

MARIA NAZARÉ MUNARI ANGELONI HAHNE

**METODOLOGIA DE APRENDIZADO BASEADO EM
ALGORITMOS GENÉTICOS PARA MODELOS
ALTERNATIVOS DE REDES IAC**

**FLORIANÓPOLIS
2005**

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**METODOLOGIA DE APRENDIZADO BASEADO EM
ALGORITMOS GENÉTICOS PARA MODELOS
ALTERNATIVOS DE REDES IAC**

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Doutor em Engenharia Elétrica.

MARIA NAZARÉ MUNARI ANGELONI HAHNE

Florianópolis, Setembro de 2005.

METODOLOGIA DE APRENDIZADO BASEADO EM ALGORITMOS GENÉTICOS PARA MODELOS ALTERNATIVOS DE REDES IAC

Maria Nazaré Munari Angeloni Hahne

‘Esta Tese foi julgada adequada para a obtenção do título de Doutor em Engenharia Elétrica, Área de Concentração em *Engenharia Biomédica*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’

Fernando Mendes de Azevedo, D.Sc.
Orientador

Alexandre Trofino Neto, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Fernando Mendes de Azevedo, D.Sc.
Presidente

Fernanda Isabel Marques Argoud, Dra.

Luiz Pereira Calôba, D.Sc.

Renato Garcia Ojeda, Dr.

Roberto C. L. de Oliveira, Dr.

Sílvia Modesto Nassar, Dra.

Aos meus pais . . .

AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer ao meu orientador, Fernando Mendes de Azevedo, pela orientação, pelo incentivo e pela amizade em todos estes anos de convívio. É um exemplo de profissional, que será sempre uma inspiração para a minha carreira.

Ao meu marido e minha filha, Rafael e Rafaela, que são meus dois grandes amores. Ao meu marido, pelo companheirismo e força, por cada palavra de carinho. À minha filha, por cada lindo sorriso. Ser mãe e acompanhar o seu crescimento enquanto desenvolvia este trabalho foi uma tarefa assustadora algumas vezes, mas, também, repleta de momentos maravilhosos.

Aos meus pais, Alfredo e Helena, que são meu exemplo de vida, meu porto seguro, que são as pessoas sempre dispostas a me ouvir, a qualquer hora. São as pessoas que sempre tem uma palavra de incentivo e que me fazem acreditar nos meus sonhos.

Aos meus irmãos, sobrinhos e cunhados, minha família querida, com quem eu posso contar incondicionalmente.

A Rudimar Luís Scaranto Dazzi, pelo cuidado com que leu meu trabalho e pelas valiosas contribuições.

A todos os colegas do Instituto de Engenharia Biomédica, colegas de mestrado e doutorado, pelo bate papo de corredor, pelas risadas, pelas conversas interessantíssimas sobre conspirações, sobre família, sobre relacionamentos, e, também, pelas dicas para o desenvolvimento do trabalho, para as apresentações dos seminários... As nossas conversas vão ficar para sempre na minha memória.

Às minhas 'irmãzinhas' de coração. Embora cada uma tenha tomado um caminho diferente, e a convivência não seja mais diária, como antes, sempre se fazem presentes por mails, pelo msn, pelo orkut, ou até numa boa lembrança dos tempos de faculdade. Espero que a nossa amizade dure para sempre, e que sempre possamos contar umas com as outras em todos os momentos, tanto alegres, quanto nos de angústia.

Ao CNPq, pelo suporte financeiro.

A **Deus**, pelo dom da vida.

METODOLOGIA DE APRENDIZADO BASEADO EM ALGORITMOS GENÉTICOS PARA MODELOS ALTERNATIVOS DE REDES IAC

Maria Nazaré Munari Angeloni Hahne

Setembro/2005

Orientador: Fernando Mendes de Azevedo, D. Sc.

Área de Concentração: Engenharia Biomédica

Palavras-chave: Redes Neurais Artificiais, Redes IAC, Algoritmos de Aprendizado, Algoritmos Genéticos

Número de Páginas: 151

Este trabalho apresenta uma metodologia para adaptar os pesos de uma rede modelo 'A', que é baseada na rede IAC original, proposta por Rumelhart. Originalmente, as redes IAC têm valores de pesos exclusivamente iguais a -1, 0 ou 1, não necessitando de um algoritmo de aprendizado. Esta característica limita sua aplicação prática. Os pesos de redes modelo 'A', propostas de forma a solucionar a restrição básica da rede IAC original, podem assumir qualquer valor pertencente ao intervalo [-1,1]. Sendo assim, são estudadas e propostas uma função de *fitness*, bem como uma representação cromossômica, que permitem a utilização de um algoritmo genético para encontrar um conjunto de pesos apropriado para a rede.

Os pontos principais envolvidos na elaboração da metodologia consistem da proposta de conjuntos de treinamento e formas de representação. O conjunto de treinamento pode ser composto por pares de entrada/saída, assim como pode ser composto por um conjunto de categorias para os pesos. Entretanto, pode-se também utilizar as categorias para os pesos na geração da população inicial, e utilizar os conjuntos de entrada/saída para o treinamento.

De forma a testar e validar o algoritmo proposto, foram utilizados três exemplos de problemas: o Jets & Sharks, um modelo simplificado de diagnóstico diferencial de Hepatite A e B, e um modelo simplificado para diagnóstico diferencial em Reumatologia. Os dois últimos exemplos não são possíveis de implementar usando a rede IAC original.

O modelo do Jets & Sharks teve seus pesos encontrados pelo Algoritmo Genético, e foram obtidas matrizes que forneceram resultados tão bons quanto aqueles fornecidos pela implementação utilizando a rede IAC original.

A metodologia proposta apresentou resultados consistentes também para os outros exemplos utilizados, sendo, portanto, considerada adequada para aplicação em outras classes de problemas que possam ser resolvidos com o modelo 'A' de redes IAC.

METHODOLOGY OF LEARNING BASED ON GENETIC ALGORITHMS TO ALTERNATIVE MODELS OF IAC NEURAL NETWORKS

Maria Nazaré Munari Angeloni Hahne

September/2005

Advisor: Fernando Mendes de Azevedo, D.Sc.

Area of Concentration: Biomedical Engineering

Keywords: Artificial Neural Networks, IAC Neural Networks, Learning Algorithms, Genetic Algorithms

Pages: 151

This work describes a methodology to derive the weights in a model 'A' network. This model is based on original IAC model, proposed by Rumelhart. The IAC networks have weights valued as -1, 0 or 1, therefore not needing a learning algorithm. This characteristic implies on little practical use. On the contrary, model 'A' networks, developed as an answer to the original IAC networks primary restriction, can assume any value within the interval $[-1,1]$. So, we present fitness functions, and a problem encoding as well, that allow the use of genetic algorithms to find a weight-set suitable to this network topology.

In this work, the principal points are to propose training sets and problem encodings. The training set can be composed by input/output pairs, as well weight categories. However, it is also possible to use the weight categories in the generation of the initial population, and to use the input/output pairs to training.

To evaluate and validate the proposed algorithm, we applied it to three examples: Jet & Sharks, a simplified model of differential diagnosis of Hepatitis A and B and a simplified model of differential diagnosis in Reumatology. Both diagnosis of Hepatitis and diagnosis in Reumatology can not be implemented using the original model.

We applied the methodology to Jets & Sharks problems, e we get weight matrices whose results are so good as those supplied by the implementation using original net IAC.

Results are consistent for all examples, showing that our algorithm can be considered a fitting and unrestricted methodology for application in any other problems solvable with the model 'A'.

Sumário

1	Introdução	1
1.1	Objetivos	8
1.1.1	Objetivo Geral	8
1.1.2	Objetivos Específicos	8
1.2	Descrição dos Capítulos	9
2	Redes de Competição e Ativação Interativa	11
2.1	Modelo de Rumelhart e McClelland	11
2.2	Rede Modelo 'A'	18
2.3	Aprendizado em Redes IAC	19
2.4	Propriedades das Redes IAC	21
3	Algoritmos Genéticos	23
3.1	Representação	25
3.2	Inicialização da População	27
3.3	Avaliação da População	28
3.4	Seleção	29
3.5	Cruzamento	30
3.6	Mutação	32

3.7	Parâmetros	32
3.8	Condições de Término	33
3.9	Características	33
4	Estudos de Caso	35
4.1	O Exemplo dos Jets & Sharks	35
4.2	Diagnóstico Diferencial de Hepatite	38
4.3	Diagnóstico Diferencial em Reumatologia	42
4.3.1	Diagnóstico Diferencial em Reumatologia: Duas Possibilidades	43
4.3.2	Diagnóstico Diferencial em Reumatologia: Quatro Possibilidades	47
4.4	Resumo dos Exemplos	52
5	Metodologia de Treinamento	54
5.1	Representações	56
5.1.1	Alfabeto Binário	60
5.1.2	Alfabeto Inteiro	61
5.2	Obtenção do Conjunto de Treinamento	62
5.2.1	Conjunto de Treinamento com Pares de Entrada e Saída	62
5.2.2	Conjunto de Treinamento por Categorias	63
5.3	Geração da População Inicial	64
5.3.1	Geração Aleatória	65
5.3.2	Geração com Restrições	65
5.4	Processamento da Rede	65
5.5	Cálculo do <i>Fitness</i>	66
5.5.1	Cálculo Simples	66
5.5.2	Cálculo por 3 Valores	68

5.5.3	Cálculo por Ponderação	69
5.5.4	Função de <i>Fitness</i> para Categorias	72
5.5.5	Considerações sobre o <i>fitness</i>	73
5.5.6	Exemplos de Aplicação das Funções de <i>Fitness</i>	73
5.6	Seleção	75
5.7	Cruzamento	75
5.8	Seleção da Nova População	76
5.9	Mutação	76
6	Estabilidade	78
6.1	Conceitos Básicos	78
6.2	Definindo Estabilidade	80
6.3	Estabilidade em Sistemas Não Lineares	82
6.3.1	Crítério de Estabilidade de Liapunov	83
6.4	Estabilidade para RNAs com <i>Feedback</i>	84
6.4.1	Estabilidade de Redes IAC	84
6.5	Conclusões	88
7	Ambiente Computacional Implementado	89
7.1	Criação de uma Rede IAC	89
7.2	Consulta de uma Rede IAC	90
7.2.1	Implementação do Algoritmo de Atualização	90
7.3	Treinamento de uma Rede Modelo 'A'	94
7.4	Outras Funções	95

8	Resultados e Discussões	96
8.1	Etapa 1: Treinamento por Pares de Entrada/Saída	97
8.1.1	Representação dos Indivíduos	98
8.1.2	Cálculo do <i>Fitness</i>	115
8.1.3	Cruzamento	117
8.1.4	Considerações	118
8.1.5	Testes com o Exemplo de Diagnóstico Diferencial em Hepatite . . .	119
8.2	Etapa 2: Treinamento por Categorias	125
8.3	Etapa 3: Geração da População Inicial com Restrição e Treinamento por Pares de Entrada/Saída	132
9	Considerações Finais	138
9.1	Trabalhos Futuros	142

Lista de Figuras

1.1	Exemplos de redes diretas.	3
1.2	Exemplo de rede com ciclos.	4
2.1	Modelo de rede IAC original. O grupo cinza representa o grupo escondido e os demais representam os grupos visíveis.	13
2.2	Gráfico ilustrativo da ativação de um neurônio exposta ao <i>decay</i>	17
2.3	Rede IAC que modela o exemplo dos Jets & Sharks. Estão ressaltadas apenas algumas conexões.	18
2.4	Exemplo de uma rede Modelo 'A'.	19
3.1	Nomenclatura associada a um AG.	24
3.2	Esquema geral de um Algoritmo Genético.	25
3.3	Exemplo de aplicação de roleta ponderada.	30
3.4	Exemplo de aplicação do operador de cruzamento, realizado em um único ponto.	31
3.5	Exemplos de cruzamentos de dois pontos, considerando ou não o indivíduo circular.	31
3.6	Exemplo de aplicação do operador de mutação, quando utilizando alfabeto binário.	32
4.1	Esquema da rede modelo 'A' que implementa o Diagnóstico Diferencial em Reumatologia, com duas possibilidades.	43
4.2	Esquema da rede modelo 'A' que implementa o Diagnóstico Diferencial em Reumatologia, com quatro possibilidades.	48
5.1	Fluxograma geral desenvolvido para a metodologia de treinamento.	55

5.2	Exemplo de rede modelo 'A', onde $M = 6$ e $L = 3$	57
5.3	Rede modelo 'A', onde os grupos 1 e 2 não têm relação entre si e representam 'conceitos' com 'valores' mutuamente exclusivos.	59
5.4	Exemplo de representação binária com 10 genes por peso.	60
5.5	Exemplo de representação inteira, usando um gene por peso.	61
5.6	Fenótipo gerado a partir do genótipo da Figura 5.5.	61
5.7	Exemplo de representação inteira, usando dois genes por peso.	62
5.8	Fenótipo gerado a partir do genótipo da Figura 5.7.	62
5.9	Exemplo de conjunto de treinamento, quando usando categorias para os pesos.	64
5.10	Exemplos de valores de <i>fitness</i> , fornecido pelas diferentes funções.	74
6.1	Sistema de malha fechada.	82
7.1	Resposta do software PDP para a entrada Jets.	91
7.2	Resposta do sistema para a entrada Jets.	91
7.3	Resposta do software PDP para a entrada Ol.	92
7.4	Resposta do sistema para a entrada Ol.	92
7.5	Resposta do software PDP para a entrada faixa etária de 20 anos e casados.	93
7.6	Resposta do sistema para a entrada faixa etária de 20 anos e casados.	93
8.1	Gráfico genérico, utilizado para os exemplos dos Jets e Sharks. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.	97
8.2	JS com 10 neurônios: Resultados obtidos quando são encontrados todos os pesos da matriz de conectividade, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.	100
8.3	JS com 21 neurônios: Resultados obtidos quando são encontrados todos os pesos da matriz de conectividade, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.	101

8.4	JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.	102
8.5	JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.	103
8.6	JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original, usando entradas gerais para treinamento. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto. .	104
8.7	JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original, usando entradas gerais para treinamento. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxa total de acertos.	104
8.8	JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	105
8.9	JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 6 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	106
8.10	JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 6 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	106
8.11	JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 6 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxa total de acertos.	106

8.12 JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 5 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	107
8.13 JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 5 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	108
8.14 JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 5 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	108
8.15 JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	109
8.16 JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	109
8.17 JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	109
8.18 JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 2 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	110
8.19 JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 2 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	110

8.20 JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 2 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	111
8.21 JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	112
8.22 JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	112
8.23 JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original, treinados com as entradas gerais. Alfabeto binário com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	115
8.24 JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original, treinados com as entradas gerais. Alfabeto binário com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	115
8.25 JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando a função de <i>fitness</i> que pondera ativos e inativos, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	116
8.26 JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando a função de <i>fitness</i> que pondera ativos e inativos, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	116
8.27 JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando cruzamento de dois pontos, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	117

8.28 JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando cruzamento de dois pontos, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	117
8.29 JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando cruzamento de dois pontos, sem considerar o indivíduo circular, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	118
8.30 JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando cruzamento de dois pontos, sem considerar o indivíduo circular, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	118
8.31 Hepatite com 13 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.	122
8.32 Hepatite com 14 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.	122
8.33 Hepatite com 17 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.	122
8.34 Exemplo de erro fornecido pelas redes treinadas, para o exemplo de Hepatite com 17 neurônios.	123
8.35 Hepatite com 44 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.	124
8.36 Hepatite com 44 neurônios: Exemplo de associação incorreta entre pesos.	124
8.37 JS com 41 neurônios: Resultados obtidos quando são encontrados pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	126
8.38 JS com 41 neurônios: Resultados obtidos quando são encontrados pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.	126

8.39	Hepatite com 44 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.	127
8.40	(a)Exemplo de consulta, indicando um quadro de sintomas mais associado à Hepatite B. (b)Exemplo de consulta, onde não se inativou o diagnóstico de Hepatite B, mesmo em presença do exame negativo para este diagnóstico.	128
8.41	Diagnóstico em Reumatologia com 29 neurônios. (a) Taxa de coincidência usando conjunto de teste com 29 entradas com 1 neurônio ativo cada. (b) Taxa de coincidência usando conjunto de teste com 29 entradas com 2 neurônios ativos cada. .	129
8.42	Diagnóstico em Reumatologia com 50 neurônios. (a) Taxa de coincidência usando conjunto de teste com 50 entradas com 1 neurônio ativo cada. (b) Taxa de coincidência usando conjunto de teste com 50 entradas com 2 neurônios ativos cada. .	130
8.43	Diagnóstico em Reumatologia com 29 neurônios: Exemplo. (a) Exemplo de consulta com resposta indevida, devido à limites das categorias aplicadas (b) Resposta esperada, de acordo com exemplo implementado por De Azevedo.	131
8.44	Diagnóstico em Reumatologia com 50 neurônios: Exemplo. (a) Exemplo de consulta com resposta indevida, devido à limites das categorias aplicadas. (b) Resposta esperada, de acordo com exemplo implementado por De Azevedo.	132
8.45	JS com 41 neurônios: Resultados obtidos quando são encontrados pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.	133
8.46	JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.	133
8.47	Hepatite com 44 neurônios: Taxas de acerto fornecidas.	134
8.48	Diagnóstico em Reumatologia com 29 neurônios. (a) Taxa de coincidência usando conjunto de teste com 29 entradas com 1 neurônio ativo cada. (b) Taxa de coincidência usando conjunto de teste com 29 entradas com 2 neurônios ativos cada. .	135
8.49	Diagnóstico em Reumatologia com 50 neurônios. (a) Taxa de coincidência usando conjunto de teste com 50 entradas com 1 neurônio ativo cada. (b) Taxa de coincidência usando conjunto de teste com 50 entradas com 2 neurônios ativos cada. .	136

8.50 Diagnóstico em Reumatologia com 50 neurônios: Exemplo. (a) Resposta de uma consulta para uma rede gerada por treinamento. (b) Resposta dada pela implementação de De Azevedo. 137

Lista de Tabelas

3.1	Nomenclatura utilizada pelos AGs.	24
4.1	Conhecimento sobre o problema dos Jets & Sharks.	36
4.2	Problema dos Jets & Sharks, considerando apenas cinco indivíduos e três 'conceitos'.	38
4.3	Parte da matriz de conectividade para uma rede que representa o problema dos Jets & Sharks com 10 neurônios, organizados em três grupos.	38
4.4	Organização dos 'conceitos' para o exemplo de diagnóstico de Hepatite.	39
4.5	Organização dos 'conceitos' para o exemplo de diagnóstico de Hepatite, com 13 neurônios.	41
4.6	Organização dos 'conceitos' para o exemplo de diagnóstico de Hepatite, com 14 neurônios.	42
4.7	Organização dos 'conceitos' para o exemplo de diagnóstico de Hepatite, com 17 neurônios.	42
4.8	Forma geral da matriz de conectividade para a rede de 29 neurônios.	44
4.9	Matriz de conectividade entre os neurônios do grupo 'diagnóstico'.	44
4.10	Matriz de conectividade entre os grupos 'diagnóstico' e 'sintomas'.	46
4.11	Matriz de conectividade entre os grupos 'sintomas' e 'pacientes'.	47
4.12	Matriz de conectividade entre os grupos 'diagnóstico' e 'pacientes'.	47
4.13	Forma geral da matriz de conectividade para a rede de 50 neurônios.	49
4.14	Matriz de conectividade entre os neurônios do grupo 'diagnóstico', para a rede de 50 neurônios.	49

4.15	Matriz de conectividade entre os grupos 'diagnóstico' e 'sintomas', para a rede de 50 neurônios.	50
4.16	Matriz de conectividade entre os grupos 'sintomas e 'pacientes', para a rede de 50 neurônios.	51
4.17	Matriz de conectividade entre os grupos 'sintomas' e 'pacientes', para a rede de 50 neurônios.	52
4.18	Matriz de conectividade entre os grupos 'diagnóstico' e 'pacientes', para a rede de 50 neurônios.	53
5.1	Matriz de conectividade da rede modelo 'A' apresentada na Figura 5.2.	57
5.2	Modelo geral da matriz de conectividade para a rede da Figura 5.3.	59
5.3	Exemplo de uma matriz MI , para o Grupo 1.	59
5.4	Exemplo de conjunto de treinamento, com $NP = 2$, para a rede da Figura 5.2.	63
5.5	Categorias de pesos para a matriz de conectividade da Figura 5.3.	64
5.6	Exemplos de cálculo do <i>fitness</i>	74
8.1	Matriz de conectividade para JS com 10 neurônios, obtida quando o indivíduo engloba todos os seus pesos - Parte 1.	99
8.2	Matriz de conectividade para JS com 10 neurônios, obtida quando o indivíduo engloba todos os seus pesos - Parte 2.	100
8.3	Taxas de acertos ativos e inativos para a rede apresentada na Tabela 8.4.	113
8.4	Submatriz C, para uma rede JS com 41 neurônios.	113
8.5	Resultados parciais, considerando as entradas específicas e as entradas gerais.	114
8.6	Resultados obtidos - Geral.	114
8.7	Conjunto de teste para o caso de Hepatite com 13 neurônios.	121
8.8	Pesos utilizados por De Azevedo.	131
8.9	Exemplo de pesos encontrados pelo AG.	131

Lista de Abreviaturas

AG	Algoritmos Genéticos
AGCR	Algoritmos Genéticos com Codificação Real
AGt	Artrite de Gota
AP	Artrite Psoriática
AR	Artrite Reumatóide
CE	Computação Evolucionária
IA	Inteligência Artificial
IAC	Interactive Activation and Competition
JS	Jets & Sharks
LES	Lupus Eritematoso Sistêmico
RNA	Redes Neurais Artificiais
SE	Sistemas Especialistas
SNC	Sistema Nervoso Central

Capítulo 1

Introdução

A Inteligência Artificial (IA) surgiu a partir de idéias de pesquisadores que pretendiam atribuir aos computadores um comportamento inteligente. Com o prosseguimento das pesquisas, observou-se que esta é uma tarefa que apresenta um alto grau de complexidade. Apesar disso, os estudos continuam nos três principais paradigmas apresentados pela IA – o simbólico, o conexionista e o evolucionário – e excelentes resultados vêm sendo obtidos nestas áreas.

Na IA Simbólica, representada, entre outros, pelos Sistemas Especialistas (SE), tenta-se reproduzir o funcionamento do raciocínio de um especialista, na solução de um problema, em uma determinada área do conhecimento. Este tipo de paradigma é interessante para resolver tarefas onde é possível articular, com relativa facilidade, o conhecimento necessário para solucioná-las. Uma das principais formas de se representar o conhecimento, ao se utilizar um SE, é por meio de regras.

Na IA Conexionista, tem-se como objetivo construir modelos que imitem o neurônio biológico e a organização neurológica do cérebro humano, simulando, desta forma, comportamentos inteligentes [1]. Aplica-se melhor em modelos onde não se tem conhecimento explícito de como realizar determinada tarefa, ou quando os especialistas não conseguem formular regras apropriadas para a sua resolução. Existem vários tipos de sistemas conexionistas, tais como Redes Neurais Artificiais, Redes de Autômatos e Redes Bayesianas.

As Redes Neurais Artificiais (RNA) permaneceram por muitos anos apenas como pesquisas de laboratório, sem muita utilização prática. A partir da década de 80, houve um grande incremento no seu uso. Diferentes fatores colaboraram para esta expansão e o maior, talvez, tenha sido a capacidade de aprender das RNAs [2], que passaram, então, a ser aplicadas às mais diversas áreas, como por exemplo, reconhecimento de padrões, reconhecimento de voz,

processamento de sinais, modelagem e controle neural, predição, processamento de imagens, classificação e diagnóstico de falhas.

Na área biomédica, há, também, uma série de trabalhos que utilizam RNAs como ferramenta. Entre estes, pode-se citar:

- Trabalhos na área de análise de imagens médicas [3, 4];
- Trabalhos na área de processamento de sinais biomédicos [5, 6, 7];
- Detecção e análise de eventos epileptiformes em EEG [8, 9];
- Detecção de hipoglicemia baseada na análise e classificação do eletroencefalograma [10];
- Detecção de fadiga muscular, por análise do Eletromiograma [11];
- Detecção de contração ventricular prematura [12].

As RNA podem ser definidas como sistemas conexionistas formados por um conjunto de unidades elementares de processamento, baseadas no neurônio biológico, e conectadas entre si. As conexões são valoradas e a rede pode apresentar um equacionamento matemático, através do qual os pesos das conexões são ajustados. Isto permite que se aprenda uma determinada tarefa, interagindo sobre exemplos de treinamento. Elas são caracterizadas por uma série de componentes, tais como [13, 14]:

- Um conjunto de unidades de processamento, também chamadas de neurônios;
- Um estado de ativação, que está definido para cada unidade da rede;
- Uma função de saída, que utiliza como argumento o valor de ativação e que fornece um valor de saída a ser transmitido para outras unidades da rede;
- Topologia, que é o padrão de conectividade entre as unidades, e determina como cada uma delas está conectada às outras pertencentes à rede. O padrão de conectividade também especifica que unidades podem se conectar a uma outra particular;
- Regra de propagação, ou função de combinação, que propaga as atividades da unidade através da rede;
- Regra de ativação, que muda a atividade de cada unidade, usando o valor de ativação corrente e as entradas recebidas de outras unidades;

- Ambiente externo, que fornece informação para a rede ou interage com ela;
- Regra de aprendizado, que modifica o padrão de conectividade, usando informação fornecida pelo ambiente externo.

Embora não haja uma norma estabelecida, algumas destas características servem como forma de classificar a rede, tais como a unidade de processamento, a forma de conexão ou topologia e o tipo de aprendizado.

A unidade de processamento, ou neurônio, é a peça fundamental que compõe uma RNA. A rede pode ser classificada como homogênea ou heterogênea, dependendo se as unidades utilizadas são iguais ou não. A partir do primeiro modelo, proposto por McCulloch e Pitts [15], foram propostos vários outros, que permitem a produção de uma saída qualquer, não necessariamente zero ou um, e com diferentes funções de ativação, tais como o encontrado em De Azevedo [16].

Na topologia, estão incluídas características como o número de unidades de processamento, o número de camadas e as conexões entre as unidades de processamento das diversas camadas. Não existe uma regra definida para a determinação destes parâmetros, que são escolhidos de forma empírica, com base na experiência do pesquisador e no tipo de aplicação que está sendo desenvolvida. Existem alguns trabalhos realizados no sentido de auxiliar a escolha destes parâmetros, tais como aquele desenvolvido por Brasil [17]. Entre as diferentes topologias, as mais comuns são as redes diretas ou *feedforward* e as redes com realimentação ou *feedback*. Existem, ainda, as redes simétricas.

Nas redes diretas, as conexões não formam ciclos, ou seja, as unidades de processamento não recebem sua própria saída como entrada, nem de forma direta, nem de forma indireta, por meio de outras unidades de processamento. São comumente representadas por redes que têm suas unidades organizadas em camadas, onde cada uma delas pode comunicar-se exclusivamente com a seguinte, conforme ilustrado na Figura 1.1. Pode haver uma ou mais camadas de pesos adaptativos. Estas são, atualmente, as redes mais populares, principalmente, por existirem métodos de aprendizado bastante difundidos e documentados.

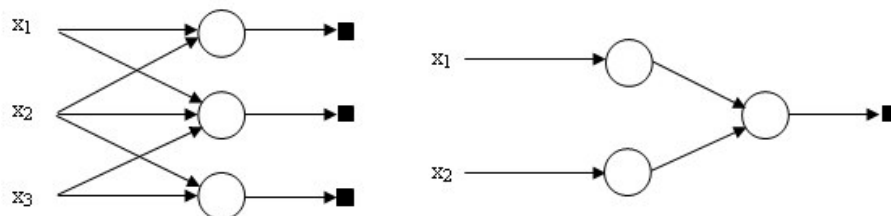


Figura 1.1: Exemplos de redes diretas.

Entretanto, as redes com realimentação ou *feedback*, ao contrário das diretas, apresentam a saída de uma unidade de processamento como entrada para ela mesma, direta ou indiretamente, formando grafos de conectividade com ao menos um ciclo. Um exemplo de rede com ciclos está ilustrado na Figura 1.2. Uma rede com ciclos pode ser constituída de uma única camada, onde cada unidade de processamento usa seu sinal de saída para alimentar as entradas de todas as outras unidades.

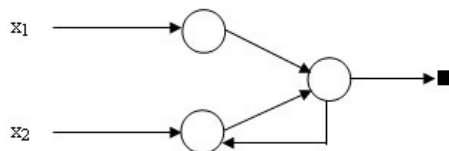


Figura 1.2: Exemplo de rede com ciclos.

As redes simétricas são um caso particular de redes com ciclos, que apresentam a matriz de conectividade simétrica. O seu grafo tem arcos não orientados. Da mesma forma que outras redes, elas podem ter unidades de processamento de entrada, de saída e internas. Aqui, destacam-se as redes IAC, que são objeto de estudo deste trabalho.

O aprendizado é uma característica de extrema importância para a maioria dos modelos de RNA, já que permite que a rede seja treinada através do ajuste de pesos das conexões, sendo assim, possível que ela aprenda os padrões que estão sendo apresentados [18]. Uma das formas mais comuns de classificação de RNA é segundo o tipo de aprendizado que ela apresenta, sendo os dois tipos principais o supervisionado e o não-supervisionado.

No aprendizado supervisionado, um conjunto de treinamento, formado por pares de entrada e de saída, é apresentado à rede, de modo que esta verifique, a cada uma das entradas, se a saída obtida é igual à desejada. Se não for, a distância entre as duas é medida, obtendo-se um erro com o qual os pesos da rede são ajustados. Isto significa que uma correção é introduzida a cada iteração, de modo a encontrar uma ligação entre os diversos pares, que são apresentados repetidamente, esperando-se que o erro diminua, até ficar dentro de um limite aceitável.

No aprendizado não-supervisionado, a saída desejada não é conhecida. Sendo assim, não existe uma medida de erro para efetuar a correção dos parâmetros da rede. Portanto, os padrões de entrada são apresentados repetidamente, até serem automaticamente particionados em classes. Utilizando-se de técnicas de inibição lateral e competição, entre outras, espera-se que a rede responda de modo semelhante a exemplos semelhantes.

Uma outra forma de aprendizado é dada pelo aprendizado evolutivo, que se utiliza de técnicas fornecidas pelo paradigma evolucionário da IA, tais como Algoritmos Genéticos

(AG). Por este método, que foi inspirado na evolução biológica, é possível determinar vários parâmetros pertencentes a uma rede.

Entre os diversos modelos de rede existentes, alguns apresentam treinamento supervisionado, tais como o Perceptron de múltiplas camadas, enquanto outros apresentam treinamento não-supervisionado, tais como as redes de Kohonen [18]. Entretanto, algumas redes não possuem uma fase real de aprendizado, ou seja, o conhecimento é inserido a priori, com a determinação e armazenamento dos pesos na sua própria estrutura. Este é o caso das redes de Competição e Ativação Interativa (do inglês *Interactive Activation and Competition - IAC*).

Nas redes IAC, o princípio de operação é o mesmo daquele das redes de Hopfield [19, 20], isto é, o projetista determina a topologia e o estado inicial da rede [21]. Esta evolui, então, para um estado de equilíbrio que representa a resposta para um determinado questionamento efetuado.

As redes IAC apresentam, como principal característica, a recuperação de informação geral e específica de padrões de informação previamente armazenados na rede. Ou seja, ela pode ser usada como uma memória associativa bidirecional, onde informação incorreta, parcial ou ruidosa apresentada, resulta na recuperação da informação correta ou completa, após um novo estado de equilíbrio ser encontrado. Esta característica teria uma grande aplicabilidade, caso a rede original IAC não apresentasse certas restrições.

Uma das principais restrições existentes é a limitação nos pesos das conexões entre os neurônios, que podem assumir somente os valores -1, 0 e +1. Numa rede IAC, as unidades de processamento se encontram organizadas em grupos, onde cada grupo representa um 'conceito' ou 'característica' relacionado a um problema, e as unidades dentro deste grupo representam 'valores' que podem ser assumidos por este 'conceito'. As conexões entre unidades pertencentes a um mesmo grupo podem assumir, exclusivamente, o valor -1, e as conexões entre unidades de diferentes grupos assumem valor +1, se existe relação entre elas, e valor 0, caso contrário.

Portanto, existe inibição completa entre os neurônios de um mesmo grupo, indicando que o 'conceito' deve apresentar 'valores' mutuamente exclusivos. Da mesma forma, quando há relação entre duas unidades de processamento, esta deve ser total. Estas características não são encontradas em muitos problemas existentes no mundo real, o que limita a aplicabilidade da rede.

Assim, outros pesquisadores estudaram as características do modelo original, pretendendo ampliar a sua aplicação. Em alguns de seus trabalhos, De Azevedo [22] usou uma rede inspirada nos princípios da rede IAC, chamada rede modelo 'A', mas que apresentava como

característica adicional o fato de possuir pesos de conexões reais, no intervalo $[-1, 1]$, permitindo que uma quantidade maior de problemas seja representado.

A rede IAC, por assumir apenas os valores 1, 0 ou -1 para as conexões, não necessita de um algoritmo que realize o ajuste de pesos. Deve-se levar em consideração, também, que estas redes são aplicadas em problemas onde o conhecimento, traduzido por meio das relações entre os diferentes 'valores' dos 'conceitos', é muito bem definido. Já no caso da rede modelo 'A', os pesos das conexões são reais, transformando o seu ajuste em um processo bastante demorado, visto que até o presente momento nenhum algoritmo para seu aprendizado foi desenvolvido. Isto representa uma forte limitação para este tipo de rede, pois o ajuste manual exige um grande esforço, tornando o seu uso, de certa forma, menos interessante.

Por causa das restrições apresentadas tanto pelo modelo original, como pelo modelo 'A', as redes IAC têm sido aplicadas em poucos trabalhos, sendo alguns destes mencionados a seguir.

Lua e Gan [23] utilizaram redes IAC em tarefas de reconhecimento de padrões, mais especificamente no reconhecimento de caracteres chineses.

Burton, Bruce e Johnston [24] utilizaram as IAC na modelagem dos processos envolvidos em reconhecimento de faces, de forma plausível biológica e psicologicamente.

Barreto e De Azevedo [25] utilizaram o modelo 'A' para implementar sistemas especialistas de auxílio ao diagnóstico.

Jaszlics, Jaszlics e Jones [26] propuseram um modelo alternativo de redes IAC, denominado Linear Interactive Activation and Competition (LINIAC). Este modelo, de alta velocidade e orientado a objeto, foi usado em simulações de 'jogos de guerra' (ou, mais precisamente, *Distributed Interactive Battle Simulations*) mostrando-se promissor.

Nascimento e Zarrop [21] provaram que redes IAC, tanto quanto as redes de Hopfield [19], podem ser usadas para resolução de problemas de otimização quadrática.

Cheng [27] descreveu um método, baseado no modelo 'A', para armazenamento e recuperação de informação científica, clínica e médica.

Tambouratzis e Antonopoulos-Domis [28] aplicaram redes IAC na estimação e monitorização do tempo de transição (*transit time*) em reatores à água fervente (*boiling water reactors* - BWRs). O sistema proposto apresentou-se robusto na presença de componentes global e local, bem como na adição de ruído branco não correlacionado.

Ni et al. [29] examinaram a possibilidade de utilizar redes IAC como uma alternativa para implementação de sistemas que utilizem técnicas de Raciocínio Baseado em Casos (RBC).

Garcez, Oliveira et al. [30] utilizaram o modelo 'A' para a construção de um sistema para apoio à decisão, relacionada à responsabilidade por danos em equipamentos de consumidores, em uma concessionária de distribuição de energia elétrica.

Saldías [31] propôs o uso do modelo 'A' em mecanismos de adaptação de interfaces. Nesta aplicação, o modelo identifica usuários de um sistema tutorial para portadores de Diabetes Mellitus, indicando qual o melhor conteúdo e a forma como este conteúdo deve ser apresentado ao usuário.

Nesta mesma linha, Barbosa [32] prevê o uso de redes IAC em mecanismos de adaptação de interfaces, para *sites* adaptativos.

Dazzi [33] pretende realizar uma implementação da metodologia proposta em [31], por meio de agentes inteligentes.

Entre os trabalhos citados, existem alguns que propõem mecanismos de aprendizado. Entretanto, estes mecanismos afetavam características fundamentais do modelo, como por exemplo, a bidirecionalidade, que permite que qualquer neurônio da rede possa ser considerado como entrada ou saída. Este é o caso do trabalho apresentado em [24].

Os trabalhos de Saldías [34], Barbosa [32] e Dazzi [33] foram desenvolvidos no Instituto de Engenharia Biomédica da UFSC (IEB). Em [34], que pretende identificar um usuário de um sistema tutorial para portadores de Diabetes Mellitus, os grupos da rede modelo 'A' utilizados podem ser divididos em atributos de interface e características do usuário. A partir das características do usuário, extraídas por meio de um questionário, a rede modelo 'A' define os atributos de interface, ou seja, define qual o conteúdo e a forma como este conteúdo deve ser apresentado. Entretanto, se durante a navegação, o usuário demonstrar mais interesse por outro conteúdo ou forma de apresentação, é possível realimentar as suas características, a partir destas informações. Isto é possível, graças à característica de bidirecionalidade inerente à rede IAC, tanto original, quanto modelo 'A'.

O desenvolvimento deste trabalho indicou que o ajuste de pesos, feito por métodos de tentativa e erro, é um processo complexo e demorado. Embora os resultados obtidos fossem aceitáveis, necessitavam de um maior refinamento. Um algoritmo de aprendizado se torna interessante, para que seja possível automatizar esta tarefa.

Entretanto, em [25] e [30], foram utilizadas técnicas de elicitação de conhecimento para montar as matrizes de conectividade. Este método, porém, não se mostra robusto o suficiente para redes maiores.

A ferramenta escolhida para solucionar o problema do mecanismo de aprendizado é originada do paradigma Evolucionário da IA: os Algoritmos Genéticos. O AG fornece uma

ferramenta de fácil utilização em mecanismos de busca e otimização. A sua estrutura básica pode ser utilizada para qualquer problema de otimização que se queira resolver. Existem, porém, dois pontos que são dependentes do problema em questão, que são a função de *fitness* e a representação para o indivíduo. Se estes não são definidos de maneira a representar adequadamente o problema, o AG acaba tendo um desempenho ruim, e pode não encontrar uma solução. Desta forma, os principais esforços devem se concentrar em propor diferentes formas de representações e funções a serem aplicadas a diferentes exemplos ilustrados.

Os AGs já têm sido utilizados com sucesso no treinamento de alguns tipos de Redes Neurais Artificiais. Em redes Perceptron Multicamadas, têm sido utilizados para auxiliar em diversas tarefas, como, por exemplo, na determinação dos parâmetros ótimos do algoritmo *Backpropagation*. Podem ser usados, também, para determinar a melhor topologia de rede a ser utilizada, incluindo o número de camadas intermediárias, assim como o número de neurônios contidos em cada uma delas [35]. A definição destes valores, ou seja, a otimização da arquitetura de uma RNA é um problema considerado bastante complexo [36]. Sendo assim, este tipo de paradigma se mostra bastante interessante em problemas críticos, onde uma estrutura otimizada de topologia de rede seja necessária. Outras redes que têm utilizado AGs no seu treinamento, ou na procura da topologia ideal, são as redes de base radial [37], redes recorrentes [38, 39, 40] e as redes neurais *fuzzy* [41].

1.1 Objetivos

A partir do que foi exposto acima, pode-se definir os objetivos deste trabalho, apresentados a seguir.

1.1.1 Objetivo Geral

O objetivo geral desta pesquisa é:

- Desenvolver propostas de algoritmos de aprendizado, baseados em Algoritmos Genéticos, para redes modelo 'A'.

1.1.2 Objetivos Específicos

Entre os objetivos específicos traçados para a pesquisa, têm-se:

- Encontrar uma representação adequada para o problema;
- Desenvolver uma ou mais funções de *fitness*, que forneçam uma medida da aptidão de cada indivíduo para solucionar o problema;
- Testar as propostas desenvolvidas;
- Realizar estudos sobre a estabilidade da rede;
- Construir um sistema computacional que permita:
 - Criar redes IAC, de ambos os modelos, automaticamente;
 - Realizar consultas às redes, tanto geradas manualmente, como geradas por treinamento;
 - Realizar o treinamento de redes IAC modelo 'A';
 - Realizar estatísticas sobre o funcionamento da rede.

1.2 Descrição dos Capítulos

Este trabalho está dividido em 9 capítulos, organizados como segue abaixo:

- Este capítulo contextualiza o problema que se pretende resolver e define os objetivos traçados para o trabalho.
- O capítulo 2 apresenta as redes IAC, tanto o modelo original, quanto o modelo 'A', seu equacionamento e algumas características que estas redes apresentam.
- O capítulo 3 fala sobre Algoritmos Genéticos. Aborda a nomenclatura utilizada, a estrutura básica e seus operadores principais, que são o cruzamento e a mutação. Traz, também, um AG simples, abordando os passos envolvidos e as possibilidades de implementação em cada passo.
- O capítulo 4 apresenta os exemplos que foram utilizados para realização dos testes com a metodologia proposta. Foram escolhidos exemplos que abrangessem tanto características mutuamente exclusivas, como não mutuamente exclusivas. Também se procurou problemas que apresentassem as relações entre unidades de diferentes grupos assumindo diferentes graus. Os três exemplos selecionados foram: Jets & Sharks, Diagnóstico Diferencial de Hepatite e Diagnóstico Diferencial em Reumatologia.

- O capítulo 5 apresenta todas as propostas elaboradas durante o desenvolvimento deste trabalho. Enfoca-se, principalmente, o desenvolvimento da representação e das funções de *fitness*. São apresentados os diferentes alfabetos utilizados, os conjuntos de treinamento criados e as funções de *fitness* específicas para cada um deles. Também são apresentados as diferentes possibilidades implementadas para os operadores genéticos, como seleção, cruzamento e mutação.
- O capítulo 6 aborda a questão da estabilidade relacionada a redes modelo 'A', com base na prova feita por Nascimento [21].
- O capítulo 7 apresenta o sistema que foi criado para implementar as propostas apresentadas. Este sistema permite criar redes IAC, tanto no modelo original, como o modelo 'A', permite treinar as redes modelo 'A' e realizar consultas às redes geradas. É possível, também, aplicar conjuntos de testes a redes obtidas por treinamento, gerando algumas estatísticas, que permitem analisar o seu comportamento.
- O capítulo 8 apresenta todos os testes realizados e as discussões pertinentes, relacionadas aos resultados obtidos. Está dividido em 3 etapas, que se diferenciam pelo conjunto de treinamento utilizado, o que implica em diferentes funções de *fitness*. Outro ponto que as diferencia é a forma como a população inicial é gerada. São utilizados exemplos completos apresentados em 4, assim como versões resumidas, para um melhor acompanhamento do desempenho das propostas elaboradas.
- O capítulo 9 traz as considerações finais a respeito do trabalho.

Capítulo 2

Redes de Competição e Ativação Interativa

As redes de competição e ativação interativa, ou redes IAC, são redes com representação local, que fornecem um mecanismo de memórias associativas. O conceito de memória associativa é intuitivo, e é uma das funções primárias do cérebro humano. É este conceito que permite que se associe um nome à face de uma pessoa conhecida, e, também, que se possa reconstituir padrões corrompidos ou incompletos.

Esta rede foi proposta por McClelland e Rumelhart para modelar reconhecimento de palavras e a recuperação de informações gerais e específicas do conhecimento armazenado sobre indivíduos [42]. Uma das características mais marcantes deste modelo de rede é o fato de seus pesos sinápticos serem fixos, ou seja, serem definidos *a priori*, assumindo apenas os valores 1, 0 e -1.

Entretanto, foram propostas variações, baseadas no modelo original, tais como a apresentada em [22], chamada de Modelo 'A', que permite que se utilize pesos no intervalo $[-1, 1]$. Estes dois modelos são abordados neste capítulo.

2.1 Modelo de Rumelhart e McClelland

O modelo original da IAC consiste numa rede com *feedback* que opera em tempo contínuo ou discreto e cujas saídas são números reais [43]. A rede é composta, basicamente, de unidades de processamento organizadas em grupos. Cada grupo representa um 'conceito' ou 'característica' para o problema em questão, e cada unidade dentro do grupo representa uma possível 'propriedade' ou 'valor' para o 'conceito'. As conexões entre as unidades têm

um peso associado, que pode assumir exclusivamente os valores -1 (conexão inibitória) e 1 (conexão excitatória), além do valor 0, correspondente à ausência de relação.

Num mesmo grupo, todas as unidades são conectadas a todas as outras, salvo a ela mesma ($w_{ii} = 0$), através de conexões inibitórias, gerando um processo de competição (daí o *competition* de IAC). Entre unidades de grupos diferentes, as conexões são excitatórias, caso haja relação direta entre elas, ou assumem o valor 0. Quando duas unidades têm conexão excitatória, ao se ativar uma delas, ela tende a ativar, também, a outra unidade com a qual está conectada (daí o *activation* de IAC).

Estes pesos formam uma matriz W de dimensões MXM , onde M é o número de unidades existentes na rede, e cada linha e cada coluna representam um neurônio. Esta matriz é simétrica, pois as conexões existentes entre as unidades são bidirecionais. Isto quer dizer que, existindo uma conexão de uma unidade i para uma unidade j , existirá também uma conexão, de mesmo valor, da unidade j para a unidade i ($w_{ij} = w_{ji}$). Desta forma, o processamento se torna interativo (daí o *interactive* de IAC), já que as unidades em um determinado grupo influenciam e são influenciadas por unidades de outros grupos da rede, que podem ser de dois tipos:

- Grupos visíveis: contêm unidades que recebem entradas externas. Podem ter três tipos de conexões, que são aquelas com unidades do mesmo grupo, com unidades de outro grupo e com o exterior;
- Grupo escondido: contém unidades que não podem receber entradas do exterior, sendo suas entradas obtidas apenas como a combinação das saídas de outras unidades às quais elas estão conectadas. Tem como finalidade fazer a ligação entre os diferentes 'conceitos'. Normalmente, tem o mesmo número de neurônios que o maior grupo visível.

Na Figura 2.1, é apresentado um modelo de rede IAC. Os círculos pretos representam os neurônios. Cada círculo transparente representa um grupo visível, que traz os neurônios associados ao 'conceito' que ele representa, podendo receber entradas vindas do exterior. O círculo cinza representa o grupo escondido, que não pode receber entradas vindas do meio externo. Pode-se observar, também, que não existem conexões entre grupos visíveis, sendo estas feitas através do grupo escondido. Neste modelo, cada neurônio do mesmo grupo inibe todos os outros. As conexões bidirecionais negativas entre neurônios do mesmo grupo não estão apresentadas nesta figura.

Cada unidade da rede tem um nível de ativação, que é modificado por três diferentes tipos de estímulos:

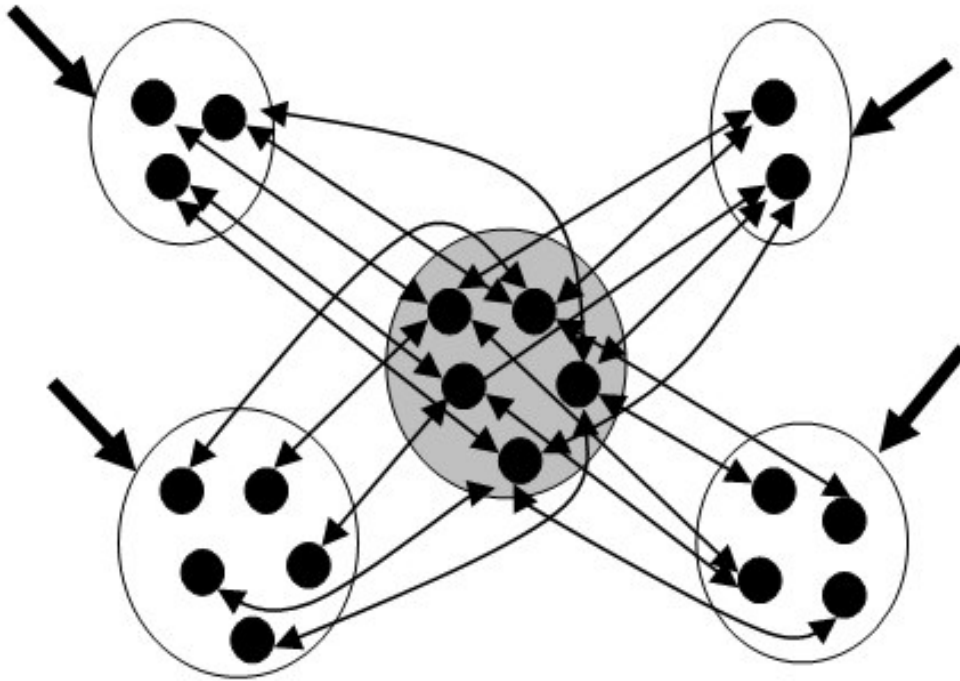


Figura 2.1: Modelo de rede IAC original. O grupo cinza representa o grupo escondido e os demais representam os grupos visíveis.

- os provenientes de fora da rede (para os grupos visíveis) e que correspondem a uma afirmação da existência daquele 'valor' particular para o 'conceito';
- os provenientes de outras unidades do mesmo grupo, através das conexões inibitórias existentes entre todos os 'valores' para o 'conceito' que este grupo representa. Estes estímulos provocam competição entre essas unidades;
- os provenientes de unidades do grupo escondido, através de conexões excitatórias. Estas conexões representam as relações existentes entre os diferentes 'conceitos', ou seja, entre os diferentes grupos visíveis.

A ativação de um neurônio na rede IAC evolui gradualmente no tempo. A unidade que teve sua ativação modificada pela entrada externa à rede influenciará as unidades com as quais ela tem conexão. Ou seja, ela tende a inibir a ativação das unidades pertencentes ao mesmo grupo, que estão conectadas a ela com pesos -1 , e excitar a unidade correspondente no grupo escondido, que está conectada a ela com valor $+1$. É esta unidade pertencente ao grupo escondido que fará com que as unidades relacionadas de outros grupos sejam excitadas, desde que também estejam conectadas a ela com conexões de valor $+1$.

Considere uma unidade simples. Se a maioria das unidades ativas na rede é conectada a esta unidade com pesos positivos, então a sua ativação tende a ser incrementada. Se a

maioria das unidades ativas é conectada a ela com pesos negativos, então sua ativação deve ser diminuída.

Desta forma, pode-se dizer que o processo de mudança de ativação ocorre de acordo com uma função de ativação, que considera tanto a ativação atual da unidade como também a entrada da rede procedente de outras unidades, ou oriunda de fora da rede para esta unidade. A equação é, então, semelhante à maioria dos modelos encontrados, ou seja, é a soma das entradas recebidas, calculadas pelo produto da saída pelo peso da respectiva conexão, mais as entradas externas à rede para a unidade.

Para calcular a entrada de uma unidade i , é necessário calcular as entradas excitatórias exc_i (dada pela Equação 2.1), que levam em conta somente as entradas com conexões positivas, e as entradas inibitórias $inib_i$ (dada pela Equação 2.2), que levam em conta somente as entradas com conexões negativas.

$$exc_i = \sum_{j=1}^M w_{ij} * ext_j, \quad \text{com } w_{ij} \geq 0 \quad (2.1)$$

$$inib_i = \sum_{j=1}^M w_{ij} * ext_j, \quad \text{com } w_{ij} < 0 \quad (2.2)$$

onde:

w_{ij} Peso da conexão entre a unidade i e a unidade j
 ext_j Valor da entrada externa para a unidade j

Assim, a Equação 2.3 mostra a relação característica da entrada de uma unidade i (net_i) considerando um modelo de uma rede IAC:

$$net_i = exc_i * \alpha + inhib_i * \gamma + ext_i * estr \quad (2.3)$$

onde:

exc_i Valor das entradas excitatórias para a unidade i
 $inib_i$ Valor das entradas inibitórias para a unidade i
 ext_i Valor da entrada externa para a unidade i
 α Gradua a força das entradas excitatórias
 γ Gradua a força das entradas inibitórias
 $estr$ Gradua a força das entradas externas

A cada nova entrada calculada, obtém-se uma mudança nas ativações desta unidade i . O resultado desta alteração é obtido através das Equações 2.4 e 2.5, que são as equações de atualização das ativações y_i . Considerando que $net_i > 0$, então:

$$\Delta y_i = (max - y_i)net_i - decay(y_i - rest), \quad \text{para } net_i > 0 \quad (2.4)$$

onde:

y_i	Ativação para a unidade i
net_i	Entrada para a unidade i
max	Máximo valor de ativação para a unidade
$decay$	Taxa de decaimento
$rest$	Nível de ativação de repouso

Por outro lado, para $net_i \leq 0$, a equação se torna:

$$\Delta y_i = (y_i - min)net_i - decay(y_i - rest), \quad \text{para } net_i \leq 0 \quad (2.5)$$

onde:

y_i	Ativação para a unidade i
net_i	Entrada para a unidade i
min	Mínimo valor de ativação para a unidade
$decay$	Taxa de decaimento
$rest$	Nível de ativação de repouso

A saída de uma unidade i (o_i) não é necessariamente idêntica ao seu valor de ativação (y_i). O valor de o_i é dado pela Equação 2.6.

$$o_i = \begin{cases} y_i, & \text{se } net_i > 0 \\ 0, & \text{se } net_i \leq 0 \end{cases} \quad (2.6)$$

onde:

y_i	Valor de ativação para a unidade i
net_i	Entrada da unidade i

Com relação às Equações 2.4 e 2.5, pode-se concluir que:

- net_i determina quando a ativação da unidade incrementa ou decrementa;

- se a ativação de uma unidade é igual ao max , então a rede acredita na hipótese completamente;
- se a ativação é igual a min , então a rede desacredita completamente na hipótese;
- se a ativação é igual a $rest$, então a rede está em um estado de repouso;
- os termos $(max - y_i)$ e $(y_i - min)$ dão certeza que a ativação da rede permanece entre min e max , evitando que ela cresça ou diminua além destes limites;
- o termo $-decay(y_i - rest)$ força a ativação a retornar ao valor de repouso na ausência de entradas externas.

Leva-se em consideração, também, no cálculo de net_i , que os estímulos externos e os estímulos excitatórios e inibitórios, vindos de unidades de dentro da rede, são ponderados por valores representados por alguns dos parâmetros apresentados pela rede. Existem vários parâmetros no modelo IAC de Rumelhart e McClelland que podem ser controlados pelo usuário:

- min : Ativação mínima. Nas simulações realizadas por Rumelhart e McClelland, o seu valor foi definido como -0,2;
- max : Ativação máxima. Seu valor ficou estabelecido como 1;
- $rest$: Nível de ativação de repouso para o qual as ativações tendem a estabelecer-se na falta de uma entrada externa. O seu valor é -0,1;
- $decay$: Taxa de decaimento, que determina a força da tendência de retorno ao nível de ativação de repouso. O seu valor nas simulações é 0,1;
- $estr$: Representa a força das entradas externas. O seu valor foi definido como 0,4;
- α : Gradua a força das entradas excitatórias, sobre as unidades, provenientes de outras unidades da rede;
- γ : Gradua a força das entradas inibitórias, sobre as unidades, provenientes de outras unidades da rede.

Poder-se-ia pensar na possibilidade de atribuir valores separados específicos a estes parâmetros para cada unidade (o modelo original IAC não permite isso, fazendo com que cada parâmetro afete toda a rede), mas este procedimento aumentaria, fortemente, a complexidade do modelo, não originando nenhum benefício conhecido.

Os valores propostos por Rumelhart e McClelland são, portanto, tomados como base em simulações realizadas, já que eles não propuseram diretrizes para a seleção de tais parâmetros. Sabe-se que max , $rest$ e min obedecem à seguinte relação: $max > 0 \geq rest \geq min$.

O gráfico da Figura 2.2 mostra a relação existente entre os parâmetros max , min , $decay$, $estr$ e $rest$.

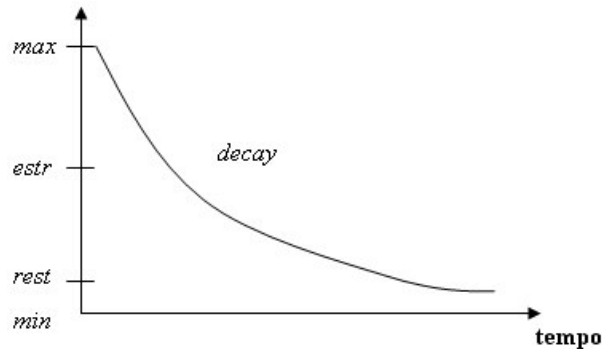


Figura 2.2: Gráfico ilustrativo da ativação de um neurônio exposta ao *decay*.

Pode-se dizer que alguns dos parâmetros citados têm significado biológico. Por exemplo, a curva de decaimento (*decay*) é uma metáfora do esquecimento dos seres humanos e o parâmetro *rest* significa que, geralmente, não se tem esquecimento total [44].

Como a rede IAC é também uma rede realimentada (com *feedback*), o tempo é discreto, pois o processamento é dividido em seqüência de passos, ou ciclos. Cada ciclo inicia-se com todas as unidades possuindo um valor de ativação, determinado no final do ciclo precedente, e os novos valores de ativação são considerados somente em um novo ciclo. Então, o processo de atualização de cada unidade é síncrono.

A Figura 2.3 apresenta uma rede IAC que contém informação sobre duas gangues rivais, os Jets e os Sharks. A rede apresenta 7 grupos: um grupo para o nome dos membros da gangue, um para a idade, um para a ocupação, um para o estado civil, um para o nível escolar, um para a gangue a qual pertencem e um que é o espelho do grupo de nomes, responsável por fazer a ligação entre os 'conceitos'. Pode-se observar que a informação dentro do grupo é mutuamente exclusiva, ou seja, se uma pessoa é casada, ela não pode ser solteira ao mesmo tempo. Entretanto, as relações existentes entre 'valores' de diferentes grupos são completamente excitatórias. Este é um ponto importante na topologia apresentada por esta rede e certamente representa uma limitação ao uso das redes IAC, pois não são muitos os problemas em que esta restrição pode ser atendida para todos os 'conceitos', ou grupos.

As conexões indicam os 'valores' de cada 'conceito' que estão relacionados entre si, para cada membro da gangue. Na Figura 2.3, estão indicadas as características relacionadas ao

nome Art. Ele está na faixa de 40 anos, é solteiro, é ladrão, é membro da gangue dos Jets e tem como escolaridade JH.

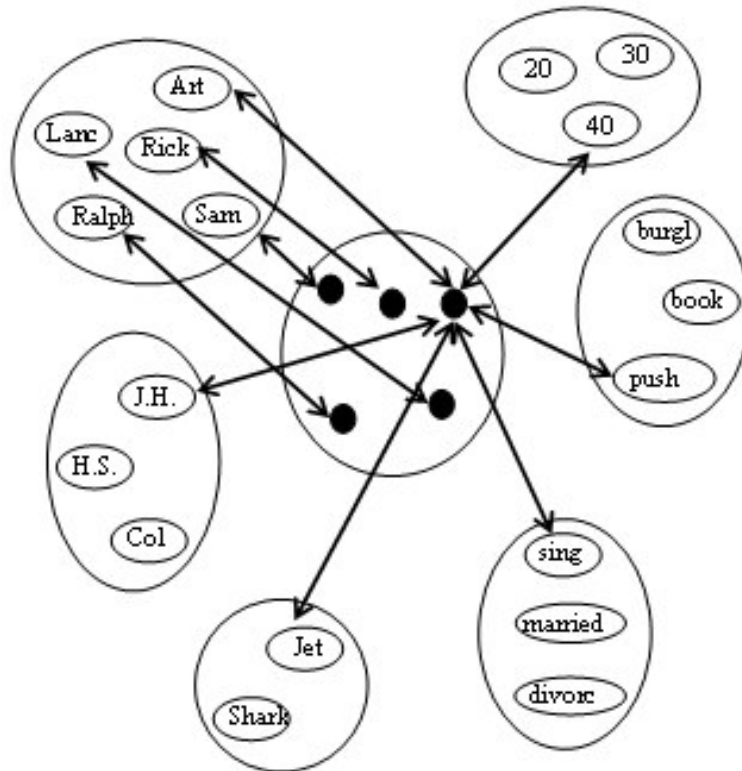


Figura 2.3: Rede IAC que modela o exemplo dos Jets & Sharks. Estão ressaltadas apenas algumas conexões.

2.2 Rede Modelo 'A'

A arquitetura sugerida por De Azevedo [22] foi inspirada no modelo IAC proposto por Rumelhart e McClelland, e foi desenvolvido de forma a solucionar as restrições de aplicabilidade prática do modelo IAC original. Um esquema é apresentado na Figura 2.4.

Neste modelo, o grupo escondido pode ou não ser utilizado, sendo esta a primeira diferença com relação ao modelo original, embora não usar o grupo escondido provoque uma melhora nas respostas, tal como descrito em [22]. A ausência do grupo escondido permite que as conexões ocorram entre quaisquer unidades de quaisquer grupos.

Outra diferença é que cada conexão pode assumir qualquer valor real no intervalo $[-1,1]$, onde valores negativos representam inibição e positivos, a excitação. O valor absoluto dos pesos representa a força da influência que existe entre dois neurônios. Dentro dos grupos, os pesos são negativos, de tal forma que os neurônios competem entre si. Entre grupos, os

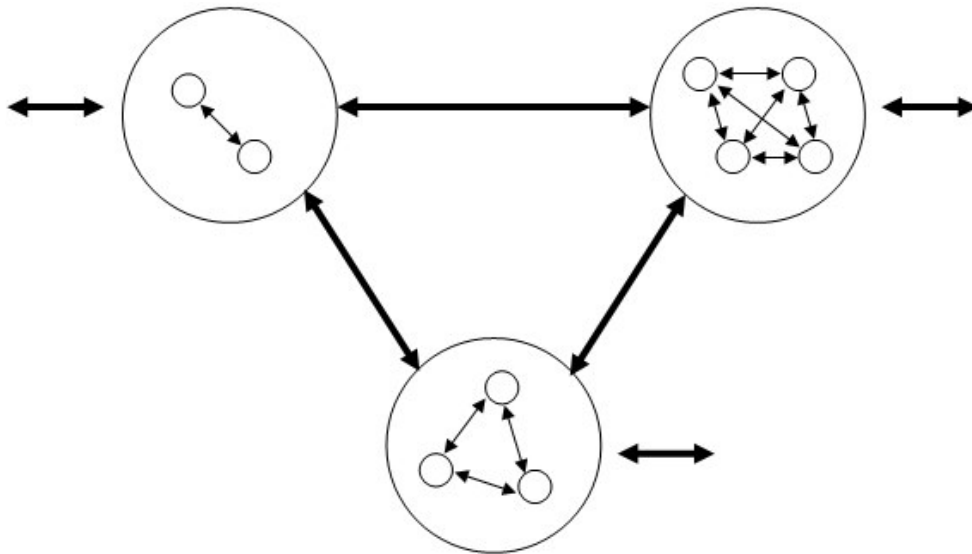


Figura 2.4: Exemplo de uma rede Modelo 'A'.

valores dos pesos dependem da força das relações entre cada 'propriedade' de cada diferente 'conceito' e podem ser tanto positivos quanto negativos. As equações e parâmetros do modelo 'A' são aqueles do modelo original de Rumelhart. O que este modelo pode perder em inspiração biológica e psicológica, características do modelo original de Rumelhart, ele ganha em utilidade. Ou seja, ele pode ser aplicado a uma imensa gama de problemas que não podem ser modelados utilizando-se a rede IAC original.

Em contrapartida, a utilização de valores reais para os pesos gera a questão de como ajustá-los para a solução de um dado problema. Primeiramente, esse ajuste foi realizado de forma similar a de um engenheiro de conhecimento, na implementação de um Sistema Especialista. Redes maiores, todavia, não se apresentam tão robustas, indicando que este procedimento não é adequado. Métodos de tentativa e erro foram utilizados em [34], implicando em muito consumo de tempo para o projetista. Ou seja, surge a necessidade do desenvolvimento de um algoritmo de treinamento que realize o ajuste adequado dos pesos sinápticos.

2.3 Aprendizado em Redes IAC

Devido às suas características intrínsecas, as redes IAC apresentam diferentes dificuldades a um projetista, em comparação com os modelos mais conhecidos, no momento da implementação da solução de um problema.

No caso das redes mais conhecidas, o projetista utiliza uma topologia pré-determinada e ajusta os pesos através de um algoritmo de treinamento, sendo esta a parte central do pro-

blema. Para exemplificar, é considerada uma rede *feedforward*. Para definir qual topologia deve ser utilizada (que forneça, ao menos, um resultado razoável) não há dificuldades, já que o número de entradas e saídas é definido pelo problema. Com relação à camada intermediária, sabe-se que uma única camada resolve a maioria dos problemas, residindo a maior dificuldade na escolha do número de neurônios desta camada. No entanto, hoje em dia, existem várias heurísticas para resolver esta questão [45]. Já quanto ao treinamento, o desenvolvimento é mais complicado.

Continuando com o exemplo do modelo *feedforward*, sabe-se que existem diversos algoritmos prontos para realizar o treinamento deste tipo de rede. Um desses algoritmos, talvez o mais conhecido hoje, é o *backpropagation* [17]. Porém, mesmo com a topologia da rede pronta e o algoritmo de aprendizado escolhido, há uma série de parâmetros a serem ajustados adequadamente como, por exemplo, a taxa de aprendizado e o termo *Momentum*. Existe também o problema do conjunto de treinamento, ou seja, deve existir um conjunto de exemplos para realizar o treinamento da rede que seja representativo do problema e em número suficiente, cuja obtenção não se constitui, necessariamente, em tarefa trivial.

Já no paradigma IAC, o centro do problema, e, certamente, sua maior dificuldade, está em se encontrar uma topologia que represente adequadamente o problema em questão, e não no ajuste de pesos, já que este passo não existe para estas redes. Em outras palavras, como não existe fase de aprendizado, a tarefa do projetista se traduz, exclusivamente, em definir uma topologia que represente o problema, ou seja:

- definir quais são os 'conceitos' envolvidos no problema, sendo que cada um deles, no momento da implementação, será representado por um grupo visível;
- definir todos os possíveis 'valores' para cada conceito e, a seguir, representar cada um destes valores através de um neurônio dentro do grupo;
- definir todas as possíveis relações existentes entre cada um dos 'valores' dos diferentes 'conceitos' e, a seguir, definir o grupo escondido através do qual todas as ligações serão realizadas;
- fazer as ligações entre todos os neurônios de todos os grupos, utilizando-se de um dos três pesos permitidos, baseado nas relações que foram definidas entre os 'valores' dos diferentes 'conceitos'.

Esta série de passos não seria um problema por demais complicado, se o modelo original de Rumelhart não apresentasse as restrições mencionadas. Exemplificando, no caso dos Jets & Sharks, ao mesmo tempo em que uma pessoa só pode ter um estado civil possível, como

casado, a relação do neurônio, que representa a pessoa no grupo escondido, com o neurônio correspondente a casado tem valor +1, pois a pessoa é ou não é casada.

Não seria, portanto, possível representar os sintomas relacionados a alguma patologia, já que uma mesma doença pode apresentar diferentes sintomas. Mais ainda, um sintoma pode ter relação maior ou menor com um determinado diagnóstico.

Como exemplo, pode-se citar o trabalho implementado por Saldías [34], que trata de Inteligências Múltiplas, para a adaptação de interfaces. Segundo a referida teoria de Howard Gardner [46], existem diferentes tipos de Inteligência (musical, lingüística, lógico-matemática, espacial, corporal-cinestésica, interpessoal, intrapessoal e ecológica), e uma mesma pessoa poderia apresentar mais de uma inteligência com diferentes graus. Este problema seria impossível de resolver utilizando-se o modelo tradicional, já que as características não são mutuamente excludentes. Por este motivo, foi utilizada o modelo 'A', que utiliza pesos reais.

Uma questão que é levantada, a partir do momento que se decide utilizar valores reais para as conexões, é se não poderiam ser utilizadas arquiteturas mais conhecidas, possuidoras de algoritmos de aprendizado bem estabelecidos, para a representação do problema. Pode-se dizer que a resposta para este questionamento é negativa, já que a rede IAC apresenta características que não são comuns a outras redes, tais como a bidirecionalidade nas conexões e organização em grupos.

Em trabalhos como o de Burton [47, 48] foram introduzidas modificações na rede IAC original, criando uma nova topologia com pesos reais para as conexões. Para treinar estas conexões, foi utilizada a regra de Hebb. Os pontos principais que se observam neste novo modelo é que ele continua usando conceitos com valores mutuamente exclusivos e as conexões, que têm pesos reais, passam a ser unidirecionais, fazendo com que o modelo perca uma das características mais interessantes do modelo original.

A partir do que foi exposto, pode-se observar que a rede criada por Rumelhart tem características bastante interessantes, mas também apresenta sérias restrições. O objetivo desta Tese é, aproveitando as características do modelo de Rumelhart e o conceito de conexões reais introduzidos nas redes Modelo 'A', desenvolver propostas de algoritmos de aprendizado, conciliando as características positivas da rede, e, ao mesmo tempo, tentando resolver as restrições que estes modelos impõem.

2.4 Propriedades das Redes IAC

A rede IAC apresenta algumas propriedades que são comuns a alguns modelos de RNA. Entre elas, pode-se citar [49]:

- Endereçabilidade por conteúdo: habilidade de acessar informações dado qualquer um dos componentes do fato ou episódio. Esta propriedade permite o acesso à informação apesar de descrições com erros. Torna-se dispensável um endereço para recuperação, em contraposição às máquinas Von Neuman, que necessitam de um endereço específico para a recuperação dos dados.
- Robustez em presença de ruído: habilidade de acessar informações, mesmo com presença de informação incompleta ou incorreta. Representa a capacidade do sistema de continuar funcionando mesmo quando danificado, operando, desta forma, apenas com dados parciais.
- Generalização: habilidade de generalizar a partir de um conjunto de exemplos. Uma RNA pode, com base em um conjunto de exemplos previamente aprendidos, estabelecer um padrão geral. Computacionalmente, esta é uma operação difícil, mas que as pessoas realizam rotineiramente.
- Associação padrão: habilidade de associar valores-padrão plausíveis, caso um fato fornecido não esteja armazenado na memória. Pode-se encontrar padrões típicos ou classes de indivíduos, embora nenhum indivíduo em especial instancie estes padrões. A propriedade final a ser examinada é a habilidade da rede IAC de fornecer padrões plausíveis quando não conhece um pedaço da informação, ou seja, associar valores padrão plausíveis para variáveis desconhecidas. O sistema de memória humana faz uso extensivo destes valores. Na rede IAC, o fornecimento de padrões plausíveis é muito próximo da generalização. Itens que são similares ao item alvo são usados para extrapolar o que a informação perdida não fornece.

Capítulo 3

Algoritmos Genéticos

A Computação Evolucionária (CE) surgiu a partir da idéia de que os mecanismos de evolução, observados na natureza, poderiam inspirar ferramentas de otimização para problemas de Engenharia e Computação [50].

Este paradigma engloba diferentes métodos, sendo que os mais difundidos são os Algoritmos Genéticos [51, 52], a Programação Evolucionária [53] e as Estratégias Evolucionárias [54]. Todos eles têm em comum o uso dos chamados 'operadores genéticos' (seleção, reprodução ou cruzamento e mutação) na evolução de uma população de soluções, tal como na evolução de populações reais, observadas na natureza [55]. No caso deste trabalho, a ferramenta de interesse são os Algoritmos Genéticos (AG), que são a técnica mais difundida, entre aquelas pertencentes a CE. Os AG foram propostos por Holland [52], e são utilizados em diversas áreas da Engenharia Elétrica, como por exemplo, Telecomunicações, Eletrônica, Sistemas Elétricos de Potência e Mecatrônica e Controle [56].

Em áreas como Engenharia e Computação, há uma série de problemas para os quais não existe uma solução satisfatória, principalmente, aqueles que envolvem otimização [57]. São problemas que apresentam um espaço de busca muito amplo, e, portanto, os mecanismos de busca tradicionais não são capazes de encontrar uma solução razoável, em um tempo razoável. Nestes casos, pode-se empregar, entre outras técnicas, os AG, que permitem que várias soluções pertencentes ao espaço de busca sejam pesquisadas simultaneamente [58]. Os AGs podem, desta forma, ser aplicados em problemas complexos de otimização: problemas com diversos parâmetros ou características que precisam ser combinadas em busca da melhor solução; problemas com muitas restrições ou condições que não podem ser representadas matematicamente; e problemas com grandes espaços de busca [59]. Além de problemas de otimização, outros que têm utilizado AGs são [50]:

- **Ecologia:** AGs têm sido usados para modelar fenômenos ecológicos;

- **Sistemas Imunes:** AGs têm sido usados para modelar vários aspectos de sistemas imunes naturais;
- **Aprendizagem de máquina:** AGs têm sido usados em muitas aplicações de aprendizado de máquina, incluindo tarefas de predição e classificação.

Uma das principais razões para o sucesso dos AGs reside na sua capacidade de explorar vários pontos do espaço de busca de forma simultânea e adaptativa. Isto permite que a informação acumulada sobre um espaço de busca desconhecido seja utilizada de maneira a induzir pesquisas subsequentes, dentro de subespaços úteis [60].

Os AGs têm uma nomenclatura baseada naquela usada pela biologia, conforme apresenta a Tabela 3.1 [50]. Um cromossomo, ou indivíduo, é composto por uma sequência de genes, sendo que em cada gene existe um valor, pertencente ao alfabeto sendo utilizado, que representa o alelo. A Figura 3.1 mostra uma possibilidade de indivíduo, utilizando um alfabeto binário $A = \{0, 1\}$.

Tabela 3.1: Nomenclatura utilizada pelos AGs.

Termo	Significado
Cromossomo	Indivíduo (string)
Gene	Cada elemento que compõe o indivíduo
Alelo	Valor de cada gene
Loco	Posição do gene no cromossomo
Genótipo	Solução candidata (i)
Fenótipo	Valor de aptidão para a solução ($f(i)$)
Geração	Ciclo

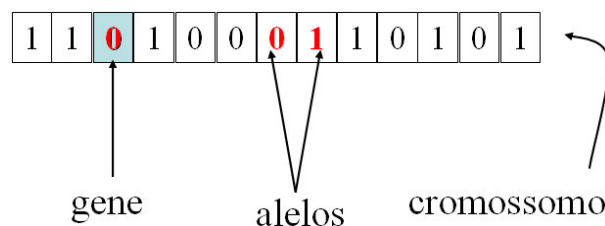


Figura 3.1: Nomenclatura associada a um AG.

Um AG simples tem os passos mostrados na Figura 3.2. A partir de um problema que se quer resolver, define-se a representação a ser utilizada. Depois disso, gera-se uma população inicial, onde cada indivíduo representa uma possível solução para o problema. Para cada indivíduo i , é calculado o seu valor de aptidão $f(i)$. Com base neste valor, realiza-se a seleção, escolhendo-se os indivíduos sobre os quais são aplicados os operadores genéticos.

Os indivíduos selecionados são cruzados dois a dois, trocando material genético entre si. E, finalmente, seleciona-se alguns indivíduos para sofrerem mutações.

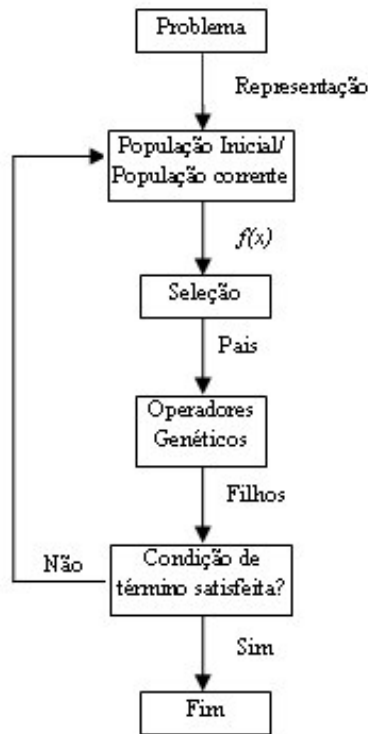


Figura 3.2: Esquema geral de um Algoritmo Genético.

Estes passos são executados, até que a população satisfaça um critério de término, que pode ser um valor de aptidão a ser alcançado, ou então, um número máximo de gerações. A partir de uma população P , composta pelos indivíduos I_1, \dots, I_{TP} , onde TP é o número de indivíduos na população, a cada ciclo do AG, é gerada uma nova população intermediária, P' , composta por indivíduos gerados pelos processos de cruzamento e mutação, e também por indivíduos vindos diretamente da população anterior. Cada um destes passos é avaliado mais detalhadamente nas próximas seções.

3.1 Representação

A representação é uma característica de extrema importância, na aplicação de AGs em problemas reais, já que estes manipulam os indivíduos, que são as variáveis codificadas, e não as variáveis propriamente ditas. Portanto, o esquema de representação pode limitar a visão que o AG tem a respeito do problema, implicando que uma boa definição é fundamental para seu desempenho [61].

Desta forma, a codificação é feita de acordo com o problema em questão, onde cada

variável deve ser discretizada e representada por um conjunto de genes, e o conjunto de variáveis codificadas forma o indivíduo [56]. Para decidir a representação a ser utilizada, deve se ter bem definido o problema a ser resolvido, ou seja, quais as variáveis que devem ser encontradas e qual a precisão necessária que estas devem apresentar.

Dependendo da natureza do problema, é necessário escolher um alfabeto para a representação, além de definir quantos genes são utilizados para representar cada variável. Com estas informações, define-se como é composto o indivíduo, que tem um tamanho TI . Na maioria das aplicações com AG, TI é mantido constante ao longo de todo o processo evolucionário.

Entre os alfabetos possíveis de serem utilizados em um AG, o binário é aquele que tem sido extensivamente utilizado nas mais diversas aplicações. Esta tem sido a representação dominante, com tamanho fixo, pois resultados teóricos mostram que são efetivas e têm representação simples [61]. Por intermédio deste alfabeto, cada gene pode assumir exclusivamente os valores 0 ou 1.

A representação binária oferece algumas vantagens, tais como simplicidade, facilidade na manipulação dos indivíduos através dos operadores genéticos, facilidade de ser transformada em inteiro ou real, e ainda, facilidade na prova de alguns teoremas. Todavia, em alguns problemas, é necessário um alfabeto de representação com mais símbolos, já que este deve ser capaz de representar todo o espaço de busca que se deseja investigar.

Apesar da representação binária ser a mais difundida, sabe-se que as características dos AGs são aproveitáveis, também, para representações não binárias. Entre outros alfabetos mais comuns, têm-se:

- **Alfabeto Inteiro:** Cada gene pode assumir qualquer valor inteiro. Algumas aplicações, que utilizam este tipo de representação, podem ser encontrados em [62, 63, 64, 65, 66, 67].

Exemplificando, em [62], é mostrado uma aplicação em classificação, onde há N padrões a serem classificados em K grupos. Na representação do indivíduo, é utilizado valores inteiros, onde o tamanho do cromossomo é igual ao número de padrões N . Assim, cada gene contém um número inteiro que varia de 1 a K , indicando o grupo k ao qual o padrão n pertence. Uma vantagem, neste tipo de representação, está relacionada ao fato de que a representação binária pode gerar valores inválidos, a menos que K seja uma potência de dois. Utilizando-se a representação inteira, este tipo de problema não ocorre.

- **Alfabeto Real:** Os AGs que utilizam esta representação são conhecidos por Algoritmos Genéticos com Codificação Real (AGCR) [61], onde cada gene pode assumir um valor

real. Alguns estudos apontam que a representação por números reais (ponto flutuante) oferece melhor desempenho do que aquele fornecido pela binária [59].

Os AGCR são utilizados, entre outros, em problemas de otimização com variáveis contínuas. Os estudos com este tipo de representação iniciaram em 1991, e já existem diversas aplicações que a utilizam [61, 68, 69, 70, 71, 72].

Existem, também, outros alfabetos sendo utilizados, como o alfabeto com símbolos [73], entre outros.

As definições feitas para a codificação são utilizadas para realizar a decodificação do cromossomo, quando se realiza a sua transformação na solução real para o problema [59]. Em alguns casos, cada gene já representa uma variável, não sendo necessário realizar nenhuma transformação. Entretanto, quando um conjunto de genes representa uma única variável, o que é comum quando se usa alfabeto binário, este conjunto deve ser transformado para um número inteiro ou real, de acordo com o especificado pela representação. Esta transformação é dependente do alfabeto que se utiliza, e da forma real como as variáveis devem ser apresentadas.

3.2 Inicialização da População

Após ser feita a definição de como é composto o indivíduo, é necessário gerar uma população inicial P_0 , que pode ser obtida aleatoriamente. É interessante que esta população tenha indivíduos com diferentes graus de aptidão. Todavia, o usuário, com base em seus conhecimentos prévios, pode estabelecer limites, fazendo com que os valores gerados fiquem mais próximos da solução desejada [45].

Em problemas de otimização de larga-escala, por exemplo, a população inicial pode incorporar conhecimento *a priori* sobre soluções. Quando se conhece bem o problema e sabe-se em que intervalo pode estar contida a resposta, a população pode ser gerada diretamente dentro deste intervalo e os elementos podem ser distribuídos uniformemente, fazendo com que a convergência seja mais rápida. Este procedimento não deve restringir drasticamente a diversidade da população, ou então, a convergência prematura pode ocorrer [74].

O número de elementos na população constitui um dos parâmetros do AG, não existindo uma regra para defini-lo. Um parâmetro que pode ser utilizado é o tamanho do cromossomo, sendo que quanto maior este tamanho, maior deve ser o tamanho da população, para que esta seja representativa do conjunto de soluções possíveis.

Um número maior de elementos da população fornece uma cobertura representativa do domínio do problema, e previne convergências prematuras para soluções locais, ao invés de globais. Portanto, aumenta a possibilidade da solução ótima estar entre estes elementos e a possibilidade de convergência. No entanto, é necessária uma maior capacidade de processamento, ou, então, um tempo maior para que o algoritmo busque a solução.

Entretanto, um número menor de indivíduos, apesar de exigir menos recursos computacionais, faz com que a probabilidade de convergência seja reduzida, resultando em uma baixa de desempenho, já que a população pode fornecer uma pequena cobertura do espaço de busca do problema, com uma distribuição ruim das soluções. Na maioria das aplicações, o tamanho da população é mantido constante ao longo das gerações.

3.3 Avaliação da População

Um elemento de extrema importância para o AG é a função de *fitness*, que é responsável por atribuir, a cada indivíduo da população, um valor que fornece uma medida do quanto este é apto para resolver o problema. Isto é, dado um indivíduo particular, que representa uma solução, a função de *fitness* retorna uma aptidão numérica simples, que deve ser proporcional à utilidade ou adaptação da solução que aquele indivíduo representa [50]. O cálculo do *fitness* é um ponto crítico para o algoritmo, já que, em última análise, é esta função que está sendo otimizada [56]. Por exemplo, quando se quer encontrar o valor que maximiza uma função, o próprio valor da função para aquele indivíduo pode ser considerado o seu *fitness* [75].

Esta função, para o AG, funciona como uma caixa preta, já que a forma como ela é gerada não importa, sendo utilizado apenas o valor final que ela produz. A função é dependente do problema e deve incorporar as restrições existentes.

Normalmente, ao início do AG, os indivíduos se localizam espalhados no espaço de busca. Por meio da função de *fitness*, à medida que a pesquisa evolui, valores particulares para cada gene iniciam a dominar a população. Assim, a variação na faixa de *fitness* diminui, à medida que a população converge.

Alguns problemas que podem resultar de uma função de *fitness* ruim, ou então, de uma má distribuição no espaço de busca, são a convergência prematura e o término lento (*slow finishing*) [74].

CONVERGÊNCIA PREMATURA

Ocorre quando genes de alguns poucos indivíduos, com alto valor de *fitness*, comparativamente, podem rapidamente vir a dominar a população, causando a convergência para um

máximo local. Uma vez que a população tenha convergido, a habilidade do AG de continuar pesquisando por melhores soluções fica bem restrita. Isto pode ocorrer devido à população reduzida ou à má distribuição da população em torno do ponto subótimo. Ou seja, um indivíduo próximo de um ótimo local possui um valor de adaptação superior aos demais indivíduos da população. Com isso, o processo de seleção fará com que este indivíduo tenha grandes chances de dominar a próxima geração e assim sucessivamente, se não aparecerem outros indivíduos com melhores valores de adaptação [45].

TÉRMINO LENTO

É o problema oposto à convergência prematura. Depois de muitas gerações, a população quase convergiu, mas é ainda possível que o máximo global não tenha sido encontrado. O *fitness* médio é alto, e a diferença entre o melhor e o indivíduo médio é pequena. Portanto, não há variância suficiente nos valores das funções de *fitness*, para localizar o máximo.

3.4 Seleção

Mais que os outros operadores, a seleção é responsável por determinar as características de convergência do AG. De um lado, tem por objetivo guiar a população em busca dos melhores indivíduos, e por outro, manter a alta diversidade genotípica da população. É através da seleção que se escolhe quais indivíduos irão participar da geração de uma nova população, por meio do *fitness* calculado para cada elemento. Indivíduos com maior grau de *fitness* têm maiores chances de participar desta nova população, enquanto indivíduos menos adaptados têm mais chances de desaparecerem [75].

Para cada indivíduo i pertencente à P , a probabilidade $P_s(i)$ de incluir a cópia de tal indivíduo dentro de P' é calculada [61]. Um método bastante conhecido para realizar esta seleção é o método da roleta ponderada, que considera uma roleta em que cada casa representa um indivíduo. O tamanho desta casa é proporcional ao seu valor de *fitness*, ou seja, indivíduos com maior valor de aptidão possuem uma casa maior, tendo, portanto, maiores chances de serem escolhidos [45]. Neste caso, $P_s(i)$, que indica a probabilidade do indivíduo i pertencer a população P' , é calculada de acordo com a Equação 3.1 [61].

$$P_s(i) = \frac{f(i)}{\sum_{j=1}^{TP} f(j)} \quad (3.1)$$

onde:

$f(i)$ Valor de aptidão do indivíduo i
 TP Número de indivíduos da população

Assim, indivíduos com aptidão acima da média tendem a receber mais cópias que aquelas com aptidão abaixo da média. A Figura 3.3 mostra um exemplo de aplicação da roleta ponderada. O indivíduo I_1 tem valor de aptidão de 2,23. Neste caso, a sua aptidão relativa, ou normalizada, é 0,14. Já o indivíduo I_2 tem aptidão relativa de 0,47, ou seja, ele tem maior probabilidade de ser escolhido para fazer parte da próxima geração.

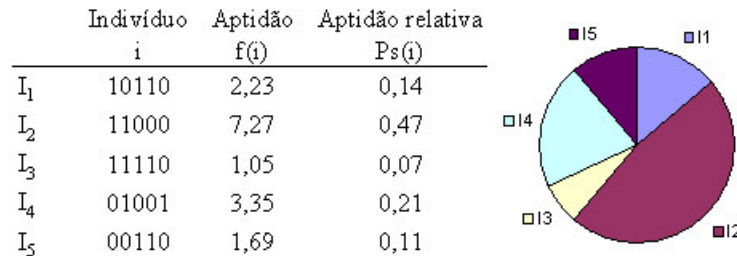


Figura 3.3: Exemplo de aplicação de roleta ponderada.

Existem outros métodos para realizar a seleção, tais como a seleção elitista e a seleção proporcional [57, 76]. Existe também a seleção por classificação (*rank selection*) [77]. Neste método, classifica-se a população em ordem crescente de aptidão e, em seguida, cada indivíduo recebe um valor de acordo com esta classificação. O pior assume valor 1, o segundo pior assume valor 2, e assim sucessivamente. O melhor cromossomo assume valor TP , que corresponde ao número de indivíduos da população. A probabilidade de um indivíduo ser selecionado para a etapa seguinte de cruzamento cresce em função de sua classificação. Este método pode apresentar convergência muito lenta, uma vez que os melhores indivíduos estão sempre próximos de seus concorrentes com menor aptidão.

3.5 Cruzamento

O cruzamento e a mutação são os operadores genéticos aplicados aos elementos escolhidos pelo método de seleção. Juntamente com a seleção, esta etapa de manipulação é que dá origem a uma nova população, que pode, ou não, ser mais apta que a anterior.

O cruzamento ocorre entre dois indivíduos diferentes, que trocam entre si fragmentos de cromossomos, e está exemplificado na Figura 3.4. Neste figura, é definido um ponto de corte que pode ou não ser aleatório e, neste ponto, os fragmentos de cromossomo são trocados. Este é um exemplo básico de como pode ocorrer o cruzamento.

Existem diversas formas de cruzamento, além da que envolve um único ponto. Em [78], pode-se encontrar referências para estudos realizados com relação ao cruzamento, onde se investiga a efetividade do cruzamento de múltiplos pontos, concluindo que cruzamento de

					P1						
Pai 1	1	0	1	1	1	0	1	0	1	1	
Pai 2	0	1	1	1	0	0	1	0	1	0	

Filho 1	1	0	1	1	0	0	1	0	1	0	
Filho 2	0	1	1	1	1	0	1	0	1	1	

Figura 3.4: Exemplo de aplicação do operador de cruzamento, realizado em um único ponto.

dois pontos dá um incremento na efetividade, mas que mais pontos além disso reduz a performance do AG.

No cruzamento de dois pontos, pode-se considerar o indivíduo circular ou não. A Figura 3.5 traz dois indivíduos pais, e mostra, também, os filhos resultantes, quando usando o cruzamento de dois pontos, sem considerar o indivíduo circular e considerando o indivíduo circular.

					P1						
Pai 1	1	0	1	1	1	0	1	0	1	1	
Pai 2	0	1	1	1	0	0	1	0	1	0	

Cruzamento de dois pontos, sem considerar o indivíduo circular:

					P1						
Filho 1	1	0	1	1	0	0	1	0	1	1	
Filho 2	0	1	1	1	1	0	1	0	1	0	

Cruzamento de dois pontos, considerando o indivíduo circular:

					P1						
Filho 1	1	1	1	0	1	0	1	0	0	1	
Filho 2	1	1	0	0	1	0	1	1	1	0	

Figura 3.5: Exemplos de cruzamentos de dois pontos, considerando ou não o indivíduo circular.

Outra técnica de cruzamento é o cruzamento uniforme. Neste tipo de cruzamento, que difere bastante do cruzamento de um ponto, uma máscara de cruzamento é gerada de forma aleatória. Esta é composta de 0's e 1's, e tem o mesmo tamanho do indivíduo. Para formar o primeiro filho, onde existe um 1 na máscara de cruzamento, o gene é copiado do primeiro pai, e onde existe 0 na máscara, o gene é copiado do segundo pai. No caso do segundo filho, os pais são trocados e o mesmo processo é realizado. Uma nova máscara é gerada aleatoriamente para cada par de pais. O número de pontos de cruzamento efetivos não é fixo, mas fica na média de $TI/2$, onde TI é o tamanho do indivíduo [78].

3.6 Mutação

A mutação corresponde a escolher um gene do cromossomo e mudar o seu valor para um outro possível, pertencente ao alfabeto sendo considerado. No caso binário, o valor pode ser mutado para 1, caso seja 0, e vice-versa. A mutação está exemplificada na Figura 3.6, onde o gene a ser modificado é escolhido aleatoriamente. A chance de ocorrer mutação é dependente de uma taxa de mutação TM , definida ao início da execução do AG.



Figura 3.6: Exemplo de aplicação do operador de mutação, quando utilizando alfabeto binário.

A mutação é um operador importante já que [55]:

- É necessária para a introdução e manutenção da diversidade genética da população;
- Fornece meios para introdução de novos elementos na população;
- Assegura que a probabilidade de se chegar a um ponto do espaço de busca nunca será zero;
- Contorna o problema de mínimos locais, pois altera levemente a direção da busca.

3.7 Parâmetros

No início do AG, são definidos os parâmetros que indicam em que probabilidade ocorrem o cruzamento e a mutação, que são a taxa de cruzamento TC , e a taxa de mutação TM . Espera-se que a probabilidade de ocorrer cruzamento seja bem maior que a probabilidade de ocorrer mutação, assim como na natureza. Quando se define uma probabilidade de mutação muito alta, o método se transforma em uma busca aleatória, e com uma taxa baixa, previne-se que uma dada posição fique estagnada em um valor, além de possibilitar que se chegue em qualquer ponto do espaço de busca.

No caso da probabilidade de cruzamento muito baixa, pode acontecer de demorar muito tempo para se encontrar o valor ótimo. E com uma taxa maior, mais rapidamente novas estruturas serão introduzidas na população, mas estruturas com boas aptidões poderão ser retiradas mais rapidamente. Deve ser encontrado um equilíbrio entre estes dois parâmetros, de maneira a garantir um bom desempenho para o AG.

3.8 Condições de Término

O ideal seria que, assim que se encontrasse o ótimo global, o algoritmo encerrasse a execução. Isto, porém, não ocorre. Pode-se usar como critério de parada o número de gerações ou um tempo limite de processamento, ou seja, mesmo que na população inicial se encontre o valor que maximize a função, ainda assim o AG não finalizará o processo de busca. Normalmente, costuma-se memorizar o indivíduo mais adaptado, independentemente deste fazer parte da população atual ou não. Ao final, este será o resultado esperado [75].

3.9 Características

O algoritmo apresentado é uma forma simples de AG, sendo que existem muitas variações, como o algoritmo conhecido por 'Algoritmo Genético de Vasconcelos' [36]. Existem, também, estudos com AGs paralelos [79, 80, 81, 82].

Os operadores genéticos citados são os mais populares. Existem, porém outros operadores, e entre eles pode-se citar a inversão, epístase, entre outros [78, 83, 84, 85].

Os AGs são facilmente adaptáveis a inúmeras classes de problemas, são robustos e são fáceis de hibridizar com outras técnicas. Além disto, exploram satisfatoriamente espaços de busca de grande dimensionalidade, assim como manipulam com certa facilidade um grande número de restrições [56].

O processo criativo e inovador implementado por AGs é facilmente compreensível: a partir de um conjunto de possíveis soluções para um problema, seleciona-se as melhores, aproveitando as suas características de interesse, recombina-as, adaptando-as e gerando-se novas soluções [56].

Em relação aos métodos de busca convencionais, os AG diferem nos seguintes pontos [55]:

- O risco de se encontrar uma solução que seja um mínimo local é reduzida, já que a busca é feita sobre um conjunto de pontos e não apenas um único;
- O único conhecimento necessário é a função custo do problema, já que os AG realizam uma busca cega;
- São utilizados operadores estocásticos e não regras determinísticas para guiar uma busca altamente exploratória e estruturada, onde as informações das gerações anteriores são utilizadas para direcionar a busca.

De forma resumida, algumas das razões para o sucesso dos AGs são:

- AGs podem resolver problemas difíceis de forma rápida e confiável;
- AGs são fáceis de adaptar para modelos e simulações existentes;
- AGs são fáceis de hibridizar;
- AGs podem não garantir uma solução ótima, mas garantem boas soluções de forma rápida.

Além disso, os AGs vêm sendo utilizados com sucesso no treinamento de diversos tipos de redes, fazendo surgir a hipótese de que poderia ser usado, também, para encontrar os pesos de uma rede modelo 'A'.

Capítulo 4

Estudos de Caso

Para a realização dos testes, são utilizados três problemas: Jets & Sharks, Diagnóstico Diferencial de Hepatite e Diagnóstico Diferencial em Reumatologia. O conhecimento sobre eles foi delimitado, e redes modelo 'A' foram utilizadas para representá-los. No caso dos Jets & Sharks, o problema foi implementado utilizando, também, uma rede IAC original. A partir dos modelos completos, foram criadas redes menores, variando tanto o número de grupos, quanto os neurônios utilizados. Isto foi feito para permitir um melhor acompanhamento dos resultados fornecidos pela metodologia sendo implementada.

4.1 O Exemplo dos Jets & Sharks

O primeiro exemplo escolhido para a realização dos testes foi o problema dos Jets & Sharks. É um *toy problem* utilizado em [42], para ilustrar o funcionamento das redes IAC, e traz informações gerais sobre indivíduos pertencentes a duas gangues rivais. Todos os 'conceitos' envolvidos no problema têm 'valores' mutuamente exclusivos, e as relações entre diferentes 'valores' de diferentes 'conceitos', quando existem, são totais. Estas características tornam este exemplo ideal para ser implementado pela rede IAC, que é considerada como um padrão-ouro para avaliação das propostas de treinamento para redes modelo 'A'. Assim, se espera que uma rede modelo 'A', que teve seus pesos encontrados por meio das propostas apresentadas, forneça resultados tão bons para este problema quanto aqueles fornecidos pela implementação usando a rede original de Rumelhart.

A Tabela 4.1 traz a informação a respeito dos 27 indivíduos pertencentes às duas gangues. Para cada indivíduo são informados o nome, a gangue a qual pertencem, a faixa etária, a escolaridade, o estado civil e a profissão exercida. Por exemplo, o indivíduo com nome

Art, pertence à gangue dos *Jets*, está na faixa etária dos 40 anos, tem ensino fundamental completo, é solteiro e é assaltante.

Tabela 4.1: Conhecimento sobre o problema dos Jets & Sharks.

Nomes	Gangue	Faixa Etária	Escolaridade	Estado Civil	Profissão
Art	Jets	40	JH	Single	Pusher
Al	Jets	30	JH	Married	Burglar
Sam	Jets	20	College	Single	Bookie
Clyde	Jets	40	JH	Single	Bookie
Mike	Jets	30	JH	Single	Bookie
Jim	Jets	20	JH	Divorced	Burglar
Greg	Jets	20	HS	Married	Pusher
John	Jets	20	JH	Married	Burglar
Doug	Jets	30	HS	Single	Bookie
Lance	Jets	20	JH	Married	Burglar
George	Jets	20	JH	Divorced	Burglar
Pete	Jets	20	HS	Single	Bookie
Fred	Jets	20	HS	Single	Pusher
Gene	Jets	20	College	Single	Pusher
Ralph	Jets	30	JH	Single	Pusher
Phil	Sharks	30	College	Married	Pusher
Ike	Sharks	30	JH	Single	Bookie
Nick	Sharks	30	HS	Single	Pusher
Don	Sharks	30	College	Married	Burglar
Ned	Sharks	30	College	Married	Bookie
Karl	Sharks	40	HS	Married	Bookie
Ken	Sharks	20	HS	Single	Burglar
Earl	Sharks	40	HS	Married	Burglar
Rick	Sharks	30	HS	Divorced	Burglar
Ol	Sharks	30	College	Married	Pusher
Neal	Sharks	30	HS	Single	Bookie
Dave	Sharks	30	HS	Divorced	Pusher

Para representar o conhecimento completo do problema, através de uma rede modelo 'A', são necessários 41 neurônios organizados em 6 grupos:

- o grupo 'nomes' contém 27 neurônios, correspondentes aos nomes dos indivíduos das duas gangues;
- o grupo 'gangue' contém 2 neurônios, correspondentes às gangues as quais os indivíduos podem pertencer (Jets e Sharks);
- o grupo 'faixa etária' contém 3 neurônios, correspondentes às possibilidades de faixa etária (20, 30, 40);

- o grupo 'escolaridade' contém 3 neurônios, correspondentes aos níveis de escolaridade possíveis (JH, HS, *College*);
- o grupo 'estado civil' contém 3 neurônios, correspondentes às possibilidades de estado civil (*single, married e divorced*).
- o grupo 'profissão' contém 3 neurônios, correspondentes às atividades exercidas dentro da gangue (*pusher, burglar e bookie*);

Quando se utiliza a rede IAC original, é necessário definir, também, qual é o grupo que será espelho para o grupo escondido. Neste caso, o grupo 'nomes' é utilizado, já que contém o maior número de neurônios. É através dele que são feitas as conexões entre todos os outros grupos. Quando se utiliza a rede modelo 'A', o grupo escondido torna-se desnecessário.

Por meio de uma rede IAC, tanto original, quanto modelo 'A', pode-se recuperar informação geral e específica a respeito de um problema. Se a rede é consultada para recuperar a informação a respeito de um indivíduo, diz-se que se quer recuperar informação específica a respeito de uma entrada. Isto ocorre, por exemplo, quando se quer acessar a informação a respeito do indivíduo com nome Art. O neurônio pertencente ao grupo 'nomes' referente a Art é ativado, processa-se a rede e recupera-se sua faixa etária, profissão, gangue, estado civil e escolaridade. Neste caso, espera-se que apenas um neurônio de cada grupo esteja ativo, já que se trata de um problema com características mutuamente exclusivas.

Contudo, quando a rede é consultada para se saber quem são os indivíduos que possuem determinada característica, ou quais outras características estão associadas a ela, diz-se que a rede deve generalizar uma determinada informação. Isto ocorre, por exemplo, quando se quer saber quem são os membros da gangue Jets, e quais características estão mais presentes nesta gangue. O neurônio do grupo 'gangue', correspondente a Jets, é ativado e as informações associadas a este neurônio são recuperadas. Neste caso, pode ocorrer de mais de um neurônio por grupo estar ativo, como no caso do grupo 'nomes', onde espera-se que todos os indivíduos pertencentes a gangue escolhida estejam ativos. Assim, são utilizados os termos 'entradas específicas' e 'entradas gerais' para referenciar as situações acima descritas.

Foram elaborados, também, exemplos resumidos, baseados no completo, contendo apenas parte da informação. Um dos exemplos traz o nome, a gangue e a ocupação sobre cinco indivíduos, conforme apresentado na Tabela 4.2. Para representar este conhecimento são necessários 10 neurônios, organizados em 3 grupos. Na Tabela 4.3, é mostrada parte da matriz de conectividade para este exemplo utilizando uma rede IAC original. São mostradas apenas as conexões dos grupos visíveis com o grupo escondido, já que as demais conexões assumem valor 0. Sabe-se também que as conexões intragrupos assumem valor -1 entre neurônios diferentes e valor 0 para conexões de um neurônio para ele mesmo.

Tabela 4.2: Problema dos Jets & Sharks, considerando apenas cinco indivíduos e três 'conceitos'.

Nome	Gangue	Profissão
Art	Jets	Pusher
Rick	Sharks	Burglar
Sam	Jets	Bookie
Ralph	Jets	Pusher
Lance	Jets	Burglar

Tabela 4.3: Parte da matriz de conectividade para uma rede que representa o problema dos Jets & Sharks com 10 neurônios, organizados em três grupos.

	_Art	_Rick	_Sam	_Ralph	_Lance
Art	1	0	0	0	0
Rick	0	1	0	0	0
Sam	0	0	1	0	0
Ralph	0	0	0	1	0
Lance	0	0	0	0	1
Jets	1	0	1	1	1
Sharks	0	1	0	0	0
Pusher	1	0	0	1	0
Burglar	0	1	0	0	1
Bookie	0	0	1	0	0

Durante a realização dos testes, foi utilizada, também, uma variação, contendo 21 neurônios, organizados em 6 grupos, que traz as informações a respeito de 10 indivíduos selecionados entre os 27 iniciais.

4.2 Diagnóstico Diferencial de Hepatite

Este exemplo aborda o diagnóstico diferencial de Hepatite, entre Hepatite aguda por vírus A e Hepatite aguda por vírus B. É um problema de auxílio à decisão na área médica, que foi, originalmente, desenvolvido como um SE baseado em regras [86], utilizando o shell *Intellec System* [87].

Traz, de uma forma simplificada, alguns sintomas e exames laboratoriais necessários a este diagnóstico conforme dados extraídos (com a concordância e colaboração do autor) de [86].

No trabalho original de Ramos [86], são enumeradas as informações necessárias para a realização do diagnóstico, que incluem informações gerais sobre o paciente, sintomas, exame físico e exames laboratoriais. Apenas parte destas informações foi utilizada na implementação deste exemplo, sendo esta escolha feita com o auxílio do referido autor.

Entre os sintomas selecionados, tem-se: náuseas, vômitos, dor abdominal, icterícia, emagrecimento, febre, astenia e anorexia. Entre as informações gerais a respeito do paciente, foram utilizados a faixa etária e a presença de fatores de risco. O exame físico detecta a presença de hepatomegalia. E, por fim, os exames laboratoriais escolhidos foram:

- ALT, que detecta a elevação da enzima hepática. Este exame apenas indica a possibilidade do paciente ter Hepatite, mas não o tipo;
- Anti-HVA IgM, que indica infecção pelo vírus da Hepatite A;
- Anti-HBC IgM, que é marcador de infecção recente de vírus de Hepatite B, encontrado até 6 meses após a infecção. Pode aparecer também em infecções crônicas;
- HBsAg, que é o primeiro marcador que aparece em infecções por vírus da Hepatite B.

Este problema, ao contrário dos JS, contém 'conceitos' cujos 'valores' não são mutuamente exclusivos, ou seja, um valor não impede a existência de outro. As relações entre os diferentes 'valores' dos 'conceitos' apresentam diferentes graus, podendo ser mais forte ou fracamente relacionadas. Como exemplo, um sintoma pode ter uma influência maior ou menor em um determinado diagnóstico. Além disso, neste problema, dada a natureza dos 'conceitos' envolvidos, fica difícil conhecer todas as relações entre todos os neurônios. Para a montagem da rede, foi necessário utilizar 44 neurônios organizados em 16 grupos, de acordo com os 'conceitos' estabelecidos na Tabela 4.4.

Tabela 4.4: Organização dos 'conceitos' para o exemplo de diagnóstico de Hepatite.

Conceitos	Valores
Diagnóstico	Hepatite Aguda por Vírus A
	Hepatite Aguda por Vírus B
Náuseas	Sim
	Não
Vômitos	Sim
	Não
Dor Abdominal	Não
	Leve a Moderada
	Intensa
Icterícia	Sim
	Não

Tabela 4.4: (continuação)

Conceitos	Valores
Emagrecimento	Não
	Leve a Moderado
	Acentuada
Febre	Alta
	Média
	Baixa
	Não apresenta
Faixa etária	Idoso
	Adulto
	Adolescente
	Criança
Fatores de risco	Transfusão
	Promiscuidade
	Piercing
	Tatuagem
	Homossexualidade masculina
	Drogas Injetáveis
Astenia	Sim
	Não
Anorexia	Sim
	Não
Exame Físico	Hepatomegalia
	Não apresenta sintomas
ALT	Negativo
	Aumentada de 2 a 4 vezes
	Aumentada de 4 a 10 vezes
	Aumentada de mais de 10 vezes
Anti HVA IgM	Positivo
	Negativo
HBsAg	Positivo
	Negativo
Anti HBc IgM	Positivo
	Negativo

Os sintomas, as informações a respeito do paciente, e o exame físico, fornecem uma indicação de qual a hepatite que está presente, já que algumas destas características são mais comuns a um diagnóstico ou outro. Por exemplo, fatores de risco, como homossexualismo e uso de agulhas injetáveis, estão associados a Hepatite Aguda por Vírus B. Alguns valores de faixa etária também podem estar mais associados a um diagnóstico ou outro. Por exemplo, é mais comum crianças terem Hepatite A. O exame de transaminase (ALT) é inespecífico, e indica somente se o fígado está afetado.

O médico utiliza estas informações para decidir sobre quais exames devem ser solicitados para o paciente em questão. No caso do resultado ser negativo, e se houver a permanência dos sintomas, então o médico solicita a realização de algum outro exame. No caso específico do problema de Hepatite, um sistema que auxilie na decisão de quais exames devem ser realizados é de grande benefício, já que o custo destes exames é bastante elevado.

Se o exame Anti-HVA IgM é positivo, então o indivíduo apresenta Hepatite A. Entretanto, se o exame HBsAg é positivo, ou o exame Anti-HBc IgM é positivo, então, o indivíduo apresenta Hepatite B. Já se o exame Anti-HVA IgM é negativo, o indivíduo não tem Hepatite A. Da mesma forma, se ambos os exames Anti-HBc IgM e HBsAg derem negativo, o indivíduo não tem Hepatite B.

Foram utilizados, também, um modelo com 13 neurônios organizados em 6 grupos (apresentado na Tabela 4.5), um modelo com 14 neurônios, organizados em 6 grupos (apresentado na Tabela 4.6) e um modelo com 17 neurônios, organizado em 7 grupos (apresentado na Tabela 4.7).

Tabela 4.5: Organização dos 'conceitos' para o exemplo de diagnóstico de Hepatite, com 13 neurônios.

Conceitos	Valores
Diagnóstico	Hepatite Aguda por Vírus A
	Hepatite Aguda por Vírus B
Vômitos	Sim
	Não
Febre	Alta
	Média
	Baixa
Anti HVA IgM	Positivo
	Negativo
HBsAg	Positivo
	Negativo
Anti HBc IgM	Positivo
	Negativo

Tabela 4.6: Organização dos 'conceitos' para o exemplo de diagnóstico de Hepatite, com 14 neurônios.

Conceitos	Valores
Diagnóstico	Hepatite Aguda por Vírus A
	Hepatite Aguda por Vírus B
vômitos	Sim
	Não
Fatores de Risco	Tatuagem
	Piercing
	Homossexualismo
	Drogas injetáveis
Anti HVA IgM	Positivo
	Negativo
HBsAg	Positivo
	Negativo
Anti HBc IgM	Positivo
	Negativo

Tabela 4.7: Organização dos 'conceitos' para o exemplo de diagnóstico de Hepatite, com 17 neurônios.

Conceitos	Valores
Diagnóstico	Hepatite Aguda por Vírus A
	Hepatite Aguda por Vírus B
Vômitos	Sim
	Não
Febre	Alta
	Média
	Baixa
Fatores de Risco	Tatuagem
	Piercing
	Homossexualismo
	Drogas Injetáveis
Anti HVA IgM	Positivo
	Negativo
HBsAg	Positivo
	Negativo
Anti HBc IgM	Positivo
	Negativo

4.3 Diagnóstico Diferencial em Reumatologia

Um outro exemplo utilizado para testes é aquele encontrado em [22], que trata do diagnóstico diferencial em reumatologia. Existem duas versões, implementadas por meio de redes modelo 'A':

- Diagnóstico diferencial entre duas possibilidades possíveis, que são Artrite Reumatóide (AR) e Lupus Eritematoso Sistêmico (LES), composto por 29 neurônios;
- Diagnóstico diferencial entre quatro possibilidades possíveis, que são AR, LES, Artrite Psoriática (AP) e Artrite de Gota (AGt), composto por 50 neurônios.

Para construir as matrizes de conectividade das redes, que representam as duas possibilidades mencionadas, De Azevedo [22] foi auxiliado por um especialista de domínio. Este definiu cada peso de conexão entre dois neurônios, de acordo com seu conhecimento a respeito das relações existentes entre os 'valores' dos 'conceitos', representados pelos neurônios em questão.

4.3.1 Diagnóstico Diferencial em Reumatologia: Duas Possibilidades

Para realizar o diagnóstico diferencial entre Artrite Reumatóide e Lupus Eritematoso Sistêmico, os grupos foram construídos seguindo a abordagem proposta por De Azevedo, para ser possível realizar, posteriormente, a comparação dos resultados, de acordo com a Figura 4.1.

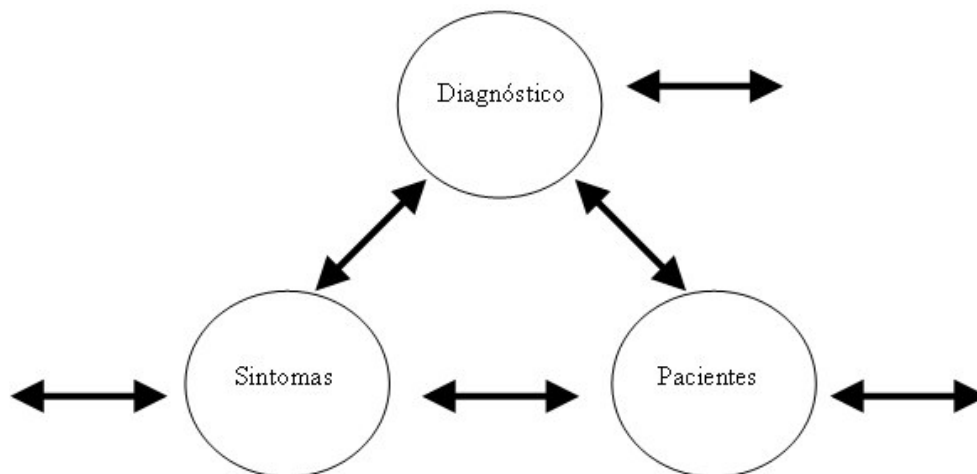


Figura 4.1: Esquema da rede modelo 'A' que implementa o Diagnóstico Diferencial em Reumatologia, com duas possibilidades.

Para representar o conhecimento envolvido no problema, são utilizados 3 grupos:

- Grupo 'diagnóstico': contém dois neurônios referentes às duas possibilidades de diagnóstico diferencial, que são Artrite Reumatóide e Lupus Eritematoso Sistêmico

- Grupo 'sintomas': contém 14 neurônios referentes aos sintomas relacionados às duas possibilidades de diagnóstico. Os sintomas considerados são: febre, artralgia, artrite, rigidez matinal, mialgia, nódulos subcutâneos, manchas borboleta, fenômeno de Raynaud, fotossensibilidade, alopecia, manifestações renais, manifestações do Sistema Nervoso Central (SNC), manifestações pulmonares e mão reumatóide.
- Grupo 'pacientes': contém 13 neurônios, sendo que 12 correspondem a pacientes já diagnosticados, e aquele restante é o paciente que está sendo consultado.

Seguindo a topologia de rede modelo 'A', todos os grupos estão interconectados, e todos os neurônios de cada grupo podem receber estímulos externos. A matriz de conectividade tem a forma geral apresentada na Tabela 4.8.

Tabela 4.8: Forma geral da matriz de conectividade para a rede de 29 neurônios.

	Diagnóstico	Sintomas	Pacientes
Diagnóstico	D/D	D/S^I	D/P^I
Sintomas	D/S	S/S	S/P^I
Pacientes	D/P	S/P	P/P

Para exemplificar a questão do grupo que representa um 'conceito' com 'valores' não mutuamente exclusivos, é considerado o grupo 'diagnóstico'. Este contém dois neurônios, cujo valor de conexão é igual a -0,9. Este valor indica que existe a possibilidade, embora pequena, de uma mesma pessoa apresentar as duas doenças. Assim, fica inviável utilizar o valor -1, que indica inibição completa. Assim, a submatriz de conectividade, que representa as conexões entre os neurônios pertencentes ao grupo 'diagnóstico', tem a forma apresentada na Tabela 4.9.

Tabela 4.9: Matriz de conectividade entre os neurônios do grupo 'diagnóstico'.

D/D	AR	LES
AR	0	-0,9
LES	-0,9	0

No caso do grupo 'sintomas', as conexões entre seus neurônios assumem valor 0, já que a presença de um sintoma não inibe nem estimula a presença de outro. Assim, a submatriz S/S é uma matriz nula.

Entre os neurônios do grupo 'pacientes' há um valor de inibição de -1, pois ao ativar um paciente os outros devem ser inativados, já que cada paciente é um caso isolado. Assim, os pesos, pertencentes a submatriz de conectividade P/P , assumem valor -1, entre dois neurônios diferentes, e valor 0, de um neurônio para ele mesmo.

Os valores de conexão entre os neurônios do grupo 'pacientes' e 'sintomas' foi feita utilizando a Engenharia do Conhecimento, por meio de conversas com o especialista de domínio. Foram extraídas as relações listadas abaixo:

- Sintoma A - Febre: Presente na AR e ausente na LES.
- Sintoma B - Artralgia: É um sintoma comum às duas doenças, mas está presente na AR com mais frequência.
- Sintoma C - Artrite: Comum às duas doenças, e está mais associada à AR.
- Sintoma D - Rigidez Matinal: Presente na AR, e ausente na LES.
- Sintoma E - Mialgia: Presente em ambas as doenças com a mesma intensidade.
- Sintoma F - Nódulos Subcutâneos: Presente na AR, e ausente na LES.
- Sintoma G - Manchas Borboleta: Presente na LES, e ausente na AR.
- Sintoma H - Fenômeno de Raynoud: Pode estar presente em ambos, mas com valores baixos, sendo um pouco mais comum para a LES.
- Sintoma I - Fotossensibilidade: Presente na LES, e ausente na AR.
- Sintoma J - Alopecia: Presente na LES e ausente na AR.
- Sintoma K - Manifestação Renal: Presente em ambas, com valores baixos, mas um pouco mais comum para LES.
- Sintoma L - Manifestação SNC: Presente em ambas, com valores baixos, mas um pouco mais comum para LES.
- Sintoma M - Manifestação Renal: Presente em ambas, com valores baixos.
- Sintoma N - Mão Reumatóide: Presente na AR, e ausente na LES.

Todas as relações acima foram transformadas em valores de conexão, como apresentado na Tabela 4.10.

Os pacientes são casos reais, e apresentam os seguintes sintomas e diagnóstico:

- Paciente 1 – **Anne**: artralgia. Diagnóstico: AR.
- Paciente 2 – **Helen**: febre, artrite, manifestações renais. Diagnóstico: AR.

Tabela 4.10: Matriz de conectividade entre os grupos 'diagnóstico' e 'sintomas'.

<i>D/S</i>	AR	LES
Febre	0,4	-0,6
Artralgia	1	0,9
Artrite	0,9	0,6
Rigidez matinal	0,7	-0,9
Mialgia	0,6	0,6
Nódulos subcutâneos	0,8	0
Manchas borboleta	-1	0,8
Fenômeno de Raynaud	0,2	0,5
Fotossensibilidade	0	0,6
Alopécia	-0,9	0,8
Manifestações renais	0,3	0,4
Manifestações do SNC	0,2	0,4
Manifestações pulmonares	0,2	0,2
Mão reumatóide	0,6	-0,9

- Paciente 3 – **Mary**: artralgia, artrite, mão reumatóide. Diagnóstico: AR.
- Paciente 4 – **Suzan**: artralgia, mialgia, nódulos subcutâneos, manifestações pulmonares. Diagnóstico: AR.
- Paciente 5 – **Lucy**: artrite, artralgia, rigidez matinal, manifestações renais, manifestações do SNC. Diagnóstico: AR.
- Paciente 6 – **Carol**: artralgia, artrite, rigidez matinal, mialgia. Diagnóstico: AR.
- Paciente 7 – **Pat**: artralgia, manchas borboleta, fenômeno de Raynaud, alopecia. Diagnóstico: LES.
- Paciente 8 – **Jackie**: artrite, fotossensibilidade e alopecia. Diagnóstico: LES.
- Paciente 9 – **Sue**: artralgia, artrite, mialgia e manchas borboleta. Diagnóstico: LES.
- Paciente 10 – **Beth**: artralgia, artrite e alopecia. Diagnóstico: LES.
- Paciente 11 – **Linda**: artralgia, artrite, nódulos subcutâneos, manifestações renais e manifestações do SNC. Diagnóstico: LES.
- Paciente 12 – **Mag**: artralgia, manchas borboletas. Diagnóstico: LES.
- Paciente 13 – **Paciente**: Paciente genérico, que tem todas as conexões com todos os neurônios dos outros grupos com valor 0. As conexões entre ele e neurônios do mesmo grupo tem valor -1.

A matriz de conectividade que representa a relação entre os pacientes e sintomas S/P é apresentada na Tabela 4.11.

Tabela 4.11: Matriz de conectividade entre os grupos 'sintomas' e 'pacientes'.

S/P	1	2	3	4	5	6	7	8	9	10	11	12	13
A	0	0,5	0	0	0	0	0	0	0	0	0	0	0
B	0,5	0	0,5	0,5	0,5	0,5	0,5	0	0,5	0,5	0,5	0,9	0
C	0	0,9	0,5	0	0,5	0,5	0	0,5	0,5	0,5	0,5	0	0
D	0	0	0	0	0,5	0,5	0	0	0	0	0	0	0
E	0	0	0	0,5	0	0,5	0	0	0,5	0	0	0	0
F	0	0	0	0,9	0	0	0	0	0	0	0,5	0	0
G	0	0	0	0	0	0	0,9	0	0,9	0	0	0,9	0
H	0	0	0	0	0	0	0,9	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0,5	0	0	0	0	0
J	0	0	0	0	0	0	0,5	0,9	0	0,9	0	0	0
K	0	0,5	0	0	0,5	0	0	0	0	0	0,5	0	0
L	0	0	0	0	0,5	0	0	0	0	0	0,5	0	0
M	0	0	0	0,9	0	0	0	0	0	0	0	0	0
N	0	0	0,9	0	0	0	0	0	0	0	0	0	0

Como observado, os 6 primeiros pacientes foram diagnosticados com AR e os 6 últimos foram diagnosticados com LES. Assim, a relação entre pacientes e diagnóstico fica estabelecida de acordo com a Tabela 4.12.

Tabela 4.12: Matriz de conectividade entre os grupos 'diagnóstico' e 'pacientes'.

D/P	AR	LER
Anne	1	-1
Helen	1	-1
Mary	1	-1
Susan	1	-1
Lucy	1	-1
Carol	1	-1
Pat	-1	1
Jackie	-1	1
Sue	-1	1
Beth	-1	1
Linda	-1	1
Mag	-1	1

4.3.2 Diagnóstico Diferencial em Reumatologia: Quatro Possibilidades

Esta versão realiza o diagnóstico entre 4 tipos de doenças: AR, LES, AGt, AP. A rede possui 50 neurônios organizados em 4 grupos interconectados, conforme ilustrado na Figura

4.2. Os grupos que constam na rede são os seguintes:

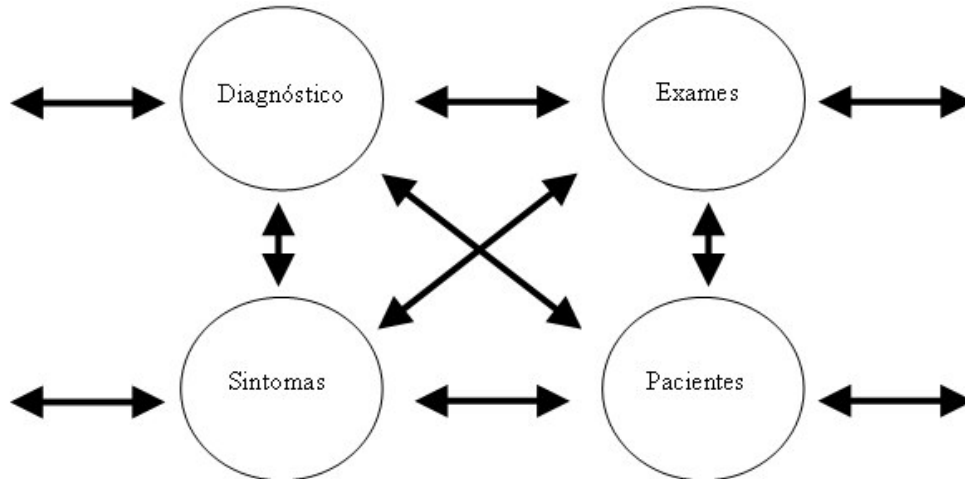


Figura 4.2: Esquema da rede modelo 'A' que implementa o Diagnóstico Diferencial em Reumatologia, com quatro possibilidades.

- grupo 'diagnóstico': contém quatro neurônios referentes às quatro possibilidades de diagnóstico.
- grupo 'pacientes': contém 25 neurônios, onde 24 se referem a pacientes já diagnosticados e 1 é um paciente genérico.
- grupo 'sintomas': contém 17 neurônios referentes aos sintomas relacionados a cada doença. São um total de 17 sintomas. Além dos 14 sintomas do exemplo anterior, contém mais 3 sintomas, associados aos dois diagnósticos acrescentados: psoríase, tophi e podagra.
- grupo 'exames': contém 4 neurônios referentes os exames necessários para realizar o diagnóstico entre as diferentes doenças: Fator Reumatóide Positivo, ANA, Fator Reumatóide Negativo e Hiperuricemia.

A matriz de conectividade tem a forma geral apresentada na Tabela 4.13, diferenciando do exemplo com 2 possibilidades de diagnóstico por possuir mais um grupo, que é o de 'exames'.

Aqui, o grupo 'diagnóstico' também tem seus valores não mutuamente excludentes, seguindo o padrão do exemplo anterior. Assim, a submatriz D/D , que traz as conexões entre os neurônios deste grupo, tem a forma apresentada na Tabela 4.14. As conexões entre os neurônios do grupo 'sintomas' e do grupo 'pacientes' assumem o mesmo padrão do exemplo anterior.

Tabela 4.13: Forma geral da matriz de conectividade para a rede de 50 neurônios.

	Diagnóstico	Sintomas	Pacientes	Exames
Diagnóstico	D/D	D/S^T	D/P^T	D/E^T
Sintomas	D/S	S/S	S/P^T	S/E^T
Pacientes	D/P	S/P	P/P	P/E^T
Exames	D/E	S/E	P/E	E/E

Tabela 4.14: Matriz de conectividade entre os neurônios do grupo 'diagnóstico', para a rede de 50 neurônios.

D/D	AR	LES	AP	AGt
AR	0	-0,9	-0,9	-0,9
LES	-0,9	0	-0,9	-0,9
AP	-0,9	-0,9	0	-0,9
AGt	-0,9	-0,9	-0,9	0

A inclusão dos três sintomas foi feita em conjunto com o especialista de domínio, e a informação obtida foi adicionada àquela já conhecida sobre o problema, sendo que os valores das conexões entre estes neurônios são apresentadas na Tabela 4.15:

- Sintoma A - Febre: Presente nas 4 doenças.
- Sintoma B - Artralgia: Presente nas 4 doenças.
- Sintoma C - Artrite: Presente nas 4 doenças.
- Sintoma D - Rigidez Matinal: Presente na AR e na AP, e ausente na LES e AGt.
- Sintoma E - Mialgia: Presente na AR, LES e AP, e ausente na AGt.
- Sintoma F - Nódulos Subcutâneos: Presente somente na AR.
- Sintoma G - Manchas Borboleta: Presente somente na LES.
- Sintoma H - Fenômeno de Raynoud: Presente na AR e LES, e ausente na AP e AGt.
- Sintoma I - Fotossensibilidade: Presente somente na LES.
- Sintoma J - Alopecia: Presente somente na LES.
- Sintoma K - Manifestação Renal: Presente na AR, LES e AGt, e ausente na AP.
- Sintoma L - Manifestação SNC: Presente na AR, LES e AGt, e ausente na AP.
- Sintoma M - Manifestação Renal: Presente na AR e LES, e ausente na AP e AGt.
- Sintoma N - Mão Reumatóide: Presente somente na AR.

Tabela 4.15: Matriz de conectividade entre os grupos 'diagnóstico' e 'sintomas', para a rede de 50 neurônios.

<i>D/S</i>	AR	LES	AP	AGt
Febre	0,4	0,6	0,4	0,4
Artralgia	1	1	0,9	0,9
Artrite	0,9	1	0,9	0,9
Rigidez matinal	0,7	-0,9	0,7	0
Mialgia	0,4	0,4	0,2	0
Nódulos subcutâneos	0,8	-0,1	-0,9	-0,9
Manchas borboleta	-1	0,8	-0,9	-0,9
Fenômeno de Raynoud	0,2	0,7	-0,9	-0,9
Fotossensibilidade	-0,1	0,6	-0,1	-0,1
Alopécia	-0,9	0,8	-0,5	-0,5
Manifestações Renais	0,1	0,4	-0,5	0,5
Manifestações do SNC	0,2	0,4	-0,5	0,1
Manifestações pulmonares	0,2	0,2	-0,1	-0,1
Mão reumatóide	0,8	-0,9	-0,9	-0,9
Psoríase	0,1	-0,1	0,7	-0,1
Tophi	-0,9	-0,9	-0,9	0,8
Podagra	-0,9	-0,9	-0,9	0,8

- Sintoma O - Psoríase: Presente na AR e AP, e ausente na LES e AGt.
- Sintoma P - Tophi: Presente somente na AGt.
- Sintoma Q - Podagra: Presente somente na AGt.

Foram inseridos 12 novos pacientes, além dos pacientes apresentados anteriormente, sendo que o paciente genérico foi adicionado ao final da listagem:

- Paciente 13 – **Marilyn**: artralgia, artrite, lesões psoriáticas. Diagnóstico: AP.
- Paciente 14 – **Gill**: artralgia, artrite, rigidez matinal, lesões psoriáticas. Diagnóstico: AP.
- Paciente 15 – **Jeanny**: artralgia, artrite, lesões psoriáticas. Diagnóstico: AP.
- Paciente 16 – **Meryl**: artralgia, rigidez matinal, lesões psoriáticas. Diagnóstico: AP.
- Paciente 17 – **Rita**: artralgia, artrite, lesões psoriáticas. Diagnóstico: AP.
- Paciente 18 – **Ingrid**: febre, artralgia, artrite, lesões psoriáticas. Diagnóstico: AP.
- Paciente 19 – **Ruth**: artralgia, artrite, rigidez matinal, tophi. Diagnóstico: AGt.

Tabela 4.17: Matriz de conectividade entre os grupos 'síntomas' e 'pacientes', para a rede de 50 neurônios.

<i>S/P</i>	13	14	15	16	17	18	19	20	21	22	23	24	25
A	0	0	0	0	0	0,5	0	0	0	0	0	0	0
B	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0
C	0,5	0,5	0,5	0	0,5	0,5	0,5	0,5	0,5	0	0,5	0,5	0
D	0	0,5	0	0,5	0	0	0,5	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	0	0,5	0	0
L	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0,9	0,9	0,9	0,9	0,9	0,9	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0,9	0,9	0,9	0,9	0	0	0
Q	0	0	0	0	0	0	0	0,9	0,9	0	0	0,9	0

4.4 Resumo dos Exemplos

Os três exemplos escolhidos para a realização dos testes fornecem diferentes graus de dificuldade no momento de sua implementação. O modelo dos Jets e Sharks é um exemplo típico para implementação utilizando a rede IAC original, já que as relações existentes entre os seus neurônios é muito bem definida. Assim após encontrar uma topologia adequada, definir os pesos entre os neurônios não é uma tarefa difícil.

Já os modelos que envolvem diagnóstico (Hepatite e Reumatologia) não podem ser implementados usando a rede IAC original. Para estes exemplos, além da dificuldade em se definir a topologia, ainda existe a dificuldade na definição do relacionamento entre os diferentes neurônios. Neste caso, a metodologia que se pretende desenvolver pode ser de extrema utilidade no momento de definir tais pesos.

Tabela 4.18: Matriz de conectividade entre os grupos 'diagnóstico' e 'pacientes', para a rede de 50 neurônios.

<i>D/P</i>	AR	LER	AP	AGt
Anne	1	-1	-1	-1
Helen	1	-1	-1	-1
Mary	1	-1	-1	-1
Susan	1	-1	-1	-1
Lucy	1	-1	-1	-1
Carol	1	-1	-1	-1
Pat	-1	1	-1	-1
Jackie	-1	1	-1	-1
Sue	-1	1	-1	-1
Beth	-1	1	-1	-1
Linda	-1	1	-1	-1
Mag	-1	1	-1	-1
Marilyn	-1	-1	1	-1
Gill	-1	-1	1	-1
Jeanny	-1	-1	1	-1
Meryl	-1	-1	1	-1
Rita	-1	-1	1	-1
Ingrid	-1	-1	1	-1
Ruth	-1	-1	-1	1
Sarah	-1	-1	-1	1
Laura	-1	-1	-1	1
Barbara	-1	-1	-1	1
Jane	-1	-1	-1	1
Evelyn	-1	-1	-1	1

Capítulo 5

Metodologia de Treinamento

A rede modelo 'A' apresenta características interessantes para alguns tipos de aplicações, tais como Mecanismos de Adaptação de Interfaces [31, 33]. Este modelo, porém, tem como dificuldade a definição dos pesos para as conexões entre os neurônios. Desta forma, torna-se necessário desenvolver uma metodologia que encontre uma matriz de conectividade para tais redes, de forma automática. Os AGs foram a ferramenta escolhida para realizar tal tarefa.

Hipótese: *'É possível o desenvolvimento de uma metodologia para realizar o ajuste de pesos de uma rede modelo 'A', utilizando Algoritmos Genéticos'.*

Para comprovação da hipótese, neste capítulo, apresenta-se a metodologia elaborada, cujo esquema geral é apresentado na Figura 5.1, e discute-se os seus detalhes. O seu objetivo é minimizar as dificuldades e tempo gasto na tarefa de realizar o ajuste de pesos de redes modelo 'A', fornecendo uma matriz de conectividade o mais apta possível para solucionar o problema.

Os principais esforços no desenvolvimento deste trabalho se concentraram na definição da representação dos indivíduos e no cálculo da aptidão, que são pontos fundamentais ao se utilizar um AG. Se estes não forem definidos de forma apropriada, acabam por inviabilizar a busca de uma solução, fazendo com que o desempenho do AG não seja satisfatório.

O indivíduo, que representa uma possível solução para o problema, é composto por uma seqüência de variáveis, sendo que cada uma delas representa um peso da matriz de conectividade. Dependendo da quantidade de neurônios apresentados pela rede, esta matriz pode ser bem grande, gerando um indivíduo bastante extenso. O seu tamanho pode ser restringido, se fatos conhecidos sobre relações entre os neurônios forem utilizados, permitindo que alguns pesos sejam definidos *a priori*, sem que seja necessário encontrá-los por meio do AG. Além disso, deve ser encontrado um alfabeto adequado para representar cada peso, de maneira que torne possível representar o espaço de soluções de maneira otimizada.

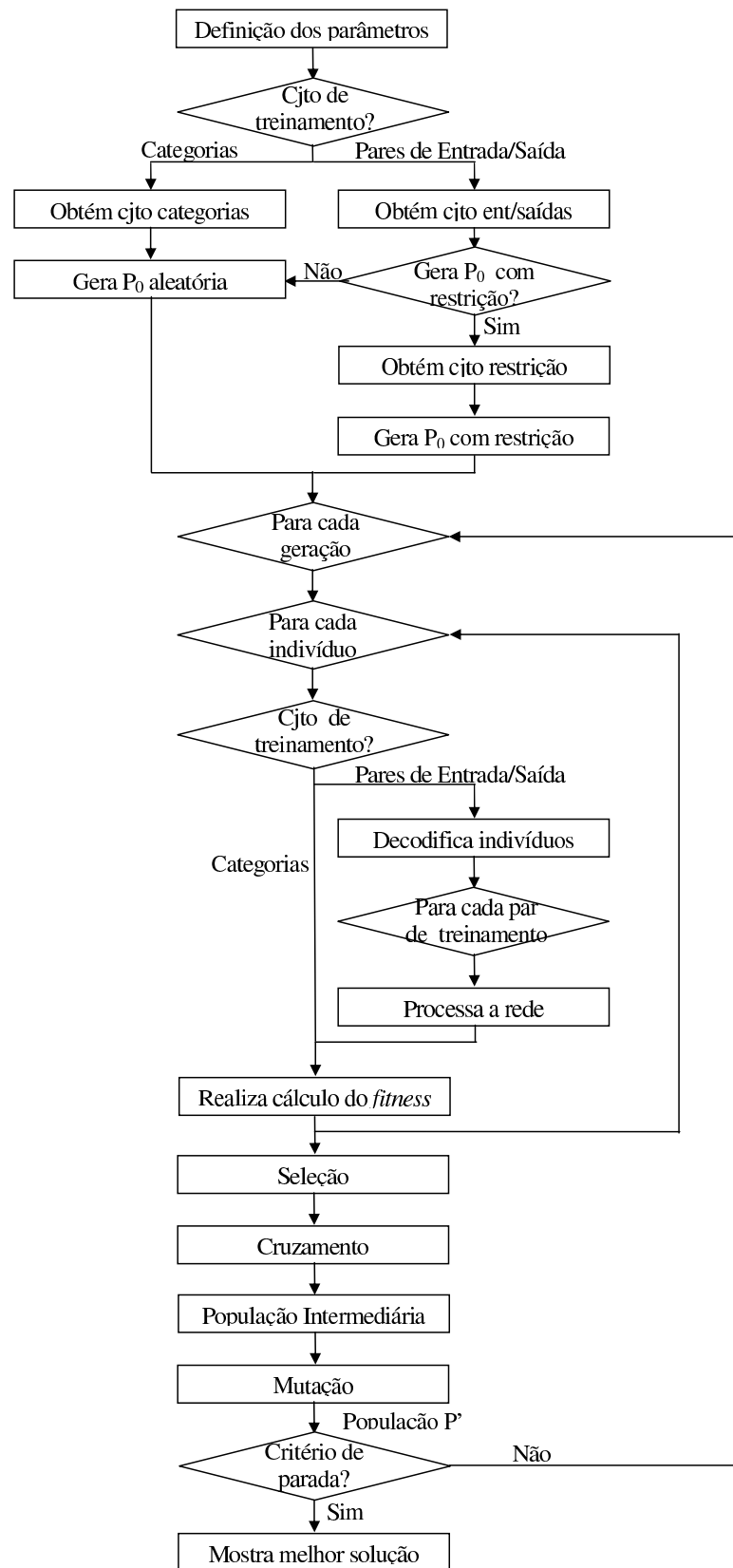


Figura 5.1: Fluxograma geral desenvolvido para a metodologia de treinamento.

Já a função de *fitness* gera, para cada indivíduo pertencente à população, um valor que traduz a sua aptidão para solucionar o problema, guiando, dessa forma, o processo de busca. Assim, somente boas soluções devem gerar bons valores de *fitness*, ou seja, a função deve medir o desempenho da rede para o problema sendo considerado.

A seguir, são detalhados os passos envolvidos na metodologia. Todos os resultados obtidos, as etapas de desenvolvimento, assim como a análise do desempenho, são apresentados no Capítulo 8.

5.1 Representações

Um AG trabalha com soluções codificadas, e não com a solução propriamente dita. Por este motivo, a definição da representação é um passo fundamental na sua utilização. Para ser possível definí-la, deve-se ter bem claro qual o problema que se quer resolver. No caso deste trabalho, o objetivo é encontrar um conjunto de pesos que represente a matriz de conectividade de uma rede modelo 'A'. Portanto, o indivíduo é codificado de forma a conter os pesos que compõem a matriz de conectividade da rede. Ou seja, supondo uma rede modelo 'A', composta por M neurônios, organizados em L grupos, devem ser encontrados os pesos w_{ij} , com $i = [1, M]$ e $j = [1, M]$.

Neste tipo de rede, porém, dois fatos são conhecidos previamente:

- Não existe conexão de um neurônio para ele mesmo. Desta forma, a matriz de conectividade W tem diagonal principal nula, ou seja, todos os elementos pertencentes à diagonal principal são iguais a zero,

$$w_{ij} = 0, \text{ se } i = j, \text{ com } i = [1, M] \text{ e } j = [1, M]$$

- As conexões entre os neurônios são bidirecionais, o que resulta numa matriz de conectividade simétrica. Isso implica que o valor da conexão do neurônio i para o neurônio j é igual ao valor da conexão do neurônio j para o neurônio i , independente dos neurônios pertencerem ou não a um mesmo grupo.

$$w_{ij} = w_{ji}, \text{ se } i \neq j, \text{ com } i = [1, M] \text{ e } j = [1, M]$$

Se uma matriz não cumpre estes dois requisitos, ela não pode ser considerada como representativa de uma rede modelo 'A'. Para exemplificar, a Figura 5.2 traz uma rede modelo 'A', que contém três grupos visíveis ($L = 3$), sem grupo escondido, cada um com dois neurônios

($M = 6$), organizados da seguinte forma: o Grupo 1 contém os neurônios a e b ; o Grupo 2 contém os neurônios c e d ; o Grupo 3 contém os neurônios e e f . A matriz de conectividade W para uma rede com estas características é apresentada na Tabela 5.1, que tem a diagonal principal composta por zeros, e a parte inferior e superior a esta diagonal simétricas.

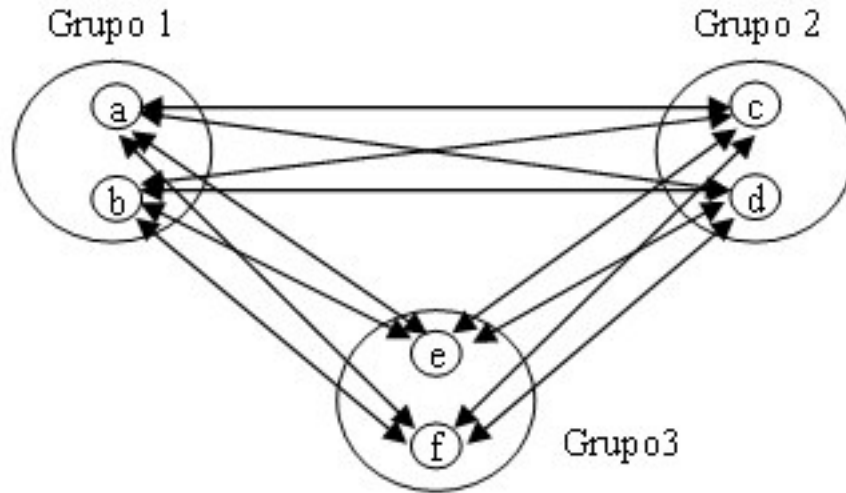


Figura 5.2: Exemplo de rede modelo 'A', onde $M = 6$ e $L = 3$.

Tabela 5.1: Matriz de conectividade da rede modelo 'A' apresentada na Figura 5.2.

		Grupo 1		Grupo 2		Grupo 3	
		a	b	c	d	e	f
Grupo 1	a	0	w_{ab}	w_{ac}	w_{ad}	w_{ae}	w_{af}
	b	w_{ab}	0	w_{bc}	w_{bd}	w_{be}	w_{bf}
Grupo 2	c	w_{ac}	w_{bc}	0	w_{cd}	w_{ce}	w_{cf}
	d	w_{ad}	w_{bd}	w_{cd}	0	w_{de}	w_{df}
Grupo 3	e	w_{ae}	w_{be}	w_{ce}	w_{de}	0	w_{ef}
	f	w_{af}	w_{bf}	w_{cf}	w_{df}	w_{ef}	0

Assim, a codificação passa a considerar estas duas restrições. Portanto, os pesos a serem encontrados são aqueles situados abaixo da diagonal principal, que é preenchida com zeros. Para encontrar os pesos acima desta diagonal, basta aplicar a propriedade de simetria. Desta forma, o número de pesos Pe a serem encontrados pelo AG é dado pela Equação 5.1.

$$Pe = \frac{M^2 - M}{2} \tag{5.1}$$

onde:

M Número de neurônios da rede

Esta abordagem foi a primeira considerada para a representação, e foi chamada de representação 'Abaixo da Diagonal Principal'.

Existem, porém, outras restrições que podem ser consideradas no momento da definição da codificação do indivíduo. A outra abordagem utilizada chama-se representação "Entre grupos que têm relação", e considera, também:

- Quais grupos têm relação: Em uma rede modelo 'A', existe a possibilidade de conexão entre todos os neurônios de todos os grupos. Entretanto, pode-se saber, previamente, que entre alguns deles, não existe relação, com todas as conexões entre seus neurônios assumindo valor 0.
- Os grupos que representam 'conceitos' com 'valores' mutuamente exclusivos: Quando um grupo representa um 'conceito' que tem 'valores' mutuamente exclusivos, existem dois pesos possíveis para as conexões entre seus neurônios, que é valor -1, quando os neurônios são diferentes e valor 0 de um neurônio para ele mesmo, garantindo que a matriz diagonal seja nula. Então, para os neurônios do grupo L_k :

$$w_{ij} = \begin{cases} -1, & \text{se } i \neq j \\ 0, & \text{se } i = j \end{cases}$$

sendo: $L_{k, inicial} \leq i, j \leq L_{k, final}$, onde:

$L_{k, inicial}$ Número correspondente ao primeiro neurônio do grupo L_k

$L_{k, final}$ Número correspondente ao último neurônio do grupo L_k

Nem sempre é possível aplicar estas duas restrições, devendo-se analisar cada problema, para verificar a viabilidade. Supondo novamente a rede modelo 'A', mostrada na Figura 5.2, mas considerando o fato de que os grupos 1 e 2 representem características com valores mutuamente exclusivos e que os neurônios destes dois grupos não tenham relação entre si, se obtém a rede da Figura 5.3.

A matriz de conectividade que representa esta rede é mostrada na Tabela 5.2. Esta matriz é composta por algumas submatrizes, denominadas de MI , CI , SR e C . A submatriz MI é padrão, e é utilizada para representar os pesos entre os neurônios pertencentes a um mesmo grupo, quando este representa um 'conceito' com 'valores' mutuamente exclusivos. Nesta matriz, todos os pesos têm valores -1, com exceção da diagonal principal, que é nula. Um exemplo de uma matriz MI é apresentado na Tabela 5.3, que traz as conexões entre os neurônios do Grupo 1. Assim, para os pesos desta matriz tem-se:

$$w_{ij} = \begin{cases} -1, & \text{se } i \neq j \\ 0, & \text{se } i = j \end{cases}$$

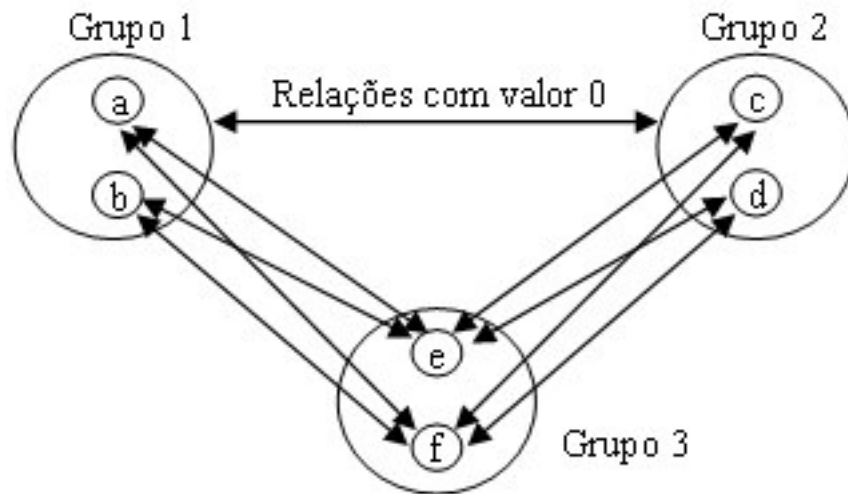


Figura 5.3: Rede modelo 'A', onde os grupos 1 e 2 não têm relação entre si e representam 'conceitos' com 'valores' mutuamente exclusivos.

Tabela 5.2: Modelo geral da matriz de conectividade para a rede da Figura 5.3.

	Grupo 1	Grupo 2	Grupo 3
Grupo 1	<i>MI</i>	<i>SR</i>	C^T
Grupo 2	<i>SR</i>	<i>MI</i>	
Grupo 3	<i>C</i>		<i>CI</i>

Tabela 5.3: Exemplo de uma matriz *MI*, para o Grupo 1.

Grupo 1	<i>a</i>	<i>b</i>
<i>a</i>	0	-1
<i>b</i>	-1	0

Contudo, as submatrizes indicadas por *SR* são matrizes nulas, e são utilizadas para indicar as conexões entre neurônios pertencentes a grupos que não têm relação. Assim, para a submatriz *SR*, para qualquer w_{ij} , $w_{ij} = 0$.

Os pesos compreendidos pela submatriz *C* são encontrados por meio do AG, sendo C^T a sua transposta. Pode-se dizer que *C* compreende o conhecimento a respeito do problema, indicando as conexões entre neurônios pertencentes a grupos que estão relacionados.

A matriz *CI* indica que o grupo tem neurônios não mutuamente excludentes. Apenas os pesos abaixo da diagonal principal desta submatriz devem ser encontrados pelo AG, já que ela é simétrica e tem diagonal principal nula.

O número total de pesos a ser encontrado, considerando todas as restrições apresentadas até o momento, varia de acordo com as definições iniciais feitas para o problema, e depende:

- do número de neurônios da rede;

- dos grupos que representam 'conceitos' com valores não mutuamente exclusivos;
- dos grupos que apresentam neurônios que se relacionam entre si.

Depois de definir quais pesos devem ser encontrados pelo algoritmo de treinamento, é necessário definir qual o alfabeto a ser utilizado e quantos genes são utilizados para representar cada peso. Entre os alfabetos escolhidos, estão o binário e o inteiro.

5.1.1 Alfabeto Binário

O alfabeto binário compreende os valores 0 e 1: $A = \{0, 1\}$. Foram propostas diferentes possibilidades usando este mesmo alfabeto, relacionadas às diferentes quantidades de genes utilizadas para representar cada peso.

DEZ GENES POR PESO

Com esta representação, são utilizados dez genes, ou dez bits, conforme o exemplo mostrado na Figura 5.4. O primeiro gene indica o sinal, sendo o valor zero o sinal positivo, e o valor um o sinal negativo. Os outros nove representam a parte fracionária do número. O valor máximo permitido é igual a 011111111_b ou $+0,998046875_d$. O valor mínimo é 111111111_b ou $-0,998046875_d$. Utilizando esta precisão, obtém-se 1024 números diferentes.

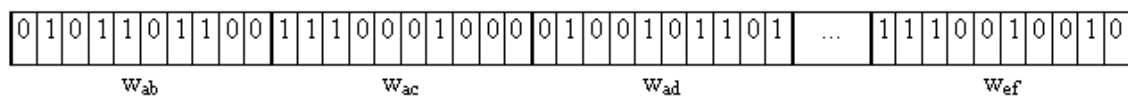


Figura 5.4: Exemplo de representação binária com 10 genes por peso.

SEIS GENES POR PESO

Outra possibilidade é a utilização de seis genes para cada peso, já que dez genes fornecem uma precisão muito grande, com muitas casas decimais após a vírgula, o que talvez não seja necessário. O primeiro gene permanece indicando o sinal e os outros cinco representam a parte fracionária do número. O valor máximo permitido é igual a 011111_b ou $+0,96875_d$. O valor mínimo é 11111_b , ou $-0,96875_d$, e obtém-se apenas 64 valores neste intervalo.

CINCO GENES POR PESO

Com as duas possibilidades acima, permite-se que existam pesos negativos entre grupos que têm relação e pesos positivos entre neurônios do mesmo grupo. A terceira possibilidade considerada foi remover o gene utilizado para indicar o sinal, mantendo apenas cinco

genes. No momento de realizar a decodificação, transforma-se o conjunto de cinco genes no peso da matriz de conectividade, verificando se o peso irá representar uma conexão intragrupo ou intergrupo. Caso seja um peso relacionado a uma conexão intragrupo, este é, automaticamente, considerado negativo, e caso seja relacionado a uma conexão intergrupo é considerado positivo.

5.1.2 Alfabeto Inteiro

Após a realização de mais estudos bibliográficos, verificou-se a possibilidade de utilização de outros alfabetos para a codificação. Assim, é possível, também, utilizar um alfabeto inteiro. Para se adequar às necessidades deste problema, que precisa representar valores no intervalo $[-1,1]$, o alfabeto compreende valores no intervalo $[-10, +10]$.

$$A = \{-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

Portanto, o cromossomo é composto por números inteiros, dentro da faixa estabelecida. Quando se utiliza a representação inteira, podem ser usados um ou dois genes na codificação.

UM GENE POR PESO

Cada gene que compõe o indivíduo passou a representar um peso. Um exemplo de indivíduo para a matriz de conectividade da Tabela 5.1 é apresentado na Figura 5.5. E a partir desta, é gerado o fenótipo apresentado na Figura 5.6.

-2	+6	-4	+5	...	-2
w_{ab}	w_{ac}	w_{ad}	w_{ae}		w_{ef}

Figura 5.5: Exemplo de representação inteira, usando um gene por peso.

-0,2	+0,6	-0,4	+0,5	...	-0,2
w_{ab}	w_{ac}	w_{ad}	w_{ae}		w_{ef}

Figura 5.6: Fenótipo gerado a partir do genótipo da Figura 5.5.

Dentro da faixa de valores estabelecida, existem duas possibilidades de representação. Pode-se utilizar apenas valores pertencentes ao intervalo $[0, 10]$, e converter o sinal de acordo com o fato da conexão ser intragrupo ou intergrupo. Ou então, utilizar valores no intervalo $[-10, 10]$ e permitir que conexões intragrupo assumam valores positivos (mais raro) e permitir que conexões intergrupo assumam valores negativos, o que pode acontecer com frequência quando se tratando de valores reais para as conexões.

DOIS GENES POR PESO

Outra opção é utilizar uma representação com dois genes, que permite representar valores entre $[-0,99,+0,99]$, com incremento de 0,01. A Figura 5.7 traz uma possibilidade de genótipo para a matriz da Tabela 5.1, enquanto a Figura 5.8 traz o fenótipo relacionado. Neste caso, utilizam-se os valores inteiros no intervalo $[-9,9]$. O sinal é definido de acordo com o sinal do primeiro gene, sendo que o sinal do segundo gene é desprezado.

-2	+6	+4	-7	...	-2	+9
w_{ab}		w_{ac}			w_{ef}	

Figura 5.7: Exemplo de representação inteira, usando dois genes por peso.

-0,26	+0,47	...	-0,29
w_{ab}	w_{ac}		w_{ef}

Figura 5.8: Fenótipo gerado a partir do genótipo da Figura 5.7.

Da mesma forma que na representação com um gene, existem duas possibilidades. É possível utilizar tanto a forma em que se converte o sinal de acordo com o fato da conexão ser intragrupo ou intergrupo, ou então, permitir que conexões intragrupo assumam valores positivos e permitir que conexões intergrupo assumam valores negativos.

5.2 Obtenção do Conjunto de Treinamento

O conjunto de treinamento é utilizado para o cálculo do *fitness*, e deve estar armazenado em um arquivo texto, a ser lido pelo programa que realiza o ajuste de pesos. Foram propostas duas abordagens diferentes, sendo que em uma o conjunto é composto por pares de entrada e saída e na outra, é composto por números que indicam, para cada peso a ser encontrado pelo AG, a categoria de valores a que devem pertencer. Estas duas abordagens são descritas a seguir.

5.2.1 Conjunto de Treinamento com Pares de Entrada e Saída

Nesta abordagem, o conjunto consiste de NP pares de entrada e saída desejada E_p/D_p , que definem, para um determinado conjunto de neurônios ativos na entrada, qual o estado de ativação dos neurônios na saída, ou seja, se eles devem estar ativos ou inativos.

O número de pares contido no arquivo pode variar, sendo que esta decisão deve ser tomada com base na complexidade do problema a ser resolvido. Assim, como em um algoritmo de treinamento comum, o número de pares e quais pares devem ser utilizados são decididos de forma empírica.

Para exemplificar, considere-se a rede IAC apresentada na Figura 5.2. Um conjunto de treinamento possível para esta rede é apresentado na Tabela 5.4. Nele, há 2 pares de padrões E_p/D_p . O primeiro par informa que se um usuário ativar o neurônio a na entrada, deve ter os neurônios a , c e e ativos na saída, após o processamento da rede. Se o usuário ativa o neurônio b como entrada, deve ter os neurônios b , d e f ativos na saída após o processamento da rede. Cada padrão contém entradas com um único neurônio ativo, mas podem ser usadas, também, entradas com mais de um neurônio.

Tabela 5.4: Exemplo de conjunto de treinamento, com $NP = 2$, para a rede da Figura 5.2.

Neurônio	E_1	D_1	E_2	D_2
a	1	1	0	0
b	0	0	1	1
c	0	1	0	0
d	0	0	0	1
e	0	1	0	0
f	0	0	0	1

5.2.2 Conjunto de Treinamento por Categorias

Uma outra abordagem para o conjunto de treinamento utiliza valores que indicam, para cada peso da matriz de conectividade, em que faixa de valores, ou categoria, dentro do intervalo $[-1,1]$, ele deve se encontrar, em contraposição ao conjunto E_p/D_p usado anteriormente.

A sua montagem é feita com base no conhecimento que um especialista de domínio do problema a ser resolvido tem a respeito das relações existentes entre os neurônios. É necessário estabelecer intervalos de pertinência, que compõem categorias para o problema. O número de categorias, assim como os intervalos que as compõem podem variar, dependendo da aplicação. Um exemplo de um conjunto de categorias é o seguinte:

- 1: Valores entre 0,7 e 1. Indica que existe relação forte de excitação entre os neurônios.
- 2: Valores entre 0,2 e 0,6. Indica que existe relação fraca de excitação entre os neurônios.
- 3: Valores entre -0,1 e 0,1. Indica que não existe relação entre os neurônios.

- 4: Valores entre -0,2 e -0,6. Indica que existe relação fraca de inibição entre os neurônios.
- 5: Valores entre -0,7 e -1. Indica que existe relação forte de inibição entre os neurônios.

Por exemplo, supondo a rede IAC apresentada na Figura 5.2, a Tabela 5.5 traz uma possível configuração de categorias para os pesos da matriz de conectividade. O arquivo de treinamento, na forma como é armazenado no arquivo texto, está mostrado na Figura 5.9. Este arquivo contém as categorias para todos os pesos abaixo da diagonal principal. Exemplificando, o primeiro valor da primeira coluna (5) é a categoria para o peso da conexão entre o neurônio b e a , e o segundo valor da primeira coluna (1) é a categoria para o peso da conexão entre o neurônio c e a . Já o primeiro valor da segunda coluna (3), indica a categoria para o peso da conexão entre o neurônio c e b , e assim sucessivamente. Os intervalos podem variar, sendo que podem ser definidos mais ou menos intervalos, dependendo do problema que se quer resolver.

Tabela 5.5: Categorias de pesos para a matriz de conectividade da Figura 5.3.

		Grupo 1		Grupo 2		Grupo 3	
		a	b	c	d	e	f
Grupo 1	a	0	5	1	3	1	3
	b	5	0	3	1	3	1
Grupo 2	c	1	3	0	5	1	3
	d	3	1	5	0	3	1
Grupo 3	e	1	3	1	3	0	5
	f	3	1	3	1	5	0

5	3	5	3	5
1	1	1	1	
3	3	3		
1	1			
3				

Figura 5.9: Exemplo de conjunto de treinamento, quando usando categorias para os pesos.

5.3 Geração da População Inicial

É necessário gerar um conjunto de possíveis soluções para compor a população P_0 , que contém um número de indivíduos definido pelo usuário, mantido fixo ao longo das gerações.

Esta é feita de acordo com a codificação e alfabeto definidos para o problema. Foram definidas duas opções para a geração de P_0 : geração aleatória e geração com restrições.

5.3.1 Geração Aleatória

Consiste na geração puramente aleatória da população inicial. É feita respeitando o alfabeto a ser utilizado e o tamanho do cromossomo, sendo que este é definido com base na quantidade de genes escolhida para representar cada peso. No caso da representação binária, para cada gene, é escolhido aleatoriamente um valor entre 0 e 1. No caso da representação inteira, considera-se se é necessário gerar valores no intervalo $[-10, 10]$, ou $[0..10]$, no caso de se usar um gene por peso, ou então no intervalo $[-9, 9]$, ou $[0..9]$, no caso de se utilizar dois genes por peso.

5.3.2 Geração com Restrições

Uma outra opção existente consiste na geração dos indivíduos respeitando certas restrições. Neste caso, utiliza-se o arquivo que contém as categorias para cada peso a ser encontrado pelo algoritmo de treinamento, sendo que cada um deles é gerado dentro do intervalo de valores definido pela respectiva categoria. Se este arquivo é utilizado na geração inicial, não pode ser utilizado como conjunto de treinamento, que passa a ser composto por pares de entrada e saída.

5.4 Processamento da Rede

Quando se utiliza pares de entrada/saída como conjunto de treinamento, é necessário que cada uma das entradas seja aplicada a cada indivíduo que compõe a população, para que seja possível calcular o seu valor de *fitness*. As respostas fornecidas pelo indivíduo, para cada entrada de cada padrão, devem ser armazenadas.

Antes de realizar o processamento, porém, é necessário realizar a decodificação de cada indivíduo, representado por um conjunto de genes, em uma matriz de conectividade de uma rede modelo 'A', que é uma matriz de números reais. No momento da montagem da matriz, observa-se como o indivíduo está codificado (ou seja, qual o alfabeto utilizado, quantos genes são usados para representar cada peso), e verificam-se quais as restrições existentes, se houver alguma, que indicam quais pesos devem ser encontrados por meio do AG, e quais pesos devem ser substituídos por valores padrão.

Em seguida, realiza-se o processamento da rede propriamente dito, que é feito com base no equacionamento, apresentado no Capítulo 2. A cada indivíduo pertencente à população, que é decodificado, é aplicado cada padrão de entrada E_p , contido na matriz de testes. Este processamento é feito por 60 ciclos, sendo este, geralmente, um número suficiente para que as redes alcancem o seu ponto de equilíbrio. As saídas Y_p^i geradas pelo indivíduo i , para cada padrão p , são armazenadas, para posterior cálculo do *fitness*, onde é feita a comparação com a saída desejada D_p .

No caso de ser utilizado um conjunto de treinamento que contém categorias, o processamento não se faz necessário, e a decodificação é realizada apenas ao final do processamento, quando o indivíduo com melhor valor de *fitness* é transformado numa matriz de conectividade, que caracteriza a resposta do AG para aquele treinamento.

5.5 Cálculo do *Fitness*

A função de *fitness* deve informar, para o AG, uma medida do desempenho que a matriz de conectividade apresenta com relação a um determinado conjunto de treinamento. No caso de ser usado o treinamento por pares de entrada e saída, o princípio básico consiste em comparar a saída Y_p^i , obtida do processamento do indivíduo i , para a entrada E_p , com a saída desejada D_p , obtida da matriz de treinamento. Com isso, calcula-se as taxas de acerto de acordo com o método escolhido, sendo que diferentes métodos foram desenvolvidos, sempre na tentativa de melhorar a convergência do algoritmo. Para o treinamento que utiliza categorias para cada peso, é necessário comparar se o peso encontrado pertence à categoria correspondente.

5.5.1 Cálculo Simples

Este método consiste na comparação simples entre a saída obtida pelo indivíduo i , para a entrada E_p , representada por Y_p^i , e a saída desejada para o padrão p , D_p . Cada entrada E_p da matriz de treinamento é processada pela matriz de conectividade que o indivíduo representa e a saída de cada neurônio é armazenada e comparada com a saída desejada D_p . Então, a cada neurônio que tem seu estado de ativação correto, o *fitness* é incrementado de um.

Ao final, o *fitness* fornece a taxa de acerto da rede, sendo este um valor real no intervalo $[0,1]$, que deve ser maximizado. Para que um indivíduo seja considerado ativo, o seu valor de ativação deve ser maior do que um valor va , definido previamente, e que pode variar, dependendo do problema sendo estudado. Sendo assim, o *fitness* $f(i)$ é dado pela Equação 5.2.

$$f(i) = \frac{Acertos(i)}{NP * M} \quad (5.2)$$

onde:

NP Número de padrões contidos no arquivo de treinamento
 M Número de neurônios apresentado pela rede

A variável $Acertos(i)$, que indica o número de neurônios que têm o estado de ativação de saída igual ao desejado, para o indivíduo i , considerando todos os padrões, é dada pela Equação 5.3.

$$Acertos(i) = \sum_{p=1}^{NP} acpad(p, i) \quad (5.3)$$

onde:

NP Número de padrões de entrada/saída

A variável $acpad(p, i)$, que indica, para o indivíduo i , após a aplicação de E_p , o número de neurônios que têm o mesmo estado de ativação que a saída desejada D_p , para o padrão p , é dada pela Equação 5.4.

$$acpad(p, i) = \sum_{j=1}^M acneuronio_p^i(j) \quad (5.4)$$

onde:

M Número de neurônios apresentado pela rede

A variável $acneuronio_p^i(j)$ indica se a saída para o neurônio j , fornecida pelo indivíduo i , para o padrão p , é igual a saída desejada deste neurônio para o padrão p . Este valor é dado pela Equação 5.5.

$$acneuronio_p^i(j) = \begin{cases} 1, & \text{se } Y_p^i(j) \geq va & \text{e } D_p(j) = 1 \\ 1, & \text{se } Y_p^i(j) < va & \text{e } D_p(j) = 0 \\ 0, & \text{em outras situações} \end{cases} \quad (5.5)$$

onde:

$Y_p^i(j)$ Saída do neurônio j , fornecida pelo individuo i , para o padrão p
 va Valor a partir do qual o neurônio é considerado ativo

5.5.2 Cálculo por 3 Valores

Esta proposta para função de *fitness* considera faixas de acerto, que indicam que o neurônio pode estar ativo, parcialmente ativo e inativo. Se o neurônio j está parcialmente ativo na saída $Y_p^i(j)$, e a saída desejada $D_p(j)$ tem valor 1, então é considerado somente meio acerto. Os estados foram assim definidos, para as saídas obtidas:

- O neurônio j é considerado inativo se o seu valor de ativação é menor que um valor mínimo a partir do qual ele é considerado parcialmente ativo, chamado v_{min} , ou seja, quando $Y_p^i(j) < v_{min}$;
- O neurônio j é considerado ativo somente se o seu valor de ativação é maior ou igual a um valor máximo, a partir do qual ele é considerado ativo, chamado v_{max} , ou seja, quando $Y_p^i(j) \geq v_{max}$;
- O neurônio j é considerado parcialmente ativo se o seu valor estiver entre os valores de v_{min} e v_{max} , ou seja, quando $v_{min} \leq Y_p^i(j) < v_{max}$.

Os valores de v_{max} e v_{min} podem ser definidos de acordo com o problema. Na maioria das simulações realizadas, v_{max} foi definido com valor de 0,5 e v_{min} com valor de 0,1. Estes valores foram escolhidos observando-se o comportamento da rede original de Rumelhart, e foram aqueles que forneceram bons resultados nos testes realizados.

Desta forma, ao se comparar as saídas obtidas e as desejadas, quando o neurônio deve estar ativo, mas a sua ativação, de acordo com o indivíduo sendo analisado, é um valor entre v_{min} e v_{max} , soma-se somente 0,5 ao número de acertos.

Neste método, o valor de *fitness* é dado pela Equação 5.6.

$$f(i) = \frac{Acertos(i)}{NP * M} \quad (5.6)$$

onde:

NP Número de padrões contidos no arquivo de treinamento

M Número de neurônios apresentado pela rede

A variável $Acertos(i)$, já definida, é dada pela Equação 5.7.

$$Acertos(i) = \sum_{p=1}^{NP} acpad(p, i) \quad (5.7)$$

onde:

NP Número de padrões de entrada/saída

A variável $acpad(p, i)$, também já definida, é dada pela Equação 5.8.

$$acpad(p, i) = \sum_{j=1}^M acneuronio_p^i(j) \quad (5.8)$$

onde:

M Número de neurônios apresentado pela rede

A variável $acneuronio_p^i(j)$ indica se a saída para o neurônio j , fornecida pelo indivíduo i , para o padrão p , é igual a saída desejada deste neurônio para o padrão p . Este valor é dado pela Equação 5.9.

$$acneuronio_p^i(j) = \begin{cases} 1, & \text{se } Y_p^i(j) \geq va_{max} & \text{e } D_p(j) = 1 \\ 0,5, & \text{se } va_{min} \leq Y_p^i(j) < va_{max} & \text{e } D_p(j) = 1 \\ 1, & \text{se } Y_p^i(j) < va_{min} & \text{e } D_p(j) = 0 \\ 0, & \text{em outras situações} \end{cases} \quad (5.9)$$

onde:

$Y_p^i(j)$ Saída do neurônio j , fornecida pelo indivíduo i , para o padrão p

5.5.3 Cálculo por Ponderação

Quando se utilizam pares de entrada e saída como conjunto de treinamento, pode ocorrer de, num determinado conjunto de padrões, haver muito mais neurônios inativos nas saídas desejadas, do que neurônios ativos. Quando isto ocorre, um indivíduo i qualquer, que represente uma rede que acerta muitos neurônios que deveriam estar inativos, pode acertar poucas ativações, e ainda assim, o valor de *fitness* ser bom. Foi definido que:

- Acertos Ativos: Informa o número de vezes que a rede ativa neurônios corretamente;
- Erros Ativos: Informa o número de vezes que a rede deixa de ativar neurônios que deveriam estar ativos;
- Acertos Inativos: Informa o número de vezes que a rede mantém inativos os neurônios que deviam estar inativos;

- Erros Inativos: Informa o número de vezes que a rede ativa neurônios que deveriam estar inativos.

Dessa forma, os acertos ativos e inativos são somados separadamente. Então, é dado um valor de ponderação VP , no intervalo $[0; 1]$, para os acertos ativos, e um valor de ponderação $|1 - VP|$ para os inativos. Assim, os neurônios ativos nos padrões, se forem em menor número, podem receber um valor de ponderação maior, trazendo um equilíbrio entre os dois tipos de acertos.

O cálculo do valor de *fitness* de cada indivíduo i é dado pela Equação 5.10.

$$f(i) = \frac{AcAtivos(i)}{TA} * VP + \frac{AcInativos(i)}{TI} * (1 - VP) \quad (5.10)$$

onde:

- $AcAtivos(i)$ Número de vezes que a rede, representada pelo indivíduo i , ativa um neurônio que deveria estar ativo, de acordo com D_p , considerando todos os padrões
- $AcInativos(i)$ Número de vezes que a rede representada pelo indivíduo i , mantém inativo um neurônio que deveria estar inativo, de acordo com D_p , considerando todos os padrões
- TA Número total de neurônios ativos nas saídas desejadas D_p
- TI Número total de neurônios inativos nas saídas desejadas D_p

A variável $AcAtivos(i)$ é dada pela Equação 5.11.

$$AcAtivos(i) = \sum_{p=1}^{NP} AcAtivoPad(p, i) \quad (5.11)$$

onde:

- $AcAtivoPad(i)$ Indica, para o indivíduo i , o número de neurônio ativos obtidos para E_p , e que estão ativos na saída desejada D_p , considerando apenas o padrão p
- NP Número de padrões de entrada/saída, contidos no conjunto de treinamento

A variável $AcAtivoPad(p, i)$, é dada pela Equação 5.12.

$$AcAtivoPad(p, i) = \sum_{j=1}^M AtivoN_p^i(j) \quad (5.12)$$

onde:

- M Número de neurônios apresentado pela rede
 $AtivoN_p^i(j)$.. Indica se a saída obtida para o neurônio j , fornecida pelo indivíduo i , para E_p , está de acordo com o estado desejado em D_p , considerando somente os neurônios ativos em D_p

A variável $AtivoN_p^i(j)$ é dada pela Equação 5.13.

$$AtivoN_p^i(j) = \begin{cases} 1, & \text{se } Y_p^i(j) \geq v_{max} & \text{e } D_p(j) = 1 \\ 0,5, & \text{se } v_{min} \leq Y_p^i(j) < v_{max} & \text{e } D_p(j) = 1 \\ 0, & \text{em outras situações} \end{cases} \quad (5.13)$$

onde:

- $Y_p^i(j)$ Saída do neurônio j , fornecida pelo indivíduo i , para E_p , considerando os neurônios ativos em D_p

A variável $AcInativos(i)$ é dada pela Equação 5.14.

$$AcInativos(i) = \sum_{p=1}^{NP} AcInativoPad(p, i) \quad (5.14)$$

onde:

- $AcInativoPad(i)$ Indica, para o indivíduo i , o número de neurônio inativos obtidos para E_p , e que estão inativos na saída desejada D_p considerando apenas o padrão p
 NP Número de padrões de entrada/saída, contidos no conjunto de treinamento

A variável $AcInativoPad(p, i)$ é dada pela Equação 5.15.

$$AcInativoPad(p, i) = \sum_{j=1}^M InativoN_p^i(j) \quad (5.15)$$

onde:

- M Número de neurônios apresentado pela rede
 $InativoN_p^i(j)$ Indica se a saída obtida para o neurônio j , fornecida pelo indivíduo i , para E_p , está de acordo com o estado desejado em D_p , considerando somente os neurônios inativos em D_p

A variável $InativoN_p^i(j)$ é dada pela Equação 5.16.

$$InativoN_p^i(j) = \begin{cases} 1, & \text{se } Y_p^i(j) < v_{min} \\ 0, & \text{em outras situações} \end{cases} \quad \text{e } D_p(j) = 0 \quad (5.16)$$

onde:

$Y_p^i(j)$ Saída do neurônio j , fornecida pelo indivíduo i , para E_p , considerando os neurônios inativos em D_p

5.5.4 Função de *Fitness* para Categorias

Utilizar qualquer uma das duas opções de conjunto de treinamento não modifica a representação do indivíduo. Apenas modifica o arquivo de treinamento e a função de *fitness* utilizados. Para um conjunto de treinamento que utiliza categorias de pertinência para os pesos, foi elaborada uma função de *fitness* que faz a verificação de quais valores de peso, fornecidos pelo indivíduo, se encaixam no padrão exigido pelo arquivo de entrada. Quanto mais valores de pesos se adequarem, melhor o valor de *fitness* gerado.

O número de categorias pode variar, de acordo com a complexidade do problema. Quanto mais complexo, mais precisa deve ser a definição de modo a se obter melhores resultados na rede final. Portanto, pode haver N intervalos, onde cada intervalo T_N compreende um intervalo $[a_N, b_N]$, sendo a_N o valor inicial do intervalo, e b_N o valor final.

O *fitness* $f(i)$ de um indivíduo i é dado pela Equação 5.17, que considera que apenas a parte inferior à diagonal principal deve ser encontrada.

$$f(i) = \frac{Acertos(i)}{Pe} \quad (5.17)$$

onde:

$Acertos(i)$... Número de vezes que o indivíduo i fornece pesos dentro do intervalo desejado
 Pe Número de pesos a ser encontrado

A variável $Acertos(i)$ é dada pela equação 5.18.

$$Acertos(i) = \sum_{k=1}^{M-1} \sum_{j=i}^{M-1} AcPeso_i(k, j) \quad (5.18)$$

onde:

- $AcPeso_i(k, j)$. Indica se o indivíduo i forneceu um valor dentro do intervalo desejado para o peso w_{kj}
- M Número de neurônios apresentado pela rede

A variável $AcPeso_i(k, j)$ é dada pela Equação 5.19.

$$AcPeso_i(k, j) = \begin{cases} 1, & \text{se } a_N \leq w_{kj} \leq b_N \text{ e } wd_{kj} = T_N \\ 0, & \text{caso contrário} \end{cases} \quad (5.19)$$

onde:

- a_N Valor inicial do intervalo I_N
- b_N Valor final do intervalo I_N
- T_N Identificador do intervalo N
- wd_{kj} Categoria desejada para o valor da conexão entre os neurônios k e j
- w_{kj} Valor da conexão entre os neurônios k e j

5.5.5 Considerações sobre o *fitness*

Resumidamente, até o momento, o *fitness* pode ser calculado por meio de:

- um conjunto de treinamento de pares de entrada e saída, com geração aleatória da população inicial;
- um conjunto de treinamento com faixas de pertinência, com geração aleatória da população inicial;
- um conjunto de treinamento de pares de entrada e saída, com geração restritiva dos elementos da população inicial, com base num arquivo de categorias de pertinência;

Estas três formas apresentam vantagens e desvantagens. Podem existir diferentes matrizes que dão bons valores de *fitness*, mas que respondem de diferentes maneiras, dando melhores ou piores resultados para o problema como um todo.

5.5.6 Exemplos de Aplicação das Funções de *Fitness*

Para exemplificar como funciona cada uma das formas de cálculo de *fitness*, que traz pares de entrada e saída como conjunto de treinamento, considera-se a rede apresentada na Tabela 5.4, que contém dois pares de entrada/saída para treinamento.

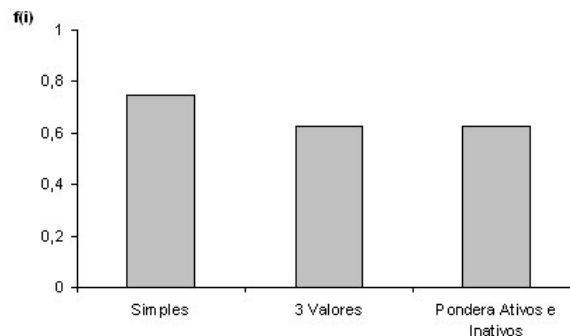
A Tabela 5.6 mostra respostas dadas por uma matriz hipotética gerada por meio de AG, fornecida por um indivíduo hipotético, chamado Indivíduo 1. Nela, consta a resposta dada por este indivíduo para cada um dos padrões de treinamento. Mostra também, o número de acertos fornecido para cada padrão pelo método simples e pelo método dos 3 valores. O método pondera ativos e inativos faz o cálculo de acertos por padrão da mesma forma que o método dos 3 valores, e por isso está sendo omitido.

Tabela 5.6: Exemplos de cálculo do *fitness*.

Neurônios	D_1	Y_1^1	Simple	3 valores	D_2	Y_2^1	Simple	3 valores
<i>a</i>	1	0,7	1	1	0	-0,1	1	1
<i>b</i>	0	0,35	0	0	1	0,7	1	1
<i>c</i>	1	0,42	1	0,5	0	0,3	0	0
<i>d</i>	0	-0,1	1	1	1	-0,2	0	0
<i>e</i>	1	0,3	1	0,5	0	-0,1	1	1
<i>f</i>	0	-0,2	1	1	1	0,3	1	0,5
			5 acertos	4 acertos			4 acertos	3,5 acertos

Como pode-se observar, o número de neurônios M multiplicado pelo número de padrões NP é 12. Pelo método simples, o total de acertos é igual a 9. Já pelo método dos 3 valores, o total de acertos é 7,5.

O valor de *fitness* fornecido pelo Indivíduo 1, pelos 3 métodos existentes é apresentado no Figura 5.10. Considerando este exemplo, não há diferença entre o método pondera ativos e inativos e o método por 3 valores, porque o número de neurônios ativos nas saídas desejadas (6) coincide com o número de neurônios inativos nas saídas desejadas (6).

Figura 5.10: Exemplos de valores de *fitness*, fornecido pelas diferentes funções.

O método por 3 valores fornece um valor de *fitness* menor, o que força o AG a procurar soluções que ativem com mais força os neurônios que devem estar ativos. O mesmo ocorre para o método pondera ativos e inativos. Se o número de neurônios inativos, fosse muito maior que o de ativos, esta forma de cálculo do *fitness* forçaria, ainda, o AG, a encontrar uma rede que respondesse bem tanto para neurônios ativos, como para neurônios inativos.

5.6 Seleção

Após a realização do cálculo do *fitness* para cada um dos indivíduos da população, realiza-se a seleção, responsável por indicar quais os indivíduos que devem ser cruzados. O número de indivíduos a ser selecionado é obtido através da taxa de cruzamento, que se mantém constante ao longo das gerações. Quando a taxa de cruzamento fornece um valor não divisível por 2, o número de indivíduos a ser cruzado é incrementado para o valor mais próximo divisível por 2. Para exemplificar, se a população contém 100 indivíduos, e se a taxa de cruzamento for de 90%, 90 indivíduos devem ser selecionados para cruzamento. Se a taxa for de 89%, também deve-se selecionar 90 indivíduos, já que 89 não é divisível por 2.

Foram utilizadas duas funções de seleção:

- **Passa mais aptos:** Por este método, são selecionados para cruzamento apenas os indivíduos considerados mais aptos, ou seja, aqueles que têm maior valor de *fitness*. Este método foi o que forneceu melhores resultados.
- **Roleta Ponderada:** Por este método, os indivíduos mais aptos têm mais chances de serem selecionados para o cruzamento, mas não impede que indivíduos menos aptos possam ser escolhidos, garantindo a variabilidade genética da população. Ao contrário do esperado, este método não guiou o AG para bons resultados.

5.7 Cruzamento

A cada dois indivíduos selecionados, estes são cruzados, gerando dois novos indivíduos. O cruzamento efetua a troca de material genético entre dois indivíduos pertencentes à população. Foram implementadas algumas possibilidades:

- **Cruzamento em um único ponto:** Escolhe-se, de forma aleatória, a cada novo cruzamento, um único ponto para realizar a troca do material genético.
- **Cruzamento de dois pontos:** Consideram-se os indivíduos circulares, escolhendo-se dois pontos para troca do material genético. Desta forma, os pontos de início e fim do cromossomo podem ser modificados.
- **Cruzamento de dois pontos sem considerar o indivíduo circular:** Escolhe-se dois pontos para realizar a troca do material genético, mas não se modificam os pontos de início e fim do cromossomo.

Normalmente, os indivíduos gerados são extensos, principalmente para os problemas mais complexos, que envolvem uma grande quantidade de neurônios. Foi cogitada a hipótese de se utilizar múltiplos pontos de cruzamento, na tentativa de melhorar a qualidade dos indivíduos gerados. Testes foram realizados, mas não houve melhoras na resposta da rede. Acredita-se que um aumento no número de pontos de cruzamento acaba por transformar a busca definida pelo AG numa busca aleatória.

5.8 Seleção da Nova População

Após o cruzamento, é realizada a seleção da nova população, a partir da população $P(t)$, que compõe uma nova geração $P(t + 1)$. Todos os indivíduos que foram gerados pelo cruzamento, ou seja, os indivíduos filhos são transportados diretamente para a nova população.

Então, dentre os indivíduos pais, é selecionado um certo número de indivíduos necessários para que o tamanho da população se mantenha constante. Os indivíduos passam por uma roleta ponderada, onde os menos aptos têm menos chances de serem escolhidos para fazerem parte da nova geração. Por exemplo, supondo uma população de 100 indivíduos, com taxa de cruzamento de 90%, há 90 indivíduos pais que geram 90 indivíduos filhos. Os filhos passam diretamente para a nova população. Os 10 restantes são escolhidos por meio de uma roleta ponderada entre os 100 indivíduos iniciais.

5.9 Mutação

A mutação é aplicada para permitir que todos os pontos do espaço de busca sejam alcançados. Para decidir se deve ou não ocorrer mutação, gera-se um número aleatório no intervalo $[0, 100]$. Se este número estiver no intervalo $[0, TM]$, onde TM é o valor da taxa de mutação, gera-se outro número aleatório pertencente ao intervalo $[1, TP]$, onde TP indica o número de indivíduos contidos na população.

Para este indivíduo, deve ser gerado, também aleatoriamente, um outro número pertencente ao intervalo $[1, TI]$, onde TI indica o tamanho do cromossomo. O alelo selecionado deve ser modificado para algum outro pertencente ao alfabeto. Caso se esteja usando a representação binária, simplesmente troca-se o valor do alelo. Se o valor for 0, troca-se para 1, e vice-versa.

Entretanto, quando se utiliza codificação inteira, escolhe-se, aleatoriamente, um outro valor pertencente ao intervalo utilizado, com exceção daquele que já está sendo utilizado no

alelo. Por exemplo, se o alelo tiver valor 0, e o alfabeto compreender o intervalo $[0, 10]$, pode-se selecionar para substituir o 0, um valor no intervalo $[1, 10]$.

Capítulo 6

Estabilidade

A estabilidade é uma das características importantes de um sistema, já que, usualmente, para este ter aplicabilidade prática deve ser projetado para ser estável. As RNA do tipo IAC são sistemas dinâmicos, conforme mostrado por De Azevedo [22, 34, 32] e, por consequência, também devem ser estáveis para serem utilizáveis.

Neste trabalho, torna-se necessário garantir que as redes geradas pelas diferentes opções apresentadas pela metodologia de treinamento elaborada representem sistemas estáveis. Isto implica em se estudar as características de estabilidade de forma a que este critério seja atendido. Este capítulo discute, sem apresentar conceitos fundamentais considerados conhecidos, o conceito de estabilidade, o critério de estabilidade de Liapunov, e a estabilidade de RNA realimentadas.

6.1 Conceitos Básicos

Aqui, o objeto de interesse é a descrição de sistemas que são modelos de sistemas físicos reais. Por conseguinte, todas as variáveis e funções são consideradas reais. Conceitos importantes relacionados a este tipo de sistema são os de Relaxação, Estacionaridade, Causalidade, Linearidade e Função de Transferência, sendo que um estudo a respeito destas propriedades pode ser encontrado em [88, 89, 90, 91, 92].

Um sistema é classificado como 'monovariável' se ele apresentar somente um terminal de entrada e um de saída, e é 'multivariável' no caso de existir mais de um terminal de entrada e/ou saída. A classe de sistemas aqui tratada (as RNA) apresenta terminais de entrada e de saída (no caso é multivariável). As entradas, ou causas ou excitações u são aplicadas aos

terminais de entrada, e as saídas, ou efeitos ou respostas y são medidas nos terminais de saída.

O primeiro passo no estudo analítico de um sistema é escrever as equações matemáticas que o descrevem. Por causa dos diferentes métodos analíticos possíveis de usar, ou por causa de diferentes questões a serem respondidas pode-se, muitas vezes, descrever o mesmo sistema por diferentes equações matemáticas [88].

Tem-se, por exemplo, a chamada 'descrição entrada-saída' do sistema (também conhecida como 'externa'), na qual a função de transferência descreve somente as propriedades terminais do sistema. Ou seja, o comportamento da saída em função da entrada. Outra descrição é a 'descrição por variáveis de estado' (também conhecida como 'interna'), onde um conjunto de equações diferenciais descreve tanto o comportamento terminal (entrada-saída) como o comportamento interno do sistema.

Um modelo freqüentemente usado na descrição entrada-saída de um sistema é através da utilização de equações diferenciais, o qual inclui derivadas no tempo e dá informação de como a resposta do sistema varia no tempo. Para a solução das equações diferenciais, ou do conjunto de equações diferenciais que representam um sistema, os métodos utilizados podem ser aqueles que tentam uma solução por aproximação de tentativa de solução ou aqueles que transformem a equação em outra forma que possa ser manuseada pela álgebra convencional.

No primeiro caso, para o estudo do comportamento do sistema, compara-se a resposta deste com a de uma excitação padronizada, que deve ser determinada a partir da resposta requerida do sistema e da forma real da excitação correspondente. Várias excitações padronizadas, tais como a função degrau, a rampa, a parábola, o impulso, entre outras, são utilizadas. O procedimento consiste na determinação da solução completa da equação diferencial referente a cada uma destas excitações. Ou seja, a determinação da resposta em regime permanente a cada um dos tipos de excitação.

Observa-se que tal abordagem é aplicável a equações diferenciais de qualquer ordem. A forma da componente transitória da resposta depende da equação característica. Os coeficientes dos termos transitórios são determinados a partir do valor instantâneo da resposta em regime permanente, das raízes da equação característica e das condições iniciais [90]. As características desejadas para um sistema de qualquer ordem podem ser especificadas em termos da resposta transitória a uma excitação degrau unitário. O desempenho do sistema pode ser avaliado em função de uma série de grandezas, tais como ultrapassagem máxima, tempo de passagem por erro nulo, tempo de acomodação, entre outras.

Um outro método utilizado para a solução dessas equações diferenciais é o da transformação por Transformada de Laplace. Neste caso, as equações diferenciais que descrevem o

comportamento do sistema são transformadas em equações algébricas simples que não envolvem o tempo, ou seja, transforma-se para o domínio s , no qual são realizadas as operações algébricas necessárias para a solução do problema.

Pode-se, ainda, afirmar que a solução de equações diferenciais, estendida às equações matriciais de estado e de resposta, torna o método aplicável a sistemas multivariáveis, não invariantes ao tempo e/ou não lineares. A formulação matricial destes sistemas conduz a soluções com a utilização de computadores digitais [90].

Estes procedimentos são utilizados em sistemas chamados 'sistemas contínuos no tempo' ou 'sistemas tempo contínuo', onde as entradas e saídas dos sistemas são definidos para todo t em $(-\infty, +\infty)$.

Todavia, o processamento de sinais tem utilizado mais e mais os computadores digitais, que não tratam sinais contínuos que ocorrem em processos naturais e técnicos, onde o tempo (ou a posição) é uma variável contínua.

É necessário, então, converter o sinal analógico para digital, para permitir a gravação/processamento por computadores digitais. Para isso é utilizado um conversor analógico-digital, efetuando o processo em duas etapas [93]: a) amostragem do sinal contínuo no tempo, $z(t)$, em instantes equidistantes, separados por um intervalo de tempo T (quantização do tempo), originando um sinal $z[k]$; e b) armazenagem da seqüência de valores de $z[k]$ na memória do computador com um número finito de bits (quantização da amplitude).

Os sistemas que tratam, processam e manipulam sinais deste tipo, digitais, são conhecidos como 'sistemas discretos no tempo' ou 'sistemas tempo discreto'. Da mesma forma que para sistemas tempo contínuo, as definições de relaxação, estacionaridade, causalidade e linearidade, entre outras, podem ser apropriadamente adequadas aos sistemas tempo discreto. Em geral, estas adequações implicam no uso de somatórios, ao invés de integrais, e na utilização da Transformada Z ao invés da de Laplace.

No caso de sistemas tempo discreto, o seu comportamento pode ser estudado tanto pela abordagem 'descrição entrada-saída', como pela abordagem 'descrição por variáveis de estado'. A maioria dos conceitos e resultados para o caso contínuo podem ser aplicados ao caso tempo discreto com pequenas modificações.

6.2 Definindo Estabilidade

Um sistema pode ser dito 'estável' se entradas limitadas, isto é, finitas, geram saídas limitadas [92]. Para exemplificar, considere-se o caso de um sistema linear: se a saída deste

sistema tender a zero, quando sujeito a uma entrada impulso, conforme o tempo tender a infinito, ele é dito 'estável'. Por outro lado, se a saída tender a infinito, quando o tempo tende a infinito, ele é dito 'instável'. O sistema pode ser ainda definido como 'criticamente estável' caso, nas mesmas condições, a saída não tender nem a zero nem a infinito.

Formalmente, diz-se que um sistema é do tipo 'entrada limitada/saída limitada' (BIBO - *bounded input/bounded output*) estável se, e somente se, toda entrada limitada resultar em uma saída limitada. A saída desse sistema não diverge se a entrada não divergir [94]. Para colocar essa condição de estabilidade BIBO em uma base formal, considere-se um sistema de tempo contínuo, cuja relação de entrada-saída é dada pela Equação 6.1.

$$y(t) = H[u(t)] \quad (6.1)$$

onde o operador H é BIBO estável se o sinal de saída $y(t)$ satisfaz a seguinte condição:

$$|y(t)| \leq M_y < \infty, \text{ para todo } t,$$

sempre que os sinais de entrada $u(t)$ satisfizerem a condição:

$$u(t) \leq M_u < \infty, \text{ para todo } t,$$

onde M_y e M_u são números positivos finitos.

Pode-se descrever a condição para a estabilidade BIBO de um sistema de tempo discreto de maneira semelhante.

No caso de sistemas lineares, existem diversas abordagens para o estudo da estabilidade, que são função da representação do sistema, ou seja, do modelo utilizado para descrever o comportamento do sistema. Para isto, um modelo deve descrever a relação entre entradas e saídas que são funções do tempo e, a partir daí, ser capaz de descrever o comportamento em regime transitório (a parte da resposta que ocorre quando há uma variação na entrada que termina depois de um curto intervalo de tempo) e em regime permanente (a parte da resposta que continua depois do transitório) [92].

A função de transferência de um sistema de malha fechada (Figura 6.1 é dada pela Equação 6.2 [89]:

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (6.2)$$

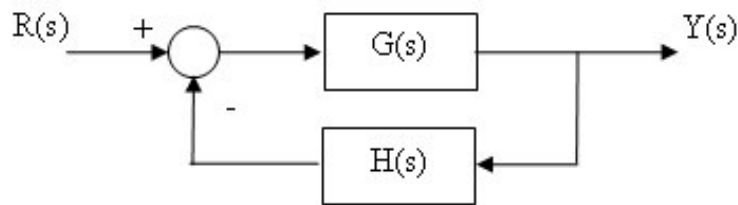


Figura 6.1: Sistema de malha fechada.

A equação característica deste sistema generalizado pode ser obtida igualando o denominador da função de transferência do sistema a zero, conforme a Equação 6.3 [89]:

$$1 + G(s)H(s) = 0 \quad (6.3)$$

Esta é a equação que determina a estabilidade do sistema. Todos os métodos de análise de estabilidade investigam esta equação de alguma maneira, sendo utilizados, em geral, os dois métodos seguintes [89]:

- Calcular as raízes exatas da Equação 6.3: Neste caso, há duas possibilidades, que são a abordagem clássica e o método de lugar de raízes;
- Determinação da região limite, onde as raízes da Equação 6.3 existem: Neste caso, tem-se a disposição algumas ferramentas, tais como o Critério de Routh-Hurwitz, o Diagrama de Nyquist, o Diagrama de Bode e o Gráfico de Nichols.

6.3 Estabilidade em Sistemas Não Lineares

O conceito de estabilidade para sistemas lineares e invariantes no tempo é fácil de ser entendido. Torna-se, no entanto, necessário estender o conceito de estabilidade a sistemas não lineares.

Mais especificamente, se tem interesse no estudo da estabilidade de RNA realimentadas, que são sistemas altamente não lineares, assim como outros modelos de RNA. Portanto, todo o arcabouço teórico citado e referenciado para sistemas lineares, até o momento, não se aplica. O ferramental teórico usado, neste caso, baseia-se nos estudos de Liapunov, mais particularmente, seu segundo método ou método direto, o qual propicia um meio de se determinar a estabilidade de um sistema sem obter explicitamente as trajetórias no espaço de estados. O segundo método é aplicável na determinação do comportamento de sistemas de ordem mais elevada, podendo ser forçados ou livres, lineares ou não lineares, invariantes ou variantes no tempo, e determinísticos ou estocásticos.

6.3.1 Critério de Estabilidade de Liapunov

Liapunov dividiu o problema geral de analisar a estabilidade de sistemas não lineares dentro de duas classes. A primeira classe consiste de todos aqueles métodos em que a equação diferencial do sistema pode ser resolvida, sendo a sua estabilidade, ou instabilidade determinada a partir desta solução. Este método, conhecido como o primeiro método de Liapunov, não traz informação importante relacionada à solução de equações diferenciais não lineares. Contudo, Liapunov indicou em seu primeiro método que a solução pode ser obtida na forma de uma série, a partir da qual a estabilidade pode ser determinada usando seu segundo método. Além disso, soluções aproximadas de equações diferenciais não lineares, freqüentemente, rendem informações de estabilidade útil.

O segundo método de Liapunov diz respeito à determinação da informação de estabilidade relacionada a sistemas não lineares sem ter que resolver sua equação diferencial. Isto contrasta com seu primeiro método, que requer a determinação dos autovalores a partir das equações linearizadas em torno de um ponto de equilíbrio. Manipulando os coeficientes da equação característica de uma certa maneira prescrita, é possível determinar se os pólos estão na metade direita ou esquerda do plano sem ter que resolver a equação diretamente. Este método é similar ao critério de estabilidade de Routh-Hurwitz para sistemas lineares.

O segundo método de Liapunov, que é baseado sobre considerações de energia, pode ser melhor estudado do plano de fase de um sistema de segunda ordem simples. Uma equação diferencial não linear de segunda ordem, que pode ser representada por duas variáveis x e y , pode ser descrita pelas Equações 6.4 e 6.5, que são duas equações de primeira ordem [89].

$$\frac{dx}{dt} = P(x,y) \quad (6.4)$$

$$\frac{dy}{dt} = Q(x,y) \quad (6.5)$$

Assumindo que $x = y = 0$ é um ponto singular, o segundo método de Liapunov diz que um sistema é estável se é possível achar a função $V(x,y)$ que tem as seguintes propriedades [89]:

- $V(x,y) = 0$, somente para $x = y = 0$;
- $dV(x,y)/dt$ nunca é positiva.

Além disso, este teorema diz que se dV/dt nunca é zero, exceto para o caso possível, quando $x = y = 0$, o equilíbrio é definido como sendo assintoticamente estável. Assim, pode

ser mostrado que a função V é análoga à energia armazenada no sistema, mas não é igual a ela.

A principal dificuldade na aplicação do método consiste em formular uma função candidata de Liapunov correta, tendo em vista que a falha de uma função em atender as condições de estabilidade não significa a inexistência de uma verdadeira função de Liapunov. Esta dificuldade é acrescida do fato de que a função de Liapunov não é a única. Para um estudo mais detalhado sobre o primeiro e segundo método de Liapunov, consulte [90, 91].

6.4 Estabilidade para RNAs com *Feedback*

Hopfield provou, em seu trabalho original [19], que a rede conhecida por seu nome, minimiza uma função de energia, constituindo-se em um sistema estável, segundo o senso de Liapunov. Zurada [95] mostra que redes BAM [96] se constituem em memórias estáveis bidirecionais, pelo mesmo critério. No caso específico deste trabalho, deve ser provado, também segundo Liapunov, que as redes IAC, bem como o modelo 'A', são estáveis. No entanto, isto já foi feito por Nascimento [21], que provou que a rede IAC é um sistema dinâmico não linear estável e que também minimiza uma função de energia, tal como a rede de Hopfield.

Como na rede de Hopfield, na rede IAC, a topologia é selecionada de forma a satisfazer restrições específicas do problema em estudo. A maior diferença na operação entre a rede de Hopfield e a rede IAC é a função de ativação não linear utilizada [21].

6.4.1 Estabilidade de Redes IAC

Nesta seção, é apresentada a prova tal como realizada por Nascimento [21]. A partir do equacionamento básico apresentado na seção 2.1, são feitas algumas considerações iniciais, para depois realizar a minimização de uma função de energia.

Sem perda de generalidade, e pelo fato de não se estar interessado no caso em que uma unidade está completamente isolada das outras, pode-se considerar que para cada unidade i , $w_{ij} \neq 0$ para ao menos um j . Ao assumir que $\min < 0 < \max$ e $|\min| = \max$, as Equações 2.4 e 2.5 podem ser combinadas, resultando na Equação 6.6.

$$\Delta y_i(k) = -|net_i(k)|y_i(k) + net_i(k)\max - \text{decay}(y_i(k) - \text{rest}) \quad (6.6)$$

Considerando que a rede opera em tempo contínuo, a Equação 6.6 é substituída pela Equação 6.7.

$$\frac{dy_i}{dt} = -|net_i|y_i + net_i max - decay(y_i - rest) \quad (6.7)$$

Para encontrar o ponto de equilíbrio do sistema y_i^e , resolve-se a Equação 6.6, estabelecendo $\Delta y_i(k) = 0$, ou a Equação 6.7, estabelecendo $dy_i/dt = 0$. Desta forma, considerando que net_i^e representa o valor de entrada da rede para a unidade i quando a rede alcança o ponto de equilíbrio, obtém-se a Equação 6.8.

$$y_i^e = \frac{maxnet_i^e + decay rest}{|net_i^e| + decay} \quad (6.8)$$

Geralmente, net_i^e é desconhecido, e portanto, a Equação 6.8 não auxilia na tarefa de encontrar a posição do ponto de equilíbrio, no caso geral. Pode-se, porém, considerar que se $decay = 0$:

- quando $net_i^e \neq 0$, o ponto de equilíbrio é caracterizado por $y_i^e = max$, se $net_i^e > 0$, ou $y_i^e = -max$, se $net_i^e < 0$;
- quando $net_i^e = 0$, a Equação 6.8 não pode ser usada para achar o ponto de equilíbrio, mas os pontos onde $net_i = 0$ para todas as unidades são também pontos de equilíbrio, já que Δy_i (ou dy_i/dt) = 0 para todo i . Um ponto onde isto é possível, mas não o único, é considerar $ext_i = 0$ para todas as unidades e, conseqüentemente, $y_i^e = 0$, para todo i , é um ponto de equilíbrio.

Além disto, para $rest = 0$ e pequenos valores de $decay$, sendo $|decay| \ll |net_i^e|$, o ponto de equilíbrio ainda se localiza próximo a max ou $-max$, e a condição $net_i = 0$ não é suficiente para causar um ponto de equilíbrio.

Observa-se que se $w_{ij} = 0$ para todo j , significando que a unidade i é completamente isolada das outras, então $net_i = ext_i$ e a condição para estabilidade é ter $-decay < |ext_i| < 2 - decay$, que pode também ser escrito como $-|ext_i| < decay < 2 - |ext_i|$. Portanto, tal unidade pode formar um sistema unidimensional estável mesmo no caso de $decay < 0$. A posição dos pontos de equilíbrio é dada pela Equação 6.8, substituindo net_i^e por ext_i .

De forma a mostrar que uma IAC também minimiza uma função de energia, inicialmente, assume-se que $decay = 0$ e que a rede está ou dentro ou nas bordas do hipercubo $[-max \ max]^M$, onde M é o número de unidades da rede. Isto significa que $-max \leq y_i \leq max$ para todo i . Pode-se definir a função quadrática da Equação 6.9 como a função de energia.

$$H(t) = -\frac{1}{2}O^T W O - ext^T O \quad (6.9)$$

onde ext e O são vetores coluna. Como no caso das redes de Hopfield, pode-se escrever a Equação 6.10.

$$\frac{dH[O(t)]}{dt} = \sum_{i=1}^M \frac{\partial H[O(t)]}{\partial O_i} \frac{dO_i}{dt} = [\nabla_O H(O)]^T \dot{O} \quad (6.10)$$

Dado que a matriz de pesos W é simétrica, tem-se a Equação 6.11.

$$\frac{dH}{dt} = -[W O + ext]^T \dot{O} = -net^T \dot{O} = -\sum_{i=1}^M net_i \frac{dO_i}{dt} \quad (6.11)$$

Sabe-se, porém, que $O_i = g(y_i)$. Desta forma, tem-se a Equação 6.12.

$$\frac{dH}{dt} = -\sum_{i=1}^M net_i \frac{dg(y_i)}{dy_i} \frac{dy_i}{dt} \quad (6.12)$$

Usando a Equação 6.7, finalmente, obtém-se a Equação 6.13.

$$\frac{dH}{dt} = \begin{cases} -\sum_{i=1}^M \frac{dg(y_i)}{dy_i} net_i^2 (max - y_i), & \text{if } net_i \geq 0 \\ -\sum_{i=1}^M \frac{dg(y_i)}{dy_i} net_i^2 (max + y_i), & \text{if } net_i < 0 \end{cases} \quad (6.13)$$

Portanto $dH/dt \leq 0$ para $-max \leq y_i \leq max$, $decay = 0$, e $dg(y_i)/dy_i \geq 0$ para todo i , sendo $g(\cdot)$ uma função monotonicamente crescente. A partir disto, pode-se também definir que $dH/dt = 0$ se e somente se $dO_i/dt = dy_i/dt = 0$ para todo i , isto é, a rede alcançou um ponto de equilíbrio. Observa-se que $net_i = 0$ para todo i implica não somente que $dH/dt = 0$, mas também que $dy_i/dt = 0$ para todo i (ver Equação 6.7).

Agora, é necessário estudar o caso em que a rede é inicializada fora do hipercubo $[-max \ max]^M$, ou seja, $-max > y_i > max$ para ao menos um i . Se $y_i \geq 0$, a Equação 6.7 pode ser escrita como a Equação 6.14.

$$\frac{dy_i}{dt} = \begin{cases} -|net_i|(|y_i| - max) - decay(|y_i| - rest) & \text{if } net_i \geq 0 \\ -|net_i|(|y_i| + max) - decay(|y_i| - rest) & \text{if } net_i < 0 \end{cases} \quad (6.14)$$

Por outro lado, se $y_i < 0$, a Equação 6.7 pode ser escrita como a Equação 6.15.

$$\frac{dy_i}{dt} = \begin{cases} |net_i|(|y_i| + max) + decay(|y_i| - rest) & \text{if } net_i \geq 0 \\ |net_i|(|y_i| - max) + decay(|y_i| - rest) & \text{if } net_i < 0 \end{cases} \quad (6.15)$$

As Equações 6.14 e 6.15 mostram respectivamente que, dado que $decay > 0$ e $|rest| < max$:

- se $y_i > max$, então $dy_i/dt < 0$;
- se $y_i < -max$, então $dy_i/dt > 0$.

Em outras palavras, considerando o espaço de ativação, se a rede está fora do hipercubo $[-max \ max]^M$ e $decay > 0$, as mudanças nos valores de ativação são tais que dado tempo suficiente, a rede alcança as bordas do hipercubo, fazendo $|y_i| \leq max$. Nota-se que, mesmo no caso onde $decay = 0$, as mudanças na ativação ainda guiam a rede para as bordas do hipercubo $[-max \ max]^M$, com a única exceção que a rede pode ser bloqueada na condição onde $net_i = 0$. Estando dentro ou nas bordas do hipercubo, a rede pesquisa o mínimo da função de energia dado pela Equação 6.9, dado que, entre outras condições, $decay = 0$.

Uma maneira de garantir que a função de energia dada pela Equação 6.9 é minimizada, é ter $decay > 0$ sempre que $|y_i| > max$, para ao menos um i . Quando $|y_i| \leq max$ para todo i , estabelece-se $decay$ para 0. Uma maneira mais simples seria definir o valor de $decay$ para algum valor positivo pequeno, e $rest$ para 0, sendo assim desnecessário considerar se a rede está dentro do hipercubo ou não.

Da Equação 6.8, pode-se ver que isto causa somente uma pequena perturbação na posição dos pontos de equilíbrio, que estão localizados onde $y_i^e = -max$ ou max , assumindo que para tal ponto de equilíbrio, a condição $|decay| \ll |net_i^e|$ é satisfeita. Se os pontos de equilíbrio, que são a solução para o problema satisfazem tal condição (em geral, tal informação não está disponível a priori), então, ainda poderia-se considerar que a função de energia dada pela Equação 6.9 está sendo minimizada. Contudo, a localização e o número dos outros pontos de equilíbrio (os pontos de equilíbrio que não estão nas esquinas do hipercubo $[-max \ max]^M$) podem mudar significativamente.

Uma possível interpretação para o fato de $decay > 0$ trazer a rede para as bordas do hipercubo, seria porque isto impede os pontos onde $net_i = 0$ de serem pontos de equilíbrio. E da Equação 6.8, pode-se ver que também força $|y_i^e| < max$. Contudo, alguns dos pontos que são pontos de equilíbrio para $decay = 0$ podem sofrer uma grande perturbação, se a condição $|decay| \ll |net_i^e|$ não é satisfeita.

6.5 Conclusões

Este capítulo teve como objetivo discutir a estabilidade tanto da rede IAC, e do modelo 'A', como das redes geradas pela metodologia proposta. Como visto anteriormente, foi provado, por outros autores, que as redes de Hopfield e a BAM são estáveis segundo o senso de Liapunov. Mais ainda, Nascimento [21] provou que as redes IAC, assim com as redes de Hopfield, também minimizam uma função de energia, sendo estáveis segundo Liapunov.

O modelo 'A' de rede utiliza a mesma arquitetura e o mesmo equacionamento e parâmetros da IAC. As diferenças entre os dois modelos não implicam em nenhuma modificação na função de energia proposta por Nascimento podendo, portanto, todas as suas conclusões para redes IAC serem estendidas para o modelo 'A'.

Mais importante, Cohen e Grossberg [97] publicaram um excelente estudo onde provam que redes realimentadas cujas matrizes de conectividade são simétricas e têm diagonal principal nula são estáveis, segundo Liapunov.

Os algoritmos propostos neste trabalho geram sempre, a cada iteração, ou seja, a cada geração, matrizes apresentando as duas condições acima. Desta forma, todas as matrizes resultantes de qualquer dos algoritmos aqui propostos representam sistemas estáveis segundo o senso de Liapunov.

Conclui-se, portanto, que as redes geradas pelos algoritmos, após o critério de parada ser atingido, representam sistemas estáveis e são, por consequência, de utilidade prática.

Capítulo 7

Ambiente Computacional Implementado

Para a realização dos testes e de estudos relacionados ao comportamento da rede, foi construído um ambiente computacional que implementa ambos os modelos, IAC e 'A'. Por meio deste, é possível, também, realizar treinamentos, aplicando-se a metodologia proposta. Neste caso, as redes geradas podem ser analisadas, permitindo que se aplique conjuntos de testes, para verificar as taxas de acerto, tornando viável, desta forma, a validação da metodologia.

Este ambiente pode ser utilizado, futuramente, por outros pesquisadores que tenham interesse em utilizar a rede IAC original e o modelo 'A'. A seguir, são descritas as principais funções implementadas.

7.1 Criação de uma Rede IAC

Este ambiente permite criar tanto uma rede IAC original, como uma rede Modelo 'A'. Para a criação de uma rede, é necessário informar a quantidade de grupos, e quantos neurônios cada grupo contém. Deve ser informado, também, um rótulo para cada neurônio.

Quando se quer criar uma rede IAC original, é necessário informar qual é o grupo escondido. Neste caso, os pesos padrão já são definidos, sendo permitido somente editar os pesos entre os diferentes grupos com o escondido. Pesos, como os pertencentes à diagonal principal, ou pesos entre neurônios do mesmo grupo já são automaticamente preenchidos. É permitido somente utilizar valores de peso 1, 0 ou -1.

No caso da rede modelo 'A', apenas os pesos da diagonal principal são preenchidos. Os outros podem ser preenchidos ou manualmente ou, então, obtidos por meio das propostas de treinamento.

7.2 Consulta de uma Rede IAC

Após a rede ser criada, e os pesos serem informados, ou diretamente, no caso da rede IAC original, ou por treinamento, no caso da rede modelo 'A', é possível realizar consultas a esta rede. Para isto, escolhe-se os neurônios referentes aos 'valores' dos 'conceitos' que compõem a consulta. É preciso definir o número de ciclos para que a rede estabilize, e também os parâmetros que se adequem melhor às necessidades do problema sendo representado. Depois de efetuadas estas definições, realiza-se a consulta, e as ativações fornecem a resposta da rede para a consulta apresentada.

É possível realizar modificações manuais na matriz de treinamento, para verificar as mudanças que estas alterações causam na resposta da rede para a consulta apresentada. Outra característica possível é realizar estatísticas para verificar se a matriz de conectividade está adequada. Arquivos de teste devem ser criados para realizar as estatísticas, que são compostos por pares de entrada/saída que informam para um conjunto de neurônios ativos na entrada, quais neurônios devem estar ativos na saída.

7.2.1 Implementação do Algoritmo de Atualização

O algoritmo de atualização consiste na implementação do equacionamento apresentado na seção 2.1. O exemplo dos Jets & Sharks foi reconstituído utilizando este sistema, com base no conhecimento que se encontrava disponível. As respostas obtidas por meio deste sistema foram comparadas com as respostas fornecidas pelo sistema implementado em [42] - PDP.

Assim, foram realizadas simulações utilizando os dois sistemas, para validar as respostas obtidas. Esperava-se que as respostas encontradas para as simulações fossem as mesmas, tanto qualitativa como quantitativamente em ambos os sistemas. Todos os testes realizados foram executados por 60 ciclos.

De acordo com as entradas fornecidas para teste, foi observado que as repostas fornecidas por ambos os sistemas são iguais. Deve-se observar, também, que foram utilizados os mesmos valores para os parâmetros.

Os 3 testes elaborados a seguir foram selecionados para demonstração:

Teste 1: Fornecimento de uma entrada composta pelo neurônio Jets. Neste caso, ativou-se o neurônio Jets, para verificação das respostas fornecidas. A Figura 7.1 mostra as respostas fornecidas pela sistema PDP, enquanto a Figura 7.2 mostra as respostas fornecidas por este sistema.


```

iac: ^C
disp/ exam/ get/ save/ set/ clear cycle do input log quit reset run
test

** Jets      84  0 Art      0  0 Phil    10  0 _Art     15  0 _Phil    16  cycle 60
0 Sharks    14  0 Al       10  0 Ike     10  0 _Al      12  0 _Ike    14
0 in20s     51  0 Sam      0  0 Nick    10  0 _Sam     15  0 _Nick   15
0 in30s     11  0 Clyde    0  0 Don     10  0 _Clyde   15  0 _Don    16
0 in40s     11  0 Mike     0  0 Ned     10  0 _Mike    15  0 _Ned    16
0 JH        51  0 Jim      0  0 Karl    10  0 _Jim     15  0 _Karl   16
0 HS        11  0 Greg     10  0 Ken     10  0 _Greg    12  0 _Ken    14
0 College   11  0 John     0  0 Earl    10  0 _John    15  0 _Earl   16
0 Single    51  0 Doug     10  0 Rick    10  0 _Doug    12  0 _Rick   16
0 Married   11  0 Lance    0  0 Ol      10  0 _Lance   15  0 _Ol     16
0 Divorce   11  0 George   0  0 Neal    10  0 _George  15  0 _Neal   15
0 Pusher    15  0 Pete     0  0 Dave    10  0 _Pete    15  0 _Dave   16
0 Burglar   15  0 Fred     0  0         0  0 _Fred    15
0 Bookie    15  0 Gene     0  0         0  0 _Gene    15
0           0  0 Ralph    0  0         0  0 _Ralph   15

```

Figura 7.1: Resposta do software PDP para a entrada Jets.

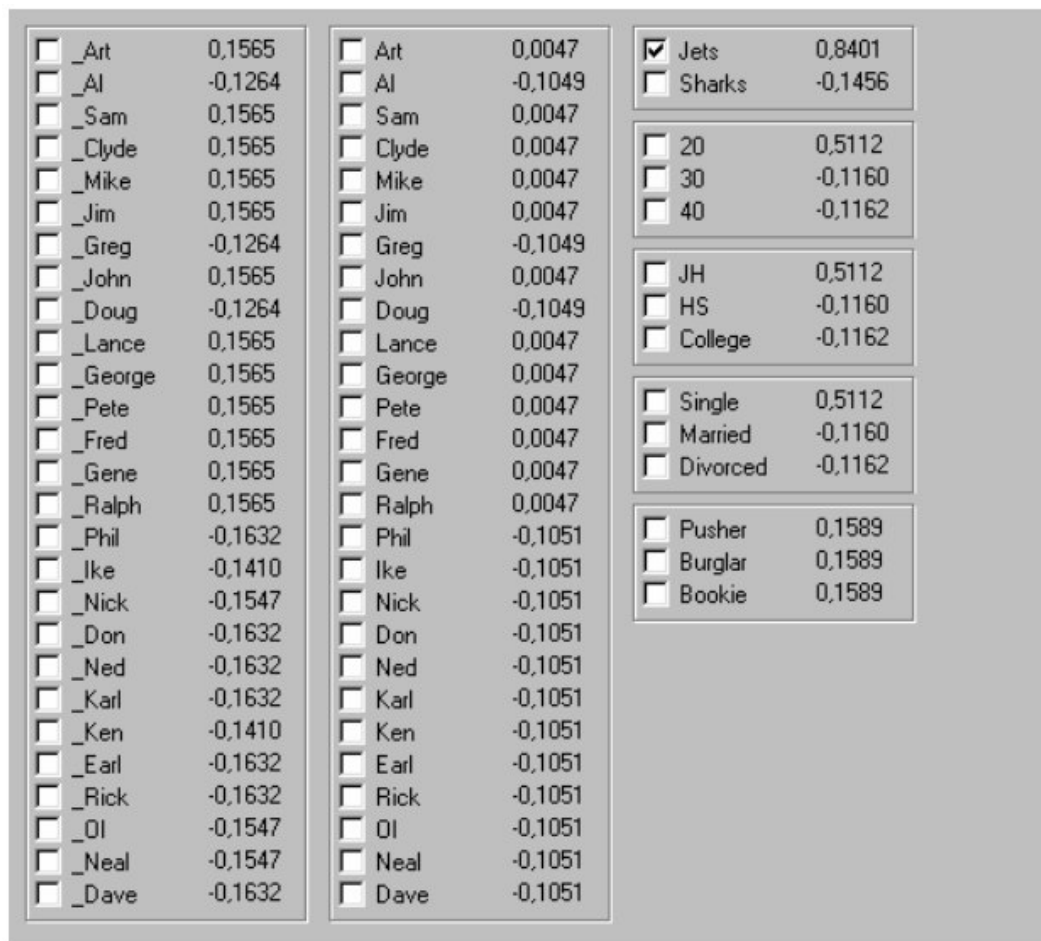


Figura 7.2: Resposta do sistema para a entrada Jets.

Teste 2: Fornecimento de uma entrada composta pelo neurônio Ol. Para o neurônio Ol, é feito o mesmo procedimento e a resposta dada pelo PDP é mostrada na Figura 7.3 e a

resposta dada por este sistema é mostrada na Figura 7.4.

```

iac:
disp/ _exam/ get/ save/ set/ clear cycle do input log quit reset run
test

0 Jets      13  0 Art      14  0 Phil    12  0 _Art    16  0 _Phil   56 cycle 60
0 Sharks   64  0 Al      14  0 Ike     14  0 _Al     14  0 _Ike   14
0 in20s    13  0 Sam     14  0 Nick    14  0 _Sam    15  0 _Nick  12
0 in30s    64  0 Clyde   14  0 Don     12  0 _Clyde  16  0 _Don   41
0 in40s    13  0 Mike    14  0 Ned     12  0 _Mike   15  0 _Ned   41
0 JH       13  0 Jim     14  0 Karl    14  0 _Jim    16  0 _Karl  14
0 HS       13  0 Greg    14  0 Ken     14  0 _Greg   14  0 _Ken   15
0 College  64  0 John    14  0 Earl    14  0 _John   15  0 _Earl  14
0 Single   13  0 Doug    14  0 Rick    14  0 _Doug   15  0 _Rick  14
0 Married  64  0 Lance  14  0 OI      80  0 _Lance  15  0 _OI    68
0 Divorce  13  0 George  14  0 Neal    14  0 _George 16  0 _Neal  14
0 Pusher   51  0 Pete    14  0 Dave    14  0 _Pete   16  0 _Dave  12
0 Burglar  11  0 Fred    14  0         0  0 _Fred   16
0 Bookie   11  0 Gene    14  0         0  0 _Gene   14
           0 Ralph  14  0         0  0 _Ralph  14

```

Figura 7.3: Resposta do software PDP para a entrada OI.

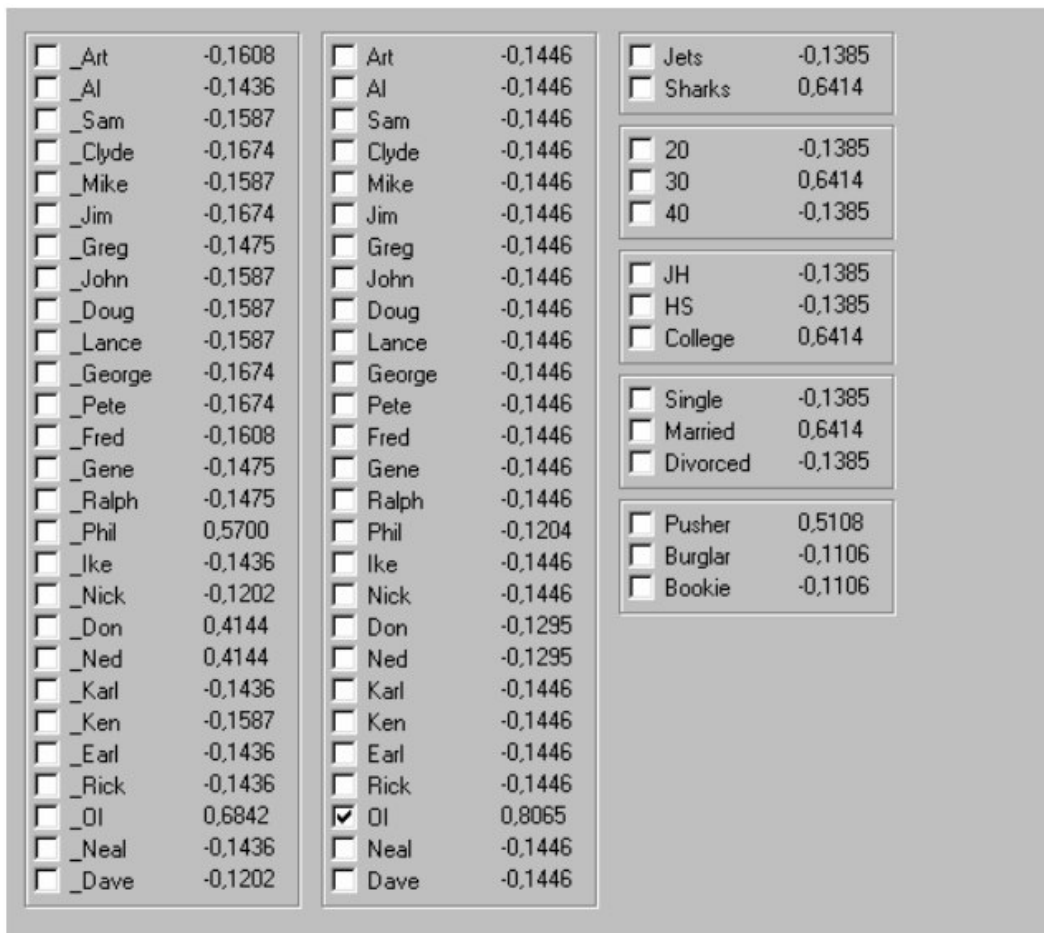


Figura 7.4: Resposta do sistema para a entrada OI.

Teste 3: Fornecimento de uma entrada composta pelos neurônios Casado + Faixa etária

20 anos. A resposta do PDP é dada na Figura 7.5 e a resposta deste sistema é mostrada na Figura 7.6

```

iac:
disp/ exam/ get/ save/ set/ clear cycle do input log quit reset run
test

0 Jets      68  0 Art      13  0 Phil     13  0 _Art     15  0 _Phil    16  cycle 60
0 Sharks   14  0 Al       10  0 Ike      13  0 _Al      38  0 _Ike    16
** in20s   84  0 Sam      13  0 Nick     13  0 _Sam     14  0 _Nick   17
0 in30s    13  0 Clyde    13  0 Don      13  0 _Clyde   15  0 _Don    14
0 in40s    14  0 Mike     13  0 Ned      13  0 _Mike    15  0 _Ned    16
0 JH       68  0 Jim      10  0 Karl     13  0 _Jim     39  0 _Karl   16
0 HS       14  0 Greg     12  0 Ken      13  0 _Greg     9  0 _Ken    14
0 College  14  0 John     22  0 Earl     13  0 _John    64  0 _Earl   14
0 Single   14  0 Doug     13  0 Rick     13  0 _Doug    16  0 _Rick   16
** Married 83  0 Lance    22  0 Ol       13  0 _Lance   64  0 _Ol     16
0 Divorce  10  0 George   10  0 Neal     13  0 _George  39  0 _Neal   17
0 Pusher   14  0 Pete     13  0 Dave     13  0 _Pete    14  0 _Dave   17
0 Burglar  68  0 Fred     13  0          0  0 _Fred    14
0 Bookie   14  0 Gene     13  0          0  0 _Gene    14
           0 Ralph   13  0          0  0 _Ralph   15

```

Figura 7.5: Resposta do software PDP para a entrada faixa etária de 20 anos e casados.

<input type="checkbox"/> _Art	-0,1525	<input type="checkbox"/> Art	-0,1311	<input type="checkbox"/> Jets	0,6831
<input type="checkbox"/> _Al	0,3885	<input type="checkbox"/> Al	-0,1069	<input type="checkbox"/> Sharks	-0,1405
<input type="checkbox"/> _Sam	-0,1484	<input type="checkbox"/> Sam	-0,1311	<input checked="" type="checkbox"/> 20	0,8447
<input type="checkbox"/> _Clyde	-0,1525	<input type="checkbox"/> Clyde	-0,1311	<input type="checkbox"/> 30	-0,1316
<input type="checkbox"/> _Mike	-0,1525	<input type="checkbox"/> Mike	-0,1311	<input type="checkbox"/> 40	-0,1458
<input type="checkbox"/> _Jim	0,3931	<input type="checkbox"/> Jim	-0,1065	<input type="checkbox"/> JH	0,6831
<input type="checkbox"/> _Greg	-0,0920	<input type="checkbox"/> Greg	-0,1298	<input type="checkbox"/> HS	-0,1401
<input type="checkbox"/> _John	0,6490	<input type="checkbox"/> John	0,2267	<input type="checkbox"/> College	-0,1405
<input type="checkbox"/> _Doug	-0,1641	<input type="checkbox"/> Doug	-0,1311	<input type="checkbox"/> Single	-0,1455
<input type="checkbox"/> _Lance	0,6490	<input type="checkbox"/> Lance	0,2267	<input checked="" type="checkbox"/> Married	0,8355
<input type="checkbox"/> _George	0,3931	<input type="checkbox"/> George	-0,1065	<input type="checkbox"/> Divorced	-0,1068
<input type="checkbox"/> _Pete	-0,1484	<input type="checkbox"/> Pete	-0,1311	<input type="checkbox"/> Pusher	-0,1401
<input type="checkbox"/> _Fred	-0,1484	<input type="checkbox"/> Fred	-0,1311	<input type="checkbox"/> Burglar	0,6831
<input type="checkbox"/> _Gene	-0,1484	<input type="checkbox"/> Gene	-0,1311	<input type="checkbox"/> Bookie	-0,1405
<input type="checkbox"/> _Ralph	-0,1525	<input type="checkbox"/> Ralph	-0,1311		
<input type="checkbox"/> _Phil	-0,1620	<input type="checkbox"/> Phil	-0,1311		
<input type="checkbox"/> _Ike	-0,1641	<input type="checkbox"/> Ike	-0,1311		
<input type="checkbox"/> _Nick	-0,1712	<input type="checkbox"/> Nick	-0,1311		
<input type="checkbox"/> _Don	-0,1487	<input type="checkbox"/> Don	-0,1311		
<input type="checkbox"/> _Ned	-0,1620	<input type="checkbox"/> Ned	-0,1311		
<input type="checkbox"/> _Karl	-0,1620	<input type="checkbox"/> Karl	-0,1311		
<input type="checkbox"/> _Ken	-0,1485	<input type="checkbox"/> Ken	-0,1311		
<input type="checkbox"/> _Earl	-0,1487	<input type="checkbox"/> Earl	-0,1311		
<input type="checkbox"/> _Rick	-0,1641	<input type="checkbox"/> Rick	-0,1311		
<input type="checkbox"/> _Ol	-0,1620	<input type="checkbox"/> Ol	-0,1311		
<input type="checkbox"/> _Neal	-0,1712	<input type="checkbox"/> Neal	-0,1311		
<input type="checkbox"/> _Dave	-0,1712	<input type="checkbox"/> Dave	-0,1311		

Figura 7.6: Resposta do sistema para a entrada faixa etária de 20 anos e casados.

7.3 Treinamento de uma Rede Modelo 'A'

Esta opção do sistema deve ser utilizada quando se quer encontrar a matriz de conectividade de uma rede modelo 'A'. Para a utilização deste sistema, existe uma série de parâmetros que devem ser ajustados ou informados, para que o desempenho do algoritmo de treinamento seja adequado. Entre estas informações, pode-se citar:

- o nome do arquivo a ser usado para treinamento: Este é um arquivo texto que contém todos os pares de entrada/saída para realizar o treinamento da rede, ou então, um arquivo com categorias para os pesos;
- taxa de cruzamento: Esta taxa, que deve ser um número entre 1 e 100, indica qual porcentagem de indivíduos da população devem ser cruzados. O número de indivíduos deve ser par, já que estes são cruzados dois a dois.
- taxa de mutação: Esta taxa, que deve ser um número entre 0 e 100, indica a probabilidade de que na população ocorra mutação em um indivíduo a ser escolhido aleatoriamente;
- número de indivíduos: indica quantos indivíduos farão parte da população. Este número é mantido constante ao longo das gerações.
- número de gerações: indica o número de gerações pelas quais os indivíduos devem ser avaliados, servindo como critério de parada.
- número de neurônios: informa o número total de neurônios que compõe a rede que está sendo treinada;
- escolher o tipo de seleção;
- escolher o tipo do cálculo do *fitness*, e os parâmetros envolvidos com o tipo escolhido;
- escolher o tipo de cruzamento;
- escolher o limiar de ativação.

A resposta do sistema é o indivíduo que forneceu melhor valor de *fitness* ao longo das gerações simuladas. Este arquivo é armazenado em um arquivo para posterior consulta e testes. Juntamente com a matriz de conectividade, são armazenadas informações diversas sobre a rede, como o *fitness* relativo ao indivíduo, a geração na qual ele foi encontrado, entre outras informações.

Também pode ser aplicado um arquivo com combinações de parâmetros. Cada linha do arquivo informa os dados necessários para cada simulação. O melhor indivíduo gerado em cada simulação é armazenado para posterior análise.

7.4 Outras Funções

O ambiente computacional permite realizar algumas estatísticas a respeito do funcionamento das redes sendo comparadas. Uma das funções permitidas é aplicar conjuntos de entrada e saída e verificar se as respostas fornecidas coincidem com as respostas desejadas. Também é permitido visualizar gráficos de ativação dos neurônios, verificando como eles se comportam ao longo dos ciclos.

É possível alterar os parâmetros relacionados ao equacionamento das redes IAC, tanto original, como Modelo A, para verificar como variações nestes parâmetros alteram o comportamento da rede como um todo.

Outra função possível é realizar alterações manuais nas matrizes de conectividade. Isto permite que pequenos ajustes possam ser feitos, com base em observações realizadas pelo usuário.

Capítulo 8

Resultados e Discussões

A metodologia, apresentada no Capítulo 5, foi desenvolvida em etapas. A partir de um modelo inicial, os testes, utilizando os estudos de caso apresentados, foram sendo realizados e, quando necessário, foram feitas alterações para melhorar o desempenho da proposta, ou então, para adequá-la a diferentes problemas. Todos os estudos de caso estão modelados através de redes modelo 'A'.

Basicamente, há três etapas principais, que diferem entre si pelos arquivos de treinamento utilizados e pela forma como é feita a geração da população inicial, entre outros. Na primeira etapa, onde foram utilizados pares de entrada/saída como conjunto de treinamento, foi definida a representação e as formas de seleção, cruzamento e mutação. Na segunda etapa, foi definido um novo conjunto de treinamento, que consiste de categorias para os pesos. E na terceira etapa, as categorias são utilizadas para geração da população inicial, e o treinamento é realizado por meio de pares de entrada/saída.

Os valores escolhidos para as taxas de cruzamento e mutação, entre outros testados, foram de 90% e 5%, respectivamente. Estes são usados em todas as simulações apresentadas neste capítulo, por serem aqueles que forneceram melhores resultados. Uma taxa de mutação mais elevada foi utilizada, pois o espaço de busca é muito grande, e é por meio deste operador que as soluções não atingidas pelo cruzamento são alcançadas. Nos casos em que foi necessário realizar o processamento da rede, os valores utilizados para os parâmetros, relacionados ao equacionamento das redes modelo 'A', são aqueles definidos por Rumelhart [42]. Na bibliografia pesquisada, observou-se que a maioria dos autores utiliza estes valores nas suas simulações, não existindo um estudo mais detalhado a respeito do comportamento destes. Isto poderia indicar a necessidade de uma pesquisa mais completa sobre a função que os parâmetros exercem sobre a rede, tanto original, quanto modelo 'A'.

8.1 Etapa 1: Treinamento por Pares de Entrada/Saída

Nesta etapa, foram realizadas diversas definições para a metodologia. Entretanto, os principais esforços se concentraram em propor formas adequadas de representação para o indivíduo, e na criação de opções de funções de *fitness*, quando utilizando pares de entrada/saída para teste. Além disto, foram definidas as formas de cruzamento e seleção.

Um dos exemplos mais utilizados nesta etapa é aquele dos JS, usando 10, 21 e 41 neurônios. O comportamento das matrizes geradas para estes exemplos, ao longo desta seção, é descrito por meio de dois gráficos tais como aqueles apresentados na Figura 8.1, que estão num formato genérico. O primeiro gráfico (8.1(a)) contém as seguintes informações:

- Acertos específicos: Informa, na cor azul, o número de neurônios com estado de ativação correto, para o conjunto de entradas específicas apresentadas.
- Erros específicos: Informa, na cor vermelha, o número de neurônios com estado de ativação incorreto, para o conjunto de entradas específicas apresentadas;
- Acertos gerais: Informa, na cor amarela, o número de neurônios com estado de ativação correto, para o conjunto de entradas gerais apresentadas;
- Erros gerais: Informa, na cor verde, o número de neurônios com estado de ativação incorreto, para o conjunto de entradas gerais apresentadas.

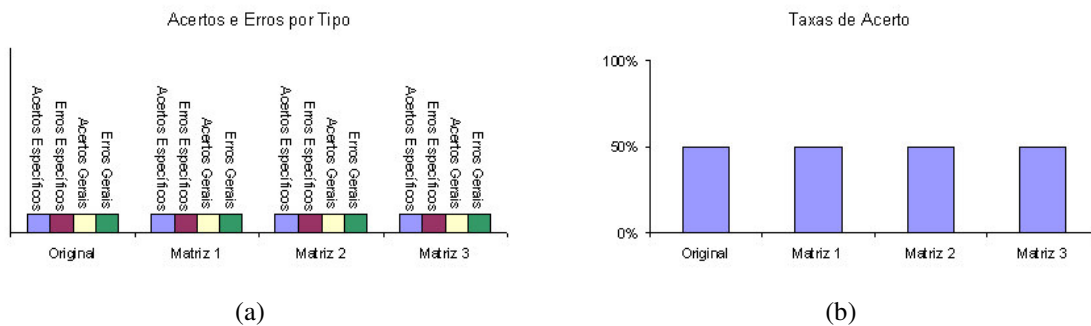


Figura 8.1: Gráfico genérico, utilizado para os exemplos dos Jets e Sharks. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.

Com esta separação entre os números de acertos e erros, pode-se verificar se o conjunto de treinamento, composto pelas entradas específicas, está sendo bem aprendido, e se as entradas gerais podem ser deduzidas a partir das entradas específicas. Neste mesmo gráfico, é apresentado o resultado para diferentes matrizes. A primeira matriz é sempre chamada

de 'Original', e representa a implementação do exemplo dos JS usando a rede IAC original. As matrizes geradas pelo treinamento são chamadas de 'Matriz 1', 'Matriz 2', e assim sucessivamente, dependendo do número de matrizes contidas no gráfico.

Para complementar, o gráfico apresentado na Figura 8.1(b), traz a taxa total de acertos para cada matriz na Figura 8.1(a), informando a porcentagem de neurônios ativados corretamente, considerando tanto as entradas gerais quanto específicas, ou seja, traz informação a respeito do comportamento geral das redes para as matrizes sendo consideradas. Este gráfico traz, também, a taxa de acerto fornecida pela implementação usando a rede IAC original. A comparação com a rede IAC original é importante, já que o exemplo dos JS funciona como um padrão ouro para a metodologia proposta.

8.1.1 Representação dos Indivíduos

Através de um AG simples, que utiliza seleção dos mais aptos e cruzamento em um único ponto, foram realizados testes relacionados à representação. Para medir o desempenho da rede gerada, com base em um conjunto de treinamento composto por pares de entrada/saída, foi utilizado o cálculo do *fitness* por 3 valores. A função de *fitness* simples foi utilizada apenas em testes básicos, por não refletir de forma adequada o comportamento das redes sendo avaliadas.

O estudo de caso escolhido para iniciar os testes foi o dos JS. Todas as entradas, pertencentes a conjuntos de treinamento relacionados a este exemplo, são compostas por um único neurônio ativo, onde cada um corresponde a um neurônio do grupo 'nomes'. Estas entradas são consideradas as entradas específicas para o problema. Portanto, o número de pares é igual ao número de neurônios deste grupo, e depende de qual versão deste exemplo está sendo utilizada. Com relação a quais pesos encontrar e como representá-los, num primeiro momento, definiu-se:

- que o indivíduo seria composto por todos os pesos pertencentes à matriz de conectividade, sem considerar o fato da matriz ser simétrica e ter diagonal 0. O número de pesos Pe a ser encontrado é dado pela Equação 8.1.

$$Pe = M^2 \quad (8.1)$$

onde:

M Número de neurônios da rede

Desta forma, o tamanho do indivíduo TI é dado pela Equação 8.2.

$$TI = Pe * NG \quad (8.2)$$

onde:

NG Número de genes utilizado para representar cada peso

Pe Número de pesos a ser encontrado pelo AG

- a utilização do alfabeto binário, considerando que o número de genes para representar cada peso, dado pela variável NG , é igual a 10. Este alfabeto foi escolhido, pois tem sido extensivamente utilizado, existindo muitos estudos a seu respeito, inclusive com provas de sua convergência, por meio do Teorema de Esquemas [98].

Como visto, uma matriz de conectividade que representa uma rede modelo 'A' é simétrica, e tem diagonal principal nula. Assim, pretendia-se observar se o AG guiaria a população para indivíduos que respeitam estas restrições. Para realizar esta verificação, utilizou-se o exemplo dos JS com 10 neurônios. Esta rede foi treinada com as entradas específicas a respeito do problema, o que resulta num conjunto de treinamento composto por 5 pares, que é o número de neurônios contidos no grupo 'nomes'. Um exemplo de matriz de conectividade obtida é apresentada nas Tabelas 8.1 e 8.2, que traz todos os valores de conexões entre os neurônios desta rede.

Tabela 8.1: Matriz de conectividade para JS com 10 neurônios, obtida quando o indivíduo engloba todos os seus pesos - Parte 1.

	Art	Rick	Sam	Ralph	Lance
Art	0,248046875	-0,6484375	-0,509765625	-0,12109375	-0,830078125
Rick	-0,646484375	-0,021484375	-0,57421875	-0,533203125	-0,291015625
Sam	-0,2109375	-0,236328125	0,64453125	-0,201171875	-0,724609375
Ralph	-0,77734375	-0,126953125	0,5703125	-0,46875	-0,779296875
Lance	-0,67578125	-0,755859375	-0,763671875	-0,822265625	0,83203125
Jets	0,72265625	-0,048828125	0,603515625	0,65625	0,947265625
Sharks	-0,73828125	0,689453125	-0,6171875	-0,583984375	-0,26953125
Pusher	0,640625	-0,181640625	-0,9765625	0,89453125	-0,958984375
Burglar	-0,716796875	0,65625	-0,046875	-0,716796875	0,83203125
Bookie	-0,462890625	0,01171875	0,958984375	0,013671875	-0,9453125

Para verificar a qualidade das respostas fornecidas por esta matriz, foi utilizado um conjunto de testes, que contém, além dos pares utilizados para treinamento, outros 5. Estes, também, são compostos por entradas com um único neurônio ativo, correspondentes às entradas gerais a respeito do problema. Assim, é possível ter uma medida a respeito do comportamento da rede, tanto relacionado às entradas específicas quanto às entradas gerais. Estes mesmos conjuntos de treinamento e testes foram utilizados em todos os exemplos que consideram o problema dos JS com 10 neurônios.

Tabela 8.2: Matriz de conectividade para JS com 10 neurônios, obtida quando o indivíduo engloba todos os seus pesos - Parte 2.

	Jets	Sharks	Pusher	Burglar	Bookie
Art	0,13671875	-0,0625	0,15234375	0,24609375	0,236328125
Rick	-0,412109375	0,46484375	0,2734375	-0,029296875	-0,7734375
Sam	-0,390625	-0,46484375	0,224609375	-0,52734375	-0,13671875
Ralph	0,1953125	-0,69921875	-0,20703125	-0,25390625	0,66015625
Lance	-0,53125	-0,833984375	-0,96875	-0,501953125	0,076171875
Jets	-0,919921875	-0,91015625	0,525390625	-0,646484375	0,1953125
Sharks	-0,435546875	-0,14453125	-0,025390625	-0,212890625	0,92578125
Pusher	-0,25	0,2890625	-0,18359375	-0,251953125	-0,35546875
Burglar	-0,33203125	0,8671875	0,60546875	-0,923828125	-0,619140625
Bookie	-0,255859375	0,1484375	-0,853515625	-0,423828125	0

A matriz das Tabelas 8.1 e 8.2, chamada de 'Matriz 2', foi obtida evoluindo uma população de 130 indivíduos, e, para o conjunto de teste usado, forneceu uma taxa geral de 84% de acerto, conforme observado na Figura 8.2(b). Este indivíduo já foi encontrado na geração 107. Para o mesmo exemplo dos JS com 10 neurônios, implementado utilizando-se a rede IAC original, a taxa de acerto é de 96%, sendo esta matriz chamada de 'Original'. Ou seja, as taxas de acerto fornecida pelas duas matrizes não são próximas.

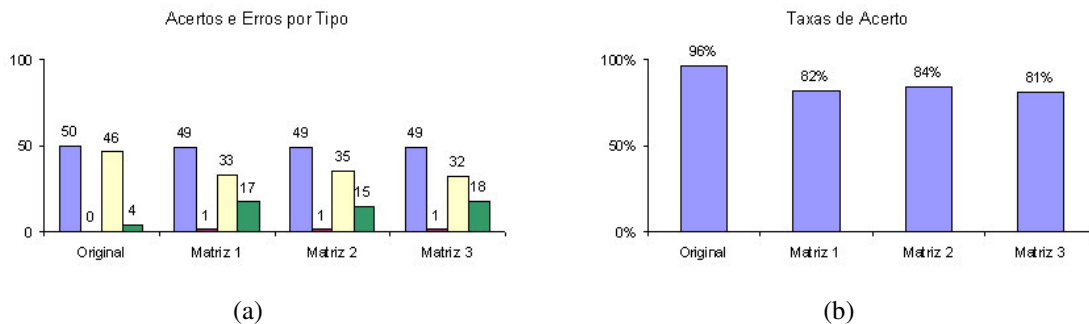


Figura 8.2: JS com 10 neurônios: Resultados obtidos quando são encontrados todos os pesos da matriz de conectividade, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.

Para a mesma rede JS com 10 neurônios, foram realizadas diferentes simulações, aumentando o número de indivíduos e gerações, e as taxas de acerto não apresentaram maiores diferenças. Ou seja, a rede estabiliza nestes valores, e alterações nos parâmetros não provocam melhores resultados.

De acordo com o mesmo gráfico da Figura 8.2(b), outras taxas de acertos obtidas por matrizes geradas por treinamento foram de 82% (Matriz 1) e 81% (Matriz 3). Observa-se pela Figura 8.2(a), que, para o conjunto de treinamento, o número de acertos e erros fornecidos

pelas diferentes matrizes é igual, considerando as entradas específicas, que foram utilizadas como conjunto de treinamento. No caso das entradas gerais, as respostas fornecidas têm poucas diferenças.

Se a taxa de acerto é inferior à esperada em redes pequenas, para redes maiores os resultados fornecidos pioram. Isto pode ser observado pela simulação realizada com o exemplo dos JS com 21 neurônios. Neste caso, a rede também foi treinada com as entradas específicas a respeito do problema, resultando num conjunto de treinamento composto por 10 pares de entrada/saída, correspondente aos neurônios do grupo 'nomes'. O conjunto de teste, seguindo o mesmo padrão, contém, além dos 10 pares de entradas específicas, outros 11 correspondentes às entradas gerais. Estes mesmos conjuntos de treinamento e testes foram utilizados em todos os exemplos que consideram o problema dos JS com 21 neurônios.

Este exemplo, implementado utilizando a rede IAC original, fornece uma taxa de acerto geral de 94,78%, conforme ilustrado pela Figura 8.3(b). Entretanto, as simulações realizadas alcançaram matrizes que fornecem taxas de acerto de 78,91% (Matriz 1), quando utilizando 500 indivíduos. Este valor foi encontrado na geração 307. Outra taxa de acerto conseguida, utilizando 550 indivíduos, foi de 79,14% (Matriz 3), encontrado na geração 288. Utilizando 600 indivíduos, as taxas encontradas foram de 79,37% (Matriz 2), na geração 370. Da mesma forma que o exemplo anterior, o número de erros para as entradas gerais e específicas, fornecidos pelas matrizes geradas por treinamento, são muito próximos, conforme observado na Figura 8.3(a).

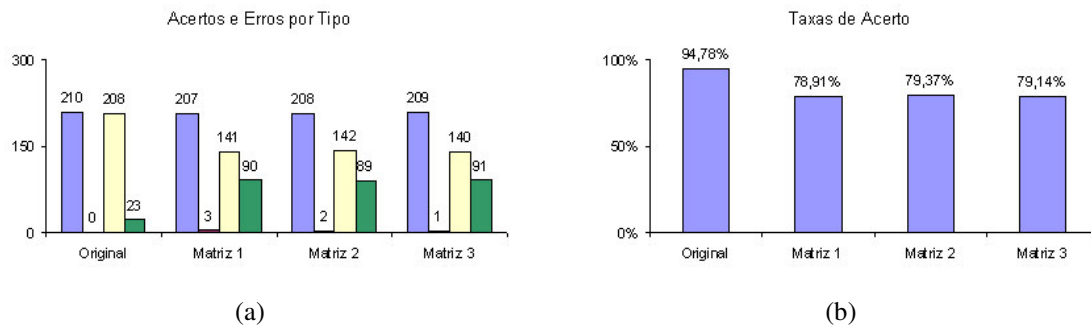


Figura 8.3: JS com 21 neurônios: Resultados obtidos quando são encontrados todos os pesos da matriz de conectividade, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.

Comparadas com as taxas fornecidas pelo modelo original, as taxas obtidas para estes exemplos não foram satisfatórias, utilizando a metodologia considerada até o momento. Isto pode ter ocorrido porque as redes geradas não respeitam as restrições básicas que caracterizam uma rede modelo 'A', ou seja, as propriedades de simetria e de diagonal principal nula.

Além de não alcançar boas taxas de acerto, os indivíduos gerados com esta representação são muito grandes. Por exemplo, para uma rede JS com 10 neurônios, é necessário encontrar 100 pesos, resultando em um indivíduo com 1000 genes. Para redes pequenas, isto não seria um problema, mas, para redes maiores, há uma forte restrição em se utilizar esta proposta, já que o tamanho dos indivíduos e das populações necessários seriam muito grandes, tornando impraticável a sua aplicação. Para melhorar o desempenho do algoritmo, decidiu-se por utilizar restrições conhecidas sobre o problema, restringindo, desta forma, os pesos a serem encontrados. Obtiveram-se, então, as duas possibilidades apresentadas: os pesos abaixo da diagonal principal ou os pesos entre grupos que têm relação.

8.1.1.1 Abaixo da Diagonal Principal

Aplicando esta restrição aos pesos que devem ser encontrados, foi possível agilizar o processo de busca, mesmo utilizando 10 genes para cada peso. Para os exemplos utilizados até o momento, houve um incremento nas taxas de acerto. Neste grupo de simulações, se mantiveram todos os parâmetros utilizados nas anteriores, modificando-se apenas a composição do indivíduo, que traz os valores relativos aos pesos abaixo da diagonal principal. A Figura 8.4(b) mostra taxas de acerto para o exemplo JS com 10 neurônios, comparadas com as respostas fornecidas pela implementação com a rede IAC original, que é de 96%.

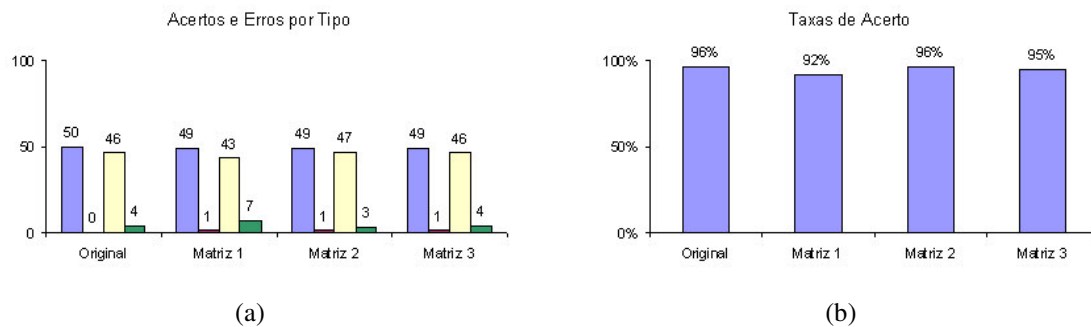


Figura 8.4: JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.

A melhor taxa de acerto obtida a partir de uma matriz gerada por treinamento foi de 96% (Matriz 2), utilizando-se 100 indivíduos. Este valor foi encontrado na geração 52. Outras simulações forneceram matrizes com taxas de acerto de 95% (Matriz 3) e 92% (Matriz 1), usando 130 e 80 indivíduos, respectivamente. Estes valores são bem mais próximos, ou até iguais, àqueles fornecidos pela implementação usando a rede IAC original. Sabe-se que redes neurais, de uma forma geral, não são modelos ótimos, que fornecem 100% de taxas de acertos. Mesmo a implementação utilizando o modelo original não fornece esta taxa.

Observa-se, também, pela Figura 8.4(a) que o número de acertos e erros separadamente foram muito próximos entre si, para as redes geradas.

Para o exemplo dos JS com 21 neurônios, tem-se os resultados observados na Figura 8.5. De acordo com a Figura 8.5(b), a melhor taxa de acerto foi de 85,26% (Matriz 3), obtida na geração 288, utilizando 500 indivíduos. Outros valores conseguidos foram 84,58% (Matriz 2) e 84,35% (Matriz 1), evoluindo 600 e 500 indivíduos, respectivamente. Mesmo com as modificações feitas na metodologia, as taxas de acerto para este exemplo não melhoraram o suficiente, se comparadas com os resultados fornecidos pela implementação utilizando o modelo original. Portanto, outras alterações se fazem necessárias.

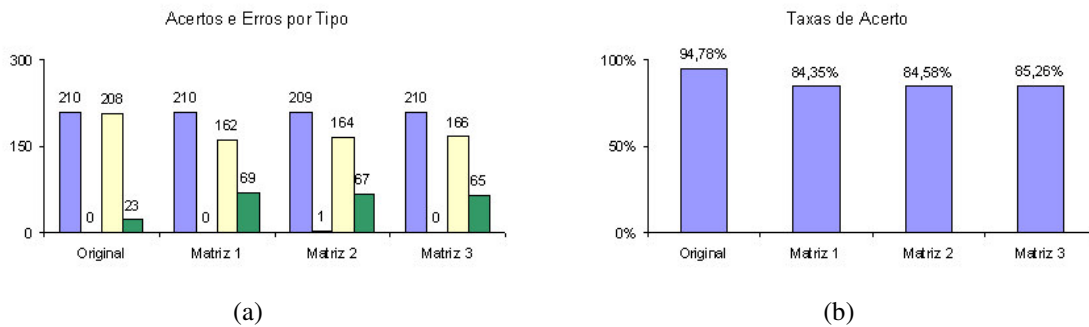


Figura 8.5: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.

Uma consideração realizada foi verificar se o conjunto de treinamento sendo utilizado, composto pelas entradas específicas a respeito do problema, era o mais adequado para o problema em questão. Então, ao invés de serem utilizadas as entradas específicas, foram utilizadas as entradas gerais para treinamento, de maneira a observar se ocorriam melhoras nos resultados fornecidos.

Desta forma, foram criados conjuntos de treinamento composto pelas entradas gerais a respeito do problema, sendo este conjunto composto por 5 pares, no caso do exemplo dos JS com 10 neurônios, e composto por 11 pares, no caso do exemplo dos JS com 21 neurônios. Foram utilizados os mesmos parâmetros das simulações anteriores, incluindo o número de gerações e de indivíduos, sendo o indivíduo composto pelos pesos abaixo da diagonal principal.

Para o exemplo de 10 neurônios, conforme ilustrado na Figura 8.6(b), as taxas de acerto foram reduzidas para 85%, em contrapartida aos 96% fornecidos anteriormente, usando as entradas específicas. Para o exemplo com 21 neurônios, conforme ilustrado na Figura 8.7(b), a taxa de acerto obtida foi de 77,10% (Matriz 3) e 78,46% (Matriz 4), que são valores menores do que as taxas médias de 85% já conseguidas.

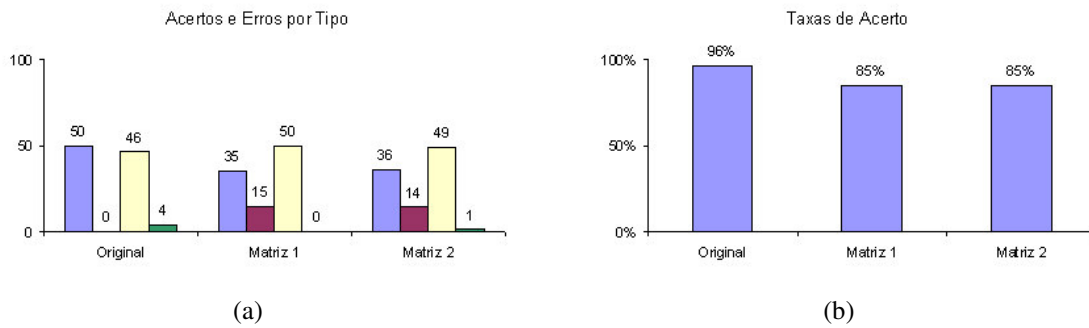


Figura 8.6: JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original, usando entradas gerais para treinamento. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

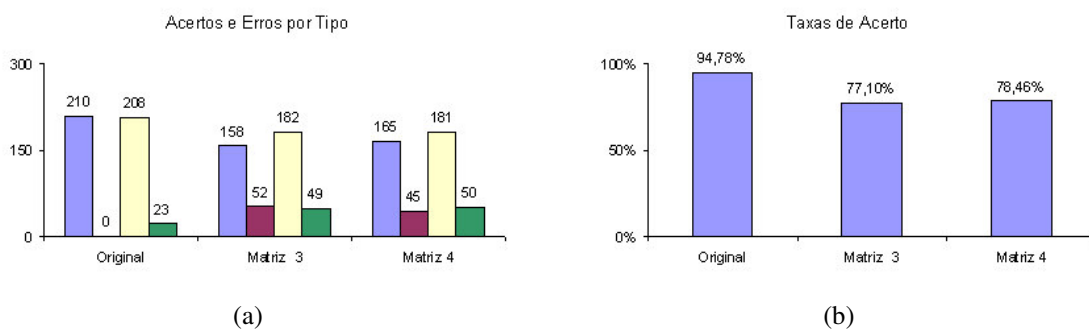


Figura 8.7: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original, usando entradas gerais para treinamento. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxa total de acertos.

Pelas Figuras 8.6(a) e 8.7(a), observa-se que há uma melhora nas respostas para as entradas gerais, mas, em contrapartida, há uma piora nas respostas fornecidas para as entradas específicas, se comparado com os resultados obtidos pelas matrizes das Figuras 8.4(a) e 8.5(a). Os mesmos testes foram realizados utilizando conjuntos híbridos e as respostas não apresentaram melhoras. Observou-se que a informação específica é a que fornece as melhores respostas, sendo mais adequada para o treinamento deste tipo de exemplo.

Voltando às simulações utilizando entradas específicas para treinamento, o algoritmo foi aplicado ao exemplo dos JS com 41 neurônios. Foi utilizado um conjunto de treinamento composto por 27 pares, onde cada entrada continha um neurônio ativo, relativo àqueles pertencentes ao grupo 'nomes'. O tempo gasto com as simulações se tornou muito grande, para que se conseguisse obter resultados razoáveis, dado o tamanho dos indivíduos. O conjunto de testes era composto por 41 pares, seguindo o padrão utilizado nos outros exemplos.

Da mesma forma, os resultados obtidos foram comparados com aqueles fornecidos pela implementação utilizando a rede IAC original, sendo estes resultados apresentados na Figura

8.8. Como pode-se observar pela Figura 8.8(a), as taxas de acerto alcançadas foram de 84,95% (Matriz 2), 85,37% (Matriz 3) e 84,71% (Matriz 4), que consideraram populações de 1000 e 2000 indivíduos. A taxa mais baixa obtida foi de 82,93% (Matriz 1), obtida em uma simulação que considerou apenas 500 indivíduos compondo a população. Estas taxas, ainda se mostram inferiores àquelas fornecidas pela rede IAC original, que é de 91,49%.

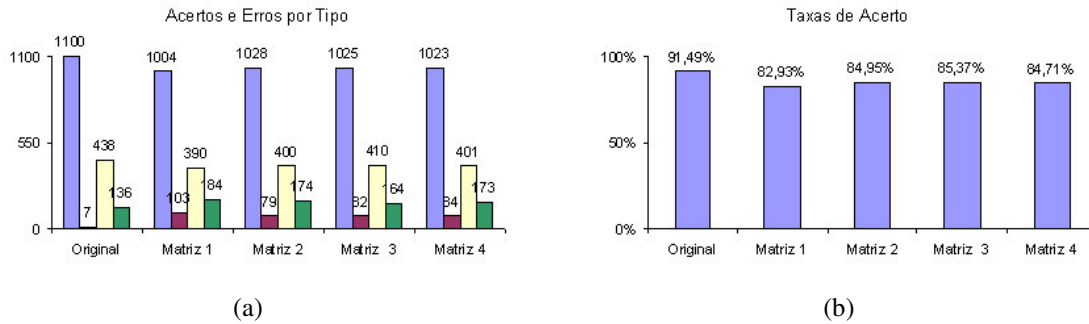


Figura 8.8: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 10 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

A alocação dos vetores necessários ao processamento do AG, para a rede de 41 neurônios, utilizando 10 genes por peso, quando se encontram os pesos abaixo da diagonal principal, torna necessário um computador com recursos de memória razoáveis. Devem ser encontrados 820 pesos, segundo a Equação 5.1, resultando num indivíduo composto por 8200 genes. Desta forma, o processamento se torna lento, e a aplicação da metodologia a redes maiores se torna inviável. Além disso, não é possível alcançar resultados tão bons quanto aqueles fornecidos pela rede IAC original.

Outra questão levantada foi o número de casas decimais geradas para cada peso. Talvez, não fosse necessário tal precisão para a qualidade dos resultados. Sendo assim, decidiu-se usar uma representação com menos genes para representar cada peso, e que, conseqüentemente, fornece menos casas decimais após a vírgula. Portanto, utilizou-se uma representação binária com 6 genes. Ainda assim, os resultados obtidos anteriormente se repetiram, pois a representação ainda era muito grande, e o processamento muito lento. Alguns resultados para os exemplos dos JS com 41 neurônios são apresentados na Figura 8.9(b), sendo possível observar que as taxas de acerto obtidas, que foram de 85,43% (Matriz 1) e 85,78%, se aproximam daquelas já alcançadas, mostradas na Figura 8.8(b).

Para as redes de 10 neurônios e 21 neurônios, utilizando 6 genes por peso, foram obtidos os mesmos valores para as taxas de acerto, conforme observado nas Figuras 8.10 e 8.11, sendo que o número de indivíduos utilizados foi o mesmo das simulações com 10 genes por peso. Entretanto, para a rede de 41 neurônios o tempo de processamento e os vetores necessários para alocação dos dados ainda se mantiveram muito grandes.

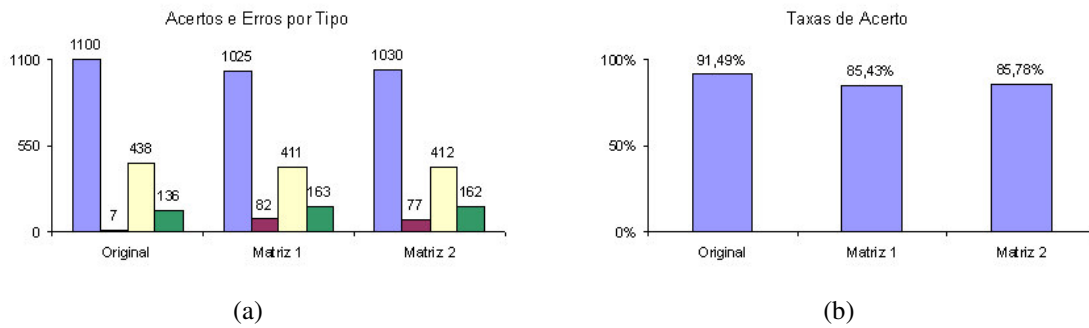


Figura 8.9: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 6 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

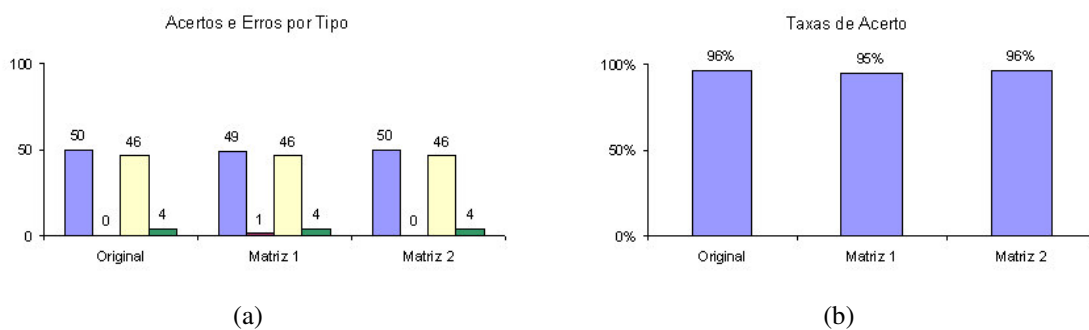


Figura 8.10: JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 6 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

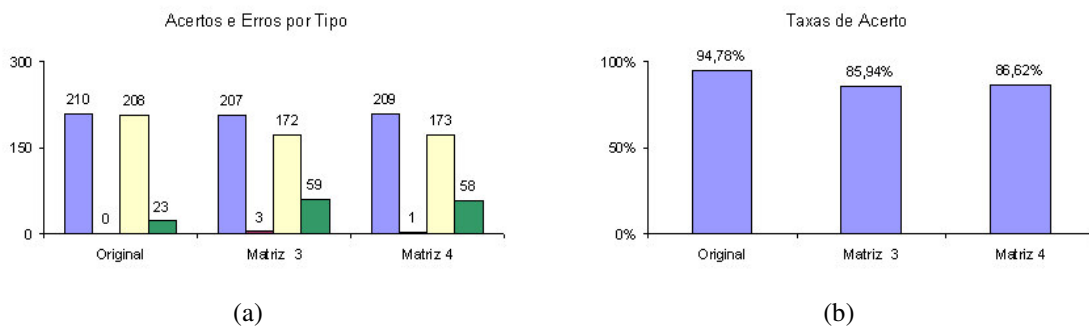


Figura 8.11: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 6 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxa total de acertos.

Observa-se pelas Figuras 8.5(b), que mostra os resultados usando 10 genes por peso, para a rede de 21 neurônios, e 8.11(b), que mostra os resultados usando 6 genes por peso, que houve um incremento mínimo nas taxas de acerto, sendo alcançados valores de até 86,62% (Matriz 4), que não foram alcançadas com o outro exemplo.

Nas representações mencionadas até o momento, os pesos eram escolhidos de maneira completamente aleatória, e esperava-se que a própria convergência levasse a pesos positivos para conexões entre neurônios pertencentes a grupos diferentes e pesos negativos para conexões entre neurônios do mesmo grupo. Como isto não aconteceu, o gene que representava o sinal foi removido, utilizando-se uma representação com apenas 5 genes, para verificar se os resultados fornecidos seriam melhores.

Ao introduzir esta modificação, e realizando o treinamento para a rede JS com 10 neurônios, não foi possível reproduzir a taxa de 96%, alcançada em treinamentos anteriores, atingindo-se uma taxa de acerto máxima de 95% obtida pela matriz chamada 'Matriz 1', conforme observado na Figura 8.12. Outros valores alcançados foram de 91% (Matriz 2) e 94% (Matriz 3).

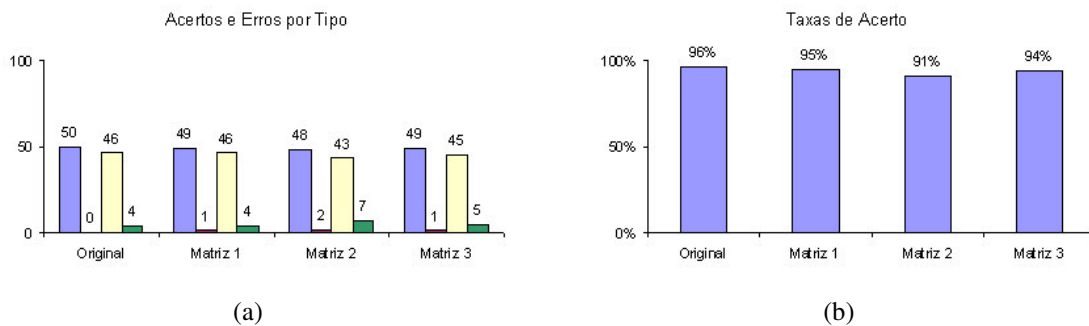


Figura 8.12: JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 5 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

No caso das redes de 21 e 41 neurônios, houve uma queda nas taxas de acertos, e além disso, a convergência se manteve lenta. Os tamanhos de populações utilizados são os mesmos aplicados quando se utiliza a representação com 6 genes. Para a rede de 21 neurônios, conforme observado na Figuras 8.13(b), foram obtidos taxas de acerto de 73,24% (Matriz 1), 75,28% (Matriz 2) e 72,34% (Matriz 3), que são bem inferiores àquelas de até 86% já alcançadas. E no caso da rede de 41 neurônios, conforme observado na Figura 8.14, as taxas foram de 57,88% (Matriz 1), 51,64% (Matriz 2) e 59,73% (Matriz 3), também inferiores àquelas de até 85,78% já obtidas.

Neste caso, aumentar o número de gerações e indivíduos não trouxe benefícios no comportamento geral da rede. Os resultados obtidos com a representação de 5 genes, que forçam o sinal a ser positivo ou negativo, podem ser um indicativo de que é mais interessante permitir ao treinamento uma maior liberdade na escolha dos sinais para os pesos.

Desta forma, foi possível concluir que, em se tratando de indivíduos muito grandes, a representação binária pode não ser a melhor escolha, apesar dos diversos estudos positivos

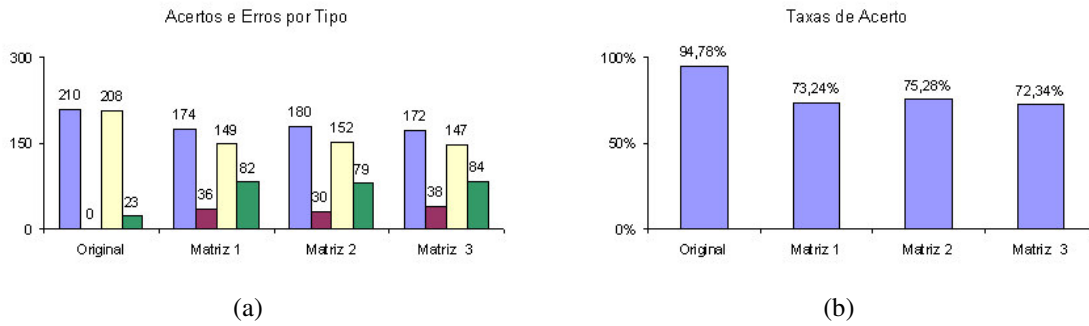


Figura 8.13: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 5 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

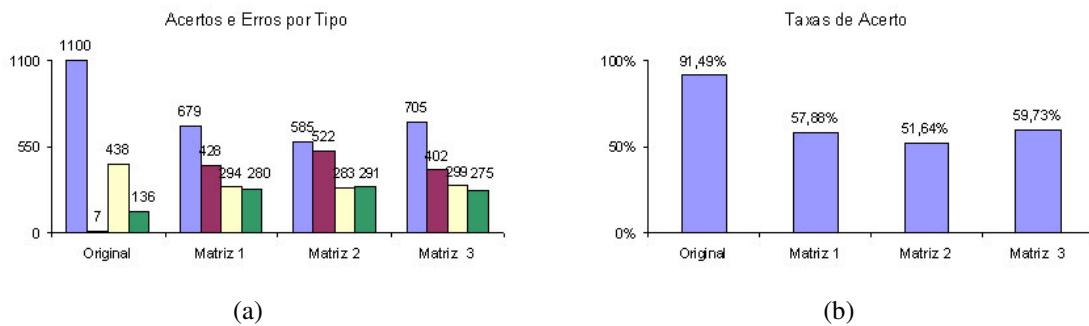


Figura 8.14: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto binário com 5 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

a seu respeito. Portanto, optou-se por uma representação inteira, como alternativa para a binária, podendo ser utilizados 1 ou 2 genes para cada peso. Para comparar as duas abordagens, foram feitos novos testes que utilizavam o mesmo sistema, modificando apenas o alfabeto utilizado, que de binário, passou a ser inteiro, e fazendo as alterações mínimas necessárias para o sistema ficar compatível com a nova representação.

Os testes mostraram que, comparativamente, as matrizes fornecidas com o alfabeto inteiro apresentaram respostas semelhantes àsquelas fornecidas pela representação binária, necessitando, porém, de um tempo menor de processamento, devido à diminuição no tamanho dos vetores, permitindo a aplicação a redes maiores.

A representação com inteiros reduziu drasticamente o tamanho das matrizes que contêm as populações, acelerando a velocidade de convergência do problema. Para a rede JS com 10 neurônio, foi possível alcançar valores de 96%, conforme apresentado na Figura 8.15(b), pela Matriz 1. Outro valor obtido foi de 95% (Matriz 2).

Para as redes de 21 e 41 neurônios, que tem alguns resultados apresentados nas Figuras 8.16 e 8.17, respectivamente, os resultados foram praticamente os mesmos daqueles obtidos

com a representação de 6 genes. O número de indivíduos contidos na população foram os mesmos.

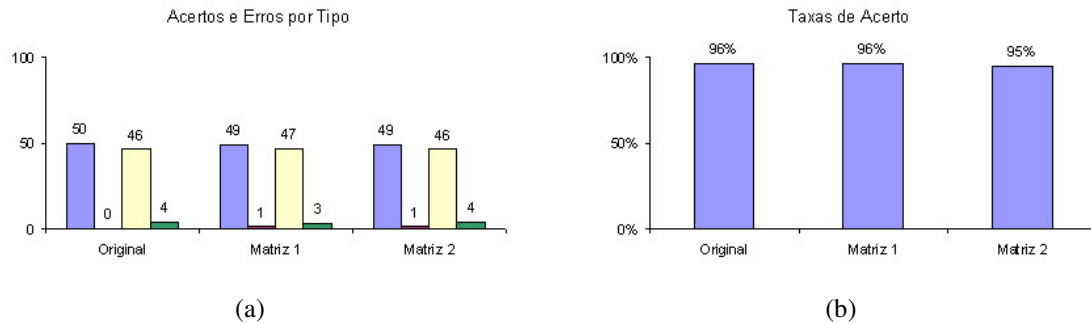


Figura 8.15: JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

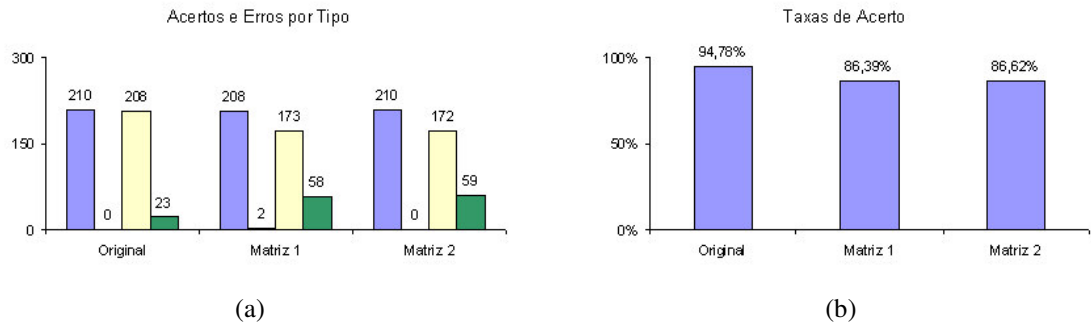


Figura 8.16: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

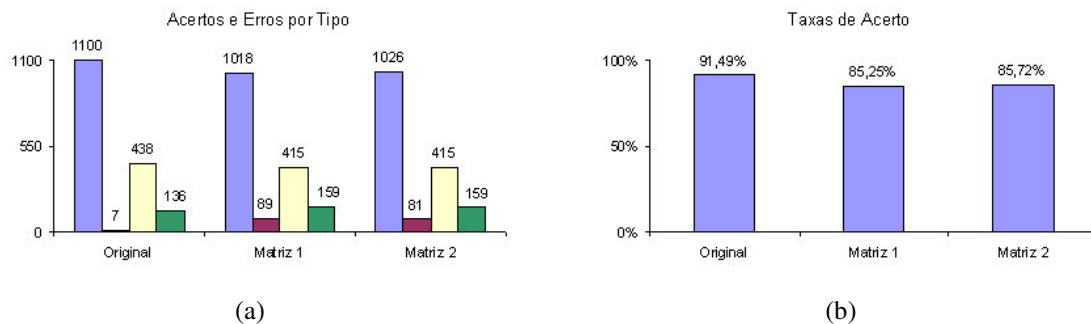


Figura 8.17: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

Com relação à precisão, esta pode depender do problema que está sendo utilizado. Utilizando os exemplos dos JS, foi realizado um teste utilizando dois genes para representar cada peso. Alguns dados obtidos utilizando-se dois genes por peso, para as redes de 10, 21 e 41 neurônios, são apresentados na Figura 8.18, 8.19, 8.20, respectivamente. Para a rede de 10 neurônios, os resultados se mantêm, comparando os gráficos das Figuras 8.15(a) e 8.18(a). No caso das redes de 21 e 41 neurônios, verifica-se um pequeno incremento na qualidade dos resultados. Comparando-se os gráficos das Figuras 8.16(a) e 8.19(a), verifica-se que para a rede de 21 neurônios, foram atingidas taxas superiores a 87% (Matriz 1 e 2) quando usando dois genes, não alcançadas utilizando um único gene para representar cada peso. E no caso da rede de 41 neurônios, comparando os gráficos das Figuras 8.17(a) e 8.20(a), observa-se que se alcançaram taxas maiores de 86% (Matriz 2), também não obtidas quando utilizando um único gene, onde se alcançaram taxas máximas de 85,72%. Embora haja uma diferença, esta é pequena. Assim, para o exemplo dos JS, a representação com uma casa decimal demonstrou ser suficiente, sendo que a utilização de dois genes trouxe pouca melhora para a resposta, sem muito incremento no tempo de processamento. Já 3 ou mais genes não trouxeram benefícios e apenas aumentaram o tempo de processamento.

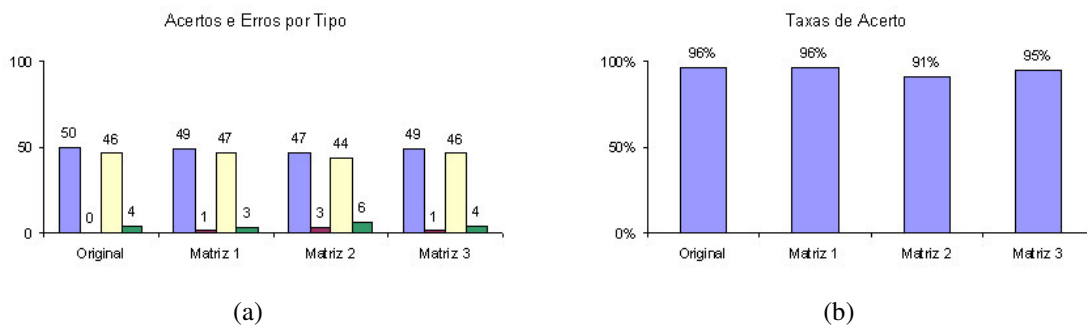


Figura 8.18: JS com 10 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 2 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

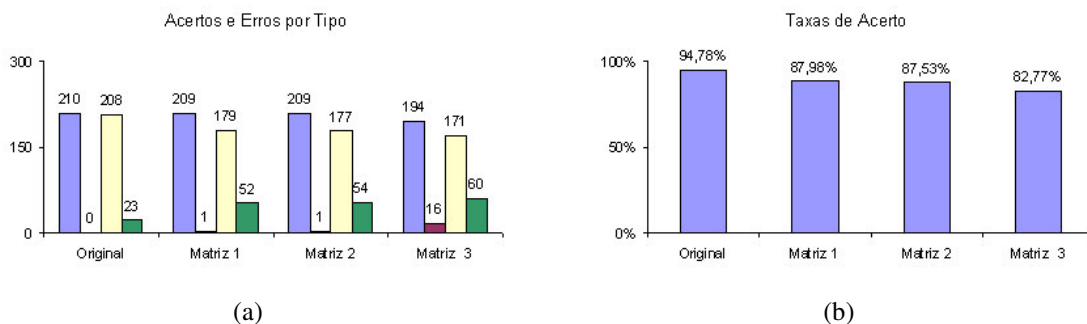


Figura 8.19: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 2 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

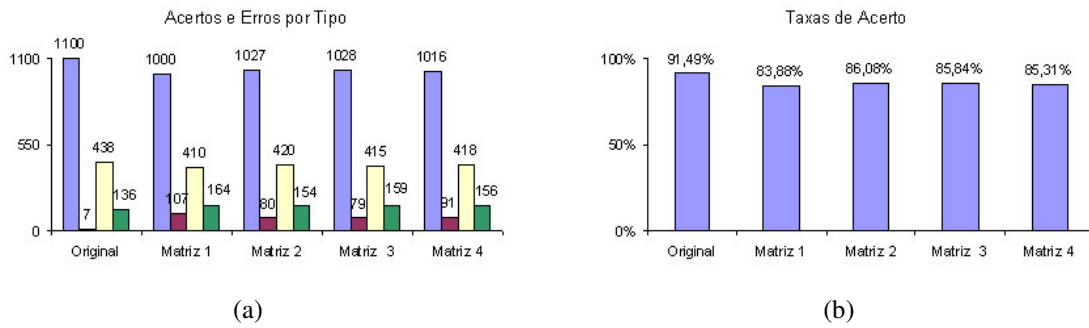


Figura 8.20: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 2 genes por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

Nesta abordagem, é permitido que existam pesos negativos entre neurônios de grupos diferentes. Já conexões positivas para neurônios do mesmo grupo não se mostraram interessantes. Normalmente, organiza-se os grupos, para no caso extremo, eles não interferirem um no outro, ou seja, conexão zero.

Com relação à precisão, foi também testada a codificação real. Observou-se que por esta representação, existe a possibilidade de se obter resultados semelhantes àqueles conseguidos utilizando-se a representação inteira.

Apesar dos resultados terem melhorado ao se encontrar os pesos abaixo da diagonal principal, além de se observar uma evolução mais rápida, ainda se observou a necessidade de procurar outras restrições para melhorar o desempenho da rede.

8.1.1.2 Grupos que têm relação

Uma outra hipótese foi inserir as restrições que consideram os grupos que têm relação e pré-definir os pesos das conexões intragrupo, quando este grupo representar um 'conceito' com 'valores' mutuamente excludentes. A representação binária, por exigir muitos recursos computacionais fica restrita a redes pequenas, fornecendo bons resultados, desde que se use um conjunto de treinamento adequado, assim como em outros modelos de rede. A definição de um conjunto de treinamento é feita de forma empírica, e é dependente do problema que se quer resolver. No caso do exemplo dos JS, o melhor conjunto de treinamento é aquele que representa o conhecimento específico a respeito do problema.

Para melhorar as respostas das redes de 41 neurônios, foi necessário utilizar novas restrições, e encontrar os pesos entre grupos que têm relação. Alguns resultados obtidos são apresentados na Figuras 8.21 e 8.22, para o problema dos JS com 21 e 41 neurônios, respectivamente.

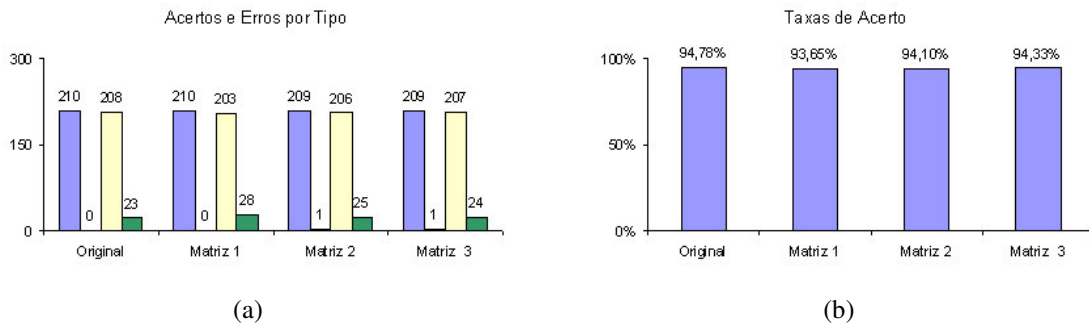


Figura 8.21: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

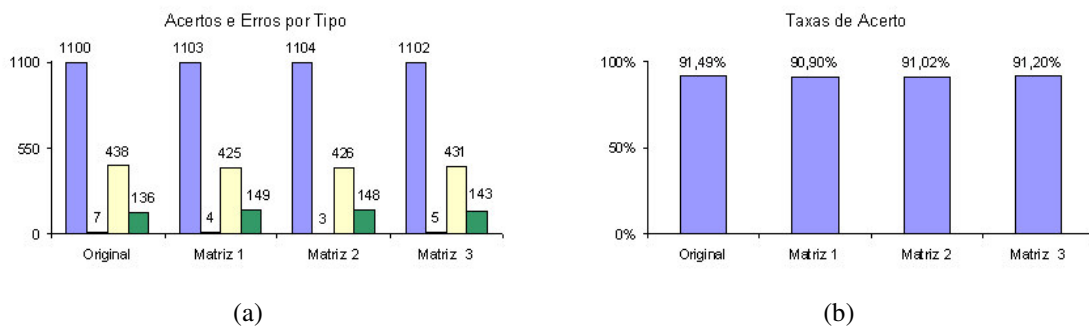


Figura 8.22: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

Utilizando-se esta nova representação para o treinamento foi possível obter uma rede que fornece taxas de acerto muito semelhantes àsquelas fornecidas pelo modelo original, tanto para a rede de 21 neurônios, como para a rede de 41 neurônios. Para a rede de 41 neurônios, foi atingida a taxa de acerto de 91,20% (Matriz 3), que é próxima do resultado fornecido pela original (91,49%), e bem melhor do que a melhor taxa conseguida anteriormente, que era de 86,08% (Matriz 2 da Figura 8.20(b)). Para a rede de 21 neurônios, foi atingida a taxa máxima de 94,33% (Matriz 3 da Figura 8.21(b)), que também é bem próxima da taxa fornecida pela implementação utilizando a original (94,76%). E, além disso, é maior do que a taxa máxima alcançada anteriormente de 87,98% (Matriz 1 da Figura 8.19(b)).

Na Tabela 8.3, é apresentado o número de acertos ativos e o número de acertos inativos, para a matriz apresentada na Tabela 8.4, onde estão apresentados os valores de uma matriz de conectividade obtida por treinamento, correspondente à submatriz C, considerando uma rede JS com 41 neurônios. Como todos os grupos representam 'conceitos' com 'valores' mutuamente exclusivos, os pesos das conexões intragrupos não precisam ser encontrados.

Tabela 8.3: Taxas de acertos ativos e inativos para a rede apresentada na Tabela 8.4.

	Original	Matriz 1
Acertos Ativos	67,45%	67,19%
Acertos Inativos	98,54%	98,46%

Tabela 8.4: Submatriz C, para uma rede JS com 41 neurônios.

	Jets	Sharks	20	30	40	HS	JH	Coll	Solt	Cas	Div	Push	Burg	Book
Art	0,4	0,2	0,2	0	0,6	0,4	0,1	0,3	0,4	0,3	0	0,9	0,3	0,7
Al	0,4	0,2	0,3	0,5	0,1	0,7	0,2	0	0,4	0,6	0,1	0,2	0,4	0,3
Sam	0,6	0,2	0,6	0,1	0,1	0,3	0,5	0,8	0,5	0,4	0	0,1	0,1	0,8
Clyde	0,5	0,3	0,1	0	0,8	0,8	0,4	0,3	0,5	0,2	0,3	0,5	0,4	0,8
Mike	0,5	0,4	0,1	0,7	0,3	0,7	0,3	0,2	0,9	0,4	0,4	0,1	0,1	0,4
Jim	0,8	0,1	0,4	0,3	0,1	0,6	0,3	0	0	0,3	0,4	0	0,4	0
Greg	0,7	0,4	0,9	0,3	0	0,3	0,6	0,2	0,1	0,7	0,5	0,6	0,2	0
John	0,5	0,4	0,5	0,2	0,1	0,9	0,4	0,6	0,3	0,4	0	0,4	0,8	0,3
Doug	0,7	0,4	0,2	0,5	0	0	0,6	0,1	0,6	0,1	0,4	0,1	0,1	0,4
Lance	0,5	0,3	0,5	0,4	0,2	0,6	0,3	0	0,1	0,7	0,1	0,1	0,6	0
George	0,4	0	0,6	0,2	0,5	0,6	0,5	0,3	0,1	0,1	0,8	0,4	0,5	0,4
Pete	0,5	0,2	0,8	0,5	0	0,2	0,8	0,4	0,7	0,3	0,1	0	0,3	0,5
Fred	0,7	0,2	0,5	0,1	0,1	0,1	0,5	0,1	0,8	0,4	0,4	0,5	0	0,1
Gene	0,8	0,3	0,4	0,1	0	0,2	0,3	0,5	0,8	0,4	0,2	0,4	0,1	0
Ralph	0,6	0,5	0	0,7	0,4	0,9	0,3	0,2	0,5	0,4	0	0,8	0,1	0,2
Phil	0,2	0,5	0,4	0,6	0,4	0,4	0,4	0,5	0	0,4	0,3	0,5	0,1	0
Ike	0,4	0,7	0,6	0,7	0,1	0,7	0,4	0,5	0,7	0,6	0,3	0,2	0,2	0,4
Nick	0,3	0,6	0,4	0,5	0,1	0,5	0,7	0	0,5	0	0,2	0,6	0	0,1
Don	0,1	0,4	0	0,7	0,2	0,4	0,1	0,9	0,4	0,5	0,1	0,4	0,8	0
Ned	0,1	0,6	0,1	0,6	0	0,2	0	0,5	0,1	0,5	0,4	0,5	0,5	0,8
Karl	0,4	0,5	0,3	0,4	0,6	0	0,9	0,3	0,1	0,6	0,2	0,2	0	0,4
Ken	0,3	0,7	0,4	0,3	0,2	0,4	0,8	0,4	0,7	0,4	0	0,2	0,5	0
Earl	0,4	0,6	0,4	0,4	0,7	0	0,5	0	0,1	0,9	0,1	0,3	0,7	0,3
Rick	0	0,5	0	0,6	0,2	0	0,5	0,4	0,3	0,3	0,9	0,2	0,9	0,3
Ol	0,1	0,7	0,1	0,4	0,1	0,1	0,2	0,7	0,4	0,5	0	0,8	0,5	0,4
Neal	0,4	0,8	0,4	0,9	0,1	0,1	0,8	0,3	0,5	0,3	0	0,2	0,2	0,6
Dave	0	0,5	0,2	0,8	0,1	0,1	0,8	0,5	0,2	0,2	0,4	0,4	0,2	0

As taxas de acerto ativos e inativos podem ser subdivididas para entradas específicas e gerais, conforme mostrado na Tabela 8.5, que traz o número de neurônios que tem estado de ativação tanto correto quanto incorreto para as entradas gerais e específicas. Observando esta mesma tabela, verifica-se que a rede acertou praticamente todos os padrões apresentados no treinamento, sendo que o número de erros apresentado para os padrões gerais é aproximadamente o mesmo fornecido pela rede original, indicando que a rede responde de forma semelhante tanto para os padrões utilizados no treinamento, como para os padrões gerais usados nos testes.

Tabela 8.5: Resultados parciais, considerando as entradas específicas e as entradas gerais.

	Original	Matriz1
Acertos Ativos Específicos	162	162
Erros Ativos Específicos	0	0
Acertos Inativos Específicos	938	942
Erros Inativos Específicos	7	3
Acertos Ativos Geral	95	94
Erros Ativos Geral	124	125
Acertos Inativos Geral	343	338
Erros Inativos Geral	12	17

A Tabela 8.6 mostra a taxa de acertos e erros geral das duas redes, que são muito próximas, considerando as entradas gerais e específicas em conjunto.

Tabela 8.6: Resultados obtidos - Geral.

	Original	Matriz 1
Total de Acertos	91,49%	91,37%
Total de Erros	8,51%	8,63%

Mesmo fazendo com que o AG evoluísse por mais gerações ou mesmo aumentando o tamanho da população, nenhuma rede que respondesse muito melhor do que o modelo original implementado por Rumelhart foi obtida. Considerando a representação entre grupos que têm relação, foi realizado novamente um teste utilizando as entradas gerais para treinamento, aplicado tanto à rede de 21 neurônios (11 pares), quanto para a rede de 41 neurônios (14 pares).

A qualidade geral das respostas da rede diminuiu, quando utilizada esta forma de treinamento, onde a rede passa a acertar mais para as entradas genéricas, mas acerta menos para as específicas e no quadro geral, como observado pelas Figuras 8.23 e 8.24, que traz resultados para as redes de 21 e 41 neurônios, respectivamente. Da mesma forma, utilizando-se um conjunto de padrões de treinamento com entradas mistas, as taxas de acerto diminuíram, sendo as entradas mistas aquelas que forneceram os piores resultados.

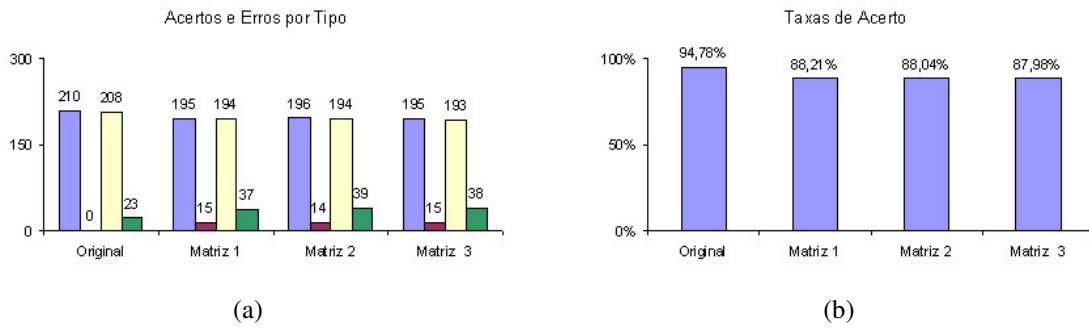


Figura 8.23: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original, treinados com as entradas gerais. Alfabeto binário com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

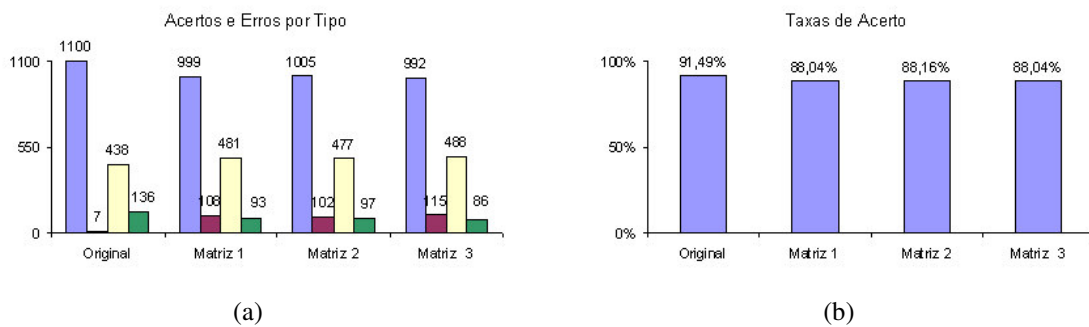


Figura 8.24: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original, treinados com as entradas gerais. Alfabeto binário com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

8.1.2 Cálculo do *Fitness*

A primeira função de *fitness* idealizada realizava um cálculo simples da quantidade de acertos fornecidos por uma matriz. Analisando melhor o problema, observou-se que esta era uma solução que não conseguiria guiar a população para uma resposta adequada, já que considerava simplesmente um valor limite, acima do qual o neurônio era considerado ativo, e abaixo do qual era considerado inativo. Desta forma, uma hipótese mais interessante seria utilizar faixas de valores que considerem valores intermediários de ativação, que foi a forma utilizada nos testes até o momento, chamada de 'Cálculo por 3 valores'.

Outro ponto levantado está relacionado ao fato de que, na maioria dos conjuntos de treinamento sendo avaliados, havia uma quantidade muito maior de neurônios que deveriam estar inativos, do que neurônios que deveriam estar ativos, principalmente quando se tratava da rede com 41 neurônios. Com esta constatação, resolveu-se ponderar os acertos ativos e os acertos inativos, de maneira que os acertos ativos tivessem uma influência maior

no valor de *fitness* final.

Como exemplo, observe-se o exemplo dos JS com 41 neurônios. Para cada entrada contida no padrão de teste, deveria haver 6 neurônios ativos e 35 neurônios inativos. Assim, como são 27 padrões de entrada (mesmo número de neurônios no grupo nomes), existem 945 neurônios que devem estar inativos, e apenas 162 que devem estar ativos.

Portanto, passou-se a somar todos os acertos ativos separados dos acertos inativos, e então, é dado o mesmo valor de peso para os dois tipos de acertos. Assim, os acertos ativos, que são em menor número, recebem um maior valor de ponderação. Alguns resultados obtidos para as redes de 21 e 41 neurônios são apresentadas nas Figuras 8.25 e 8.26, respectivamente. Pode-se observar que as taxas de acerto obtidas são próximas daquelas conseguidas utilizando cálculo do *fitness* por 3 valores, podendo ser observada uma pequena melhora, como na rede de 41 neurônios, onde foram alcançadas taxas de até 91,85% (Matriz 2 da Figura 8.26(b)).

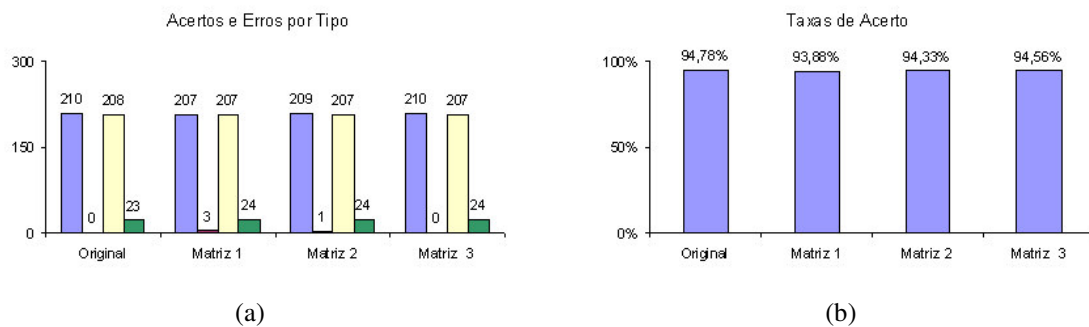


Figura 8.25: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando a função de *fitness* que pondera ativos e inativos, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

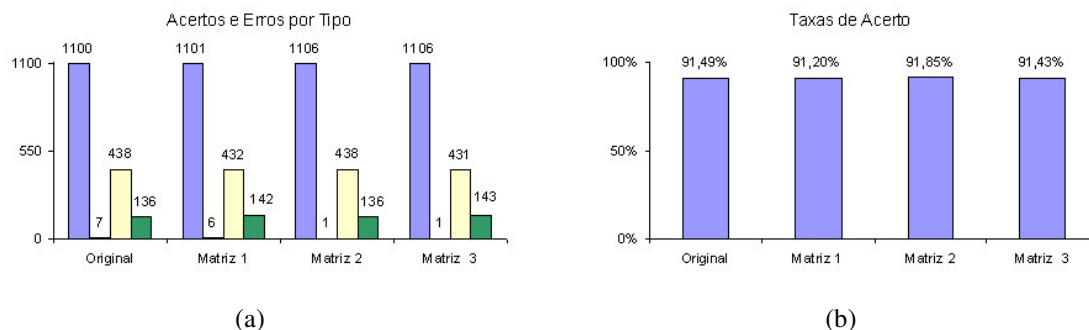


Figura 8.26: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando a função de *fitness* que pondera ativos e inativos, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

8.1.3 Cruzamento

Além do cruzamento em um único ponto, outras formas foram testadas, que são o cruzamento de dois pontos considerando o indivíduo circular e o cruzamento de dois pontos, sem considerar o indivíduo circular.

Utilizando o cruzamento de dois pontos, com indivíduo circular, para as redes de 21 e 41 neurônios se obtém os resultados ilustrados nas Figuras 8.27 e 8.28, respectivamente, onde se observa uma piora da qualidade das respostas fornecidas. Houve um decréscimo na qualidade fornecida para o conjunto de treinamento com as entradas específicas, o que resulta numa piora para o conjunto de teste, que utiliza as entradas gerais.

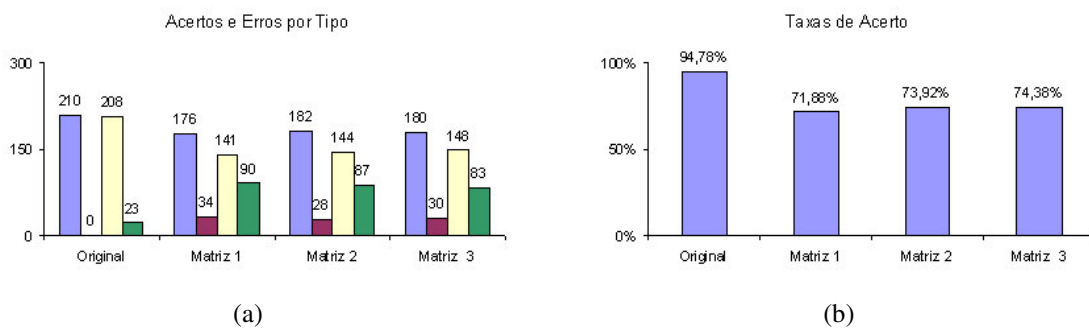


Figura 8.27: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando cruzamento de dois pontos, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

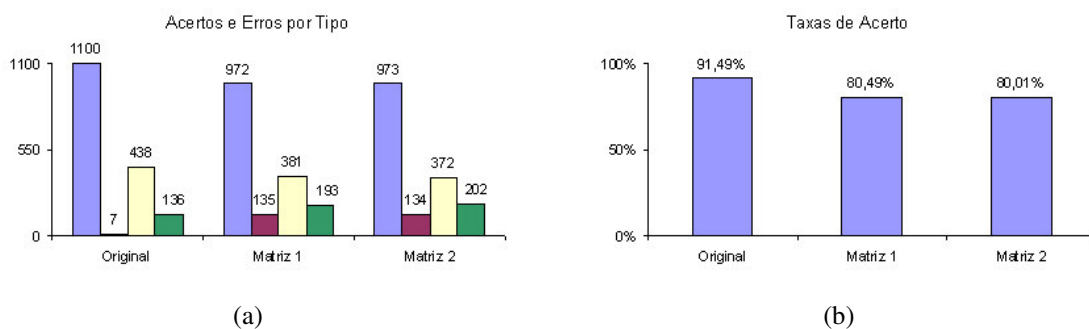


Figura 8.28: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando cruzamento de dois pontos, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

O cruzamento em dois pontos, mas sem considerar o indivíduo circular, não traz nenhum benefício em relação aos resultados fornecidos pela rede, como pode-se observar pelas Figuras 8.29 e 8.30, que trazem os resultados, usando este tipo de cruzamento, para as redes de 21 e 41 neurônios, respectivamente, onde os resultados se mantiveram estáveis.

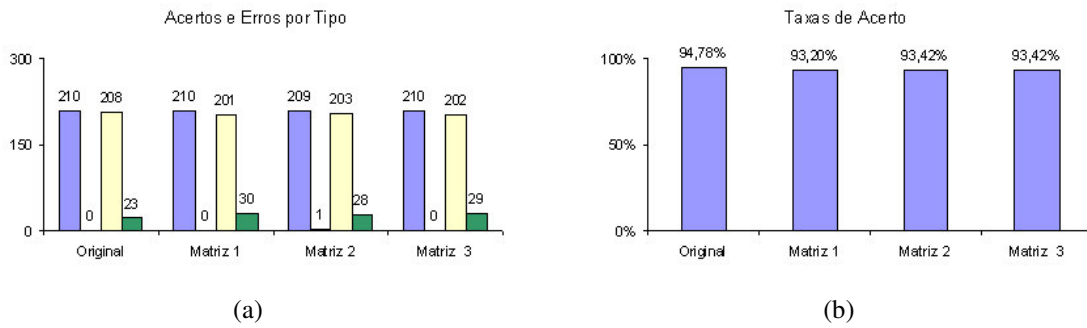


Figura 8.29: JS com 21 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando cruzamento de dois pontos, sem considerar o indivíduo circular, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

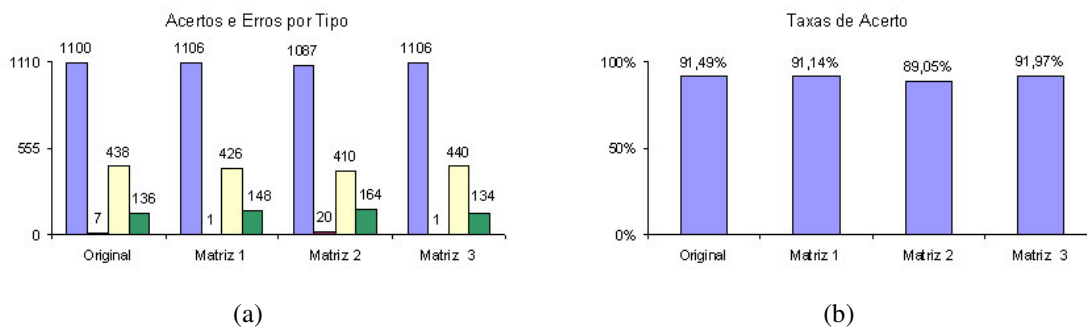


Figura 8.30: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação, usando cruzamento de dois pontos, sem considerar o indivíduo circular, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

Uma outra tentativa realizada foi utilizar mais pontos de cruzamento. Observou-se, porém, que o treinamento utilizando esta proposta não trouxe benefícios para rede, mesmo quando os indivíduos analisados são grandes.

8.1.4 Considerações

Baseado em todas as propostas apresentadas, é possível fazer algumas conclusões relacionadas às combinações que fornecem melhores resultados, considerando a rede de 41 neurônios.

No contexto geral, os melhores resultados foram obtidos pelas seguintes combinações:

- 1 ou 2 genes para representar cada peso, considerando os pesos entre neurônios que têm relação, cálculo do *fitness* pelo método dos 3 valores, método de seleção dos mais aptos e cruzamento em um único ponto.

- 1 gene para representar cada peso, considerando os pesos entre neurônios que têm relação, cálculo do *fitness* pelo método de ponderar os valores ativos e inativos, método de seleção dos mais aptos e cruzamento em um único ponto.
- 1 ou 2 genes para representar cada peso, considerando os pesos entre neurônios que têm relação, cálculo do *fitness* pelo método dos 3 valores, método de seleção dos mais aptos e cruzamento em dois pontos com indivíduo não circular.
- 1 gene para representar cada peso, considerando os pesos entre neurônios que têm relação, cálculo do *fitness* pelo método de ponderar os valores ativos e inativos, método de seleção dos mais aptos e cruzamento em dois pontos com indivíduo não circular.

Os piores resultados, que forneceram taxas de erro mais altas:

- 1 ou 2 genes para representar cada peso, considerando os pesos abaixo da diagonal principal, cálculo do *fitness* pelo método dos 3 valores, método de seleção dos mais aptos e cruzamento em dois pontos, considerando o indivíduo circular.
- 1 ou 2 genes para representar cada peso, considerando os pesos abaixo da diagonal principal, cálculo do *fitness* pelo método de ponderar os valores ativos e inativos, método de seleção dos mais aptos e cruzamento em dois pontos, considerando o indivíduo circular.

Observa-se que representando cada peso utilizando um gene ou dois genes proporciona resultados muito semelhantes, obtendo-se redes com performances semelhantes. Isto significa, que neste caso, utilizar apenas uma casa decimal após a vírgula para cada peso já é suficiente, não necessitando de uma resolução maior. A partir destas conclusões, iniciaram-se os testes com um novo exemplo: diagnóstico diferencial de Hepatite.

8.1.5 Testes com o Exemplo de Diagnóstico Diferencial em Hepatite

Os testes realizados indicam que a metodologia, da forma como está definida até o momento, é capaz de gerar um conjunto de pesos para as redes dos JS de 10, 21 e 41 neurônios, que fornecem resultados tão bons quanto aqueles fornecidos pelas implementações utilizando a rede IAC original. Contudo, estes exemplos apresentam 'conceitos' com 'valores' mutuamente exclusivos, e as relações entre estes 'valores', quando existem são totais. Desta forma, escolheu-se um outro exemplo, que trata do diagnóstico diferencial de Hepatite. Neste problema, alguns 'conceitos' têm 'valores' não mutuamente exclusivos e,

além disto, as relações entre eles podem assumir diferentes graus. Problemas com este tipo de característica são o grande motivador do desenvolvimento de redes modelo 'A', e, dessa forma, o motivador deste trabalho.

Como este é um problema onde os neurônios podem estar mais forte ou fracamente relacionados, fica difícil estabelecer os pesos que estas conexões assumem, o que justifica o uso de um mecanismo que realize esta tarefa. As características assumidas por este exemplo acabam por dificultar a criação dos conjuntos de treinamento e testes. O grupo 'diagnóstico', que compõe as entradas específicas para o problema, contém apenas dois neurônios, não sendo suficiente para definir um conjunto de treinamento. Então, outra forma de definir este conjunto foi elaborada, para que seja possível transmitir à rede as relações existentes entre os neurônios. O novo conjunto de treinamento contém pares que informam, para cada neurônio pertencente à rede que está ativo isoladamente na entrada, quais os neurônios que devem estar ativos após o processamento. O conjunto foi montado com base em [86].

Os conjuntos de testes utilizados para este exemplo também são diferentes daqueles usados com o problema dos JS. Foram utilizadas, principalmente, entradas compostas por um ou mais neurônios ativos, simulando casos clínicos, verificando se o diagnóstico fornecido é compatível com o esperado. Convém ressaltar que ao ativar um conjunto de sintomas, o que importa é o diagnóstico que a rede apresenta. Embora, ao se utilizar este tipo de rede em situações práticas, como no auxílio ao diagnóstico, a ativação de outros neurônios, a partir daqueles de entrada podem servir como um indicativo de quais outros sintomas são possíveis de se encontrar naquele paciente.

Inicialmente, foram utilizados os exemplos de Diagnóstico de Hepatite com 13, 14 e 17 neurônios. Além dos casos clínicos, a rede também foi consultada com entradas com um único neurônio ativo, representando as possibilidades de diagnóstico, para verificar se a rede recupera os sintomas associados.

A Tabela 8.7 traz o conjunto de teste utilizado para o problema de Hepatite com 13 neurônios, onde o número de entradas é igual a 10. A entrada E_1 , por exemplo, consulta a rede para verificar os sintomas e exames associados à Hepatite A. Já a entrada E_3 informa que o paciente apresenta sintomas que não estão associados a um quadro de hepatite (febre alta e ausência de vômitos). Assim, espera-se que nenhuma das duas possibilidades de diagnóstico seja ativada. No caso da entrada E_4 , são ativados sintomas característicos de Hepatite, tanto A quanto B, que são vômitos e febre média. Neste caso, espera-se que os dois diagnósticos sejam ativados com valores de ativações semelhantes.

Para os exemplos com 14 e 17 neurônios, foram criados conjuntos de 14 e 17 pares, respectivamente, seguindo a mesma linha do exemplo mostrado na Tabela 8.7. Estes dois exemplos trazem o grupo 'fatores de risco', que é um grupo com neurônios não mutuamente

exclusivos.

Tabela 8.7: Conjunto de teste para o caso de Hepatite com 13 neurônios.

Grupos	Neurônios	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	E_{10}
Diagnóstico	Hepatite A	1	0	0	0	0	0	0	0	0	0
	Hepatite B	0	1	0	0	0	0	0	0	0	0
Vômitos	Sim	0	0	0	1	1	1	1	1	1	1
	Não	0	0	1	0	0	0	0	0	0	0
Febre	Alta	0	0	1	0	0	0	0	0	0	0
	Média	0	0	0	1	1	1	1	1	1	1
	Baixa	0	0	0	0	0	0	0	0	0	0
Anti HVA IgM	Positivo	0	0	0	0	1	0	0	1	0	0
	Negativo	0	0	0	0	0	1	0	0	1	1
HBsAg	Positivo	0	0	0	0	0	0	1	0	1	0
	Negativo	0	0	0	0	0	0	0	1	0	1
Anti HBc IgM	Positivo	0	0	0	0	0	0	1	0	1	0
	Negativo	0	0	0	0	0	0	0	1	0	1

Sabe-se que os neurônios do grupo 'diagnóstico' também não são mutuamente exclusivos. Além disso, outra informação a respeito do problema é que não existe relação direta entre os grupos que representam sintomas, informações gerais sobre os pacientes, exame físico e exames laboratoriais. Estas são feitas por meio do grupo 'diagnóstico'.

Portanto, para o exemplo com 13 neurônios, o número de pesos a ser encontrado é igual a 23 ($Pe = 23$). Seguindo a mesma linha de raciocínio, $Pe = 31$ para o exemplo de Diagnóstico de Hepatite com 14 neurônios, e $Pe = 37$ para o exemplo de Hepatite com 17 neurônios. Já para o problema de Hepatite com 44 neurônios, $Pe = 100$.

Voltando ao caso de diagnóstico de Hepatite com 13 neurônios, foram geradas diferentes redes, o conjunto com 10 pares de testes foi aplicado e as respostas analisadas. O treinamento foi feito usando cálculo do *fitness* por 3 valores (que é o mesmo usado na Etapa 1) e alfabeto inteiro com 1 gene por peso.

As duas primeiras entradas, aplicadas a estas redes, consistiam em ativar o neurônio correspondente a cada um dos dois diagnósticos possíveis. Foi observado que os sintomas apropriados foram ativados, além dos exames correspondentes. Depois disso, foram aplicadas possibilidades de casos clínicos, num total de 8. As taxas de acerto, considerando os 10 pares possíveis, para 3 diferentes matrizes, estão ilustradas na Figura 8.31. Observou-se que as 3 redes erraram o mesmo caso, sendo que a Matriz 2, teve um outro erro além deste.

Para a rede de 14 neurônios, que tem os resultados apresentados na Figura 8.32, duas das três redes apresentadas como exemplo forneceram taxas de acerto de 85,71% (Matriz 1 e 3), que implica em doze casos com resposta correta, de um total de quatorze. A Matriz 2

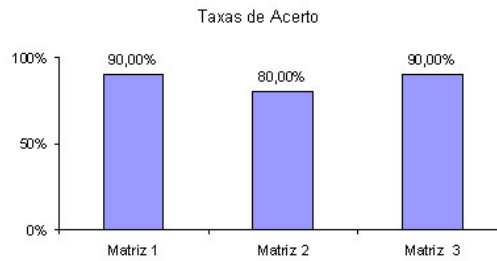


Figura 8.31: Hepatite com 13 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.

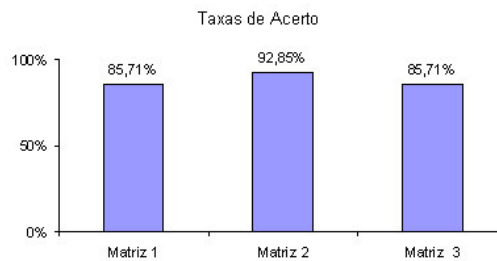


Figura 8.32: Hepatite com 14 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.

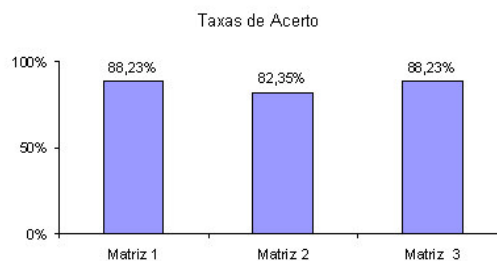


Figura 8.33: Hepatite com 17 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.

teve taxa de acerto de 92,85%, que implica em treze casos com resposta correta. No caso da rede de 17 neurônios, que tem os resultados apresentados na Figura 8.33, duas redes obtiveram taxas de acerto de 88,23% (Matriz 1 e 3), que implicam em quinze casos com resposta correta, de um total de dezessete. A Matriz 2 forneceu taxa de 82,35%, ou seja, acertou quatorze casos.

Para os exemplos de 14 e 17 neurônios, foi observada uma fonte de erros, que ocorre quando são ativados muitos sintomas relacionados a um diagnóstico. Por exemplo, no caso de uma pessoa apresentar muitos fatores de risco, relacionados à presença de Hepatite B, mesmo informando que o exame para este diagnóstico deu negativo, ainda assim, não é possível desativá-lo.

Este erro foi observado quando há mais de dois fatores de risco ativos na entrada, além de outros sintomas. Um exemplo é ilustrado na Figura 8.34. Apesar de se informar que os exames para Hepatite B forneceram resultados negativos, ainda assim, mediante o conjunto de neurônios de entrada, o diagnóstico se mantém ativo. Isto ocorre porque as conexões de inibição, que ligam os neurônios correspondentes aos exames negativos para Hepatite B e o diagnóstico de Hepatite B, não tem força suficiente para torná-lo inativo, considerando as ativações positivas recebidas.

<input type="checkbox"/>	Hepatite Aguda Vírus A	-0,1602
<input type="checkbox"/>	Hepatite Aguda Vírus B	0,5697
<input checked="" type="checkbox"/>	Com vômitos	0,7928
<input type="checkbox"/>	Sem vômitos	-0,1469
<input type="checkbox"/>	Febre alta	-0,1540
<input checked="" type="checkbox"/>	Febre média	0,7947
<input type="checkbox"/>	Febre baixa	-0,1303
<input type="checkbox"/>	Tatuagem	-0,1634
<input checked="" type="checkbox"/>	Piercing	0,7176
<input checked="" type="checkbox"/>	Homossexualismo	0,7231
<input checked="" type="checkbox"/>	Drogas injetáveis	0,7234
<input type="checkbox"/>	Anti-HVA IgM Positivo	-0,1211
<input type="checkbox"/>	Anti-HVA IgM Negativo	-0,1283
<input type="checkbox"/>	HBs Ag Positivo	-0,1323
<input checked="" type="checkbox"/>	HBs Ag Negativo	0,7585
<input type="checkbox"/>	Anti-HBc IgM Positivo	-0,1322
<input checked="" type="checkbox"/>	Anti-HBc IgM Negativo	0,7711

Figura 8.34: Exemplo de erro fornecido pelas redes treinadas, para o exemplo de Hepatite com 17 neurônios.

Já para a rede com 44 neurônios, os resultados apresentados não são tão eficientes. Foi criado um conjunto de teste com 50 casos clínicos. Entre estes, diversos casos simples informados não obtiveram uma resposta coerente. Alguns resultados estão ilustrados na Figura 8.35, onde obtiveram-se taxas de acerto de 74% (Matriz 1), 78% (Matriz 2) e 72% (Matriz 3), implicando em 37, 39 e 36 casos com resposta correta, respectivamente. Para chegar a estas respostas, foram utilizadas populações de 3000 indivíduos, avaliados ao longo de 300 gerações.

Conforme definido, o treinamento desta rede foi feito com 44 pares de entrada/saída, tornando o treinamento bastante lento. Foram testadas outras possibilidades de conjuntos de treinamento com menos pares, mas as respostas pioravam a medida que a quantidade de pares de treinamento diminuía.

Nas matrizes ilustradas, se observaram algumas associações inadequadas de sintomas

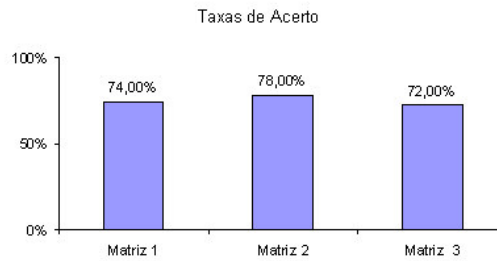


Figura 8.35: Hepatite com 44 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.

com diagnóstico, já que o AG não alcança 100% como valor de *fitness*. Pelo fato da rede não aprender o conjunto de treinamento completo, isto acabava gerando casos em que conexões que deveriam ter valores de inibição, assumiram valores de excitação, e vice-versa. Por exemplo, associar um valor de inibição entre resultados positivos para os exame de Hepatite B e diagnóstico de Hepatite B, o que gera resultados como aquele apresentado na Figura 8.36, de forma que mesmo informando que os exames para Hepatite B tiveram resultados positivos, ainda assim o diagnóstico não foi ativado. Devido a estes problemas, tornou-se necessário encontrar um novo conjunto de treinamento, dando origem a uma nova etapa do desenvolvimento da metodologia.

<input type="checkbox"/> Hepatite Aguda Vírus A	-0,1412	<input type="checkbox"/> Com astenia	0,0540
<input type="checkbox"/> Hepatite Aguda Vírus B	-0,1353	<input type="checkbox"/> Sem astenia	-0,1497
<input type="checkbox"/> Com náusea	-0,1395	<input type="checkbox"/> Com anorexia	-0,1515
<input type="checkbox"/> Sem náusea	-0,1436	<input type="checkbox"/> Sem anorexia	-0,1537
<input type="checkbox"/> Com vômitos	-0,1473	<input type="checkbox"/> Com hepatomegalia	-0,1378
<input type="checkbox"/> Sem vômitos	-0,1529	<input type="checkbox"/> Sem hepatomegalia	0,0586
<input type="checkbox"/> Não apresenta dor abdomi	-0,0288	<input type="checkbox"/> Transaminase normal	-0,1074
<input type="checkbox"/> Dor leve a moderada	-0,1523	<input type="checkbox"/> 2 a 4 vezes maior	-0,1400
<input type="checkbox"/> Dor intensa	-0,1342	<input type="checkbox"/> 4 a 10 vezes maior	0,2279
<input type="checkbox"/> Com icterícia	-0,1249	<input type="checkbox"/> Maior de 10 vezes	-0,1221
<input type="checkbox"/> Sem icterícia	-0,1556	<input type="checkbox"/> Anti-HVA IgM Positivo	-0,1458
<input type="checkbox"/> Não apresenta emagrecime	-0,1619	<input type="checkbox"/> Anti-HVA IgM Negativo	-0,1444
<input type="checkbox"/> Emagrecimento moderado	-0,1520	<input checked="" type="checkbox"/> HBs Ag Positivo	0,7851
<input type="checkbox"/> Emagrecimento acentuado	-0,1208	<input type="checkbox"/> HBs Ag Negativo	-0,1605
<input type="checkbox"/> Febre alta	-0,1517	<input checked="" type="checkbox"/> Anti-HBc IgM Positivo	0,7820
<input type="checkbox"/> Febre média	-0,1332	<input type="checkbox"/> Anti-HBc IgM Negativo	-0,1059
<input type="checkbox"/> Febre baixa	0,0682		
<input type="checkbox"/> Sem febre	-0,1361		
<input type="checkbox"/> Idoso	-0,1579		
<input type="checkbox"/> Adulto	-0,1166		
<input type="checkbox"/> Adolescente	-0,1142		
<input type="checkbox"/> Criança	-0,1559		
<input type="checkbox"/> Transusão	-0,1532		
<input type="checkbox"/> Promiscuidade	-0,1429		
<input type="checkbox"/> Piercing	-0,1564		
<input type="checkbox"/> Tatuagem	-0,1590		
<input type="checkbox"/> Homossexualismo	-0,1473		
<input type="checkbox"/> Drogas injetáveis	-0,0838		

Figura 8.36: Hepatite com 44 neurônios: Exemplo de associação incorreta entre pesos.

8.2 Etapa 2: Treinamento por Categorias

Foi observado, na etapa anterior, ao se realizar os testes com o problema de Hepatite, associações inadequadas que comprometiam a utilidade da rede gerada, por fornecer respostas incoerentes para os casos clínicos apresentados. Observou-se, também, que o conjunto de treinamento necessário é muito grande, o que acaba resultando num processamento muito lento. Outra dificuldade é definir claramente quais neurônios devem estar ativos na saída, para um determinado neurônio ativo na entrada.

Foram feitas outras tentativas de encontrar funções de *fitness* para o conjunto de treinamento por pares de entrada/saída, mas não se chegou a alguma outra, além das apresentadas, que guiasse a população para melhores resultados. Desta forma, não havendo outra função de *fitness* que melhorasse a resposta do AG, a alternativa utilizada foi modificar o arquivo de treinamento utilizado.

Este novo tipo de conjunto de treinamento considera faixas de pertinência, ou categorias, para os pesos, sendo esta a principal mudança introduzida nesta etapa. Para medir o desempenho das redes geradas com relação a este novo conjunto de treinamento, foi necessário desenvolver uma função de *fitness* adequada, apresentada na seção 5.5.4.

Utilizando esta nova abordagem, houve uma diminuição abrupta no tempo de processamento do AG, já que não é necessário consultar cada rede pertencente à população com as entradas do conjunto de treinamento, como era feito da forma anterior. Os operadores e mecanismos de seleção, utilizados na Etapa 1, foram mantidos.

Os treinamentos realizados consideram 5 categorias para os pesos, conforme apresentado na seção 5.2.2. Para a montagem do conjunto de treinamento, é necessário que o especialista informe, aproximadamente, o valor que a relação entre dois neurônios pode assumir, com base nas categorias definidas. O número de categorias e os intervalos para os pesos podem variar de acordo com o problema sendo analisado.

O primeiro exemplo testado foi o dos JS com 41 neurônios, para verificar se era possível reproduzir os resultados alcançados anteriormente. Foram implementadas duas possibilidades, sendo que uma consiste em encontrar todos os pesos abaixo da diagonal principal, e a outra em encontrar os pesos entre grupos que têm relação. Das 5 categorias definidas, dada a natureza do problema, apenas 3 foram efetivamente utilizadas, que são as que indicam forte relação de excitação, forte relação de inibição, e ausência de relação.

Foram realizados testes tanto para encontrar os pesos entre grupos que têm relação, como para encontrar os pesos abaixo da diagonal principal. Para o primeiro caso, alguns resultados obtidos para matrizes geradas por treinamento estão apresentados na Figura 8.37. Foram

alcançadas taxas de 89,71% (Matriz 1) e 89,59% (Matriz 2). No caso em que foram encontrados os pesos abaixo da diagonal principal, alguns resultados são apresentados na Figura 8.38, obtendo-se taxas de 88,10% e 89,29%.

Pode-se observar por estas figuras, que as taxas de acerto ficaram um pouco abaixo daquelas já conseguidas, sendo também possível observar que os dois métodos fornecem resultados muito próximos, sendo que a representação entre grupos que têm relação traz resultados ligeiramente melhores do que aquela que considera os pesos abaixo da diagonal principal. Entretanto, embora as taxas de acerto não tenham sido muito menores, estes resultados poderiam ser melhorados.

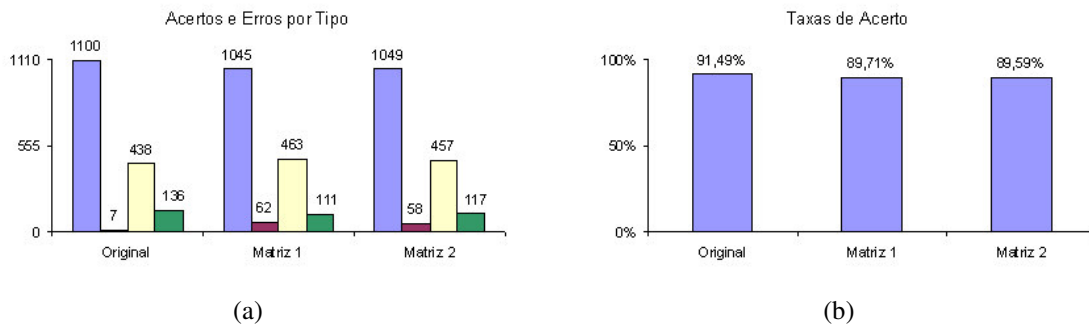


Figura 8.37: JS com 41 neurônios: Resultados obtidos quando são encontrados pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

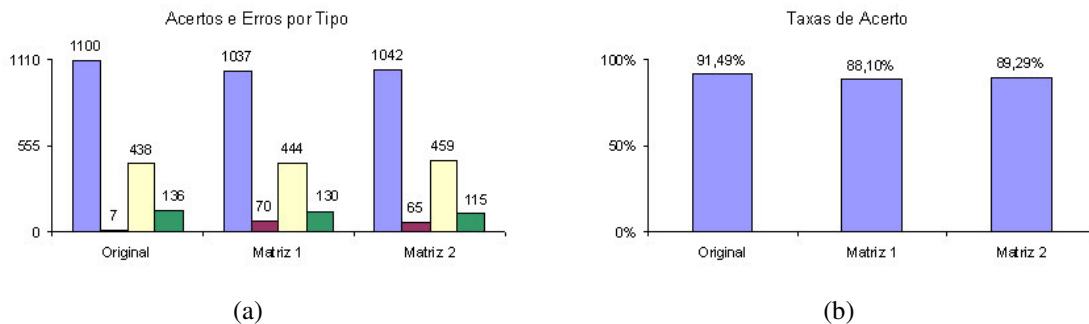


Figura 8.38: JS com 41 neurônios: Resultados obtidos quando são encontrados pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas. (b) Taxas de acerto.

Depois de observar as respostas fornecidas para o exemplo dos Jets & Sharks, a metodologia foi aplicada ao exemplo de Hepatite com 44 neurônios. Neste caso, para cada peso da matriz de conectividade, que deve ser encontrado pelo AG, define-se a categoria à qual este pertence, sendo que todas as categorias foram utilizadas, inclusive as que indicam relação fraca de excitação e inibição.

Utilizando este novo conjunto de treinamento, as respostas obtidas para o problema de

Hepatite foram bem melhores. Inclusive no caso de se informar todos os pesos abaixo da diagonal principal. Além do mais, no treinamento, é quase sempre possível chegar a 100% como valor de *fitness*. Desta forma, não ocorrem associações incorretas entre neurônios, que provocam as respostas incoerentes, que comprometem a utilidade da rede, como aqueles observados quando utilizado o treinamento por pares de entrada e saída. Isto ocorria, porque a metodologia considerava o comportamento global, ao invés de cada conexão individualmente. Por utilizar pares de entrada e saída, considera-se apenas a resposta final fornecida pela rede, podendo haver combinações diversas de pesos fornecendo a mesma resposta.

O mesmo conjunto de testes composto por 50 pares de entrada saída foi aplicado a redes obtidas pelo treinamento utilizando categorias para os pesos. As taxas de acerto de diagnósticos se elevaram, como observado pelos resultados ilustrados na Figura 8.39, onde se alcançaram taxas de acerto de 90% (Matriz 1), 86% (Matriz 2) e 88% (Matriz 3). Estes treinamento foram feitos evoluindo uma população de 3000 indivíduos, ao longo de 200 gerações. A taxa máxima alcançada anteriormente foi de 78%, como ilustrado na Figura 8.35.

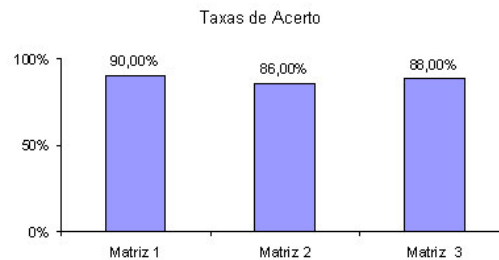


Figura 8.39: Hepatite com 44 neurônios: Resultados obtidos quando são encontrados os pesos entre grupos que têm relação. Alfabeto inteiro com 1 gene por peso.

Considerando a Matriz 1, que forneceu 90% de acerto, uma das fontes de erros observada em 3 exemplos do conjunto de testes repetiu o comportamento das redes de 14 e 17 neurônios, no treinamento realizado na Etapa 1. Quando existem muitos sintomas ativos, associados a um diagnóstico, no caso do exame relacionado ser negativo, esta conexão, mesmo tendo um valor que indique forte relação de inibição, não tem força suficiente para tornar o diagnóstico correspondente inativo. Este tipo de comportamento pode ser associado ao funcionamento da rede modelo 'A', que considera o somatório das ativações, tanto relacionadas à inibição, como relacionadas à excitação. Ela não tem uma característica de ignorar a informação, como ocorre, por exemplo, com sistemas especialistas. Num SE, ao se informar que uma determinada característica é negativa, as regras que a contêm são simplesmente ignoradas, não sendo computadas ao final do processamento.

Para ilustrar este problema, é apresentado um exemplo, utilizando uma rede obtida por treinamento. Considere-se um paciente com as seguintes características: Paciente adulto

com náusea, vômitos, dor abdominal leve a moderada, sem icterícia, emagrecimento moderado, febre média, com astenia, sem anorexia, com hepatomegalia, e apresentando exame de transaminase alterada de 2 a 4 vezes. A resposta para esta consulta é apresentada na Figura 8.40(a). Como o paciente é adulto e apresenta fatores de risco, a probabilidade de Hepatite B é maior, exatamente como na resposta fornecida pela rede.

<input type="checkbox"/> Hepatite Aguda Vírus A	0,4864	<input checked="" type="checkbox"/> Com astenia	0,8065	<input type="checkbox"/> Hepatite Aguda Vírus A	0,5398	<input checked="" type="checkbox"/> Com astenia	0,8053
<input type="checkbox"/> Hepatite Aguda Vírus B	0,7883	<input type="checkbox"/> Sem astenia	-0,1526	<input type="checkbox"/> Hepatite Aguda Vírus B	0,6522	<input type="checkbox"/> Sem astenia	-0,1523
<input checked="" type="checkbox"/> Com náusea	0,8049	<input type="checkbox"/> Com anorexia	-0,1117	<input checked="" type="checkbox"/> Com náusea	0,8034	<input type="checkbox"/> Com anorexia	-0,1149
<input type="checkbox"/> Sem náusea	-0,1543	<input checked="" type="checkbox"/> Sem anorexia	0,7705	<input type="checkbox"/> Sem náusea	-0,1537	<input checked="" type="checkbox"/> Sem anorexia	0,7716
<input checked="" type="checkbox"/> Com vômitos	0,8049	<input checked="" type="checkbox"/> Com hepatomegalia	0,8049	<input checked="" type="checkbox"/> Com vômitos	0,8034	<input checked="" type="checkbox"/> Com hepatomegalia	0,8034
<input type="checkbox"/> Sem vômitos	-0,1495	<input type="checkbox"/> Sem hepatomegalia	-0,1536	<input type="checkbox"/> Sem vômitos	-0,1494	<input type="checkbox"/> Sem hepatomegalia	-0,1535
<input type="checkbox"/> Não apresenta dor abdomi	-0,1145	<input type="checkbox"/> Transaminase normal	-0,1551	<input type="checkbox"/> Não apresenta dor abdomi	-0,1173	<input type="checkbox"/> Transaminase normal	-0,1547
<input checked="" type="checkbox"/> Dor leve a moderada	0,8076	<input checked="" type="checkbox"/> 2 a 4 vezes maior	0,7956	<input checked="" type="checkbox"/> Dor leve a moderada	0,8057	<input checked="" type="checkbox"/> 2 a 4 vezes maior	0,7947
<input type="checkbox"/> Dor intensa	-0,1553	<input type="checkbox"/> 4 a 10 vezes maior	-0,1136	<input type="checkbox"/> Dor intensa	-0,1549	<input type="checkbox"/> 4 a 10 vezes maior	-0,1166
<input type="checkbox"/> Com icterícia	-0,1115	<input type="checkbox"/> Maior de 10 vezes	-0,1136	<input type="checkbox"/> Com icterícia	-0,1148	<input type="checkbox"/> Maior de 10 vezes	-0,1166
<input checked="" type="checkbox"/> Sem icterícia	0,7682	<input type="checkbox"/> Anti-HV A IgM Positivo	-0,1181	<input checked="" type="checkbox"/> Sem icterícia	0,7690	<input type="checkbox"/> Anti-HV A IgM Positivo	-0,1045
<input type="checkbox"/> Não apresenta emagrecim	-0,1142	<input type="checkbox"/> Anti-HV A IgM Negativo	-0,1392	<input type="checkbox"/> Não apresenta emagrecim	-0,1172	<input type="checkbox"/> Anti-HV A IgM Negativo	-0,1401
<input checked="" type="checkbox"/> Emagrecimento moderado	0,8038	<input type="checkbox"/> HBs Ag Positivo	0,2398	<input checked="" type="checkbox"/> Emagrecimento moderado	0,8030	<input type="checkbox"/> HBs Ag Positivo	-0,1319
<input type="checkbox"/> Emagrecimento acentuado	-0,1514	<input type="checkbox"/> HBs Ag Negativo	-0,1518	<input type="checkbox"/> Emagrecimento acentuado	-0,1510	<input checked="" type="checkbox"/> HBs Ag Negativo	0,7438
<input type="checkbox"/> Febre alta	-0,1536	<input type="checkbox"/> Anti-HBc IgM Positivo	0,2398	<input type="checkbox"/> Febre alta	-0,1535	<input type="checkbox"/> Anti-HBc IgM Positivo	-0,1318
<input checked="" type="checkbox"/> Febre média	0,8049	<input type="checkbox"/> Anti-HBc IgM Negativo	-0,1529	<input checked="" type="checkbox"/> Febre média	0,8034	<input checked="" type="checkbox"/> Anti-HBc IgM Negativo	0,7406
<input type="checkbox"/> Febre baixa	-0,1143			<input type="checkbox"/> Febre baixa	-0,1172		
<input type="checkbox"/> Sem febre	-0,1081			<input type="checkbox"/> Sem febre	-0,1125		
<input type="checkbox"/> Idoso	-0,1375			<input type="checkbox"/> Idoso	-0,1392		
<input checked="" type="checkbox"/> Adulto	0,7880			<input checked="" type="checkbox"/> Adulto	0,7861		
<input type="checkbox"/> Adolescente	-0,1419			<input type="checkbox"/> Adolescente	-0,1408		
<input type="checkbox"/> Criança	-0,1365			<input type="checkbox"/> Criança	-0,1345		
<input type="checkbox"/> Transusão	-0,1655			<input type="checkbox"/> Transusão	-0,1663		
<input type="checkbox"/> Promiscuidade	-0,1645			<input type="checkbox"/> Promiscuidade	-0,1655		
<input checked="" type="checkbox"/> Piercing	0,7178			<input checked="" type="checkbox"/> Piercing	0,7114		
<input checked="" type="checkbox"/> Tatuagem	0,7178			<input checked="" type="checkbox"/> Tatuagem	0,7114		
<input type="checkbox"/> Homossexualismo	-0,1643			<input type="checkbox"/> Homossexualismo	-0,1650		
<input checked="" type="checkbox"/> Drogas injetáveis	0,7115			<input checked="" type="checkbox"/> Drogas injetáveis	0,7060		

(a)

(b)

Figura 8.40: (a)Exemplo de consulta, indicando um quadro de sintomas mais associado à Hepatite B. (b)Exemplo de consulta, onde não se inativou o diagnóstico de Hepatite B, mesmo em presença do exame negativo para este diagnóstico.

Então, a rede é novamente consultada acrescentado o resultado negativo para os exames de Hepatite B. A resposta à nova consulta é apresentada na Figura 8.40(b). Esperava-se que o neurônio associado à Hepatite B fosse desativado. Isto, porém, não ocorreu. Como este neurônio recebe ativação positiva de vários neurônios na rede, o estímulo de inibição, recebido dos neurônios relacionados ao exame negativo para Hepatite B, não são suficientes para desativá-lo.

Mesmo na rede original IAC, é considerado o conjunto de excitações externas positivas e negativas. Em outras palavras, um neurônio, mesmo que tenha forte conexão negativa vinda de outro neurônio com excitação externa, pode ser ativado (ao invés de "inativado") se ele possuir um conjunto grande de outras pequenas conexões positivas vindas de neurônios que tenham excitações externas. E vice-versa.

Mesmo assim, as respostas obtidas para o exemplo de Hepatite foram mais próximas do desejado, quando utilizando o conjunto de treinamento por categorias. E assim, a metodologia foi aplicada ao problema de diagnóstico em reumatologia.

O conjunto de treinamento foi elaborado para o problema de 29 e 50 neurônios, considerando as 5 categorias já apresentadas. Para o caso de diagnóstico em reumatologia, todos os grupos se relacionam entre si, tornando-se necessário encontrar todos os pesos abaixo da diagonal principal, já que todos os neurônios estão relacionados entre si. Assim, para a rede de 29 neurônios, aplicando a Equação 5.1, $Pe = 406$. Para a rede de 50 neurônios, aplicando a mesma Equação, $Pe = 1225$.

De forma a validar as redes geradas, foram criados conjuntos de testes com número variável de neurônios ativos. Estes conjuntos são aplicados à rede desenvolvida por De Azevedo [22]. As respostas obtidas para cada entrada do conjunto é armazenada. Em seguida, este conjunto é aplicado à rede sendo testada. As respostas fornecidas pelas duas redes são comparadas de forma qualitativa, verificando se fornecem as mesmas ativações e inativações, e se, em cada grupo, a ordem de ativação é a mesma.

A aplicação da metodologia para os dois exemplos de diagnóstico em reumatologia, resultou em redes que fornecem muitos padrões com respostas inadequadas para o problema, sendo necessário melhorar as taxas apresentadas. Três redes geradas por treinamento para a rede de 29 neurônios estão apresentadas na Figura 8.41. Na Figura 8.41(a), foi aplicado às três redes um conjunto de teste com 29 entradas contendo apenas um neurônio ativo, e se observaram taxas de coincidência de 82,75% (Matriz 1), 81,92% (Matriz 2) e 82,40% (Matriz 3). Na Figura 8.41(b), foi aplicado às mesmas três redes um conjunto de teste com 29 entradas contendo dois neurônios ativos em cada uma, e se observaram taxas de coincidência de 81,09% (Matriz 1), 82,16% (Matriz 2) e 81,45% (Matriz 3).

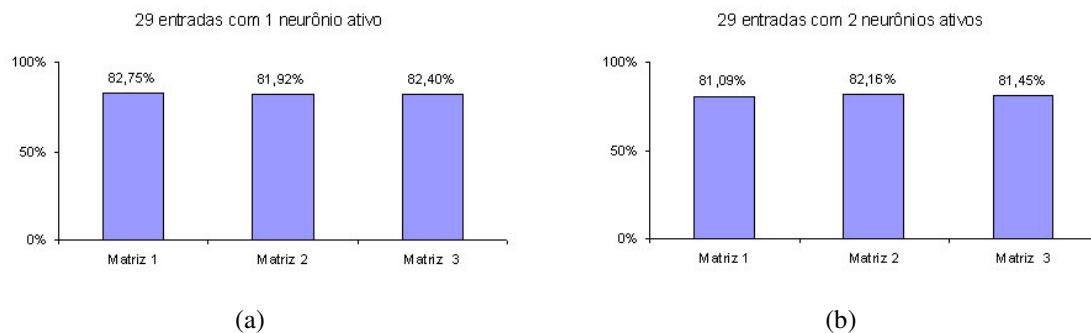


Figura 8.41: Diagnóstico em Reumatologia com 29 neurônios. (a) Taxa de coincidência usando conjunto de teste com 29 entradas com 1 neurônio ativo cada. (b) Taxa de coincidência usando conjunto de teste com 29 entradas com 2 neurônios ativos cada.

Na Figura 8.42, são apresentadas três matrizes geradas por treinamento para a rede de 50 neurônios. Na Figura 8.42(a), são apresentadas as taxas de coincidência quando aplicado um

conjunto de teste com 50 entradas, onde cada uma continha apenas um neurônio ativo. Os resultados observados foram de 82,56% (Matriz 1), 81,84% (Matriz 2) e 82,72% (Matriz 3). Na Figura 8.41(b), as taxas de coincidência aplicadas se referem ao caso em que é aplicado um conjunto de teste com 50 entradas contendo 2 neurônios ativos em cada uma. As taxas obtidas foram de 82,28% (Matriz 1), 81,96% (Matriz 2) e 82,12% (Matriz 3).

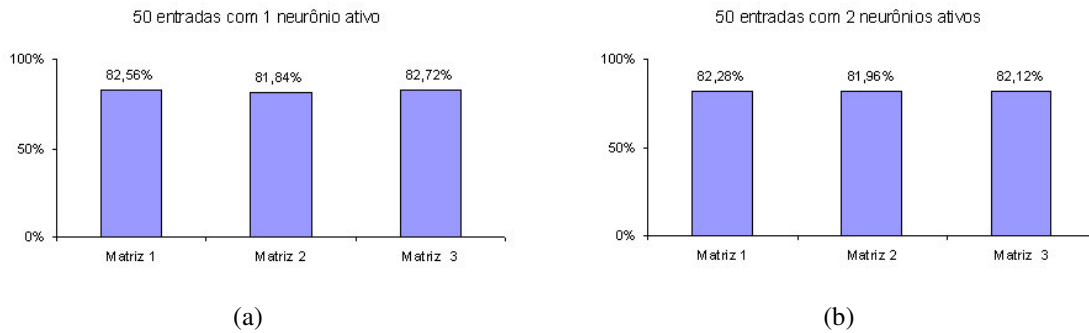


Figura 8.42: Diagnóstico em Reumatologia com 50 neurônios. (a) Taxa de coincidência usando conjunto de teste com 50 entradas com 1 neurônio ativo cada. (b) Taxa de coincidência usando conjunto de teste com 50 entradas com 2 neurônios ativos cada.

Ao se realizar uma análise da qualidade das respostas fornecidas, detectou-se a necessidade de realizar um direcionamento para os pesos obtidos. De acordo com as redes geradas, tanto para a rede de 29, como para a de 50 neurônios, observou-se que os pesos variavam dentro dos valores permitidos pela faixa indicada, sem que houvesse alguma regra ou guia, indicando mais próximo de qual extremo este valor deveria ficar. E esta característica acabou por resultar em alguns problemas no comportamento da rede.

Em [22], existem alguns exemplos ilustrativos para as redes de diagnóstico em reumatologia implementadas. Um destes exemplos é apresentado na Figura 8.43 para ilustrar um mau comportamento da rede, onde deve-se ativar os neurônios associados os sintomas de artrite e artralgia, além do paciente genérico, e verificar qual as possibilidades de diagnóstico.

Como pode-se observar pelas figuras, ao invés de manter as duas possibilidades de diagnóstico ativas, como era de se esperar, já que artralgia e artrite não são definitivas para nenhuma das duas possibilidades, a rede mostrada como exemplo ativa apenas Artrite Reumatóide. Uma das explicações para este tipo de comportamento reside nos pesos utilizados por De Azevedo e aqueles encontrados pela rede. Os pesos utilizados por De Azevedo são aqueles apresentados na Tabela 8.8. Já os pesos encontrados por treinamento são apresentados na Tabela 8.9.

Como pode-se observar pelas Tabelas, os pesos encontrados para a matriz exemplo, estão dentro dos limites necessários, porém, estão em limites opostos, quando deveriam ser valores mais próximos. É claro que todo o conjunto de pesos da matriz de conectividade acaba por

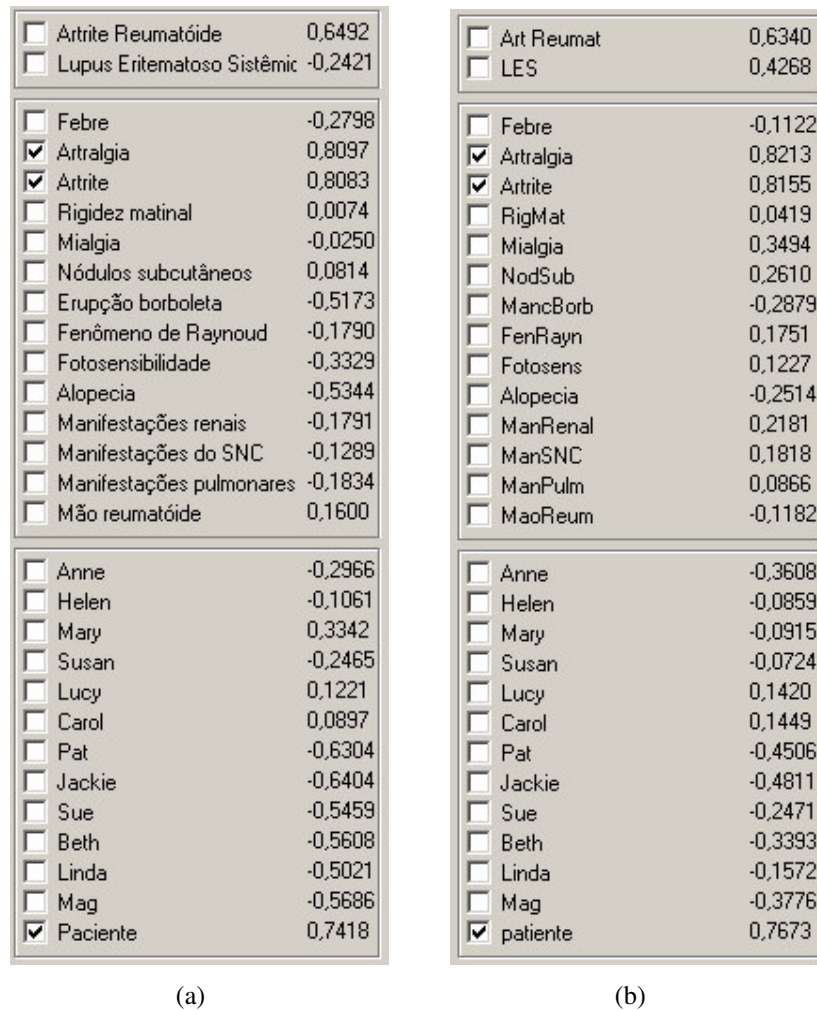


Figura 8.43: Diagnóstico em Reumatologia com 29 neurônios: Exemplo. (a) Exemplo de consulta com resposta indevida, devido à limites das categorias aplicadas (b) Resposta esperada, de acordo com exemplo implementado por De Azevedo.

Tabela 8.8: Pesos utilizados por De Azevedo.

	Artrite Reumatóide	Lupus Eritematoso Sistêmico
Artralgia	1	0,9
Artrite	0,9	0,6

Tabela 8.9: Exemplo de pesos encontrados pelo AG.

	Artrite Reumatóide	Lupus Eritematoso Sistêmico
Artralgia	1	0,8
Artrite	1	0,5

influenciar a resposta final. Porém, uma diferença entre pesos além da necessária pode trazer resultados bem diferentes ao final do processamento.

O mesmo tipo de erro foi observado para a rede de 50 neurônios (Figura 8.44) para o

exemplo onde deveriam ser ativados os neurônios artralgia, artrite, manifestações renais e paciente genérico. A rede de De Azevedo traz pelo menos 3 possibilidades de diagnóstico, enquanto a rede gerada por treinamento traz duas possibilidades, e com ativações diferentes daquelas fornecidas pela outra rede.

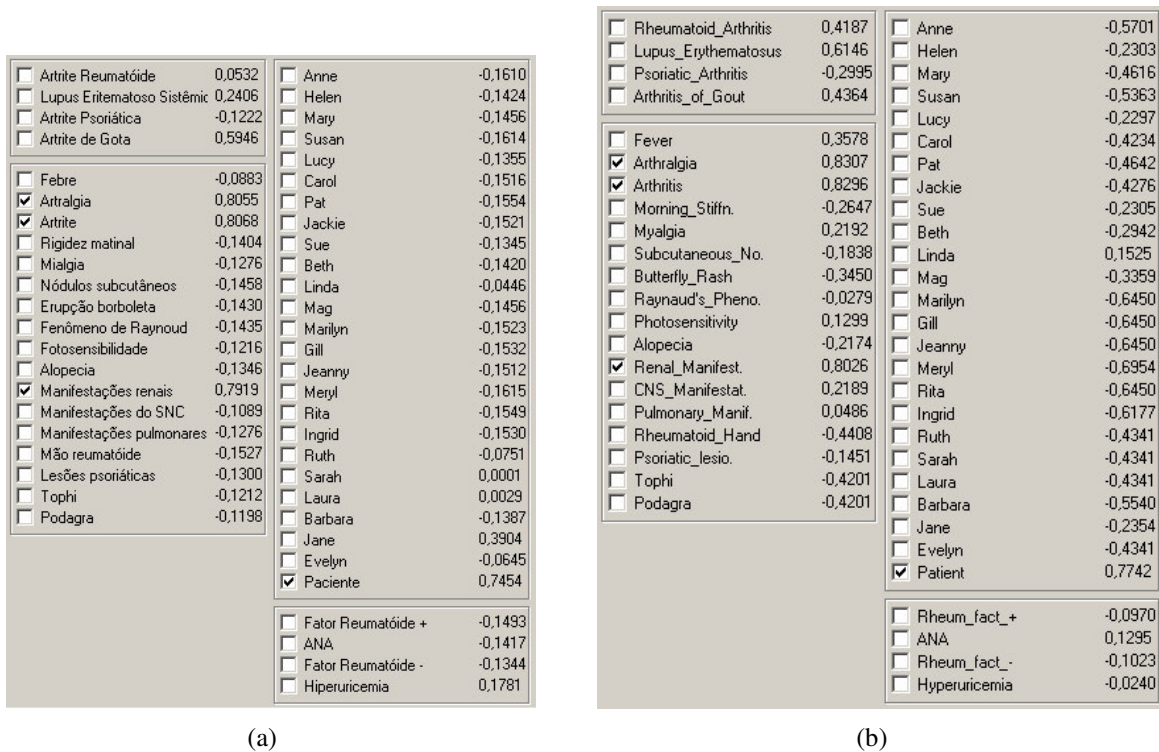


Figura 8.44: Diagnóstico em Reumatologia com 50 neurônios: Exemplo. (a) Exemplo de consulta com resposta indevida, devido à limites das categorias aplicadas. (b) Resposta esperada, de acordo com exemplo implementado por De Azevedo.

No caso de redes muito grandes, as variações nos pesos podem provocar variações maiores na sua resposta final. Desta forma, é necessário um ajuste mais fino que melhore a resposta fornecida. Diante desta constatação, a pesquisa prosseguiu para a próxima etapa.

8.3 Etapa 3: Geração da População Inicial com Restrição e Treinamento por Pares de Entrada/Saída

Esta nova abordagem é uma forma híbrida das propostas anteriores. O conjunto de treinamento da Etapa 2 passa a ser utilizado na geração da população inicial. Quando o indivíduo que representa a solução para um problema é muito grande, é interessante utilizar restrições na geração da população inicial, limitando o espaço de busca para o problema.

Desta forma, o arquivo de treinamento volta a ser aquele utilizado na Etapa 1, composto por pares de entrada/saída. A nova proposta foi aplicada a todos os exemplos, verificando se bons desempenhos obtidos se mantêm, e se é possível melhorar desempenhos não tão adequados.

A primeira aplicação testada foi o exemplo dos JS com 41 neurônios. Já neste problema foi possível reproduzir os resultados obtidos com a primeira simulação e até melhorá-los, em alguns casos, comparados aos resultados fornecidos pela rede original. A metodologia foi aplicada tanto para encontrar os pesos situados abaixo da diagonal principal, como os pesos entre grupos que têm relação.

Quando encontrando os pesos entre grupos que têm relação, são obtidos as taxas de acerto encontradas na Figura 8.45(b), que traz taxas de 90,30% (Matriz 1) e 91,02% (Matriz 2). Já a Figura 8.46(a) traz as taxas de acerto quando encontrados os pesos abaixo da diagonal principal. Neste caso, as taxas foram de 92,27% (Matriz 1) e 91,91% (Matriz 2). As simulações foram feitas ao longo de 150 gerações, utilizando 100 indivíduos.

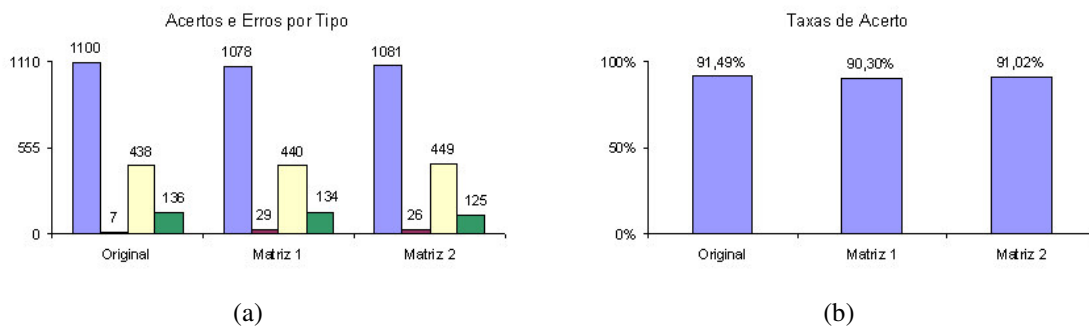


Figura 8.45: JS com 41 neurônios: Resultados obtidos quando são encontrados pesos entre grupos que têm relação, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.

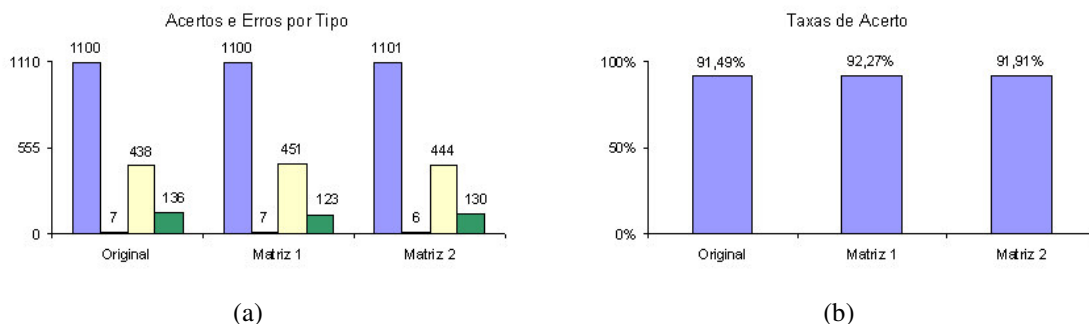


Figura 8.46: JS com 41 neurônios: Resultados obtidos quando são encontrados os pesos abaixo da diagonal principal, comparados com aqueles fornecidos pela IAC original. Alfabeto inteiro com 1 gene por peso. (a) Número de acertos e erros para entradas gerais e específicas (b) Taxas de acerto.

Observa-se, por estas figuras, que este tipo de treinamento faz um ajuste na população

gerada inicialmente. Os valores de *fitness* já são razoáveis na população inicial, e há uma evolução em direção das redes que fornecem melhores resultados, guiados pelo conjunto de treinamento indicado.

Taxas de acerto melhores são conseguidas quando se encontram todos os pesos abaixo da diagonal principal, embora a diferença entre os dois tipos de resultados não seja tão grande. Nos treinamentos realizados nas Etapas 1 e 2, são obtidas melhores taxas quando se encontram os pesos entre grupos que têm relação.

O que pode ser observado é que, em se tratando de redes modelo 'A', permitir uma maior flexibilidade nos pesos, desde que esta flexibilidade seja limitada pela geração com restrições da população inicial, pode ser benéfica para a resposta final da rede. Esta flexibilidade para os valores permitiu uma diferença nas respostas, embora pequena, ou seja, permitiu que se conseguissem redes com resultados até melhores do que aqueles fornecidos pela IAC original.

Convém notar, também, que os resultados obtidos foram conseguidos utilizando-se muito menos gerações e número de indivíduos na população, tornando a evolução do AG bem mais rápida. Após a verificação de que a qualidade das respostas obtidas para a rede dos JS é adequada, passou-se a testar esta metodologia com os outros exemplos existentes.

Para o exemplo de Hepatite com 44 neurônios, foi aplicado o mesmo conjunto de teste composto de 50 casos clínicos, como apresentado nas etapas anteriores. Os resultados são apresentados na Figura 8.47, onde foram alcançadas taxas de acerto de 92% (Matriz 1) e 94% (Matriz 2 e Matriz 3).

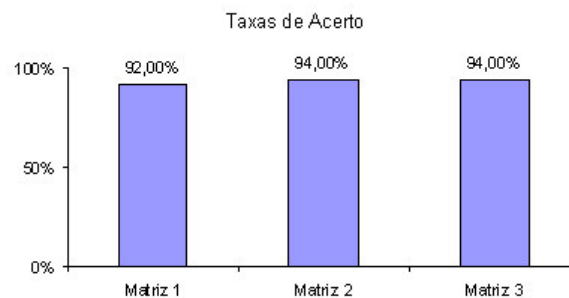


Figura 8.47: Hepatite com 44 neurônios: Taxas de acerto fornecidas.

As taxas de acerto para os diagnósticos aumentou, porém, se manteve a característica da rede, em considerar o conjunto total de ativações, e não informações isoladas. Passando para o exemplo de diagnóstico em reumatologia, os testes foram realizados tanto com a rede de 29 neurônios, como com a rede de 50 neurônios, para as quais nenhuma metodologia havia trazido resultados dentro do esperado. Aqui, novamente, foram aplicados os mesmos

conjuntos utilizados nos testes anteriores nas redes geradas por treinamento e naquelas desenvolvidas por De Azevedo [22], um composto por entradas com um único neurônio ativo, e um composto por entradas com dois neurônios ativos.

Os resultados referentes ao exemplo de 29 neurônios são apresentados na Figura 8.48, que contém resultados para três redes geradas por treinamento. Na Figura 8.48(a), foi aplicado às três redes um conjunto de teste com 29 entradas contendo apenas um neurônio ativo, e se observaram taxas de coincidência de 91,31% (Matriz 1), 90,96% (Matriz 2) e 91,67% (Matriz 3). Na Figura 8.48(b), foi aplicado às mesmas três redes um conjunto de teste com 29 entradas contendo dois neurônios ativos em cada uma, e se observaram taxas de coincidência de 91,91% (Matriz 1), 91,43% (Matriz 2) e 91,79% (Matriz 3).

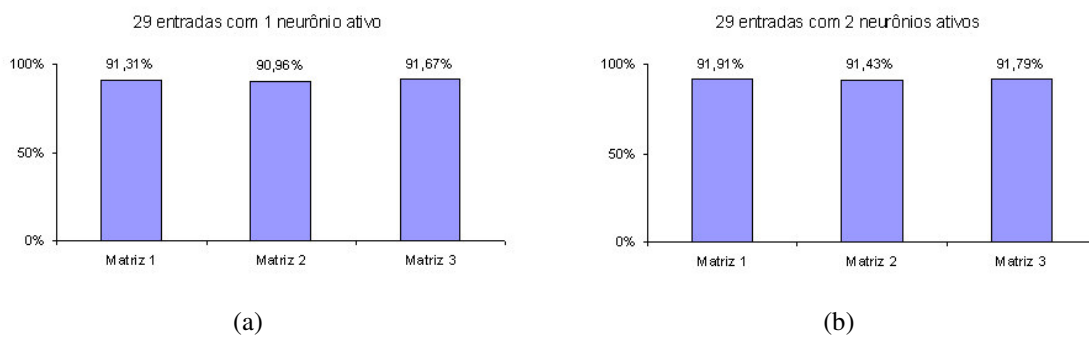


Figura 8.48: Diagnóstico em Reumatologia com 29 neurônios. (a) Taxa de coincidência usando conjunto de teste com 29 entradas com 1 neurônio ativo cada. (b) Taxa de coincidência usando conjunto de teste com 29 entradas com 2 neurônios ativos cada.

Na Figura 8.49, são apresentadas três matrizes geradas por treinamento para a rede de 50 neurônios. Na Figura 8.49(a), são apresentadas as taxas de coincidência quando aplicado um conjunto de teste com 50 entradas, onde cada uma continha apenas um neurônio ativo. Os resultados observados foram de 90,32% (Matriz 1), 90,27% (Matriz 2) e 92,30% (Matriz 3). Na Figura 8.49(b), as taxas de coincidência aplicadas se referem ao caso em que é aplicado um conjunto de teste com 50 entradas contendo 2 neurônios ativos em cada uma. As taxas obtidas foram de 90,82% (Matriz 1), 91,02% (Matriz 2) e 92,7% (Matriz 3).

Observamos que as taxas de coincidência cresceram de cerca de 83% para cerca de 90%, alcançando taxas de coincidência de até 92,7% com esta metodologia, conforme observado na Figura 8.49(b). Ressalta-se que, quando os resultados não coincidem, isto não significa que a rede gerada por treinamento esteja errada. A rede implementada por De Azevedo para este problema, assim como qualquer outra rede neural, não apresenta 100% de taxa de acerto, já que este não é um comportamento esperado para este tipo de ferramenta.

Utilizando novamente os exemplos apresentados por De Azevedo em [22], considere-se o caso onde é ativado o sintoma Podagra e são recuperadas as informações pertinentes, como

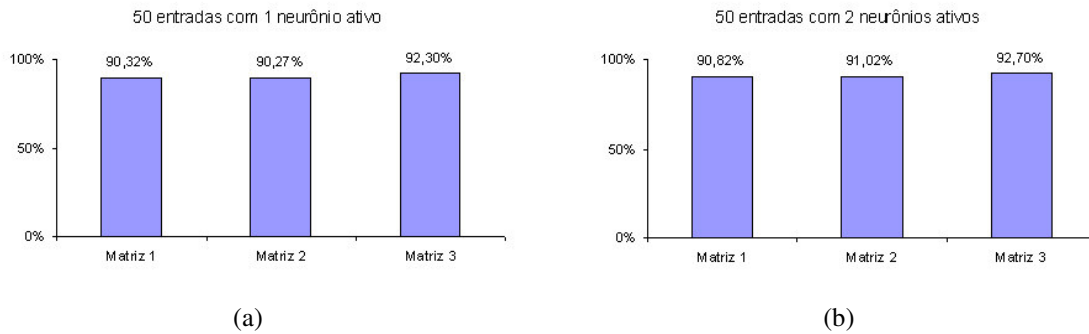


Figura 8.49: Diagnóstico em Reumatologia com 50 neurônios. (a) Taxa de coincidência usando conjunto de teste com 50 entradas com 1 neurônio ativo cada. (b) Taxa de coincidência usando conjunto de teste com 50 entradas com 2 neurônios ativos cada.

mostrado na Figura 8.50. Ao se ativar este neurônio, a paciente chamada Barbara é recuperada pela rede implementada por De Azevedo. Na verdade, esta paciente não apresenta este sintoma. Entretanto, a rede encontrada por treinamento, não ativa esta característica. Ou seja, este é um acerto na recuperação da informação dado pela rede encontrada por treinamento e um erro fornecido pela rede de Azevedo. No caso da paciente chamada Ruth, esta também não deveria estar ativa, considerando o sintoma Podagra. Pode-se observar que a rede fornecida por treinamento, ativou com valor bem menor do que a implementação fornecida por De Azevedo.

Outro fator que se observou é que, mesmo utilizando 50 pares no conjunto de treinamento, que é um valor considerável, são necessárias apenas cerca de 100 a 200 indivíduos sendo avaliados por cerca de 100 gerações para ser possível encontrar as respostas fornecidas.

Uma outra opção considerada foi utilizar uma abordagem mista. Neste caso, se realizaria a geração aleatória da população inicial, e o valor de *fitness* consideraria as duas possibilidades de conjuntos de treinamento. Ou seja, com dois arquivos de teste, sendo que os melhores *fitness* seriam atribuídos às redes que respondessem adequadamente para o arquivo de faixas de pesos, e respondesse bem para um possível conjunto de treinamento para entradas e saídas. Este tipo de treinamento, porém, se mostrou muito lento, ficando inviável para redes maiores.

Também foi cogitada a possibilidade de utilizar um outro tipo de conjunto de treinamento que utiliza pares de entrada e saída, mas ao invés de utilizar apenas os valores 0 e 1 na saída, utiliza-se faixas de ativação para os neurônios. Esta abordagem leva em consideração o fato de que, a menos que um neurônio receba ativação externa, ele dificilmente vai estar ativo com valor maior que 0,5. Entretanto, esta implementação também não trouxe melhoras para a qualidade das respostas fornecidas.

	(a)	(b)
<input type="checkbox"/> Anne	-0,1708	-0,1740
<input type="checkbox"/> Helen	-0,1709	-0,1706
<input type="checkbox"/> Mary	-0,1684	-0,1717
<input type="checkbox"/> Susan	-0,1713	-0,1740
<input type="checkbox"/> Lucy	-0,1684	-0,1704
<input type="checkbox"/> Carol	-0,1690	-0,1713
<input type="checkbox"/> Pat	-0,1723	-0,1740
<input type="checkbox"/> Jackie	-0,1704	-0,1740
<input type="checkbox"/> Sue	-0,1692	-0,1717
<input type="checkbox"/> Beth	-0,1684	-0,1717
<input type="checkbox"/> Linda	-0,1694	-0,1708
<input type="checkbox"/> Mag	-0,1695	-0,1721
<input type="checkbox"/> Marilyn	-0,1683	-0,1717
<input type="checkbox"/> Gill	-0,1690	-0,1713
<input type="checkbox"/> Jeanny	-0,1673	-0,1717
<input type="checkbox"/> Meryl	-0,1720	-0,1736
<input type="checkbox"/> Rita	-0,1679	-0,1717
<input type="checkbox"/> Ingrid	-0,1685	-0,1710
<input type="checkbox"/> Ruth	0,2186	0,4333
<input type="checkbox"/> Sarah	0,6406	0,6130
<input type="checkbox"/> Laura	0,6333	0,6130
<input type="checkbox"/> Barbara	-0,0253	0,1806
<input type="checkbox"/> Jane	-0,1168	-0,0998
<input type="checkbox"/> Evelyn	0,3970	0,4780
<input type="checkbox"/> Paciente	-0,1692	-0,1699

(a) (b)

Figura 8.50: Diagnóstico em Reumatologia com 50 neurônios: Exemplo. (a) Resposta de uma consulta para uma rede gerada por treinamento. (b) Resposta dada pela implementação de De Azevedo.

Capítulo 9

Considerações Finais

Baseado no fato dos AGs já serem utilizados no treinamento de diversos tipos de redes, tais como apresentado em [28], [29] e [31], estes foram utilizados para o treinamento de redes modelo 'A'.

De acordo com o desenvolvimento deste trabalho, a hipótese '*É possível o desenvolvimento de uma metodologia para realizar o ajuste de pesos de uma rede modelo 'A', utilizando Algoritmos Genéticos*' se mostrou verdadeira.

A metodologia apresentada oferece uma alternativa viável para o ajuste de pesos de uma rede modelo 'A', em contrapartida aos métodos de tentativa e erro, utilizados em trabalhos como [34]. A utilização de AGs implica em algumas dificuldades, como a montagem do arquivo de treinamento adequado, já que em algumas situações fica difícil estabelecer qual a relação existente entre dois neurônios. Além disso, se torna necessário estabelecer faixas de pertinência para os pesos (ou categorias). O resultado final, porém, pode fornecer resultados satisfatórios e uma maior confiança nas respostas fornecidas pela rede. Com esta formulação, a partir de uma medida aproximada que os valores dos pesos devem assumir, um conjunto de treinamento guia a população em direção da resposta mais adequada. Portanto, apesar das dificuldades apresentadas, a metodologia é capaz de fornecer resultados mais confiáveis, após o treinamento.

Além disto, todas as características pertinentes às redes IAC são preservadas, como aquela que implica em bidirecionalidade. Em [47], esta característica não é mantida. A estabilidade das redes geradas também pode ser demonstrada.

Para realizar a validação da metodologia apresentada foram utilizados exemplos. Procurou-se escolher problemas que fossem representativos, para que seja possível generalizar a metodologia para outros semelhantes.

O exemplo dos JS, no caso, foi escolhido porque funciona como padrão ouro, já que é um problema onde o conhecimento está muito bem estabelecido. O exemplo de Hepatite foi escolhido por apresentar uma modelagem intermediária entre o JS, que tem todos os grupos mutuamente exclusivos e as relações totais, e o exemplo de diagnóstico em reumatologia, que tem grupos com mais neurônios, onde os neurônios não são mutuamente exclusivos. Além disso, foi possível contar com um especialista de domínio na modelagem do problema. E o exemplo de diagnóstico em reumatologia, além das características de interesse, foi implementado em [22] por métodos de elicitación de conhecimento, sendo assim possível comparar os dois métodos.

No caso do exemplo dos Jets & Sharks, foi possível encontrar uma rede tão ou um pouco mais eficiente que a fornecida pela implementação utilizando a rede IAC original, utilizando apenas os pesos 1, 0 e -1, além do grupo escondido. Nas 3 etapas desenvolvidas, bons resultados foram alcançados.

Sabe-se que a generalização é um dos problemas desta rede, que funciona muito bem para recuperar informação específica a respeito de um padrão fornecido na entrada. Como exemplo, a rede responde adequadamente as informações sobre os membros da gangue, mas não responde tão bem quando é perguntado quem são os Jets.

O exemplo de diagnóstico diferencial de Hepatite seria impossível de resolver utilizando a rede IAC original, pois apresenta características com valores não mutuamente exclusivos, além de ter relações entre neurônios assumindo diferentes graus. A rede modelo 'A' permite a sua implementação, mas traz a dificuldade no ajuste de pesos. Este pode ser feito, de forma automática, pela metodologia apresentada.

Os erros gerados para o problema de Hepatite, quando a rede deixava um diagnóstico ativo, mesmo sendo informado que o exame para ele era negativo, são atribuídos a uma característica própria das redes modelo 'A'. Ou seja, mesmo na rede original IAC, é considerado o conjunto de excitações externas positivas e negativas. Assim, não se pode esperar que neurônios conectados a uma quantidade maior de outros com excitação positiva, se tornem inativos.

Em relação ao exemplo que trata do diagnóstico em reumatologia, que também não poderia ser implementado pela rede original, observou-se que a metodologia foi capaz de gerar matrizes que fornecem resultados semelhantes àqueles trazidos por [22], que foram elaborados com base no conhecimento do especialista. Embora as redes comparadas tenham fornecido respostas diferentes para as mesmas situações, não se pode afirmar qual é melhor. O que ocorre, quando usando a metodologia, são respostas mais coerentes em relação ao conjunto de treinamento, que reflete nas taxas de acerto fornecidas. Ou seja, garante uma resposta mais segura com relação ao que foi ensinado à rede, o que torna a rede mais robusta

e passível de ser aplicada em uma quantidade maior de aplicações.

Com relação aos exemplos escolhidos, pode-se sugerir que a metodologia pode ser aplicada à problemas semelhantes. Os exemplos envolvem vários grupos, que tem valores não mutuamente exclusivos, e com neurônios se relacionando com diferentes graus. Além disso, é possível encontrar tanto os pesos abaixo da diagonal principal, como os pesos entre grupos que têm relação. Portanto, pode-se considerar esta metodologia como genérica o suficiente para ser aplicada a diferentes problemas, que possam ser representados através de redes IAC modelo 'A', as quais, por sua vez, resolvem as restrições apresentadas pelo modelo original de Rumelhart, que limitam, grandemente, sua aplicabilidade prática.

O uso destes exemplos levou a algumas considerações a respeito de alguns dos parâmetros utilizados, que foram testados de forma empírica:

- As taxas de cruzamento e mutação foram mantidas constantes. Observou-se que pequenas variações nestes parâmetros não provocavam grandes alterações nos resultados, não sendo necessário um ajuste tão fino. Porém, variações maiores nestas taxas faziam com que o AG tivesse um desempenho ruim.
- O número de indivíduos e o número de gerações eram sempre definidos dependentes do exemplo que se estava utilizando, sempre considerando que quanto maior o tamanho do indivíduo, maior o número de indivíduos e gerações necessário.
- As variáveis v_{max} e v_{min} foram definidas com base no comportamento de redes IAC em geral, e assumiram os valores de 0,5 e 0,1, respectivamente.
- O número de pares de treinamento é igual ao número de neurônios que a rede contém, sendo que cada entrada deve ser composta por um neurônio ativo.

O tempo de processamento necessário foi considerado bastante razoável. Para exemplificar, no caso do treinamento do exemplo de Diagnóstico em Reumatologia, com 50 neurônios, o tempo de processamento ficava em torno de 2 horas e trinta a duas horas e quarenta e cinco minutos, quando utilizando a metodologia descrita na Etapa 3.

O fato de melhores resultados terem sido conseguidos quando utilizados conjuntos de entrada e saída para treinamento em conjunto com as restrições nos conjuntos de pesos está relacionado ao fato das redes IAC, e modelos nela baseados, terem representação localizada.

Sabe-se que a rede modelo 'A', assim como a rede IAC original, tem seu conhecimento localizado, e não distribuído, como a maioria das redes neurais. Isto indica, que a cada neurônio ou sinapse é atribuído um conhecimento determinado. Quando o conhecimento

é distribuído, este é compartilhado por vários neurônios ou sinapses. Talvez, por ser mais econômica do ponto de vista de neurônios necessários, a representação distribuída seja mais utilizada [34].

Utilizar as categorias leva esta característica em consideração. Utilizando a abordagem anterior, várias matrizes diferentes podem representar a solução para o problema, inclusive matrizes que não respeitam as relações existentes entre dois neurônios específicos. Isto porque, na outra abordagem, se leva em consideração somente a saída final fornecida pela matriz de conectividade sendo analisada.

Os intervalos das categorias podem variar, sendo possível definir uma quantidade maior ou menor destes, em função do problema que se quer resolver. Além disso, a restrição na geração inicial dos pesos traz uma maior liberdade com relação ao número de intervalos, que pode ser menor.

Com relação à complexidade envolvida nos dois tipos de conjunto de treinamento, esta seria a mesma, já que o conhecimento do especialista a respeito do problema seria útil tanto para montar um conjunto de entradas e saídas, como para montar uma matriz de conectividade aproximada.

Um outro objetivo traçado foi a elaboração do sistema de treinamento. Este sistema pode facilitar a execução de diferentes treinamentos, variando apenas os parâmetros envolvidos, e a elaboração do conjunto de teste, permitindo que testes variados sejam executados mais rapidamente.

Por meio do sistema é possível inserir mais neurônios na rede, bastando refazer o treinamento, incluindo a informação pertinente àquele neurônio, tanto no conjunto de treinamento utilizado quanto no conjunto utilizado para restringir a geração da população inicial. A criação de um sistema que facilite estas atividades também pode ser levantado como um ponto positivo na elaboração deste trabalho.

Apesar de não ter sido utilizadas redes com grupo escondido, é perfeitamente possível a aplicação da metodologia para este caso, bastando considerar que não existe relação entre grupos visíveis, mas sim destes com o grupo escondido. Isto quer dizer que basta utilizar a opção da metodologia que permite encontrar os pesos entre grupos que têm relação.

Assim, pode-se dizer que os objetivos traçados para o trabalho foram atingidos, permitindo automatizar o treinamento de uma rede modelo 'A'. Espera-se que esta metodologia contribua para uma maior utilização deste tipo de redes, já que permite, agora, uma aplicação a uma quantidade maior de problemas, mantendo todas as características de interesse da rede IAC original.

9.1 Trabalhos Futuros

Existem algumas propostas que podem ser elaboradas para melhorar o desempenho das redes modelo 'A' e da metodologia proposta:

- Elaborar um modelo híbrido que use redes IAC e Sistemas Especialistas. Este tipo de sistema permitiria que fosse possível resolver algumas restrições impostas pelo modelo 'A', assim como pelo modelo original. Um modelo como esse, resolveria questões como aquelas apontadas para o problema de Hepatite, onde a ativação do neurônio relacionado ao exame negativo não torna o diagnóstico correspondente negativo.
- A utilização do alfabeto decimal se mostrou viável, fornecendo os mesmos resultados da representação inteira. É necessário, porém, a realização de mais testes, com a implementação de operadores apropriados para que seja possível utilizar este tipo de representação.

Referências Bibliográficas

- [1] BITTENCOURT, G. *Inteligência Artificial: Ferramentas e Teorias*. Florianópolis: Editora da UFSC, 1998.
- [2] RUSSEL, S.; NORVIG, P. *Inteligência Artificial*. Rio de Janeiro: Editora Campus, 2004.
- [3] CHRISTODOULOU, C.; PATTICHIS, C.; PANTZIARIS, M.; NICOLAIDES, A. Texture-Based Classification of Atherosclerotic Carotid Plaques. *IEEE Transactions on Medical Imaging*, v. 22, n. 7, p. 902–912, 2003.
- [4] JOO, S.; YANG, Y. S.; MOON, W. K.; KIM, H. C. Computer-aided Diagnosis of Solid Breast Nodules: Use of an Artificial Neural Network Based on Multiple Sonographic Features. *IEEE Transactions on Medical Imaging*, v. 23, n. 10, p. 1292–1300, 2004.
- [5] DUA, R.; BEETNER, D. G.; STOECKER, W. V.; WUNSCH, D. C., I. Detection of Basal Cell Carcinoma Using Electrical Impedance and Neural Networks. *IEEE Transactions on Biomedical Engineering*, v. 51, n. 1, p. 66–71, 2004.
- [6] SUBASI, A.; ERÇELEBI, E. Classification of EEG Signals Using Neural Network and Logistic Regression. *Computer Methods and Programs in Biomedicine*, v. In Press, Corrected Proof, n. Available online 25 February 2005, 2005.
- [7] BHATIKAR, S. R.; DEGROFF, C.; MAHAJAN, R. L. A Classifier Based on the Artificial Neural Network Approach for Cardiologic Auscultation in Pediatrics. *Artificial Intelligence in Medicine*, v. In Press, Corrected Proof, p. Available online 21 December 2004, 2004.
- [8] ARGOUD, F. I. M. *Contribuição à Automatização da Detecção e Análise de Eventos Epileptiformes em EEG*. Tese (Doutorado em Engenharia Elétrica) - Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, 2001.

- [9] PANG, C.; UPTON, A.; SHINE, G.; KAMATH, M. A Comparison of Algorithms for Detection of Spikes in the Electroencephalogram. *IEEE Transactions on Biomedical Engineering*, v. 50, n. 4, p. 521–526, 2003.
- [10] IAIONE, F. *Detecção de Hipoglicemia Baseada na Análise e Classificação do Eletroencefalograma*. Tese (Doutorado em Engenharia Elétrica) - Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, 2003.
- [11] MOSHOU, D.; HOSTENS, I.; PAPAIOANNOU, G.; RAMON, H. Dynamic Muscle Fatigue Detection Using Self-Organizing Maps. *Applied Soft Computing*, v. In Press, Corrected Proof, p. Available online 17 November 2004, 2004.
- [12] SHYU, L.-Y.; WU, Y.-H.; HU, W. Using Wavelet Transform and Fuzzy Neural Network for VPC Detection from the Holter ECG. *IEEE Transactions on Biomedical Engineering*, v. 51, n. 7, p. 1269–1273, 2004.
- [13] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning Representations by Backpropagation Errors. *Nature*, London, v. 323, n. 9, p. 451–453, 1986.
- [14] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning Internal Representations by Error Propagation. In: *Parallel Distributed Processing*, 1986. MIT Press. v. 1. p. 319–362.
- [15] BRAGA, A. P.; LUDERMIR, T. B.; CARVALHO, A. C. P. L. F. *Redes Neurais Artificiais: Teoria e Aplicações*. Rio de Janeiro: Editora LTC, 2000.
- [16] DE AZEVEDO, F. M. Uma proposta de modelos formais de neurônios e redes neurais artificiais. In: *III CONGRESSO BRASILEIRO DE REDES NEURAI*S, Florianópolis 1997. p. 608–614.
- [17] BRASIL, L. M. *Proposta de Arquitetura para Sistema Especialista Híbrido e a Correspondente Metodologia de Aquisição de Conhecimento*. Tese (Doutorado em Engenharia Elétrica) - Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, 1999.
- [18] HAYKIN, S. *Redes Neurais: Princípios e Prática*. Porto Alegre: Bookman, 2004.
- [19] HOPFIELD, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. In: *Proceedings of the National Academy of Sciences*, 1982. v. 79. p. 2554–2558.

- [20] HOPFIELD, J. J. Neurons with Graded Response Have Collective Computational Properties like those of Two-state Neurons. In: *Proceedings of the National Academy of Sciences*, 1984. v. 81. p. 3088–3092.
- [21] NASCIMENTO JR, C. L.; ZARROP, M. B. Using the IAC Neural Network to Solve Optimization Problems. In: *Proceedings of the 2nd Brazilian Congress on Neural Networks*, 1995.
- [22] DE AZEVEDO, F. M. Contribution to the Study of Neural Networks in Dinamical Expert System. PHD Thesis - Facultés Universitaires Notre-Dame de la Paix (FUNDP), Namur, Bélgica, 1993.
- [23] LUA, K. T.; W., G. K. Recognizing Chinese Characters through Interactive Activation and Competition. *Pattern Recognition*, v. 23, n. 12, p. 1311–1321, 1990.
- [24] BURTON, A. M.; BRUCE, V.; JOHNSTON, R. A. Understanding Face Recognition with a Interactive Activation and Competition Model. *British Journal of Psychology*, v. 81, p. 361–380, 1990.
- [25] BARRETO, J. M.; DE AZEVEDO, F. M. Connectionist Expert System as Medical Decision Aid. *Artificial Intelligence in Medicine*, v. 5, p. 515–523, 1993.
- [26] JASZLICS, I.; JASZLICS, S.; JONES, S. Automated Battlefield Simulation Command and Control using Artificial Neural Networks. In: *Proceedings of the Fifth Annual Conference on Distributed Interactive Simulation Environments*, 1994. p. 100–105.
- [27] CHENG, K. J. Retrieval of Clinical Science Information Using an Interactive Activation and Competition Network. *Artificial Intelligence in Medicine*, v. 8, p. 359–375, 1996.
- [28] TAMBOURATZIS, T.; ANTONOPOULOS-DOMIS, M. A Non-Stationary Signal Correlator for On-line Transit Time Estimation. *Annals of Nuclear Energy*, v. 29, n. 11, p. 1299–1313, 2002.
- [29] NI, Z. W.; LU, Y. J.; LI, L. S.; JIA, R. Y. A Neural Network Case-Based Reasoning and its Application. *International Conference on Machine Learning and Cybernetics*, v. 1, p. 529–532, 2002.
- [30] GARCEZ, J. N.; OLIVEIRA, R.; ET AL. Módulo de Consulta Inteligente para Apoio à Decisão Sobre Ressarcimento de Danos ao Consumidor em Rede de Distribuição de Energia Elétrica. In: *Anais do XIV Congresso Brasileiro de Automática*, 2002. p. 1767–1772.

- [31] SALDÍAS, G. M. J. C.; DE AZEVEDO, F. M. Didactic Ergonomy for the Interface of Intelligent Tutoring Systems. In: *In: Computers and Education: Towards a Lifelong Learning Society*, 2003. p. 75–88.
- [32] BARBOSA, A. T. R. *Mecanismo de Adaptação baseado em Redes Neurais Artificiais em Sistemas Hípermidia Adaptativos*. Tese (Doutorado em Engenharia Elétrica) - Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, 2003.
- [33] DAZZI, R. L. S. Uma Nova Abordagem Multiagentes para a Construção de Sistemas Tutores Inteligente. Exame de qualificação (Doutorado em Engenharia Elétrica) - PGEEL - UFSC, 2004.
- [34] SALDÍAS, G. M. J. C. *Metodologia para a Construção de Interfaces Adaptáveis em Sistemas Tutores Inteligentes*. Tese (Doutorado em Engenharia Elétrica) - Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina (UFSC), Santa Catarina, 2002.
- [35] KURI, A. *A Comprehensive Approach to Genetic Algorithms in Optimization and Learning, Theory and Applications*. Cambridge, USA: Ed. Politecnico, 1999. v. 1.
- [36] KURI, A. A Universal Eclectic Algorithm for Constrained Optimization. In: *PROCEEDINGS OF 6TH EUROPEAN CONGRESS ON INTELLIGENT TECHNIQUES & SOFT COMPUTING, EUFIT'98*, 1998. p. 518–522.
- [37] HARPHAM, C.; DAWSON, E. C. W.; BROWN, E. M. R. A Review of Genetic Algorithms Applied to Training Radial Basis Function Network. *Neural Computing and Applications*, v. 13, p. 193–201, 2004.
- [38] BLANCO, A.; DELGADO, M.; PEGALAJAR, M. C. A genetic algorithm to obtain the optimal recurrent neural network. *International Journal of Approximate Reasoning*, v. 23, n. 1, p. 67–83, 2000.
- [39] BLANCO, A.; DELGADO, M.; PEGALAJAR, M. C. A Real-Coded Genetic Algorithm for Training Recurrent Neural Networks. *Neural Networks*, v. 14, n. 1, p. 93–105, 2001.
- [40] DIMOPOULOS, K. P.; KAMBHAMPATI, C.; CRADDOCK, R. Efficient Recurrent Neural Training Incorporating a priori Knowledge. *Mathematics and Computers in Simulation*, v. 52, n. 2, p. 137–162, 2000.
- [41] ALIEV, R. A.; FAZLOLLAHI, B.; VAHIDOV, R. M. Genetic Algorithm-Based Learning of Fuzzy Neural Networks. part 1: Feed-Forward Fuzzy Neural Networks. *Fuzzy Sets and Systems*, v. 118, n. 2, p. 351–358, 2001.

- [42] MCCLELLAND, J. L.; RUMELHART, D. E. *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs and Exercises*. Cambridge, USA: MIT Press, 1986. v. 1.
- [43] MCCLELLAND, J. L.; RUMELHART, D. E. An Interactive Activation Model of Context Effects in Letter Perception: Part 1, An Account of Basic Findings. *Psychological Review*, Washington, v. 88, p. 357–407, 1981.
- [44] SIGAKI, N. A. *Estudo da Influência da Base de Casos em Redes IAC na Implementação de Sistemas Especialistas*. Dissertação (Mestrado em Engenharia Elétrica) - Curso de Pós-Graduação em Engenharia Elétrica, UFSC, Santa Catarina, 1997.
- [45] BARRETO, J. M. *Inteligência Artificial no Limiar do Século XXI*. Florianópolis: ppp Edições, 1997.
- [46] GARDNER, H. Inteligências Múltiplas. *sss*, p. 1–2, 1983.
- [47] BURTON, M. A. Learning new faces in an Interactive Activation and Competition Model. *Visual Cognition*, p. 313–348, 1994.
- [48] BURTON, M. A.; BRUCE, V. From Pixels to People: A Model of Familiar Face Recognition. *Cognitive Science*, v. 23, p. 1–31, 1999.
- [49] DENNIS, S. The IAC Network: How Neural Networks Process Information. Disponível em: <<http://www2.psy.uq.edu.au/brainwav/Manual/IAC.html>>, Acesso em: 01 abril 2003. 2003.
- [50] MITCHELL, M. *An Introduction to Genetic Algorithms*. USA: MIT Press, 1998.
- [51] GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [52] HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. Cambridge, USA: MIT Press, 1975.
- [53] FOGEL, L. J.; OWENS, A. J.; WALSH, M. J. *Artificial Intelligence through Simulated Evolution*. Wiley, 1966.
- [54] RECHENBERG, I. Evolution Strategy. In: ZURADA, J. M.; MARKS, R. J.; ROBINSON, C. J. (Eds.) *Computational Intelligence - Imitating Life*, 1994. Editors ZURADA, J. M.; MARKS, R. J.; ROBINSON, C. J. IEEE Press. p. 147–159.
- [55] DE AZEVEDO, F. M. Algoritmos Genéticos em Redes Neurais Artificiais. In: *V ESCOLA DE REDES NEURAIAS*, São José dos Campos 1999.

- [56] LOPES, H. S. Algoritmos Genéticos em Projetos de Engenharia: Aplicações e Perspectivas Futuras. In: *Anais do IV Simpósio Brasileiro de Automação inteligente*, São Paulo, 1999. p. 64–74.
- [57] MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. 3. ed. New York: Springer, 1999.
- [58] MANTERE, T.; ALANDER, J. T. Evolutionary Software Engineering, a Review. *Applied Soft Computing*, v. 5, p. 315–331, 2005.
- [59] RENTERÍA, A. R.; PACHECO, M. A. C. Algoritmos Genéticos: Princípios e Aplicações. In: *V Congreso Internacional de Ingeniería Electrónica, Eléctrica Y Sistemas*, Lima, Peru, 1999. p. 11–16.
- [60] KALOGIROU, S. A. Artificial Intelligence for the Modeling and Control of Combustion Processes: A Review. *Progress in Energy and Combustion Science*, v. 29, p. 515–566, 2003.
- [61] HERRERA, F.; LOZANO, M.; VERDEGAY, J. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*, v. 12, p. 265–319, 1998.
- [62] JIANG, J.-H.; WANG, J.-H.; CHU, X.; YU, R.-Q. Clustering Data using a Modified Integer Genetic Algorithm (IGA). *Analytica Chimica Acta*, v. 354, p. 263–274, 1997.
- [63] DAHAL, K. P.; MCDONALD, J. R. Generator Maintenance Scheduling of Electrical Power Systems Using Genetic Algorithms with Integer Representation. In: *Proceeding of Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1997. p. 456–461.
- [64] DAHAL, K. P.; ALDRIDGE, C. J.; MCDONALD, J. R. Generator Maintenance Scheduling using a Genetic Algorithm with a Fuzzy Evaluation Function. *Fuzzy Sets and Systems*, v. 102, p. 21–29, 1996.
- [65] RENTERÍA, A. R.; PACHECO, M. A. C. Identificação Robusta de Modelos Lineares por Partes Através de Algoritmos Genéticos. In: *ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL (ENIA 2001)*, Fortaleza, Ceará, 2001.
- [66] TÚPAC, Y. J.; VELLASCO, M. M.; PACHECO, M. A. C. Planejamento e Otimização do Desenvolvimento de um Campo de Petróleo por Algoritmos Genéticos. In: *VIII International Conference on Industrial Engineering and Operations Management*, Curitiba, PR, BRASIL, 2002.

- [67] TAO, H.; LIAO, G.; WANG, L. Integer Coded Genetic Algorithm Design of Staggered Sampling MTI. In: *Proceeding of IEE Int. Conf. Neural Networks and Signal Processing*, Naging, China, 2003. p. 558–562.
- [68] DE MACEDO, R. A.; DA SILVA FILHO, D.; COURY, D. V.; DE FRANÇA MENDES CARNEIRO, A. A. An Evolutionary Optimization Approach to Track Voltage and Current Harmonics in Electrical Power Systems. In: *Power Engineering Society General Meeting*, 2003. v. 2. p. 1250–1255.
- [69] MAULIK, U.; BANDYOPADHYAY, S. Fuzzy Partitioning Using a Real-Coded Variable-Length Genetic Algorithm for Pixel Classification. *IEEE Transactions on Geoscience and Remote Sensing*, v. 41, n. 5, p. 1075–1081, 2003.
- [70] ACHICHE, S.; BARON, L.; BALAZINSKI, M. Real/Binary-like Coded Versus Binary Coded Genetic Algorithms to Automatically Generate Fuzzy Knowledge Bases: a Comparative Study. *Engineering Applications of Artificial Intelligence*, v. 17, p. 313–325, 2004.
- [71] KUOA, R.; CHEN, J. A Decision Support System for Order Selection in Electronic Commerce Based on Fuzzy Neural Network Supported by Real-Coded Genetic Algorithm. *Expert Systems with Applications*, v. 26, p. 141–154, 2004.
- [72] LIU, Z.; LIU, A.; WANG, C.; NIU, Z. Evolving Neural Network Using Real Coded Genetic Algorithm (GA) for Multispectral Image Classification. *Future Generation Computer Systems*, v. 20, p. 1119–1129, 2004.
- [73] CHANG, C.-C.; SHIH, K.-J.; LIN, I.-C. An intellectual placement of Chinese phonetic symbols. In: *Natural Language Processing and Knowledge Engineering*, 26-29 Oct. 2003. p. 506–511.
- [74] DA SILVA, A. A. Tutorial Genetics Algorithms. *Learning and Nonlinear Models*, v. 1, n. 1, p. 45–60, 2002.
- [75] TANOMARU, J. Motivação, Fundamentos e Aplicações de Algoritmos Genéticos. In: *Anais do II Congresso Brasileiro de Redes Neurais, III Escola de Redes Neurais*, Curitiba, 1995.
- [76] SCHMITT, L. M. Theory of Genetic Algorithms. *Theoretical Computer Science*, v. 259, p. 1–61, 2001.
- [77] BÄCK, T.; HOFFMEISTER, F. Extended Selection Mechanisms in Genetic Algorithms. In: *Proceedings of IV International Conference on Genetic Algorithms - ICGA*, San Diego, USA, 1991. p. 92–99.

- [78] BEASLEY, D.; BULL, D.; MARTIN, R. An Overview of Genetic Algorithms: Part 2, Research Topics. *University Computing*, v. 15, n. 4, p. 170–181, 1993.
- [79] MOORE, M. An Accurate Parallel Genetic Algorithm to Schedule Tasks on a Cluster. *Parallel Computing*, v. 30, p. 567–583, 2002.
- [80] LIENIG, J. A Parallel Genetic Algorithm for Performance-Driven VLSI Routing. *IEEE Transactions on Evolutionary Computation*, v. 1, n. 1, p. 29–39, 1997.
- [81] PARK, B. J.; CHOI, H. R.; KIM, H. S. A Hybrid Genetic Algorithm for the Job Shop Scheduling Problems. *Computers & Industrial Engineering*, v. 45, n. 4, p. 597–613, 2003.
- [82] CANTÚ-PAZ, E. A Survey of Parallel Genetic Algorithms. IlliGAL Technical Report 97003, 1997.
- [83] HECKERLINGA, P. S.; GERBERA, B. S.; TAPEC, T. G.; WIGTONC, R. S. Use of Genetic Algorithms for Neural Networks to Predict Community-Acquired Pneumonia. *Artificial Intelligence in Medicine*, v. 30, p. 71–84, 2004.
- [84] LI, J.; KANG, L.; WU, Z. An Adaptive Neighborhood-based Multi-parent Crossover Operator for Real-coded Genetic Algorithms. In: *The 2003 Congress on Evolutionary Computation. CEC '03.*, 8-12 Dec., 2003. v. 1. p. 14 – 21.
- [85] HILL, S.; NEWELL, J.; O'RIORDAN, C. Analysing the Effects of Combining Fitness Scaling and Inversion in Genetic Algorithms. In: *16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2004*, 15-17 Nov. 2004. p. 380–387.
- [86] RAMOS, M. T. V.; D'ACAMPORA, A. J.; DE AZEVEDO, F. M. Sistema Inteligente de Auxílio ao Diagnóstico de Hepatites Virais. In: *Anais do IV Workshop de Informática Médica Aplicada à Saúde - IV Congresso Brasileiro de Computação*, 2004. p. 595–598.
- [87] DE AZEVEDO, F. M.; FERRARI, G. L.; ANGELONI, M. N. M.; ARGOUD, F. I. M.; ALGARVE, A. S. Shell para Desenvolvimento de Sistemas Especialistas Fuzzy usando Prolog. In: *Anais do III Congresso Latino Americano de Engenharia Biomédica*, 2004. p. 919–92.
- [88] CHEN, C. *Introduction to Linear System Theory*. USA: Holt, Rinehart and Winston, Inc., 1970.
- [89] SHINNERS, S. M. *Control System Design*. USA: John Wiley & Sons, 1964.
- [90] D'AZZO, J.; HOUPIS, C. *Análise e Projeto de Sistemas de Controle Lineares*. Brasil: Editora Guanabara Dois Ltda., 1978.

- [91] CHEN, C. *Linear System Theory and Design*. USA: Holt, Rinehart and Winston, Inc., 1984.
- [92] BOLTON, W. *Engenharia de Controle*. Brasil: Makron Books, 1995.
- [93] GIROD, B.; RABENSTEIN, R.; STENGER, A. *Sinais e Sistemas*. Brasil: LTC - Livros Tecnicos e Cientificos Editora Ltda., 2003.
- [94] HAYKIN, S.; VAN VEEN, B. *Sinais e Sistemas*. Brasil: Editora Bookman, 2001.
- [95] ZURADA, J. M. *Artificial Neural Systems*. USA: West Publishing, 1992.
- [96] KOSKO, B. *Neural Networks and Fuzzy Systems - a dynamical systems approach to machine intelligence*. New Jersey: Prentice-Hall, 1992.
- [97] COHEN, M. A.; GROSSBERG, S. Absolute Stability of Global Pattern Formation an Parallel Memory Storage by Competitive Neural Networks. *IEEE Transactions on Systems, Man and Cybernetics*, v. 13, n. 5, p. 815–826, 1983.
- [98] WHITLEY, D. A Genetic Algorithm Tutorial. Disponível em: <http://samizdat.mines.edu/ga_tutorial/>, Acesso em: 22 de março de 2005. 2005.