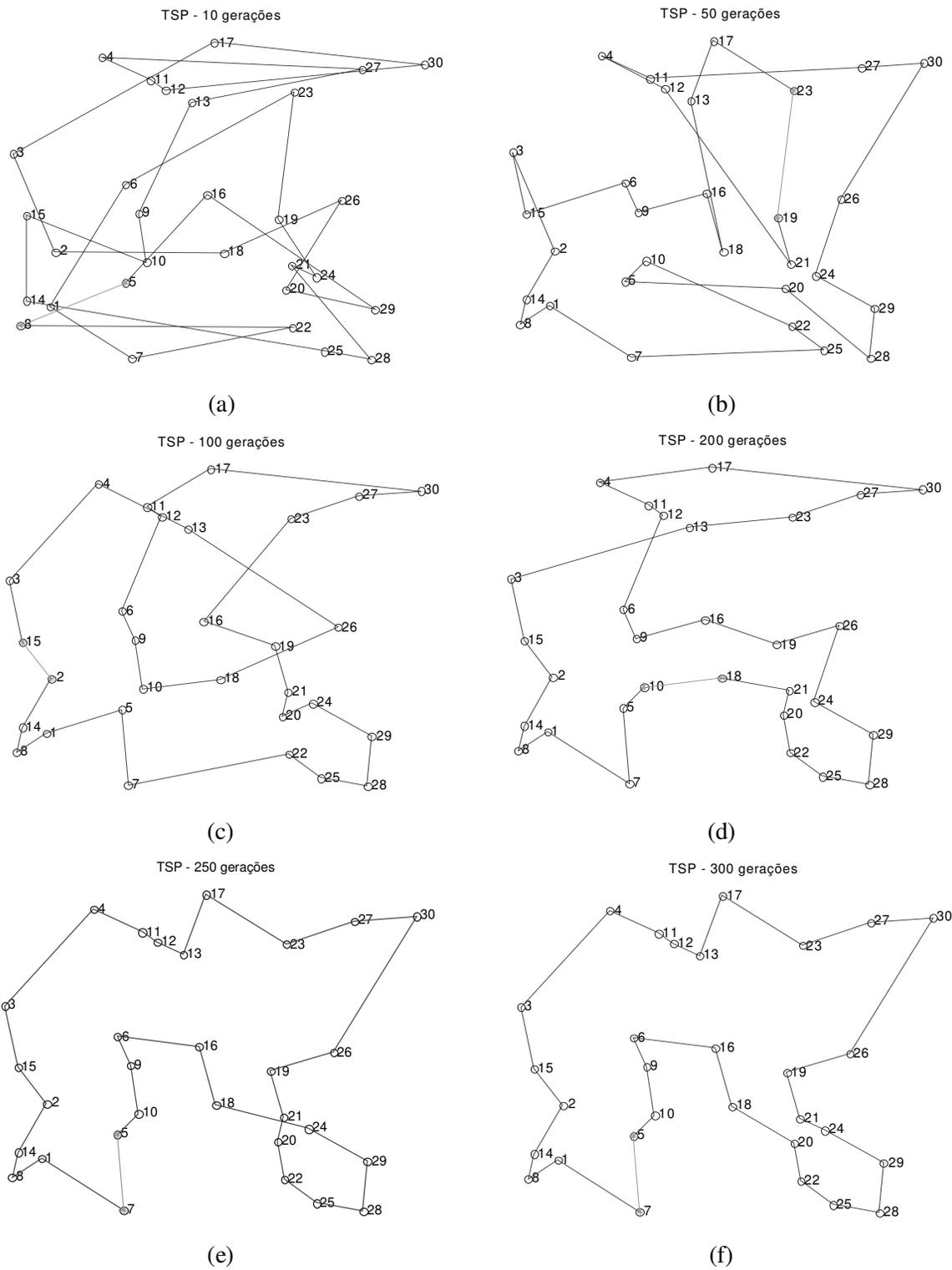


Figura 6.10. Evolução do algoritmo CLONALG (melhor indivíduo da população) aplicado ao problema TSP. Comprimento da rota (em u.m.d. – unidade de medida de distância): (a) 107932, (b) 75627, (c) 60913, (d) 54540, (e) 49433 e (f) 48872.

Para concluir a avaliação de desempenho do algoritmo CLONALG, ele foi aplicado ao *problema do caixeiro viajante* (TSP – *Travelling Salesman Problem*). De forma simples, o caixeiro viajante precisa visitar L cidades em seu território, exatamente uma única vez cada cidade, e depois retornar à cidade de partida. O problema é: dado o custo de viagem entre todos os pares de cidades, qual rota possui o menor custo? Neste trabalho, o custo de uma rota é basicamente o comprimento do caminho percorrido pelo caixeiro viajante (distância total). O problema TSP é do tipo combinatorial e aparece em diversas aplicações, de projeto de circuitos VLSI até sistemas de entregas de *fast food*. Neste caso, a utilização de um espaço de formas Inteiro é apropriada, onde vetores de comprimento L , compostos por uma permutação de elementos do conjunto $C = \{1, 2, \dots, L\}$, representam as possíveis rotas a serem percorridas. Cada componente do vetor inteiro indexa uma cidade. O inverso do comprimento total de cada rota fornece a medida de afinidade do vetor correspondente ao anticorpo.

A Figura 6.10 apresenta a evolução do melhor indivíduo da população determinado pelo algoritmo CLONALG. Após 300 gerações (Figura 6.10(f)) foi possível determinar o ótimo global do problema (Moscato & Fontanari, 1990). O tamanho da população é $N = 300$, e os parâmetros adotados foram $[\mathbf{Ab}, \mathbf{f}] = \text{clonalg}(\mathbf{Ab}, 30, 300, 150, 10, 60)$.

O algoritmo CLONALG como proposto nesta tese constitui uma implementação computacional do princípio imunológico da seleção clonal. Sendo assim, este algoritmo poderia ser melhorado para solucionar diversos problemas, como o TSP, onde a utilização de heurísticas ou mecanismos de busca local são capazes de aumentar significativamente a eficiência de algoritmo similares (como o GA) em problemas específicos.

6.2.3. Análise de Sensibilidade

Nesta seção, verificaremos a sensibilidade do algoritmo CLONALG quanto à definição dos seguintes parâmetros:

- n : quantidade de anticorpos a serem selecionados para clonagem, gerando a população $\mathbf{Ab}_{\{n\}}$;
- N_c : quantidade de clones gerada a partir de $\mathbf{Ab}_{\{n\}}$; e
- d : quantidade d de anticorpos com baixa afinidade a serem substituídos.

Sem perda de generalidade, analisaremos o problema de otimizar a função unidimensional $g_1(x) = \text{sen}^6(5\pi x)$ apresentada na Figura 6.2(a). A população terá um tamanho fixo $N = 10$, correspondente a duas vezes a quantidade de ótimos globais da função a ser maximizada, e será codificada como anteriormente. Nas Figuras 6.11 e 6.12, os resultados apresentados correspondem aos valores máximo, mínimo e médio obtidos a partir de 10 simulações. Desejamos verificar dois aspectos: 1) a capacidade do algoritmo determinar os cinco picos da função (ótimos globais), e o número de gerações necessárias para a determinação destes picos.

Inicialmente, fixamos o tamanho de cada clone gerado como sendo igual a 10 para cada um dos n indivíduos selecionados, portanto $N_c = 10n$. Feito isso, verificou-se a quantidade de gerações necessárias para a determinação de pelo menos um ótimo global da função.

Os resultados estão apresentados na Figura 6.11(a). Neste caso, verifica-se que não existe uma forte relação entre o parâmetro n e a velocidade de convergência do algoritmo, a não ser por um aumento no custo computacional de cada iteração. É importante lembrar, que quanto menor o valor de n em relação a N , menor a capacidade do algoritmo realizar uma busca multimodal, e mais seu comportamento se aproxima de um algoritmo evolutivo clássico. A Figura 6.11(b) mostra que a afinidade média da população sofre um aumento diretamente (e quase linearmente) proporcional a n .

Para analisar a sensibilidade do algoritmo em relação a N_c , a quantidade de clones gerada para cada um dos $n = N (= 10)$ anticorpos do repertório, foram assumidos os seguintes valores: $N_c = \{5, 10, 20, 40, 80, 160\}$. É possível verificar pela Figura 6.12, que quanto maior a quantidade de clones gerada para cada anticorpo, menor a quantidade de gerações necessária para a convergência do algoritmo, sendo que a convergência foi verificada quando o fitness médio da população atingiu um valor maior ou igual a 0.92.

Por sua vez, o parâmetro d possui uma importância fundamental na manutenção da diversidade da população e potencialidade de explorar novas regiões do espaço de busca. Isso foi verificado da seguinte forma. Gerou-se uma população inicial \mathbf{Ab} com $N = 10$ anticorpos iguais $\mathbf{Ab}_i = \langle 0, \dots, 0 \rangle$, $i = 1, \dots, 10$. A afinidade de todos estes anticorpos é igual a 0. Executou-se o algoritmo CLONALG para $d = 0$ (Figura 6.13(a)), $d = 1$ e $d = 2$ (Figura 6.13(b)). Como pode ser verificado, para um valor de $d > 0$, o algoritmo foi capaz de localizar todos os máximos globais da função. Por outro lado, é importante salientar que valores muito elevados deste parâmetro podem causar uma busca quase aleatória dentro do espaço de formas, sendo sugerido valores entre 5% e 20% do tamanho total N da população \mathbf{Ab} .

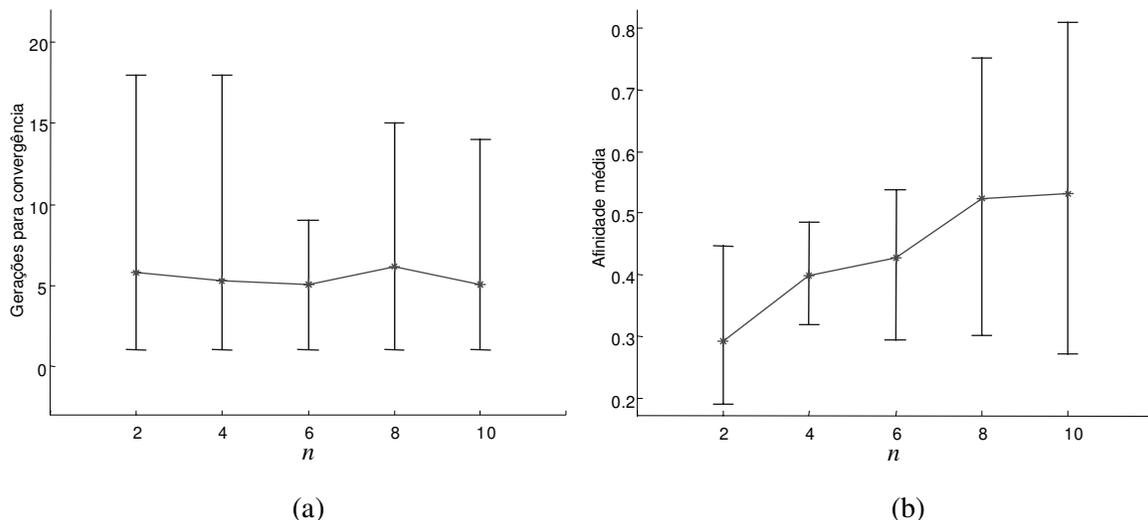


Figura 6.11. Análise de sensibilidade do algoritmo CLONALG em relação ao parâmetro n . (a) Número de gerações necessárias para a determinação de pelo menos um ótimo global da função. (b) Valor da afinidade média da população até a determinação de pelo menos um ótimo global da função.

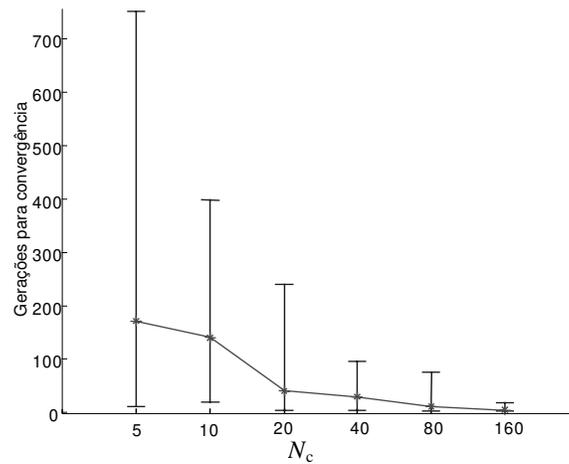


Figura 6.12. Análise de sensibilidade do algoritmo CLONALG em relação ao parâmetro N_c .

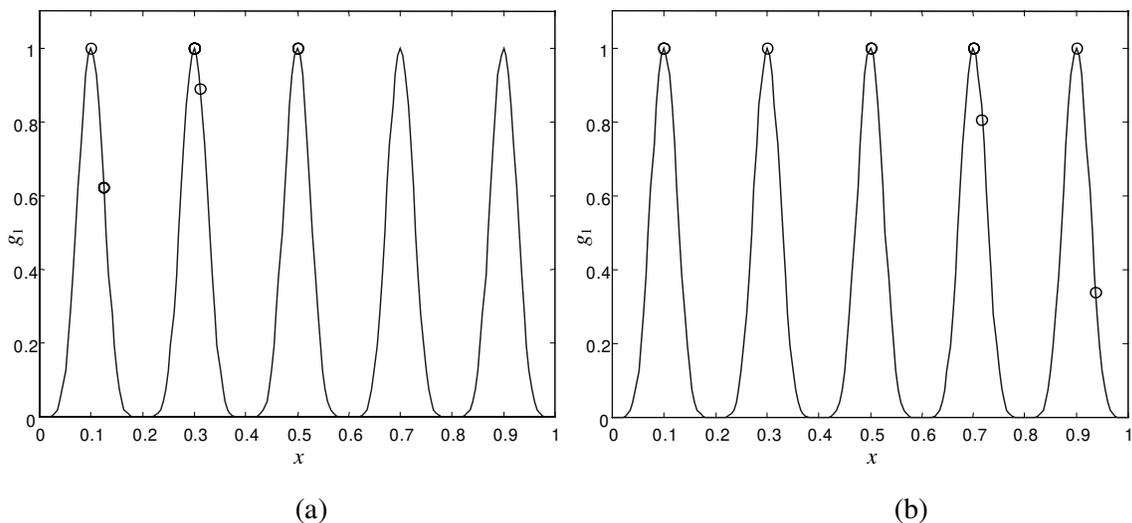


Figura 6.13. Análise de sensibilidade do algoritmo CLONALG em relação ao parâmetro d , $N = 10$. (a) $d = 0$. (b) $d = 2$, correspondendo a 20% da população. Todos os indivíduos da população inicial são idênticos, correspondendo a $x = 0$.

6.2.4. Discussão

Sabemos que o reconhecimento Ag-Ab ocorre por complementaridade de formas e é geralmente medido por um critério de dissimilaridade, como discutido na Seção 3.4. Porém, sem perda de generalidade, utilizamos uma medida de similaridade para avaliar a afinidade Ag-Ab no problema de reconhecimento de caracteres (Figura 6.1), apenas para que pudessemos representar o conjunto de memória como uma “imagem interna” dos antígenos a serem reconhecidos.

Uma outra forma de apresentar os resultados, que surtiria o mesmo efeito e estaria em maior concordância com a motivação biológica, seria utilizar uma medida de dissimilaridade, como formalmente proposto, e complementar as cadeias de atributos representativas dos anticorpos resultantes para que pudéssemos obter o efeito visual desejado. Note também, que um reconhecimento perfeito não é necessário para a correta classificação do padrão. Poderia ser empregada a idéia de reatividade cruzada introduzindo-se um limiar de afinidade ε que definiria uma região V_ε no espaço de formas de cada elemento da população, permitindo que um anticorpo pudesse reconhecer mais de um antígeno. Outra implicação direta da introdução desse limiar de afinidade seria em uma possível redução no tamanho do repertório de memória. Por exemplo, assuma que um anticorpo Ab_i possui afinidade superior a ε aos antígenos Ag_j e Ag_k , ($f_{i,j}f_{i,k} > \varepsilon$), então este anticorpo representa uma imagem interna dos antígenos j e k , e não seria necessário um anticorpo de memória específico para cada um deles.

Ao compararmos o algoritmo CLONALG aplicado a problemas de otimização multimodal com o método de fitness sharing (GA_{sh}), verificamos que o primeiro algoritmo é bastante eficiente na determinação de múltiplas soluções, enquanto o GA_{sh} possui algumas limitações:

- Ele executa a comparação de cada membro da população com todos os outros, requerendo um esforço computacional da ordem $O(N^2)$; e
- A definição do parâmetro σ_{share} é crítica e depende de uma distribuição uniforme dos picos no espaço de busca, o que pode ser verificado no caso da função $g_3(x,y)$, ilustrada na Figura 6.8(c), onde o desempenho desta estratégia foi baixo.

Verificamos que a velocidade de convergência do CLONALG é diretamente proporcional ao tamanho N_c do clone gerado a cada iteração do algoritmo. Porém existe um custo computacional associado a este parâmetro que não pode ser desconsiderado (ver Seção 5.3.4). Os resultados apresentados também mostram que o algoritmo CLONALG é capaz de gerar diversidade, através do parâmetro d (que simula a morte programada das células – anticorpos – pouco estimuladas e inserção de novos indivíduos na população) e da mutação somática, permitindo a determinação de todos os ótimos da função mesmo quando uma população inicial pequena e com baixa (ou nenhuma) diversidade é empregada.

6.3. SAND

O algoritmo de *simulated annealing* para geração de diversidade populacional (SAND), descrito na Seção 5.2, visa produzir de forma antecipatória um repertório de anticorpos que apresenta máxima cobertura do espaço de antígenos a serem reconhecidos. Esta estratégia não assume conhecimento prévio do problema a ser abordado (conjunto de antígenos) e possui aplicabilidade a problemas de reconhecimento de padrões (antígenos) e definição de condições iniciais ótimas para métodos de busca, como em algoritmos evolutivos e redes neurais artificiais.

O algoritmo SAND foi desenvolvido em duas versões: uma para espaços de formas binários de Hamming e outra para espaços Euclidianos.

Com o objetivo de avaliar o desempenho da versão para espaços de Hamming, ele será aplicado à “otimização” (maximização da diversidade) de populações de anticorpos aleatoriamente geradas. Estas populações otimizadas pelo SAND serão empregadas no reconhecimento de populações binárias de antígenos, e seus resultados comparados com a capacidade de reconhecimento apresentada pelas populações originais de anticorpos, ou seja, com as populações de anticorpos antes de serem otimizadas pelo SAND. Além disso, o algoritmo para espaços binários de Hamming também foi utilizado para inicializar os processos de busca do algoritmo CLONALG e de um algoritmo genético (GA) clássico, aplicados a problemas de otimização multimodal e global, respectivamente.

No caso de espaços de formas Euclidianos, o SAND foi aplicado ao problema de inicialização de redes neurais artificiais do tipo perceptron de múltiplas camadas (MLP) treinadas via um algoritmo de otimização não-linear irrestrita de segunda ordem. O método SAND para inicialização de redes MLP será aplicado a diversos problemas de teste e de mundo real e seu desempenho comparado com o de várias outras estratégias propostas na literatura.

6.3.1. Espaço de Formas Binário de Hamming

Serão estudados dois problemas: o reconhecimento de populações aleatórias de antígenos e a inicialização de métodos de busca exploratória.

6.3.1.1. Reconhecimento de Antígenos

O aumento na cobertura do espaço de formas binário de Hamming pode ser verificado através da aplicação do algoritmo SAND no reconhecimento de populações binárias de antígenos.

Tabela 6.1. Medida de energia (Equação 5.5) de populações aleatoriamente geradas (P_{RP}) versus a energia de populações evoluídas pelo algoritmo SAND (P_{SAND}). N é o tamanho da população e L o comprimento das cadeias binárias. Os resultados são o máximo, mínimo, média e desvio padrão obtidos a partir de 20 simulações.

L	2^L	N	P_{RP} (%)				P_{SAND} (%)			
			<i>Max</i>	<i>Min</i>	<i>Média</i>	δ	<i>Max</i>	<i>Min</i>	<i>Média</i>	δ
3	8	8	81.25	50.00	68.12	9.06	100	100	100	0
4	16	16	78.12	62.50	69.06	4.76	100	100	100	0
5	32	32	73.44	62.50	68.28	3.61	100	100	100	0
6	64	64	73.44	64.06	69.45	2.91	100	100	100	0
7	128	128	71.48	66.80	68.32	1.46	100	100	100	0
8	256	128	71.68	66.21	69.30	1.62	100	100	100	0
9	512	256	68.18	63.48	65.38	1.26	100	100	100	0

A Tabela 6.1 apresenta uma comparação da energia da população (Equação 5.5) considerando populações aleatoriamente geradas (P_{RP}) e populações evoluídas pelo algoritmo P_{SAND} . Os resultados apresentados são o máximo, mínimo, média e desvio padrão obtidos a partir de 20 simulações. Pode ser observado que o SAND é capaz de maximizar a energia das populações geradas aleatoriamente, ou seja, ele é capaz de maximizar a cobertura do espaço de buscas, ou a diversidade populacional.

Considerando o algoritmo apresentado na Seção 5.2.3.2, a seguinte linha de comando foi utilizada: $[Ab] = sand(N, L, 100 \cdot N, 0.8, 3, 100)$. O comprimento L das cadeias binárias assumiu os valores $L = \{3, 4, 5, 6, 7, 8, 9\}$ e a quantidade de anticorpos $N = \{8, 16, 32, 64, 128, 256\}$, respectivamente.

É possível concluir, pela Tabela 6.1, que se uma população Ag de antígenos fosse gerada aleatoriamente com cadeias binárias de comprimento $L = 3, 4, 5, 6, 7, 8$ e 9 , as populações de anticorpos evoluídos seriam capazes de se ligar a qualquer antígeno do conjunto Ag para qualquer valor do limiar de afinidade ϵ , indicando uma ligação perfeita (Figura 6.14(a)). Neste caso, um valor de $\epsilon > 1$ implica que haverá anticorpos na interseção dos conjuntos definidos por cada um deles, gerando anticorpos com reatividade cruzada que podem se ligar a mais do que um antígeno, iniciando uma resposta reativa cruzada (Figura 6.14(b)). Se o repertório de anticorpos tivesse sido gerado aleatoriamente, seria necessário um limiar de afinidade $\epsilon > 0$ para que alguns dos antígenos fossem reconhecidos, indicando que a população gerada aleatoriamente não apresenta uma máxima cobertura do espaço de formas (Figura 6.14(c)).

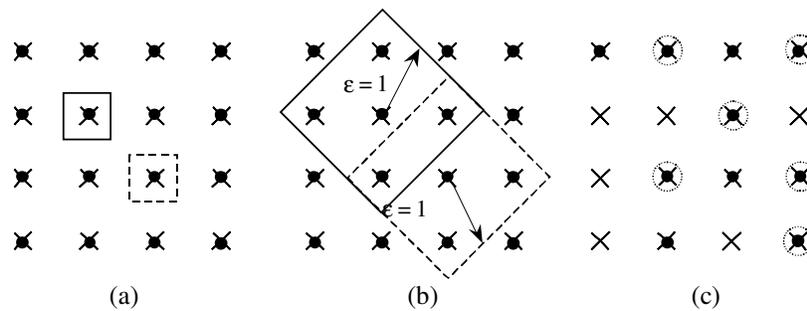


Figura 6.14. Idéia ilustrativa de cobertura do espaço de formas, onde (x) corresponde aos antígenos que representam todo o espaço de formas, e (•) são os anticorpos. (a) Existem anticorpos suficientes para cobrir todo o universo de antígenos. O algoritmo SAND evoluiu uma população capaz de apresentar a máxima cobertura do espaço, sem permitir que um anticorpo reconheça outro anticorpo do repertório ($\epsilon < 1$). Existe uma superposição entre todos os antígenos e os anticorpos. (b) Além da superposição Ag-Ab, para $\epsilon > 1$, interseções dos conjuntos de anticorpos estimulados pelo antígeno ocorrerão. (c) A população aleatoriamente gerada pode conter anticorpos suficientes para cobrir todo o espaço de formas ($N = 16$), porém existe a superposição de alguns deles (representada pelos círculos em torno de • e x).

Dada uma cadeia binária de comprimento L , para que seja possível medir a reatividade cruzada (capacidade de generalização ou cobertura do espaço de formas) dos repertórios de anticorpos evoluídos e aleatoriamente gerados, foi adotado o seguinte procedimento. De todos os $M = 2^L$ possíveis antígenos, uma quantidade N_{Ag} desta população foi aleatoriamente escolhida. N anticorpos também foram aleatoriamente gerados, compondo uma população denominada P_{RP} , e seguidamente evoluídos pelo algoritmo SAND, compondo uma nova população denominada P_{SAND} .

Tabela 6.2. Reatividade cruzada. Percentual de antígenos reconhecidos corretamente pelas populações P_{RP} e P_{SAND} . Valores máximo, mínimo, média e desvio padrão obtidos a partir de 10 simulações para $L = 9$ e $N_{Ag} = 50$.

N	ε	$P_{RP}(\%)$				$P_{SAND}(\%)$			
		<i>Max</i>	<i>Min</i>	<i>Média</i>	δ	<i>Max</i>	<i>Min</i>	<i>Média</i>	δ
5	0	2	0	0.4	0.84	2	0	1.2	1.03
	1	16	4	8.2	3.33	16	6	10.8	3.91
	2	46	30	37.4	5.97	52	30	40.2	6.21
	3	86	64	77.6	7.82	92	78	86.2	4.94
10	0	4	0	1.4	1.35	6	0	2.8	1.69
	1	20	8	15.8	3.33	22	6	17	5.01
	2	68	58	64	3.65	76	54	67.4	7.24
	3	100	86	94.2	4.16	100	96	98.8	1.69
25	0	12	2	4.6	3.41	10	2	5.2	2.35
	1	54	32	39.6	6.10	60	36	46.8	7.73
	2	98	86	92.6	3.78	98	86	93.6	4.50
	3	100	100	100	0	100	100	100	0
50	0	14	2	8.2	3.94	16	4	8.4	4.20
	1	72	54	62.8	5.59	74	56	64	5.08
	2	100	96	99.2	1.40	100	98	99.8	0.63
	3	100	100	100	0	100	100	100	0
75	0	22	4	12.2	4.66	24	8	14.8	4.34
	1	84	68	76.8	5.90	88	78	81.4	2.99
	2	100	98	99.8	0.63	100	100	100	0
	3	100	100	100	0	100	100	100	0
100	0	26	12	18	4.62	30	12	20.8	5.09
	1	92	76	86.4	4.50	98	80	90.6	5.42
	2	100	100	100	0	100	100	100	0
	3	100	100	100	0	100	100	100	0

Para diversos valores de ε e N , o percentual de reconhecimento correto das populações de antígenos pelos repertórios P_{RP} e P_{SAND} foi avaliado, onde o reconhecimento ocorre caso, para cada antígeno i do conjunto \mathbf{Ag} , exista pelo menos um anticorpo j do conjunto \mathbf{Ab} cuja distância de Hamming ($\mathbf{Ag}_i - \mathbf{Ab}_j$) $< \varepsilon$.

A Tabela 6.2 apresenta os valores máximo, mínimo, médio e o desvio padrão do percentual de antígenos reconhecidos corretamente para 10 simulações, assumindo $L = 9$ e $N_{Ag} = 50$. Como pode ser observado por esta tabela, as populações evoluídas pelo SAND apresentam um percentual de reconhecimento maior do que aquelas aleatoriamente geradas. Neste caso, um antígeno é considerado reconhecido corretamente quando a distância de Hamming (Equação 3.3) entre este antígeno e pelo menos um anticorpo do repertório imunológico for menor ou igual a ε , ou seja, $D \leq \varepsilon$. A linha de comando utilizada foi igual à anterior para $L = 9$.

6.3.1.2. Inicialização de Algoritmos Evolutivos

A inicialização da população a ser empregada nos algoritmos evolutivos pode influenciar o resultado obtido ao final do processo de busca. Por exemplo, se considerarmos problemas de busca multimodal, a população inicial poderá determinar a quantidade de ótimos locais e globais obtidos. Para os casos em que apenas o ótimo global da função é desejado, a população inicial poderá aumentar ou não a probabilidade de determinação deste ótimo.

Na Seção 6.2.3, foi feita uma análise de sensibilidade do algoritmo CLONALG em relação à diversos parâmetros definidos pelo usuário. Foi verificado que, para $d = 0$, a capacidade deste algoritmo executar uma busca multimodal fica reduzida, principalmente quando as populações iniciais não são inicializadas apropriadamente (Figura 6.13). Sendo assim, faz-se necessário que a população inicial de candidatos à solução esteja bem distribuída no espaço de formas, de modo que o algoritmo possa determinar a maior quantidade de ótimos locais da função a ser otimizada. Esta característica também é importante quando algoritmos de otimização local como métodos de gradiente são empregados, pois estes métodos geralmente assumem, a priori, que a função a ser otimizada é unimodal.

Para avaliar a influência da condição inicial na determinação da maior quantidade de ótimos de uma função, aplicamos o algoritmo CLONALG na otimização da função $g_1(x) = \text{sen}^6(5\pi x)$, apresentada na Figura 6.2(a). Para populações com cardinalidades distintas, verificou-se a quantidade de ótimos globais obtidos pelo CLONALG quando inicializado com uma população aleatória P_{RP} e quando inicializado pelo algoritmo SAND (P_{SAND}). Sabendo que esta função possui 5 ótimos, todos globais, a Tabela 6.3 apresenta a quantidade de máximos, tomados a partir de 10 simulações, determinados pelo algoritmo CLONALG para os 2 métodos de inicialização. É possível perceber que, mesmo para populações de dimensões reduzidas, como $N = 5$, o algoritmo proposto é capaz de proporcionar um aumento significativo na quantidade de ótimos determinados. Foi utilizada a mesma linha de comando anterior para cadeias de comprimento $L = 22$, como descrito na Seção 6.2.2: `[Ab, f]=clonalg(Ab, 22, 50, 50, 10, 0)`.

Tabela 6.3. Quantidade de ótimos globais da função $g_1(x) = \text{sen}^6(5\pi x)$ determinados pelo algoritmo CLONALG quando a população inicial é gerada aleatoriamente (P_{RP}) e quando ela é gerada pelo algoritmo SAND (P_{SAND}).

N	P_{RP}				P_{SAND}			
	<i>Max</i>	<i>Min</i>	<i>Média</i>	δ	<i>Max</i>	<i>Min</i>	<i>Média</i>	δ
5	4	1	2.9	0.88	5	3	3.8	0.63
6	5	2	3.5	0.85	5	2	3.8	0.92
7	5	3	4.1	0.57	5	3	4.4	0.70
8	5	3	3.8	0.63	5	3	4.5	0.71
9	5	3	4.2	0.79	5	4	4.6	0.52
10	5	3	4.2	0.63	5	4	4.7	0.48
11	5	4	4.5	0.53	5	4	4.8	0.42
12	5	3	4.3	0.67	5	5	5	0

Como último exemplo de aplicação do SAND para inicialização de algoritmos evolutivos, seja um algoritmo genético (GA) com seleção bi-classista, crossover simples e mutação de múltiplos pontos, aplicado à otimização da função $g_2(x) = 2^{(-2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$, ilustrada na Figura 6.2(b). Os parâmetros escolhidos para a evolução do GA foram $gen = 100$, $p_c = 0.6$ e $p_m = 0.01$. Neste caso, deseja-se verificar quantas vezes, tomadas a partir de 10 simulações, o GA é capaz de determinar o ótimo global desta função correspondente a um valor de fitness do melhor indivíduo maior do que 0.97, sabendo que o ótimo global desta função corresponde a 1.0. A Figura 6.15 faz uma comparação do percentual de determinação do ótimo global da função $g_2(x) = 2^{(-2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$ para populações inicializadas aleatoriamente (P_{RP}) e via o algoritmo SAND (P_{SAND}).

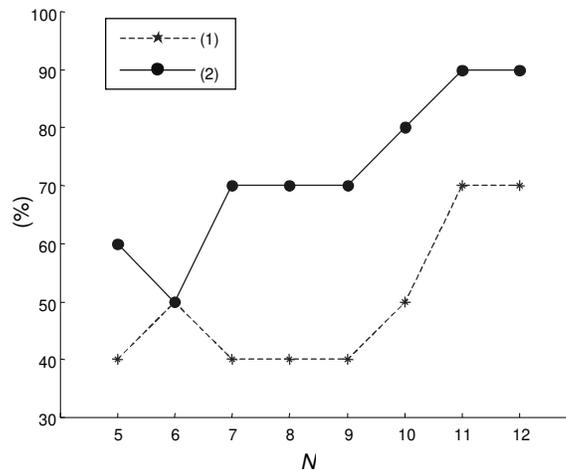


Figura 6.15. Percentual de convergência, a partir de 10 simulações, de um GA aplicado à maximização da função $g_2(x) = 2^{(-2((x-0.1)/0.9)^2)} \text{sen}^6(5\pi x)$ para populações de candidatos à solução de diferentes tamanhos N . P_{RP} (1) e P_{SAND} (2).

6.3.2. Espaço de Formas Euclidianos

Para avaliar o desempenho da versão projetada para espaços de formas Euclidianos do algoritmo SAND, ele será aplicado ao problema de inicialização dos vetores de pesos a serem utilizados no treinamento de redes neurais artificiais do tipo perceptron de múltiplas camadas (MLP) com treinamento via algoritmos de backpropagation. Este conjunto inicial de pesos possui grande influência na velocidade de treinamento e qualidade da solução obtida após a convergência da rede. Uma escolha inadequada deste conjunto, a ser expresso na forma de um vetor pode fazer com que o algoritmo de treinamento fique preso em um mínimo local insatisfatório, ou que o algoritmo apresente problemas de instabilidade numérica.

Cada anticorpo evoluído pelo SAND corresponde a um vetor contendo os pesos de um determinado neurônio em uma das camadas da rede MLP. Assim, gerar uma população de anticorpos com máxima diversidade em S^L corresponde a produzir um conjunto de neurônios com vetores de pesos bem distribuídos em \mathfrak{R}^L . O algoritmo SAND para espaços de forma Euclidianos (Seção 5.2.3.3) será aplicado separadamente a cada camada da rede. Um aspecto importante desta estratégia é o fato de que, como estamos gerando vetores com normas unitárias, estes vetores podem ser escalonados (normalizados) de forma que a ativação de cada neurônio ocorra dentro da região aproximadamente linear de suas funções de ativação, evitando a saturação durante o processo de inicialização e treinamento da rede.

O método SAND para inicialização de redes MLP foi comparado com 5 outras estratégias e aplicado a 7 problemas de teste. Os métodos cujos desempenhos foram comparados são: BOERS (Boers & Kuiper, 1992), WIDROW (Nguyen & Widrow, 1990), KIM (Kim & Ra, 1991), OLS (Lehtokangas *et al.*, 1995), INIT (de Castro & Von Zuben, 1998a,b), e SAND (de Castro & Von Zuben, 2001c).

Para especificar os problemas de teste utilizados, seja M a quantidade de amostras (antígenos), SSE a soma desejada dos erros quadráticos (critério de parada da rede MLP) e net a arquitetura da rede representada por $[n_i-n_h-n_o]$, onde n_i é o número de entradas da rede, n_h a quantidade de unidades intermediárias e n_o o número de saídas. Os problemas tomados para comparação foram:

- problema de paridade 2 (XOR): $M = 4$, net : [2-2-1], SSE = 0.01;
- problema de paridade 3: $M = 8$, net : [3-3-1], SSE = 0.01;
- aproximação da função $\sin(x).\cos(2x)$: $M = 25$, net : [1-10-1], SSE = 0.01;
- ESP: problema de mundo real utilizado por Barreiros *et. al.* (1997); $M = 75$, net : [3-10-5], SSE = 0.1;
- SOYA: problema de mundo real utilizado por de Castro *et al.* (1998b), $M = 116$, net : [36-10-1], SSE = 0.1;
- IRIS: classificação de tipos de plantas do gênero Iris disponível em (URL 3); $M = 150$, net : [4-10-3], SSE = 0.15; e
- ENC/DEC: problema de codificação/decodificação descrito por Fahlman (1988); $M = 10$, net : [10-7-10].

Tabela 6.4. Resultados de simulação para inicialização de redes neurais. Os valores apresentados são o máximo, mínimo, média e desvio padrão (δ) do número de épocas para convergência dos algoritmos para os valores do SSE pré-definidos, a partir de 10 simulações.

Problema	Método	Max	Min	Média	δ
XOR	BOERS	129	6	42.90	48.32
	WIDROW	93	8	20.50	25.92
	KIM	84	6	32.50	25.91
	OLS	166	5	46.70	46.50
	SAND	76	7	27.11	28.61
	INIT	47	8	18.60	12.76
Paridade 3	BOERS	62	10	25.10	16.01
	WIDROW	25	10	19.60	5.17
	KIM	46	10	18.90	10.41
	OLS	148	29	65.70	36.16
	SAND	19	14	16.00	2.00
	INIT	24	8	16.00	4.97
$\sin(x).\cos(2x)$	BOERS	187	79	135.10	35.39
	WIDROW	243	123	178.60	41.27
	KIM	231	95	164.50	44.89
	OLS	133	39	91.40	33.06
	SAND	165	84	127.30	30.43
	INIT	449	181	254.80	80.69
ESP	BOERS	1618	340	883.40	467.52
	WIDROW	2280	236	825.40	639.74
	KIM	1763	425	715.40	397.82
	OLS	2052	35	545.42	586.62
	SAND	766	385	551.50	136.12
	INIT	762	383	479.20	118.19
SOYA	BOERS	174	136	158.60	13.14
	WIDROW	464	176	266.30	79.27
	KIM	280	177	219.40	28.34
	OLS	311	155	220.44	52.53
	SAND	185	125	147.20	17.50
	INIT	236	136	188.00	29.30
IRIS	BOERS	1568	735	1102.40	281.33
	WIDROW	1240	676	918.60	183.72
	KIM	2063	767	1294.80	438.34
	OLS	5000	2075	4140.20	1384.74
	SAND	1169	688	852.80	169.71
	INIT	1407	662	869.80	216.73

Tabela 6.5. Capacidade de convergência dos algoritmos para o problema ENC/DEC. Os valores apresentados incluem o máximo, mínimo, média e desvio padrão (δ) do número de épocas para convergência dos algoritmos para os valores do SSE pré-definidos, a partir de 20 simulações

Problema	Método	Max	Min	Mean	δ	Convergência
ENC/DEC	BOERS	40	12	22.21	7.94	14/20
	WIDROW	32	7	12.36	6.81	14/20
	KIM	298	10	38.17	69.07	18/20
	OLS	276	7	97.78	102.03	9/20
	SAND	14	7	10.00	2.24	19/20
	INIT	16	6	9.44	2.53	16/20

O algoritmo de treinamento empregado foi o gradiente conjugado escalonado de Moller (1993), com cálculo exato da informação de segunda ordem, como proposto por Pearlmutter (1994). Para cada algoritmo e cada problema testados, foram efetuadas 10 simulações. Os resultados apresentados na Tabela 6.4 são o máximo, mínimo, média e desvio padrão (δ) da quantidade de épocas para convergência dos algoritmos para os valores do SSE pré-definidos.

Para os espaços Euclidianos, é suposto que as matrizes \mathbf{Ab} , contendo os vetores de pesos de cada camada, são fornecidas ao algoritmo e, portanto, a linha de comando do SAND passa a ser $[\mathbf{Ab}] = \text{sand}(\mathbf{Ab}, \text{gen}, \beta, \delta, \text{mE})$. Os parâmetros $\text{gen} = 5000$, $\beta = 0.9$, $\delta = 5$ e $\text{mE} = 0.995$ foram adotados em todas as simulações.

Para o problema ENC/DEC, os algoritmos não foram capazes de convergir em todos os casos. Sendo assim, para apresentar os resultados foram executadas 20 simulações e listada a quantidade de sucessos (convergência) obtida por cada algoritmo (Tabela 6.5), juntamente com os valores máximo, mínimo, médio e desvio padrão dos casos em que as redes convergiram.

Pode ser observado, nas Tabelas 6.4 e 6.5, que o SAND apresentou bons resultados quando comparado às outras abordagens avaliadas. É importante salientar que todos os métodos devem apresentar desempenhos superiores a uma inicialização aleatória, cujos resultados não foram mostrados. A Figura 6.16 faz uma comparação de desempenho dos algoritmos avaliados. Um método é considerado superior a outro quando ele apresenta o menor valor máximo, mínimo, média e desvio padrão do número de épocas para convergência. Os algoritmos OLS, SAND e INIT foram superiores aos outros em praticamente todos os casos.

Resultados empíricos demonstraram a necessidade de re-escalonar os vetores de pesos gerados pelo algoritmo SAND quando aplicado a problemas de mundo real (ESP, SOYA e IRIS). O procedimento de re-escalonamento tende a evitar que os neurônios saturem durante o treinamento. Mesmo assim, a complexidade computacional associada à implementação do SAND é inferior àquela produzida, por exemplo, pelos algoritmos OLS e INIT.

Durante o projeto de uma rede neural, geralmente desejamos que seu desempenho seja o melhor possível para dados não utilizados no seu treinamento, ou seja, tem-se por objetivo que a RNA treinada apresente um boa capacidade de generalização. Entretanto, todas as arquiteturas neurais básicas como o MLP estão sujeitas a um sobre-treinamento (Prechelt, 1998): enquanto a rede parece estar aprendendo a representar o problema cada vez melhor, ou seja, o erro para o conjunto de treinamento decresce, em algum ponto do processo de treinamento o seu desempenho de teste está se deteriorando, ou seja, o erro para amostras não utilizadas no treinamento está aumentando. Existem duas abordagens básicas para evitar, ou aliviar, o sobre-treinamento (Sarle, 1995; Prechelt, 1998): 1) reduzir a dimensão do espaço de parâmetros (dimensão da rede) ou 2) reduzir o número efetivo de parâmetros. Métodos aplicados à redução do espaço de parâmetros são as estratégias construtivas e de poda (Kwok & Yeung, 1997; Reed, 1993), enquanto os métodos utilizados para reduzir o número efetivo de parâmetros são as técnicas de regularização e validação cruzada (Prechelt, 1998).

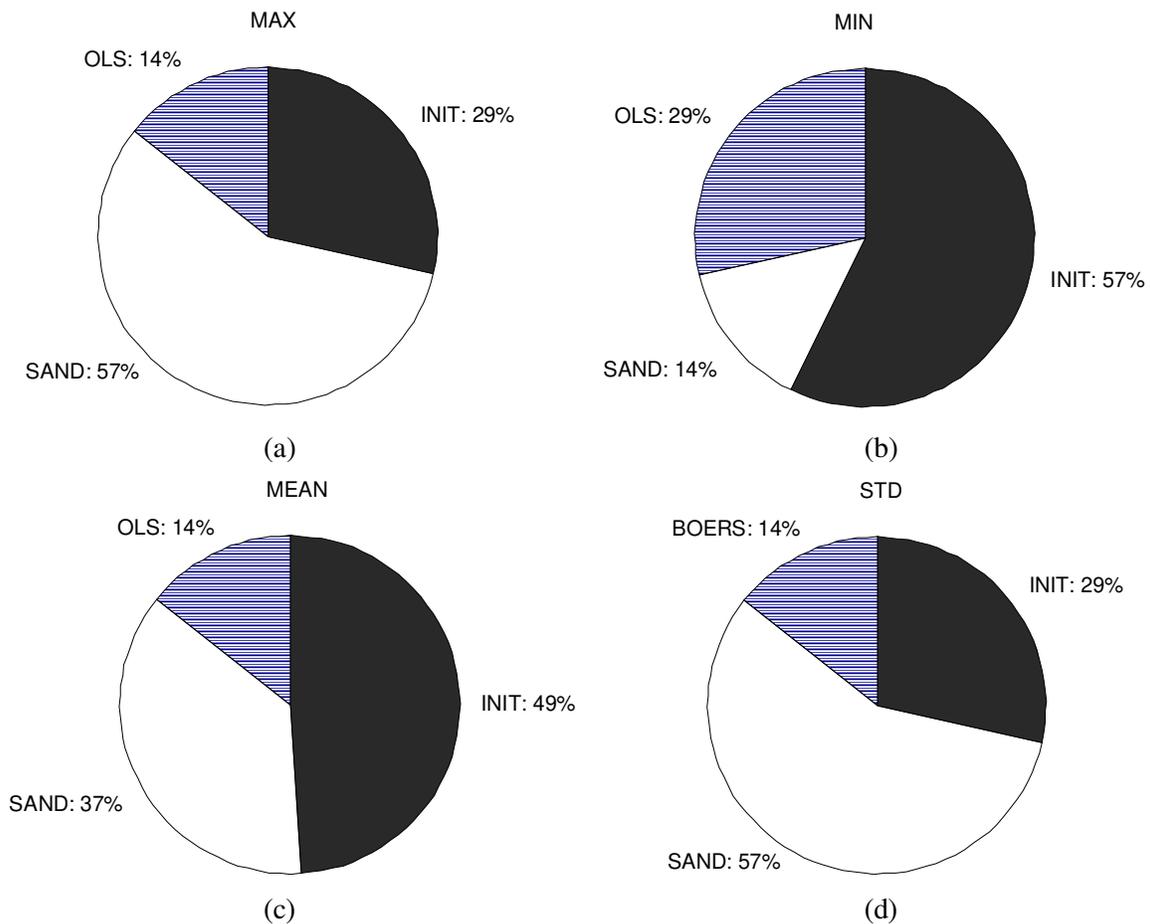


Figura 6.16. Comparação de desempenho de todos os métodos. (a,d) Em 57% dos casos, o algoritmo SAND necessitou do menor número máximo de épocas para convergência e apresentou o menor desvio padrão. (b) Em 14% dos casos, o SAND apresentou o número mínimo de épocas para convergência. (c) Em 37% dos casos, o SAND apresentou o menor número médio de épocas para convergência.

O objetivo não é o de aplicar nenhuma destas estratégias para melhorar a capacidade de generalização das redes, mas apenas o de avaliar a influência de cada método de inicialização na capacidade de generalização das redes neurais treinadas. Para que isso seja feito, todo o conjunto amostral será particionado em dois subconjuntos independentes: um conjunto de treinamento (P) e um conjunto de teste (X). A rede será treinada utilizando o conjunto P e seu desempenho avaliado para o conjunto X. Para cada problema, 10 partições distintas serão feitas considerando 90% das M amostras para treinamento da rede, e os 10% restantes para teste. Os valores máximo, mínimo e a média do SSE para o conjunto de teste estão apresentados na Figura 6.17. Os problemas avaliados foram apenas os problemas de mundo real, ou seja, ESP, SOYA e IRIS.

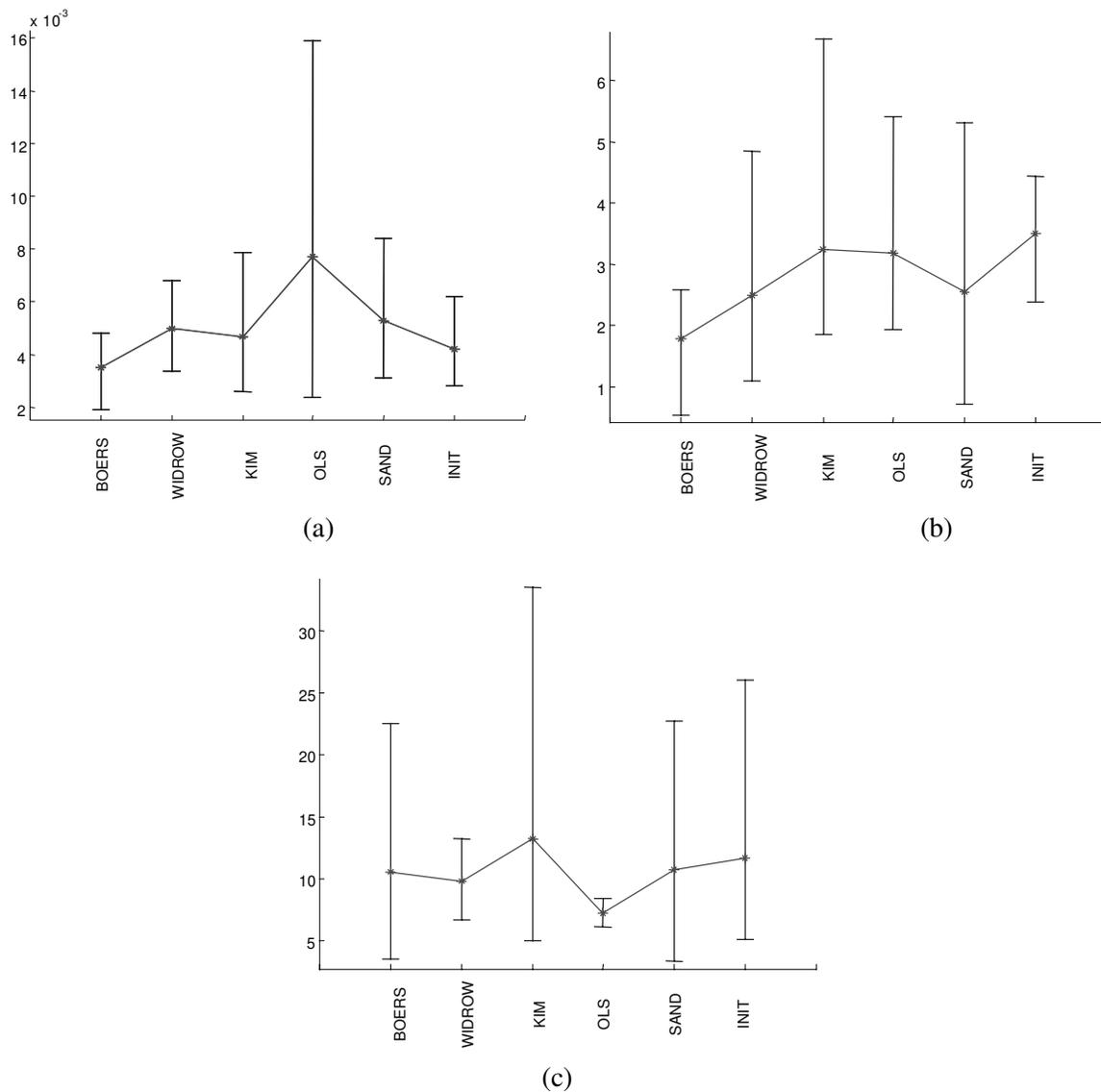


Figura 6.17. Capacidade de generalização dos métodos avaliada para os 3 problemas de mundo real. São apresentados os valores máximo, mínimo e médio (de 10 simulações) do SSE para os conjuntos de teste. (a) ESP. (b) SOYA. (c) IRIS.

Dos resultados apresentados na Figura 6.17, pode-se concluir que nenhum algoritmo se mostrou superior aos demais em termos de capacidade de generalização.

6.3.3. Análise de Sensibilidade

A linha de comando proposta para o algoritmo SAND recebe como entradas os seguintes parâmetros: N , L , gen , β , δ e mE . N e L irão depender do problema a ser tratado, ou de uma escolha do usuário. Como a função de energia proposta, tanto para o espaço de formas de Hamming quanto para o Euclidiano, assume valores de no máximo 100%, o critério de parada mE está bem definido (qualquer valor próximo ou igual a 100), fazendo com que a quantidade gen de gerações possa ser arbitrada de forma a permitir que o algoritmo atinja a diversidade desejada. Portanto, os únicos parâmetros que precisam de uma análise mais criteriosa são:

- β : taxa de decrescimento geométrico da temperatura; e
- δ : limiar de estado estacionário para redução da temperatura.

Estes parâmetros influenciam principalmente na velocidade de convergência do algoritmo. Para avaliar esta influência, foi feito o seguinte experimento: dada uma população binária de $N=16$ anticorpos de comprimento $L=5$, verificou-se a quantidade de gerações necessárias para a convergência do algoritmo para uma diversidade máxima da população ($mE = 100$). A Figura 6.18 apresenta a quantidade média de gerações para convergência do SAND, tomadas a partir de 10 simulações. É possível verificar que, para valores pequenos de δ e β , o algoritmo requer um menor número de iterações para convergência.

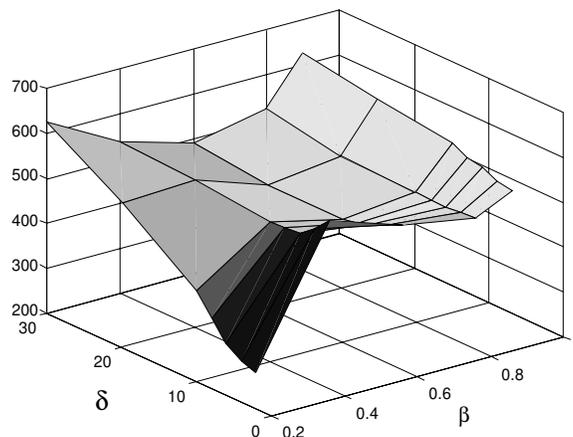


Figura 6.18. Sensibilidade do algoritmo SAND aos parâmetros β e δ . Valores médios da quantidade de gerações para convergência: $mE = 100\%$.

6.3.4. Discussão

Ao comparar a capacidade de reconhecimento antigênico por populações de anticorpos aleatoriamente geradas e populações otimizadas pelo algoritmo SAND, é possível perceber que o método proposto é capaz de gerar repertórios de anticorpos com uma maior cobertura do espaço de formas. O mesmo ocorre quando analisamos a reatividade cruzada (capacidade de generalização) das populações evoluídas.

Esta estratégia demonstrou ser interessante não apenas para o problema de reconhecimento de padrões, mas também para a inicialização de estratégias de busca, incluindo o CLONALG proposto na Seção 5.3, e redes neurais artificiais. Verificou-se que o método permite que um algoritmo de busca multimodal (CLONALG) encontre uma maior quantidade de ótimos locais da função, e também pode aumentar o percentual do número de vezes que um GA converge para o ótimo global da função, principalmente quando populações de tamanho reduzido estão sendo utilizadas. Considerando a inicialização de redes neurais do tipo MLP, o SAND permitiu que, em alguns casos, o algoritmo de treinamento convergisse em um menor número de iterações, mas não apresentou influência significativa na capacidade de generalização da rede treinada.

6.4. ABNET

O algoritmo para a geração de uma rede neural Booleana, ABNET, descrito na Seção 5.4, foi desenvolvido com o objetivo de ilustrar como diversos princípios imunológicos podem ser utilizados para o desenvolvimento e/ou aperfeiçoamento de algoritmos de treinamento de redes neurais artificiais (RNA). Para avaliar o desempenho da ABNET, ela foi aplicada a 3 problemas distintos e seus resultados comparados com outros algoritmos de treinamento de RNAs, como as redes competitivas e os mapas auto-organizáveis de Kohonen (Seção 4.2.2.2) e as redes de Hopfield discretas (Seção 4.2.2.3). Todos os resultados apresentados constituem a média de 10 simulações. Os parâmetros empregados no treinamento da ABNET serão discutidos na Seção 6.4.4 que trata da análise de sensibilidade da rede em relação aos parâmetros de entrada definidos pelo usuário.

6.4.1. Decisão de Lógica Majoritária

Existem diversos problemas na área de telecomunicações que podem ser descritos através de mapeamentos binários, incluindo o problema de Decisão de Lógica Majoritária (MLD – *major logic decision*), que é a regra de decisão que minimiza a probabilidade de erro de mensagem no receptor de uma linha de comunicação. Para avaliar o desempenho da ABNET quando aplicada a problemas de mundo real, seu algoritmo de treinamento foi empregado na obtenção da solução do problema MLD.

6.4.1.1. Descrição do Problema

Uma possível aplicação do serviço de comunicação digital é a investigação da modulação do espectro de dispersão utilizando um detector não coerente FH-FSK (*frequency hopping-*

frequency shift keying), como proposto pelos laboratórios Bell (Goodman *et al.*, 1980; Einarson, 1980).

O diagrama de blocos de um transmissor FH-CDMA de múltiplos níveis é apresentado na Figura 6.19(a). Todas as operações binárias consideradas foram tomadas sobre o *campo de Galois* $GF(Q)$. A palavra código resultante enviada através do canal por uma seqüência de L sinais FSK é dada por

$$\bar{y} = \bar{a} + \bar{x}, \quad (6.1)$$

onde \bar{x} representa uma mensagem Q -ária gerada pelo m -ésimo usuário, e \bar{a} seu endereço, ambos de comprimento L ($L < Q$). \bar{a} é adicionado componente a componente à mensagem do usuário.

Estas seqüências FSK podem ser representadas como entradas para uma matriz de Q linhas e L colunas (Einarson, 1980). A duração destas seqüências é denominada de *tempo de chip* (*chip time*). Como exemplo, considere o sistema com $Q = 7$ e $L = 5$, ilustrado na Figura 6.19(b). A matriz de transmissão corresponde à mensagem do usuário, tomada arbitrariamente como sendo $\bar{x} = (1, 1, 1, 1, 1)$, adicionada a seu endereço $\bar{a} = (3, 0, 6, 1, 4)$, resultando em uma palavra código $\bar{y} = (4, 1, 0, 2, 5)$. Note que, neste caso, o endereço da palavra código foi movido ciclicamente (adição de módulo 7).

A seqüência FSK pode ser afetada pelas seqüências de outros usuários e, também, por ruído, que foi desconsiderado nesta aplicação. Assim, apenas outros usuários poderiam interferir na transmissão e recepção dos dados.

O receptor é composto por um despallhador de freqüência, que extrai o endereço da seqüência FSK recebida, e Q detectores de energia. Durante os L intervalos de chip, as saídas dos detectores são comparadas à um limiar e é preciso decidir se a freqüência correspondente está presente ou não.

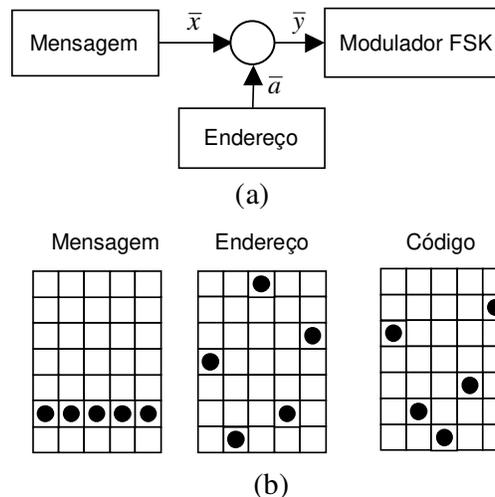


Figura 6.19. Receptor FH-CDMA de múltiplos níveis. (a) Diagrama de blocos. (b) Exemplo para $Q = 7$ e $L = 5$: $\bar{x} = (1, 1, 1, 1, 1)$, $\bar{a} = (3, 0, 6, 1, 4)$ e $\bar{y} = (4, 1, 0, 2, 5)$.

Um receptor de lógica majoritária com regra de máxima verosimilhança, para a mensagem m , é dado pela seguinte regra:

$$\max_m p(\bar{r} | \bar{y}_m), \quad m = 0, 1, 2, \dots, Q-1 \quad (6.2)$$

onde \bar{r} é a matriz do receptor, \bar{y}_m é uma das possíveis matrizes (transmitidas pelo usuário m), que está sendo codificada, e $p(\cdot | \cdot)$ é a probabilidade condicional.

Assim, um receptor de máxima verosimilhança, que minimiza a probabilidade de erro para mensagens de mesma probabilidade a priori, pode ser implementado por um receptor de regra majoritária que seleciona a mensagem \bar{y} como a palavra código transmitida (matriz) com um maior número de entradas na matriz recebida.

6.4.1.2. Receptores Neurais para o Problema MLD

Diversas abordagens utilizando redes neurais artificiais foram propostas e testadas para a solução do problema de decisão de lógica majoritária. De Deus Jr. *et al.* (1999a) propuseram uma nova arquitetura neural e algoritmo de treinamento, especialmente desenvolvidos para resolver o problema de decisão de lógica majoritária. Esta rede, denominada, NTNET (*non-trainable neural network*), é composta por 2 módulos, onde o primeiro executa a soma das linhas da matriz, e o segundo módulo é competitivo, sendo responsável pela determinação da mensagem decodificada. A NTNET permite o projeto do receptor de lógica majoritária com um conjunto de neurônios do tipo threshold. Outras arquiteturas de RNAs também foram empregadas para comparação (de Deus Jr. *et al.* (1999a,b)): o perceptron de múltiplas camadas (MLP), treinado com o algoritmo de backpropagation, e um método de poda para os mapas auto-organizáveis de Kohonen com grid de saída uni-dimensional (PSOM) proposto por de Castro & Von Zuben (1999b). Note que estas arquiteturas de rede neural operam com conexões assumindo valores reais, enquanto a ABNET e a NTNET possuem pesos binários para as conexões. Para aplicar o MLP a este problema, os 7 padrões de entrada foram rotulados, de forma que o MLP pudesse mapeá-los em diferentes unidades de saída.

Para comparar o desempenho dos 4 algoritmos de treinamento de RNA: MLP, NTNET, PSOM e ABNET, considere um sistema com $Q = 7$ e $L = 3$. Neste caso, resolver o problema de decisão de lógica majoritária implica em projetar um receptor com um desempenho, em termos de probabilidade de erro de palavra, como dado pela Figura 6.20.

Todas as redes foram capazes de resolver o problema, ou seja, atingir a probabilidade de erro de palavra desejada. Entretanto, seus algoritmos de treinamento são completamente distintos, tornando difícil uma comparação de desempenho em termos de custo computacional e acurácia dos resultados obtidos. Assim, as diferentes redes neurais foram comparadas em relação à complexidade de suas arquiteturas resultantes, ou seja, quantidade de conexões (pesos e limiares) ao final do processo de treinamento.

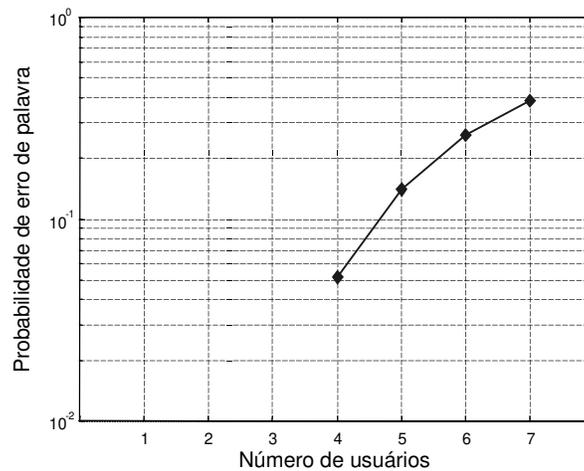


Figura 6.20. Desempenho do problema MLD para $Q = 7$ e $L = 3$. A probabilidade de que uma mensagem enviada seja decodificada corretamente, denominada probabilidade de erro de palavra, é igual a 0 para um número de usuários inferior a 4.

A rede MLP com arquitetura mais simples capaz de resolver o problema continha 181 conexões, com 21 entradas, 6 unidades intermediárias e 7 saídas. A NTNET precisou de 21 conexões. Se desconsiderarmos as conexões cujos pesos associados assumem o valor 0, a arquitetura resultante da ABNET também apresenta apenas 21 conexões, de acordo com a Figura 6.21. O algoritmo de treinamento do PSOM resultou em uma rede com 147 conexões de valores reais (21×7), entretanto, os pesos do PSOM podem ser quantizados (decodificados) para assumirem os valores $\{0,1\}$ e, caso as conexões com pesos associados 0 sejam descartadas, a rede resultante também apresentará 21 conexões.

A Tabela 6.6 resume a comparação de desempenho dos algoritmos em relação à complexidade de suas arquiteturas resultantes. Esta tabela também mostra que a representação binária intrínseca dos pesos, faz com que a NTNET e a ABNET sejam particularmente adequadas para a solução do problema MLD.

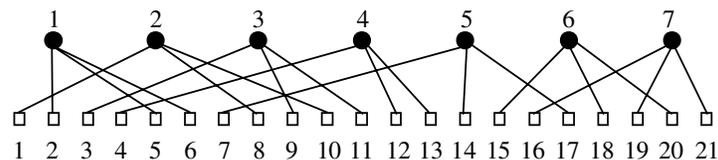


Figura 6.21. Exemplo de ABNET resultante para o problema MLD, onde apenas as conexões com pesos associados correspondentes a 1 foram apresentadas. A rede possui 21 entradas e 7 saídas, resultando em um vetor de pesos de comprimento 21.

Tabela 6.6. Complexidade das redes (MLP, NTNET, PSOM e ABNET) para resolver o problema MLD. A quantidade final de conexões apresentada para a ABNET e PSOM são tomadas após a eliminação dos pesos com valores nulos.

Rede Neural	Número de Conexões	Complexidade da Arquitetura	Representação Binária
MLP	181	Alta	Não
NTNET	21	Baixa	Sim
PSOM	21	Baixa	Não
ABNET	21	Baixa	Sim

6.4.2. Classificação de Lentes de Contato

Este problema de reconhecimento (classificação) de padrões foi utilizado por Witten & McDonald (1988) para estudar a capacidade de aquisição de conhecimentos. O banco de dados está disponível para download no repositório de banco de dados para aprendizagem de máquina da Universidade da Califórnia (URL 3). Nove regras cobrem o conjunto de treinamento e os exemplos são completos, sem ruídos e constituem uma simplificação do problema. Estão disponíveis $M = 24$ instâncias com $L = 4$ atributos, mapeadas em 3 classes.

Trata-se de um problema simples de classificação e nosso objetivo é testar a capacidade da ABNET em aprender os padrões de entrada. É importante salientar que este problema não é binário, requerendo uma modelagem binária (quantização) dos dados para que a ABNET possa ser aplicada. As entradas originais assumem apenas os valores {1,2,3}, que serão representados por 1: [1 1 1]; 2: [0 1 0] e 3: [0 0 1], correspondentes a uma quantidade total de 12 bits para representar cada amostra. Os números 1, 2 e 3 poderiam ter sido modelados com apenas 2 bits, mas isso resultaria em números com diferentes distâncias de Hamming (HD) entre eles, por exemplo, se 1: [1 1]; 2: [0 1] e 3: [0 0], a HD entre os números 1 e 2 seria HD = 1, enquanto entre os números 1 e 3 seria HD = 2. A quantização escolhida mantém uma distância de Hamming simétrica entre os três valores.

A ABNET foi comparada com uma rede competitiva (Seção 4.2.2.2) denominada RC, com o mapa auto-organizável de Kohonen com grid de saída uni-dimensional (Seção 4.2.2.2.1) denominado SOM, e com o PSOM, proposto por de Castro & Von Zuben (1999b), para a rede de Kohonen. A RC apresenta um número fixo de saídas N , pré-especificado antes do treinamento. Os resultados apresentados na Tabela 6.7 são a média e o desvio padrão, obtidos a partir de 10 simulações, do percentual de classificação correta (PCC) das diferentes RNAs avaliadas. Os resultados da ABNET levam em consideração o limiar de afinidade ϵ e arquiteturas com dimensão final distinta. O algoritmo de treinamento do PSOM requer a especificação do limiar de poda ξ e a quantidade final de neurônios na rede também é variável. Como as redes RC, SOM e PSOM possuem pesos ou conexões com valores reais, as entradas foram tomadas assumindo seus valores originais, porém normalizadas no intervalo [-1,1].

Tabela 6.7. Aplicação das redes RC, SOM, ABNET e PSOM ao problema de classificação de lentes de contato. PCC é o percentual de classificação correta, N a quantidade de unidades de saídas da rede, ϵ o limiar de afinidade e ξ o limiar de poda do PSOM. Os resultados apresentados são a média e desvio padrão obtidos a partir de 10 simulações. No casos das redes RC e SOM, foram escolhidos arbitrariamente diferentes valores para N .

RC		SOM		ABNET			PSOM		
N	PCC (%)	N	PCC (%)	ϵ	N	PCC (%)	ξ	N	PCC (%)
24	92.50 ± 1.75	24	97.08 ± 3.43	0	24 ± 0.00	100 ± 0.00	0.001	13.7 ± 1.30	95.00 ± 4.30
19	92.50 ± 1.75	19	96.67 ± 2.64	1	24 ± 0.00	100 ± 0.00	0.005	12.7 ± 1.06	92.08 ± 4.99
14	92.92 ± 2.01	14	96.25 ± 3.07	2	21.9 ± 0.88	97.08 ± 3.43	0.01	13.2 ± 1.81	92.92 ± 4.83
9	92.09 ± 1.32	9	92.09 ± 1.32	3	21 ± 1.41	95.83 ± 3.93	0.05	8 ± 0.00	91.67 ± 0.00
4	75.00 ± 4.81	4	77.50 ± 8.83	4	18.2 ± 1.23	93.75 ± 4.05	0.1	8 ± 0.00	91.67 ± 0.00

Pode ser observado que, devido à quantização dos dados de entrada, a ABNET apresentou as arquiteturas finais menos parcimoniosas, quando comparadas ao algoritmo PSOM, por exemplo. Por outro lado, a ABNET demonstrou ser muito poderosa na aquisição de conhecimentos, sendo capaz de classificar os padrões de entrada corretamente com 100% de acerto para um limiar de afinidade de 0 ou 1. Nestes casos, a rede evoluída contém anticorpos altamente específicos aos antígenos a serem reconhecidos.

6.4.3. Reconhecimento de Caracteres Binários

Este problema objetiva avaliar a tolerância a ruídos e capacidade de generalização da ABNET, quando caracteres binários são apresentados à rede. Neste caso, a ABNET foi comparada com a rede de Hopfield discreta, descrita na Seção 4.2.2.3 e utilizando a regra de aprendizagem da pseudo-inversa.

Os padrões de entrada utilizados para testar a tolerância a ruídos da rede foram artificialmente gerados e estão apresentados na Figura 6.22. Um ruído aleatório foi inserido nas amostras simplesmente revertendo-se bits em 0 em bits em 1, e vice-versa. Foram testados 4 níveis de ruído distintos: 5.7%, 11.4%, 20% e 40%, correspondendo à troca de 2, 4, 7 e 14 bits, respectivamente. A ABNET foi testada para diferentes valores do limiar de afinidade ϵ ($\epsilon = 0, 1, 2, 3, 4$ e 5) e uma classificação correta é considerada quando a rede mapeia o padrão corrompido (com ruído) no mesmo neurônio de saída em que foi mapeado o padrão corresponde na ausência de ruído. No caso da rede de Hopfield, uma classificação correta é considerada quando a distância de Hamming entre o padrão restaurado e o armazenado é menor do que δ , onde $\delta = 0, 1, 2, 3, 4$ e 5.



Figura 6.22. 10 caracteres com resolução 7×5 empregados na avaliação da tolerância a ruídos da ABNET.

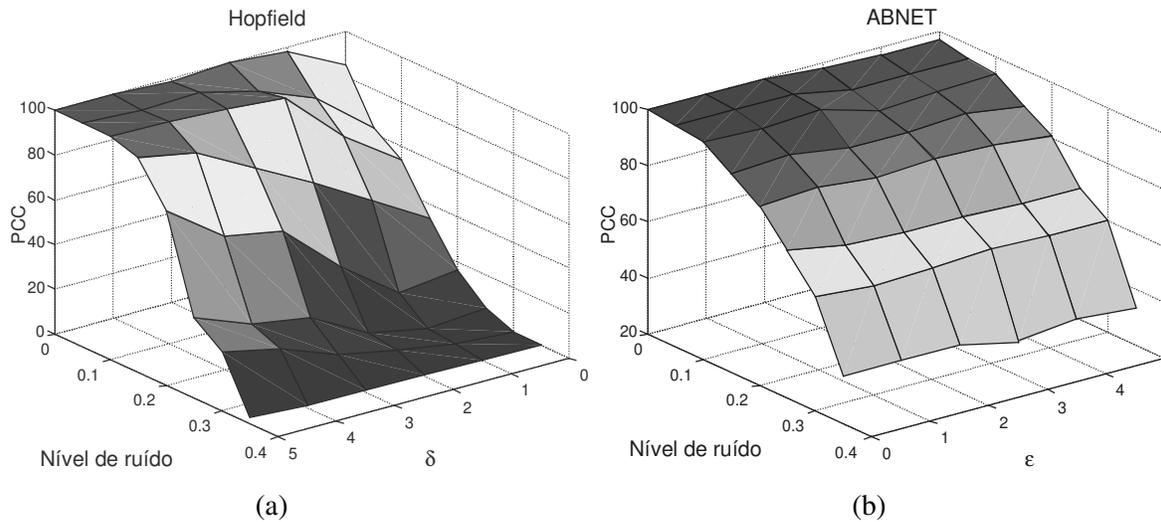


Figura 6.23. Tolerância a ruídos (percentual de classificação correta – PCC). (a) Rede de Hopfield, onde δ é a distância de Hamming permitida entre o padrão recuperado e o padrão armazenado. (b) ABNET, onde ϵ é o limiar de afinidade da rede.

A Figura 6.23 apresenta os resultados da rede de Hopfield e ABNET quando aplicadas ao problema de reconhecimento de caracteres binários. É possível observar que, caso seja requerido um reconhecimento perfeito pela rede de Hopfield ($\delta = 0$) entre o padrão armazenado e o corrompido, seu desempenho apresenta uma forte degradação com o nível de ruído. A ABNET demonstrou ser menos sensível ao nível de ruído do que a rede de Hopfield. É importante ressaltar que os parâmetros δ e ϵ não são diretamente comparáveis. Enquanto δ representa a acurácia do estado estacionário alcançado pela rede de Hopfield ao final do processo iterativo de treinamento (medindo a existência de estados espúrios), ϵ indica o limiar de afinidade necessário para a aprendizagem da ABNET. Entretanto, para efeitos práticos, o papel desempenhado por estes parâmetros nos problemas de classificação é equivalente.

Para avaliar a capacidade de generalização da ABNET, foi empregado um conjunto de caracteres distintos do anterior, como ilustrado na Figura 6.24. Este conjunto amostral é composto por $M = 29$ caracteres de comprimento $L = 400$ (resolução 20×20), onde 6 deles correspondem a diferentes fontes da letra A, e os outros representam outras letras do alfabeto. A idéia é treinar as redes ABNET e Hopfield com os dados da Figura 6.24(a) e verificar se elas são capazes de reconhecer os caracteres da Figura 6.24(b), que são diferentes fontes para a letra A, como pertencentes a grupos representativos dos caracteres A. Para isso, é necessário que estas redes sejam capazes de criar estes grupos representativos da letra A.

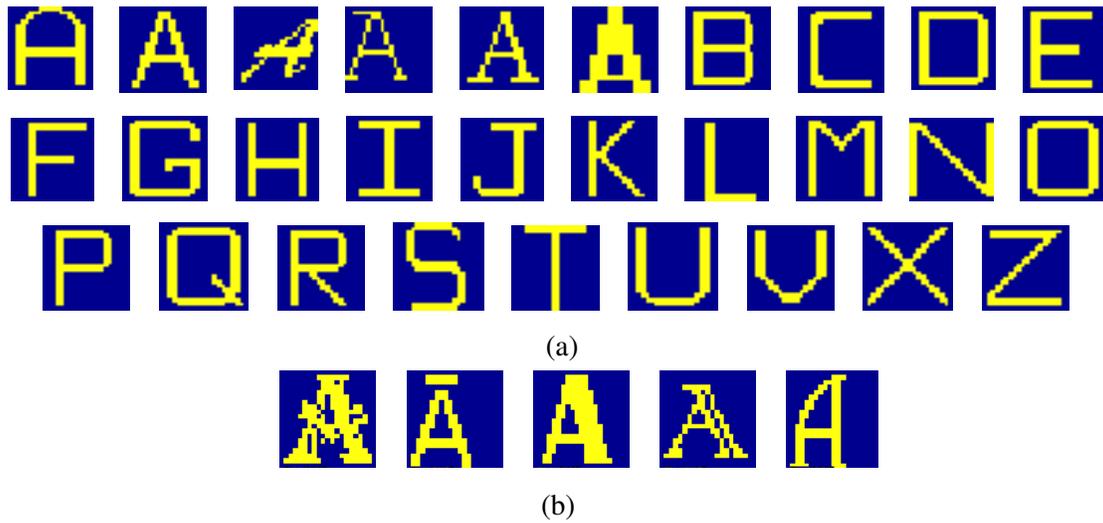


Figura 6.24. Caracteres para testar a capacidade de generalização da ABNET. (a) Amostras do alfabeto com 6 letras A distintas. (b) Conjunto de teste independente.

A Figura 6.25(a) ilustra os padrões recuperados pela rede de Hopfield e a associação de padrões realizada pela ABNET. Pode ser visto que a rede de Hopfield não foi capaz de recuperar um dos caracteres, resultando em um estado espúrio. A ABNET agrupou as diferentes fontes de A em dois grupos representativos do caractere A, resultando em uma classificação correta.

Hopfield					ABNET ($\epsilon \geq 55$)				
Teste					Teste			Teste	
Padrões associados					Padrões associados			Padrões associados	
(a)					(b)				

Figura 6.25. Resultados obtidos para as redes de Hopfield e ABNET aplicadas ao problema de capacidade de generalização para caracteres binários. (a) Hopfield. (b) ABNET.

6.4.4. Análise de Sensibilidade

O pseudocódigo para o algoritmo de treinamento proposto para a ABNET na Seção 5.4.3 apresenta a seguinte linha de comando $[\mathbf{Ab}, \mathbf{F}] = \text{abnet}(\mathbf{Ag}, L, \text{gen}, \varepsilon, \alpha_0, \beta)$. Assim como para o algoritmo CLONALG discutido anteriormente, os dados de treinamento correspondem ao conjunto antigênico \mathbf{Ag} de comprimento L e, portanto, são determinados pelo problema a ser estudado. Para todos os problemas abordados, adotou-se $\beta = 3$ e $\alpha_0 = 3$.

No caso da ABNET, a quantidade mínima de gerações para o treinamento da rede merece uma análise um pouco mais cuidadosa do que no caso do algoritmo CLONALG. Por se tratar de uma rede construtiva cujos neurônios são inseridos a cada β iterações (ou gerações), é preciso que a quantidade gen de gerações de treinamento seja suficiente para que o processo construtivo da rede permita a inserção de todos os neurônios necessários à aprendizagem e classificação correta do conjunto de treinamento. Na Seção 3.4.2.1 foi demonstrado analiticamente que a cobertura de cada anticorpo em um espaço de formas binário de Hamming pode ser determinada pela seguinte expressão

$$C = \sum_{i=0}^{\varepsilon} \binom{L}{i} = \sum_{i=0}^{\varepsilon} \frac{L!}{i!(L-i)!}, \quad (6.3)$$

onde C é a cobertura de cada anticorpo e L o comprimento das cadeias.

Para cadeias de comprimento L , é possível gerar um conjunto total de 2^L cadeias distintas e, portanto, a quantidade máxima N_{\max} de anticorpos que precisam ser inseridos na ABNET para que ela seja capaz de apresentar a cobertura máxima do espaço de formas pode ser dada pela Equação (6.4), que é função do limiar de afinidade ε .

$$N_{\max} = \frac{2^L}{C}. \quad (6.4)$$

Ou seja, N_{\max} corresponde ao número total de cadeias que podem ser geradas dividido pela cobertura de cada anticorpo. Se N_{\max} for maior do que a quantidade M de antígenos a serem reconhecidos ($N_{\max} > M$), então $N_{\max} = M$, já que não é necessário mais do que um anticorpo para reconhecer cada antígeno do conjunto amostral. Dado N_{\max} , e considerando que a cada β gerações um anticorpo é verificado para poder entrar na rede, a seguinte regra pode ser derivada para a determinação da quantidade mínima gen de gerações necessária para que a ABNET convirja para uma rede de dimensão apropriada

$$\text{gen} = (\beta + 1) N_{\max}. \quad (6.5)$$

Para o primeiro problema testado na Seção 6.4.1, o limiar de afinidade foi assumido nulo, indicando que um reconhecimento perfeito entre um antígeno e um anticorpo faz-se necessário para que a ABNET classifique os dados de treinamento. No caso dos dois outros problemas abordados, Seções 6.4.2 e 6.4.3, o desempenho da rede foi avaliado para diferentes valores de ε . No problema de classificação de lentes de contato, os resultados apresentados na Tabela 6.7 indicam que ε influencia na dimensão final da rede e também no seu percentual de classificação correta.

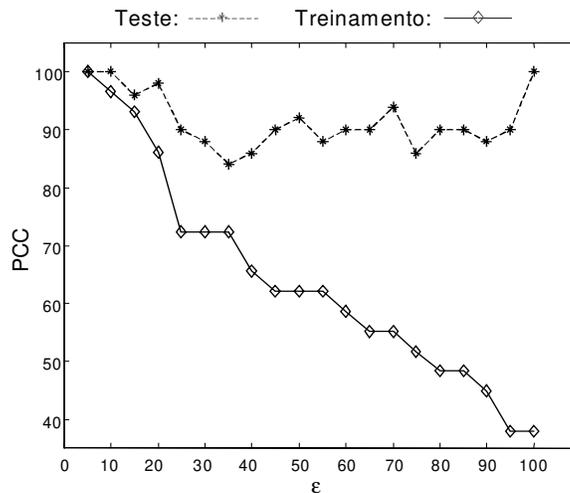


Figura 6.26. Capacidade de generalização da ABNET (percentual de classificação correta – PCC) como função do limiar de afinidade ϵ para os padrões de treinamento e teste apresentados na Figura 6.24.

A Figura 6.26 mostra como a capacidade de generalização da ABNET varia em relação ao limiar de afinidade ϵ . Esta figura mostra que a classificação para os padrões de treinamento sofre uma degradação com ϵ , mas a capacidade de generalização para a classe A não sofre uma degradação significativa, mantendo uma classificação correta em torno de 92%. Este fenômeno permite concluir que a ABNET está mantendo pelo menos uma classe separada para os padrões A, e está misturando os outros caracteres. Isto pode ser esperado se considerarmos que a classe A é predominante no conjunto amostral.

Por último, a taxa inicial de hipermutação α_0 exerce influência na velocidade de convergência do algoritmo de treinamento da ABNET. Para avaliar como se dá esta influência, foi feito o seguinte experimento. Uma ABNET foi inicializada com $N = 10$ anticorpos de comprimento $L = 35$ com o objetivo de aprender os caracteres da Figura 6.22 para $\epsilon = 0$. Como a ABNET possui a quantidade mínima de anticorpos necessária para reproduzir os dados de treinamento para $\epsilon = 0$, resta-lhe a tarefa de aprender o conjunto amostral. A Figura 6.27 mostra a sensibilidade do algoritmo em relação ao parâmetro α_0 para os dados de treinamento da Figura 6.22, $\epsilon = 0$ e $N = 10$. É possível verificar que para valores de $\alpha_0 < 6$ a velocidade de convergência da rede é menor. Também é possível perceber que valores elevados de α_0 não reduzem significativamente o tempo de treinamento da rede.

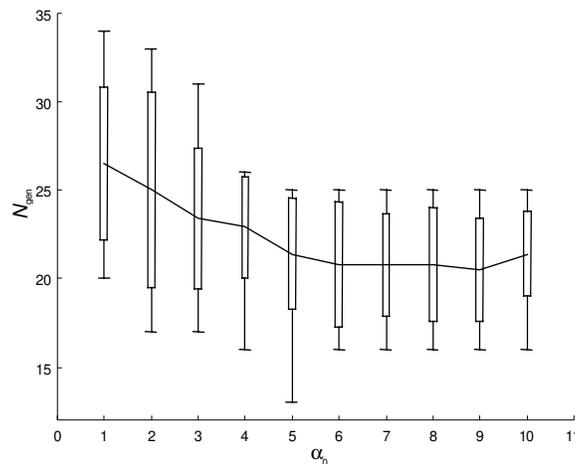


Figura 6.27. Sensibilidade da ABNET em relação ao parâmetro α_0 para os dados de treinamento da Figura 6.22, $\varepsilon = 0$ e $N = 10$. Valores máximo, mínimo, média e desvio padrão da quantidade de épocas necessária para convergência (N_{gen}).

6.4.5. Discussão

As redes neurais testadas não apresentaram dificuldade para solucionar o problema de decisão de lógica majoritária. Entretanto, a principal vantagem da ABNET sobre as outras redes neurais testadas, é que ela representa uma arquitetura genérica, com estratégia de aprendizagem inerentemente Booleana, que pode ser aplicada a diversos problemas. A outra rede que apresentou bom desempenho para este problema foi a NTNET, especialmente projetada para resolver o problema MLD.

No exemplo descrito na Seção 6.4.2, a ABNET foi capaz de realizar a aquisição de conhecimentos com altas taxas de reconhecimento de padrões. Também foi possível observar que, como proposto na teoria (Seção 5.4) e verificado na análise de sensibilidade (Seção 6.4.4), valores pequenos de ε (0 ou 1) implicam em uma rede resultante com anticorpos altamente específicos aos estímulos antigênicos e muitas unidades são inseridas para que a rede possa representar apropriadamente os dados de treinamento. Quanto mais generalistas os anticorpos, menos unidades são inseridas e mais parcimoniosa é a rede.

A ABNET apresenta algumas vantagens sobre a rede de Hopfield discreta na solução de problemas de reconhecimento de caracteres binários. Ela implementa uma espécie de classificador ótimo de erro mínimo quando ruído é inserido de forma aleatória e independente nos padrões de treinamento. A ABNET também necessitou de menos conexões do que a rede de Hopfield para o problema testado: 400×400 (160.000) conexões na rede de Hopfield, contra 400×29 (11.600) conexões da ABNET para $\varepsilon = 0$. A ABNET também pode ser modificada para tornar-se um classificador ótimo de erro mínimo quando os padrões de entrada são bipolares. Além disso, ela não apresenta estados espúrios.

Durante a apresentação do algoritmo de backpropagation, Rumelhart *et al.* (1986) discutiram 7 problemas, dos quais podemos citar, por conveniência, 6: XOR, paridade, codificação, simetria, adição e negação. Para todos estes exemplos, a rede a ser treinada

pode ser classificada como Booleana, onde a informação a ser adquirida está codificada sob a forma de cadeias binárias. Devido ao padrão de conexões de ABNET e ao seu algoritmo de treinamento, ela é capaz de resolver problemas como estes de forma direta e eficiente, enquanto algumas redes que operam com pesos reais, como o MLP treinado com o algoritmo de backpropagation, teriam dificuldades de convergência e, além disso, demandariam um esforço computacional muito superior.

Finalmente, a implementação em hardware da ABNET é simplificada, pois ela não estaria sujeita aos efeitos de quantização do conjunto de pesos e seu algoritmo de treinamento opera diretamente com valores binários.

6.5. aiNet

Para avaliar o desempenho da rede imunológica artificial proposta, denominada aiNet, ela foi aplicada a diversos problemas de teste e seu desempenho comparado com os mapas auto-organizáveis de Kohonen (SOM – *self-organizing map*) com grid unidimensional de saída (ver Seção 4.2.2.2). Os parâmetros de treinamento do SOM são: valor inicial da taxa de aprendizagem $\alpha_0 = 0.9$, valor final da taxa de aprendizagem $\alpha_f = 10^{-3}$, com um decrescimento geométrico de 0.9 a cada 5 iterações. Os pesos foram inicializados empregando-se uma distribuição uniforme no intervalo $[-0.1, 0.1]$. A quantidade de neurônios m do grid de saída irá depender do problema a ser abordado. Ao final do processo de treinamento do SOM, aquelas unidades de saída que não forem responsáveis pela classificação de nenhum dado de entrada serão removidas. Dentre as características avaliadas, destacam-se a capacidade de compressão de informação (amostras de treinamento) medida pelo índice CR (*compression rate*), e de separação e representação de clusters. Uma versão deste algoritmo também foi empregada na determinação da quantidade e localização de centros de redes neurais com funções de ativação de base radial (RBF), discutidas na Seção 4.2.2.1.

6.5.1. Análise de Dados

Foram avaliados quatro problemas de teste, como ilustrados na Figura 6.28, CLASSLS, CLASSNLS, SPIR e CHAINLINK.

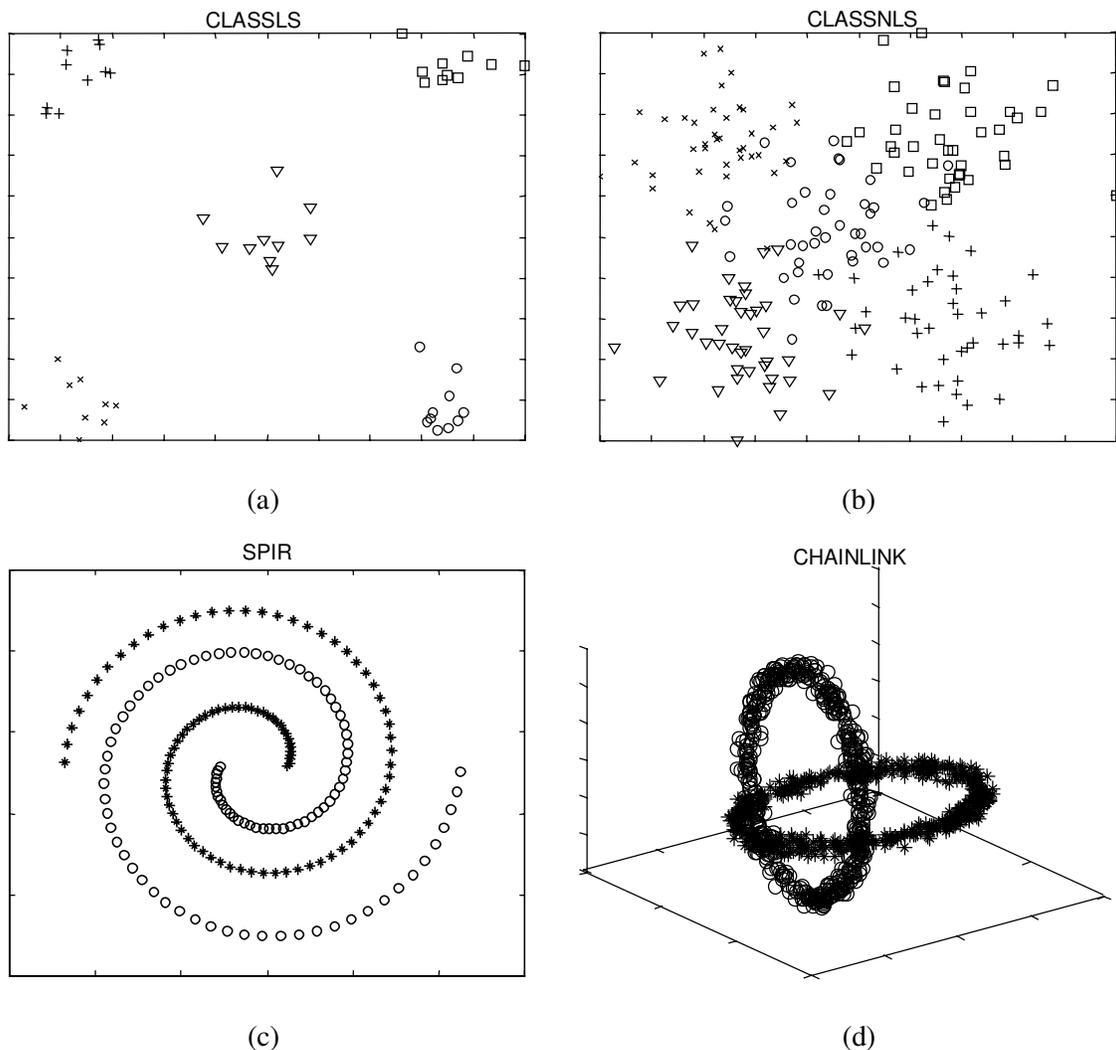


Figura 6.28. Antígenos (dados de treinamento) para avaliação de desempenho do algoritmo aiNet. (a) CLASSLS, $M = 50$. (b) CLASSNLS, $M = 200$. (c) SPIR, $M = 190$. (d) CHAINLINK, $M = 500$.

6.5.1.1. CLASSLS

Este problema é composto por cinco classes linearmente separáveis, cada qual contendo 10 amostras e está ilustrado na Figura 6.28(a). As amostras foram geradas utilizando-se distribuições gaussianas com diferentes médias e variâncias fixas. O objetivo é ilustrar a capacidade da aiNet em separar, definir e representar clusters, construir a arquitetura final da rede e reduzir a complexidade do problema, ou seja, comprimir o conjunto amostral, o qual será apresentado à aiNet sem os rótulos, ou seja, sem o conhecimento prévio da classe a que cada dado pertence .

Considerando o algoritmo de treinamento apresentado na Seção 5.5.3, a seguinte linha de comando foi utilizada: $[\mathbf{Ab}_m, \mathbf{S}] = \text{aiNet}(\mathbf{Ag}, 2, 10, 4, 0.2, 1, 0.14, 10)$.

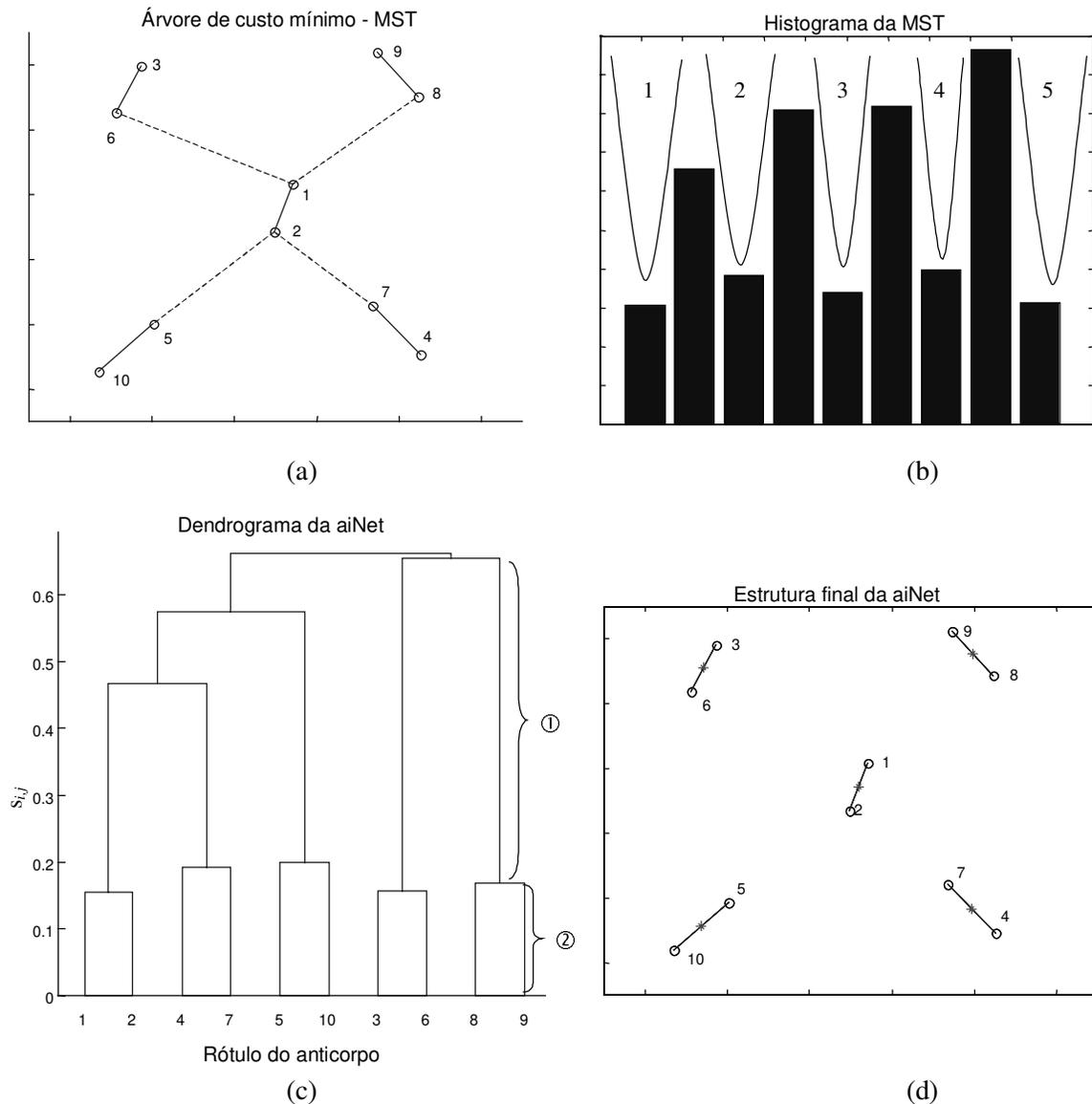


Figura 6.29. Resultados apresentados pela aiNet para o problema CLASSLS. (a) Árvore de custo mínimo (MST), onde as linhas tracejadas indicam conexões a serem removidas. (b) Histograma da MST. (c) Dendrograma da rede indicando grandes diferenças entre os valores de fusão (① em relação a ②). (d) Arquitetura final da aiNet com os cinco clusters representados pelos seus respectivos centros de massa.

A Figura 6.29(a) apresenta a árvore de custo mínimo (MST) resultante, representando uma rede composta por $m = 10$ anticorpos de memória, o que corresponde a uma compressão $CR = 80\%$ do conjunto amostral. As linhas tracejadas indicam conexões inconsistentes, a serem detectadas pelo histograma da MST e removidas (podadas). A quantidade de vales na Figura 6.29(b) permite identificar 5 clusters distintos no conjunto de dados. Também é possível utilizar o critério do tamanho do passo (*stepsize*) para empregar o dendrograma da aiNet na definição dos clusters resultantes da rede.

Tabela 6.8. Valores de pertinência de cada anticorpo c_i , $i = 1, \dots, 10$, da aiNet em relação ao centróide v_j , $j = 1, \dots, 5$, de cada cluster.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}
v_1	1.00	1.00	0.67	0.71	0.76	0.69	0.84	0.75	0.71	0.66
v_2	0.58	0.63	0.50	1.00	0.68	0.50	1.00	0.60	0.56	0.64
v_3	0.67	0.50	0.63	0.56	0.50	0.58	0.57	1.00	1.00	0.50
v_4	0.50	0.55	0.54	0.62	1.00	0.57	0.64	0.50	0.50	1.00
v_5	0.60	0.50	1.00	0.50	0.59	1.00	0.50	0.60	0.56	0.64

Um passo de tamanho igual a 1.5 seria adequado para classificar corretamente os anticorpos da rede, como pode ser visto na Figura 6.29(c). A Figura 6.29(d) apresenta a arquitetura final da aiNet com cinco clusters distintos, onde cada um deles está representado pelo seu respectivo centro de massa, ou centróide (estrelas).

Uma vez definida a arquitetura final da rede (Figura 6.29(d)) e os clusters representados pelos seus respectivos centros de massa, é possível aplicar os conceitos de clusterização gradual discutidos na Seção 5.5.4.2 para a aiNet resultante. A matriz de proximidades U especificando o grau de pertinência de cada anticorpo da rede em relação aos centróides determinados é apresentada na Tabela 6.8. A partir desta tabela percebe-se que os anticorpos c_1 e c_2 pertencem ao cluster v_1 com pertinência 1.00, ao cluster v_2 com pertinência 0.58 e 0.63 ($u_{2,1}$ e $u_{2,2}$) respectivamente, e assim por diante.

Considere agora o caso de aplicar um SOM unidimensional ao mesmo problema. Os resultados estão apresentados na Figura 6.30 para uma quantidade final de saídas $m = 16$ ($m_0 = 20$). É possível verificar que esta rede também é capaz de solucionar este problema.

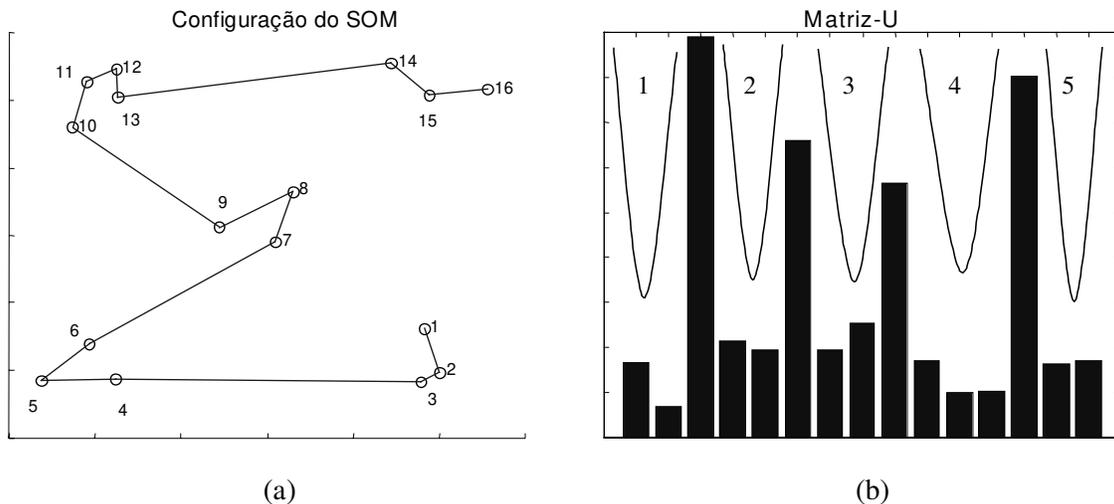


Figura 6.30. Aplicação de um SOM para o problema CLASSLS. (a) Posição final dos vetores de pesos e ligação com seus vizinhos mais próximos. (b) Matriz-U com cinco vales, cada qual representando um cluster diferente.

6.5.1.2. CLASSNLS

O problema ilustrado na Figura 6.28(b) é composto por 5 classes não-linearmente separáveis, onde cada classe possui 40 amostras de treinamento. Como pode ser observado, se os dados não estivessem rotulados nesta figura, como no caso dos modelos avaliados, seria difícil identificar diferenças entre os grupos de dados, até mesmo para um observador humano. Este problema foi utilizado por de Castro & Von Zuben (1999b) para avaliar o desempenho de um método de poda, denominado PSOM, para os mapas auto-organizáveis de Kohonen.

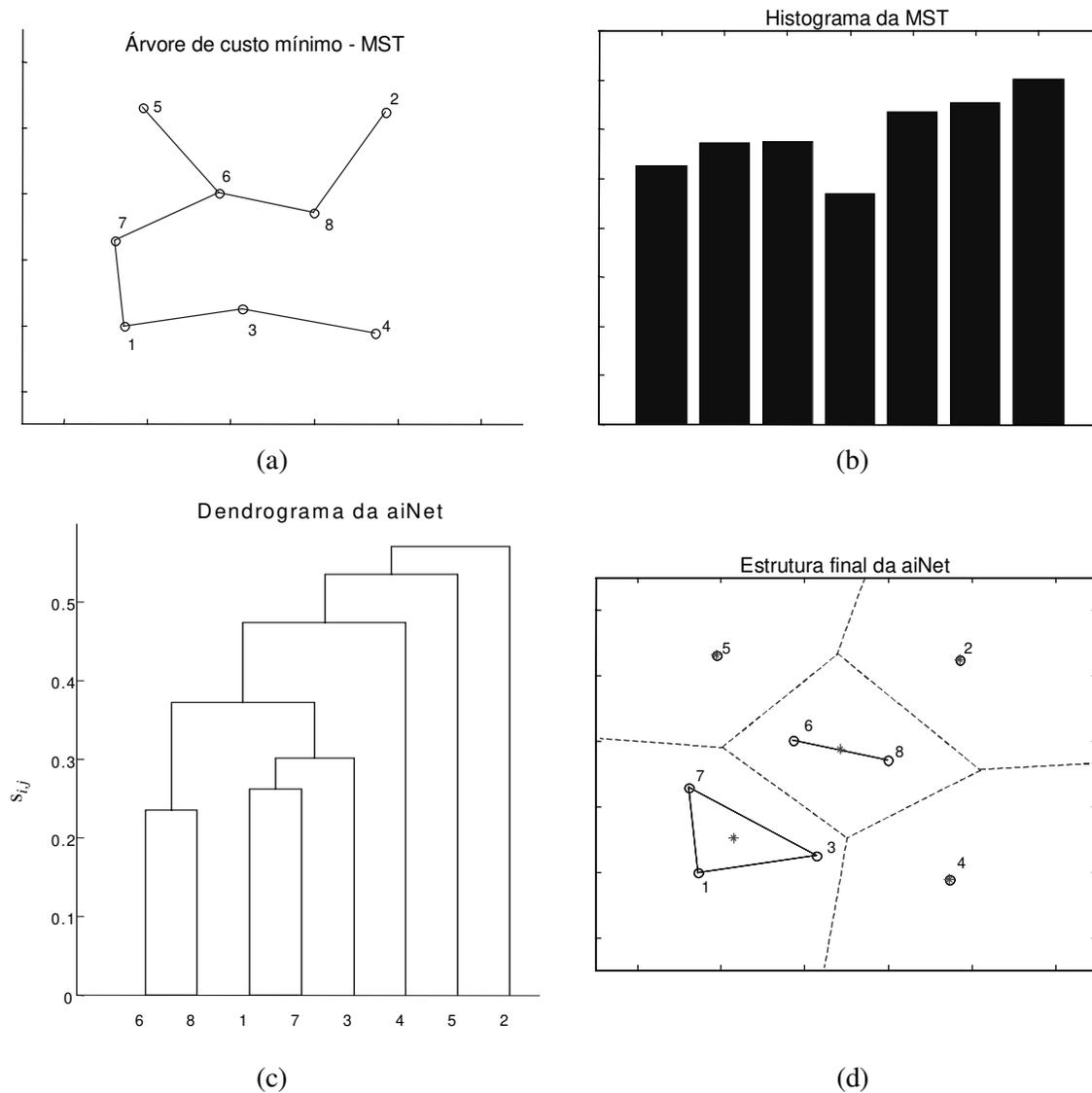


Figura 6.31. aiNet aplicada ao problema CLASSNLS. (a) Árvore de custo mínimo (MST). (b) Histograma da MST. (c) Dendrograma da rede indicando os cinco clusters de anticorpos. (d) Arquitetura final da aiNet com os cinco clusters representados pelos seus respectivos centros de massa e o diagrama de Voronoi particionando o espaço de formas em relação aos centros de massa de cada cluster.

Foram apresentados resultados referentes ao percentual de classificação dos dados de treinamento para diferentes dimensões do PSOM. Estes resultados serão reproduzidos objetivando comparar a classificação apresentada pela aiNet com a classificação obtida pelo PSOM.

Considerando o algoritmo de treinamento da Seção 5.5.3, a seguinte linha de comando foi utilizada: $[\mathbf{Ab}_m, \mathbf{S}] = \text{aiNet}(\mathbf{Ag}, 2, 10, 4, 0.2, 1, 0.2, 10)$. A aiNet apresentou uma taxa de compressão $CR = 96\%$, correspondendo a $m = 8$ anticorpos de memória na rede.

Como discutido na Seção 5.5.4, quando a estrutura final da rede está próxima da quantidade de clusters existente no conjunto antigênico (5 para este problema), o método da árvore de custo mínimo (MST) apresenta dificuldades para detectar e separar os clusters da aiNet. Neste caso, a utilização do dendrograma como critério de decisão se apresenta como uma boa alternativa. Como pode ser observado na Figura 6.31(b), nada pode ser concluído quanto à quantidade de clusters da aiNet partindo-se do histograma da MST. Entretanto, analisando a Figura 6.31(c), é possível determinar quantos clusters resultaram na aiNet e quais anticorpos pertencem a cada um deles: basta particionar o dendrograma tomando $s_{i,j} \approx 0.3$. A Figura 6.31(d) apresenta a arquitetura final da rede e as superfícies de decisão determinadas pelo diagrama de Voronoi dos anticorpos resultantes. Os clusters estão representados pelos seus respectivos centros de massa. A Figura 6.32 apresenta o diagrama de Voronoi da Figura 6.31(d) plotado sobre os dados de entrada.

A Tabela 6.9 apresenta os valores de pertinência de cada anticorpo da rede (c_i , $i = 1, \dots, 8$) em relação aos cinco clusters (v_i , $i = 1, \dots, 5$) detectados.

A Tabela 6.10 reproduz os resultados apresentados por de Castro & Von Zuben (1999b) adicionados do percentual de classificação obtido pela aiNet. O PSOM e a aiNet possuem estruturas cuja dimensão varia dinamicamente ao longo do processo de treinamento, enquanto o SOM possui um grid de saída com tamanho definido a priori. É possível verificar que o PSOM e a aiNet não são capazes de determinar arquiteturas com todas as dimensões listadas na Tabela 6.10; o que é representado pelas linhas tracejadas.

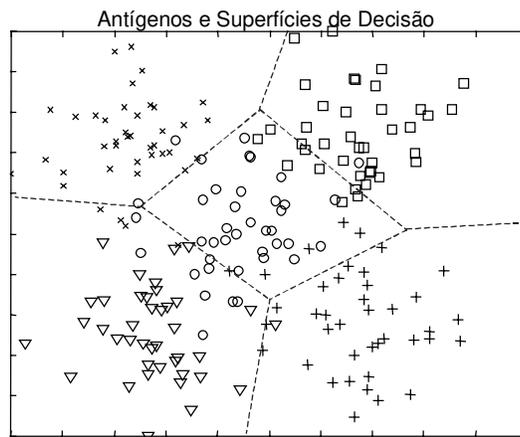


Figura 6.32. Superfícies (hiperplanos) de decisão dados pelo diagrama de Voronoi aplicado à aiNet resultante na Figura 6.31(d) e antígenos da Figura 6.28(b).

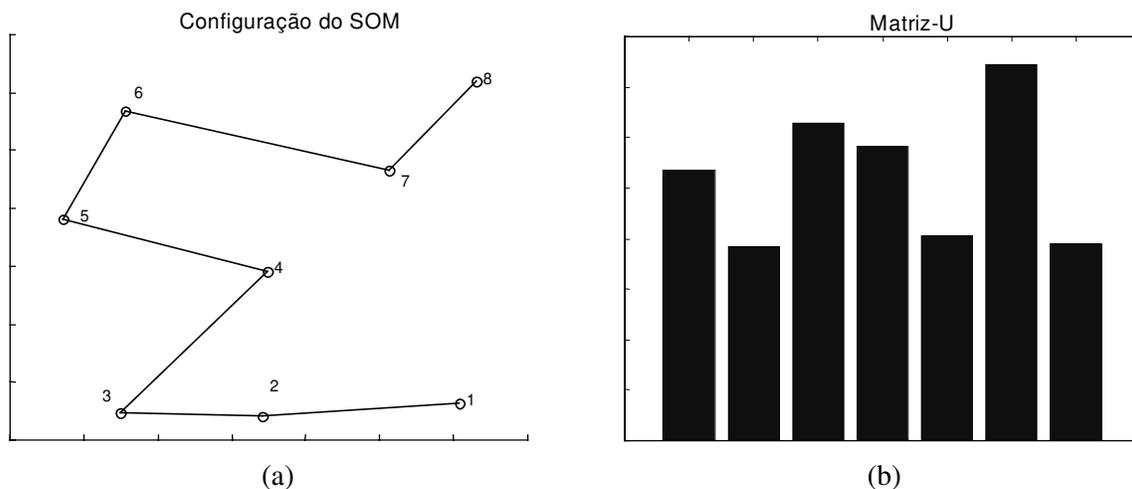
Tabela 6.9. Valores de pertinência de cada anticorpo c_i , $i = 1, \dots, 8$, em relação ao centróide de cada cluster v_j , $j = 1, \dots, 5$.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
v_1	1.00	0.50	1.00	0.50	0.50	0.86	1.00	0.63
v_2	0.80	0.50	1.00	0.50	0.50	1.00	0.97	1.00
v_3	0.65	0.50	0.55	0.50	1.00	0.89	0.94	0.50
v_4	0.71	0.50	1.00	1.00	0.50	0.50	0.59	0.72
v_5	0.50	1.00	0.50	0.50	0.50	0.63	0.50	0.80

Tabela 6.10. Comparação de desempenho entre os algoritmos SOM, PSOM e aiNet quanto à capacidade de classificação correta do dados de treinamento do problema CLASSNLS, onde m é a dimensão final da rede (de Castro & Von Zuben, 1999b).

m	Classificação Correta (%)				
	6	7	8	9	10
SOM	79.0	74.0	80.5	76.5	82.5
PSOM	-----	79.5	-----	-----	83.0
aiNet	-----	-----	83.5	67.0	76.5

Para os mesmos dados de treinamento adotados no problema anterior, os resultados da aplicação do SOM a este problema estão ilustrados na Figura 6.33, assumindo $m = 8$. Note que não é possível inferir a quantidade de clusters determinados partindo da matriz-U. Sendo assim, para avaliar o desempenho do SOM treinado, aplicou-se os dados de treinamento à rede e verificou-se qual unidade de saída era a vencedora para cada padrão. As unidades que venceram mais para os padrões de determinada classe eram consideradas as unidades representativas daquela classe, permitindo avaliar o percentual de classificação correta do SOM.

**Figura 6.33.** Resultado da aplicação do SOM ao problema CLASSNLS. (a) Configuração final dos vetores de pesos e ligação com seus vizinhos mais próximos. (b) Matriz-U.

6.5.1.3. SPIR

O problema das duas espirais, denominado SPIR, está ilustrado na Figura 6.28(c). O conjunto de antígenos é composto por $M = 190$ elementos divididos em duas classes de mesma cardinalidade em um espaço de formas Euclidianas de dimensão $L = 3$. O objetivo deste problema é avaliar a capacidade da aiNet em distinguir clusters que não são linearmente separáveis, mas que apresentam uma distância relativa entre as classes. Os parâmetros de treinamento foram $[\mathbf{Ab}_m, \mathbf{S}] = \text{aiNet}(\mathbf{Ag}, 3, 50, 4, 0.1, 1, 0.08, 10)$.

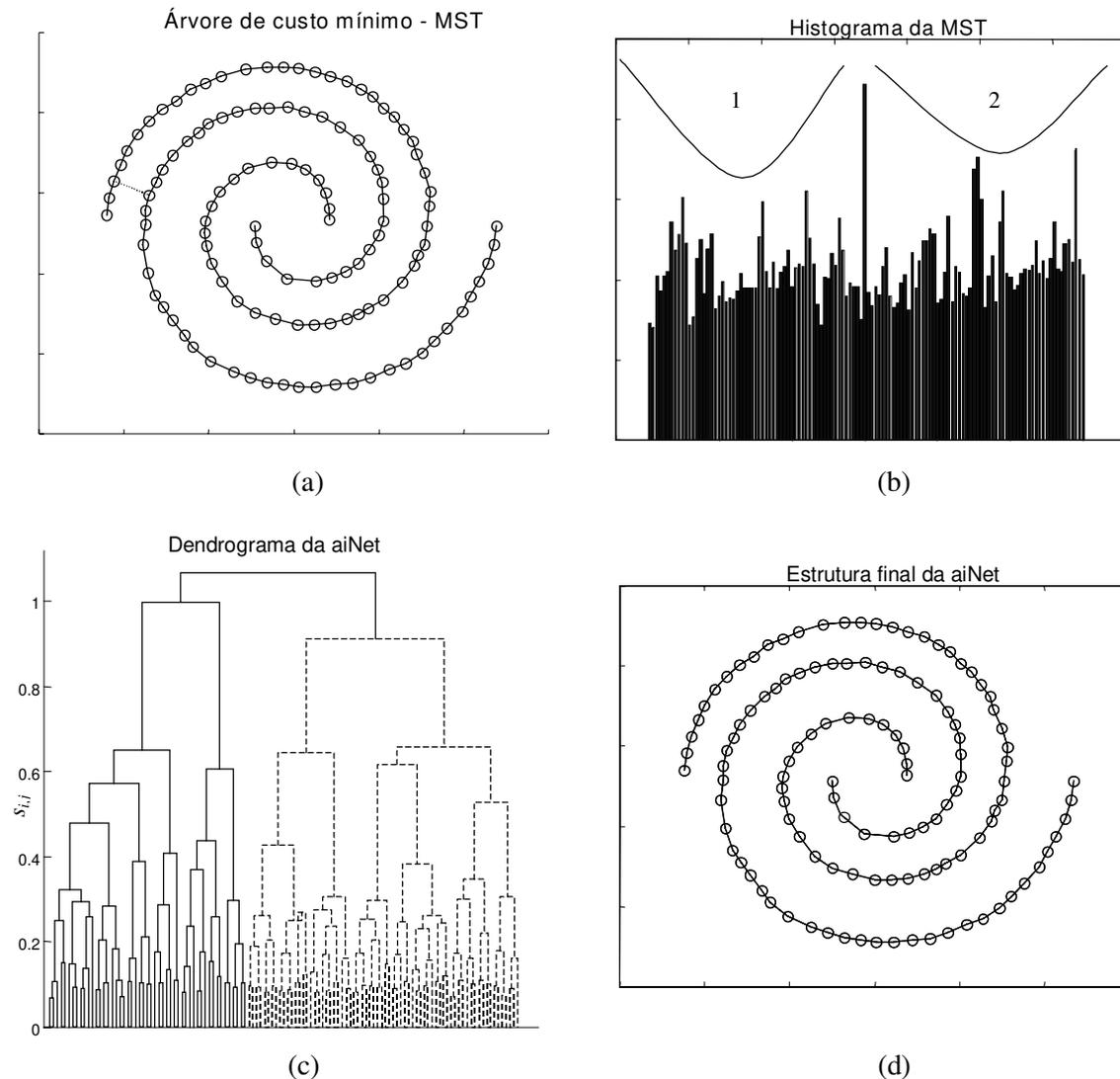


Figura 6.34. Resultados da aplicação da aiNet ao problema SPIR. (a) Árvore de custo mínimo (MST), onde a linha tracejada indica a conexão a ser removida. (b) Histograma da MST. (c) Dendrograma da rede indicando grandes diferenças entre os valores de fusão. (d) Arquitetura final da aiNet com dois clusters.

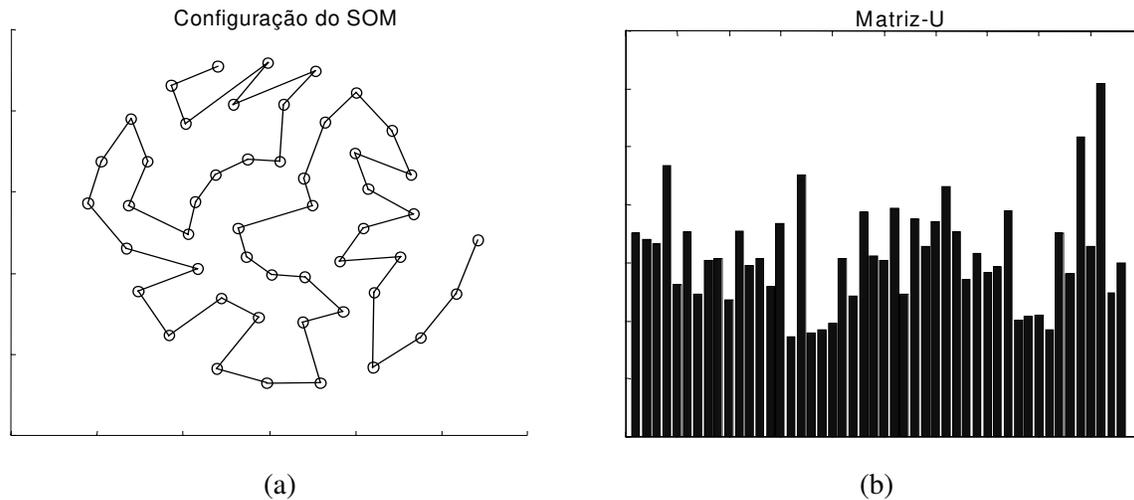


Figura 6.35. Resultado da aplicação do SOM ao problema SPIR. (a) Configuração final dos vetores de pesos e ligação com seus vizinhos mais próximos. (b) Matriz-U.

A Figura 6.34(a) e (b) apresenta a árvore de custo mínimo e seu respectivo histograma. Partindo do histograma da MST é possível identificar a existência de dois clusters de anticorpos na aiNet empregando-se um fator $f=2$. Neste caso, a rede resultante foi composta por $m = 121$ anticorpos, indicando uma taxa de compressão $CR = 36.32\%$. Note que a compressão foi maior em regiões do espaço onde a quantidade de antígenos similares é maior, ou seja, onde existe uma maior quantidade de informações redundantes. O dendrograma da aiNet também permite identificar dois grandes grupos (clusters) de anticorpos, como ilustrado na Figura 6.34(c).

A Figura 6.35 apresenta o resultado da aplicação do SOM com $m = 46$ ($m_0 = 50$) neurônios na camada de saída ao problema SPIR. A configuração final do SOM e sua respectiva matriz-U não nos permitem concluir nada a respeito da quantidade de clusters existente no conjunto amostral.

6.5.1.4. CHAINLINK

1000 dados amostrais em um espaço \mathcal{R}^3 foram arranjados sob a forma de dois anéis entrelaçados, dos quais um deles segue na direção x - y e o outro na direção x - z . Os dois anéis podem ser interpretados como pertencentes a uma corrente, cada qual composta por 500 amostras. Os dados são produzidos por um gerador de números aleatórios dentro de dois toróides de raio $R = 1.0$ e $r = 0.1$, de acordo com a Figura 6.28(d).

A análise estatística dos dados mostra que eles possuem as seguintes propriedades: seus diferentes componentes (x,y,z) não estão correlacionados (utilizando o coeficiente de correlação de Pearson) e a distribuição de cada componente pode ser vista como normal. Uma análise de componentes principais mostra que não existe nenhuma dimensão mais representativa do que as demais. Além disso, não existe nenhum hiperplano que possa

separar os dados corretamente em dois subconjuntos, ou seja, eles são não-linearmente separáveis. Por outro lado, da forma pela qual os dados foram criados, é possível identificar claramente duas classes distintas, por simples inspeção visual. A distância entre os dados amostrais de cada classe é uma ordem de magnitude menor do que a distância entre as classes (Ultsch, 1995).

Os parâmetros de treinamento foram $[Ab_m, S]=aiNet(Ag, 3, 5, 4, 0.1, 1, 0.15, 10)$. A Figura 6.36(a) e (b) apresenta a árvore de custo mínimo resultante e seu histograma correspondente. Partindo do histograma da MST é possível determinar de forma contundente os dois clusters da rede, utilizando-se um fator $f = 2$.

Note que, neste caso, a avaliação dos valores de fusão (tamanho do passo) do dendrograma da aiNet (Figura 6.36(c)) torna-se difícil, podendo levar a interpretações incorretas. A taxa de compressão foi de $CR = 94.5\%$ ($m = 55$).

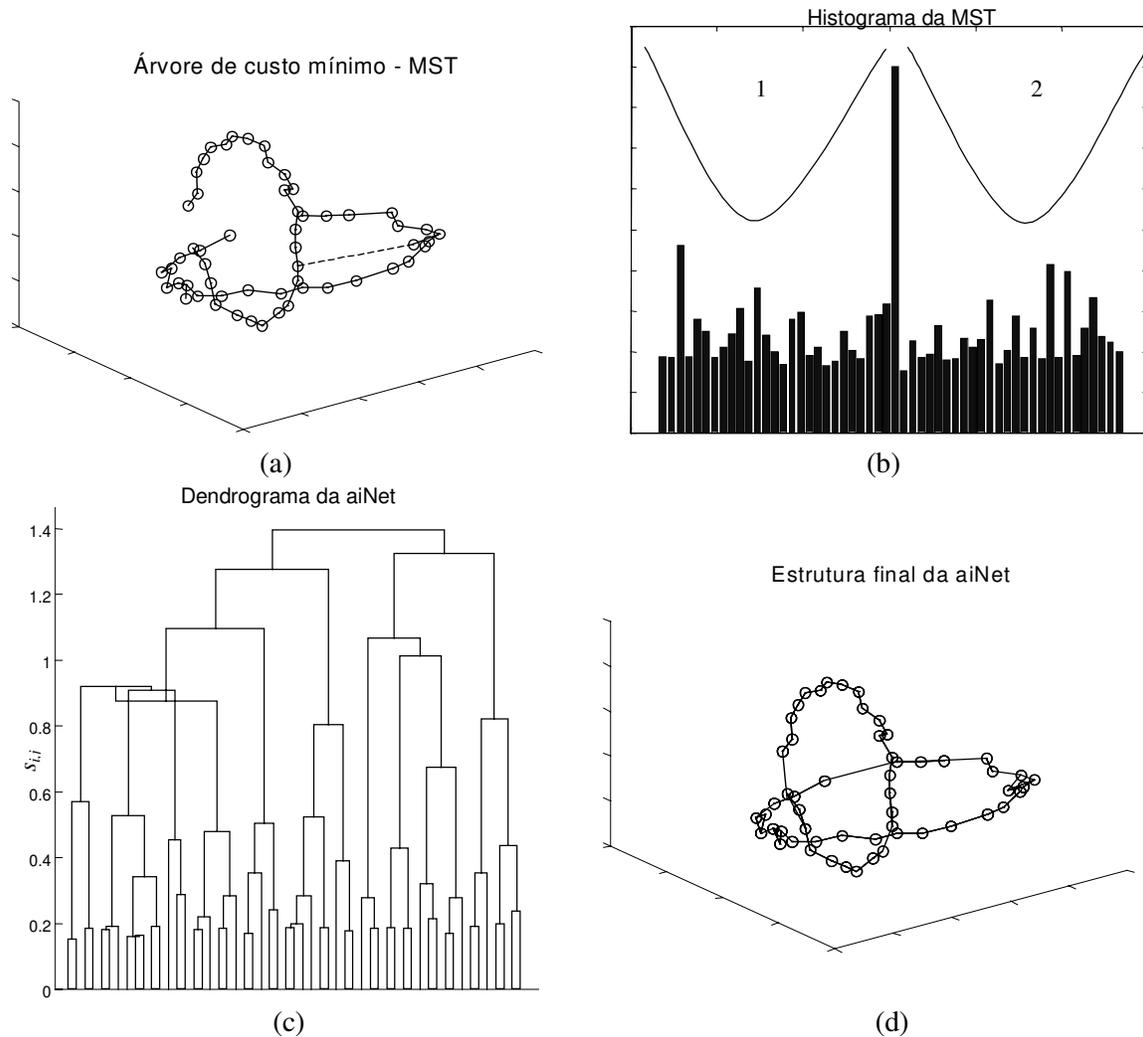


Figura 6.36. Aplicação da aiNet ao problema CHAINLINK. (a) Árvore de custo mínimo (MST), onde a linha tracejada indica a conexão a ser removida. (b) Histograma da MST. (c) Dendrograma da rede. (d) Arquitetura final da aiNet com dois clusters.

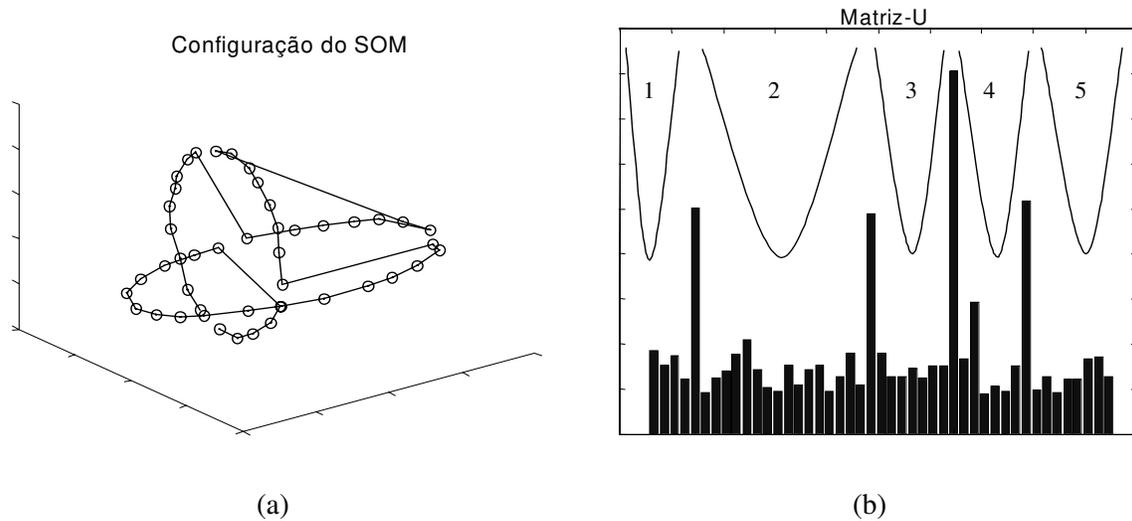


Figura 6.37. Resultado da aplicação do SOM ao problema CHAINLINK. (a) Configuração final dos vetores de pesos e ligação com seus vizinhos mais próximos. (b) Matriz-U.

O SOM foi aplicado na solução deste problema com um grid de saída composto por $m = 46$ neurônios ($m_0 = 50$). Os resultados estão apresentados na Figura 6.37. Para este problema, a matriz-U foi composta por 5 vales, indicando 5 clusters distintos. Note que estes 5 clusters podem ser obtidos desenhando-se 5 hiperplanos separando as partes dos anéis nos neurônios correspondentes aos picos da matriz-U.

6.5.2. Projeto Automático de Redes Neurais de Funções de Base Radial

Como discutido na Seção 4.2.2.1, uma das principais limitações para o funcionamento apropriado de uma rede neural com funções de ativação de base radial, ou simplesmente RBF, está na determinação dos parâmetros das RBFs, inclusive a quantidade de neurônios da rede. Juntamente com um algoritmo para a determinação automática da dispersão das funções de base radial, o algoritmo de treinamento da aiNet foi empregado na determinação da quantidade e posicionamento da base destas funções, dando origem ao algoritmo denominado ICS (*immunity-based center selection*), cujos resultados foram apresentados em de Castro & Von Zuben (2001b,e) e serão reproduzidos nesta tese.

Cada anticorpo de memória resultante do treinamento da aiNet corresponde ao centro de uma RBF da rede neural, portanto, resta-nos apenas descrever o algoritmo empregado na determinação da dispersão destes centros definidos pela aiNet. Trata-se de uma abordagem não-supervisionada que pode ser descrita como a seguir.

Seja ρ_j a dispersão associada ao centro \mathbf{c}_j , $j = 1, \dots, m_1$, localizado pelo algoritmo de treinamento da aiNet. Neste caso, assumimos que a RBF é composta por funções Gaussianas descritas como na Equação (6.6):

$$g(\|\mathbf{x} - \mathbf{c}_j\|^2) = \exp\left(-\frac{1}{2} \frac{\|\mathbf{x} - \mathbf{c}_j\|^2}{\rho_j^2}\right) \quad j = 1, \dots, m_1. \quad (6.6)$$

onde \mathbf{x} é o vetor de entrada (antígeno).

Para definir o processo iterativo de atualização dos centros determinados, foi empregado um algoritmo não-supervisionado semelhante ao descrito na Seção 4.2.2.1.2:

1. *Inicialização*: Inicialize as dispersões $\rho_j, j = 1, \dots, m_1$, com o valor das distâncias dos centros posicionados \mathbf{c}_j em relação ao seu padrão de entrada de maior afinidade $\mathbf{x}_i, i = 1, \dots, M$.
2. *Amostragem*: Selecione uma amostra de treinamento \mathbf{x} aleatoriamente.
3. *Afinidade*: Determine o índice k do centro com maior afinidade em relação ao vetor de entrada \mathbf{x} , de acordo com a medida de distância

$$d_k = \|\mathbf{x}(t) - \mathbf{c}_k(t)\|, \tag{6.7}$$

$$k(\mathbf{x}) = \arg \min_k d_k, k = 1, \dots, m_1,$$

onde $\mathbf{c}_k(t)$ é o centro da k -ésima RBF e d_k sua distância ao padrão de entrada $\mathbf{x}(t)$, ambos na iteração t .

4. *Atualização*: Ajuste os centros das RBFs através das seguintes expressões

$$\text{Se } d_k < \rho_k, \text{ então } \rho_k(t+1) = \rho_k(t) - \gamma(t) \|\mathbf{x}(t) - \mathbf{c}_k(t)\|, k = k(\mathbf{x}), \tag{6.8}$$

$$\text{Senão, se } d_k > \rho_k, \text{ então } \rho_k(t+1) = \rho_k(t) + \lambda(t) \|\mathbf{x}(t) - \mathbf{c}_k(t)\|, k = k(\mathbf{x}),$$

onde $\gamma(t)$ e $\lambda(t)$ são as taxas de aprendizagem para as dispersões das RBFs na iteração t . A taxa γ é inicializada com um valor próximo da unidade e decresce geometricamente ao longo das iterações até que um valor mínimo γ_{\min} seja atingido.

5. *Ciclo*: repita os Passos 2 a 5 até que $\gamma(t) \leq \gamma_{\min}$.

Nas nossas implementações, $\lambda(t) = \gamma_{\min}/\gamma(t)$, $\gamma(0) = 0.9$, $\gamma_{\min} = 10^{-3}$ e, a cada 5 iterações, o valor de γ é decrescido geometricamente por um fator 0.95.

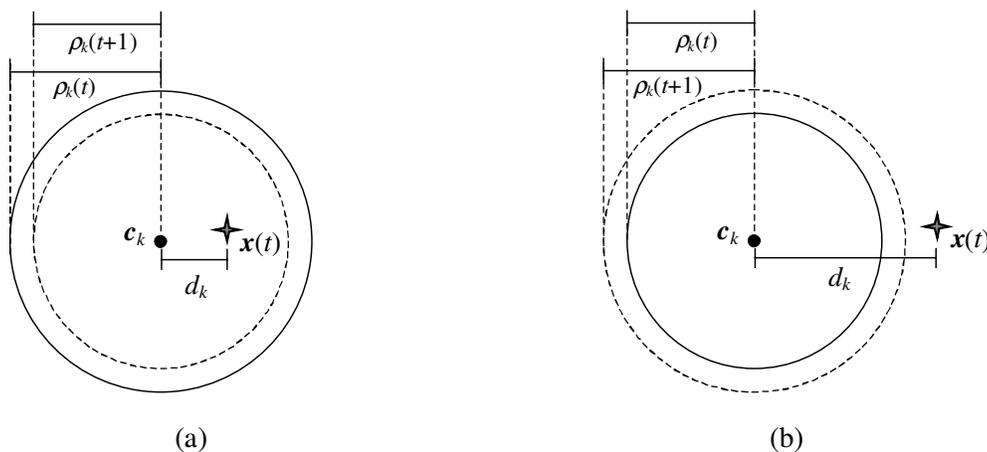


Figura 6.38. Determinação iterativa das dispersões das RBFs. (a) Redução da dispersão. (b) Aumento da dispersão.

Tabela 6.11. Parâmetros de treinamento da aiNet para a determinação de centros de RBFs.

	σ_s	σ_d	n	β	$\zeta(\%)$	gen
<i>sin(x)</i>	0.50	1.0	4	10	20	5
CLASS	0.15				10	10
SPIR	0.80				10	5
IRIS	0.20				20	10

Esta regra de atualização para o vetor ρ de dispersões pode ser explicada da seguinte forma. Se a dispersão ρ_k do centro c_k de maior afinidade ao padrão $\mathbf{x}(t)$ é menor do que a distância d_k , Equação (6.8), então sua dispersão ρ_k pode ser reduzida, senão sua dispersão ρ_k deve ser aumentada para que o padrão de treinamento seja apropriadamente classificado. Este processo está ilustrado na Figura 6.38.

Com o objetivo de avaliar o desempenho do algoritmo, ele foi aplicado a diversos problemas de regressão e classificação: *sin(x)*, CLASSNLS, SPIR e IRIS. Seus resultados foram comparados com o método de inicialização aleatória, apresentado na Seção 4.2.2.1.1, com o método *k*-means, apresentado na Seção 4.2.2.1.2 e com alguns resultados da literatura. Os parâmetros de entrada da aiNet estão resumidos na Tabela 6.11. Os resultados a serem apresentados são os valores médios, obtidos a partir de 10 simulações computacionais, e o desvio padrão ($\pm\delta$) em relação a esta média.

6.5.2.1. Função *sen(x)*

Trata-se de um problema de aproximação de funções, com $M = 100$ amostras de treinamento para um período da função *sen(x)*. Estas amostras foram geradas com ruído uniformemente distribuído no intervalo $[-0.7, 0.7]$. A função *sen(x)* e os dados utilizados para a avaliação do algoritmo ICS estão apresentados na Figura 6.39(a). A Figura 6.39(b) apresenta o desempenho (aproximação e posição dos centros) dos três algoritmos aplicados a este problema. A quantidade média de centros determinados pela aiNet foi $m_1 = 6.8 (\pm 0.8)$, correspondente a uma taxa média de compressão $CR = 97.84\%$. As aproximações apresentadas na Figura 6.39(b) foram obtidas para $m_1 = 6$ centros.

Se os centros determinados pelo método de inicialização aleatória estiverem aproximadamente uniformemente distribuídos no espaço de entradas, então esta estratégia pode resultar em uma boa aproximação, caso contrário, a aproximação pode ser polarizada em relação a alguns dados de entrada, como pode ser visto na Figura 6.39(b). Observando esta figura, é possível verificar que o método *k*-means foi capaz de realizar uma aproximação semelhante à do algoritmo ICS, porém com a necessidade de se definir a priori a quantidade m_1 de centros e assumir que todos as funções de base radial possuem a mesma dispersão ρ , dada pela Equação 4.13.

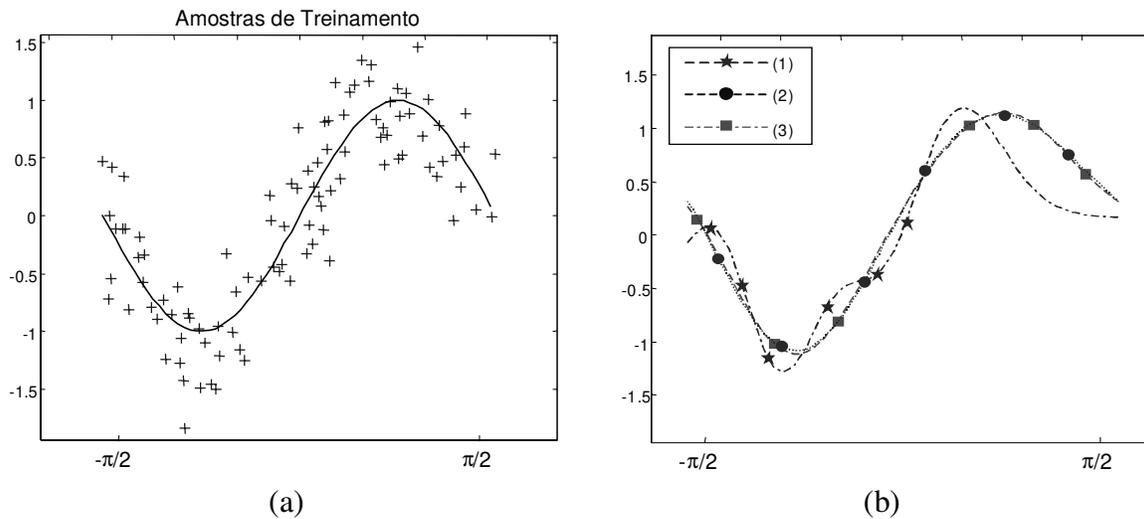


Figura 6.39. Aproximação da função $sen(x)$. (a) 100 amostras de treinamento (+) e função $sen(x)$ definida no intervalo $[-\pi, +\pi]$ (linha sólida). (b) Desempenho dos algoritmos: (1) seleção aleatória; (2) k -means e (3) ICS.

O algoritmo proposto permitiu determinar $\rho = [0.88, 1.38, 0.92, 1.14, 1.02, 1.04]$ como sendo o vetor de dispersões das RBFs, enquanto a Equação 4.13 produziu $\rho = 1.63$ e $\rho = 1.52$ para a inicialização aleatória e via k -means, respectivamente. Orr (1995) obteve uma boa regularização para $m_1 = 3$ RBFs na rede, enquanto o algoritmo proposto por Chen *et al.* (1995) selecionou $m_1 = 15$ RBFs. No caso do método ICS, a quantidade de RBFs é controlada por σ_s , porém existe uma relação entre o valor escolhido para σ_s e a qualidade da solução. Por exemplo, com $\sigma_s = 1.0$, uma rede com $m_1 = 3$ pode ser obtida, porém com um baixo nível de aproximação. Resultados satisfatórios para o algoritmo proposto ocorrem para $m_1 \geq 5$.

6.5.2.2. CLASSNLS

Este problema foi descrito na Seção 6.5.1.2, Figura 6.28(b). A diferença é que, enquanto lá os rótulos dos dados não foram usados em nenhum momento, aqui eles não serão usados no posicionamento dos centros e na definição das dispersões mas serão usados no ajuste dos pesos de saída da rede neural. A Figura 6.40 apresenta a posição dos centros e as superfícies de decisão determinadas pelos três algoritmos aplicados ao problema CLASSNLS. A quantidade média de centros determinados pela aiNet foi $m_1 = 6.6 (\pm 1.4)$, correspondente a uma taxa média de compressão $CR = 96.67\%$. Foram utilizados $m_1 = 7$ RBFs para os métodos de inicialização aleatória e via k -means. Note que a inicialização aleatória pode resultar em uma quantidade de clusters menor do que o número real de classes existente no conjunto amostral, como pode ser observado na Figura 6.40(b).

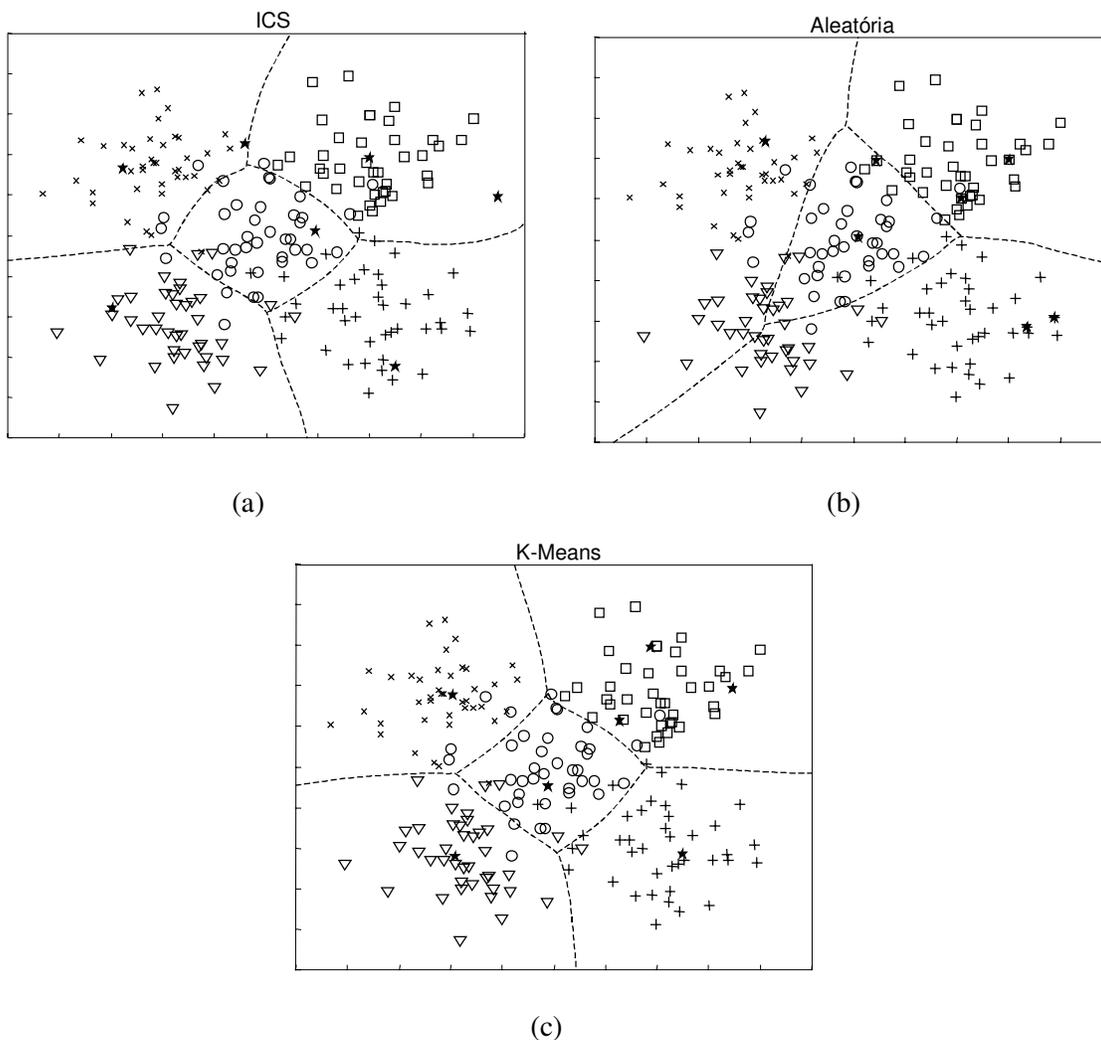


Figura 6.40. Posição dos centros (estrelas em preto) e superfícies de decisão para o problema CLASSNLS para $N = 7$. (a) ICS. (b) Inicialização aleatória. (c) k -means.

Este problema foi utilizado na literatura para avaliar a capacidade de classificação de algoritmos não-supervisionados (de Castro & Von Zuben, 1999b), onde um erro de classificação da ordem de 17% foi verificado, enquanto o método ICS apresenta um erro médio de 9.5%. Entretanto, é importante salientar que a RBF resultante opera de forma supervisionada, resultando, geralmente, em erros de classificação inferiores aos obtidos pelos métodos não-supervisionados.

6.5.2.3. SPIR

Este problema foi descrito na Seção 6.5.1.3, Figura 6.28(c). A diferença é que, enquanto lá os rótulos dos dados não foram usados em nenhum momento, aqui eles não serão usados no posicionamento dos centros e na definição das dispersões mas serão usados no ajuste dos pesos de saída da rede neural. A Figura 6.41 apresenta a posição dos centros e superfícies

de decisão determinadas pelos três algoritmos aplicados ao problema SPIR. A quantidade média de centros determinados pela aiNet foi $m_1 = 52.7 (\pm 3.1)$, correspondente a uma taxa média de compressão $CR = 72.26\%$. Para efeito de comparação, foram utilizados $m_1 = 52$ RBFs para os métodos de inicialização aleatória e via k -means. Novamente o método de inicialização aleatória pode resultar em uma classificação incorreta (Figura 6.41(b)), enquanto o algoritmo ICS sempre resulta em uma classificação correta para os parâmetros de treinamento adotados. Neste caso, foi possível verificar que o algoritmo k -means pode resultar em centros posicionados em regiões do espaço sem nenhum dado, como ilustrado na Figura 6.41(c).

O problema SPIR já foi utilizado na literatura para avaliar a capacidade de novos algoritmos de determinação automática de arquiteturas de redes do tipo RBF. Uma versão supervisionada da rede construtiva proposta por Fritzke (1994a) foi aplicada na solução do problema SPIR (Fritzke, 1994b). Nesta aplicação, o autor utilizou $M = 194$ amostras de treinamento o que resultou em uma rede RBF com $m_1 = 145$ unidades intermediárias, uma quantidade muito superior à média $m_1 = 52.7 (\pm 3.1)$ obtida pelo algoritmo ICS.

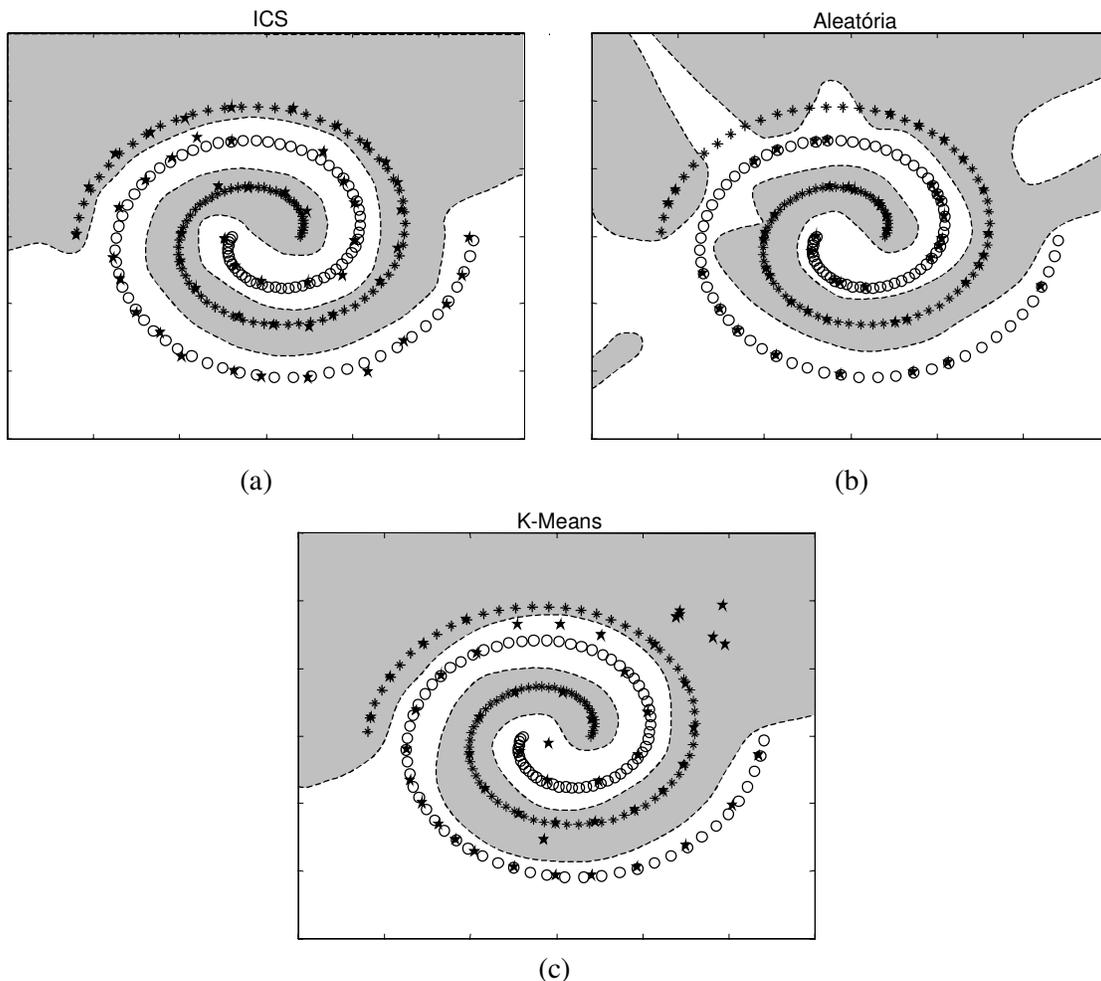


Figura 6.41. Posição dos centros e superfícies de decisão para o problema SPIR, $N = 52$. (a) ICS. (b) Inicialização aleatória. (c) k -means.

6.5.2.4. IRIS

Para testar a capacidade de generalização do algoritmo ICS, foi utilizado o problema IRIS (URL 3). Este conjunto de dados é composto por 150 amostras distribuídas em três classes distintas, onde apenas uma delas é linearmente separável das demais. Todo o conjunto amostral foi particionado para gerar conjuntos de treinamento e teste.

As simulações foram executadas considerando-se 10 partições distintas para cada tamanho do conjunto de treinamento, com 10% a 50% dos dados empregados exclusivamente para treinamento, e o restante para a avaliação da capacidade de generalização da rede. A Tabela 6.12 apresenta o erro médio de classificação obtido considerando cada uma das três estratégias: ICS, inicialização aleatória e via k -means. A quantidade média de RBFs determinada pelo algoritmo ICS também é apresentada. Os outros dois métodos requerem este parâmetro como dado de entrada.

Existem vários resultados na literatura referentes à aplicação de algoritmos de treinamento modificados ao problema da IRIS (Karayiannis & Mi, 1997; Jayasuriya & Halgamuge, 1998). A grande maioria destes algoritmos são construtivos, ou seja, a dimensão da rede RBF é determinada durante o processo de aprendizagem. A Tabela 6.13 resume os melhores desempenhos dos melhores resultados apresentados na literatura (maiores detalhes podem ser encontrados na literatura citada) contrastando-os com o melhor desempenho obtido pelo método ICS proposto.

Tabela 6.12. Classificação correta (valores médios e desvio padrão) dos algoritmos (ICS, Inicialização Aleatória e k -means) para o problema IRIS, onde Etr é o erro médio ($\pm\delta$) para o conjunto de treinamento e Ete é o erro médio ($\pm\delta$) para o conjunto de teste. Ntr é a quantidade de amostras de treinamento.

Ntr	ICS		Aleatória		k -means		m_1
	Etr (%)	Ete (%)	Etr (%)	Ete (%)	Etr (%)	Ete (%)	
75	3.33 \pm 2.61	4.67 \pm 2.20	5.07 \pm 1.97	4.27 \pm 1.51	3.60 \pm 2.81	5.87 \pm 3.68	9.40 \pm 1.90
90	3.00 \pm 1.29	3.83 \pm 2.61	6.11 \pm 2.11	4.83 \pm 4.12	4.22 \pm 3.26	5.17 \pm 2.77	9.11 \pm 1.62
105	3.71 \pm 1.58	4.44 \pm 4.06	4.76 \pm 1.74	5.77 \pm 5.05	4.38 \pm 2.25	5.56 \pm 3.19	8.70 \pm 1.64
120	3.00 \pm 0.98	3.33 \pm 1.57	7.58 \pm 9.72	9.33 \pm 11.3	4.58 \pm 1.58	4.66 \pm 5.71	8.80 \pm 1.48
135	3.04 \pm 1.46	1.33 \pm 2.81	4.66 \pm 4.54	6.00 \pm 12.8	3.48 \pm 1.40	2.67 \pm 3.44	11.4 \pm 2.99

Tabela 6.13. Melhor desempenho do algoritmo ICS aplicado ao problema IRIS e comparado com os melhores resultados dos melhores métodos da literatura (Karayiannis & Mi, 1997 – Tabela II; Jayasuriya & Halgamuge, 1998 – Tabela V), para Ntr = 75.

Metódo	Etr (%)	Ete (%)	m_1
ICS	0.0	0.0	7
MSBFN	-----	2.9	5
ELEANNE 2	0.0	1.3	11
Gradient descent	1.3	1.3	7

6.5.3. Análise de Sensibilidade

Do Passo 1.1.1 ao Passo 1.1.7 do algoritmo de treinamento da aiNet, descrito na Seção 5.5.2, o algoritmo CLONALG é reproduzido, portanto a análise de sensibilidade do CLONALG em relação aos parâmetros n , N_c e d também se aplica à aiNet. Resumidamente (ver Seção 6.2.3), foi verificado que n é diretamente proporcional ao valor médio da afinidade dos anticorpos em relação aos antígenos, enquanto N_c implica uma maior velocidade de convergência do algoritmo e d aumenta a capacidade de exploração do espaço de formas. Sendo assim, será feita uma análise de sensibilidade da aiNet em relação aos parâmetros da rede imunológica, como σ_s , σ_d e ζ . Além disso, a relação entre n e a quantidade final m de anticorpos de memória na aiNet também será verificada.

Na Seção 6.5.1 a aiNet foi aplicada a diversos problemas de teste objetivando ilustrar sua adequação e potencialidade. Para que a aiNet possa ser aplicada a qualquer problema, vários parâmetros devem ser definidos pelo usuário, destacando:

- σ_s : limiar de supressão;
- σ_d : limiar de mortalidade natural;
- n : quantidade de anticorpos a serem selecionados para clonagem; e
- ζ : percentual de anticorpos maduros a serem re-selecionados.

A Figura 6.42 apresenta a relação entre o limiar de supressão σ_s e a quantidade final m de anticorpos de memória na aiNet para os problemas SPIR e CHAINLINK. Como discutido na Seção 5.5.2, σ_s controla a quantidade de anticorpos na rede sendo responsável por sua plasticidade. Valores elevados para σ_s indicam anticorpos mais generalistas, enquanto valores pequenos para σ_s resultam em anticorpos altamente específicos. Este parâmetro é crítico, pois uma escolha inadequada pode resultar em classificações incorretas.

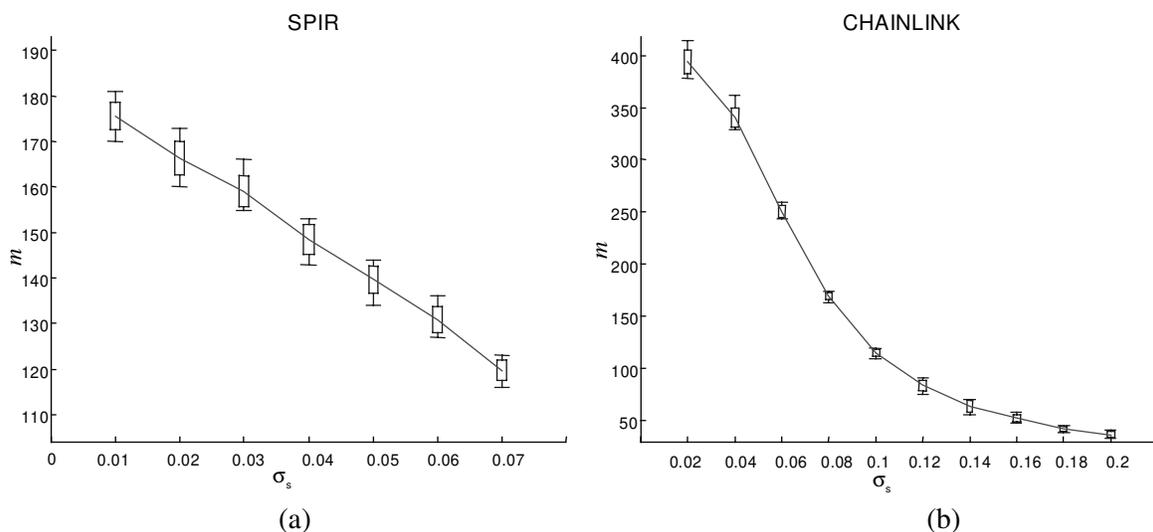


Figura 6.42. Relação entre a quantidade final de anticorpos na aiNet (N) e o limiar de supressão (σ_s). Os resultados são os valores máximo, mínimo, médio e o desvio padrão de 10 simulações para os problemas SPIR e CHAINLINK.

Tabela 6.14. Quantidade de anticorpos N_{Ab} que sofreram mortalidade natural (poda) na primeira geração do algoritmo de treinamento da aiNet para o problema CLASSLS em 10 simulações.

	1	2	3	4	5	6	7	8	9	10	Estatística
N_{Ab}	74	87	73	60	81	54	55	58	62	77	68.1±11.7

Se algum conhecimento sobre a distribuição dos dados (ou classes) do problema é conhecido, ele pode ser empregado na determinação de σ_s . Para os problemas testados, $\sigma_s = 0.2$, $\sigma_s = 0.2$, $\sigma_s = 0.08$ e $\sigma_s = 0.2$, respectivamente, resultaram em bons desempenhos do algoritmo. Valores acima destes causaram classificações incorretas em algumas simulações.

O limiar de morte natural (σ_d) é responsável pela eliminação de anticorpos cuja afinidade antigênica é baixa. Sem perda de generalidade, considerando o primeiro problema, CLASSLS (Seção 6.5.1.1), é possível avaliar a relevância deste parâmetro para a aprendizagem da rede. A Tabela 6.14 apresenta a quantidade de anticorpos que sofreram mortalidade natural durante a primeira geração do algoritmo de treinamento. Os resultados são apresentados para 10 simulações distintas. Em todos os casos, a mortalidade natural ocorreu apenas durante a primeira geração. Nas gerações subsequentes o parâmetro σ_d não foi responsável pela morte de nenhum anticorpo. Este resultado permite-nos concluir que os processos de maturação de afinidade e forte pressão seletiva fazem com que o algoritmo seja capaz de absorver rapidamente (em apenas uma geração) as características mais relevantes do conjunto antigênico.

Para estudar a sensibilidade da aiNet em relação aos parâmetros n e ζ , foi escolhido um valor para o limiar de supressão tal que a rede resultante apresentasse aproximadamente $m = 50$ anticorpos para o problema CLASSLS. O objetivo é verificar a capacidade da aiNet em reproduzir (aprender) o conjunto antigênico através da definição de um critério de parada que avalie a distância média (ou erro médio) entre os anticorpos da rede e os antígenos, cujo valor foi escolhido igual a 10^{-2} . Sendo assim, adotou-se $\sigma_s = 0.01$.

A Figura 6.43(a) apresenta a relação entre n e m e a Figura 6.43(b) ilustra a relação entre n e a quantidade de gerações necessária para convergência (N_{gen}). É possível verificar que quanto maior n , maior a dimensão final da aiNet (maior m), indicando que n tem uma influência direta na plasticidade da rede. Por outro lado, da Figura 6.43(b), é possível verificar que quanto maior n , menor a quantidade de gerações necessária para convergência (aprendizagem do conjunto antigênico). Os resultados apresentados são os valores máximo, mínimo e médio obtidos a partir de 10 simulações.

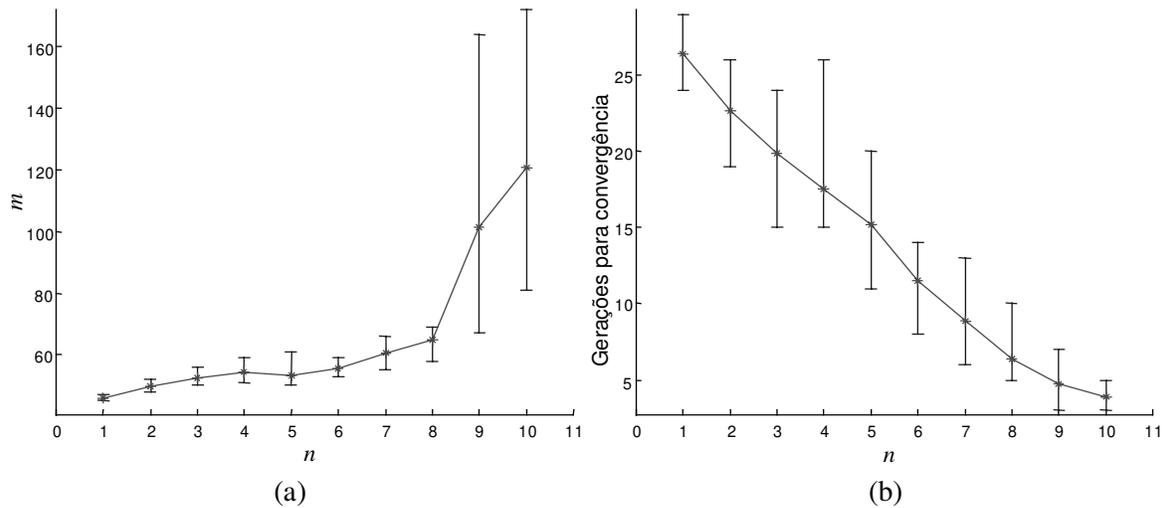


Figura 6.43. Sensibilidade da aiNet em relação à quantidade n de anticorpos a serem selecionados para clonagem. Valores máximo, mínimo e média tomadas de 10 simulações. (a) Relação $n \times m$. (b) Relação $n \times N_{\text{gen}}$.

Para estudar a sensibilidade da aiNet em relação ao parâmetro ζ , foi utilizado o mesmo critério de parada e limiar de supressão anteriores, variando-se ζ de 2% a 24% em intervalos de 2%. A Figura 6.44 apresenta a relação entre ζ , m e a quantidade de gerações para convergência (média de 10 simulações). Desta figura, é possível perceber que ζ não tem grande influência na dimensão final da rede, mas valores altos de ζ implicam em convergência mais lenta.

Intuitivamente poderíamos imaginar que quanto maior o valor de ζ mais rápida deveria ser a convergência da aiNet. Entretanto, diversos fatores devem ser considerados. Primeiramente, o critério de parada empregado é a diferença (distância) entre os anticorpos da rede e os antígenos a serem reconhecidos. Neste caso, dado um limiar de supressão $\sigma_s = 0.01$, a quantidade m de anticorpos de memória gerados é aproximadamente igual à quantidade de antígenos a serem reconhecidos ($m \approx M$). Portanto, existe aproximadamente um anticorpo de memória servindo como imagem interna para cada antígeno. Nesta situação, um valor de $\zeta > 2\%$ implica que, para cada antígeno, mais de um clone maduro está sendo substituído do conjunto de memória, portanto, enquanto o antígeno apresentado na iteração atual está sendo melhor aprendido, um elemento de memória com alta afinidade a algum outro antígeno está sendo “perdido”, reduzindo a velocidade de convergência do algoritmo para o critério de parada adotado.

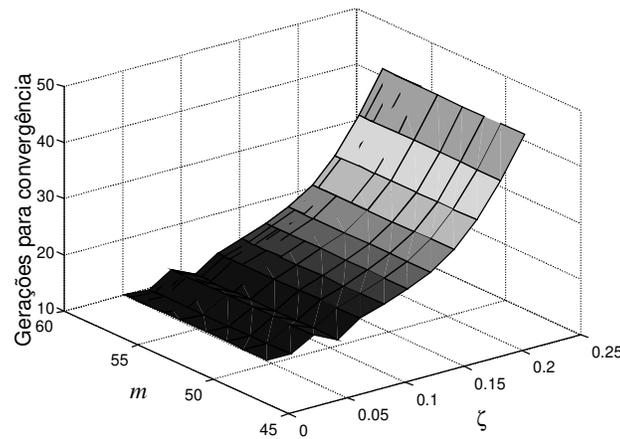


Figura 6.44. Relação entre ζ , m e a quantidade de gerações para convergência (média de 10 simulações).

6.5.4. Discussão

Os objetivos principais da aiNet foram definidos como sendo primeiramente a capacidade automática de extrair informações de um conjunto de dados (antígenos) através da construção de imagens internas, e reduzir a redundância do conjunto de anticorpos da rede dada a interação dos elementos do modelo de rede imunológica. A utilização de anticorpos de mesma dimensão que os antígenos permite manter a topologia original das classes, que é geralmente perdida quando alguma técnica de redução de dimensionalidade (como o SOM) é empregada. A aiNet permite a manutenção de relações de vizinhança em um espaço L -dimensional.

Os principais objetivos do SOM são reduzir a dimensionalidade dos dados preservando uma métrica e as relações de vizinhança entre os padrões de entrada (Kohonen, 1982). Com uma abordagem diferente, a aiNet reduz a redundância dos dados de treinamento permitindo a reconstrução métrica e das relações de vizinhança sem reduzir a dimensionalidade dos dados. Devido à reprodução das relações topológicas, informações similares (baseadas em uma métrica de distância) são mapeadas em anticorpos vizinhos (próximos), eventualmente o mesmo anticorpo, caracterizando uma quantização e clusterização do espaço de entrada.

A definição do limiar de supressão (σ_s) é crucial para a determinação da estrutura final da rede e, conseqüentemente, da quantidade e formas dos clusters definidos pela árvore de custo mínimo. A quantidade n de anticorpos com alta afinidade a serem selecionados para reprodução também demonstrou ser importante para o tamanho final da aiNet. Entretanto, este parâmetro, assim como vários outros, foi mantido fixo para a grande maioria das aplicações apresentadas. No caso dos problemas com uma quantidade maior de dados de treinamento, empregou-se $\zeta = 10\%$ para reduzir o tempo de processamento do algoritmo. A Tabela 6.11 apresenta valores default para alguns parâmetros de treinamento da aiNet como

σ_a , n , β e ζ . É importante mencionar também que em nenhum caso a aiNet demonstrou ser sensível à condição inicial, ou seja, ao repertório inicial de anticorpos, assumido de tamanho $N_0 = 10$ em todos os casos.

No caso da rede neural do tipo SOM, a definição a priori do tamanho do grid de saída impõe uma arquitetura de rede que nem sempre é capaz de mapear todas as interações importantes, presentes nos dados de entrada, nos nós de saída. Mesmo para o caso dos modelos com arquitetura dinâmica (Fritzke, 1994a; de Castro & Von Zuben, 1999b; Cho, 1997), se os clusters dos dados de treinamento não são representáveis em espaços de menor dimensão ou então apresentarem uma conformação não-radial, estes algoritmos apresentam dificuldades de clusterização. No caso da aiNet, desde que exista uma distância relativa entre as classes, uma classificação correta pode ser obtida em qualquer situação. Exemplos claros desta afirmação foram apresentados nas Seções 6.5.1.3 e 0. Outros exemplos de problemas que poderiam ser resolvidos pela aiNet, mas para os quais o SOM e suas variantes provavelmente falhariam estão apresentados na Figura 6.45.

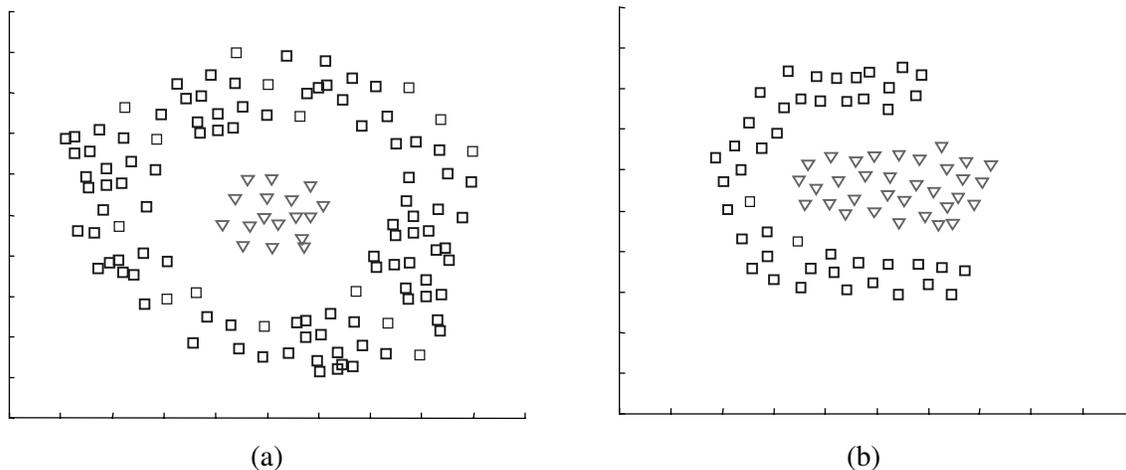


Figura 6.45. Exemplos de problemas que poderiam ser resolvidos pela aiNet, mas para os quais uma rede neural artificial competitiva como o SOM falharia na determinação correta dos clusters.

6.5.4.1. Projeto Automático de RBFs

Os resultados apresentados na Seção 6.5.2 mostram a dependência do algoritmo k -means em relação à escolha inicial dos centros. Pode ser observado na Figura 6.41(c) que a definição a priori da quantidade de RBFs a ser utilizada na rede neural pode resultar no desperdício de recursos computacionais, com centros posicionados em locais do espaço com baixa densidade de dados. Este tipo de comportamento não ocorre para o algoritmo proposto, pois ele elimina os centros que não estão mapeando nenhum dado de treinamento. Assim como de Castro & Von Zuben (1999b) propuseram um método de poda para redes do tipo SOM, uma metodologia similar poderia ser empregada para o k -means no sentido de eliminar unidades pouco representativas do conjunto amostral.

O algoritmo ICS gera um conjunto de funções de base radial que satisfazem o teorema de Micchelli (1986). Este algoritmo permite naturalmente a produção de uma solução regularizada para problemas de aproximação de funções, como pôde ser visto na Figura 6.39(b). Também pôde ser verificado nesta figura, que o algoritmo ICS não necessariamente determina os centros das RBFs uniformemente distribuídos, como geralmente é feito pelo k -means. O ICS distribui as RBFs de acordo com a função densidade de probabilidade dos dados de treinamento e ajusta a dispersão de cada função de forma que as amostras sejam apropriadamente classificadas ou aproximadas.

O problema IRIS foi empregado para avaliar a capacidade de generalização do método ICS e seu desempenho foi comparado satisfatoriamente com os melhores resultados encontrados na literatura.

CAPÍTULO 7

CONCLUSÕES E PERSPECTIVAS FUTURAS

As diversas implicações resultantes da introdução da engenharia imunológica como um novo paradigma computacional são discutidas neste capítulo. Como complemento, uma discussão sobre a tendência da área de sistemas imunológicos artificiais e a proposta de possíveis extensões deste trabalho e novas direções a serem tomadas fazem parte deste capítulo.

“ . . . partindo sempre de proposições prováveis, se atingirá uma conclusão também provável . . . Assim, o argumento mais incisivo é aquele que põe a sua conclusão em pé de igualdade com as proposições solicitadas” – Aristóteles

7.1. Introdução

Não é possível se afirmar até que ponto a ciência, mais especificamente a teoria de sistemas inteligentes, poderá alcançar no propósito de criar máquinas capazes de modelar e simular comportamentos complexos na presença de incertezas e inconsistências. Serão ainda necessários muitos anos de pesquisa para que se compreenda grande parte dos fenômenos (conscientes, inconscientes, cognitivos, instintivos, orgânicos, celulares, etc.) envolvidos nos processos biológicos, principalmente quando se procura agregá-los. No entanto, alguns aspectos dos sistemas naturais já podem ser implementados e utilizados para auxiliar ou substituir o homem na realização de tarefas. Mesmo se chegarmos à compreensão absoluta dos fatores envolvidos nestes processos, ainda haverá o desafio de propor mecanismos de formalização, e ainda desenvolver técnicas adequadas para viabilizar sua implementação. Esta tese reforça ainda mais a premissa de que os sistemas naturais (ou biológicos) constituem excelentes fontes inspiradoras para o desenvolvimento da ciência e tecnologia.

A partir do momento em que se detém um certo conhecimento de como funcionam alguns mecanismos naturais, como por exemplo, como se dá uma resposta imune adaptativa a um determinado agente infeccioso, tornam-se possíveis os processos de formulação matemática e implementação computacional destes procedimentos naturais. O mais interessante talvez, seja o fato de que algumas vezes um sistema, princípio, ou mecanismo, é modelado objetivando-se obter um certo desempenho e, sem que se preveja, um comportamento novo, diferente do planejado, emerge.

O desenvolvimento desta tese foi, certamente, uma experiência acadêmica ímpar. Coursar as disciplinas de Imunologia (BI 583) e Imunologia Celular (NM 140) no Departamento de Microbiologia e Imunologia (DMI) do Instituto de Biologia (IB) da Unicamp e interagir com especialistas de outra área de pesquisa foi muito gratificante. Por mais simples que

tenham sido, realizar experimentos *in vivo* foi uma experiência nova para mim, que vinha obtendo uma formação técnica em eletricidade, eletrônica e computação desde o ensino médio. Alguns processos observados experimentalmente no cotidiano dos biólogos podem apresentar uma riqueza de idéias muito grande para um engenheiro ou cientista da computação.

Baseado nesta experiência pessoal e nas evidências que se apresentam através do desenvolvimento das redes neurais artificiais, algoritmos evolutivos, e outros paradigmas de inteligência computacional, é possível afirmar que os sistemas naturais ainda têm muito à contribuir para a engenharia e vice-versa.

Por ser uma tese voltada para a área de engenharia, sua contribuição se restringiu a mecanismos de formalização e desenvolvimento de ferramentas computacionais para a solução de problemas de engenharia, embora seja possível crer que a biologia também possa fazer bom uso não apenas das ferramentas aqui propostas, mas também de um ponto de vista mais voltado para as ciências exatas, que diverge (às vezes não tão sutilmente) do ponto de vista dos biólogos. Uma interação maior precisa, e pode, existir.

7.2. Contribuições

O principal propósito desta tese era o de utilizar idéias extraídas dos sistemas imunológicos biológicos e artificiais para o desenvolvimento de novas ferramentas computacionais a serem empregadas na solução de problemas de engenharia. Estas ferramentas deveriam apresentar um incremento significativo em pelo menos dois aspectos: autonomia e adaptabilidade.

Dentre as principais contribuições desta tese, a serem listadas a seguir, destaca-se o pioneirismo no Brasil e no mundo em praticamente todas elas:

- Proposição de um novo paradigma de computação, denominado *engenharia imunológica*, como uma nova linha de pesquisa dentro dos sistemas imunológicos artificiais (Capítulo 3);
- Apresentação de um estudo histórico dos sistemas imunológicos artificiais (Capítulo 3);
- Uma das primeiras tentativas de se apresentar uma estrutura formal genérica para os sistemas imunológicos artificiais (Capítulo 3);
- O primeiro esforço no sentido de concatenar a grande maioria dos trabalhos em SIA, feita em duas partes: uma inicial na forma de um Relatório Técnico (RT DCA 02/00) e a outra no Capítulo 3;
- Iniciativa de correlacionar os SIA com outras abordagens de sistemas inteligentes (Capítulo 4);
- Discussão sucinta das relações entre o sistema imunológico e o sistema nervoso central, tentando salientar como as ciências que estudam estes sistemas podem se beneficiar uma da outra (Capítulo 4);

- Discussão de como a teoria da evolução pode ser estudada dentro do sistema imunológico e qual a influência do sistema imunológico na evolução das espécies (Capítulo 4);
- Proposição de quatro novas ferramentas computacionais fortemente fundamentadas em conceitos e teorias imunológicas, com aplicações às mais diversas áreas, como reconhecimento e classificação de padrões, análise de dados e otimização (Capítulos 5 e 6);
- Apresentação de soluções alternativas para problemas de engenharia através das novas ferramentas computacionais desenvolvidas (Capítulo 6); e
- Sugestão de diversas extensões para a própria tese e para os sistemas imunológicos artificiais (Capítulo 7).

7.3. Perspectivas

Antes de começar a listar um série de possíveis extensões da tese e tendências da área de sistemas imunológicos artificiais, gostaria de salientar que, devido principalmente à insipiência da área, muitas das propostas a serem apresentadas podem ainda estar distantes da factibilidade.

Como toda nova área de atuação, os sistemas imunológicos artificiais, mais especificamente a engenharia imunológica, apresenta uma vasta lista de tópicos a serem desenvolvidos e aperfeiçoados. Sem nos restringirmos apenas à EI, é possível dividir as perspectivas futuras desta tese em três grandes grupos: 1) propostas de novos algoritmos inspirados nos sistemas imunológicos biológicos, 2) variações a serem introduzidas nos algoritmos propostos, e 3) especulações sobre o futuro da área.

7.3.1. Novos Algoritmos Inspirados em Mecanismos Imunológicos

No Capítulo 4, foi discutida a relação entre os sistemas imunológicos biológicos, a engenharia imunológica e diversas outras abordagens de sistemas inteligentes. A apresentação da computação molecular, por exemplo, permite a proposição de uma implementação molecular dos vários algoritmos de SIA apresentados. O processamento massivamente paralelo da computação molecular pode oferecer uma alternativa capaz de aliviar o custo computacional dos algoritmos de SIA. Além disso, a representação molecular dos algoritmos de SIA pode permitir gerar modelos de medula óssea (Seção 3.4.3.1) e estudar a evolução de bibliotecas genéticas para a geração de repertórios de anticorpos de forma mais plausível sob o ponto de vista biológico.

Irun Cohen (1992b) propõe que as células apresentadoras de antígenos (APCs) são capazes de identificar o contexto do encontro entre uma célula imunológica e um antígeno através de seu processamento e apresentação, servindo como um filtro para o ruído molecular e uma lente que focaliza a atenção em padrões moleculares específicos. Sendo assim, as APCs podem ser comparadas com a *retina*, sendo fundamentais como *detectores de características* de diversos elementos encontrados no ambiente.

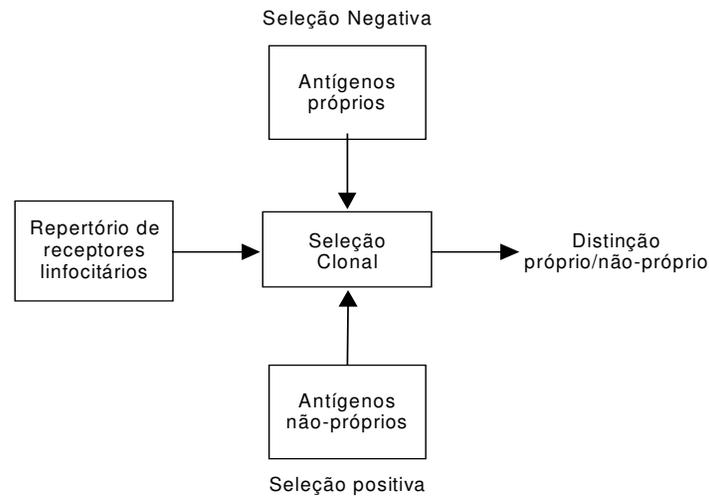


Figura 7.1. Algoritmo genérico de distinção próprio/não-próprio gerado a partir do algoritmo de seleção negativa, do algoritmo CLONALG e de um possível algoritmo de seleção positiva.

Grossberg *et al.* (1989) propuseram modelos neurais artificiais para a visão. Apoiando-se no ponto de vista de I. Cohen, modelos similares aos desenvolvidos por Grossberg *et al.* (1989) poderiam ser propostos para a atividade das células APCs do sistema imunológico.

Tomando o algoritmo de seleção negativa apresentado por Forrest *et al.* (1994), discutido na Seção 3.3.3.2, pode ser possível propor um *algoritmo de seleção positiva* baseado na seleção positiva dos linfócitos T (Seção 2.9.1). Juntando o algoritmo de seleção e expansão clonal (CLONALG) proposto nesta tese, com os algoritmos de seleção positiva e negativa, poderíamos desenvolver um algoritmo genérico de distinção próprio/não-próprio como ilustrado na Figura 7.1.

Devido ao grau de similaridade existente entre o sistema imunológico e os sistemas neurais, destacados na Seção 4.3.2, é de se admirar que haja tão poucas referências propondo modelos conexionistas baseados em princípios imunológicos e vice-versa (Seção 3.3.7), tendo sido alguns destes propostos nesta tese. Certamente as áreas de redes neurais e sistemas imunológicos artificiais têm muito a lucrar uma com a outra e novas ferramentas computacionais, hibridizando princípios de ambas, tendem a surgir.

Na Seção 3.4.2, foi proposta uma representação genérica (Figuras 3.9 e 3.10) para as células que compõem a engenharia imunológica. Neste modelo, as conexões celulares podem apresentar pesos individuais para cada componente da cadeia reconhecida e de reconhecimento. Até o presente momento, todos os algoritmos conhecidos na literatura para modelar as redes imunológicas operam com um peso único para quantificar o grau de interação das células. Uma das propostas que pode ser levantada é a utilização de um peso diferente para cada um dos parâmetros de cada uma das cadeias representantes das células do SIA. A complexidade do SIA aumentaria proporcionalmente à sua flexibilidade.

Na Seção 4.4.4 foram apresentados conceitos básicos sobre a programação genética (PG) que tenta evoluir programas computacionais utilizando princípios de computação evolutiva.

A principal diferença entre a PG e os algoritmos genéticos está na representação. Enquanto o GA utiliza cadeias binárias ou vetores de comprimento fixo para representar os indivíduos da população, a PG geralmente codifica os indivíduos através de estruturas em árvore de comprimentos variáveis. Uma abordagem semelhante poderia ser empregada para a engenharia imunológica e até mesmo para os sistemas imunológicos artificiais, gerando uma possível *programação imunológica* (PI). Neste caso, um algoritmo como o CLONALG poderia ser empregado para evoluir um conjunto de programas computacionais com características similares às apresentadas na Figura 4.22, ou seja, todos os programas apresentariam um valor de fitness diferente, porém eles poderiam ter atributos distintos como quantidade de funções empregadas na solução do problema, custo computacional, entre outros.

A herança genética proposta por Lamarck na teoria da evolução sugere que características adquiridas durante a vida de um indivíduo podem ser passadas para seus descendentes. Esta teoria foi demonstrada ser inconsistente com o estudo mais aprofundado de células somáticas e germinativas. Entretanto, o efeito Baldwin é um mecanismo no qual a aprendizagem facilita a assimilação indireta de novas informações genéticas ao longo das gerações do processo evolutivo (Bäck *et al.*, 2000a). Na Seção 4.5 foi discutida a relação entre o processo de evolução celular que ocorre no sistema imunológico, chamado de microevolução, e a evolução das espécies. Hightower *et al.* (1996) foram os primeiros a estudar o efeito Baldwin no SI, considerando um espaço de formas de Hamming com afinidades não-lineares. Os resultados apresentados indicaram que o efeito Baldwin não é uma relação universal entre evolução e aprendizagem, e depende da forma da função de afinidade. Estes resultados têm sido refutados na literatura (Bäck *et al.*, 2000a – Cap. 34, p. 312). Muito ainda precisa ser feito para que se possa inferir como a microevolução (evolução dentro do indivíduo) influencia na macroevolução (evolução da espécie), tanto nos sistemas naturais quanto nos artificiais. Modelos utilizando outras representações, medidas de afinidade, etc., precisam ser avaliados.

Apesar da grande quantidade de aplicações de SIA a problemas de teste e mundo real ainda há muito por se fazer no desenvolvimento de novos algoritmos partindo de mecanismos imunológicos ainda não modelados. Como exemplos podemos citar: 1) um modelo da reação em cascata do sistema complemento poderia constituir um novo algoritmo para a eliminação de vírus computacionais; 2) os mecanismos de regulação do sistema imunológico poderiam ser utilizados como forma de controle de processos industriais caracterizados pela não-estacionariedade e distributividade.

Finalmente, um algoritmo como o CLONALG proposto nesta tese e outros algoritmos imunológicos aplicados na solução de problemas de otimização multimodal (Seção 3.3.5) permitem verificar que estes modelos imunológicos tratam o problema de múltiplas espécies e nichos de forma direta. As células do sistema imunológico apresentam grande diversidade, cooperam e competem entre si para garantir uma defesa eficiente contra os microorganismos infecciosos. Na área da computação evolutiva, vários modelos têm sido desenvolvidos no intuito de gerar populações de indivíduos com alto grau de diversidade e cobertura de múltiplas soluções de um dado problema (Goldberg, 1989; Mahfoud, 1995). As características de diversidade, cooperação e competição celulares do sistema

imunológico sugerem novos paradigmas para o desenvolvimento de algoritmos evolutivos eficientes na solução de problemas de clusterização e otimização multimodal e multiobjetivo.

7.3.2. Variações dos Algoritmos Propostos

Dentre as possíveis melhorias, variações e novas interpretações dos algoritmos propostos, é possível destacar um conjunto de itens comuns a todos:

- Emprego de algoritmos probabilísticos de seleção;
- Execução de testes com diferentes operadores de mutação;
- Possibilidade de inclusão de operadores de recombinação e inversão nos SIAs. Diversos trabalhos existentes na literatura já abordam estes operadores dentro dos sistemas imunológicos artificiais, porém para manter uma maior inspiração biológica, nos restringimos apenas ao operador de mutação (somática); e
- Aplicação dos algoritmos a problemas de mundo real de complexidade elevada.

7.3.2.1. SAND

Como principais extensões do algoritmo SAND, podemos citar sua aplicação a espaços de formas Inteiras para a solução de problemas como o TSP e a introdução de um limiar de afinidade ϵ nas funções de energia.

7.3.2.2. CLONALG

No Passo (3) da versão para o reconhecimento de padrões, o processo de seleção escolhe um subconjunto n de anticorpos baseado em suas afinidades ao antígeno \mathbf{Ag}_j . É possível desenvolver uma versão em lote (batch) do algoritmo tal que se inclua um nível de estimulação τ_i de cada anticorpo \mathbf{Ab}_i antes de se definir qual deles será selecionado para reprodução. Neste caso, todos os antígenos \mathbf{Ag}_j , $j = 1, \dots, M$, deverão ser apresentados a todo o repertório de anticorpos antes que um anticorpo específico seja selecionado.

Assim como na ABNET e na aiNet, é possível utilizar um limiar de afinidade ϵ durante o reconhecimento antigênico. Este limiar permitiria um ajuste dinâmico da cardinalidade da população de memórias $\mathbf{Ab}_{\{m\}}$.

Nos Passos 3 e 7 do algoritmo CLONALG para o problema de reconhecimento de padrões, o método de seleção empregado foi determinístico, simulando os processos biológicos descritos na literatura. Entretanto, qualquer mecanismo de seleção probabilístico baseado na afinidade Ag-Ab poderia ser empregado, como o algoritmo de Roulette Wheel, seleção por torneio e outras estratégias utilizadas em computação evolutiva (Seção 4.4).

Também é possível aplicar procedimentos direcionados de maturação de afinidade, como por exemplo, supondo a existência de uma população de antígenos a ser reconhecida, os anticorpos podem ser mutados de forma a aumentar continuamente suas afinidades aos respectivos antígenos. Este procedimento foi adotado durante a aprendizagem da aiNet, que utiliza esta versão do CLONALG como parte de seu processamento.

Da forma como o CLONALG foi apresentado para o problema de reconhecimento de padrões, o algoritmo gera um anticorpo de memória para cada antígeno. É possível introduzir o conceito de limiar de afinidade ϵ , de forma a controlar dinamicamente a quantidade de anticorpos de memória. Este limiar de afinidade pode estar relacionado apenas ao conjunto de memórias $\mathbf{Ab}_{\{m\}}$, da seguinte maneira: se vários elementos do conjunto de memórias possuem afinidade superior a um limiar ϵ , então basta um deles (ou uma quantidade mínima) para reconhecer e classificar uma classe de antígenos com características tais que eles recaiam dentro da região de reconhecimento V_ϵ deste anticorpo.

7.3.2.3. ABNET

A principal extensão da ABNET está no desenvolvimento de um algoritmo de treinamento construtivo, utilizando o princípio da seleção clonal, para redes com pesos reais. O algoritmo CLONALG poderia ser empregado como parte do processamento da rede, sendo responsável pela determinação de quais neurônios iriam se multiplicar e serem selecionados para entrar na rede.

No processo de crescimento da ABNET (Seção 5.4.2.1), o anticorpo selecionado para se reproduzir gera um clone cujo vetor de pesos associado é o melhor complemento para o antígeno de menor afinidade a este anticorpo. Uma outra abordagem para se atribuir pesos associados ao clone do anticorpo mais estimulado seria colocá-lo no centro de massa de todos os antígenos reconhecidos por este anticorpo.

Utilizar um histograma para determinação do número de clusters da ABNET, assim como a matriz-U, para grid de saída unidimensional, discutida na Seção 4.2.2.2, e o histograma empregado na aiNet (Seção 5.5.4).

7.3.2.4. aiNet

Dentre os principais estudos e extensões da aiNet é possível citar:

- A utilização de parâmetros adaptativos, como σ_d , σ_s , n , d e ζ . Idéias extraídas de algoritmos coevolutivos e metaevolutivos poderiam ser empregadas como fonte de inspiração (Bäck *et al.*, 2000a,b);
- A implementação e comparação de diferentes medidas de afinidade;
- A análise detalhada de convergência, incluindo a estabilidade da população e erro de classificação;
- A comparação de desempenho da aiNet com outros modelos de redes imunológicas artificiais;
- Adoção de mecanismos alternativos para interpretação da configuração da rede resultante do processo de exposição ao conjunto de antígenos, principalmente na definição de vizinhança e discriminação dos clusters; e
- O desenvolvimento de uma ferramenta mais apropriada para a visualização da aiNet, principalmente para os casos em que a dimensão das células é superior a 3 ($L > 3$).

7.3.3. Especulações Sobre o Futuro da Área

Apesar da estrutura formal para a engenharia imunológica proposta nesta tese, em nível mundial a área ainda sofre pela falta de uma estrutura formal para o desenvolvimento de sistemas imunológicos artificiais. Como discutido no Capítulo 3, não existe nenhum livro texto com este conteúdo. Talvez ainda leve alguns anos até que uma revista específica para a área surja, provavelmente pelo IEEE (Transactions on Artificial Immune Systems?!), mas já é possível perceber a necessidade desta publicação.

Provavelmente uma das tendências mais fortes da área está no desenvolvimento de sistemas híbridos, contendo uma ou mais abordagens de computação inteligente, como redes neurais artificiais, computação evolutiva e sistemas nebulosos. *Redes neuroimunes, sistemas imunofuzzy, algoritmos imunogenéticos e redes neurofuzzyimunes* tornar-se-ão comuns. A idéia reside no aproveitamento das potencialidades de cada abordagem de forma a desenvolver algoritmos capazes de superar as deficiências isoladas de cada uma.

No Capítulo 4, foi discutido brevemente que um perfeito equilíbrio entre o sistema nervoso central, o sistema imunológico e o sistema endócrino permite a garantia da homeostase do indivíduo. Estas discussões nos levam a conjecturar acerca da possibilidade de que o sistema endócrino também possa oferecer uma grande quantidade de mecanismos potenciais para o desenvolvimento de *sistemas endócrinos artificiais*. Indo ainda mais longe, poderíamos pensar também que, uma vez proposto um aparato respeitável de conceitos, teorias e técnicas computacionais associadas aos sistemas nervoso, imunológico e endócrino, seria possível projetar um sistema cibernético com características mais próximas da *vida artificial* do que os já existentes.

Finalmente, a adoção de uma visão semiótica do sistema imunológico ou a hibridização da semiótica computacional com os sistemas imunológicos artificiais, pode levar ao desenvolvimento de novos tipos de sistemas autônomos inteligentes capazes de apresentar comportamentos emergentes como percepção, capacidade de julgamento, etc. Um ponto de partida na busca por uma visão semiótica do sistema imunológico pode ser o trabalho desenvolvido por Secarz *et al.* (1988).

BIBLIOGRAFIA

Com o objetivo de fornecer ao leitor uma grande quantidade de referências sobre os assuntos abordados nesta tese, principalmente os sistemas imunológicos artificiais, ao invés de apresentar apenas as referências bibliográficas, é listada toda a bibliografia consultada durante o desenvolvimento da tese. Aqueles autores cujos nomes aparecem no Índice Remissivo compõem as referências da tese.

- [1] **Aamodt, A. & Plaza, E. (1994)**, “Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches”, *AI Communications*, **7**(i), pp. 39-59.
- [2] **Abbattista, F., Di Gioia G., Di Santo G. & Farinelli A. M. (1996)**, “An Associative Memory Based on the Immune Networks”, *2nd. Online Workshop on Evolutionary Computation*, Nagoya, Japan.
- [3] **Abbas, A. K. & Janeway, C. A. (2000)**, “Immunology: Improving on Nature in the Twenty-First Century”, *Cell*, **100**, pp. 129-138.
- [4] **Abbas, A. K., Litchman, A. H. & Pober, J. S. (1995)**, *Imunologia Celular e Molecular*, Revinter.
- [5] **Ada, G. L. & Nossal, G. J. V. (1987)**, “The Clonal Selection Theory”, *Scientific American*, **257**(2), pp. 50-57.
- [6] **Adams, D. (1996)**, “How the Immune System Works and Why it Causes Autoimmune Diseases”, *Imm. Today*, **17**(7), pp. 300-302.
- [7] **Ader, R., Felten, D. L. & Cohen, N. (Eds.) (1991)**, *Psychoneuroimmunology*, 2^a Ed., Sand Diego, Academic Press.
- [8] **Adleman, L. M. (1994)**, “Molecular Computation of Solutions to Combinatorial Problems”, *Science*, **266**, pp. 1021-1024.
- [9] **Ahmed, R. & Sprent, J. (1999)**, “Immunological Memory”, *The Immunologist*, **7/1-2**, pp. 23-26.
- [10] **Aisu, H. & Mizutani H. (1996)**, “Immunity-Based Learning – Integration of Distributed Search and Constraint Relaxation”, *Proc. do IMBS'96*.
- [11] **Aleksander, I. & Morton, H. B. (1991)**, “General Neural Unit: Retrieval Performance”, *Electronic Letters*, **27**(19), pp. 1776-1778
- [12] **Allen, D. et al. (1987)**, “Timing, Genetic Requirements and Functional Consequences of Somatic Hypermutation During B-cell Development”, *Imm. Rev.*, **96**, pp. 5-22.
- [13] **Al-Sultan, K. S. & Khan, M. M. (1996)**, “Computational Experience on Four Algorithms for the Hard Clustering Problem”, *Pattern Recognition Letters*, **17**, pp. 295-308.
- [14] **Anderson, G., Hare, K. J. & Jenkinson, E. J. (1999)**, “Positive Selection of Thymocytes: the Long and Winding Road”, *Imm. Today*, **20**(10), pp. 463-468.
- [15] **Anderson, K., Wrammert, J. & Leanderson, T. (1998)**, “Affinity Selection and Repertoire Shift Paradoxes as a Consequence of Somatic Mutation?”, *Imm. Rev.*, **162**, pp. 172-182.
- [16] **Andrade, W. N., Johnston, M. G. & Hay, J. B. (1998)**, “The Relationship of Blood Lymphocytes to the Recirculating Lymphocyte Pool”, *Blood*, **91**(5), pp. 1653-1661.

- [17] **Axelrod, R. (1997)**, *The Complexity of Cooperation*, Princeton University Press, Princeton, New Jersey.
- [18] **Babu, G. P. & Murty, M. N. (1994)**, “Clustering With Evolution Strategies”, *Pattern Recognition*, **27**(2), pp. 321-329.
- [19] **Bäck, T., Fogel, D. B. & Michalewicz, Z. (2000a)**, *Evolutionary Computation 1 Basic Algorithms and Operators*, Institute of Physics Publishing (IOP), Bristol and Philadelphia.
- [20] **Bäck, T., Fogel, D. B. & Michalewicz, Z. (2000b)**, *Evolutionary Computation 2 Advanced Algorithms and Operators*, Institute of Physics Publishing (IOP), Bristol and Philadelphia.
- [21] **Bäck, T. & Schwefel, H. -P. (1993)**, “An Overview of Evolutionary Algorithms for Parameter Optimization”, *Evolutionary Computation*, **1**(1), pp. 1-23.
- [22] **Baggish, J. (1994)**, *Como Funciona seu Sistema Imunológico*, Quark Books.
- [23] **Ballet, P., Tisseau, J. & Harrouet, F (1997)**, “A Multiagent System to Model an Human Humoral Response”, *Proc. do IEEE SMC'97*, **1**, pp. 357-362.
- [24] **Banchereau, J. & Steinman, R. M. (1998)**, “Dendritic Cells and the Control of Immunity”, *Nature*, **392**, pp. 245-252.
- [25] **Barreiros, J. A. L., Ribeiro, R. R. P., Affonso, C. M. & Santos, E. P. (1997)**, “Estabilizador de Sistemas de Potência Adaptativo com Pré-Programação de Parâmetros e Rede Neural Artificial”, *III Congresso Latino Americano de Geração e Transmissão de Eletricidade*, pp.538-542.
- [26] **Bauer, H.-U & Villmann, Th. (1997)**, “Growing a Hypercubical Output Space in a Self-Organizing Feature Map”, *IEEE Trans. on Neural Networks*, **8**(2), pp. 218-226.
- [27] **Bell, G. I. & Perelson, A. S. (1978)**, “An Historical Introduction to Theoretical Immunology”, *Theoretical Immunology*, (Eds.) G. I. Bell, A. S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc., pp. 3-41.
- [28] **Berek, C. & Ziegner, M. (1993)**, “The Maturation of the Immune Response”, *Imm. Today*, **14**(8), pp. 400-402.
- [29] **Bernardes, A. T. & dos Santos, R. M. Z. (1997)**, “Immune Network at the Edge of Chaos”, *J. theor. Biol.*, **186**, pp. 173-187.
- [30] **Bernaschi, M., Castiglione, F. & Succi, S. (1998)**, “A Parallel Simulator of the Immune Response”, *Lect. Notes in Computer Sc.*, **1401**, pp. 163-172.
- [31] **Bersini, H. & Calenbuhr, V. (1997)**, “Frustrated Chaos in Biological Networks”, *J. theor. Biol.*, **188**, pp. 187-200.
- [32] **Bersini, H. (1991)**, “Immune Network and Adaptive Control”, *Proc. of the First European Conference on Artificial Life*, MIT Press.
- [33] **Bersini, H. & Varela, F. J. (1990)**, “Hint for Adaptive Problem Solving Gleaned from Immune Networks”, *Parallel Problem Solving from Nature*, pp. 343-354.
- [34] **Bezdek, J. C. (1981)**, *Pattern Recognition With Fuzzy Objective Function Algorithms*, Plenum Press, New York.
- [35] **Bezdek, J. C. & Pal, S. K. (1992)**, *Fuzzy Models for Pattern Recognition: Methods That Search For Structures in Data*, New York, IEEE.
- [36] **Bezzi, M., Celada, F., Ruffo, S. & Seiden, P. E. (1997)**, “The Transition Between Immune and Disease States in a Cellular Automaton Model of Clonal Immune Response”, *Physica A*, **245**(1-2), pp. 145-163.

-
- [37] **Bishop, C. M., Svensén, M. & Williams, C. K. I. (1997)**, “GTM: The Generative Topographic Mapping”, *Neural Computation*, **9**.
- [38] **Bishop, C. M., Svensén, M. & Williams, C. K. I. (1997)**, “GTM: A Principled Alternative to the Self-Organizing Map”, *Advances in Neural Information Processing Systems*, **9**.
- [39] **Biswas, N. N. & Kumar, R. (1990)**, “A New Algorithm for Learning Representations in Boolean Neural Networks”, *Current Science* 59(12), pp. 595-600.
- [40] **Blalock, E. J. (1994)**, “The Immune System Our Sixth Sense”, *The Immunologist*, **2/1**, pp. 8-15.
- [41] **Blalock, E. J. (1992)**, *Neuroimmunoendocrinology*, 2^a Ed., Chemical Immunology.
- [42] **Boers, E. G. W. & Kuiper, H. (1992)**, “Biological Metaphors and the Design of Modular Artificial Neural Networks”, Tese de Mestrado, Leiden University, Leiden, Netherlands.
- [43] **Bonamin, L. V. (1994)**, “O Estresse e as Doenças”, *Ciência Hoje*, **17(99)**, pp. 25-30 .
- [44] **Bonabeau, E. W. & Theraulaz, G. (1995)**, “Why Do We Need Artificial Life”, *Artificial Life An Overview*, Langton, C. G. (Ed.), The MIT Press, Cambridge Massachusetts, pp. 303-325.
- [45] **Bonna, C. A. & Kohler, H. (1983)**, “Immune Networks”, *Annals of the New York Academy of Sciences*, **418**.
- [46] **Borghans, J. A. M., Taams, L. S., Wauben, M. H. M. & De Boer, R. J. (1999)**, “Competition for Antigenic Sites During T Cell Proliferation: A Mathematical Interpretation of *in vitro* data”, *Proc. Natl. Ac. Sci. USA*, **96**, pp. 10782-10787.
- [47] **Broomhead, D. S. & Lowe, D. (1988)**, “Multivariable Functional Interpolation and Adaptive Networks”, *Complex Systems*, **2**, pp. 321-355.
- [48] **Browne, C. & Aleksander, I. (1996)**, “Digital General Neural Unit with Controlled Transition Probabilities”, *Electronic Letters*, **32(9)**, pp. 824-825.
- [49] **Burnet, F. M. (1978)**, “Clonal Selection and After”, *Theoretical Immunology*, (Eds.) G. I. Bell, A. S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc., pp. 63-85.
- [50] **Burnet, F. M. (1959)**, *The Clonal Selection Theory of Acquired Immunity*, Cambridge University Press.
- [51] **Calenbuhr, V., Bersini, H., Stewart, J. & Varela, F. J. (1995)**, “Natural Tolerance in a Simple Immune Network”, *J. theor. Biol.*, **177**, pp. 199-213.
- [52] **Calich, V. L. G. & C. A. C. Vaz (1989)**, *Imunologia Básica*, Artes Médicas.
- [53] **Carneiro, J., Coutinho, A., Faro, J. & Stewart, J. (1996a)**, “A Model of the Immune Network with B-T Cell Co-Operation. I – Prototypical Structures and Dynamics”, *J. theor. Biol.*, **182**, pp. 513-529.
- [54] **Carneiro, J., Coutinho, A. & Stewart, J. (1996b)**, “A Model of the Immune Network with B-T Cell Co-Operation. II – The Simulation of Ontogenesis”, *J. theor. Biol.*, **182**, pp. 531-547.
- [55] **Carol, M. C. & Prodeus, A. P. (1998)**, “Linkages of Innate and Adaptive Immunity”, *Current Opinion in Imm.*, **10**, pp. 36-40.
- [56] **Carroll, J. D. (1995)**, “ ‘Minimax Length Links’ of a Dissimilarity Matrix and Minimum Spanning Trees”, *Psychometrika*, **60(3)**, pp. 371-374.
- [57] **Celada, F. & Seiden, P. E. (1996)**, “Affinity Maturation and Hypermutation in a Simulation of the Humoral Immune Response”, *Eur. J. Imm.*, **26**, pp. 1350-1358.
-

- [58] **Celada, F. & Seiden, P. E. (1992)**, “A Computer Model of Cellular Interaction in the Immune System”, *Imm. Today*, **13**, pp. 56-62.
- [59] **Chandra, R. K. (1997)**, “Nutrition and the Immune System: An Introduction”, *Am. J. Clin. Nutr.*, **66**, pp. 460S-463S.
- [60] **Chen, M.-S., Han, J. & Yu, P. S. (1996)**, “Data Mining: An Overview from a Database Perspective”, *IEEE Trans. on Knowledge Engineering*, **8**(6), pp. 866-883.
- [61] **Chen, S., Chng, E. S. & Alkadhim, K. (1995)**, “Regularised Orthogonal Least Squares Algorithm for Constructing Radial Basis Function Networks”, *Int. J. of Control*, **64**, pp. 829-837.
- [62] **Chen, S., Cowan, C. F. N. & Grant, P. M. (1991)**, “Orthogonal Least Squares Algorithm for Radial Basis Function Networks”, *IEEE Trans. on Neural Networks*, **2**(2), pp. 302-309.
- [63] **Cho S.-B. (1997)**, “Self-Organizing Map with Dynamical Node Splitting: Application to Handwritten Digit Recognition”, *Neural Computation*, **9**, pp. 1345-1355.
- [64] **Çiringiroglu, U. (1993)**, “A Charge-Based Neural Hamming Classifier”, *IEEE J. of Solid-State Circuits*, **28**(1), 59-67.
- [65] **Cohen, J. J. (1993)**, “Apoptosis”, *Imm. Today*, **14**(3), pp. 126-130.
- [66] **Cohen, I. R. (1992a)**, “The Cognitive Principle Challenges Clonal Selection”, *Imm. Today*, **13**(11), pp. 441-444.
- [67] **Cohen, I. R. (1992b)**, “The Cognitive Paradigm and the Immunological Homunculus”, *Imm. Today*, **13**(12), pp. 490-494.
- [68] **Colaco, C. (1998)**, “Acquired Wisdom in Innate Immunity”, *Imm. Today*, **19**(1), pp. 50.
- [69] **Cook, M. (2000)**, “Receptor Editing (and the Evolution of Sex)”, *Imm. Today*, **21**(1), pp. 55.
- [70] **Costa, J. A. F. (1999)**, *Classificação Automática e Análise de Dados por Redes Neurais Auto-Organizáveis*, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp, Campinas, Brasil, Novembro.
- [71] **Coutinho, A. (1995)**, “The Network Theory: 21 Years Later”, *Scand. J. Imm.*, **42**, pp. 3-8.
- [72] **Coutinho, A. (1989)**, “Beyond Clonal Selection and Network”, *Imm. Rev.*, **110**, pp. 63-87.
- [73] **Coutinho, A., Forni, L., Holmberg, D., Ivars, F. & Vaz, N. (1984)**, “From an Antigen-Centered, Clonal Perspective of Immune Responses to an Organism-Centered, Network Perspective of Autonomous Activity in a Self-Referential Immune System”, *Imm. Rev.*, **79**, pp. 152-168.
- [74] **Cunha, P. (1994)**, *Por Dentro do Sistema Imunológico*, Atual Editora, 3^a. ed.
- [75] **Cunningham, A. J. (1978)**, “ ‘Gestalt Immunology’: A Less Reductionist Approach to the Subject”, *Theoretical Immunology*, (Eds.) G. I. Bell, A. S. Perelson & G. H. Pimbley Jr., marcel Dekker Inc., pp. 45-85.
- [76] **Cziko, G. (1995)**, “The Immune System: Selection by the Enemy”, *Without Miracles*, G. Cziko, A Bradford Book, The MIT Press, pp. 39-48.
- [77] **Dasgupta, D. (2000)**, “An Immune Agent Architecture for Intrusion Detection”, *Proc. of GECCO’00*, Workshop on Artificial Immune Systems and Their Applications.
- [78] **Dasgupta, D. (1999a)**, “Information Processing in the Immune System”, *New Ideas in Optimization*, (Eds.) D. Corne, M. Dorigo & F. Glover, McGraw Hill, London, pp. 161-165.
- [79] **Dasgupta, D., (1999b)**, “Immunity-Based Intrusion Detection System: A General Framework”, *Proc. of the 22nd NISSC*.

-
- [80] **Dasgupta, D., Cao, Y. & Yang, C. (1999)**, “An Immunogenetic Approach to Spectra Recognition”, *Proc. do GECCO'99*, pp. 149-155.
- [81] **Dasgupta, D. & Forrest, S. (1999)**, “Artificial Immune Systems in Industrial Applications”, *Proc. do IPMM'99*.
- [82] **Dasgupta, D. (Ed.) (1998a)**, *Artificial Immune Systems and Their Applications*, Springer-Verlag.
- [83] **Dasgupta, D., (1998b)**, “An Overview of Artificial Immune Systems and Their Applications”, *Artificial Immune Systems and Their Applications*, Springer-Verlag, pp. 3-21.
- [84] **Dasgupta, D., (1997)**, “Artificial Neural Networks and Artificial Immune Systems: Similarities and Differences”, *Proc. of the IEEE SMC*, **1**, pp. 873-878.
- [85] **Dasgupta, D. & Attoh-Okine, N., (1997)**, “Immunity-Based Systems: A Survey”, *Proc. of the IEEE SMC*, **1**, pp. 369-374.
- [86] **Dasgupta, D. & Forrest, S. (1996)**, “Novelty Detection in Time Series Data Using Ideas From Immunology”, *Proc. of the ISCA'96*.
- [87] **Dawkins, R. (1989)**, *The Selfish Genes*, Oxford University Press.
- [88] **Deaton, R, Garzon, M, Rose, J. A., Murphy, R. C., Stevens Jr., S. E. & Franceschetti, D. R. (1997)**, “A DNA Based Artificial Immune System for Self-Nonself Discrimination”, *Proc. of the IEEE SMC'97*, pp. 862-866.
- [89] **De Boer, R. J., Segel, L. A. & Perelson, A. S. (1992)**, “Pattern Formation in One- and Two-Dimensional Shape-Space Models of the Immune System”, *J. theor. Biol.*, **155**, pp. 295-333.
- [90] **De Boer, R. J. & Perelson, A. S. (1991)**, “Size and Connectivity as Emergent Properties of a Developing Immune Network”, *J. theor. Biol.*, **149**, pp. 381-424.
- [91] **De Boer, R. J. & Perelson, A. S. (1993)**, “How Diverse Should the Immune System Be”, *Proc. R. Soc. London B*, **252**, pp. 171-175.
- [92] **De Boer, R. J. (1989)**, “Information Processing in Immune System: Clonal Selection Versus Idiotypic Network Models”, *Cell to Cell Signalling: From Experiments to Theoretical Models*, Academic Press Limited, pp. 285-302.
- [93] **De Castro, L. N. & Von Zuben, F. J., (2001a)**, “Learning and Optimization Using the Clonal Selection Principle”, aceito para publicação no *IEEE Transaction on Evolutionary Computation*, Special Issue sobre Sistemas Imunológicos Artificiais.
- [94] **De Castro, L. N. & Von Zuben, F. J., (2001b)**, “An Immunological Approach to Initialize Centers of Radial Basis Function Neural Networks”, *Proc. do CBRN'01*, pp. 79-84.
- [95] **De Castro, L. N. & Von Zuben, F. J., (2001c)**, “An Immunological Approach to Initialize Feedforward Neural Network Weights”, *Proc. do ICANNGA'01*, pp. 126-129.
- [96] **De Castro, L. N. & Von Zuben, F. J., (2001d)**, “A Pruning Self-Organizing Algorithm to Select Centers of Radial Basis Function Neural Network”, *Proc. do ICANNGA'01*, pp. 114-117.
- [97] **De Castro, L. N. & Von Zuben, F. J., (2001e)**, “Automatic Determination of Radial Basis Function: An Immunity-Based Approach”, aceito para publicação no *International Journal of Neural Systems*, Special Issue sobre Non-Gradient Learning Techniques.
- [98] **De Castro, L. N. & Von Zuben, F. J., (2001f)**, “The Construction of a Boolean Competitive Neural Network Using Ideas From Immunology”, submetido ao jornal *International Journal of Neurocomputing*.
-

- [99] **De Castro, L. N. & Von Zuben, F. J., (2001g)**, “aiNet: An Artificial Immune Network for Data Analysis”, Capítulo do livro *Data Mining: A Heuristic Approach*, H. A. Abbass, R. A. Sarker, and C. S. Newton (Eds.), Idea Group Publishing, USA.
- [100] **De Castro, L. N. & Von Zuben, F. J. (2000a)**, “The Clonal Selection Algorithm with Engineering Applications”, *GECCO'00 – Workshop Proceedings*, pp. 36-37.
- [101] **De Castro, L. N. & Von Zuben, F. J. (2000b)**, “Artificial Immune Systems: Part II – A Survey of Applications”, *Technical Report – RT DCA 02/00*, p. 65.
- [102] **De Castro, L. N. & Von Zuben, F. J., (2000c)**, “An Evolutionary Immune Network for Data Clustering”, *Proc. do IEEE SBRN*, pp. 84-89.
- [103] **De Castro, L. N. & Von Zuben, F. J. (1999a)**, “Artificial Immune Systems: Part I – Basic Theory and Applications”, *Technical Report – RT DCA 01/99*, p. 95.
- [104] **De Castro, L. N. & Von Zuben, F. J. (1999b)**, “An Improving Pruning Technique with Restart for the Kohonen Self-Organizing Feature Map”, *Proc. do IJCNN*, **3**, pp. 1916-1919.
- [105] **De Castro, L. N. & Von Zuben, F. J. (1999c)**, “Neural Networks with Adaptive Activation Functions: A Second Order Approach”, *Proc. do SCI/ISAS'99*, **3**, pp. 574-581.
- [106] **De Castro, L. N., Iyoda E. M., Santos, E. P. & Von Zuben F. J. (1999a)**, “Redes Neurais Construtivas: Uma Abordagem Comparativa”, *Anais do IV CBRN*, pp. 102-107.
- [107] **De Castro, L. N., Ramírez L. A., Gomide F. A. C. & Von Zuben, F. J. (1999b)**, “Hybrid Tuning of Activation Functions in Feedforward Networks”, *Proc. do IJCNN'99*, **6**, pp. 4263-4267.
- [108] **De Castro, L. N. & Von Zuben F. J. (1998a)**, “A Hybrid Paradigm for Weight Initialisation in Supervised Feedforward Neural Network Learning”, *Proc. do ICS'98, Workshop on Artificial Intelligence*, pp. 30-37, Taipei/Taiwan, R.O.C.
- [109] **De Castro, L. N. & Von Zuben F. J. (1998b)**, “Uma Abordagem Simples e Robusta para Inicialização de Pesos em Redes de Múltiplas Camadas”, *Anais do V SBRN*, pp. 97-102.
- [110] **De Castro, L. N. & Von Zuben, F. J. (1998c)**, “Optimized Training Techniques for Feedforward Neural Networks”, Relatório Técnico RT DCA – 03/98, FEEC/UNICAMP, 41 p.
- [111] **De Castro, L. N., Iyoda E. M., Von Zuben F. J. & Gudwin R. R. (1998a)**, “Feedforward Neural Network Initialisation: an Evolutionary Approach”, *Proc. do IEEE V SBRN*, pp. 43-48.
- [112] **De Castro, L. N., Von Zuben, F. J. & Martins, W. (1998b)**, “Hybrid and Constructive Learning Applied to a Prediction Problem in Agriculture”, *Proc. do IJCNN'98*, **3**, pp. 1932-1926.
- [113] **De Castro, L. N., Lima, W. da S. & Martins, W. (1997a)**, “Classificador Neural para Previsão de Carga a Curto Prazo”, *Anais do IV SBRN*, pp. 68-70.
- [114] **De Castro, L. N., Lima, W. da S. & Martins, W (1997b)**, “Redes Neurais Artificiais Aplicadas a Previsão de Produtividade de Soja”, *Anais do III CBRN*, pp. 308-312.
- [115] **De Deus Jr., G. A., Portugheis, J. & Netto, M. L. A. (1999a)**, “Projeto de Receptores FH-CDMA Utilizando Redes Neurais Artificiais”, *Proc. do XVII SBT*, pp. 265-270.
- [116] **De Deus Jr., G. A., Portugheis, J. & de Castro, L. N. (1999b)**, “Mapa Auto-Organizável Não-Paramétrico Aplicado ao Problema de Decisão de Lógica Majoritária”, *Proc. do IV SBA*, pp. 150-155.

-
- [117] **De Jong, K. A. (1994)**, “Genetic Algorithms: A 25 Year Perspective”, *Computational Intelligence: Imitating Life*, IEEE Press, pp. 125-134.
- [118] **Denker, J. S. (1986)**, “Neural Network Models of Learning and Adaptation”, *Physica 22D*, pp. 216-232.
- [119] **Dennett, D. C. (1978)**, *Brainstorms: Philosophical Essays on Mind of Psychology*, Brighton Harvester.
- [120] **Detours, V., Sulzer, B. & Perelson, A. S. (1996)**, “Size and Connectivity of the Idiotypic Network are Independent of the Discreteness of the Affinity Distribution”, *J. theor. Biol.*, **183**, pp. 409-416.
- [121] **D’haeseleer, P., Forrest, S. & Helman, P. (1996)**, “An Immunological Approach to Change Detection: Algorithms, Analysis and Implications”, *Proc. of the IEEE Symposium on Computer Security and Privacy*.
- [122] **Dilger, W. (1997)**, “Decentralized Autonomous Organization of the Intelligent Home According to the Principle of the Immune System”, *Proc. do IEEE SMC’97*, pp. 351-356.
- [123] **Dilger, W. (1996)**, “The Immune System of the Smart Home”, *Y. Ishida (ed.), Workshop Notes 4, Proc. of the ICMAS’96*, pp. 72-81.
- [124] **Dos Santos, E. P. & Von Zuben, F. J. (2000)**, “Efficient Second-Order Learning Algorithms for Discrete-Time Recurrent Neural Networks”, L.R. Medsker and L.C. Jain (Eds.), *Recurrent Neural Networks: Design and Applications*, CRC Press, pp. 47-75.
- [125] **Dote, Y. (1996)**, “Soft Computing (Immune Networks) in Artificial Intelligence”, URL: <http://bank.csse.muroran-it.ac.jp/research/scai/scaip.htm>
- [126] **Dreher H. (1995)**, *The Immune Power Personality*, Penguin Books.
- [127] **Edelman, G. M. (1992)**, *Bright Air, Brilliant Fire, On the Matter of the Mind*, Basic Books.
- [128] **Edelman, G. M. (1987)**, *Neural Darwinism The Theory of Neuronal Group Selection*, Basic Books.
- [129] **Edelman, G. M. (1975)**, “Molecular Recognition in the Immune and Nervous Systems”, *The Neurosciences: Paths of Discovery*, MIT Press, pp. 65-74.
- [130] **Einarson, G. (1980)**, Address Assignment for a Time-Frequency-Coded Spread-Spectrum System, *Bell System Technical Journal*, **59**(7), pp. 1241-1255.
- [131] **Eisenfeld, J. & Prueitt, P. (1988)**, “Systematic Approach to Modeling Immune Response”, *Theoretical Immunology*, Part One, (Ed.) A. S. Perelson, pp. 223-255.
- [132] **Endoh, S., Toma, N. & Yamada, K. (1998)**, “Immune Algorithm for n-TSP”, *Proc. of the IEEE SMC’98*, pp. 3844-3849.
- [133] **Epstein J. & Axtell R. (1996)**, “Disease Processes”, *Growing Artificial Societies*, MIT Press, pp. 138-152.
- [134] **Erwin, E., Obermayer, K. & Schulten, K. (1992)**, “Self-Organizing Maps: Ordering, Convergence Properties and Energy Functions”, *Biol. Cybern.*, **67**, pp. 47-55.
- [135] **Everitt, B. (1993)**, *Cluster Analysis*, Heinemann Educational Books.
- [136] **Fahlman, S. E. (1988)**, “An Empirical Study of Learning Speed in Back-Propagation Networks”, Relatório Técnico CMU-CS-88-162, School of Computer Science, Carnegie Mellon University, Pittsburg, PA.
- [137] **Farmer, J. D. (1990)**, “A Rosetta Stone for Connectionism”, *Physica 42D*, pp. 153-187.
-

- [138] **Farmer, J. D., Kauffman, S. A. Packard, N. H. & Perelson, A. S. (1987)**, “Adaptive Dynamic Networks as Models for the Immune System and Autocatalytic Sets”, *Ann. of the N. Y. Acad. of Sci.*, **504**, pp. 118-131.
- [139] **Farmer, J. D., Packard, N. H. & Perelson, A. S. (1986)**, “The Immune System, Adaptation, and Machine Learning”, *Physica 22D*, pp. 187-204.
- [140] **Fayyad, U. M. (1996)**, “Data Mining and Knowledge Discovery: Making Sense Out of Data”, *IEEE Expert Intelligent Systems and Their Applications*, **11**(5), pp. 20-25.
- [141] **Fischetti, M. & Martello, S. (1988)**, “A Hybrid Algorithm for Finding the k th Smallest of n Elements in $O(n)$ time”, *Ann. Op. Res.*, **13**, pp. 401-419.
- [142] **Fearon, D. T. & Locksley, R. M. (1996)**, “The Instructive Role of Innate Immunity in the Acquired Immune Response”, *Science*, **272**, pp. 50-53.
- [143] **Fogel, D. B. (1994)**, “An Introduction to Simulated Evolutionary Optimization”, *IEEE Trans. on Neural Networks*, **5**(1), pp. 3-14.
- [144] **Fogel, L. J., Owens, A. J. & Walsh, M. J. (1966)**, *Artificial Intelligence Through Simulated Evolution*, Wiley, New York.
- [145] **Forrest, S. & Hofmeyr, S. A. (2000)**, “Immunology as Information Processing”, *Design Principles for the Immune System and Other Distributed Autonomous Systems*, I. Cohen & L. A. Segel (Eds.), Oxford University Press, pp. 361-387.
- [146] **Forrest, S. & Hofmeyr, S. A. (1999)**, “John Holland's Invisible Hand: An Artificial Immune System”, *FESTSCHIRIFT para John Holland*.
- [147] **Forrest, S., Somayaji, A & Ackley, D. H. (1997)**, “Building Diversity Computer Systems”, *Proc. do 6th Workshop on Hot Topics in Operating Systems*, pp. 67-72.
- [148] **Forrest, S., Hofmeyr S. A. & Somayaji A. (1997)**, “Computer Immunology”, *Communications of the ACM*, **40**(10), pp. 88-96.
- [149] **Forrest, S., A. Perelson, Allen, L. & Cherukuri, R. (1994)**, “Self-Nonself Discrimination in a Computer”, *Proc. do IEEE Symposium on Research in Security and Privacy*, pp. 202-212.
- [150] **Forrest, S., Javornik, B., Smith, R. E. & Perelson, A. S. (1993)**, “Using Genetic Algorithms to Explore Pattern Recognition in the Immune System”, *Evolutionary Computation*, **1**(3), pp. 191-211.
- [151] **Forrest, S. & A. Perelson (1992)**, “Computation and the Immune System”, *SIGBIO Newsletter, Association for Computing Machinery*, **12**(2), pp. 52-57.
- [152] **Elanayar, S. & Shin, Y. C. (1994)**, “Radial Basis Function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic Systems”, *IEEE Trans. on Neural Networks*, **5**(4), pp. 594-603.
- [153] **Freeman, J. A. S. & Saad, D. (1995)**, “Learning and Generalization in Radial Basis Function Networks”, *Neural Computation*, **7**, pp. 1000-1020.
- [154] **Fritzke B. (1997)**, “A Self-Organizing Network that Can follow Non-Stationary Distributions”, *Proc. do ICANN*, Springer, pp. 613-618.
- [155] **Fritzke B. (1996)**, “Growing Self-Organizing Networks – Why?”, *Proc. do ESANN*, D-Facto Publishers, Brussels, pp. 61-72.
- [156] **Fritzke B. (1994a)**, “Growing Cell Structures – A Self-Organizing Network for Unsupervised and Supervised Learning”, *Neural Networks*, **7**(9), pp. 1441-1460.

-
- [157] **Fritzke, B. (1994b)**, “Supervised Learning with Growing Cell Structures”, *Advances in Neural Information Processing Systems*, **6**, pp. 255-262.
- [158] **Fukuda, T., Mori, K. & Tsukiyama, M. (1998)**, “Parallel Search for Multi-Modal Function Optimization with Diversity and Learning of Immune Algorithm”, In (Ed.) D. Dasgupta, *Artificial Immune Systems and Their Applications*, Springer-Verlag, pp. 210-220.
- [159] **Gardner, H. (1985)**, *The Mind’s New Science: A History of the Cognitive Revolution*, Basic Books, New York.
- [160] **Garzon, M. H. & Deaton, R. J., (1999)**, “Biomolecular Computing and Programming”, *IEEE Trans. on Evol. Computation*, **3**(3), pp. 236-250.
- [161] **Gaspar, A. & Collard, P. (2000)**, “Two Models of Immunization for Time Dependent Optimization”, *Proc. do IEEE SMC’00*, **1**, pp. 113-118.
- [162] **Gaspar, A. & Collard, P. (1999)**, “From GAs to Artificial Immune Systems: Improving Adaptation in Time Dependent Optimization”, *Proc. do CEC’99*, **3**, pp. 1859-1866.
- [163] **George, A. J. T. (2000)**, “Jumping or Walking: Which is Better”, *Imm. Today*, **21**(1), pp. 56.
- [164] **George, A. J. T. & Gray, D. (1999)**, “Receptor Editing during Affinity Maturation”, *Imm. Today*, **20**(4), pp. 196.
- [165] **Germain, R. N. (1995)**, “MHC-Associated Antigen Processing, Presentation, and Recognition Adolescence, Maturity and Beyond”, *The Immunologist*, **3/5-6**, pp. 185-190.
- [166] **Germain, R. N. (1994)**, “MHC-Dependent Antigen Processing and Peptide Presentation: Providing Ligands for T Lymphocyte Activation”, *Cell*, **76**, pp. 287-299.
- [167] **Goldberg, D. E. (1989)**, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Reading, Massachusetts.
- [168] **Goldberg, D. E. & Richardson, J. (1987)**, “Genetic Algorithms With Sharing for Multimodal Function Optimization”, *Genetic Algorithms and Their Applications: Proc. of the II ICGA*, pp. 41-49.
- [169] **Gomm, J. B. & Yu, D. L. (2000)**, “Selecting Radial Basis Function Network Centers with Recursive Orthogonal Least Squares Training”, *IEEE Trans. on Neural Networks*, **11**(2):306-314.
- [170] **Good, I. J. (1986)**, “(Neo) Darwinism”, *Physica 22D*, pp. 13-30.
- [171] **Goodman, D. J., Henry, P. S. & Prabhu, V. K. (1980)**, “Frequency-Hopped Multilevel FSK for Mobile Radio”, *Bell System Technical Journal*, **59**(7), pp. 1257-1275.
- [172] **Goonatiake, S. & S. Khebbal (1995)**, “Intelligent Hybrid Systems”, John Wiley & Sons.
- [173] **Gray, D. L. & Michel, A. N. (1992)**, A Training Algorithm for Binary Feedforward Networks, *IEEE Trans. on Neural Networks*, **3**(2), pp. 176-194.
- [174] **Grossberg, S., Mingolla, E. & Todorovic, D. (1989)**, “A Neural Network Architecture for Preattentive Vision”, *IEEE Trans. on Biomedical Engineering*, **36**(1), pp. 65-84.
- [175] **Grossman, Z. (1993)**, “Cellular Tolerance as a Dynamic State of the Adaptable Lymphocyte”, *Imm. Rev.*, **133**, pp. 45-73.
- [176] **Gu, J., Lee, D., Park, S. & Sim, K. (2000)**, “An Immunity-based Security Layer Model”, *Proc. do GECCO’00*, Workshop on Artificial Immune Systems and Their Applications, pp. 47-48.
- [177] **Gudwin, R. R. (1996)**, *Contribuições ao Estudo Matemático de Sistemas Inteligentes*, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, Unicamp.
-

- [178] **Hajela, P., & Yoo, J. S. (1999)**, “Immune Network Modelling in Design Optimization”, *New Ideas in Optimization*, (Eds.) D. Corne, M. Dorigo & F. Glover, McGraw Hill, London, pp. 203-215.
- [179] **Hajela, P., Yoo, J. & Lee, J. (1997)**, “GA Based Simulation of Immune Networks - Applications in Structural Optimization”, *Journal of Engineering Optimization*, **29**, pp. 131-149.
- [180] **Hajela, P., & Lee, J. (1996)**, “Constrained Genetic Search via Schema Adaptation: An Immune Network Solution”, *Structural Optimization*, **12**(1), pp. 11-15.
- [181] **Hamann, A. & Rebstock, S. (1993)**, “Migration of Activated Lymphocytes”, *Current Topics in Microbiology and Immunology*, **184**, pp. 110-124.
- [182] **Han, S., Dillon, S. R., Zheng, B., Shimoda, M., Schlissel, M. S. & Kelsoe, G. (1997)**, “V(D)J Recombinase Activity in a Subset of Germinal Center B Lymphocytes”, *Science*, **278**, pp. 301-305.
- [183] **Harmer, P. K. & Lamont, G. B. (2000)**, “An Agent Based Architecture for a Computer Virus Immune System”, *Proc. do GECCO'00*, Workshop on Artificial Immune Systems and Their Applications, pp. 45-46.
- [184] **Hart, E. & Ross, P. (1999a)**, “The Evolution and Analysis of a Potential Antibody Library for Use in Job-Shop Scheduling”, *New Ideas in Optimization*, D. Corne, M. Dorigo & F. Glover (Eds.), McGraw Hill, London, pp. 185-202.
- [185] **Hart, E. & Ross, P. (1999b)**, “An Immune System Approach to Scheduling in Changing Environments”, *Proc. do GECCO'99*, pp. 1559-1566.
- [186] **Hart, E., Ross, P. & Nelson, J. (1998)**, “Producing Robust Schedules Via An Artificial Immune System”, *Proc. do ICEC'98*.
- [187] **Hartigan, J. A. (1967)**, “Representations of Similarity Matrices by Trees”, *Journal of the American Statistical Association*, **62**, pp. 1440-1158.
- [188] **Hay, J. B. & Andrade, W. N. (1998)**, “Lymphocyte Recirculation, Exercise, and Immune Responses”, *Can. J. Physiol. Pharmacol.*, **76**, pp. 490-496.
- [189] **Haykin S. (1999)**, *Neural Networks – A Comprehensive Foundation*, Prentice Hall, 2nd Ed.
- [190] **Hightower, R. R., Forrest S. & Perelson, A. S. (1996)**, “The Baldwin Effect in the Immune System: Learning by Somatic Hypermutation”, *R. K. Belew and M. Mitchel (Eds.), Adaptive Individuals in Evolving Populations*, Addison-Wesley, Reading, MA, pp. 159-167.
- [191] **Hightower R. R., Forrest, S. A & Perelson, A. S. (1995)**, “The Evolution of Emergent Organization in Immune System Gene Libraries”, *Proc. of the Sixth ICGA*, L. J. Eshelman (Ed.), Morgan Kaufmann, San Francisco, CA, pp. 344-350.
- [192] **Hodgkin, P. D. (1998)**, “Role of Cross-Reactivity in the Development of Antibody Responses”, *The Immunologist*, **6/6**, pp. 223-226.
- [193] **Hoffmann, G. W. (1986)**, “A Neural Network Model Based on the Analogy with the Immune System”, *J. theor. Biol.*, **122**, pp. 33-67.
- [194] **Hoffmann, G. W., Benson, M. W., Bree, G. M. & Kinahan, P. E. (1986)**, “A Teachable Neural Network Based on an Unorthodox Neuron”, *Physica 22D*, pp. 233-246.
- [195] **Hoffmann, G. W. (1975)**, “A Theory of Regulation and Self-Nonself Discrimination in an Immune Network”, *Eur. J. Imm.*, **5**, pp. 638-647.

-
- [196] **Hofmeyr S. A. (2000)**, “An Interpretative Introduction to the Immune System”, *Design Principles for the Immune System and Other Distributed Autonomous Systems*, I. Cohen & L. A. Segel (Eds.), Oxford University Press.
- [197] **Hofmeyr S. A. & Forrest, S. (2000)**, “Architecture for an Artificial Immune System”, *Evolutionary Computation*, **7**(1), pp. 45-68.
- [198] **Hofmeyr S. A. & Forrest, S. (1999)**, “Immunity by Design: An Artificial Immune System”, *Proc. of GECCO'99*, pp. 1289-1296.
- [199] **Hofmeyr S. A. (1997)**, “An Overview of the Immune System”, *Tutorial sobre imunologia computacional*, URL: <http://www.cs.unm.edu/~steveah/imm-html/immune-system.html>.
- [200] **Holcombe, M. & Bell, A. (1998)**, “Computational Models of Immunological Pathways”, *Information Processing in Cell and Tissues*, M. Molcombe & R. Paton (Eds.), Plenum Press, pp. 213-226.
- [201] **Holland, J. H. (1998)**, *Adaptation in Natural and Artificial Systems*, 5^a Ed., MIT Press.
- [202] **Hopfield, J. J. (1984)**, “Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons”, *Proc. Natl. Acad. Sci. USA*, **81**, pp. 3088-3092.
- [203] **Hopfield, J. J. (1982)**, “Neural Networks and Physical Systems with Emergent Collective Computational Capabilities”, *Proc. Natl. Acad. Sci. USA*, **79**, pp. 2554-2558.
- [204] **Horowitz, E & Sahni, S. (1978)**, *Fundamentals of Computer Algorithms*, London, Pitman.
- [205] **Huberman, B. A. & Hoog, T. (1986)**, “Complexity and Adaptation”, *Physica 22D*, pp. 376-384.
- [206] **Hubert, L., Arabie, P. & Meulman, J. (1998)**, “Graph-Theoretic Representations for Proximity Matrices Through Strongly-Anti-Robinson or Circular Strongly-Anti-Robinson Matrices”, *Psychometrika*, **63**(4), pp. 341-358.
- [207] **Hunt, J. E., Timmis, J., Cooke, D. E., Neal, & King, C. (1998)**, “JISYS: The Development of An Artificial Immune System for Real World Applications”, D. Dasgupta (Ed.), *Artificial Immune Systems and Their Applications*, Springer-Verlag, pp 157-186.
- [208] **Hunt, J. E. & Cooke, D. E. (1996)**, “Learning Using an Artificial Immune System”, *Journal of Network and Computer Applications*, **19**, pp. 189-212.
- [209] **Hunt, J. E. & Fellows, A. (1996)**, “Introducing an Immune Response into a CBR system for Data Mining”, *BCS ESG'96 Conference*.
- [210] **Hunt, J. E., Cooke, D. E. & Holstein, H. (1995)**, “Case Memory and Retrieval Based on the Immune System”, *ICCBR*, Published as Case-Based Reasoning Research and Development, Ed. Manuela Weloso and Agnar Aamodt, Lecture Notes in Artificial Intelligence 1010, pp 205 -216.
- [211] **Hwang, Y-S & Bang, S-Y. (1997)**, “An Efficient Method to Construct a Radial Basis Function Neural Network Classifier”, *Neural Networks*, **10**(8), pp. 1495-1503.
- [212] **Ienne, P. & Kuhn, G. (1995)**, “Digital Systems for Neural Networks”, In P. Papamichalis and R. Kerwin (Eds.), *Digital Signal Processing Technology*, **CR57**, pp. 314-345.
- [213] **Inman, J. K. (1978)**, “The Antibody Combining Region: Speculations on the Hypothesis of General Multispecificity”, *Theoretical Immunology*, (Eds.) G. I. Bell, A. S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc., pp. 243-278.
- [214] **Ishida, Y. (1996a)**, “An Immune Network Approach to Sensor-based Diagnosis by Self-Organization”, *Complex Systems*, **10**, pp. 73-90.
-

- [215] **Ishida, Y. (1996b)**, “The Immune System as a Self-Identification Process: A Survey and A Proposal”, *Proc. of the IMBS’96*.
- [216] **Ishida, Y. (1996c)**, “Agent-Based Architecture of Selection Principle in the Immune System”, *Proc. of the IMBS’96*, pp. 92-104.
- [217] **Ishida, Y. (1993)**, “An Immune Network Model and Its Applications to Process Diagnosis”, *Systems and Computers in Japan*, **24**(6), pp. 646-651.
- [218] **Ishiguro, A., Watanabe, Y. & Kondo, T. (1997)**, “A Robot with a Decentralized Consensus-Making Mechanism Based on the Immune System”, *Proc. ISADS’97*, pp. 231-237.
- [219] **Ishiguro, A. Watanabe, Y., Kondo, T., Shirai, Y. & Uchikawa, Y. (1996)**, “Immunoid: A Robot with a Decentralized Behavior Arbitration Mechanisms Based on the Immune System”, *Proc. of the IMBS’96*.
- [220] **Jain, A. K., Murty, M. N. & Flynn, P. J. (2000)**, “Data Clustering: A Review”, *submetido*.
- [221] **Jain, A. K. & Mao, J. (1994)**, “Neural Networks and Pattern Recognition”, J. M. Zurada, R. J. Marks II & C. J. Robinson (Eds.), *Computational Intelligence Imitating Life*, IEEE Press, pp. 194-212.
- [222] **Janeway, C. A., P. Travers, Walport, M. & Capra, J. D. (2000)**, “Imunobiologia: O Sistema Imunológico na Saúde e na Doença”, Artes Médicas, 4^a Ed.
- [223] **Janeway Jr, C. A. (1993)**, “How the Immune System recognizes Invaders”, *Scientific American*, **269**(3), pp. 41-47.
- [224] **Janeway Jr., C. A. (1992)**, “The Immune System Evolved to Discriminate Infectious Nonself from Noninfectious Self”, *Imm. Today*, **13**(1), pp. 11-16.
- [225] **Jayasuriya, A. & Halgamuge, S. K. (1998)**, “An Enhanced Clustering Method for Multiple Shape Basis Function Networks”, In *Proc. of the IJCNN’98*, pp. 7-11.
- [226] **Jerne, N. K. (1985)**, “The Generative Grammar of the Immune System”, *Science*, **229**, pp. 1057-1059.
- [227] **Jerne, N. K. (1984)**, “Idiotypic Networks and Other Preconceived Ideas”, *Imm. Rev.*, **79**, pp. 5-24.
- [228] **Jerne, N. K. (1974a)**, “Towards a Network Theory of the Immune System”, *Ann. Immunol. (Inst. Pasteur)* **125C**, pp. 373-389.
- [229] **Jerne, N. K. (1974b)**, “Clonal Selection in a Lymphocyte Network”. *Cellular Selection and Regulation in the Immune Response*, G. M. Edelman (Ed.), Raven Press, N. Y., p. 39.
- [230] **Jerne, N. K. (1973)**, “The Immune System”, *Scientific American*, **229**(1), pp. 52-60.
- [231] **Johnsonbaugh, R. (1997)**, *Discrete Mathematics*, Prentice Hall, 4th ed.
- [232] **Jun, J.-H., Lee, D.-W. & Sim, K.-B., (1999)**, “Realization of Cooperative and Swarm Behavior in Distributed Autonomous Robotic Systems Using Artificial Immune System”, *Proc. do IEEE SMC’99*, **4**, pp. 614-619.
- [233] **Karayannis, N. B. & Mi, G. W. (1997)**, “Growing Radial Basis Neural Networks: Merging Supervised and Unsupervised Learning with network Growth Techniques”, *IEEE Trans. on Neural Networks*, **8**(6), pp. 1492-1506.
- [234] **Kauffman, S. A. & Weinberger, E. D. (1989)**, “The NK Model of Rugged Fitness Landscapes and Its Application to Maturation of the Immune Response”, *J. theor. Biol.*, **141**, pp. 211-245.

-
- [235] **Kephart, J. O., Sorkin, G. B. & Swimmer, M. (1997a)**, “An Immune System for Cyberspace”, *Proc. do IEEE SMC’97*, pp. 879-884.
- [236] **Kephart, J. O., Sorkin, G. B., Swimmer, M. & White S. R. (1997b)**, “Blueprint for a Computer Immune System”, *Apresentado na Conferência Internacional Virus Bulletin*.
- [237] **Kephart, J. O. (1994)**, “A Biologically Inspired Immune System for Computers”, R. A. Brooks & P. Maes (Eds.), *Artificial Life IV Proc. of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, MIT Press, pp. 130-139.
- [238] **Kepler, T. B. & Perelson, A. S. (1993a)**, “Somatic Hypermutation in B Cells: An Optimal Control Treatment”, *J. theor. Biol.*, **164**, pp. 37-64.
- [239] **Kepler, T. B. & Perelson, A. S. (1993b)**, “Cyclic Re-enter of Germinal Center B Cells and the Efficiency of Affinity Maturation”, *Imm. Today*, **14**(8), pp. 412-415.
- [240] **Kim, J. & Bentley, P. (1999a)**, “The Human Immune System and Network Intrusion Detection”, *Proc. of the EUFIT’99*, CD ROM.
- [241] **Kim, J. & Bentley, P. (1999b)**, “Negative Selection and Niching by an Artificial Immune System for Network Intrusion Detection”, *Proc. of GECCO’99*, pp. 149-158.
- [242] **Kim, Y. K. & Ra, J. B. (1991)**, “Weight Value Initialization for Improving Training Speed in the Backpropagation Network”, *Proc. do IEEE IJCNN*, **3**, pp. 2396-2401.
- [243] **Kinnear Jr., K. E. (1994)**, *Advances in Genetic Programming*, MIT Press.
- [244] **Kirkpatrick, S., Gelatt Jr., C. D. & Vecchi, M. P. (1987)**, “Optimization by Simulated Annealing”, *Science*, **220**(4598), pp. 671-680.
- [245] **Klein, J. (1995)**, “Ehrlich and Darwin: Homochauvinism in Immunology”, *Immunology and Cell Biology*, **73**, pp. 103-108.
- [246] **Klein, J. (1990)**, *Immunology*, Blackwell Scientific Publications.
- [247] **Kleinstein, S. & Seiden, P. E. (2000)**, “Simulating the Immune System”, *Computing in Sciences and Engineering*, July/August, pp. 69-77.
- [248] **Kohonen T. (1982)**, “Self-Organized Formation of Topologically Correct Feature Maps”, *Biological Cybernetics*, **43**, pp. 59-69.
- [249] **Koza., J. R. (1992)**, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press.
- [250] **KrishnaKumar, K. & Neidhoefer, J. (1997a)**, “Immunized Adaptive Critics for Level 2 Intelligent Control”, *Proc. do IEEE SMC’97*, **1**, pp. 856-860.
- [251] **KrishnaKumar, K. & Neidhoefer, J. (1997b)**, “Immunized Neurocontrol”, *Expert Systems with Applications*, **13**(3), pp. 201-214.
- [252] **KrishnaKumar, K. Satyadas, A. & Neidhoefer, J. (1995)**, “An Immune System Framework for Integrating Computational Intelligence Paradigms with Applications to Adaptive Control”, *Computational Intelligence A Dynamic System Perspective*, M. Palaniswami, Y. Attikiouzel, R. J. Marks II, D. Fogel and T. Fukuda (Eds.), IEEE Press, pp. 32-45.
- [253] **Kruisbeek, A. M. (1995)**, “Tolerance”, *The Immunologist*, **3/5-6**, pp. 176-178.
- [254] **Kwok T. Y., & Yeung. D. Y. (1997)**, “Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems”, *IEEE Trans. on Neural Networks*, **8**(3), pp. 630-645.
-

- [255] **Lamont, G. B., Marmelstein, R. E. & Van Veldhuizen D. A. (1999)**, “A Distributed Architecture for a Self-Adaptive Computer Virus Immune System”, *New Ideas in Optimization*, D. Corne, M. Dorigo & F. Glover (Eds.), McGraw Hill, London, pp. 167-183.
- [256] **Langman, R. E. & Cohn, M. (1986)**, “The ‘Complete’ Idiotypic Network is an Absurd Immune System”, *Imm. Today*, **7**(4), pp. 100-101.
- [257] **Langton, C. G. (Ed.), (1995)**, *Artificial Life An Overview*, The MIT Press, Cambridge Massachussets.
- [258] **Lapedes, A. & Farber, R. (1986)**, “A Self-Organizing, Nonsymmetrical Neural Net for Content Addressable Memory and Pattern Recognition”, *Physica 22D*, pp. 247-259.
- [259] **Lapointe, F. -J. & Legendre, P. (1995)**, “Comparison Tests for Dendrograms: A Comparative Evaluation”, *Journal of Classification*, **12**, pp. 265-282.
- [260] **Lapointe, F. -J. & Legendre, P. (1991)**, “The Generation of Random Ultrametric Matrices Representing Dendrograms”, *Journal of Classification*, **8**, pp. 177-200.
- [261] **Leake, D. B. (1996)**, *Case-Based Reasoning, Experiences, Lessons & Future Directions*, AAAI Press/The MIT Press.
- [262] **Leclerc, B. (1995)**, “Minimum Spanning Trees for Tree Metrics: Abridgements and Adjustments”, *Journal of Classification*, **12**, pp. 207-241.
- [263] **Lederberg, J. (1988)**, “Ontogeny of the Clonal Selection Theory of Antibody Formation”, *Annals of the New York Ac. of Sc.*, **546**, pp. 175-182.
- [264] **Lee, D-W., Jun, H-B. & Sim, K-B. (1999)**, “Artificial Immune System for Realization of Cooperative Strategies and Group Behavior in Collective Autonomous Mobile Robots”, In *Proc. of the AROB'99*, pp. 232-235.
- [265] **Lehtokangas, M., Saarinen, J., Kaski, K. & Huuhtanen, P. (1995)**, “Initializing Weights of a Multilayer Perceptron by Using the Orthogonal Least Squares Algorithm”, *Neural Computation*, **7**, pp. 982-999.
- [266] **Lerner, B., Guterman, H., Aladjem, M. & Dinstein, I. (1999)**, “A Comparative Study of Neural Network Based Feature Extraction Paradigms”, *Pattern Recognition Letters*, **20**, pp. 7-143.
- [267] **Lichtenstein, L. M. (1993)**, “Allergy and the Immune System”, *Scientific American*, **269**(3), pp. 85-93.
- [268] **Linnemeyer, P. A. (1993)**, “The Immune System – An Overview”, *A Fact Sheet from the Seattle Treatment Education Project*, URL: <http://www.thebody.com/step/immune.html>.
- [269] **Lippmann, R. P. (1989)**, “Pattern Classification Using Neural Networks”, *IEEE Communications Magazine*, pp. 47-63.
- [270] **Lippmann, R. P. (1987)**, “An Introduction to Computing with Neural Nets”, *IEEE ASSP Magazine*, pp. 4-22.
- [271] **Lo, D., Feng, L., Li, L. et al. (1999)**, “Integrating Innate and Adaptive Immunity in the Whole Animal”, *Imm. Rev.*, **169**, pp. 225-239.
- [272] **Lu, H., Setiono, R & Liu, H. (1996)**, “Effective Data Mining Using Neural Networks”, *IEEE Trans. on Knowledge and Data Engineering*, **8**(6), pp. 957-961.
- [273] **Mahfoud, S. W. (1995)**, *Niching Methods for Genetic Algorithms*, Tese de Doutorado, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.
- [274] **Mannie, M. D. (1999)**, “Immunological Self/Nonself Discrimination”, *Immunologic Research*, **19**(1), pp. 65-87.

-
- [275] **Mao, J. & Jain, A. K. (1996)**, "A Self-Organizing Network for Hyperellipsoidal Clustering (HEC)", *IEEE Trans. on Neural Networks*, **7**(1), pp. 16-29.
- [276] **Mao, J. & Jain, A. K. (1995)**, "Artificial Neural Networks for Feature Extraction and Multivariate Data Projection", *IEEE Trans. on Neural Networks*, **6**(2), pp. 296-317.
- [277] **Marrack, P. & Kappler, J. W. (1993)**, "How the Immune System Recognizes the Body", *Scientific American*, **269**(3), pp. 49-55.
- [278] **Mason, D. (1998)**, "Antigen Cross-Reactivity: Essential in the Function of TCRs", *The Immunologist*, **6/6**, pp. 220-222.
- [279] **Matsui, K. (1999)**, "New Selection Method to Improve the Population Diversity in Genetic Algorithms", *Proc. do IEEE SMC'99*, **I**, pp. 625.
- [280] **McConkey, D. J., Orrenius, S. & Jondal, M. (1990)**, "Cellular Signalling in Programmed Cell Death (Apoptosis)", *Imm. Today*, **11**(4), pp. 120-121.
- [281] **McCulloch W. & Pitts W. (1943)**, "A Logical Calculus of the Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics*, **5**, pp. 115-133.
- [282] **Medzhitov, R. & Janeway Jr., C. A. (1998)**, "Innate Immune Recognition and Control of Adaptive Immune Responses", *Seminars in Imm.*, **10**, pp. 351-353.
- [283] **Medzhitov, R. & Janeway Jr., C. A. (1997a)**, "Innate Immunity: Impact on the Adaptive Immune Response", *Current Opinion in Imm.*, **9**, pp. 4-9.
- [284] **Medzhitov, R. & Janeway Jr., C. A. (1997b)**, "Innate Immunity: The Virtues of a Nonclonal System of Recognition", *Cell*, **91**, pp. 295-298.
- [285] **McCoy, D. F. & Devaralan, V. (1997)**, "Artificial Immune Systems and Aerial Image Segmentation", *Proc. do SMC'97*, pp. 867-872.
- [286] **Micchelli, C. A. (1986)**, "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions", *Construct. Approx.*, **2**, pp. 11-22.
- [287] **Michalewicz, Z. (1996)**, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, 3^a Ed.
- [288] **Miller, R. A. (1999)**, "Aging and the Immune Function", *Fundamental Immunology*, 4^a ed., W. E. Paul (Ed.), Lippincott-Raven Publishers, Philadelphia, pp. 947-966.
- [289] **Milligan, G. W. & Cooper, M. C. (1985)**, "An Examination of Procedures for Determining the Number of Clusters in a Data Set", *Psychometrika*, **50**(2), pp. 159-179.
- [290] **Mitchel, M. (1998)**, *An Introduction to Genetic Algorithms*, The MIT Press.
- [291] **Mitchel, M. & Forrest, S. (1997)**, "Genetic Algorithms and Artificial Life", *Artificial Life An Overview*, G. Langton (Ed.), pp. 267-290.
- [292] **Mitchison, N. A. (1994)**, "Cognitive Immunology", *The Immunologist*, **2/4**, pp. 140-141.
- [293] **Moller, M. F. (1993)**, "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning", *Neural Networks*, **6**, pp. 525-533.
- [294] **Mohler, R. R., Bruni, C. & Gandolfi, A. (1980)**, "A Systems Approach to Immunology", *Proc. of the IEEE*, **68**(8), pp. 964-989.
- [295] **Moody, J. & Darken, C. (1989)**, "Fast Learning in Networks of Locally-Tuned Processing Units", *Neural Computation*, **1**, pp. 281-294.
- [296] **Moscato, P. & Fontanari, J. F. (1990)**, "Stochastic Versus Deterministic Update in Simulated Annealing", *Physics Letters A*, **146**(4), pp. 204-208.
-

- [297] Müller, W. (1995), "Dissecting the Cytokine Network", *The Immunologist*, **3/5-6**, pp. 216-218.
- [298] Murtag, F. & Hernández-Pajares, M. (1995), "The Kohonen Self-Organizing Method: An Assessment", *Journal of Classification*, **12**, pp. 165-190.
- [299] Nagano, S. & Yonezawa, Y., (1999), "Generative Mechanism fo Emergent Properties Observed with the Primitive Evolutional Phenomena by Immunotic Recognition", *Proc. do IEEE SMC'99*, **I**, pp. 223-228.
- [300] Nguyen, D. & Widrow, B. (1990), "Improving the Learning Speed of Two-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights", *Proc. do IJCNN*, **3**, pp. 21-26.
- [301] Nikolaev, N. I., Iba, H. & Slavov, V. (1999), "Inductive Genetic Programming with Immune Network Dynamics", *Advances in Genetic Programming 3*, MIT Press, pp. 355-376.
- [302] Nossal, G. J. V. (1994), "Negative Selection fo Lymphocytes", *Cell*, **76**, pp. 229-239.
- [303] Nossal, G. J. V. (1993a), "Life, Death and the Immune System", *Scientific American*, **269**(3), pp. 21-30.
- [304] Nossal, G. J. V. (1993b), "The Molecular and Cellular Basis of Affinity Maturation in the Antibody Response", *Cell*, **68**, pp. 1-2.
- [305] Nussenzweig, M. C. (1998), "Immune Receptor Editing; Revise and Select", *Cell*, **95**, pp. 875-878.
- [306] Okamoto, T. & Ishida, Y. (1999a), "A Distributed Approach to Computer Virus Detection and Neutralization by Autonomous and Heterogeneous Agents", *Proc. do ISADS'99*, pp. 328-331.
- [307] Okamoto, T. & Ishida, Y. (1999b), "Multiagent Approach Against Computer Virus: An Immunity-Based System", *Proc. do AROB'99*, pp. 69-72.
- [308] Oei, C. K., Goldberg, D. E. & Chang, S.-J. (1991), "Tournament Selection, Niching and the Preservation of Diversity", *IlligAL Report 91011*, University of Illinois, Illinois, genetic Algorithm Laboratory.
- [309] Ootsuki, J. T. & Sekiguchi, T. (1999), "Application of the Immune System Network Concept to Sequential Control", *Proc. do IEEE SMC'99*, **3**, pp. 869-874.
- [310] Oprea, M. (1999), *Antibody Repertoires and Pathogen Recognition: The Role of Germline Diversity and Somatic Hipermutation*, Tese de Doutorado, Universidade do Novo México, Albuquerque, Novo México, EUA.
- [311] Oprea, M. & Kepler, T. B. (1999), "Genetic Plasticity of V Genes Under Somatic Hypermutation: Statistical Analyses Using a New Resampling-Based Methodology", *Genome Research*, **9**, pp. 1294-1304.
- [312] Oprea, M. & Forrest, S. (1999), "How the Immune System Generates Diversity: Pathogen Space Coverage with Random and Evolved Antibody Libraries", *Proc. do GECCO'99*, **2**, pp. 1651-1656.
- [313] Oprea, M. & Forrest, S. (1998), "Simulated Evolution of Antibody Gene Libraries Under Pathogen Selection", *Proc. do IEEE SMC'98*.
- [314] Orr, M. J. L. (1995), "Regularisation in the Selection of Radial Basis Function Centres", *Neural Computation*, **7**(3):606-623.

-
- [315] **Papavasiliou, F., Casellas, R., Suh, H., Qin, X.-F., Besmer, E., Pelanda, R., Nemazee, D., Rajewsky, K. & Nussenzweig, M. C., (1997)**, “V(D)J Recombination in Mature B Cells: A Mechanism for Altering Antibody Responses”, *Science*, **278**, pp. 298-301.
- [316] **Parijs, L. V. & Abbas, A. K. (1998)**, “Homeostasis and Self-Tolerance in the Immune System: Turning Lymphocytes Off”, *Science*, **280**, pp. 243-248.
- [317] **Parikh, J. A. Da Ponte, J. S. Vitale, J. N. & Tselioudis, G. (1999)**, “An Evolutionary System for Recognition and Tracking of Synoptic-Scale Storm Systems”, *Pattern Recognition Letters*, **20**, pp. 1389-1396.
- [318] **Parish, C. R. & O’Neill, E. R. (1997)**, “Dependence of the Adaptive Immune Response on Innate Immunity: Some Questions Answered but New Paradoxes Emerge”, *Imm. and Cell Biol.*, **75**, pp. 523-527.
- [319] **Pedrycz, W. & Gomide, F. A. C. (1998)**, *An Introduction to Fuzzy Sets Analysis and Design*, MIT Press.
- [320] **Percus, J. K., Percus, O. E. & Perelson, A. S. (1993)**, “Predicting the Size of the T-cell Receptor and Antibody Combining Region from Consideration of Efficient Self-Nonself Discrimination”, *Proc. Natl. Acad. Sci. USA*, **90**, pp. 1691-1695.
- [321] **Perelson, A. S. & Weigel, F. W. (1998)**, “Some Design Principles for Immune System Recognition”, *submetido ao jornal Complexity*.
- [322] **Perelson, A. S. & Weisbuch, G. (1997)**, “Immunology for Physicists”, *Rev. of Modern Physics*, **69**(4), pp. 1219-1267.
- [323] **Perelson, A. S., Hightower, R. & Forrest, S. (1996)**, “Evolution and Somatic Learning of V-Region Genes”, *Research in Immunology*, **147**, pp. 202-208.
- [324] **Perelson, A. S. (1988)**, “Towards a Realistic Model of the Immune System”, *Theoretical Immunology*, Part Two, A. S. Perelson (Ed.), pp. 377-401.
- [325] **Perelson, A. S. (1989)**, “Immune Network Theory”, *Imm. Rev.*, **110**, pp. 5-36.
- [326] **Perelson, A. S. & Oster, G. F. (1979)**, “Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-Nonself Discrimination”, *J. theor.Biol.*, **81**, pp. 645-670.
- [327] **Perelson, A. S., Mirmirani, M. & Oster, G. F. (1978)**, “Optimal Strategies in Immunology II. B Memory Cell Production”, *J. Math. Biol.*, **5**, pp. 213-256.
- [328] **Perelson, A. S., Mirmirani, M. & Oster, G. F. (1976)**, “Optimal Strategies in Immunology I. B-Cell Differentiation and Proliferation”, *J. Math. Biol.*, **3**, pp. 325-367.
- [329] **Piatelli-Palmarini, M. (1986)**, “The Rise of Selective Theories: A Case Study and Some Lessons from Immunology”, *Language Learning and Concept Acquisition*, Demopoulos, W. & Marros, A. (Eds.).
- [330] **Poggio, T. & Girosi, F. (1990)**, “Networks for Approximation and Learning”, *Proc. of the IEEE*, **78**(9), pp. 1481-1497.
- [331] **Potter, M. A. & De Jong K. A. (1998)**, “The Coevolution of Antibodies for Concept Learning”, *Proc. do 5th ICPPSN*, pp. 530-539.
- [332] **Prechelt L. (1998)**, “Automatic Early Stopping Using Cross Validation: Quantifying the Criteria”, *Neural Networks*, **11**(4), pp. 761-767.
- [333] **Prim, R. C. (1957)**, “Shortest Connection Networks and Some Generalizations”, *Bell Sys. Tech. J.*, pp. 1389-1401.
-

- [334] **Qi, X. & Palmieri, F. (1994a)**, “Theoretical Analysis of Evolutionary Algorithms with an Infinite Population Size in Continuous Space, Part I: Basic Properties of Selection and Mutation”, *IEEE Trans. on Neural Networks*, **5**(1), pp. 102-119.
- [335] **Qi, X. & Palmieri, F. (1994b)**, “Theoretical Analysis of Evolutionary Algorithms with an Infinite Population Size in Continuous Space, Part I: Analysis of the Diversification Role of Crossover”, *IEEE Trans. on Neural Networks*, **5**(1), pp. 120-129.
- [336] **Reed, R. (1993)**, “Pruning Algorithms – A Survey”, *IEEE Trans. on Neural Networks*, **4**(5), pp. 740-747.
- [337] **Rechenberge, I. (1994)**, “Evolution Strategy”, *Computational Intelligence Imitating Life*, J. M. Zurada, R. J. Marks II and C. J. Robinson (Eds.), IEEE Press, pp. 147-159.
- [338] **Rechenberge, I. (1973)**, *Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stuttgart.
- [339] **Rensberger, B. (1996)**, “In Self-Defense”, *Life Itself*, B. Resenberger, Oxford University Press, pp. 212-228.
- [340] **Richter, P. H. (1978)**, “The Network Idea and the Immune Response”, *Theoretical Immunology*, G. I. Bell, A. S. Perelson & G. H. Pimbley Jr. (Eds.), Marcel Dekker Inc., pp. 539-569.
- [341] **Richter, P. H. (1975)**, “A Network Theory of the Immune System”, *Eur. J. Imm.*, **5**, pp. 350-354.
- [342] **Robinson, M. E., Yoneda, H. and Sinencio, E. S- (1992)**, “A Modular CMOS Design of a Hamming Network”, *IEEE Trans. on Neural Networks*, **3**(3), pp. 444-456.
- [343] **Rudolf, G. (1994)**, “Convergence Analysis of Canonical Genetic Algorithms”, *IEEE Trans. on Neural Networks*, **5**(1), pp. 96-101.
- [344] **Rumelhart, D. E., McClelland, J. L. & The PDP Research Group, eds. (1986)**, *Parallel Distributed Processing*, Cambridge MIT Press.
- [345] **Rumelhart, D. E. & Zipser, D. (1985)**, “Feature Discovery by Competitive Learning”, *Cognitive Science*, **9**, pp. 75-112.
- [346] **Russel, S. & Norvig, P. (1995)**, *Artificial Intelligence A Modern Approach*, Prentice Hall.
- [347] **Sarle, W. S. (1994)**, “Neural Networks and Statistical Models”, *Proc. of the Nineteenth Annual SAS Users Group International Conference*.
- [348] **Salomon, R. (1998)**, “Evolutionary Algorithms and Gradient Search; Similarities and Differences”, *IEEE Trans. on Evolutionary Computation*, **2**(2), pp. 45-55.
- [349] **Schmid, A., Leblebici, Y. & Mlyneck, D. (1998)**, “Hardware realization of a Hamming Neural Network with On-Chip Learning”, *IEEE Int. Symposium on Circuits and Systems*.
- [350] **Schwartz, R. S. & Banchereau, J. (1996)**, “Immune Tolerance”, *The Immunologist*, **4/6**, pp. 211-218.
- [351] **Schwefel, H. -P. (1994)**, “On the Evolution of Evolutionary Computation”, *Computational Intelligence Imitating Life*, J. M. Zurada, R. J. Marks II & C. J. Robinson (Eds.), IEEE Press, pp. 116-124.
- [352] **Scrimshaw, N. S. & SanGiovanni, J. P. (1997)**, “Synergism of Nutrition, Infection, and Immunity: An Overview”, *Ann. J. Clin. Nutr.*, **66**, pp. 464S-477S.
- [353] **Scroferneker, M. L. & Pohlmann, P. R. (1998)**, *Imunologia Básica e Aplicada*, Sagra Luzzatto.

-
- [354] **Secarz, E., Celada, F., Mitchinson, N. A. & Tada, T. (1988)**, *The Semiotics of Cellular Communication in the Immune System*, NATO ASI Series, Vol. H23, Springer-Verlag, New York.
- [355] **Sedgewick, R. (1988)**, *Algorithms*, 2nd ed., Reading, MA: Addison-Wesley.
- [356] **Segel, L. A. (1999)** “How Can Perception of Context Improve the Immune Response?”, *não publicado*. [Online]
- [357] **Segel, L. A. & Bar-Or, R. L. (1998)**, “Immunology Viewed as the Study of an Autonomous Decentralized System”, *Artificial Immune Systems and Their Applications*, Springer-Verlag, D. Dasgupta (Ed.), pp. 65-88.
- [358] **Segel, L. A. (1997)**, “The Immune System as a Prototype of Autonomous Decentralized Systems”, *Proc. do IEEE SMC'97*, **1**, pp. 375-384.
- [359] **Segel, L. & Perelson, A. S. (1988)**, “Computations in Shape Space: A New Approach to Immune Network Theory”, *Theoretical Immunology*, Parte II, A. S. Perelson (Ed.), pp. 321-343.
- [360] **Seiden, P. E. & Celada, F. (1992a)**, “A Model for Simulating Cognate Recognition and Response in the Immune System”, *J. theor. Biol.*, **158**, pp. 329-357.
- [361] **Selman, B., Brooks, R. A., Dean, T. Horvitz, E., Mitchell, T. M. & Nilsson, N. J. (1996)**, “Challenge Problems for Artificial Intelligence”, *Proc. do AAAI'96*.
- [362] **Sieburg, H., McCutchan, H., Clay, O., Caballero, L. & Ostlund, J. (1990)**, “Simulation fo HIV-infection in Artificial Immune Systems”, *Physica D*, **45**, pp. 208-228.
- [363] **Silverstein, A. M. (1985)**, “History of Immunology”, *Cellular Immun.*, **91**, pp. 263-283.
- [364] **Skinner, B. F. (1971)**, *Beyond Freedom and Dignity*, Alfred A. Knopf: New York.
- [365] **Slavov, V. & Nikolaev, N. I. (1998)**, “Immune Network Dynamics for Inductive Problem Solving”, *VPPSN*, pp. 712-721.
- [366] **Smith, D. J. (1997)**, “The Cross-Reactive Immune Response: Analysis, Modelling and Application to Vaccine Design”, Tese de Doutorado, Computer Science, The University of New Mexico.
- [367] **Smith, D. J., S. Forrest, D. A. Ackley & A. S. Perelson (1998)**, “Using Lazy Evaluation to Simulate Realistic-Size Repertoires in Models of the Immune System”, *Bulletin of Mathematical Biology*, **60**, pp. 647-658.
- [368] **Smith, D. J., Forrest, S., Hightower, R. R. & Perelson, S. A. (1997)**, “Deriving Shape Space Parameters from Immunological Data”, *J. theor. Biol.*, **189**, pp. 141-150.
- [369] **Smith, D. J., Forrest, & Perelson, S. A. (1993)**, “Searching for Diverse, Cooperative Populations with Genetic Algorithms”, *Evolutionary Computation*, **1**, pp. 127-149.
- [370] **Sprent, J. (1994)**, “T and B Memory Cells”, *Cell*, **76**, pp. 315-322.
- [371] **Springer, T. A. (1994)**, “Traffic Signals for Lymphocyte Recirculation and Leukocyte Emigration: The Multistep Paradigm”, *Cell*, **76**, pp. 301-314.
- [372] **Starlab**, URL: <http://www.starlab.org/genes/ais/>
- [373] **Stewart, J. & Varela, F. J. (1991)**, “Morphogenesis in Shape-space. Elementary Metadynamics in a model of the Immune Network”, *J. theor. Biol.*, **153**, pp. 477-498.
- [374] **Storb, U. (1998)**, “Progress in Understanding the Mechanism and Consequences of Somatic Hypermutation”, *Immun. Rev.*, **162**, pp. 5-11.
- [375] **Sutanto, E. L., Mason, J. D. & Warwick, K. (1997)**, “Mean-Tracking Clustering Algorithm for Radial Basis Function Centers Selection”, *Int. Journal of Control*, **67(6)**:961-977.
-

- [376] **Sutton, R. S. & Barto, A. G. (1998)**, *Reinforcement Learning an Introduction*, A Bradford Book.
- [377] **Suzuki, J. & Yamamoto, Y. (2000a)**, "A Decentralized Policy Coordination facility in Open WebServer", *Proc. do SPA'00*.
- [378] **Suzuki, J. & Yamamoto, Y. (2000b)**, "iNet: A Configurable Framework for Simulating Immune Network", *Proc. do IEEE SMC'00*, pp. 119-124.
- [379] **Suzuki, J. & Yamamoto, Y. (1998)**, "The Reflection Pattern in the Immune System", *Proc. do OOPSLA'98, workshop on Non-Software Examples of Software Architecture*.
- [380] **Tada, T. (1997)**, "The Immune System as a Supersystem", *Ann. Rev. Imm.*, **15**, pp. 1-13.
- [381] **Tarakanov A., Sokolova, S., Abramov, B. & Aikimbayev, A. (2000)**, "Immunocomputing of the Natural Plague Foci", *Proc. do GECCO'00*, Workshop on Artificial Immune Systems and Their Applications, pp. 38-39.
- [382] **Tarlinton, D. (1998)**, "Germinal Centers: Form and Function", *Current Opinion in Imm.*, **10**, pp. 245-251.
- [383] **Tauber, A. I. (1997)**, "Historical and Philosophical Perspectives on Immune Cognition", *Journal of the History of Biology*, **30**, pp. 419-440.
- [384] **Tauber, A. I. (1994)**, "The Immune Self: Theory or Metaphor", *Imm. Today*, **15**(3), pp. 134-136.
- [385] **Taylor, C. & Jefferson, D. (1995)**, "Artificial Life as a Tool for Biological Inquiry", *Artificial Life An Overview*, Langton, C. G. (Ed.), The MIT Press, Cambridge Massachussets, pp. 1-13.
- [386] **Timmis, J. (2000)**, *Artificial Immune Systems: A Novel Data Analysis Technique Inspired by the Immune Network Theory*, Tese de Doutorado, Department of Computer Science, University of Whales, September.
- [387] **Timmis, J., Neal, M. & Hunt, J. (2000)**, "An Artificial Immune System for Data Analysis", *BioSystems*, **55**, pp. 143-150.
- [388] **Timmis, J. & Neal, M. (2000)**, "Investigating the Evolution and Stability of a Resource Limited Artificial Immune System", *Proc. do GECCO'00*, Workshop on Artificial Immune Systems and Their Applications, pp. 40-41.
- [389] **Timmis, J., Neal, M. & Hunt, J. (1999)**, "Data Analysis Using Artificial Immune Systems, Cluster Analysis and Kohonen Networks: Some Comparisons", *Proc. do IEEE SMC'99*, **3**, pp. 922-927.
- [390] **Tizard, I. R. (1995)**, *Immunology An Introduction*, Saunders College Publishing, 4th Ed.
- [391] **Toma, N., Endo, S. & Yamada, K. (1999)**, "Immune Algorithm with Immune Network and MHC for Adaptive Problem Solving", *Proc. do IEEE SMC'99*, **IV**, pp. 271-276.
- [392] **Tonegawa, S. (1985)**, "The Molecules of the Immune System", *Scientific American*, **253**(4), pp. 104-113.
- [393] **Tonegawa, S. (1983)**, "Somatic Generation of Antibody Diversity", *Nature*, **302**, pp. 575-581.
- [394] **Unanue, E. R. & Benacerraf, B. (1986)**, *Imunologia*, Editora Guanabara.
- [395] **Ultsch, A. (1993)**, "Knowledge Extraction from Artificial Neural Networks and Applications", *Information and Classification*, O. Optiz et al. (Eds.), Springer, Berlin, pp. 307-313.

-
- [396] **Ultsch, A. (1995)**, “Self-Organizing Neural Networks Perform Different from Statistical k-means”, *Gesellschaft für Klassifikation*.
- [397] **URL 1 (1995)** “Immune System: An Internal Force Armed and Ready for Battle”, *Medical Essay, Mayo Clinic Health Letter*,
URL: <http://www.mayohealth.org/mayo/9502/htm/immunesy.htm>.
- [398] **URL 2** “Understanding the Immune System”, *National Institute of Health and National Cancer Institute*, U.S. Department of Health and Human Services, Public Health Service, URL: http://rex.nci.nih.gov/PATIENTS/INFO_TEACHER/bookshelf/NIH_immune/index.html.
- [399] **URL 3** Machine Learning Databases of the Institute of Computer Science of the University of Irvine, California [Online] <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>.
- [400] **Varela, F. J., Thompson, E. & Rosch, E. (1991)**, *The Embodied Mind*, MIT Press.
- [401] **Varela, F. J. & Coutinho, A. (1991)**, “Second Generation Immune Networks”, *Imm. Today*, **12**(5), pp. 159-166.
- [402] **Varela, F. J., Coutinho, A. Dupire, E. & Vaz, N. N. (1988)**, “Cognitive Networks: Immune, Neural and Otherwise”, *Theoretical Immunology*, Parte Dois, A. S. Perelson (Ed.), pp. 359-375.
- [403] **Varela, F. J. & Stewart, J. (1990)**, “Dynamics of a Class of Immune Networks. I. Global Stability of Idiotypic Interactions”, *J. theor. Biol.*, **144**, pp. 93-101.
- [404] **Varela, F. J. & Stewart, J. (1990)**, “Dynamics of a Class of Immune Networks II. Oscillatory Activity of Cellular and Humoral Components”, *J. theor. Biol.*, **144**, pp. 103-115.
- [405] **Vertosick, F. T. & Kelly, R. H. (1991)**, “The Immune System as a Neural Network: A Multi-epitope Approach”, *J. theor. Biol.*, **150**, pp. 225-237.
- [406] **Vertosick, F. T. & Kelly, R. H. (1989)**, “Immune Network Theory: A Role for Parallel Distributed Processing?”, *Immunology*, **66**, pp. 1-7.
- [407] **Von Boehmer, H. (1994)**, “Positive Selection of Lymphocytes”, *Cell*, **76**, pp. 219-228.
- [408] **Wang, Z-o & Zhu, T. (2000)**, “An Efficient Learning Algorithm for improving Generalization Performance of Radial Basis Function Neural Networks”, *Neural Networks*, **13**(4-5):545-553.
- [409] **Watanabe, Y., Ishiguro, A., Shirai, Y. & Uchikawa, Y. (1998)**, “Emergent Construction Behavior Arbitration Mechanism Based on the Immune System”, *Proc. do ICEC'98*, pp. 481-486.
- [410] **Weinand, R. G. (1990)**, “Somatic Mutation, Affinity Maturation and the Antibody Repertoire: A Computer Model”, *J. theor. Biol.*, **143**, pp. 343-382.
- [411] **Weisbuch, G. (1990)**, “A Shape-Space Approach to the Dynamics of the Immune System”, *J. theor. Biol.*, **143**, pp. 507-522.
- [412] **Weissman, I. L. & Cooper, M. D. (1993)** “How the Immune System Develops”, *Scientific American*, **269**(3), pp. 33-40.
- [413] **Wettschereck, D. & Dietterich, T. (1992)**, “Improving the Performance of Radial Basis Function Networks by Learning Center Locations”, *Advances in Neural Information Processing Systems*, **4**, pp. 1133-1140.
- [414] **Whitehead, B. A. & Choate, T. D. (1996)**, “Cooperative-Competitive genetic Evolution of Radial Basis Function Centers and Widths for Time Series Prediction”, *IEEE Trans. on Neural Networks*, **7**(4), pp. 869-880.
-

- [415] **Wierzhón, S. T. (2000a)**, “Discriminative Power of the Receptors Activated by k -Contiguous Bits Rule”, *J. of Computer Science and Technology*, **1**(3), pp. 1-13.
- [416] **Wierzhón, S. T. (2000b)**, “Generating Optimal Repertoire of Antibody Strings in an Artificial Immune System”, *Intelligent Information Systems*, M. A. Kłopotek, M. Michawicz & S. T. Wierzhon (Eds.), Physica-Verlag, Berlin-Heidelberg-NY, pp. 119-134.
- [417] **Witten, I. H. & MacDonald, B. A. (1988)**, “Using Concept Learning for Knowledge Acquisition”, *Int. J. Man-Machine Studies*, **29**, pp. 171-196.
- [418] **Wolfram, S. (1986)**, “Approaches to Complexity Engineering”, *Physica 22D*, pp. 385-399.
- [419] **Yingwei, L., Sundararajan, N. & Saratchandran, P. (1996)**, “A Sequential Learning Scheme for Function Approximation Using Minimal Radial Basis Function Neural Networks”, *Neural Computation*, 461-478.
- [420] **Yip, P. P. C. & Pao, Y-H. (1995)**, “Combinatorial Optimization with Use of Guided Evolutionary Simulated Annealing”, *IEEE Trans. on Neural Networks*, **6**(2), pp. 290-295.
- [421] **Young, A. J., Hay, J. B. & Mackay, C. R. (1993)**, “Lymphocyte Recirculation and Life Span in Vivo”, *Current Topics in Microbiology and Immunology*, **184**, pp. 161-173.
- [422] **Zadeh, L. A. (1965)**, “Fuzzy Sets”, *Information and Control*, **8**, pp. 338-353.
- [423] **Zahn, C. T. (1971)**, “Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters”, *IEEE Trans. on Computers*, **C-20**(1), pp. 68-86.
- [424] **Zinkernagel, R. M. & Kelly, J. (1997)**, “How Antigen Influences Immunity”, *The Immunologist*, **4/5**, pp. 114-120.

APÊNDICE

GLOSSÁRIO DA TERMINOLOGIA BIOLÓGICA

Este glossário visa apresentar uma descrição sucinta dos principais termos e expressões da biologia empregados no desenvolvimento desta tese.

A

Adenóides: formadas por tecido linfóide associado a mucosas e localizadas na cavidade nasal.

Afinidade: força de ligação de uma molécula à outra.

AIDS (síndrome da imunodeficiência adquirida): doença fatal causada por um vírus e caracterizada pela destruição das defesas do sistema imunológico.

Alelo: variação intra-espécie num loco gênico em particular.

Alérgeno: antígeno que causa reações alérgicas ou de hipersensibilidade.

Alergia: reação sintomática face a um antígeno ambiental normalmente inócuo.

Amígdalas: são grandes agregados de células linfóides organizadas como parte do sistema imune associado a mucosas e ao intestino.

Anafilaxia: reação imune específica que resulta na vasodilatação e contração de músculos lisos, como os dos brônquios, podendo resultar na morte do animal.

Anergia: estado de não-responsividade ao antígeno.

Anticorpo: molécula solúvel de proteína plasmática produzida e secretada pelas células B em resposta a um antígeno. Ligando-se especificamente a estes agentes infectantes, ou antígenos, os anticorpos podem neutralizá-los ou prepará-los para o processamento (ingestão e digestão) pelos fagócitos.

Anticorpos monoclonais: anticorpos produzidos por um único clone de células B.

Antígeno: qualquer substância que, quando introduzida no organismo, é reconhecida pelo sistema imune adaptativo. Seu nome deriva de sua propriedade de gerar anticorpos.

Apêndice: tecido linfóide associado ao intestino, localizado na porção inicial do cólon.

Apoptose: ou morte celular programada. É caracterizada pela degeneração do DNA nuclear, degeneração e condensação nuclear pela fagocitose dos resíduos celulares.

Auto-anticorpo: anticorpos específicos para antígenos próprios ao hospedeiro.

Auto-imunidade: resposta imune contra antígenos próprios.

B

Baço: órgão linfóide secundário da cavidade abdominal que é importante para as atividades do sistema imunológico. Possui uma polpa vermelha envolvida na remoção de células sanguíneas envelhecidas e uma polpa branca de células linfóides que responde aos antígenos levados ao baço pelo sangue.

Bactéria: microorganismo procariótico.

Basófilo: leucócitos que contêm grânulos.

C

Cadeia leve (L): é a menor das duas cadeias que constituem todas as moléculas de imunoglobulinas, sendo formada por um domínio variável (V) e um domínio constante (C), com ligação dissulfídrica com a cadeia pesada.

Cadeia pesada (H): todas as moléculas de imunoglobulinas possuem dois tipos de cadeia, uma pesada (H) e uma leve (L). a unidade básica de uma imunoglobulina possui duas cadeias pesadas idênticas e duas cadeias leves também idênticas.

Câncer: designação genérica de qualquer tumor maligno.

Célula apresentadora de antígeno (APC – antigen presenting cell): célula altamente específica que podem processar os antígenos e exibir seus fragmentos peptídicos na superfície celular, juntamente com moléculas necessárias para a ativação linfocitária. Linfócitos B, monócitos (incluindo macrófagos e células dendríticas) e vários outros tipos de células que apresentam o antígeno sob uma forma reconhecível pelos linfócitos T.

Células B: pequenas células brancas cruciais para as defesas imunes. Também conhecidas por linfócitos B, são produzidas na medula óssea e diferenciam-se em plasmócitos, geradoras dos anticorpos, ou células de memória, capazes de responder rápida e efetivamente contra re-infecções.

Células dendríticas: células brancas encontradas no baço e outros órgãos linfóides, cuja principal função é apresentar antígenos a células T.

Células matadoras naturais (NK – natural killer cell): são linfócitos não-T e não-B, portadores de morfologia granular, que matam certas células tumorais. São importantes na imunidade inata frente aos vírus e outros agentes intracelulares.

Células T: subgrupo de linfócitos definidos pelo seu desenvolvimento no timo. Participam diretamente nas respostas imunes adaptativas e secretam linfocinas quando ativadas.

Células T auxiliares: subclasse das células T que tipicamente carregam os marcadores CD4 e são fundamentais para o início da produção de anticorpos, ativando células T citotóxicas, e iniciando muitas outras respostas imunes.

Células T citotóxicas: subconjunto dos linfócitos T CD8 que podem matar células infectadas por vírus ou transformadas por câncer.

Células T supressoras: subconjunto de células T que interrompe a produção de anticorpos e outras respostas imunes.

Centro germinativo: locais em tecidos linfóides secundários onde ocorre intensa proliferação, seleção, maturação e morte de células B durante respostas de formação de anticorpos.

Choque anafilático: reação alérgica a antígenos administrados por via sistêmica, capaz de causar colapso circulatório e asfixia por edema de vias aéreas superiores.

Citocinas: termo genérico para moléculas solúveis mediadoras da interação celular. São proteínas sintetizadas e secretadas por células que afetam o comportamento de outras células. Incluem as linfocinas ou interleucinas, secretadas pelos linfócitos, e monocinas produzidas pelos monócitos e macrófagos.

Clone: grupo de células derivadas de um progenitor comum.

Complemento: grupo de proteínas séricas envolvidas no controle da inflamação, na ativação de fagócitos e no ataque lítico às membranas celulares. O sistema complemento pode ser ativado por interação com o sistema imune.

Complexo de histocompatibilidade principal (MHC): constitui um conjunto de genes que codificam as moléculas MHC e proteínas envolvidas no processamento de antígenos e na defesa do hospedeiro.

Cromossomo: estrutura física do núcleo celular que hospeda os genes. Cada ser humano possui 23 pares de cromossomos.

Crossover: processo de recombinação de material genético que ocorre na reprodução sexuada.

D

Deleção clonal: é responsável pela tolerância ao próprio através da eliminação dos linfócitos imaturos que se ligam aos antígenos próprios.

Determinante antigênico: é a porção da molécula de um antígeno que reage com um determinado anticorpo; também é conhecido com epítopo.

Doença auto-imune: doenças nas quais a patologia é causada por uma resposta imune a antígenos próprios.

E

Edição de receptor: substituição de uma cadeia leve de um receptor de antígeno auto-reativo em células B imaturas por uma cadeia leve que não confere auto-reatividade.

Eosinófilos: são leucócitos admitidos como importantes especialmente na defesa contra infecções parasitárias; são ativados pelos linfócitos da resposta imune adaptativa.

Epítopo: um único determinante antigênico. Funcionalmente é a porção do antígeno que liga-se ao paratopo do antígeno.

Exon: segmento de genes codificando uma proteína.

Expansão clonal: proliferação de linfócitos antígeno específicos em resposta ao estímulo antigênico e precede sua diferenciação em células efetoras. Constitui uma etapa essencial

da resposta imune adaptativa, permitindo que as células antígeno-específicas aumentem em número, de modo a poder combater efetivamente o patógeno que iniciou a resposta.

F

Fagocitose: internalização de matéria particulada por células.

Fagócitos: células que fagocitam. Contribuem para a defesa imune ingerindo e digerindo micróbios ou outras células e partículas invasoras.

Fenótipo: características físicas e mentais de um organismo conseqüentes do genótipo.

Fragmento Fab: consiste da cadeia leve e da metade aminoterminal da cadeia pesada, mantidas juntas por uma ponte de enxofre.

Fragmento Fc: compreende as metades carboxiterminais das duas cadeias pesadas, unidas por pontes de enxofre na região móvel da dobradiça.

Fungo: microorganismos eucarióticos unicelulares, como levedura e bolores, que podem causar diversas doenças. A imunidade aos fungos é complexa e envolve tanto as respostas imunes humorais quanto as celulares.

G

Gameta: cromossomo único.

Gene: unidade de material genético (DNA) que carrega as direções que uma célula usa para desenvolver determinadas tarefas.

Genoma: conjunto de todo o material genético contido dentro da célula.

Genótipo: conjunto particular de genes contidos em um genoma.

Granulócitos: denominação alternativa para leucócito polimorfonucleares. Células brancas contendo grânulos com químicos que permitem a célula digerir microorganismos, ou produzir reações inflamatórias. Os neutrófilos, eosinófilos e basófilos são exemplos de granulócitos.

H

Haptenos: são as moléculas que podem reagir com o anticorpo, mas não são capazes de, por si só, induzir uma resposta imune adaptativa.

Hipermutação somática: durante as respostas das células B aos antígenos, os genes da região-V sofrem mutação somática com altas taxas, visando a geração de vários anticorpos, alguns dos quais se ligam com maior afinidade ao estímulo antigênico.

Histamina: amina vasoativa liberada a partir de mastócitos e basófilos.

HIV (vírus da imunodeficiência humana): vírus causador da AIDS.

Homeostase: termo genérico para descrever o estado fisiológico normal.

I

Idiotipo: característica antigênica da região-V do anticorpo. É constituído de uma série de idiotopos, que podem estimular respostas reguladoras de uma rede idiotípica.

Idiotopo: um único antígeno na região-V do anticorpo.

Ignorância clonal: descreve uma forma de autotolerância.

Imagem interna: quando um anticorpo antígeno específico é utilizado para imunizar um indivíduo, alguns dos anticorpos estimulados se assemelham ao antígeno original e são referidos como uma imagem interna.

Imunidade celular: imunidade produzida principalmente pela ação das células T.

Imunidade humoral: imunidade mediada por fatores solúveis, como os anticorpos que circulam no fluído do corpo.

Imunidade inata: as fases iniciais da resposta do hospedeiro contra a infecção dependem da imunidade inata, na qual uma variedade de mecanismos de resistência inata reconhece e responde à presença de agentes patogênicos. Está presente em todas as pessoas em qualquer tempo e não aumenta com a repetida exposição a um dado microorganismo.

Imunidade passiva: imunidade resultante da transferência de anticorpos, ou soro, imune.

Imunoglobulinas: famílias de moléculas de grandes proteínas, também conhecidas por anticorpos.

Imunologia: estudo de todos os aspectos da defesa do hospedeiro contra infecção e das conseqüências das respostas imunes.

Inflamação: termo usado para descrever o acúmulo local de fluído, de proteínas plasmáticas e de células brancas, iniciado pelo dano físico, pela infecção ou por uma resposta imune local.

Interferon: citocinas que podem levar as células a resistirem face à replicação de vírus.

Interleucinas: termo usado para citocinas elaboradas por leucócitos e envolvidas na sinalização entre as células do sistema imunológico.

Íntron: segmento gênico entre os exons que não codificam proteína.

L

Latência: alguns vírus podem penetrar em umas células, mas não se replicam, constituindo um estado conhecido como latência.

Leucócitos: todas as células brancas, incluindo os linfócitos, leucócitos polimorfonucleares e monócitos.

Linha: fluído transparente, levemente amarelado que transporta os linfócitos, lubrifica os tecidos do corpo e deságua nos linfonodos.

Linfocinas: citocinas produzidas pelos linfócitos que atuam na regulação das respostas imunes.

Linfócitos: pequenas células brancas produzidas pelos órgãos linfóides e responsáveis pela resposta imune.

Linfócitos virgens: linfócitos que nunca encontraram seus antígenos específicos e, assim, nunca responderam ao mesmo, distinguindo-se do linfócitos de memória e efetores.

Linfonodos: órgãos linfóides secundários do sistema imunológico, com forma de grãos de feijão distribuídos pelo organismo e ligados pelos vasos linfáticos, aonde se iniciam as respostas imunes adaptativas.

M

Macrófago: grandes células fagocitárias que atuam devorando micróbios, apresentando antígenos a outras células e expelindo secreções imunes.

Mastócito: célula fagocitária que contém grânulos preenchidos com químicos capazes de danificar e destruir invasores.

Maturação de afinidade: aumento da afinidade média dos anticorpos produzido durante o curso de uma resposta imune humoral.

Medula óssea: tecido esponjoso encontrado nas cavidades dos ossos. É o local de geração dos elementos celulares do sangue, incluindo as hemácias, os monócitos os leucócitos polimorfonucleares e as plaquetas. Nos mamíferos corresponde, também, no local de desenvolvimento das células B.

Memória imune: quando um antígeno é encontrado mais de uma vez, a resposta imune adaptativa a cada encontro sucessivo é mais rápida e eficaz, característica conhecida como memória imune.

MHC: ver complexo de histocompatibilidade principal.

Micróbios: organismos vivos inferiores, incluindo bactérias, vírus, fungos e protozoários.

Microorganismos: organismos microscópicos, unicelulares com exceção de alguns fungos, que incluem bactérias, leveduras e protozoários, podendo causar doenças humanas.

Microorganismos patogênicos: microorganismos que podem causar doenças quando infectarem um hospedeiro.

Molécula: menor quantidade de substâncias químicas específicas que podem existir isoladamente.

Monocinas: citocinas secretadas por macrófagos.

Monócitos: grandes fagócitos (células brancas) que ao entrar no tecido transformam-se em macrófagos.

Monoespecificidade: os linfócitos possuem apenas um tipo de receptor sendo, portanto, monoespecíficos em relação aos antígenos.

Morte celular programada: ver apoptose.

Mutação somática: ver hipermutação somática.

N

Neutralização: anticorpos que podem inibir a infectividade de um vírus ou a toxicidade de uma molécula de toxina são ditos neutralizantes, e o processo de inativação recebe o nome de neutralização.

Neutrófilo: também conhecidos como leucócitos polimorfonucleares, constituem a principal categoria de células brancas no sangue periférico humano.

O

Opsonização: alteração da superfície de uma agente patogênico ou de outra partícula, permitindo que ele possa ser ingerido por fagócitos.

Organismo: ser vivo individual.

Órgãos linfóides: órgãos que compõem o sistema imunológico, onde os linfócitos se desenvolvem. Incluem a medula óssea, o timo, os linfonodos, o baço e vários outros grupos de tecidos linfóides. Os vasos linfáticos também podem ser considerados órgãos linfóides.

Órgãos linfóides primários (ou *centrais*): locais de desenvolvimento dos linfócitos.

Órgãos linfóides secundários (ou *periféricos*): são os linfonodos, o baço e os tecidos linfóides associados a mucosas, onde são induzidas as respostas imunes.

P

Parasita: planta ou animal que vive, cresce e se alimenta sobre ou dentro de outro ser.

Paratopo: marcador contido na superfície da molécula de anticorpo e que se liga ao antígeno correspondente.

Patógenos: microorganismos infecciosos externos.

Patologia: é o estudo científico da doença. O termo também é empregado para descrever lesões tissulares detectáveis.

Peptídeo: marcador antigênico reconhecido pela região-V de um anticorpo.

Placas de Peyer: coleção de tecidos linfóides no intestino.

Plaquetas: pequenos fragmentos celulares encontrados no sangue e decisivos para o processo de coagulação sanguínea.

Plasma: é o componente fluido do sangue contendo água, eletrólitos e proteínas plasmáticas.

Plasmócitos: células grandes produtoras de anticorpos diferenciadas a partir de células B.

Proteínas: compostos orgânicos feitos de amino-ácidos. São os maiores componentes das plantas e animais.

Proteínas de fase aguda: uma série de proteínas encontradas no sangue pouco depois do início de uma infecção. Participam nas etapas iniciais de defesa.

Q

Quimiocina: citocinas envolvidas na migração e ativação de células, principalmente, fagocitárias e linfocitárias.

Quimiotaxia: aumento da migração direcional de células, especialmente em respostas a gradientes crescentes de concentração de certos fatores quimiotáticos.

R

Reatividade cruzada: reação do anticorpo à um antígeno diferente, porém relacionado, ao utilizado para evocar uma resposta imune humoral.

Rearranjo: a expressão de receptores antigênicos requer o rearranjo de segmentos gênicos no desenvolvimento dos linfócitos. As seqüências expressas da região variável são compostas pelo rearranjo dos segmentos gênicos.

Receptor de antígenos: tanto os linfócitos B quanto os T carregam em suas superfícies receptores de antígenos muito distintos, capazes de reconhecer uma grande variedade de antígenos.

Receptor de célula B: receptor de superfície das células B específicos para determinado antígeno.

Receptor de célula T: consiste de um heterodímero de cadeias α e β ligadas por pontes dissulfídicas altamente variáveis, expressas na membrana celular. Um receptor alternativo feito de cadeias variáveis γ e δ é expresso em um subconjunto de células T.

Rede idiotípica: os receptores linfocitário podem se reconhecer mutuamente através de interações idiotípicas, formando uma rede idiotípica.

Região constante: parte da estrutura do anticorpo característica a cada classe de anticorpos.

Região variável: parte da estrutura do anticorpo que difere de um anticorpo para o outro.

Resposta auto-imune: é a resposta imune adaptativa contra substâncias próprias do hospedeiro.

Resposta imune: reações do sistema imunológico a fim de se defender contra um agente patogênico.

Resposta inflamatória: vermelhidão, queimação, dor, inchaço e perda de função produzidos em resposta a uma infecção, como resultado do aumento de fluxo e secreções das células imunes.

S

Segmentos gênicos: os domínios variáveis dos receptores imunes são codificados em segmentos gênicos que devem, em primeiro lugar, sofrer recombinação somática a fim de formarem um exon complexo, codificando a região-V.

Segundo sinal: sinal co-estimulatório requerido pela ativação linfocitária.

Seleção negativa: durante o desenvolvimento intratímico, os timócitos que reconhecem o próprio são deletados do repertório.

Seleção positiva: apenas as células T cujos receptores podem reconhecer os antígenos apresentados pelas moléculas MHC próprias podem amadurecer no timo.

Sinal co-estimulatório: a proliferação dos linfócitos requer tanto a união com o antígeno como a recepção de um sinal co-estimulatório, usualmente emitido por uma molécula de superfície de uma APC.

Sistema complemento: ver complemento.

Soro: líquido claro que separa do sangue quando este coagula. Retém os anticorpos presentes no sangue.

T

Teoria da seleção clonal: é o paradigma central da imunidade adaptativa. Estabelece que as respostas imunes adaptativas derivam de linfócitos individuais antígeno-específicos que são autotolerantes. Estes linfócitos proliferam em resposta ao antígeno e se diferenciam em células efetoras antígeno-específicas, a fim de eliminarem o agente patogênico, e em células de memória, destinadas à manutenção da imunidade duradoura.

Timo: órgão linfóide primário, localizado um pouco acima do peito, onde os linfócitos T se proliferam e amadurecem.

Timócitos: células linfóides encontradas no timo.

Tolerância: estado passivo de resposta a um determinado antígeno ou grupo de antígenos.

Toxinas: agentes produzidos por plantas e bactérias, normalmente perigosos para os seres humanos.

V

Vacina: substância que contém componentes antigênicos de um organismo infeccioso para estimular a resposta imune.

Vasos linfáticos: rede de canais espalhada pelo organismo, semelhante aos vasos sanguíneos, que transportam a linfa para os órgãos imunes e para a corrente sanguínea.

Vírus: micróbio submicroscópico causador de doenças infecciosas. Se reproduzem apenas em células vivas.

ÍNDICE REMISSIVO

ASSUNTOS E AUTORES

A

Aamodt, A., 79
Abbas, A. K., 11, 32
Abbattista, F., 50
ABNET, 3, 127, 150, 203
abordagem
 baseada em agentes, 51
 conexionista, 157
ácido nucléico, 123
Ada, G. L., 25, 105
Adams, D., 28, 121
adaptabilidade, 110
adaptação
 filogenética, 120
 ontogenética, 121
 sociogenética, 120
adenóides, 15
Ader, R., 99
Adleman, L. M. , 41, 122
afinidade, 25, 54, 59
 de Hunt, 59
agentes
 autônomos, 47
 inteligentes, 47
Ahmed, R., 12, 27
aiNet, 3, 127, 159, 214
Aisu, H., 50
alelo, 110
alérgeno, 11
álgebra Booleana, 124, 158
algoritmo
 de seleção clonal, 48, 64, 67, 127, 141, 150
 de seleção negativa, 47, 48, 64, 65
 de treinamento, 86
 evolutivo básico, 111
 evolutivo, 147
 genético, 46, 78, 108, 140
 genético clássico, 113
 genético co-evolutivo, 51
Allen, D., 27, 28, 30
amígdala, 15
aminoácido, 10

análise de um algoritmo, 129
Anderson, G., 33
anergia, 28, 31
anticorpo, 8, 20
antígeno, 9
 próprio, 10, 25, 31
APC, 13
apêndice, 15
apoptose, 28
aprendizagem, 7, 84, 120
 auto-organizada, 89
 competitiva, 151
 de máquina, 51, 141
 imunológica, 25, 27
 não-supervisionada, 89, 151
 por reforço, 2, 25, 49, 91
 supervisionada, 88
aptidão, 110
Aristóteles, 237
arquitetura, 86
árvore, 168
 geradora mínima, 168, 170
aspectos cognitivos, 150
Atmar, W., 109, 111
atratores, 98
atualização de pesos, 155
audição, 104
autômato de estado finito, 52
axioma da completude, 31
Axtell, R., 52

B

Bäck, T., 108, 117, 120, 148, 241
baço, 15
bactéria, 8
Ballet, P., 52
Banchereau, J., 19, 28, 31, 34
barreira cérebro-sangue, 105
Barreiros, J. A. L., 197
Barto, A. G., 25, 91
base nucleotídica, 22
basófilo, 12

- BCR, 18
 Bell, G. I., 10
 Bentley, P., 47
 Berek, C., 27, 30
 Bernardes, A. T., 35
 Bernaschi, M., 52
 Bersini, H., 37, 49
 Bezdek, J. C., 173
 Bezzi, M., 52
 Biswas, N. N., 158
 Blalock, E. J., 100, 103
 Boers, E. G. W., 197
 Bonabeau, E. W., 40
 Bonamin, L. V., 99
 Bonna, C. A., 35
 Bordet, J., 9
 Breinl, 10
 Broomhead, D. S., 91, 92
 Burnet, F. M., 10, 25, 28
 busca genética restrita, 48
- C**
- cadeia
 de atributos, 56, 128
 leve, 20
 pesada, 20
 polipeptídica, 20
 caixeiro viajante, 57
 Calenbuhr, V., 35
 camada de anticorpo, 48
 caminho, 168
 capacidade de generalização, 26, 46, 200
 características dos algoritmos genéticos, 113
 Carroll, J. D., 170
 Case-Based Reasoning, 46
 CDR, 20
 Celada, F., 52, 56, 61
 célula
 apresentadora de antígeno, 12, 13, 19
 auto-reativa, 10
 B, 13
 branca, 12
 de memória, 14, 23, 25, 27
 dendrítica, 19, 33
 epitelial, 33
 imunocompotente, 31
 T auxiliar, 18, 19
 T citotóxica, 18, 19
 T, 13
 centro de massa, 172
 centróide, 167, 168, 172
 Chen, S., 92, 227
 Cho, S. -B., 95, 158, 235
 Choate, T. D., 92
 Çilingiroglu, U., 158
 circuito, 168
 lógico, 158
 clonagem, 22, 110
 CLONALG, 3, 67, 127, 141, 159, 180
 clonar, 10
 clone, 18, 141
 cluster, 90
 clusterização
 gradual, 172
 hierárquica, 53
 nebulosa, 173
 clusters, 95
 cobertura, 61
 co-estimulação, 33
 cognição, 140
 Cohen, I. R., 101, 239
 Cohen, J. J., 28
 Cohn, M., 35
 coleta de lixo, 48
 Collard, P., 49
 colônias de insetos, 39
 compartilhamento de fitness, 120
 competição, 39
 complementaridade, 18
 complemento, 9
 complexidade de um algoritmo, 129
 complexo de histocompatibilidade principal,
 13, 18
 complexo MHC/peptídeo, 13, 18, 19, 33
 comportamento coletivo emergente, 48
 comprimento de onda, 46
 computação
 de DNA, 41
 evolutiva, 17, 41, 108
 imunológica, 44
 molecular, 122
 conexões, 85
 conjunto nebuloso, 124
 construção da rede, 153
 contexto, 102
 controle autônomo, 49
 Cooke, D. E., 43, 46, 175

cooperação, 39
 coordenação de servidores Web, 52
 Coutinho, A., 27, 28, 30, 31, 35, 37, 72, 102
 cromossomo, 110
 crossover, 110, 140
 simples, 115
 uniforme, 116
 cross-reactive response, 26
 custo computacional, 129, 141, 149, 159
 Cziko, G., 9, 121

D

D'haeseleer, P., 47, 67
 dado rotulado, 89
 Darwin, C., 2, 109
 Dasgupta, D., 43, 46, 47, 49, 50, 59, 104
 Dawkins, R., 122
 de Boer, R. J., 30, 35, 56
 de Castro, L. N., , 45, 48, 50, 53, 57, 59, 61, 67, 88, 95, 132, 141, 150, 154, 158, 159, 197, 205, 224, 235
 de Deus Jr, G. A., 205
 De Jong, K. A., 51, 114
 Deaton, R. J., 123
 Deaton, R., 51
 decisão de lógica majoritária, 203
 definição
 algoritmos genéticos, 113
 aprendizagem, 88
 árvore geradora mínima, 170
 busca em um espaço de buscas, 111
 caminho minimax, 170
 clusterização, 160
 dendrograma, 168
 engenharia imunológica, 44
 epigênese, 112
 evolução, 109
 homúnculo imunológico, 102
 imunologia, 7
 método de niching, 118
 programação genética, 117
 rede imunológica artificial, 160
 rede neural artificial, 84
 sistema complexo, 40
 sistema imunológico artificial, 43
 sistema imunológico, 7
 superfície de adaptação, 111
 teoria da rede, 35

deleção clonal, 33
 dendrograma, 167
 Denys, J., 9
 detecção
 detecção
 de anomalias, 2, 47, 50
 de falhas, 50
 detector, 75
 de característica, 90
 Detours, V., 35, 56
 Devarajan, V., 51
 Dilger, W., 51
 dinâmica, 37, 160, 174
 diplóide, 110
 distância minimax, 170
 distinção próprio/não-próprio, 28, 35, 47
 diversidade, 2, 28, 31, 46, 118, 127, 133
 DNA, 21, 51, 110
 doença
 auto-imune, 31, 34, 47
 psicossomáticas, 99
 dos Santos, R. M. Z., 35
 Dreher, H., 18

E

edição de receptor, 22, 27, 28
 efeito Baldwin, 51
 Ehrlich, P., 9
 Einarson, G., 204
 Elanayar, S., 92
 eletroforese de gel, 123
 Endo, S., 48, 57
 energia, 134
 enfermidade, 8
 engenharia, 39
 imunológica, 2, 127, 128
 enzima, 12, 123
 eosinófilo, 12
 epítipo, 18, 35
 Epstein, J., 52
 equação diferencial, 45
 espaço
 de buscas, 140
 de estados, 98
 de formas, 54
 de formas de Hamming, 56
 de formas de Manhattan, 56
 de formas Euclidiano, 56

de formas Inteiro, 57
espécie, 118
esquemas, 46
estado espúrio, 98
estratégia
 evolutiva, 108
 evolutiva, 140, 148
estrutura, 37
 formal, 53
evento seletivo, 10
evolução natural, 109
evolução, 109
expansão clonal, 32

F

Fab, 20
fagócito, 9, 12
Fahlman, S. E., 197
Farmer, J. D., 25, 35, 45, 46, 50, 52, 56, 71, 176
Fc, 20
Fellows, A., 52, 79
fenótipo, 53, 65, 110
ferramentas computacionais, 127
Fischetti, M., 130
física quântica, 97
fitness, 110, 140
fitness sharing, 119
fluxograma, 134
Fogel, L. J., 108
Fontanari, J. F., 188
forma generalizada, 54
Forrest, S., 43, 46, 47, 48, 50, 52, 53, 56, 59, 61, 65, 75, 123, 140, 240
Freeman, J. A. S., 92
Fritzke, B., 95, 158, 229, 235
Fukuda, T., 48
função
 de ativação, 61, 85
 de compartilhamento de fitness, 120
 de custo, 113
 de energia, 133
 de energia, 98
fungo, 8
furthest neighbor, 168
fuzzy c-means, 173
fuzzy k-means, 173

G

gameta, 110
Garzon, M. H., 123
Gaspar, A., 49
gene, 110
genoma, 110
genótipo, 53, 65, 110
George, A. J. T., 27, 29
geração, 144
Germain, R. N., 19
Giroso, F., 92
Goldberg, D. E., 118, 120, 182, 241
Gomide, F. A. C., 40, 125
Goodman, D. J., 204
grafo, 53
 conexo, 168
 G ponderado, 168
grânulo, 12
granulócito, 12
grau de ligação, 56
Gray, D. L., 27, 29, 158
Grossberg, S., 240
Gu, J., 48

H

Hajela, P., 48, 59, 139, 162
Halgamuge, S. K., 230
haplóide, 110
haptenos artificiais, 9
Harmer, P. K., 47, 59, 139
Hart, E., 52, 57, 77, 162
Hartigan, J. A., 168
Haurowitz, 10
Haykin, S., 26, 40, 84, 92, 133
hemácia, 15
hematopoese, 15
Hightower, R. R., 51, 53, 56, 59, 61, 65, 241
hill climbing, 113
hipermutação somática, 21, 27, 68
histograma, 97, 170
Hodgkin, P. D., 26, 31
Hoffmann, G. W., 26, 35, 49, 104
Hofmeyr, S. A., 16, 43, 48, 52, 59, 75
Holland, J. H., 41, 45, 71, 108, 122
Hopfield, J. J., 50, 97, 140
Horowitz, E., 129
Hubert, L., 168

Hunt, J. E., 43, 46, 52, 59, 79, 175

I

identidade molecular, 38
 idioto, 10, 31, 35
 IEEE Transactions on Evolutionary
 Computation, 41
 Jenne, P., 56
 ignorância clonal, 32
 imagem interna, 36, 161
 imunidade, 7
 imunização, 8
 imunoglobulina, 20
 imunoide, 80
 inconsistência, 170
 infecção crônica, 27
 informações sensoriais, 85
 inicialização de pesos, 50
 Inman, J. K., 31
 inteligência
 artificial, 40
 computacional, 3
 intencionalidade, 103
 intervalo de amadurecimento, 153
 Introdução, 179
 inversão, 140
 Ishida, Y., 47, 51, 53, 104
 Ishiguro, A., 48, 80

J

janela de evolução, 148
 Janeway, C. A., 8, 11, 12, 16, 20, 27
 Java, 52
 Jayasuriya, A., 230
 Jefferson, D., 41
 Jenner, E., 8
 Jerne, N. K., 10, 30, 31, 35, 49, 70, 101, 104,
 159, 179
 Johnsonbaugh, R., 129
 Jun, H. -B., 125
 Jun, J. H., 48

K

Karayiannis, N. B., 92, 230
 Kaski, S., 96

Kelly, J., 33
 Kelly, R. H., 49, 104
 Kephart, J. O., 47
 Kepler, T. B., 30, 157
 Kim, J., 47, 197
 Kinnear Jr., K. E., 117
 Kirkpatrick, S., 3, 53, 132
 Kitasato, S., 8
 Klein, J., 17
 Kleinstein, S., 52
 Kohler, H., 35
 Kohonen, T., 91, 95, 158, 234
 Koza, J. R., 117
 Krishnakumar, K., 49, 125
 Kruisbeek, A. M., 31
 Kuhn, G., 56
 Kuiper, H., 197
 Kumar, R., 158
 Kwok, T. Y., 88, 200

L

Lamont, G. B., 47, 59, 139
 Landsteiner, K., 9
 Langman, R. E., 35
 Langton, C. G., 41
 Lapointe, F. -J., 168
 latência, 26
 Leclerc, B., 170
 Lederberg, J., 17
 Lee, D. -W., 48, 125
 Lee, J., 48, 59, 139
 Legendre, P., 168
 Lehtokangas, M., 197
 leucócito, 12
 ligação, 54
 ligante, 54
 limiar
 de afinidade, 55, 57
 de compartilhamento, 120
 linfa, 15
 linfocina, 13
 linfócito virgem, 17
 linfócito, 12, 13, 14
 linfonodo, 15
 Lippman, R. P., 158, 181
 lisozima, 16
 locus, 110
 lógica, 124

binária, 124
 multi-valorada, 124
 nebulosa, 124
 Lowe, D., 91, 92

M

macrófago, 12, 16, 19, 33
 Mahfoud, S. W., 118, 140, 241
 Manni, M. D., 32
 mapas auto-organizáveis de Kohonen, 153
 Martello, S., 130
 Mason, D., 26, 31
 match score, 56
 matching, 54
 matriz-U, 96
 Matsui, K., 140
 maturação de afinidade, 27, 155
 Mazutani, H., 50
 McConkey, D. J., 28
 McCoy, D. F., 51
 McCulloch, W., 85
 McDonald, B. A., 207
 mecanismo
 de seleção, 116
 sensorial, 103
 Medawar, P., 10
 medula óssea, 15, 64
 memória, 7
 associativa, 50
 endereçável por conteúdo, 97
 imunológica, 2, 27, 28, 103, 142
 metadinâmica, 37, 49, 160, 174
 meta-evolução, 148
 Metchnikoff, E., 9
 método de gradiente, 113
 MHC, 13
 MHC-I, 19
 MHC-II, 19
 Mi, G. W., 92, 230
 Micchelli, C. A., 93, 236
 Michel, A. N., 158
 microevolução, 142
 microorganismos patogênicos, 8
 Milligan, G. W., 169
 mineração de dados, 79
 minimal spanning tree, 168
 Mitchel, M., 110, 113, 140
 modelo

computacional, 52
 contínuo da rede imunológica, 64
 de medula óssea, 64
 molécula
 receptora de antígeno, 18
 de DNA, 122
 Moller, M. F., 199
 morte celular programada, 28
 Moscato, P., 188
 multi-especificidade, 31
 mutação, 110, 113
 indutiva, 116
 somática, 10, 142

N

Nagano, S., 51, 123
 não-próprio, 31
 nearest neighbor, 167
 Neidhoefer, J., 49
 neodarwinismo, 109
 neurônio, 151
 artificial, 85
 biológico, 84
 neutrófilo, 12
 Nguyen, D., 197
 niching, 48
 nicho ecológico, 118
 Nikolaev, N. I., 52
 nível
 de pertinência, 172
 de concentração, 153
 de estimulação, 125
 Norvig, P., 40
 Nossal, G. J. V., 13, 25, 33, 105
 notação, 3, 134, 142, 151, 162
 nucleotídeo, 110, 123
 Nussenzweig, M. C., 27

O

Oei, C. K., 184
 Okamoto, T., 47
 olfato, 104
 Otsuki, J. T., 49
 operador
 de mutação, 130
 de seleção, 130

de recombinação, 115
 Oprea, M., 21, 53, 56, 139, 162
 órgãos
 linfóides, 14
 linfóides primários, 33
 linfóides secundários, 33
 origem das espécies, 109
 Orr, M. J. L., 227
 Oster, G. F., 31
 otimização, 141, 145
 de segundo tipo, 148
 dependente do tempo, 49
 multimodal, 48, 145
 ótimo local, 29

P

padrão de reflexão, 57
 padrões de treinamento, 151
 Pal, N. R., 173
 paladar, 104
 paradigma funcional, 117
 paradoxo, 31
 parasita, 8
 paratopo, 35
 Parijs, L. V., 32
 partição, 173
 Pasteur, L., 8
 patógeno, 8
 patologia, 8
 Pauling, L., 10
 Pearlmutter, B. A., 199
 Pedrycz, W., 40, 125
 peptídeo, 13, 19
 Perelson, A. S., 10, 20, 24, 25, 30, 31, 35, 46,
 53, 54, 56, 59, 61, 65, 121, 157
 personalidade, 103
 pesos sinápticos, 84
 Piattelli-Palmarini, M., 9
 Pitts, W., 85
 placas de Peyer, 15
 plasmócito, 14, 23
 Plaza, E., 79
 poda da rede, 154
 Poggio, T., 92
 Pohlmann, P. R., 13
 polpa
 branca, 15
 vermelha, 15

portas lógicas, 158
 Potter, M. A., 51
 Powell, M. J. D., 92
 Prechelt, L., 200
 prédios inteligentes, 51
 presa-predador, 25
 pressão seletiva, 119
 Prim, R. C., 170
 princípio
 da seleção clonal, 23, 121
 da seleção natural, 109
 problema
 do caixeiro viajante, 122
 do caminho Hamiltoniano, 122
 NP-completo, 52
 programação
 genética, 52, 117
 imunológica, 241
 projetar, 39
 proliferação celular, 22
 próprio, 31
 protocolo molecular, 123
 pseudocódigo, 131, 135, 144, 146, 157, 165
 psiconeuroimunologia, 99

Q

Q-learning, 49
 quebra pelo operador, 119

R

Ra, J. B., 197
 raciocínio baseado em casos, 46, 79
 RBF, 91
 r-bits complementares, 59
 reação em cadeia de polimerase, 123
 reatividade cruzada, 31
 reatividade cruzada, 55
 receptor, 9
 auto-reativo, 28
 de antígeno, 13
 Rechenberg, I., 108, 148
 recombinação genética, 110
 recombinação V(D)J, 22
 reconhecimento, 54
 de espectros, 46
 de padrões, 2, 18, 46, 128, 141, 150

- rede
 de computador, 47
 de Hamming, 158
 de Petri, 49
 idiotípica, 36
 imunológica artificial, 159
 neural artificial Booleana, 158
 neural artificial, 26, 40, 46, 49
 neural auto-organizada, 150
 neural Booleana, 50, 150
 neural competitiva, 95
 neural de Hopfield, 97
 neural RBF, 50
 Reed, R., 200
 região
 constante, 20
 de reconhecimento, 55
 região-V, 20, 27, 35
 regra
 da projeção, 98
 de Hebb, 98
 regulação, 18, 24
 Rensberger, B., 16
 repertório
 ativo, 31
 celular, 30
 disponível, 31
 potencial, 31
 pré-imune, 35
 representação, 112, 128
 de incertezas, 124
 pictórica, 62, 129
 reprodução
 assexuada, 110
 sexuada, 110
 resposta
 imune adaptativa, 11, 12, 23, 34, 141
 imune específica, 12
 negativa, 36
 positiva, 36
 primária, 24, 26, 27
 reativa cruzada, 26, 55
 secundária, 26, 27
 Richardson, J., 120, 182
 Richter, P. H., 35
 RNA, 21
 Robert, K., 8
 Robinson, M. E., 158
 robustez, 39
 Ross, P., 52, 57, 77, 162
 roulette wheel, 81, 114
 ruído de seleção, 119
 Rumelhart, D. E., 49
 Russel, S., 40
- S**
- Saad, D., 92
 Sahni, S., 129
 SAND, 127, 132, 191
 Sarle, W. S., 200
 scheduling, 57, 77
 Schmid, A., 158
 Schwartz, R. S., 28, 31, 34
 Schwefel, H. -P., 108, 148
 Scroferneker, M. L., 13
 Sedgewick, R., 130
 Segel, L. A., 35, 54, 56
 segmentação de imagens, 51
 segunda geração de redes imunológicas, 72
 segundo sinal, 23
 segurança computacional, 47, 74
 Seiden, P. E., 52, 56, 61
 Sekiguchi, T., 49
 seleção, 109, 112
 molecular, 28
 natural, 17, 120
 negativa, 10, 31, 32, 33, 47, 51
 tímica, 33, 47
 sexto sentido, 104
 Shin, Y. C., 92
 side-chain theory, 9
 Silverstein, A. M., 9
 simulated annealing, 3, 53, 127, 132, 133
 sinal co-estimulatório, 23
 sinalização, 18
 celular, 35
 sistema
 preventivo, 17
 biológico, 39
 classificador, 45, 46, 49, 52, 140
 complemento, 9, 14, 20
 computacional imunizado, 125
 distribuído, 39
 endócrino, 99
 imune adaptativo, 12
 imune inato, 12, 34
 imunológico artificial, 2, 41

imunológico, 39
 natural, 39
 nebuloso, 40, 124
 nervoso central, 39, 49, 99
 Slavov, V., 52
 Smith, D. J., 26, 46, 56, 105, 140
 special issue, 41
 special track, 41
 splicing, 21
 Sprent, J., 12, 26, 27
 Steinman, R. M., 19
 Stewart, J., 35
 Storb, U., 28, 30
 superfície de afinidade, 29
 supervisor, 88
 supressão clonal, 164
 supressão da rede, 164
 Sutton, R. S., 25
 Suzuki, J., 52, 53, 57

T

tamanho do passo, 169
 tato, 104
 Tauber, A. I., 101
 taxonomia, 44
 Taylor, C., 41
 TCR, 18
 técnicas de clusterização hierárquica, 160, 166
 temperatura, 134
 template, 10
 teoria

- da cadeia lateral, 9
- da evolução natural, 17
- da rede imunológica, 10, 27, 31, 35, 45, 70, 127
- da seleção clonal, 10, 18, 35
- de agentes, 51
- de grafos, 160
- germinal, 9
- instrucionista, 10
- seletivista, 9, 10

 terminologia biológica, 109
 Theraulaz, G., 40
 Timmis, J., 43, 52, 176
 timo, 15, 18
 timócitos, 34
 Tizard, I. R., 8, 22

tolerância, 10, 34, 35
 Toma, N., 48, 57
 tomada de decisões, 48, 51
 Tonegawa, S., 10, 27, 31
 transcrição, 21
 treinamento, 84

U

Ultsch, A., 96
 unicidade, 2, 84

V

vacinação, 8
 vacínia, 8
 valor de ligação, 60, 125
 valores de fusão, 169
 Varela, F. J., 31, 35, 37, 72, 102
 variáveis lingüísticas, 124
 vasos linfáticos, 16
 vermes, 47
 vertebrados, 17
 Vertosick, F. T., 49, 104
 vida artificial, 41
 vírus, 8, 47
 visão, 104
 vizinho mais distante, 168
 vizinho mais próximo, 167
 von Behring, E., 8
 von Boehmer, H., 33
 Von Zuben, F. J., 45, 48, 50, 53, 57, 59, 61, 67, 88, 95, 132, 141, 150, 154, 158, 159, 197, 205, 224, 235

W

Watanabe, Y., 48, 80
 Weigel, F. W., 30
 Weisbuch, G., 20, 25, 30, 31, 54
 Whitehead, B. A., 92
 Widrow, B., 197
 Wierzhón, S. T., 47, 67
 Witten, I. H., 207
 Wolfram, S., 39
 Wright, A., 9
 WWW Starlab, 43

Y

Yamamoto, Y., 52, 53, 57

Yeung, D. Y., 88, 200

Yonezawa, Y., 51, 123

Yoo, J., 48, 162

Z

Zadeh, L. A., 124

Zahn, C. T., 170

Ziegner, M., 27, 30

Zinkernagel, R. M., 33