

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

**SENSIBILIDADE PARAMÉTRICA COMO
GUIA PARA O TREINAMENTO HÍBRIDO DE
REDES NEURAIS**

Tese submetida como parte dos requisitos para a
obtenção do grau de Doutor em Engenharia Elétrica

João da Silva Dias

Florianópolis, Dezembro de 1998.

SENSIBILIDADE PARAMÉTRICA COMO GUIA PARA O TREINAMENTO HÍBRIDO DE REDES NEURAIAS

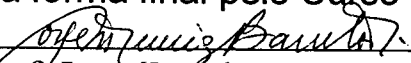
João da Silva Dias

Esta tese foi julgada adequada para obtenção do Título de

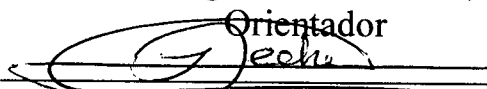
DOUTOR EM ENGENHARIA ELÉTRICA

**Área de concentração em
Sistemas de Informação**

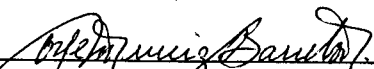
e aprovada em sua forma final pelo Curso de Pós-Graduação

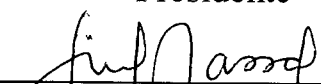

Prof. Jorge Muniz Barreto, Dr.

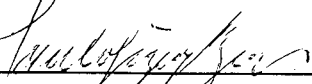
Orientador

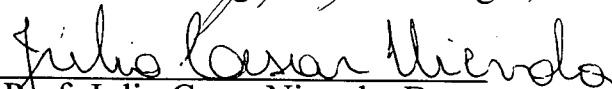

Prof. Idemar Cassana Decker, D. Sc.
Coordenador do Curso

Banca Examinadora:

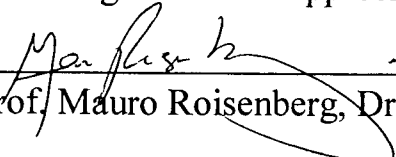

Prof. Jorge Muniz Barreto, Dr.
Presidente


Prof. Silvia Modesto Nassar, Dr.^a
Co-Orientadora


Prof. Paulo Sérgio da Silva Borges, Dr.


Prof. Julio Cesar Nievola, Dr.


Prof. Eugenio Kahn Epprecht, Dr.


Prof. Mauro Roisenberg, Dr.

À minha esposa CRISTIANE
À minha filha MARINA
Aos meus pais GIL e MARIA ALICE

AGRADECIMENTOS

Aos professores Jorge M. Barreto e Silvia M. Nassar pela orientação competente, apoio e amizade.

Ao professor Carlos Inácio Zanchin pelo incentivo dado desde o mestrado e também em reconhecimento pelo seu trabalho sério e competente.

Ao colega John Wisbeck pelas inúmeras discussões e divagações ao longo deste trabalho.

Ao colega Emil Kupek pelas discussões e apoio.

A todos os amigos que de alguma forma apoiaram este trabalho.

Aos brasileiros que parcialmente financiaram este trabalho através do Conselho Nacional de Pesquisa – CNPq.

PUBLICAÇÕES

1. Dias, João da Silva; Barreto, Jorge Muniz; Nassar, Silvia; e Brasil, L. M.. "Genetic and Back-Propagation Algorithm in Hybrid Training of Artificial Neural Networks: A Unimodal Search Procedure". Proceedings of the 16th IASTED - International Conference Applied Informatics – ISSN: 1027-2666, ISBN: 0-88986-250-8 February 23-25, 1998, Garmisch-Partenkirchen, Germany.
2. Dias, João da Silva; Barreto, Jorge Muniz; e Nassar, Silvia. "Algoritmo Genético Guiado pela Sensibilidade no Treinamento Híbrido de Redes Neurais com Back-Propagation". I Congresso Latinoamericano de Ingenieria Biomédica – MAZATLAN 98 – ISBN: 968-5063-00-1, Outubro de 1998, México.
3. Dias, João da Silva, Barreto, Jorge Muniz e Vilvahuamán, Luis A. "Otimização do Algoritmo Back-propagation com Algoritmo Genético". VII Simposio Latinoamericano de Ingeniería Biomédica, Bucaramanga - Colombia, Outubro de 1996.
4. Dias, João da Silva; Zimmermann, A. C.; Borges, P. S. da S.; Barreto, Jorge Muniz. "Aprendizado e Evolução: de Lamarch a Baldwin". I Simpósio Brasileiro de Redes Neurais – SBRN 98, Belo Horizonte, Dezembro de 1998.

5. Dias, João da Silva, Barreto, Jorge Muniz. "Heurísticas na Implementação de Algoritmo Genético no Treinamento de Redes Multicamadas". Submetido ao I Simpósio de Logística da Marinha, Rio de Janeiro, Dezembro de 1996.
6. Dias, João da Silva; e Barreto, Jorge Muniz. "Algoritmo Genético: Inspiração Biológica na Solução de Problemas – Uma Introdução". Pesquisa Naval – ISSN: 1414-8595 – Suplemento Especial da Revista Marítima Brasileira, N. 11 – Outubro de 1998, Rio de Janeiro.
7. Dias, João da Silva; Barreto, Jorge Muniz; e Nassar, Silvia. "Algoritmo Genético Guiado pela Análise de Sensibilidade". IV Fórum Nacional de Ciência e Tecnologia em Saúde – FNCTS 98 – ISBN: 85-7014-006-1, Outubro de 1998, Curitiba, Brasil.
8. D'angelo, Guido G.; Zimmermann, A. C.; Borges, P. S. da S.; Barreto, J. M.; e Dias, J. da S.. "Algoritmos de Redes Neurais como Filtro de Imagens Radiográficas". IV Fórum Nacional de Ciência e Tecnologia em Saúde – FNCTS 98 – ISBN: 85-7014-006-1, Outubro de 1998, Curitiba, Brasil.

SUMÁRIO

AGRADECIMENTOS.....	IV
PUBLICAÇÕES.....	V
SUMÁRIO.....	VII
LISTA DE FIGURAS.....	IX
LISTA DE TABELAS.....	X
LISTA DE ABREVIATURAS.....	XI
RESUMO.....	XII
ABSTRACT.....	XIII
1 INTRODUÇÃO.....	1
1.1 CONSIDERAÇÕES INICIAIS.....	1
1.2 OBJETIVOS.....	4
1.2.1 <i>Objetivo Geral</i>	4
1.2.2 <i>Objetivos Específicos</i>	5
1.3 PROPOSTA DO TRABALHO.....	5
1.4 ESPECIFICAÇÃO DO DOMÍNIO DA APLICAÇÃO "BENCHMARK".....	8
1.5 DESCRIÇÃO DO DOMÍNIO DA APLICAÇÃO.....	10
1.6 SUPERFÍCIE DE ERRO DO XOR.....	11
1.7 PONTOS ORIGINAIS DO TRABALHO.....	13
1.8 ESTRUTURA DO TRABALHO.....	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 REDES NEURAIS.....	15
2.1.1 <i>Introdução</i>	15
2.1.2 <i>Modelo Generalizado do Neurônio</i>	18
2.1.3 <i>Redes Multicamada do Tipo Direta (RMD)</i>	19
2.1.4 <i>Algoritmo de Treinamento</i>	21
a) <i>Algoritmo de Retropropagação</i>	22
b) <i>Algoritmo Quickprop</i>	25
2.1.5 <i>Problemas no Treinamento com o Algoritmo de Retropropagação</i>	26
a) <i>Mínimo Local</i>	26
b) <i>Paralisia</i>	27
2.2 COMPUTAÇÃO EVOLUTIVA.....	28
2.2.1 <i>Inspiração Biológica para Evolução e Aprendizado</i>	31
2.2.2 <i>Conceitos Fundamentais e Terminologia do Algoritmo Genético</i>	35
2.2.3 <i>Operadores Genéticos (OGs)</i>	38
a) <i>Cruzamento ("crossover")</i>	39

b) <i>Mutação</i>	41
2.2.4 <i>Operação do Algoritmo Genético</i>	43
a) <i>Geração da População Inicial</i>	43
b) <i>Adaptação da População</i>	44
c) <i>Convergência</i>	45
d) <i>Seleção</i>	46
e) <i>Aplicação dos Operadores Genéticos</i>	49
f) <i>Nova População</i>	50
2.2.5 <i>Teorema Fundamental do Algoritmo Genético</i>	51
2.2.6 <i>Evolução e Aprendizado Individual</i>	58
2.2.7 <i>Inclusão do Aprendizado</i>	61
a) <i>Aprendizado Lamarckiano</i>	62
b) <i>Aprendizado Baldwiniano</i>	63
2.3 ESTADO DA ARTE NO TREINAMENTO HÍBRIDO DE RNA	64
2.3.1 <i>Introdução</i>	64
a) <i>Otimização da Topologia</i>	64
b) <i>Otimização do Treinamento</i>	65
2.3.2 <i>Revisão da Literatura</i>	67
2.4 SENSIBILIDADE	71
2.4.1 <i>Gradiente Descendente (GD)</i>	72
2.4.2 <i>Variação dos Pesos (VP)</i>	73
3 METODOLOGIA DA PROPOSTA	75
3.1 INTRODUÇÃO	75
3.2 IMPLEMENTAÇÕES DA BUSCA UNIMODAL	77
a) <i>Busca Unimodal Direta</i>	78
b) <i>Busca Unimodal Reversa</i>	80
c) <i>Busca Unimodal com Sensibilidade Global</i>	80
d) <i>Busca Unimodal Parcial com Sensibilidade</i>	81
e) <i>Busca Unimodal Parcial com Sensibilidade por Padrão</i>	82
3.3 INSPIRAÇÃO DA PROPOSTA DE BUSCA UNIMODAL	84
a) <i>Inspiração Biológica</i>	84
b) <i>Inspiração Matemática</i>	86
4 RESULTADOS	88
4.1 REPRESENTATIVIDADE DA FUNÇÃO CUSTO	88
4.2 A ANÁLISE DE SENSIBILIDADE E A DISTRIBUIÇÃO DO CONHECIMENTO SOBRE OS PESOS DA REDE	89
4.2.1 <i>Delineamento do Experimento</i>	90
4.2.2 <i>Amostra</i>	91
4.2.3 <i>Resultados</i>	92
4.2.4 <i>Conclusões do experimento</i>	94
4.3 TRANSIÇÃO ENTRE O ALGORITMO GENÉTICO E O ALGORITMO BASEADO NO GRADIENTE DESCENDENTE	94
4.4 CRUZAMENTO UNIFORME E MUTAÇÃO: O QUE HÁ DE COMUM?	98
4.5 ASPECTOS GERAIS DA TÉCNICA HÍBRIDA	100
4.6 ALGORITMO DE RETROPROPAGAÇÃO E QUICKPROP	103
5 CONCLUSÕES	106
5.1 CONCLUSÕES	106
5.2 TRABALHOS FUTUROS	109
6 REFERÊNCIAS BIBLIOGRÁFICAS	111

LISTA DE FIGURAS

Figura 2. 1 Modelo do neurônio artificial	18
Figura 2.2 Exemplo de rede multicamada do tipo direta com 3 camadas de neurônios e 2 camadas de pesos	20
Figura 2.3 Paralisia devido a presença de planalto	28
Figura 2.4 Exemplo de cruzamento uniforme	40
Figura 2.5 Exemplo de cruzamento com 1-partição	41
Figura 2.6 Exemplo de cruzamento com 2-partições	41
Figura 2.7 Exemplo de mutação (troca simples).	42
Figura 2.8 Ciclo básico do algoritmo genético.	43
Figura 2.9 Comparação da aptidão: roleta simples e ponderada	47
Figura 2.10 Pressão seletiva da evolução e do aprendizado	60
Figura 2.11 Variação dos pesos originais	74
Figura 3.1 Codificação do cromossomo conforme sequência dos pesos	78
Figura 3.2 Representação do cromossomo e características	85
Figura 4.1 Histograma da distribuição dos pesos	92
Figura 4.2a Intervalo de confiança para a média dos pesos segundo o Gradiente Descendente	93
Figura 4.2b Intervalo de confiança para a média dos pesos segundo a Variação dos Pesos	94
Figura 4.3 Detalhamento da metodologia de transição do algoritmo genético para o algoritmo de retropropagação	96
Figura 4.4 Desempenho comparativo do cruzamento e mutação para uma população de 20 indivíduos	99
Figura 4.5 Desempenho comparativo do cruzamento e mutação para uma população de 100 indivíduos	99

LISTA DE TABELAS

Tabela 1	Relação da terminologia do AG com a Biologia	37
Tabela 2	Dados para roleta simples	47
Tabela 3	Roleta ponderada	48
Tabela 4	Indivíduos x adaptação	52
Tabela 5	Exemplo de um indivíduo e 4 possíveis esquemas	53
Tabela 6	Exemplo de um indivíduo e 2 possíveis esquemas	55
Tabela 7	Valores de média e desvio padrão para o gradiente descendente e a variação dos pesos	93

LISTA DE ABREVIATURAS

AE	Algoritmo Evolutivo
AG	Algoritmo Genético
AGS	Algoritmo Genético Simples
AS	Análise de Sensibilidade
CE	Computação Evolutiva
ECG	Eletrocardiograma
EE	Estratégia Evolutiva
EEG	Eletroencefalograma
EMG	Eletromiograma
GD	Gradiente Descendente
IA	Inteligência Artificial
IAC	Inteligência Artificial Conexionista
OAG	Operadores do Algoritmo Genético
OG	Operador Genético
P_i	Probabilidade de seleção do indivíduo i
P_c	Probabilidade de sofrer cruzamento
PE	Programação Evolutiva
PG	Programação Genética
PID	Controlador proporcional integral derivativo
P_m	Probabilidade de sofrer mutação
P_{sc}	Probabilidade de sobrevivência após cruzamento
P_{sm}	Probabilidade de sobrevivência após mutação
QP	Quickprop
RMD	Redes Multicamada do tipo Direta
RNA	Redes Neurais Artificiais
RNN	Redes Neurais Naturais
RoIP	Roleta Ponderada
RP	Retropropagação
RS	Roleta Simples
SEP	Sensibilidade do Erro em relação ao Peso
SMI	Sistema Multivariável Interdependente
SNC	Sistema Nervoso Central
VP	Variação dos Pesos

RESUMO

A técnica híbrida de treinamento de redes neurais artificiais através do algoritmo genético e do algoritmo de retropropagação apresenta limitações de convergência e é restrita a pequenas redes. Este trabalho discute as razões destas deficiências e baseado nos resultados obtidos, propõe a análise de sensibilidade como forma de guiar o algoritmo genético para uma condição do espaço de busca mais promissora.

A principal alteração proposta é a abordagem unimodal do algoritmo genético utilizando a sensibilidade paramétrica como guia. Ao contrário da maioria das propostas, esta processará somente alguns pesos do conjunto segundo a seqüência estipulada pela análise de sensibilidade. Os pesos serão processados individualmente, enquanto os demais pesos serão mantidos fixos. Finalizada a última geração, o melhor indivíduo é inserido no conjunto de pesos e mantido fixo, enquanto que o próximo peso é processado. Isto ocorrerá até que todos os pesos da rede sejam processados pelo algoritmo genético. Ao final, ter-se-á passado uma etapa. Após completar um certo número de etapas, o melhor conjunto de pesos é repassado a um algoritmo baseado na técnica de gradiente descendente, para que este continue o ajuste fino do conjunto de pesos.

Entre os estudos efetuados cita-se o da distribuição de conhecimento sobre a rede neural, bem como a forma de utilizar esta informação como guia para a busca a ser realizada pelo algoritmo genético, além de discutir a viabilidade do emprego do algoritmo de retropropagação na técnica híbrida com o algoritmo genético.

Palavras-Chave: Algoritmo Genético, Redes Neurais, Inteligência Artificial, Treinamento Híbrido de Redes Neurais, Análise de Sensibilidade.

ABSTRACT

Parametric Sensitivity as Guide to Hybrid Training of the Neural Network

Hybrid techniques to training artificial neural networks through genetic algorithm and back-propagation, present convergence limitations and are restricted to small nets. This work discusses the reasons of these deficiencies and, based on the results obtained, proposes the sensitivity analysis as a form of guiding the genetic algorithm to achieve the convergence faster. This leads to a methodology of incorporating the sensitivity to the genetic algorithm.

The main alteration proposal is the unimodal approach of the genetic algorithm using a parametric sensitivity as guide. Unlike most other proposals, it processes only some of the weights according to the sequence specified by the sensitivity analysis while the rest is kept fixed. At the end of the last generation, the best individual weight is kept fixed, while the next one is being processed. This cycle is repeated until the genetic algorithm processes all the network weights, which constitutes a phase. After completing a certain number of phases, the best group of the weights is passed on to an algorithm based on the technique of gradient descent to continue their fine adjustment.

Among the performed studies it is also cited the distribution of the knowledge about the neural network, as well as the form of using this information as guide for the search to be accomplished by the genetic algorithm.

Key-words: Genetic Algorithm, Neural Networks, Artificial Intelligence, Hybrid Training for Neural Network, Sensitivity Analysis

1 INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

A motivação para a busca de inspiração nos neurônios e no sistema nervoso central (SNC) para a resolução de problemas deve-se à elevada capacidade do cérebro em realizar certas atividades corriqueiras, como a classificação de padrões. Além desta, há problemas que são resolvidos mais efetivamente pelo cérebro humano do que por computadores. Tipicamente esses problemas têm duas características em comum: eles geralmente são problemas mal definidos, e via de regra requerem enorme quantidade de processamento[1].

As redes neurais artificiais (RNA) nasceram da tentativa de imitar as redes neurais naturais (RNN). Inicialmente esperava-se que estes modelos matemáticos apresentassem características semelhantes às RNN, apresentando bom desempenho em tarefas não algorítmicas e dificuldades com as de natureza intrinsecamente algorítmica [2,3].

A RNA apresenta-se como um sistema computacional distribuído, composto de um número de elementos de processamento (neurônios) operando em paralelo, conectados conforme alguma topologia específica (arquitetura) e tendo a capacidade de auto-ajustar a intensidade de suas conexões (peso) e os parâmetros dos elementos de processamento (treinamento). Em geral a rede neural representa uma função de processamento de alguma informação (reconhecimento de padrões, compressão de dados, etc.). As RNAs podem aprender regras complexas. Elas diferem em sua eficiência e rapidez de treinamento, sua habilidade de fazer distinções, sua capacidade em generalizar, e o tipo de máquina ou hardware no qual elas podem ser executadas.

Dentre as vantagens potenciais de um sistema maciçamente paralelo, destaca-se a velocidade de processamento, bem como a habilidade de processar dados com ruído, dados corrompidos ou com perdas [4].

O emprego de RNAs tem-se mostrado promissor principalmente onde o ambiente apresenta modelo complexo, não-linear ou com presença de incertezas. Portanto, as RNAs são um bom paradigma para resolver problemas cuja solução algorítmica seja desconhecida ou em que exista uma má definição do problema. As aplicações de RNA inicialmente desenvolveram-se com maior ênfase em áreas tais como reconhecimento de padrões e na área da saúde, onde a maioria das técnicas clássicas falhavam. Mais tarde, as RNAs começaram a entrar em áreas onde as técnicas clássicas dominavam ou não existiam técnicas gerais para problemas não-lineares, como é o caso dos sistemas de controle [5].

Na área médica é crescente a utilização de técnicas de Inteligência Artificial (IA)¹, tanto no processamento de sinais biomédicos (eletroencefalograma (EEG), eletrocardiograma (ECG), eletromiograma (EMG), etc.) como em sistemas de apoio a decisão médica [6-10]. As RNAs apresentam-se como uma ferramenta poderosa no tratamento destes sinais e principalmente na sua análise, onde as dificuldades para o ser humano são elevadas devido principalmente a quantidade e qualidade dos dados disponíveis .

A escolha de uma classe de redes neurais, antes de mais nada, é uma função do problema ao qual ela se destina. No tocante à topologia, existem duas principais subdivisões de redes, as diretas (feed-forward) e as recorrentes (feedback)[11]. Enquanto que uma rede recorrente pode ser utilizada tanto para problemas dinâmicos quanto para estáticos, as redes estáticas destinam-se exclusivamente aos problemas estáticos. Pelo fato deste trabalho estar vinculado à área de Engenharia Biomédica, mesmo não tendo sido utilizado um problema específico da área, há preocupação de que o conhecimento adquirido seja utilizado na área na qual este trabalho se vincula. Assim, em linhas gerais, pretende-se que o conhecimento gerado

¹ "IA é o estudo das faculdades mentais com o uso de modelos computacionais"[104]

seja aplicável em RNAs para processamento de informação biomédica². Este tipo de problema pode ser caracterizado por uma função associando a cada sinal um significado dentro de um conjunto de possíveis significados. Este é um problema tipicamente estático. Portanto, se poderia escolher qualquer uma das duas classes de redes (estática ou recorrente). No entanto, as dificuldades de trabalho e a possível instabilidade no treinamento por parte das redes recorrentes, aliada à facilidade de uso das redes estáticas, tornam estas redes preferíveis [2].

Dentre os vários tipos de RNAs descritas até o momento, as redes multicamada do tipo direta (RMD), proposta inicialmente por Rumelhart et al em 1986[12], têm-se mostrado práticas e interessantes para aplicações nas mais variadas áreas do conhecimento.

Após a definição da rede a ser utilizada, classe, tipo e topologia, é necessário escolher o método de treinamento da rede. Dentre muitos métodos de treinamento de uma RMD, destaca-se o algoritmo de treinamento de retropropagação (*back-propagation*).

Basicamente, o algoritmo de treinamento de retropropagação (RP) é uma técnica de otimização que usa o gradiente descendente, buscando minimizar algum critério de erro. No entanto, este método não garante ser imune a falha, podendo apresentar problemas, principalmente devido à paralisia da rede, mínimos locais e lenta convergência. A paralisia da rede está ligada, por exemplo, ao valor dos pesos. Valores elevados para os pesos podem levar a função de ativação a saturação e devido a falta de habilidade do algoritmo de RP em manipular estes valores, a convergência é lenta ou pode ocorrer até mesmo a paralisia da rede. O problema dos mínimos locais está relacionado com a quantidade e qualidade dos mínimos locais existentes na superfície de erro. Como o algoritmo de treinamento de RP baseia-se na técnica do gradiente descendente, pode acontecer de, na busca de um mínimo global, passar por um mínimo local mais profundo e não mais conseguir sair. Desta forma, no caso de problemas com muitos mínimos

² Conhecimento adquirido por determinado meio, podendo ser: sinal, sintoma e/ou exames laboratoriais.

locais a convergência pode ser lenta e haver uma convergência para um destes mínimos locais. No entanto, passada a fase de inicialização e na proximidade de um mínimo global, a técnica do gradiente descendente é bastante eficaz.

Já o algoritmo genético (AG) é uma abordagem para problemas complexos de busca e treinamento através de modelos computacionais de processos evolutivos. O AG constitui-se em um ótimo processo de busca da região onde a probabilidade de se encontrar o mínimo global é maior, tendo em vista que este trabalha com uma população de indivíduos candidatos a solução e não somente com um único indivíduo.

Com este método atenuam-se os dois principais problemas do algoritmo de RP que são a paralisia e os mínimos locais. Porém, a convergência na região próxima ao mínimo global é lenta, tendo em vista que a convergência ocorre com praticamente toda a população. Logo, a técnica de AG é recomendável para iniciar a busca de uma região de provável localização do mínimo global, enquanto que o algoritmo de RP é indicado para a finalização do processo de busca ou ajuste fino.

O treinamento híbrido de redes neurais utilizando o AG e o RP visa aumentar a imunidade do treinamento à paralisia e aos mínimos locais. Contudo, apesar da aparente possibilidade de sucesso, a junção destas técnicas não tem respondido às expectativas. Torna-se relevante a análise destas técnicas, visando identificar os seus pontos críticos e principalmente responder se a ferramenta é adequada ou não a tarefas como o treinamento de redes neurais, ou se é mal empregada.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Estudar o treinamento híbrido de redes neurais artificiais com o algoritmo genético e a técnica baseada no gradiente descendente, levantando e propondo alternativas para os problemas da técnica híbrida.

1.2.2 Objetivos Específicos

Dentre os objetivos específicos a serem buscados, cita-se:

- a) Estudar a influência da representatividade da função custo no desempenho do algoritmo genético;
- b) Desenvolvimento de uma proposta de abordagem de sistemas interdependentes por algoritmo genético (como é o caso do conjunto de pesos);
- c) Analisar a diferença de sensibilidade dos pesos entre as camadas e a melhor forma de abordar esta sensibilidade com o algoritmo genético;
- d) Propor uma metodologia para determinar o momento da transição automática e otimizada entre o algoritmo genético e o algoritmo de retropropagação;
- e) Analisar o desempenho do AG com a utilização do cruzamento uniforme ao invés do cruzamento com 1-partição;
- f) Estudar as características e o tamanho da população, número de etapas e a taxa de cruzamento e de mutação;
- g) Avaliar a importância da diversidade da população no processo de busca para a proposta em questão;
- h) Avaliar a influência das técnicas baseadas no gradiente descendente no desempenho da proposta híbrida com algoritmo genético.

1.3 PROPOSTA DO TRABALHO

Na maioria das técnicas híbridas de treinamento de RNA, envolvendo o AG e o RP, os pesos formam um cromossomo e este é processado pelo algoritmo genético com o auxílio da função custo representada pelo inverso do erro da época, que é tomado como o erro médio quadrático de todos os padrões do conjunto de treinamento. Desta forma todos os pesos, ou cada parte deste (bit), tem a mesma probabilidade de serem processados geneticamente. Isto seria relevante se todos os pesos tivessem o mesmo grau de importância em todos os momentos do treinamento. Esta técnica será denominada daqui por diante de técnica convencional.

A técnica convencional foi estudada de forma bastante consistente por Kitano[13] que através dos resultados de vários experimentos avaliou o desempenho do treinamento de RNA usando AG e a técnica baseada no gradiente descendente (GD), como é o caso do algoritmo de retropropagação (RP). Neste trabalho, ele afirma que no tocante ao treinamento de RNA a proposta híbrida de treinamento, AG e RP, sistematicamente é melhor do que somente com AG e inferior ou próximo da variante mais rápida do RP, o QuickProp [14]. Contudo, o desempenho degrada a medida que aumenta o número de conexões da rede. Com os resultados obtidos após vários experimentos sob a técnica convencional, Kitano afirmou que: "...tarefas como treinamento de RNA's não são boas tarefas para AG devido a interdependência entre as regiões do cromossomo". Conclusões semelhantes são relatadas por Whitley³ [15].

A partir da implementação da técnica convencional de treinamento híbrido de RNA através do AG e do algoritmo baseado em GD, observou-se o comportamento descrito por Kitano[13]. Contudo, a conclusão de que a ferramenta (AG) não era apropriada ao problema (treinamento de RNA), não foi visualizada.

O presente trabalho inicia-se a partir da premissa de que a deficiência não reside na ferramenta e sim na forma como esta é empregada. Assim, apresentar-se-á como um dos motivos para a falha na utilização do AG, a função custo comumente empregada (inverso do erro médio quadrático dos padrões de treinamento)[4,16-18]. Apesar de sua representatividade para o conjunto de pesos, ela não informa qual ou quais os pesos são significativos no momento da avaliação. Isto implica em dizer que não se sabe a priori as regiões do espaço de busca mais promissoras. Este problema é tanto mais relevante quanto maior for o número de pesos constituintes da rede neural. Portanto, falta uma orientação ao AG no seu trabalho de aproximar o conjunto de pesos iniciais do provável local onde localiza-se um mínimo global.

³ O domínio da aplicação adotado por Kitano e por Whitley são diferentes, conforme será visto adiante.

Após a percepção de que a escolha da função custo poderia ser um grande problema para a implementação do AG, estudou-se o comportamento dos pesos aleatoriamente gerados e distribuídos na topologia de uma RNA, segundo um conjunto de padrões de treinamento para um problema específico. Estes estudos foram executados principalmente com o auxílio da análise de sensibilidade (AS) a qual foi utilizada para avaliar a contribuição de cada peso (do conjunto de pesos que compõe a rede) para o erro médio quadrático produzido pelo conjunto de padrões de treinamento [19].

A aplicação da AS mostrou que é possível determinar regiões do espaço de busca mais promissoras para o treinamento através do AG. Com esta possibilidade e com base nas premissas anteriores, estudou-se várias alternativas complementares à forma tradicional de aplicar o AG no treinamento híbrido de RNA.

A inovação da sistemática apresentada aqui está na forma como o AG atuará sob cada peso do conjunto que compõe a rede, bem como no conjunto de heurísticas empregadas.

O AG, ao contrário da maioria das propostas, não irá abordar o conjunto de pesos como sendo um único indivíduo. Será realçada a individualidade de cada peso no conjunto, o qual será trabalhado pelo AG individualmente, enquanto que os demais são mantidos fixos. Apresenta-se algumas formas de como o AG deve atuar sobre o conjunto de pesos, mas de uma forma geral, somente alguns pesos do conjunto é que são treinados a cada geração pelo AG. Os pesos que serão processados pelo AG são escolhidos conforme a sensibilidade do erro global do conjunto de treinamento a cada peso da rede. No início de cada geração, após a aplicação dos operadores do AG, a análise de sensibilidade é refeita para definir quais os pesos que serão processados nesta geração. Além da sistemática de atuação do AG em cada peso do conjunto, este trabalho preocupa-se com os detalhes envolvidos no processo de otimização, principalmente na atuação do AG e na transição entre o AG e o GD. A preocupação com o AG justifica-se por ser este o encarregado pela busca

inicial do conjunto de pesos, pois da qualidade da busca e dos valores de pesos a serem passados ao GD depende o sucesso do treinamento.

Foram estudados vários detalhes da implementação do AG, seja buscando a sua própria otimização bem como o desempenho da técnica híbrida de treinamento, AG e GD. Quanto ao GD comparou-se o desempenho do RP com o QuickProp. A comparação efetuada não visou a velocidade de convergência, a qual foi exaustivamente apresentada em vários trabalhos [13,14,20-22]. A preocupação aqui reside na qualidade dos pesos entregues pelo AG ao RP ou qualquer outro algoritmo baseado em GD e a habilidade deste em manipular estes pesos. Esta preocupação justifica-se por ser comum a passagem de valores numéricos elevados pelo AG ao RP, o que conduz a uma lenta velocidade de convergência devido aos valores obtidos pela derivada da função de ativação.

Dentre os pontos originais deste trabalho destacam-se:

- a) Implementação da análise de sensibilidade como guia para o algoritmo genético no espaço de busca do treinamento de RNA;
- b) Proposta e análise de viabilidade da implementação da variação dos pesos como alternativa ao cálculo da análise de sensibilidade clássica;
- c) Determinação da localização preferencial do conhecimento na camada de saída de uma rede 2-2-1 para o domínio da aplicação do XOR;
- d) Uma nova visão da inspiração biológica aplicada ao algoritmo genético;
- e) Abordagem unimodal do algoritmo genético em sistemas multivariáveis.

1.4 ESPECIFICAÇÃO DO DOMÍNIO DA APLICAÇÃO “BENCHMARK”

A fase inicial, onde foram avaliadas de forma preliminar algumas heurísticas, também tinha como objetivo a especificação de um domínio de aplicação “benchmark”. Assim, no transcorrer dos trabalhos foram levantados vários requisitos necessários, tais como:

- a) Ser não linearmente separável: para forçar o algoritmo de treinamento a trabalhar todos os pesos por necessidade do problema e não por potencialidade do algoritmo;

- b) Ser matematicamente bem definido, para que em caso de dúvidas quanto ao desempenho de alguma variável dos experimentos seja possível, ou no mínimo facilitada, a identificação da causa do sintoma observado;
- c) Ser simples o bastante para que seja possível uma depuração rápida e confiável do software em caso de dúvida e/ou erro;
- d) Possuir uma boa base de informação sobre o treinamento realizado por outros autores, para que possam ser comparadas e/ou explicadas particularidades dos desempenhos obtidos durante os experimentos e desenvolvimento do presente trabalho;
- e) Possuir um conjunto de treinamento viável e representativo: viável a nível de quantidade de dados, o que poderia inviabilizar uma depuração completa, e representativo quanto à completude do conjunto de treinamento em relação ao problema;
- f) O mesmo problema deve ser utilizado em todos os experimentos para que na medida do possível os resultados parciais permitam a superposição das informações e heurísticas encontradas.

O método de treinamento híbrido proposto pode ser considerado como sendo o de aprendizado supervisionado. O AG, por trabalhar com uma população, tem forte tendência a eliminar problemas de paralisia, soluções locais, etc. Além disso, sua aplicação não exige derivabilidade da função erro com relação aos valores das conexões sinápticas. Por outro lado, o algoritmo de RP ou outro algoritmo de treinamento baseado no gradiente exige que a função erro seja derivável com relação aos valores dos pesos. Assim, o algoritmo proposto e estudado exige apenas derivabilidade da função erro para valores próximos da solução.

A partir dos requisitos levantados notou-se que a maioria dos problemas ou tinha um conjunto de treinamento pouco representativo ou muito grande, o que dificultava em muito a compreensão do desempenho durante o treinamento realizado com observações em tempo real. Desta forma, optou-se por manter o domínio da aplicação no problema do ou-exclusivo (XOR). Este atende plenamente os requisitos levantados, além de

possuir extensa base de comparação com outras publicações [23-28]. Logo, uma vez definido o domínio a ser utilizado nas avaliações, torna-se importante compreender a superfície de erros produzida pelo XOR. Este conhecimento será utilizado de forma a direcionar os experimentos a serem realizados no transcorrer deste trabalho.

1.5 DESCRIÇÃO DO DOMÍNIO DA APLICAÇÃO

O ou-exclusivo ou XOR é um dos mais populares “benchmarks” para treinamento de RNA. Este “benchmark” tomou notoriedade com a exposição realizada por Minsky and Pappert no livro *Perceptrons*[25], onde os autores provaram que o perceptron não era capaz de implementar o XOR por este ser linearmente não separável.

Outro fator que torna o XOR relevante é a sua utilização como parte da implementação de um autômato da paridade. Enquanto que a determinação da paridade através de uma rede neural do tipo direta permite a determinação da paridade em seqüências fixas de 0's e 1's, o autômato da paridade pode ser utilizado em aplicações onde o comprimento da seqüência seja variável. Isso porque este aprende o conceito da paridade [29].

A implementação do XOR geralmente é realizada através de duas arquiteturas:

- a) Rede 2-2-1 [23]: é composta de três camadas de neurônios sendo uma de entrada com dois neurônios, uma intermediária também com dois neurônios e a última com um neurônio. Com a utilização dos três pesos de “bias”, há 9 pesos treináveis;
- b) Rede 2-1-1 [23,27,28]: também é composta de três camadas sendo que na camada de entrada há dois neurônios, na intermediária um neurônio e na de saída também um neurônio. A particularidade desta arquitetura é que os dois neurônios da entrada possuem conexão direta com o neurônio da saída. Esta rede tem 7 pesos treináveis ao todo, incluindo os dois pesos do “bias”.

O treinamento geralmente é executado até que a rede responda perfeitamente aos 4 padrões do conjunto de treinamento. A precisão requerida para valores de saída binário varia de estudo para estudo. Isto geralmente dificulta as comparações. Para uniformidade dos resultados, alguns pesquisadores têm sugerido usar o critério 40-20-40 para saídas booleanas o reinício do treinamento quando não houver convergência em tempo razoável. Este critério, em outras palavras, propõe que os valores inferiores a 40% da faixa de saída são considerados nível lógico zero. Valores de saída superiores a 60% da faixa de saída é considerado nível lógico um. Os 20% restantes ficam entre o nível lógico zero e um, criando uma faixa de valores indeterminados. Este critério é largamente empregado por não exigir extrema precisão na saída (carga computacional) mas requer que os valores de saída sejam distintos o bastante para uma certa quantidade de imunidade ao ruído. Os valores estipulados pelo critério 40-20-40 não devem ser confundidos com os valores desejados que são utilizados para calcular o erro a ser utilizado na retropropagação [30].

1.6 SUPERFÍCIE DE ERRO DO XOR

A preocupação com a forma da superfície de erro é importante principalmente quando o algoritmo de treinamento a ser empregado é baseado nas técnicas de gradiente descendente. Vários pesquisadores têm analisado a superfície de erro criada pelo conjunto de pesos pertinentes ao XOR[23,24,27,28].

Sprinkhuizen-Kuyper e Boers [27] realizaram um dos estudos analiticamente mais completos sobre superfícies de erro. Os autores iniciaram provando que um mínimo global é um ponto estacionário estável e caracteriza-se por apresentar gradiente zero para todos os padrões individualmente. O mínimo instável foi definido como sendo o mínimo onde o gradiente é zero para todo o conjunto de treinamento, mas não para cada padrão separadamente. Esta distinção é importante, desde que uma solução exata pode ser representada pela rede, então somente o mínimo absoluto com erro zero é mínimo estável, sendo os demais mínimos instáveis. O fato

de que todo mínimo local é instável, pode ser explorado pelo algoritmo de treinamento para escapar deste mínimo. Os pontos de sela por exemplo, apresentam gradiente zero para o erro de um conjunto fixo de padrões de treinamento, mas não para o erro dos padrões individualmente. Assim, o treinamento “on-line” pode escapar destes pontos[27].

Para o XOR com arquitetura 2-1-1, Sprinkhuizen-Kuyper e Boers[27] estudaram-na e afirmaram que a superfície de erro da rede não apresenta mínimos locais, somente ponto de sela, onde alguns algoritmos podem paralisar o treinamento ou retardá-lo. A solução correta com erro zero será encontrada no limite, com probabilidade 1[27].

Já para as redes 2-2-1, Sprinkhuizen-Kuyper e Boers[27] provaram analiticamente que a superfície de erro possui um mínimo global estável com erro zero e não apresenta mínimos locais, desde que os pesos sejam finitos. Mas se houver pesos tendendo ao infinito, aparecem regiões com mínimos locais as quais são envolvidas por pontos de sela, que caracterizam-se por serem pontos estacionários e erro não igual a zero[26-28].

Quando um ponto de sela é encontrado, o processo de treinamento em lote (ajuste dos pesos só ocorre após a apresentação de todos os padrões de treinamento) com momento igual a zero pode paralisar. Mas com o treinamento “on-line” (ajuste dos pesos após a apresentação de cada padrão de treinamento) provavelmente pode escapar do ponto de sela. Experimentos conduzidos iniciando o treinamento “on-line” exatamente sob um ponto de sela, com taxa de treinamento pequena (0,01) e momento igual a zero, o treinamento escapou do ponto de sela e buscou uma solução com erro zero num tempo finito. Já com o processamento em lote, nenhum progresso foi feito que permitisse escapar do ponto de sela. A paralisia do treinamento nestes pontos de sela é que tem levado alguns pesquisadores a afirmar que há mínimos locais em problemas como o XOR [27].

Como pode ser visto, é raro encontrar um problema “benchmark” com tal quantidade e qualidade de informações. Algumas avaliações conduzidas no transcorrer deste trabalho só foram possíveis exatamente pela

disponibilidade de informações e a facilidade de identificação visual do que estava ocorrendo em determinados momentos durante o treinamento.

1.7 PONTOS ORIGINAIS DO TRABALHO

Dentre os pontos originais do trabalho destacam-se:

- a) Aplicação da análise de sensibilidade como guia para o algoritmo genético;
- b) Obtenção da análise de sensibilidade através da variação dos pesos;
- c) Maior sensibilidade dos pesos da camada de saída em uma rede 2-2-1 para o domínio da aplicação do XOR;
- d) Nova abordagem da inspiração biológica na aplicação do algoritmo genético;
- e) Abordagem unimodal do algoritmo genético no treinamento de redes neurais artificiais.

1.8 ESTRUTURA DO TRABALHO

A seguir, no capítulo 2, será apresentado uma rápida revisão das técnicas envolvidas, RNA e AG, sendo dado maior ênfase ao AG por este ser mais recente e não ser tão consolidado quanto a RNA. Também apresenta-se o estado da arte no treinamento híbrido de RNA através de AGs e o algoritmo de treinamento baseado em GD, bem como a fundamentação teórica da análise de sensibilidade (AS). O capítulo 3 abordará a proposta de trabalho da tese. No capítulo 4 apresentam-se os resultados do processo observacional e empírico realizado no transcorrer do presente trabalho. As conclusões são apresentados no capítulo 5. As Referências Bibliográficas são apresentadas no capítulo 6.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas as principais técnicas a serem utilizadas no transcorrer deste trabalho. Apesar de aparentemente independentes, aos poucos o papel de cada uma das técnicas e suas interligações serão compreendidos.

Apesar da vasta literatura existente sobre RNA, apresenta-se um mínimo de exposição sobre o tema, tendo em vista a inexistência de uma padronização da terminologia. Já no caso do AG, por ser este uma técnica mais recente e apresentar ainda muitas controvérsias e/ou pontos dúbios, será dada maior ênfase.

Após a apresentação da RNA e do AG, mostra-se o estado da arte no treinamento híbrido de RNA através do algoritmo de RP e AG. Neste ponto, além de mostrar a metodologia convencional, também são apresentadas as críticas a esta metodologia. Apresenta-se a análise de sensibilidade como uma das possíveis ferramentas a serem empregadas na avaliação da veracidade destas críticas bem como na atenuação dos problemas da técnica híbrida convencional.

No transcorrer da apresentação são usados os termos minimização e maximização com certa frequência. O termo maximização está relacionado aos objetivos do AG enquanto que a minimização do erro refere-se ao objetivo final do algoritmo de treinamento de uma RNA. Portanto, para que seja possível empregar o AG como algoritmo de treinamento de uma RNA, torna-se necessária adequação da função custo do AG e da RNA. Assim, a função custo a ser maximizada pelo AG será proporcional ao inverso da função custo da RNA a qual deverá necessariamente ser minimizada. Desta forma, as particularidades de ambas as técnicas foram consideradas.

Ao final deste capítulo, ter-se-á visto as técnicas envolvidas no treinamento híbrido de RNAs, os problemas inerentes a esta e uma proposta de ferramentas a serem utilizadas na avaliação e atenuação dos problemas comumente encontrados na técnica convencional.

2.1 REDES NEURAIS

Esta revisão faz-se necessária tão somente pela ausência de uma normatização da terminologia na área de inteligência artificial (IA). Portanto, será restrita a presente abordagem aos elementos essenciais aos capítulos subseqüentes.

Em caso de necessidade de aprofundar-se em algum assunto relatado neste capítulo sugere-se o livro de Simon Haykin [17] e Barreto [31].

2.1.1 Introdução

A motivação inicial para buscar nos neurônios e no sistema nervoso central (SNC) alguma inspiração para a resolução de problemas, deve-se à elevada capacidade do cérebro humano em realizar certas atividades corriqueiras, como a classificação de padrões. Desta inspiração nasceram as redes neurais artificiais (RNAs) das quais inicialmente esperava-se que emergisse algum comportamento inteligente. Esperava-se também que as RNAs apresentassem características semelhantes às redes neurais naturais (RNN), dos seres vivos, por exemplo, apresentando bom desempenho em tarefas não algorítmicas e tendo dificuldades em realizar tarefas de natureza intrinsecamente algorítmica [2,3].

A principal característica da RNA é sua capacidade de generalização, ou seja, uma vez tendo aprendido um conceito, a RNA é capaz de operar com conceitos similares que não foram aprendidos, sem esforço suplementar [2].

A viabilidade do emprego da RNA nas mais diversas áreas fez dela uma das principais áreas da IA, denominada de inteligência artificial conexionista (IAC). A IAC apresenta excelente desempenho com problemas mal definidos ou seja, onde o modelo não está disponível ou disponível

porém com imprecisão. A atuação da IAC nestes casos torna-se possível devido a sua capacidade de “adquirir o modelo” a partir de exemplos do problema em estudo.

No início dos anos 40, os pioneiros da área, McCulloch e Pitts [32], estudavam o potencial e a capacidade emergente da interconexão de vários componentes baseados no modelo do neurônio e células nervosas naturais. Com base nestes estudos, os pesquisadores propuseram o primeiro modelo de neurônio artificial (McCulloch and Pitts - 1943). Outros como Hebb (1949), preocuparam-se com as leis de ajuste dos pesos envolvidos em sistemas neurais. A introdução dos perceptrons por Rosenblatt no final dos anos 50 criou grande agitação, principalmente sobre como os sistemas inteligentes poderiam ser construídos a partir dos perceptrons. Rosenblatt (1962), através do teorema da convergência dos perceptrons, garante que o perceptron encontrará um estado de soluções se este existir, isto é, ele aprenderá a classificar qualquer conjunto de entradas desde que para este problema exista uma solução [33]. Contudo, no livro *Perceptrons* (1969), Minsky and Papert [25] colocaram um fim nessas especulações, através de uma análise rigorosa do perceptron, provando muitas propriedades e apontando limitações de muitos modelos. Observaram que, embora o teorema da convergência garantisse a classificação correta, estes dados deveriam obrigatoriamente ser linearmente separáveis. Infelizmente a maioria dos problemas encontrados não são linearmente separáveis. Logo, o perceptron inicialmente proposto não é capaz de processar problemas linearmente não separáveis, como é o caso do famoso exemplo do ou-exclusivo ou XOR como também é conhecido [17].

Inspirado na regra delta, introduzida por Widrow et al [1], em 1974 Paul Werbos apresentou a base conceitual do algoritmo de treinamento de retropropagação (RP), (em inglês “back-propagation”), que independentemente foi reinventado por David Parker em 1982 e Rumelhart and McClelland em 1986 (no livro “*Parallel Distributed Processing*”). Devido a inspiração na regra delta, o algoritmo de RP também é conhecido como regra delta generalizada [34].

Com o advento da rede multicamada do tipo direta (RMD), bem como do algoritmo de retropropagação (RP), a IAC teve um grande impulso, imergindo nos anos 80 do marasmo em que se encontrava nos anos 70, devido as afirmações de Minsky and Papert. Devido a estas afirmações, a grande maioria das pesquisas na área da IAC tinham sido praticamente canceladas. Com a falta de investimentos a paralisia das pesquisas foi inevitável, apenas quebrada algumas vezes por trabalhos de alguns poucos pesquisadores, tais como os do próprio Widrow, Werbos, Grosberg, Kohonen, dentre outros.

Conforme já descrito anteriormente, este trabalho mescla as técnicas do algoritmo genético e RNA. Dos vários tipos de RNA o trabalho focalizará as redes multicamada do tipo direta ("feed-forward"), o algoritmo de treinamento mais difundido, o de retropropagação e o algoritmo Quickprop.

As RMDs, devido à sua versatilidade, apresentam uma vasta gama de aplicações. De uma forma geral, podem ser empregadas na classificação de padrões, generalização, memorização ("look-up table"), aproximador universal, etc. A habilidade para este desempenho depende fundamentalmente da utilização de uma função de ativação adequada, do número de camadas de neurônios e do conjunto de treinamento da RNA.

Devido à versatilidade e a relativa facilidade de uso, a RMD é até os dias de hoje a mais popular das redes neurais artificiais. Atualmente, é comum a aplicação de RMD em classificação, controle de processos, processamento de sinais, diagnóstico médico e previsões financeiras.

Este trabalho focaliza somente a RMD de 3 camadas sendo totalmente conectada (pesos). Esta escolha deve-se ao domínio da aplicação especificado no capítulo anterior, o XOR, a ser implementado com a RMD 2-2-1.

Após a exposição do modelo do neurônio adotado, apresentar-se-á alguns detalhes da rede multicamada direta bem como as fórmulas essenciais do algoritmo de RP.

2.1.2 Modelo Generalizado do Neurônio

Apesar da diferença entre as RNNs e as RNAs, estas foram inspiradas no cérebro humano. Apesar da simplicidade do primeiro neurônio artificial, proposto em 1943 por McCulloch and Pitts, este serviu de modelo para estudos do sistema nervoso e para estimular a área de inteligência artificial conexionista (IAC). Atualmente, há uma grande variedade de modelos que em sua larga maioria estão deixando de ter como princípio básico a modelagem do SNC e sim o desejo de construir neurocomputadores. Assim sendo, adotar-se-á como modelo geral do neurônio, o descrito por Barreto [2], conforme mostra a figura 2.1. Este modelo é uma generalização do modelo proposto por McCulloch and Pitts [2,32].

O neurônio artificial, elemento básico das RNAs, é uma unidade de processamento de informações [17]. Basicamente, consiste de uma unidade de soma (no presente caso), denominada no modelo de Φ e da função de ativação, denominada de η .

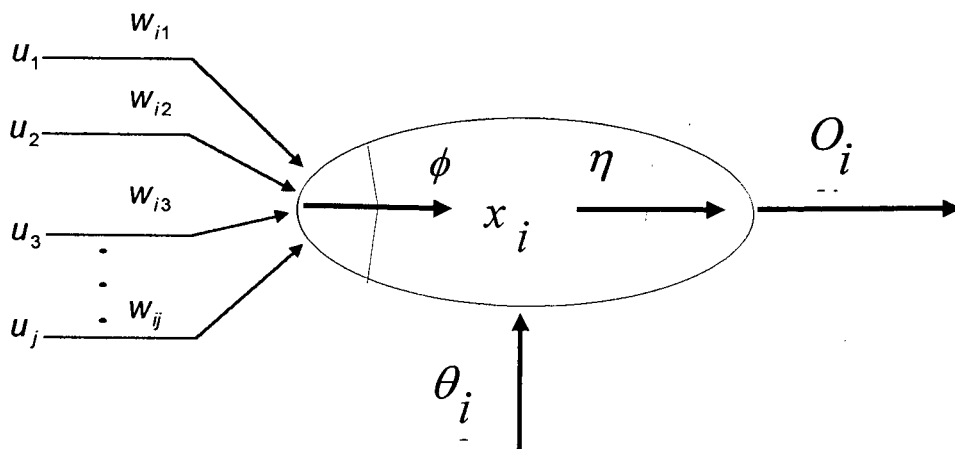


Figura 2. 1 Modelo do neurônio artificial [31]

A função Φ combina as entradas u_j ponderadas por W_j constituindo uma combinação linear. O estado de ativação do neurônio produzido pela função Φ através da função η , produzirá a saída do neurônio. A função η ,

também denominada de função de ativação, limitará o sinal de saída do neurônio em algum valor finito. Tipicamente, este valor é normalizado no intervalo $[0,1]$ no caso da função de ativação do tipo sigmóide e no intervalo $[-1, 1]$ no caso da utilização da função de ativação tangente hiperbólica.

Em termos matemáticos para um neurônio estático⁴, x_i , temos:

$$x_i = \Phi(w_i, u_i) = \sum w_{ij} \cdot o_j + \Theta_i \quad (1)$$

e a saída O_i do neurônio dada por:

$$O_i = \eta(x_i) \quad (2)$$

2.1.3 Redes Multicamada do Tipo Direta (RMD)

O presente trabalho, como já mencionado, utilizará as RNAs multicamada do tipo direta ("feed-forward"), as quais são caracterizadas por um conjunto de neurônios arranjados em camadas, como mostra a figura 2.2.

Nestas, todos os sinais se propagam no sentido direto, da camada de entrada para a de saída passando pela camada intermediária. Não existe realimentação.

Os neurônios em uma rede multicamada do tipo direta podem ser de três tipos, conforme a camada em que se localizam: camada de neurônios de entrada; camada de neurônios internos e camada de neurônios de saída.

Os neurônios da camada de entrada, como o próprio nome sugere, são os responsáveis por transmitir o vetor de dados da entrada para o interior da rede. Isso é realizado sem que nenhum processamento propriamente dito seja realizado, ou seja, pode-se fazer uma analogia destes neurônios com condutores passivos, os quais propagam uma informação sem alterá-la.

⁴ O neurônio pode ser estático ou dinâmico[31]. Neste trabalho será empregado somente o neurônio estático.

A camada de saída é encarregada de gerar a saída da RMD. Cada neurônio desta camada é conectado aos neurônios da camada interna mais próxima (no caso de haver mais de uma camada interna). Estes, geralmente, tem a mesma capacidade e a forma de processamento dos neurônios da(s) camada(s) interna(s).

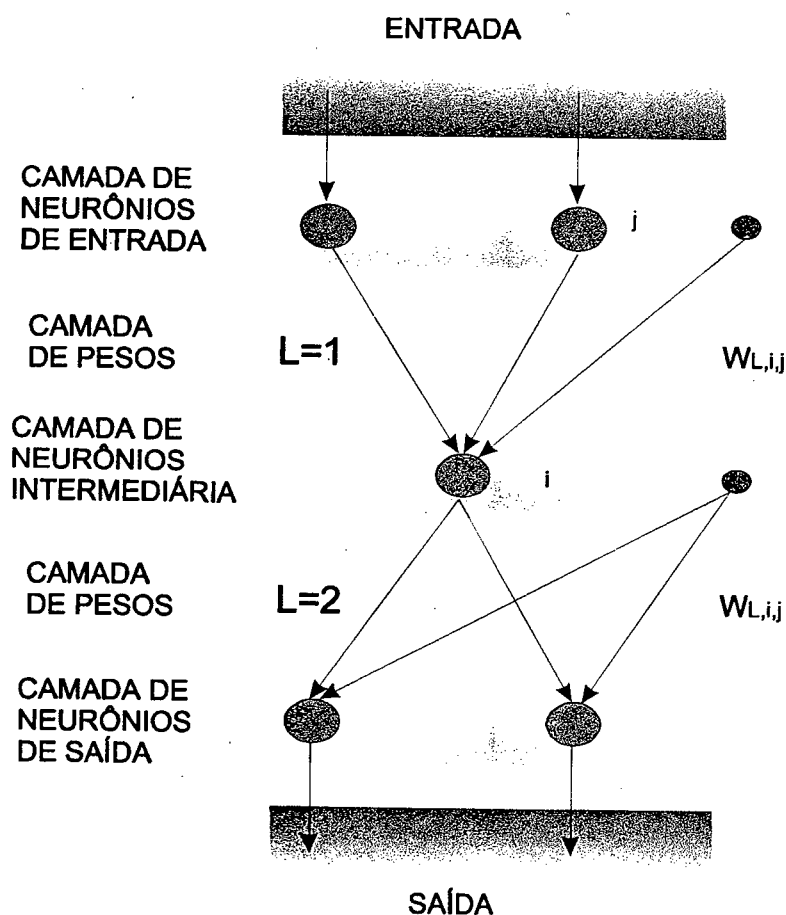


Figura 2.2 Exemplo de rede multicamada do tipo direta com 3 camadas de neurônios e 2 camadas de pesos.

Os neurônios da(s) camada(s) interna(s) são invisíveis para o ambiente externo, localizando-se entre a camada de entrada e a de saída. Estes neurônios são fundamentais para permitir que a RMD possa extrair características e generalizações [35].

A RMD (também denominada de perceptron multicamada) aprende por exemplos, através do processamento repetido de um conjunto de

treinamento que contém uma série de vetores de entrada e a correspondente saída. Cada passagem do conjunto de treinamento pela rede constitui uma época. Durante cada época, a rede compara na saída, o resultado desejado com o resultado atual, calcula o erro, e modifica os pesos da rede visando minimizar o erro. Através deste processo, denominado de treinamento supervisionado, a rede aprende a associar padrões de entrada com os correspondentes padrões de saída. Ao final, o conjunto dos pesos obtidos representa o que a rede aprendeu.

A seguir será feito uma breve revisão do procedimento de ajuste dos pesos durante o treinamento segundo o algoritmo de RP e o Quickprop. Um equacionamento mais detalhado do algoritmo de RP pode ser encontrado em Kröse e Smargt [18] e do Quickprop em [14,36,37].

2.1.4 Algoritmo de Treinamento

Basicamente, o problema do treinamento é encontrar um conjunto de pesos para a RNA que minimize o erro produzido pela diferença entre a saída obtida na rede e o valor desejado. A estratégia comumente adotada para encontrar um mínimo no espaço de busca definido pelos pesos, usa uma função iterativa na vizinhança do ponto especificado pelo peso no instante da análise. Uma aproximação da função pode ser obtida através da primeira ou da segunda ordem da expansão de Taylor. O próximo ponto é determinado em relação ao corrente, obtendo-se a direção de busca e o tamanho do passo a ser dado nessa direção. Se a direção de busca é fixada como sendo o gradiente negativo e o tamanho do passo é constante, então teremos o algoritmo de RP (o qual utiliza a aproximação de primeira ordem da expansão de Taylor). Se, contudo, for utilizada a segunda ordem da expansão de Taylor, obter-se-á tanto a direção de busca como o tamanho do passo. Provavelmente um passo será suficiente para obter um mínimo no espaço de busca[22].

Entretanto, os métodos de segunda ordem não são aplicáveis na prática devido aos requisitos de cálculo da matriz Hessiana, a qual apresenta grande complexidade e alto custo computacional. Normalmente, para

contornar estes problemas, emprega-se uma aproximação do método de segunda ordem. Assim, dentre as técnicas que são aproximações dos métodos de segunda ordem, cita-se: Quickprop (Fahlman[14]), Second Order Momentum (Pearlmutter[38]), Newton, Quasi- and Pseudo-Newton (Watrous and Shastri [39]), Conjugate Gradient (Johansson [40]).

A aplicação destas técnicas inicialmente necessita de um conjunto de padrões, o qual consiste de pares de vetores de entrada e saída. Esta espécie de treinamento é denominada de treinamento supervisionado, em contraste ao treinamento não-supervisionado, onde somente o vetor de entrada é apresentado à rede durante o treinamento.

Independente do algoritmo de treinamento, o ajuste dos pesos da RNA pode ser realizado após a apresentação de todos os padrões de treinamento (treinamento em lote) ou então, após a apresentação de cada padrão (treinamento por padrão).

A seguir, apresenta-se uma revisão do algoritmo de RP e do Quickprop. O algoritmo de RP é apresentado por ser o mais comumente empregado no treinamento híbrido de RNA em conjunto com o AG. Já o algoritmo Quickprop é mostrado como uma alternativa ao algoritmo de RP, não somente por causa de seu desempenho superior ao algoritmo de RP quanto à velocidade de convergência, mas principalmente pela capacidade de processar valores de pesos mais elevados. Como será visto neste trabalho, esta última característica é essencial quando trabalhando de forma híbrida com AG.

a) Algoritmo de Retropropagação

O algoritmo de RP é um algoritmo de primeira ordem, ou seja, baseia-se no gradiente descendente para obter a direção de busca do mínimo da superfície de erro e o passo empregado é dado pela taxa de treinamento. No caso do treinamento por padrão, para cada padrão de treinamento (p) as conexões ou pesos (W_{ij}) (que interligam o neurônio i ao seu antecessor j) são modificadas por uma pequena quantidade (ΔW_{ij}), na direção da derivada

negativa do erro quadrático com respeito ao correspondente peso. Onde o erro quadrático pode ser escrito como:

$$E_p = \frac{1}{2} \sum_{i=1}^{N_0} (d_i^p - o_i^p)^2 \quad (3)$$

Onde d_i^p é o valor desejado no neurônio i para o padrão p e N_0 representa o número de neurônios na camada.

O algoritmo de RP requer que a função de ativação seja contínua e diferenciável, devido a sua utilização na derivação da regra de atualização dos pesos. Supondo que seja utilizada a sigmóide como função de ativação, a saída o_i^p é dada por:

$$O_i^p(x) = \frac{1}{1 + e^{-x_i}} \quad (4)$$

Assim,

$$\Delta w_{ij}(t+1) = -\varepsilon \frac{\partial E_p}{\partial w_{ij}(t)} \quad (5)$$

onde ε é a taxa de treinamento que regula a amplitude de cada passo de renovação dos pesos.

O gradiente descendente presente na equação 5 pode ser reescrito (equação 6) como sendo função do erro na saída do neurônio i (δ_i^p) produzido pela saída do neurônio j (O_j^p) propagado pela conexão comum a ambos, W_{ij} .

$$\Delta w_{ij} = -\varepsilon \frac{\partial E_p}{\partial w_{ij}} = \varepsilon \cdot \delta_i^p \cdot o_j^p \quad (6)$$

A determinação do erro δ_i^p depende da camada onde se localiza o neurônio i . Para um neurônio localizado na camada de saída, o erro δ_i^p é dado por:

$$\delta_i^p = (d_i^p - o_i^p) o_i^p (1 - o_i^p) \quad (7)$$

Já se o neurônio estiver localizado nas camadas intermediárias, o erro δ_j^p é dado por:

$$\delta_j^p = o_j^p (1 - o_j^p) \sum_{i=1}^{N_o} \delta_i^p W_{ij} \quad (8)$$

As equações 7 e 8 fornecem um procedimento computacional recursivo para calcular o erro, δ , para todas as unidades da rede. Isto permite calcular a renovação dos pesos, ΔW_{ij} , conforme a equação 5.

O algoritmo de RP requer que as mudanças dos pesos sejam de forma infinitesimal. Esta correção é tão pequena quanto se queira devido a inclusão da taxa de treinamento (ε). Como valores muito pequenos de ε são irrealizáveis e valores grandes podem causar oscilações, acrescenta-se o termo denominado momento (α), o qual tem por objetivo atenuar estas possíveis oscilações. Quando a correção atual tem direção oposta (sinal) à correção anterior, o momento atenua a correção atual. Já no caso da correção atual possuir a mesma direção, ao invés de atenuar, aumenta o passo. Com isso, a inclusão do termo momento faz com que as mudanças dos pesos tendam a manter a mesma direção. Desta forma, permite-se que alguns mínimos locais pequenos sejam ignorados [33], bem como é atenuada a influência de planaltos onde o gradiente é muito pequeno [41]. Com a inclusão do momento a correção dos pesos segue a forma dada por:

$$\Delta w_{ij}(t+1) = \varepsilon \cdot \delta_i \cdot o_j + \alpha \cdot \Delta w_{ij}(t) \quad (9)$$

onde t indexa temporalmente o ajuste do peso.

b) Algoritmo Quickprop

O algoritmo Quickprop (QP), proposto por Fahlman (1988)[36], é uma aproximação do método de segunda ordem. O próprio Fahlman ao comentar a inspiração do Quickprop, afirma que o algoritmo é baseado no método de Newton e que em essência, é uma técnica fundamentada mais em heurística do que no formalismo matemático [14]. Este algoritmo é a variante mais rápida do algoritmo de RP; vários pesquisadores têm mostrado o seu desempenho em relação à velocidade de convergência [13,14,22,36].

A renovação dos pesos $\Delta W(t+1)$ pelo QP, equação 10, aparentemente é a mesma do algoritmo de RP com o acréscimo de um segundo termo. Este termo é uma aproximação da derivada de segunda ordem do erro com respeito ao peso, multiplicada pela renovação dos pesos no instante $t-1$.

$$\Delta W(t+1) = -\varepsilon \cdot \frac{\partial E}{\partial W}(t) + \frac{\frac{\partial E}{\partial W}(t)}{\frac{\partial E}{\partial W}(t-1) - \frac{\partial E}{\partial W}(t)} * \Delta W(t-1) \quad (10)$$

Além das alterações sob a equação de renovação dos pesos, a aplicação do algoritmo QP consiste também na aplicação de várias heurísticas.

Fahlman introduziu um “fator máximo de crescimento”, μ , o qual é utilizado como limite de renovação dos pesos, evitando oscilações. Assim, quando a renovação de um determinado peso for superior a μ , esta é ignorada e a renovação assume o valor como sendo igual a μ . Este fator também é usado como um indicador de presença de mínimo local, o qual deve causar um rápido crescimento para os pesos sem um treinamento adicional. O valor sugerido por Fahlman para μ , dependendo do problema e da taxa de treinamento, pode variar de 1,75 a 2,25 [22,37].

Outra heurística importante refere-se ao gradiente da sigmóide, o qual foi alterado devido a possibilidade de paralisia quando a saída o_i aproxima-

se de 1,0 ou 0,0, conforme é apresentado no próximo item. Desta forma, Fahlman acrescentou um “offset” ao gradiente da sigmóide (equação 11).

$$\frac{\partial E}{\partial W} \propto o_i(1-o_i)+0,1 \quad (11)$$

Uma nova função de erro é utilizada. As unidades onde o valor absoluto do erro torna-se menor do que 0,1 (ou erro quadrático menor do que 0,01) não serão mais treinadas. Esta modificação evita o sobre-treinamento das unidades da rede.

Outra heurística consiste em aplicar uma pequena redução no peso para prevenir o crescimento elevado do valor dos pesos. O equacionamento completo do algoritmo Quickprop podem ser encontrados em [22,37].

2.1.5 Problemas no Treinamento com o Algoritmo de Retropropagação

Rumelhart, Hinton, and Williams [12] providenciaram uma prova da convergência do algoritmo de RP. Esta foi realizada em termos de equações diferenciais parciais, fazendo com que seja válida somente se os pesos da rede são ajustados em passos infinitesimais, o que tornaria a sua aplicação prática proibitiva. De fato, não existe prova de que o algoritmo de RP convergirá com um passo de tamanho finito. Observações empíricas mostram que a rede usualmente aprende, mas a duração do processo de treinamento é imprevisível e geralmente longo [42].

O algoritmo de RP tem a vantagem de efetuar uma busca direta, ou seja, os pesos são sempre ajustados na direção que minimiza a superfície de erros. No entanto, alguns problemas podem dificultar o seu desempenho. Adiante são descritos os principais fatores que podem comprometer o desempenho não somente do algoritmo de RP mas de todos os algoritmos de treinamento baseados na técnica de gradiente descendente.

a) Mínimo Local

O algoritmo de RP utiliza o gradiente descendente para ajustar os pesos da rede, seguindo o declive da superfície de erro para um mínimo. Desta forma

consegue trabalhar bem com superfícies de erro convexas, as quais têm um único mínimo. Mas freqüentemente conduz a uma solução não-ótima em uma superfície de erro altamente convoluída ou não convexa, normalmente encontrada na maioria das aplicações. Em alguns casos um mínimo local é uma solução aceitável; contudo na maioria é inadequada.

Após a finalização do treinamento através do algoritmo de RP, a apresentação do conjunto de teste permite avaliar se a solução encontrada é aceitável ou não. Caso não seja aceitável, deve-se inicializar os pesos da rede com um novo conjunto de pesos aleatórios e treiná-la novamente. Entretanto, não há garantias de que o novo treinamento seja bem sucedido até o momento da avaliação com o conjunto de teste.

b) Paralisia

Em algumas circunstâncias, a rede pode dirigir-se para um estado no qual a modificação dos pesos acarreta paralisia. Esta paralisia do treinamento da rede é um problema sério; uma vez ocorrido, o tempo de treinamento pode se estender por várias ordens de magnitude [42].

No algoritmo de RP, durante a fase de retropropagação dos erros, o gradiente da função erro (E) é proporcional a $o_i(1-o_i)$, no caso da aplicação da sigmóide, onde o_i é o valor de saída atual para o neurônio i . Contudo, quando o_i está próximo de 1,0 ou 0,0, o termo $o_i(1-o_i)$ torna-se muito pequeno, reduzindo a velocidade de descida do gradiente. Isto causa uma saída próxima dos extremos da função de ativação $\eta(x)$; neste ponto, a sua derivada aproxima-se de zero. Como o algoritmo de RP calcula a magnitude da mudança dos pesos usando esta derivada, é de se esperar que a correção do peso também seja nula ou próxima de zero. Se esta condição for difundida pela rede, o treinamento pode tornar-se lento ou até mesmo paralisar-se.

Outra razão para a paralisia é a presença de "planalto" [21]. Seja por exemplo, a busca do mínimo da função mostrada na figura 2.3, o gradiente é

nulo para todo um conjunto de valores de W_{ij} , o algoritmo não avança, fica paralisado.

A paralisia ocorre também quando há saturação da função de ativação dos neurônios, o que é um fato mais freqüente do que o mencionado acima.

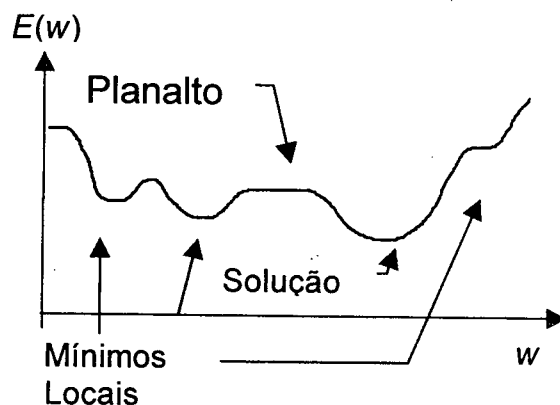


Figura 2.3 Paralisia devido a presença de planalto.

2.2 COMPUTAÇÃO EVOLUTIVA

A busca por métodos de solução de problemas surgiu logo após o aparecimento da vida, e certamente o primeiro problema a resolver foi o da sobrevivência. Deste momento em diante a busca por soluções passou a crescer. Porém, a complexidade dos problemas cresceu mais rapidamente ainda.

No final da década de 50, alguns pesquisadores buscaram na natureza inspiração para novas técnicas de busca de soluções. O motivo para a atenção ter se voltado para a natureza deve-se ao fato desta conseguir resolver de forma satisfatória problemas altamente complexos, como é o caso da sobrevivência das espécies. Aliado a este fato, é possível explicar a grande maioria dos seres vivos através de poucos operadores que podem ser modelados utilizando-se processos estocásticos (cruzamento, mutação e seleção), agindo sobre uma população de uma espécie [43].

A tentativa de imitação do cérebro humano na expectativa de um comportamento emergente deu origem às RNAs. Já a tentativa de imitar a

evolução dos seres vivos na natureza originou a computação evolutiva (CE). A CE é uma designação genérica, dada aos métodos computacionais com inspiração na teoria de evolução das espécies e na genética natural. Dentre os métodos computacionais, também denominados de algoritmos evolutivos (AE), pode-se citar os três principais⁵: programação evolutiva (PE), estratégia evolutiva (EE) e algoritmo genético (AG) [43,44]. A seguir serão apresentados cada um destes métodos com ênfase especial ao AG, tendo em vista este ser a principal ferramenta a ser estudada neste trabalho, junto com uma técnica baseada em gradiente descendente para efetuar o treinamento híbrido de RMD.

A PE concebida por Lawrence J. Fogel nos EUA no início da década de 60 foi difundida a partir de 1966 com o livro "Artificial Intelligence Through Simulated Evolution" escrito por Fogel, Owens and Walsh [45]. Em poucas palavras poderia se definir esta técnica como sendo caracterizada pela presença geralmente única do operador genético da mutação.

A EE foi concebida independentemente por Ingo Rechenberg e Hans-Paul Schwefel em 1965 na Alemanha. Ambos trabalhavam juntos no início dos anos sessenta, visando a otimização do projeto de bico de aviões no tocante ao coeficiente aerodinâmico de penetração. A EE trabalha da mesma forma que a PE, porém, diverge na forma de implementação do operador de mutação. A EE, em vez da mutação, adiciona um ruído gaussiano com média zero e com um certo desvio padrão ao descendente [43].

O AG foi proposto inicialmente por John H. Holland em 1975 no trabalho intitulado "Adaptation in Natural and Artificial Systems"[46]. Holland inspirou-se no mecanismo de evolução das espécies, tendo como base os trabalhos de Darwin sobre a origem das espécies [47] e na genética natural devido principalmente a Mendel [48]. Atualmente, há também uma variação desta, a programação genética (PG), onde além da presença dos operadores de seleção e mutação há também a do cruzamento.

⁵ A Programação Genética (PG) é considerada aqui como sendo parte do AG.

O Algoritmo Genético (AG) é um método robusto, utilizado basicamente para resolver problemas em pesquisa numérica, otimização de funções e treinamento de máquina, dentre outras áreas [43,46,49-51].

Os maiores atrativos do AG tem sido a sua simplicidade de implementação e a eficiência [51], sendo que grande parte desta deve-se a eficácia do AG em dar um passo na direção de uma busca global [52].

Fogel [43] apresenta uma classificação muito elucidativa das pesquisas de AG em seis áreas, conforme o interesse dos pesquisadores:

- a) Desenvolvimento de fundamentação matemática para aplicação do AG como técnica de otimização;
- b) Comparação do AG com outros métodos de otimização visando incorporações que melhorem seu desempenho;
- c) Aplicação em problemas de engenharia e desenvolvimento de programas computacionais, também denominada de Programação Genética (PG);
- d) Aplicação de AG em sistemas classificadores baseados em regras;
- e) Utilização de AG em simulação de vida artificial; e
- f) Implementação do AG em máquinas paralelas.

A breve introdução histórica das áreas pertinentes à CE acima não apresenta os inúmeros trabalhos anteriores e que de certa forma formaram o alicerce que mais tarde permitiu aos demais construir as teorias hoje disponíveis. Uma visão histórica mais detalhada da CE pode ser encontrada em Fogel [43].

Tendo localizado o AG no contexto da CE, apresenta-se a seguir uma revisão dos principais fatos históricos que marcaram o desenvolvimento das teorias biológicas da evolução e aprendizado aceitas atualmente. Assim, espera-se que ao final tenha-se uma fonte de inspiração para a área de CE, bem como uma forma de atenuar os equívocos e erros de nomenclatura. Apesar da inspiração biológica não ser integralmente respeitada na CE, é mister que no mínimo a nomenclatura original seja considerada.

2.2.1 Inspiração Biológica para Evolução e Aprendizado

A revisão inicia-se com o zoólogo francês Jean-Baptiste de Lamarck (1744–1829) ao qual atribuí-se a primeira teoria evolucionista tratada cientificamente. Lamarck, já em 1801 começou a divulgar alguns trabalhos sobre a sua teoria de evolução, mas foi no livro “Philosophie Zoologique” de 1809 que este apresentou de forma mais completa a sua teoria. Na verdade, a idéia da transmissão de caracteres adquiridos durante a existência do indivíduo já era admitida desde a antiguidade. Coube a Lamarck uma abordagem mais científica, onde este procurava mostrar que a evolução realmente se dava através da hereditariedade dos caracteres adquiridos porém, de forma direta. Assim, o indivíduo incorporava as características ao seu patrimônio genético e desta forma poderia transmiti-la a seus descendentes. Esta teoria ficou praticamente obscura até que o inglês Charles R. Darwin (1809–1882), apresentasse em 1859 uma nova abordagem para a evolução, no trabalho intitulado “Sobre a origem das espécies”[47]. Este trabalho baseia-se nas observações realizadas de 1831 a 1836 quando de sua viagem, principalmente às ilhas Galápagos a bordo do navio Beagle. Nesta época a teoria de Lamarck retornou com mais força do que quando apresentada inicialmente. Isto se deu principalmente devido à disputas religiosas entre gnósticos e agnósticos, bem como intrigas domésticas entre ingleses e franceses. Como Lamarck, Darwin partiu da observação de uma adaptação e de uma adequação entre a forma e o desempenho dos órgãos dos seres vivos e seu modo de vida. A diferença é que Darwin voltou-se para um mecanismo de evolução baseado na seleção natural enquanto que a proposta Lamarckiana defendia a evolução pelos mecanismos da hereditariedade dos caracteres diretamente adquiridos durante a vida, também denominada de lei do uso e desuso [48,53].

Para a seleção natural pouco importa saber qual o mecanismo que copia o modelo dos pais e transmite-o a seus descendentes, ou seja: se trata de um esforço individual ou de variação inata ao acaso. O essencial é que

certos indivíduos realizam melhor essa tarefa que outros e, como consequência, deixam mais descendentes que estes.

No entanto, descontando as inúmeras controvérsias de cunho político, filosófico e religioso, a teoria da evolução das espécies como inicialmente proposta apresentava-se frágil em alguns pontos. Principalmente porque na época a teoria da hereditariedade aceita era a da combinação, a qual estipulava que o cruzamento de dois indivíduos diferentes para uma dada característica, originava descendentes exatamente intermediários (semelhante a mistura de líquidos – café com leite por exemplo). Se esta teoria estivesse certa e um dos ascendentes estivesse próximo de um ótimo, os descendentes não poderiam ser melhores do que os que lhe deram origem[53].

No século XX é que as pesquisas forneceram os dados que faltavam à seleção natural de Darwin, abolindo a teoria da hereditariedade por combinação e adotando a teoria da hereditariedade baseada em partículas elementares chamada genes. Estas partículas não se combinam como líquidos, mantendo-se bem distintos durante a transmissão hereditária. É o que prega as leis da hereditariedade também denominadas de Leis de Mendel. Estas leis foram efetivamente estabelecidas em 1865 pelo monge morávio Gregor Mendel (1822–1884) mas ficaram ignoradas até a sua redescoberta em 1900, pelo biólogo holandês Hugo De Vries e dois outros biólogos, Karl Erich Correns na Alemanha e Gustav Tschermak von Seysenegg na Áustria[48].

As Leis de Mendel e a descoberta da dominância, também por Mendel [48,53], resolveram o problema dos descendentes não poderem ser melhores do que o melhor indivíduo que lhes deram origem. Através de suas leis, Mendel defendeu que os descendentes poderiam no conjunto serem melhores que seus pais, desde que herdassem as características favoráveis de cada um, ou seja: os descendentes poderiam ser tão bons quanto fosse o patrimônio genético acumulado pelos ascendentes.

No entanto, a seleção natural e as Leis de Mendel ao longo do tempo tenderiam a saturar o processo de evolução. Assim, a tendência seria de que

todos os indivíduos se aproximassem do melhor indivíduo permitido pelo patrimônio genético da população. No início esta aproximação seria rápida e depois de algumas gerações, tenderia a reduzir a aproximação. Este mecanismo pode ser matematicamente explicado pelo teorema fundamental do algoritmo genético (AG) [46], excluindo a mutação que só veio a ser conhecida mais tarde. As Leis de Mendel explicam a troca de material genético (cruzamento), e a seleção natural à busca pelo indivíduo mais apto. Mas se não houvesse a inclusão de material genético inexistente no patrimônio genético inicial haveria uma estagnação ao longo do tempo.

Apesar do relativo sucesso em explicar a evolução dos indivíduos, as teorias de Mendel e Darwin ainda não eram suficientes para explicar a evolução dos seres vivos quando interagindo com o meio no qual encontram-se inseridos. Darwin já tinha observado que isso acontecia, chegando a admitir a hereditariedade dos caracteres diretamente adquiridos, como proposto por Lamarck, porém, numa escala reduzida.

Somente em 1896 é que James M. Baldwin apresentou o trabalho "A new factor in evolution" [54]. Neste trabalho Baldwin, à semelhança de Lamarck, acreditava que o aprendizado adquirido através das iterações dos indivíduos com o meio poderia ser transmitido às gerações futuras. A divergência das duas abordagens se deve porque Lamarck acreditava que as características adquiridas poderiam ser diretamente transferidas ao patrimônio genético. Já Baldwin pregava que estas características poderiam ser herdadas, porém de forma indireta. Isto pela conscientização de que a capacidade em aprender e do que se aprende depende fortemente do meio. Assim, segundo Baldwin, as iterações com o meio fazem com que os indivíduos que tenham maior habilidade em aprender a sobreviver neste meio, tenham o seu grau de adaptação melhorado. Desta forma, o indivíduo também terá uma probabilidade maior de sobrevivência e de reprodução. Consequentemente se houvesse reprodução viável, o patrimônio genético dos mais hábeis estaria sendo preservado para as gerações futuras.

Por outro lado, pelo fato das características adquiridas serem diretamente herdadas, a proposta de Lamarck requer o mapeamento inverso,

do fenótipo para o genótipo, o que não é plausível biologicamente e tão pouco na maioria das aplicações práticas (ex: problema do caixeiro viajante).

Já a proposta de Baldwin é Darwiniana, pois não envolve o mapeamento inverso. As características adquiridas são herdadas indiretamente [55].

Outra descoberta importante e que, aliada às Leis de Mendel, deu maior sustentação à teoria de Darwin, foi a mutação. O botânico holandês Hugo Marie De Vries (1848-1935) além de redescobrir as Leis de Mendel, foi o primeiro a utilizar o termo mutação. De Vries, em 1903, publicou o livro "The mutation theory", descrevendo seus estudos sobre o fenômeno da variação e mutação das plantas, onde esperava que amplas variações poderiam produzir uma espécie nova em uma única geração, discordando desta forma de Darwin, que enfatiza o desenvolvimento lento de novas espécies por diferenças individuais quase imperceptíveis.

Outros estudos sobre a mutação foram conduzidos no laboratório do biólogo americano Thomas H. Morgan, na Universidade Columbia em Nova York. Em 1910, durante estudos sobre a transmissão hereditária na mosca drosófila (mosca da banana), notou-se que sem causa aparente, surgiu inesperadamente uma nova característica que era herdável segundo as Leis de Mendel. A questão, daí por diante, era saber qual a frequência dessas mutações. Por volta do final dos anos 30, essas pesquisas conduziram à noção de que, entre organismos multicelulares (animais e vegetais), a taxa de mutação por gene é de mais ou menos 1/100.000 por célula sexual. Na espécie humana cada ser humano é portador, em média, de duas mutações. Estas mutações porém, na maior parte das vezes, não se manifestam, pois os genes mutados são geralmente recessivos[48,53].

A reunião destas teorias em torno da denominação de "neodarwinismo" ou também de "Teoria Sintética da Evolução" foi realizada em Princeton (EUA) em uma conferência internacional realizada em janeiro de 1947, onde estavam presentes geneticistas, naturalistas e paleontólogos.

Atualmente, a ambição dos cientistas é o Projeto Genoma que através de um esforço internacional, pretende-se até o ano 2005 mapear um a um, os 100.000 genes contidos em cada célula do corpo humano.

Até o momento, enfocou-se a inspiração biológica da CE com ênfase no AG. Para tanto, a apresentação envolveu a teoria de evolução das espécies e a genética natural. Antes de abordar a metodologia de implementação do AG, principalmente quanto a seus operadores, é necessário estabelecer um vocabulário mínimo comum tanto ao AG como a suas fontes de inspiração biológica.

2.2.2 Conceitos Fundamentais e Terminologia do Algoritmo Genético

A nível biológico, um ser vivo (indivíduo) geralmente é composto por um conjunto de cromossomos. No caso do AG é comum o emprego dos dois termos (indivíduo e cromossomo) indistintamente. Contudo, em uma analogia da biologia com os problemas a serem otimizados pelo AG, esta afirmação só é válida se o problema for uma função de uma única variável. Caso o problema seja uma função de várias variáveis, todas passíveis de otimização, então o indivíduo é composto de vários cromossomos. Este geralmente é o caso das RNAs.

Na literatura de AG é comum aparecer o termo em inglês "string" como sinônimo de cromossomo e indivíduo. Isso se deve ao fato da representação algorítmica do cromossomo, em AG, geralmente ser implementada pelo alfabeto binário {0, 1}, em analogia com a natureza que utiliza quatro bases: a Adenina representada por A, a Guanina representada por G, a Timina representada por T e por último a Citosina representada por C [43].

O cromossomo é composto de genes, sendo que cada gene possui um local fixo no cromossomo. Este local é denominado de locus. Cada gene pode assumir um certo valor pertencente a um certo conjunto de valores, os quais são denominados de alelo [56]. Em termos de AG, o cromossomo corresponde ao indivíduo, e este é representado por uma "string" de comprimento finito. O termo gene é denominado de "bit". O termo alelo refere-se ao conjunto de valores possíveis de serem atribuídos a um

determinado “bit”, ou seja, é o alfabeto binário {0, 1}, no presente exemplo. Desta forma o valor de um “bit” (alelo) refletido no fenótipo depende da posição que este ocupa na “string” (locus).

Ao conjunto de cromossomo, genes e alelos denomina-se genótipo e as características conferidas por este denomina-se fenótipo. Em termos de AG, o genótipo é a variável independente (x) e o fenótipo é a variável dependente ou função, $f(x)$. Na tabela 1 apresenta-se um resumo da terminologia comum ao AG e à biologia.

No algoritmo genético trabalha-se com um conjunto de indivíduos (população) no qual cada elemento é candidato a ser a solução desejada. A função a ser otimizada representa o ambiente no qual a população inicial vai ser posta. Espera-se que através dos mecanismos de evolução das espécies e da genética natural, os indivíduos mais aptos tenham maior probabilidade de se reproduzirem e que a cada nova geração esteja mais apto ao ambiente (função a ser otimizada).

A aptidão, em inglês “fitness”, é obtida pela avaliação do indivíduo através da função a ser otimizada. Se o objetivo for maximizar, a aptidão é diretamente proporcional ao valor da função. Caso o objetivo seja a minimização, a aptidão será inversamente proporcional ao valor da função. Contudo, o termo minimização não é bem aceito por alguns pesquisadores por não ter inspiração biológica, haja visto que os indivíduos mais aptos é que deverão ter maiores chances de sobreviverem [52,57].

Após ter sido realizado o teste de todos os indivíduos da população na função a ser otimizada, obtém-se a aptidão para cada um. A adaptação de um indivíduo, no contexto biológico, designa o desempenho deste no ambiente em que se encontra inserido [46]. A aptidão é a quantificação da adaptação do indivíduo ou seja é o valor obtido com a aplicação do indivíduo à função custo.

A aptidão é um valor que exprime quão adaptado está o indivíduo ao meio; quanto maior a aptidão, maior é a probabilidade do indivíduo se reproduzir. Assim, a aptidão é obtida através da função $f(x)$, a qual é uma aproximação do meio onde se encontra inserido o indivíduo x que deverá

competir com os demais indivíduos da população. Dependendo do meio e do genótipo do indivíduo x , ao longo do tempo pode haver uma variação na sua adaptação, Δx . A magnitude desta variação é denominada de grau de adaptação do indivíduo. Como exemplo de grau de adaptação, poder-se-ia citar a barata que, hoje, tem uma aptidão baixa, mas em caso de catástrofe nuclear ela sobreviveria. Isso quer dizer que ela possui um alto grau de adaptação, quando considerados estes dois meios.

Tabela 1 Relação da terminologia do AG com a biologia

Biologia	Algoritmo Genético
Cromossomo	Indivíduo ("string")
Gene	Bit
Alelo	Valor do bit
Locus	Posição de um bit específico no indivíduo ou "string"
Genótipo	Indivíduo candidato a solução – x
Fenótipo	Valor da função para um dado indivíduo - $f(x)$

Finalizada a avaliação dos indivíduos da geração atual, espera-se que a próxima geração seja uma evolução da anterior. Para que isso ocorra, os mais aptos deverão possuir maior probabilidade de serem selecionados para dar origem à nova geração. Contudo, alguns poucos não muito aptos também poderão ser selecionados. Ao mecanismo responsável por esta escolha seletiva, denomina-se de seleção. Desta forma, se o processo for bem conduzido, espera-se que a nova geração seja, em média, melhor do que a que lhe deu origem [58].

Realizada a seleção, o próximo passo é a aplicação dos operadores genéticos (OG), também denominados de mecanismos de busca [46]. Dentre tais mecanismos, os mais comumente empregados em AGs são o

cruzamento (“crossover”) e a mutação [49,58,59]. Os OGs destinam-se a manipular o patrimônio genético visando obter um indivíduo mais apto ao ambiente em questão (função custo).

2.2.3 Operadores Genéticos (OGs)

Neste momento, é importante se fazer uma distinção entre operador genético(OG) e operadores do algoritmo genético (OAG). Dentre os OAG mais conhecidos estão a seleção, o cruzamento e a mutação. No entanto destes somente o cruzamento e a mutação são operadores genéticos. A seleção tem inspiração na teoria da evolução das espécies de Charles Darwin (1842) [47] e portanto não pode ser considerada um operador genético. Conforme já descrito antes, o conhecimento a respeito da genética no tocante ao cruzamento foi apresentado 124 anos após a publicação de Darwin [48].

No presente trabalho, o operador denominado de seleção não é considerado um operador genético, mas sim um operador Darwiniano. Nesta pesquisa, a denominação de algoritmo genético refere-se a duas classes de operadores: o operador Darwiniano (seleção) e os operadores genéticos (cruzamento e mutação).

Os operadores genéticos são mecanismos de busca que se destinam à manipulação dos indivíduos selecionados a partir da geração anterior, visando a obtenção de algum indivíduo candidato mais apto. O real papel dos OG e suas características de controle ainda não estão bem definidos. No entanto, é atribuída ao cruzamento a tarefa de explorar o patrimônio genético já existente nos pais, ou seja, este faria uma busca local (por local entenda-se a região que pode ser encontrada pela manipulação do patrimônio genético existente em todos os indivíduos da população). Atribui-se à mutação a função de repor o material genético perdido em gerações anteriores e também a introdução de material inexistente, promovendo uma busca global. Logo, o cruzamento busca uma solução a partir do conhecimento dos indivíduos já existentes “exploitation”, e a mutação

promove uma avaliação em áreas do espaço de busca ainda não avaliadas “exploration” [58].

Realmente, até o momento não existe uma possibilidade de afirmar como e quando usar o cruzamento e a relação deste com o tamanho da população. Contudo, no momento é grande a quantidade de trabalhos estudando as funções de cada um e a relação entre estes parâmetros [56,60-63]. Cada um desses operadores é apresentado a seguir de forma mais detalhada.

a) Cruzamento (“crossover”)

O OG denominado de cruzamento ou também de recombinação (“crossover”), em AG corresponde a uma generalização do que ocorre na reprodução sexuada. Existem na verdade, raros seres unicelulares que apresentam este esquema, antes da mitose. O cruzamento é típico de seres mais evoluídos [61], e se dá pela aproximação dos cromossomos dos dois indivíduos (pais) que trocam entre si partes de seus cromossomos. Isso resulta em dois cromossomos diferentes, porém que ainda guardam influências dos pais. Não é sempre que o cruzamento é efetuado, por isso a taxa de cruzamento representa a probabilidade de parte da população sofrer cruzamento. Os pares (pais) que não sofrerem cruzamento serão copiados como filhos.

A troca de partes do cromossomo pode ser realizada de várias formas. Basicamente tem-se o cruzamento uniforme, cruzamento com 1-partição, cruzamento com 2-partições e cruzamento com n-partições [64-66].

O cruzamento uniforme consiste no emparelhamento dos dois cromossomos pais e cada locus do cromossomo tem 50% de chance de ser trocado⁶. A figura 2.4, mostra um exemplo onde supõem-se que um determinado cromossomo que possui 8 locus, sofreu cruzamento uniforme em 3 locus, o primeiro, o quarto e o quinto locus.

⁶ Há controvérsias em torno do valor de 50%.

O cruzamento com 1-partição, consiste na escolha aleatória de somente um ponto de corte. Todo o material genético, dos pais, existente à direita deste ponto será intercambiado (figura 2.5).

No caso do cruzamento com 2-partições, há a escolha aleatória de dois pontos de corte. Todo o material genético, dos pais, existente entre os dois pontos de corte são trocados, e o restante é mantido inalterado (figura 2.6).

Já o cruzamento com n-partições consiste de n cruzamentos com 2-partições. Pode-se considerar o cruzamento com 1 e 2-partições como sendo casos particulares do cruzamento com n-partições.

Spears[61,64] e outros têm analisado o desempenho das várias formas de aplicação do cruzamento. Contudo, até o momento, o que existe são inúmeras controvérsias sobre qual forma de cruzamento é a melhor, qual a taxa de cruzamento é indicada e a relação destes fatores entre si e com o tamanho da população [67]. Vários autores [56,64,68] concordam que a presença de cruzamento aumenta o desempenho do AG em relação ao uso exclusivo da mutação.

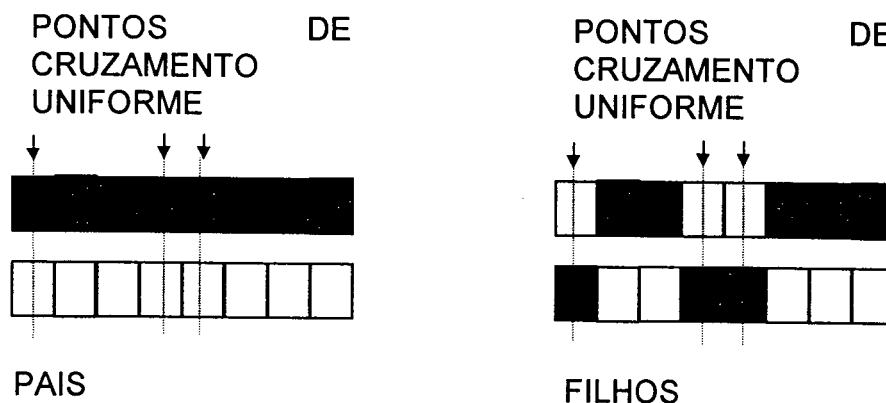


Figura 2.4 Exemplo de cruzamento uniforme.

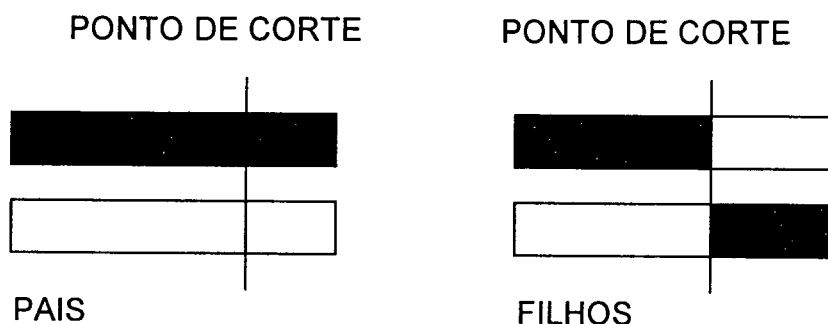


Figura 2.5 Exemplo de cruzamento com 1-partição.

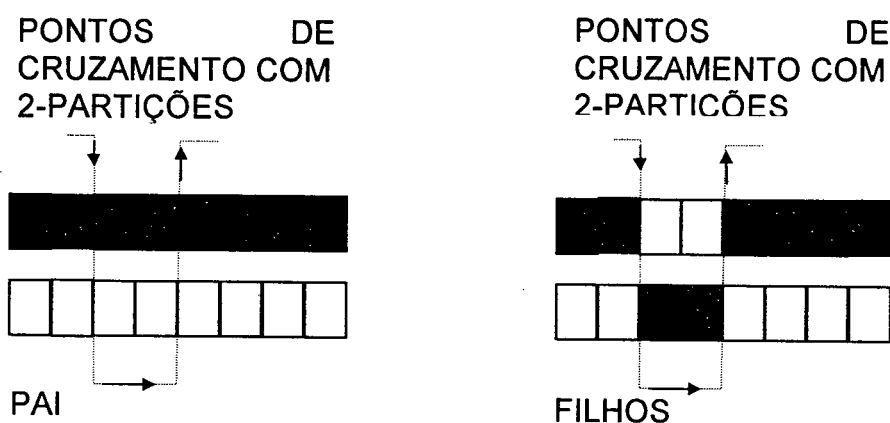


Figura 2.6 Exemplo de cruzamento com 2-partições.

b) Mutação

A mutação consiste em perturbações na cadeia do cromossomo. Desta forma, originará uma nova cadeia de "bits" que poderá guardar pouca ou nenhuma informação da cadeia original. Na realidade, mutação é a denominação dada aos vários mecanismos de alteração genética, os quais têm em comum o fato de alterarem o novo cromossomo. Dentre os principais mecanismos de alteração genética que recebem a denominação global de mutação, destacam-se: troca simples, translocação, inversão, deleção e adição [48,61]. A alteração do comprimento da cadeia do cromossomo não inviabiliza a utilização destes tipos de mutação, apenas dificulta a implementação.

Na adição, ocorre a inserção de mais um gene na cadeia e na deleção é justamente o oposto, ocorre a retirada de um gene da cadeia. Geralmente

estes mecanismos não são utilizados em algoritmos genéticos⁷, pois alteram o comprimento da cadeia do cromossomo [61].

A troca simples consiste de um erro de cópia de um ou mais genes da cadeia. Se um gene for considerado como sendo um “bit” com valor lógico V , a ocorrência de troca simples levaria este bit (gene) para nível lógico F e vice-versa. Na figura 2.7 tem-se um exemplo utilizando-se o alfabeto binário $\{0, 1\}$. Já a inversão consiste na retirada e inserção de uma parte da cadeia, porém na ordem inversa da que foi retirada. Ao contrário da inversão, na translocação uma parte do cromossomo é retirada e colocada em outra posição do mesmo cromossomo, guardando a mesma ordem com que foi retirada. Estes três mecanismos não alteram o comprimento original da cadeia e como a maior parte dos trabalhos em algoritmo genético utilizam cadeia de comprimento fixo, estes são os mais comumente utilizados [61]. No entanto como na maioria dos trabalhos com AG, este também utiliza o termo mutação como sinônimo de troca simples [46,49,59,61].

Por último, após a seleção e a aplicação dos OGs, tem-se uma nova geração a qual deve ser avaliada. A avaliação consiste da obtenção da aptidão de cada indivíduo da população atual para análise da convergência e/ou continuidade do processamento. Caso a geração atual não esteja adaptada o suficiente, deve-se repetir o processo até que a aptidão seja aceitável. Quando a aptidão média da população for aceitável, o melhor indivíduo desta geração provavelmente será a solução desejada [49,52,59,61].

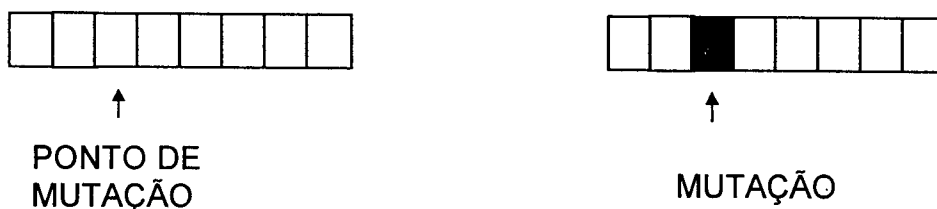


Figura 2.7 Exemplo de mutação (troca simples).

⁷ Isso se dá somente pela “dificuldade” de implementação algorítmica.

2.2.4 Operação do Algoritmo Genético

Desde que Holland [46] propôs os algoritmos genéticos em 1975, inúmeras contribuições têm surgido com o intuito de sugerir alterações na idéia original [52,69]. No entanto, a abordagem seguirá o modelo de AG proposto por Holland [46], o qual é denominado atualmente de algoritmo genético simples (AGS) ou canônico [49].

O procedimento básico do AGS, mostrado na figura 2.8, basicamente segue os passos:

- a) Geração da população inicial;
- b) Avaliação da população;
- c) Teste de convergência;
- d) Seleção;
- e) Aplicação dos operadores do algoritmo genético;
- f) Nova geração.

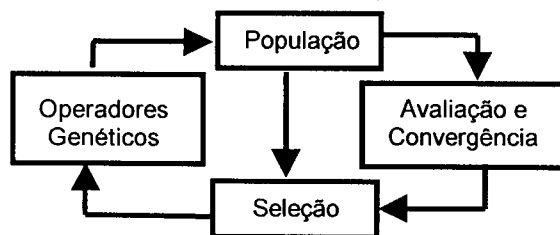


Figura 2.8 Ciclo básico do algoritmo genético.

A seguir cada uma destas etapas será abordada de forma mais detalhada, visando acrescentar algumas informações necessárias à compreensão da operação de um AGS.

a) Geração da População Inicial

A população inicial pode ser obtida através da geração aleatória com distribuição uniforme dos indivíduos em um intervalo previamente definido pelo usuário. Este intervalo é definido levando-se em consideração, algum conhecimento prévio do problema a ser otimizado. Quanto mais restrito for o intervalo inicial, mais rápida será a convergência, isso porque os valores

gerados estarão mais próximos da solução desejada. Contudo, caso o intervalo seja muito restrito e por erro ou desconhecimento esteja longe da solução desejada, a convergência será mais lenta e poderá até mesmo nem ocorrer devido, por exemplo, a perda da diversidade, de que se falará adiante.

O número de elementos que comporá a população ainda é uma heurística, ou seja, depende muito da experiência do usuário e do seu conhecimento prévio sobre a função a ser otimizada. É claro que quanto maior o número de indivíduos da população, maior é a probabilidade de convergência, tendo em vista que aumenta a chance da solução desejada constar entre os elementos da população inicial. Em contrapartida, o tempo de processamento também aumenta. Muitos autores têm sugerido valores para o tamanho da população, taxa de cruzamento e mutação [52,56,68]. Porém, estes valores, de forma geral, são válidos para o domínio da aplicação em que foram testados.

O número de elementos na população, a probabilidade de ocorrer cruzamento e a probabilidade de acontecer mutação (a serem vistos adiante) são denominados de parâmetros de controle do AG [49].

b) Adaptação da População

A adaptação da população consiste na apresentação de cada indivíduo à função custo e após, é atribuído um valor proporcional ao seu desempenho (aptidão).

A função custo é uma função matemática representativa do problema, isto é, do ambiente onde a população de indivíduos está inserida. Contudo, não precisa ser o modelo do processo a ser otimizado, até mesmo porque se o modelo existisse e fosse bem comportado, os métodos clássicos de resolução seriam mais eficientes e eficazes. Portanto, quanto mais representativa do problema for a função, maiores são as chances de sucesso na otimização com o AG.

Geralmente o valor atribuído a cada indivíduo, após tê-lo submetido a função custo, é o próprio valor resultante. Este valor representa a aptidão do indivíduo ao ambiente em questão.

c) Convergência

Na convergência, analisa-se o desempenho da população para ver se o objetivo foi atingido. Isto pode ser feito através de vários fatores, tais como: valores máximo, mínimo, média, desvio padrão da população, etc. O parâmetro mais comumente utilizado é o desvio padrão dos valores de aptidão dos indivíduos da população. Assim, tem-se uma comparação do desempenho da geração atual com a anterior e se o desvio padrão for igual ou menor ao estabelecido como aproximação aceitável, o processo de busca é finalizado [69].

Como o AG é regido por população, se na população inicial houver um elemento que seja a resposta exata do problema, o AG ainda assim poderá não finalizar o processo de busca da solução. A finalização ou convergência só ocorrerá quando houver pequena variação na adaptação média da população atual em relação a anterior. Isto indica que a população se adaptou ao meio, ou seja, os elementos da população, suficientemente próximos, levam a função ao valor otimizado e/ou desejado [49].

Contudo, na prática pode ocorrer, nas primeiras gerações, uma rápida convergência para uma solução sub-ótima, porém, não o desejado ótimo global. Este problema é denominado de convergência prematura, podendo ocorrer principalmente devido à má distribuição da população inicial em torno de um ponto sub-ótimo no espaço de busca. 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 grande chance de dominar a próxima geração e assim sucessivamente, se não aparecerem outros indivíduos com melhor valor de aptidão [52,56]. Este caso agrava-se quando for utilizada a roleta simples (a ser vista adiante).

Esta má distribuição dos indivíduos no espaço de busca também recebe a denominação de perda da diversidade [52,69]. Segundo Júlio Tanomaru [52], o conceito de diversidade “indica o grau em que as mais diversas regiões de busca estão representadas no espaço de busca”. Este problema pode ser amenizado de várias formas, tais como: escolha criteriosa do número de indivíduos na população; melhoria da distribuição dos indivíduos da população inicial no espaço de busca; e principalmente pela escolha do processo de seleção visando impedir ou atenuar a perda de diversidade nas primeiras gerações.

d) Seleção

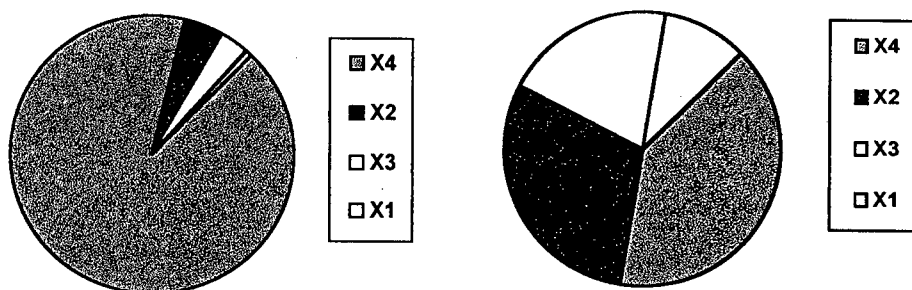
A seleção basicamente tem por objetivo fazer com que os indivíduos mais aptos da geração anterior tenham maior probabilidade de participarem do processo que irá gerar a nova população.

O processo de seleção tem início após a verificação da aptidão de cada indivíduo e a análise da não convergência dos valores.

O processo de validação fornece os elementos da população em ordem de adaptação. Uma das formas mais comumente empregadas para o processo de seleção é a roleta (denominada doravante de roleta simples) [47,63]. No entanto, conforme demonstrado a seguir, esta técnica promove uma rigorosa pressão seletiva, podendo reduzir bruscamente a diversidade das populações iniciais. Assim, é grande a chance da convergência ser guiada para um máximo local. Como forma de amenizar este problema, utiliza-se uma variante da roleta simples (RS), também conhecida por roleta ponderada (RoIP). No caso da RS, cada indivíduo da população anterior terá uma probabilidade de ser sorteado proporcional a sua aptidão, conforme mostra o exemplo da tabela 2. A soma da aptidão dos 4 indivíduos é 220. Portanto, o primeiro indivíduo apresenta 91% de adaptação relativa à somatória da adaptação da população. Na figura 2.9(a) mostra-se o domínio do primeiro indivíduo sobre os demais.

Tabela 2 Dados para roleta simples.

INDIVÍDUO (ordenados)	APTIDÃO	
	Valor	% do TOTAL
X ₄	201	91
X ₂	10	5
X ₃	7	3
X ₁	2	1
TOTAL	220	100%



a) Roleta Simples

b) Roleta Ponderada

Figura 2.9 – Comparação da aptidão: roleta simples e ponderada.

A roleta ponderada também apresenta os indivíduos ordenados conforme desempenho. Contudo, a distância entre os indivíduos próximos é reduzida, ou seja, a pressão seletiva é atenuada. Na RoIP a cada indivíduo é atribuído um posto (rank), conforme sua ordem na população. A figura 2.9(b) mostra a RoIP construída conforme os dados da tabela 3. Uma análise comparativa das duas técnicas de seleção, RS e RoIP, conclui que a pressão seletiva causada pela RoIP é menor do que a da RS. Isso porque despreza o valor da distância entre as aptidões de dois vizinhos, considerando apenas

que tem aptidões diferentes. Os exemplos apresentados nas tabelas 2 e 3 são bastante representativos. Principalmente quando nas primeiras gerações aparece um indivíduo com um alto valor de aptidão em relação aos demais, por exemplo um máximo local. Neste caso, pode-se finalizar a comparação entre as técnicas evidenciando que no caso da RS, o melhor elemento terá uma relação de 91:1 de ser sorteado em relação ao pior, enquanto que no caso da RoIP a relação é de 4:1.

Tabela 3 Roleta ponderada.

INDIVÍDUOS (ordenados)	APTIDÃO	
	Valor	Posto
X ₄	201	4
X ₂	10	3
X ₃	7	2
X ₁	2	1
TOTAL	220	10

A primeira vista até pode parecer interessante aumentar o foco sobre o melhor indivíduo. Entretanto, não é essa a realidade dos fatos. Por exemplo, considere-se que antecipadamente fosse conhecido que o fenótipo (indivíduo) procurado (ótimo global) fosse 1,1 e que o melhor e o pior indivíduo da população em um determinado momento fossem 1,0 e 0,1, respectivamente, e caso não existisse nenhum outro indivíduo na população com o material genético correspondente a parte decimal do pior indivíduo, o único meio de obter a resposta desejada seria através do cruzamento do melhor com o pior indivíduo. Considerando-se um caso hipotético, em que não houvesse mutação, se fosse utilizada a RS, a chance do pior ser escolhido para cruzamento com o melhor pode ser muito pequena até mesmo desprezível. Este problema, também conhecido como pressão seletiva, é bastante atenuado no caso da utilização da RoIP. Desta forma fica

claro que o cruzamento só levará a uma solução ótima se existir na população o material genético necessário para tal. Caso contrário, se uma resposta exata for imprescindível será necessário a participação da mutação para repor ou introduzir o material genético necessário ao patrimônio genético da população.

e) Aplicação dos Operadores Genéticos

A aplicação dos OGs consiste na manipulação genética através da aplicação dos operadores de cruzamento e/ou mutação. Estes podem ser aplicados em toda ou parte da população sorteada para compor o processo que irá gerar a nova população. Ao final desta etapa terá sido criada uma nova população que deverá repetir os passos anteriores até que a adaptação da população seja aceitável.

Após a definição da composição da nova população (a ser visto no próximo tópico) inicia-se a manipulação genética. Quanto aos operadores genéticos, geralmente costuma-se executá-los em seqüência no AGS. Inicialmente aplica-se o OG de cruzamento e em seguida o da mutação [2,50,52].

O cruzamento com 1-partição, implica na geração de dois números aleatórios com distribuição uniforme: o primeiro, entre 0 e 1, indicará a probabilidade de ocorrer cruzamento e o segundo, o local da realização do cruzamento (ponto de corte). Caso o primeiro número gerado for inferior à probabilidade escolhida para haver cruzamento, por exemplo 0,6, realiza-se o cruzamento propriamente dito. Caso contrário, copia-se os pais para a nova geração. O segundo número aleatório (gerado no caso da ocorrência de cruzamento) indicará a posição de corte do cromossomo para efetuar o cruzamento, conforme mostrado na figura 2.5 (no caso de cruzamento com 1-partição) [49,59,70]. Para tanto, o número aleatório gerado deverá estar entre 1 e $g-1$, onde g é o número de bits do indivíduo (ou genes do cromossomo).

Uma das formas de aplicar o operador genético da mutação (troca simples) necessita da geração de um número aleatório entre 0 e 1 para cada

bit de cada indivíduo. Se este número aleatório for inferior à probabilidade de mutação (definida inicialmente para o problema em questão), executa-se a mutação. Caso contrário, o valor lógico do bit não é alterado. Este processo deve ser repetido para o próximo bit do indivíduo e assim por diante, até que todos os bits do indivíduo tenham sido analisados [52,69].

A probabilidade de ocorrer mutação, principalmente devido a inspiração biológica, é sempre bem menor que a de ocorrer cruzamento, quase que na totalidade das aplicações.

Conforme J. L. Ribeiro [59] existe um compromisso entre os três parâmetros de controle do AGS: tamanho da população, probabilidade de cruzamento e probabilidade de mutação. Muitos autores têm proposto valores para estes parâmetros visando garantir um bom desempenho do AG, porém estes valores ainda fazem parte de heurísticas [49,52,56,60,64].

Resumindo, os indivíduos selecionados para serem manipulados pelos OGs são inicialmente divididos em casais e a estes é aplicado o operador genético de cruzamento. A cada novo indivíduo gerado aplica-se o operador genético da mutação. Ao final, tem-se uma nova geração e desta forma espera-se que, em média, a nova população tenha melhor adaptação que a anterior e assim sucessivamente.

f) Nova População

As populações seguintes à população inicial geralmente apresentam o mesmo número de indivíduos (tamanho). A população inicial tem seus indivíduos gerados aleatoriamente dentro de um intervalo previamente definido. Já a população seguinte é obtida principalmente através da manipulação genética da população inicial.

Uma forma de compor a nova população é fazer com que através do processo de seleção, sejam pareados todos os indivíduos sorteados da geração anterior e a estes casais aplicam-se os OGs de cruzamento e mutação.

Outra forma muito efetiva de compor a nova população é dividi-la em partes. Uma parte seria obtida através da aplicação dos OGs. Já a outra

parte poderia ser obtida através da seleção dos indivíduos da população anterior ou através da cópia dos indivíduos mais aptos. Assim, o melhor indivíduo da população atual e o melhor da anterior seriam mantidos na população; caso o melhor fosse o da população anterior, este substituiria o pior da atual que seria eliminado. O processo de manter na população atual o melhor indivíduo já obtido, denomina-se de população elitista. A forma exposta aqui é apenas um exemplo; geralmente a composição é adaptada ao problema a ser resolvido.

Embora existam vários trabalhos sobre a composição da população, o que ainda persiste são heurísticas.

2.2.5 Teorema Fundamental do Algoritmo Genético

Após a compreensão do mecanismo de funcionamento dos AGS, torna-se importante mostrar matematicamente o funcionamento do mecanismo apresentado até o momento. Além disso, a abordagem matemática realizada a seguir é importante para a compreensão do papel dos OAG.

O esquema ou esquemas (“schema” ou no plural “schemata”) descrito inicialmente por Holland [46] e posteriormente por Goldberg [62], é justamente a explicação matemática do mecanismo de operação do AGS. Esta análise é conhecida pelo título de “Teorema Fundamental do Algoritmo Genético” ou também por “Teorema dos Esquemas” [62].

O processo de busca tem no valor da aptidão a principal informação a respeito da qualidade dos pontos do espaço de busca da população. No entanto, é importante encontrar outras informações que de alguma forma indiquem o melhor e mais rápido caminho de busca. Logo, torna-se importante levantar informações contidas em uma população e que auxiliem no direcionamento da procura no espaço de busca.

Exemplificando, considere o espaço de busca representado por quatro indivíduos com os seus respectivos valores de adaptação, conforme mostra a tabela 4.

Analisando os dados da tabela 4 conclui-se que os dois indivíduos mais aptos, além do melhor valor, possuem em comum o primeiro bit como

sendo 1. Portanto, há uma relação entre esta similaridade dos dois indivíduos e os seus respectivos valores de aptidão. Esta similaridade pode ajudar a guiar o processo de busca [46,69].

Holland definiu esquema (E) como sendo um subconjunto de um indivíduo com certas posições similares [69].

A demonstração do Teorema Fundamental do Algoritmo Genético será realizada a partir do alfabeto binário {0, 1}, acrescido do símbolo asterisco (*) que inserido em uma certa posição do esquema, indica que o valor desta posição não é importante, ou seja, pode ser "0" ou "1" [52,69]. Para o exemplo da tabela 5, as posições de interesse são denominadas de "0" ou "1" em relação ao indivíduo.

Tabela 4 Indivíduos x Aptidão.

Indivíduo	Aptidão
01101	169
11000	576
01000	64
10011	361

Sendo os esquemas uma poderosa ferramenta para análise do grau de similaridade entre os indivíduos, torna-se importante avaliar o número de possíveis indivíduos e esquemas que podem ser obtidos a partir de uma "string". O número de indivíduos em uma "string" de comprimento g (número de bits) e cardinalidade k (número de símbolos no alfabeto), é de K^g diferentes indivíduos [69]. Já para o número de diferentes esquemas, a cardinalidade deve ser acrescida de 1, para levar em consideração o locus com valor de alelo indiferente (*). Logo, o número de diferentes esquemas é dado por $(k+1)^g$. Como exemplo, em uma "string" de 5 bits de comprimento ($g = 5$) tem-se $2^5 = 32$ possíveis indivíduos diferentes e $(2+1)^5 = 243$ possíveis esquemas.

Tabela 5 Exemplo de um indivíduo e 4 possíveis esquemas.

Indivíduo	Esquema		
	E1	E2	E3
11011	1***1	11**1	**01*

A quantificação das diferenças entre os esquemas mostrados na tabela 5, fica a cargo de duas propriedades: a ordem do esquema (denotado por $O(E)$) e comprimento do esquema (denotado por $\delta(E)$) [69].

A ordem de um esquema E , $O(E)$, representa o número de 1's e 0's fixos no esquema E [69]. Por exemplo o esquema $E1 = 1***1$ tem ordem 2, $O(1***1) = O(E1) = 2$ e o esquema $E2 = 11**1$ tem ordem de $O(E2) = 3$.

O comprimento do esquema E , $\delta(E)$, representa a distância entre a primeira e a última posição de interesse no esquema [69]. Por exemplo, o esquema $E1 = 1***1$ tem comprimento $\delta(E1) = 5 - 1 = 4$, pois a última posição de interesse é 5 e a primeira é 1.

Os esquemas e suas propriedades são importantes ferramentas para discussão e classificação de similaridades e permitem analisar os efeitos da seleção, cruzamento e mutação de geração para geração [69]. Estes efeitos são abordados de forma individual e ao final em conjunto, obtendo-se desta forma a equação do Teorema Fundamental do Algoritmo Genético.

A seleção, como já visto anteriormente, tem por objetivo garantir aos indivíduos mais aptos da geração anterior a maior probabilidade de participarem da composição da próxima geração.

Suponha-se que na geração t e população $P(t)$ existam $m(E,t)$ possíveis esquemas E [46,58], e também que a probabilidade de seleção de um indivíduo é a razão de sua aptidão pela somatória da aptidão de todos os indivíduos da população, conforme mostra a equação 12.

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j} \quad (12)$$

onde: p_i é a probabilidade de seleção de um determinado indivíduo; f_i é o valor de aptidão do indivíduo; $\sum f_j$ é a somatória dos valores de aptidão dos n indivíduos que compõem a população em questão.

Neste caso, pode-se escrever que o número de esquemas $m(E,t)$ que serão selecionados para a próxima geração, $m(E, t+1)$, a partir da população de n indivíduos é:

$$m(E, t+1) = [m(E,t) * n * f(E)] * \left(\sum_{j=1}^n f_j \right)^{-1} \quad (13)$$

onde: $f(E)$ é a adaptação média do indivíduo representando o esquema E na geração t .

Considerando a adaptação média (f_M) da população $P(t)$, tem-se:

$$f_M = \left(\sum_{j=1}^n f_j \right) * n^{-1} \quad (14)$$

Logo:

$$m(E, t+1) = (m(E,t) * f(E)) * (f_M)^{-1} \quad (15)$$

A equação 15 mostra que o número de esquemas E da geração t que possuir valor de aptidão $f(E)$ acima do valor médio da aptidão da população f_M , crescerá exponencialmente nas futuras gerações e os que tiverem valor inferior ao valor da média da população, decrescerão também exponencialmente [46].

Contudo, apesar do comportamento altamente promissor apresentado pelo processo de seleção, este sozinho não consegue explorar novas

regiões do espaço de busca. Isto ocorre porque no processo de seleção não há geração de nenhum novo ponto do espaço de busca, todos são copiados da geração anterior sem alteração. A função de promover exploração de novos pontos/regiões do espaço de busca cabe ao cruzamento e a mutação [46].

O cruzamento, conforme já descrito, ocorre entre dois indivíduos (pais) visando obter novos indivíduos (filhos) mais adaptados, sendo que os pais são escolhidos na população atual através do processo de seleção [69].

No exemplo abaixo, tabela 6, tem-se um indivíduo e dois possíveis esquemas (E1 e E2):

Tabela 6 Exemplo de um indivíduo e dois possíveis esquemas.

Indivíduo	Esquema	
	E1	E2
11011	**01*	1**1*

No indivíduo mostrado na tabela 6, caso o ponto de cruzamento (escolhido de forma aleatória) tenha sido o segundo bit, o esquema E1 não será destruído, ao contrário de E2, que com o mesmo ponto de cruzamento será destruído. Observando isto, conclui-se que a probabilidade de destruição será maior quanto maior for o comprimento do esquema $\delta(E)$ [46,69]. Como o esquema E2 possui comprimento $\delta(E2) = 3$ e o indivíduo tem $g = 5$ bits de comprimento, o número de possíveis locais de cruzamento é de $(g - 1) = 5 - 1 = 4$ locais. O que nos permite dizer que o esquema E2 tem 3/4 de chance de ser destruído por cruzamento. Em termos matemáticos pode-se escrever:

$$p_{dc} = \frac{\delta(E)}{g - 1} \quad (16)$$

onde: p_{dc} é a probabilidade de destruição por cruzamento do esquema E de comprimento $\delta(E)$ e cujo indivíduo tenha g bits de comprimento.

No entanto, a equação 16 só é válida se todos os indivíduos da população sofrerem cruzamento. Como cada indivíduo tem a probabilidade de sofrer cruzamento de p_c , a equação 16 torna-se:

$$p_{dc} \geq p_c * (\delta(E)) * (g - 1)^{-1} \quad (17)$$

Logo, a probabilidade de sobrevivência de um esquema E por cruzamento, p_{sc} , é:

$$p_{sc} \geq 1 - p_{dc} \geq 1 - p_c * (\delta(E)) * (g - 1)^{-1} \quad (18)$$

Isso nos permite escrever a equação 19 que representa os efeitos combinados da seleção e do cruzamento.

$$m(E, t + 1) \geq m(E, t) * (f(E)) * (f_M)^{-1} * \left[1 - p_c * \frac{\delta(E)}{g - 1} \right] \quad (19)$$

A equação 19 mostra que a sobrevivência de um determinado esquema depende do valor da aptidão do esquema em relação ao valor médio da aptidão da população, f_M , e do comprimento do esquema $\delta(E)$, devido a seleção e ao cruzamento respectivamente [69].

Enquanto que o cruzamento promove uma busca de um provável melhor indivíduo dentro do patrimônio genético existente na população atual, a mutação visa repor algum material genético bom e que tenha sido perdido ou que não esteja presente na população. Logo, a mutação é uma forma de se obter novos pontos do espaço de busca através da inserção de novo material genético [69]. Em virtude de sua forte atuação aleatória, este operador tem baixa probabilidade de ocorrência. Caso contrário, faria com

que o algoritmo genético se aproximasse do processo de busca aleatória [52,69].

Há várias formas de emprego da mutação. A primeira é escolher um indivíduo e gerar um número aleatório entre 0 e 1. Caso o valor gerado seja inferior à probabilidade de ocorrer mutação, p_m , gera-se um segundo número aleatório inteiro entre 1 e g (comprimento do indivíduo), sendo que este número indica qual a posição que deve ocorrer a mutação [61]. Com esta técnica somente um bit do elemento sorteado sofrerá mutação. A segunda forma consiste em gerar um número aleatório entre 0 e 1 para cada bit de cada indivíduo da população. Sendo este valor inferior à probabilidade de mutação, p_m , realiza-se a mutação [52,69]. Nesta técnica, um único indivíduo pode sofrer mutação em vários bits, ao contrário da primeira, em que somente um bit de cada indivíduo escolhido é que sofre mutação.

Para um esquema E sobreviver à mutação, todos os bits deste têm que sobreviver. Como a sobrevivência de um bit é $(1 - p_m)$, a sobrevivência de um esquema à mutação (p_{sm}) é a sobrevivência de cada bit do esquema tantas vezes quanto for a ordem do esquema, ou seja:

$$p_{sm} \geq (1 - p_m)^{O(E)} \quad (20)$$

Para pequenos valores de p_m , como é usual em AG ($p_m \ll 1$), a equação 20 pode ser aproximada para:

$$p_{sm} \geq (1 - O(E) * p_m) \quad (21)$$

A influência da mutação, dada pela equação 21, é inserida na equação 19 e gera a equação 22, segundo a qual temos a população de um determinado esquema E na próxima geração, $m(E, t+1)$, após a aplicação do processo de seleção e dos operadores genéticos de cruzamento e mutação na população atual do esquema E , $m(E, t)$.

$$m(E, t + 1) \geq m(E, t) * (f(E)) * (f_M)^{-1} * \left[1 - p_c * \frac{\delta(E)}{g - 1} - O(E) * p_m \right] \quad (22)$$

A equação 22 é conhecida como Teorema Fundamental do Algoritmo Genético ou Teorema dos Esquemas, segundo o qual os esquemas que tiverem aptidão superior à aptidão média da população crescerão exponencialmente, enquanto que os que tiverem aptidão média inferior decrescerão exponencialmente [52,58,69].

Essa característica é altamente promissora. No entanto, ela depende de fatores tais como probabilidade de ocorrer cruzamento, p_c , e a probabilidade de ocorrer mutação, p_m , cujos valores são determinados com forte predomínio de heurísticas. Não há um procedimento objetivo ou fórmula para determinar os valores para esses parâmetros e que proporcionarão o melhor desempenho ao AGS. Há também a influência do número de indivíduos necessários à composição da população ou espaço de busca e do inter-relacionamento desses parâmetros entre si em função do problema a ser otimizado.

2.2.6 Evolução e Aprendizado Individual

É comum encontrar, nos artigos de CE, equívocos envolvendo a evolução e o aprendizado. A evolução está relacionada com toda a população e ocorre a nível de população de indivíduos que se reproduzem, segundo a seleção natural e estão sujeitos aos mecanismos de cruzamento e mutação. Já o aprendizado ocorre a nível do indivíduo, através da iteração deste com o meio no qual encontra-se inserido. Conforme a habilidade do indivíduo em se adaptar ao meio, pode haver um desempenho melhor ao longo do tempo, com a experiência adquirida. Outra diferença básica é que as mudanças evolutivas geralmente são cumulativas, ao contrário do aprendizado. Com exceção do aprendizado Lamarckiano, somente as mudanças evolutivas é que alteram o genótipo [71].

Hinton and Nowlan (1987)[72] foram os primeiros pesquisadores a estudar a inclusão do aprendizado utilizando AG. Estes analisaram o desempenho do AG sozinho e com a inclusão do aprendizado (efeito Baldwin) sob uma superfície com um único ótimo global. Aos indivíduos foi permitido o aprendizado através de uma busca aleatória. Os resultados segundo os autores, mostraram que o AG sozinho falhava enquanto que com a inclusão do aprendizado através do efeito Baldwin foi possível encontrar o ótimo global.

Apesar dos resultados obtidos até o momento, onde geralmente, o aprendizado aparece como benfeitor, este acarreta em custos para o processo de busca. O aprendizado requer uma certa dose de experimentação na região onde se localiza o indivíduo. Isto significa mais dados, mais processamento e, portanto, mais tempo despendido na busca de um ótimo global experimentando localmente. Além disso, Turney [55] esclarece que o aprendizado suaviza a superfície de adaptação. Assim, se esta superfície já é suave, esta ficará ainda mais suave, tornando a aproximação do ótimo global mais lenta. Uma abordagem mais detalhada sobre os custos e benefícios do aprendizado pode ser encontrada em [55,73,74].

Enquanto na evolução Lamarckiana o conhecimento adquirido por um indivíduo (fenótipo) é incorporado ao seu código genético (genótipo), no aprendizado Baldwiniano, o mesmo conhecimento adquirido pelo indivíduo não é incorporado ao código genético. Neste último caso, a habilidade do indivíduo em adquirir conhecimento ou melhorar o seu desempenho ao longo de sua vida é que será utilizada como forma de melhorar a sua aptidão. Logo, também estará sendo melhorada a probabilidade deste indivíduo em se reproduzir e incorporar o seu patrimônio genético ao da população[75,76].

Em resumo, a evolução não é uma propriedade emergente de uma população de indivíduos que aprendem. Entretanto ela é indiretamente afetada pelo aprendizado individualmente dos elementos de sua população.

Na figura 2.10 pode ser visto de forma mais detalhada o que é e qual a contribuição esperada da evolução, do aprendizado e também uma das

formas de implementar o aprendizado Lamarckiano e o Baldwiniano em conjunto com o AG.

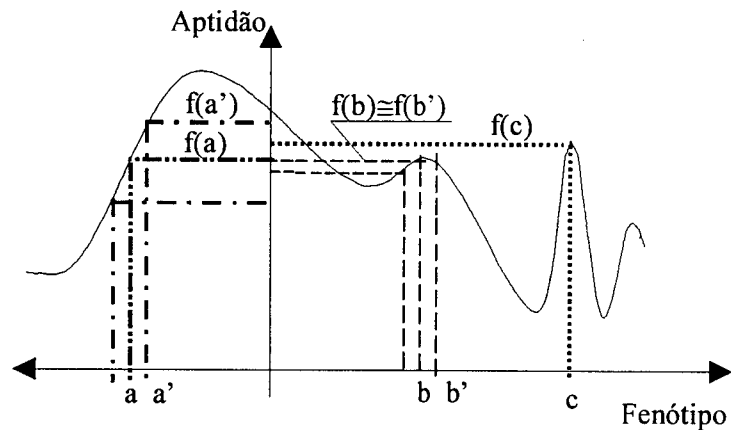


Figura 2.10 Pressão seletiva da evolução e do aprendizado

No eixo horizontal está representado o genótipo de três indivíduos “a”, “b” e “c” sendo o grau de adaptação (aptidão) de cada indivíduo $f(a)$, $f(b)$ e $f(c)$, respectivamente, medido no eixo vertical.

Após a avaliação dos três indivíduos (conforme figura 2.10), caso fosse empregada somente a técnica evolutiva, o “rank” relativo a adaptação forneceria como indivíduo mais adaptado o indivíduo “c”, seguido por um empate entre os indivíduos “a” e “b”. Contudo, uma análise da figura 2.10 mostraria que o ponto “c” é um máximo local e apesar do desempenho semelhante dos indivíduos “a” e “b”, o indivíduo “a” está mais próximo do máximo global do que o indivíduo “b”.

Já a aplicação do aprendizado sob o genótipo obtido por processos evolutivos, dependendo da avaliação local realizada em torno de cada um dos 3 indivíduos “a”, “b” e “c”, acrescentaria mais uma informação para guiar o AG na sua busca através de processos evolutivos. Esta informação mostra que há maior habilidade em encontrar o máximo global com o indivíduo “a” seguido do “b” e por último com o indivíduo “c”. Esta informação é exatamente oposta à obtida por processos puramente evolutivos. Assim,

enquanto a busca evolutiva faria maior pressão seletiva sob o indivíduo “c”, a busca através do aprendizado localizaria a pressão seletiva sob o indivíduo “a”, neste momento do treinamento.

2.2.7 Inclusão do Aprendizado

O bom desempenho do AG na busca do máximo global em grande parte é atribuído ao fato do AG trabalhar com uma população de indivíduos, ao contrário de outros métodos, como os que utilizam o gradiente que trabalha somente com um indivíduo. Com isso consegue-se explorar uma grande parte do espaço de busca de uma única vez [56]. Já o fato de ser aplicável a situações onde o modelo matemático não é claramente definido e a funções lineares e não-lineares deve-se principalmente à necessidade de se ter somente uma função custo (função aptidão) que represente o problema a ser otimizado.

Contudo, em virtude da convergência lenta e até mesmo crítica dos AGs quando o erro torna-se pequeno, recomenda-se utilizá-lo de forma híbrida [52]. Nesse caso, o AG seria encarregado da aproximação necessária do máximo global e outros métodos, como o gradiente, ficariam encarregados do ajuste fino.

A computação evolutiva ou evolucionária ainda não se encontra plenamente estabelecida; por esse motivo é comum encontrar divergências em vários pontos tais como, por exemplo, os tipos de operadores e seu real papel, o tamanho da população e o relacionamento destes itens com o problema a ser resolvido. No entanto é crescente o interesse por esta área e grandes contribuições têm surgido.

O algoritmo genético simples ou canônico apresentado aqui, considera que os indivíduos tem de se adaptar ao meio que lhes é apresentado (função custo), não havendo possibilidade de interação entre estes dois elementos (indivíduo e meio). Atualmente as pesquisas tem mostrado que o AG está ganhando um forte aliado, quando passa a admitir que as características aprendidas pelos indivíduos em uma geração podem ser passadas para os indivíduos da próxima geração. Assim, passa-se a

admitir a interação com o meio. Na revisão realizada, foi analisado o benefício do aprendizado, bem como as duas formas básicas de implementação. Também mostrou-se a forma pela qual a evolução e o aprendizado podem interagir. Após ter sido explanado o que é e como opera o AG, será apresentada a metodologia de inclusão do aprendizado no AG, formando o algoritmo genético híbrido (AGH).

a) Aprendizado Lamarckiano

Conforme já explicado anteriormente, o aprendizado Lamarckiano, também denominado de evolução Lamarckiana, defende a hereditariedade das características adquiridas diretamente para o patrimônio genético.

Uma forma de empregar a evolução Lamarckiana pode ser vista na figura 2.10. A avaliação local de cada um dos 3 indivíduos, mostrou que a aptidão do indivíduo “a” é inferior à obtida em torno de sua vizinhança, por exemplo no ponto “a’ “. O mesmo acontece com o indivíduo “b” que segundo a avaliação local, poderia ser melhor representado por “b’ “. Já para o indivíduo “c”, por ser um máximo local, uma avaliação local nas mesmas proporções da realizada em torno de “a” e “b” mostrou que só haveria piora do desempenho. Portanto, a informação do aprendizado seria utilizada para substituir o genótipo “a” por “a’ “ e “b” por “b’ “. Já as novas aptidões seriam $f(a')$ e $f(b')$. A aptidão é utilizada para selecionar os pares para cruzamento e mutação. Logo, a aplicação do aprendizado Lamarckiano fez com que o genótipo anterior (“a” e “b”) fosse perdido. Portanto, o aprendizado Lamarckiano é disruptivo à capacidade de processamento dos esquemas (“schematas”)[75,77]. Esta crítica é forte por parte dos usuários de AG porque estes têm no crescimento dos esquemas favoráveis a sua base funcional e, portanto, qualquer fator prejudicial a este pode descaracterizá-lo. Talvez, dependendo da intensidade da quebra do processamento baseado nos esquemas favoráveis, o AG passe a operar de forma mais aleatória.

Apesar das críticas, é crescente o número de trabalhos que procuram avaliar a viabilidade de aplicação do aprendizado Lamarckiano bem como compará-lo ao aprendizado Baldwiniano.

b) Aprendizado Baldwiniano

Baldwin em 1896 introduziu o que ele denominou de seleção orgânica para explicar como as características adquiridas por membros de uma população durante suas vidas, poderiam ser incorporadas ao patrimônio genético das novas gerações, mas de forma indireta, também denominado de efeito Baldwin[74].

À semelhança do aprendizado Lamarckiano, os indivíduos são avaliados em sua vizinhança, porém, esta informação obtida não é utilizada para alterar o patrimônio genético e sim a sua aptidão. Assim, o AG receberia a informação de que o indivíduo "a" inicialmente avaliado como tendo aptidão $f(a)$, teria a sua aptidão atualizada para $f(a')$. O mesmo aconteceria para "b" que teria a aptidão $f(b)$ trocada por $f(b')$. A aptidão de "c" não seria trocada porque ao redor dele não há como desenvolver nada melhor do que o próprio indivíduo, ou seja, apresenta baixo grau de adaptação.

O aprendizado Baldwiniano troca apenas a aptidão dos indivíduos, com isso a pressão seletiva é direcionada para os indivíduos que apresentarem maior desempenho naquele momento da busca, na esperança de que isso seja uma tendência. No entanto, caso a tendência não se mantenha na próxima geração, uma simples mudança da aptidão alterará a pressão seletiva da busca sem que no entanto o patrimônio genético seja afetado, em um primeiro momento. Ao contrário da pressão seletiva de origem puramente evolutiva, a pressão seletiva devido ao aprendizado leva em consideração o desempenho evolutivo do indivíduo, além do próprio aprendizado[71].

O aprendizado Baldwiniano tenta privilegiar os indivíduos que apesar de sua aptidão inicial, apresentam uma grande capacidade de elevar o seu desempenho em relação ao estado inicial. Logo, o que é herdado é a habilidade em adquirir características ou seja, a habilidade em aprender e isso é possível se o indivíduo tiver um alto grau de adaptação. Baldwin propôs que o aprendizado é uma vantagem onde um novo comportamento começa a envolver uma dada população[55].

2.3 ESTADO DA ARTE NO TREINAMENTO HÍBRIDO DE RNA

2.3.1 Introdução

Na utilização de RNA destacam-se dois grandes problemas a serem sanados pelo projetista: a definição da topologia e o treinamento desta para que atinja satisfatoriamente o objetivo a que se destina.

A escolha da topologia, no caso de ser a primeira abordagem do problema e de não haver conhecimento prévio disponível, ainda hoje depende necessariamente da experiência do projetista e de algumas possíveis tentativas frustradas. A escolha de uma topologia não otimizada estruturalmente acarreta uma carga computacional que pode limitar a aplicação prática da RNA e encarecer ou inviabilizar a sua implementação em "hardware".

Nos casos onde já se tem uma topologia bem definida ou quando a experiência permite defini-la satisfatoriamente, resta ao projetista executar o treinamento. O treinamento poderá ser realizado para uma rede ainda não treinada, bem como para redes já treinadas, mas que por algum motivo tenham seu desempenho degradado.

a) Otimização da Topologia

Considerando novamente o cérebro humano como fonte de inspiração, sabe-se que este não é totalmente interconectado [78]. Este fato tem reforçado a conjectura de que uma topologia otimizada não necessariamente apresenta-se totalmente conectada. Devido à inexistência de uma metodologia de cálculo para a determinação da melhor topologia para um determinado problema, o projetista e/ou especialista acabam por escolher uma topologia não otimizada estruturalmente (número de interconexões, de neurônios e de camadas).

O problema da busca de uma topologia otimizada é uma tarefa difícil tendo em vista a sua complexidade. A inexistência de um modelo que permita a aplicação de técnicas de busca clássica obriga à utilização de funções custo, as quais devem ser otimizadas. No entanto, a função custo associada ao espaço de busca da topologia é complexa, não-diferenciável,

multimodal e enganosa [70]. Para contornar estas dificuldades, tem-se atualmente no AG uma boa opção de trabalho.

Os vários trabalhos de otimização da topologia seguem basicamente dois tipos de esquema:

- a) nível baixo;
- b) nível alto.

No esquema de representação de nível baixo, codifica-se diretamente a topologia da rede a ser otimizada. Cada neurônio e cada peso são especificados separadamente, e por isso diz-se que esse tipo é transparente e fácil de ser empregado.

Já no esquema de nível alto tem-se uma codificação mais complexa da topologia da rede. Não há uma codificação direta da topologia, esta é apresentada por uma representação. Dependendo da capacidade da representação, pode-se trabalhar com redes de maior tamanho quando comparados ao esquema de nível baixo [37]. Exemplos dessa técnica são as propostas de Gruau [79], Schiffman [78] e Palmer et al [80].

Ao término da seleção de uma topologia, tem-se também os pesos apropriados ao mapeamento do conjunto de entrada na correspondente saída. Contudo, se houver variações no processo e o desempenho da rede degradar, haverá necessidade de treiná-la novamente. A definição de uma nova topologia será cogitada caso a degradação seja elevada, a ponto do ajuste dos pesos não corresponder à necessidade do processo.

b) Otimização do Treinamento

Haykin [17] define treinamento no contexto de RNA como: "... processo pelo qual os parâmetros de uma rede neural são adaptados, através de um processo contínuo de estimulação pelo ambiente onde a rede se encontra inserida. O tipo de aprendizado é determinado pela forma como as mudanças de parâmetros são executadas".

Os métodos baseados em cálculo são inevitavelmente superiores no domínio de problemas onde eles possam ser usados. Contudo, em

problemas onde os métodos clássicos falham ou não existem, o AG apresenta-se como um método a ser considerado.

Conforme Miller et al [81], a determinação do conjunto de pesos se caracteriza por ser um espaço de busca complexo, que como tal não é eficientemente explorado por técnicas enumerativas, aleatórias ou mesmo por métodos de busca guiados por conhecimento heurístico. Em contraste, a natureza altamente adaptativa do AG lhe confere certa habilidade em se mover em torno do espaço de soluções, sem uma forte dependência da estrutura ou localidade, providenciando um método robusto para a busca em questão.

As abordagens para otimização dos pesos de RNAs com AG, geralmente diferem segundo uma ou mais das seguintes características:

- a) tamanho da população;
- b) operadores genéticos usados;
- c) estratégia de renovação da população.

Inicialmente, as tentativas de melhoria do treinamento com AG foram implementadas sob a taxa de treinamento. Talvez isso tenha ocorrido por inspiração no teorema da convergência [42], segundo o qual o algoritmo de RP sempre converge para um mínimo, desde que as correções dos pesos sejam infinitesimais. Além disso, outro fator que pode ter inspirado um procedimento desse tipo é que, se para cada peso for atribuído uma taxa de aprendizado e esta for adaptativa, a convergência é acelerada [20]. Porém, não é garantida a convergência para um ponto desejado [82].

As potencialidades do AG são muitas, mas existem limitações quanto ao seu emprego. O AG é bastante sensível aos parâmetros de controle, tais como: tamanho da população, faixa inferior e superior dos pesos iniciais, taxa de cruzamento e mutação, estratégia de renovação da população, codificação das variáveis a serem otimizadas e principalmente a representatividade da função custo. Como se já não bastasse a quantidade de parâmetros, tem-se ainda como agravante a interdependência dos pesos.

2.3.2 Revisão da Literatura

A primeira abordagem sobre evolução dos pesos em uma rede fixa foi realizada por Montana e Davis (1989) [83]. Neste trabalho os autores utilizaram somente o AG como algoritmo de treinamento para encontrar um bom conjunto de pesos para uma RMD. O objetivo era evitar a tendência à paralisia e mínimos locais do algoritmo de RP. O domínio da aplicação empregado foi a classificação de sons provenientes do oceano, tendo como meta a detecção de sinais interessantes no meio de uma larga variedade de ruído acústico e interferências existentes no próprio oceano. Cada cromossomo da população era formado pelo vetor de 126 pesos lidos em uma ordem fixa (da entrada para a saída e de cima para baixo), a função custo adotada era a soma dos quadrados dos erros de todo o conjunto de treinamento. Como resultado dos experimentos os autores relatam que o conjunto de pesos obtido pelo AG foi melhor do que o obtido pelo algoritmo de RP e de forma mais rápida. No entanto, o algoritmo Quickprop foi superior a ambos (AG e Algoritmo de RP).

Após a publicação de Montana e Davis em 1989, Kitano publicou em 1990 [13] uma dura crítica à aplicação do AG como forma de treinamento de RNA. Nesta crítica, o autor implementou vários experimentos envolvendo AG sozinho, AG híbrido com o algoritmo de RP, algoritmo de RP sozinho e por último o algoritmo Quickprop. Nestes experimentos a representação do cromossomo e da função custo foi a mesma da adotada por Montana e Davis [83]. Portanto, uma população de cromossomos é uma população de redes com diferentes distribuições de pesos. O domínio da aplicação foi o XOR e o codificador/decodificador. Os resultados obtidos através dos vários experimentos levaram Kitano a afirmar que a proposta híbrida de treinamento de RNA com AG e algoritmo de RP sistematicamente é melhor do que somente com AG e inferior ou próximo da variante mais rápida do algoritmo de RP, que é o algoritmo Quickprop. Afirmou também que o desempenho do treinamento híbrido degrada à medida que o número de conexões da rede aumenta. Assim, concluiu "... uma tarefa como o treinamento dos pesos de uma rede neural não é uma boa tarefa para AG, devido à interdependência

entre as regiões do cromossomo” [13]. Portanto, além de não propor uma solução, a crítica recaiu sobre o AG que é uma ferramenta e não na forma como a ferramenta foi empregada.

Os trabalhos que sucederam ao de Kitano podem ser englobados em 3 grandes áreas:

- a) Repetem parte dos experimentos realizados e basicamente chegam à mesma conclusão [34,84];
- b) A partir da conclusão obtida por Kitano, de que o problema do AG reside na interdependência entre as regiões do cromossomo, vários trabalhos têm buscado justificar porque isso ocorre. Nesta linha cita-se: Whitley [15], Radcliffe [85,86] entre outros;
- c) Baseado no fato de que quanto maior o número de neurônios da rede, menor é a eficiência da técnica convencional do AG, buscam-se técnicas de codificação que reduzam o tamanho do espaço de busca [86,87].

Assim, Whitley [15] ao abordar as possíveis formas de emprego do AG em RNA, diz que as aplicações de AG no treinamento de RNA têm-se reduzido por dois fatores:

- a) Os métodos baseados em gradiente tem sido desenvolvidos a tal ponto que se tornaram eficientes para esta tarefa;
- b) O problema de treinar uma RNA representa uma aplicação que inerentemente não é uma boa tarefa para AG, já que este depende pesadamente do cruzamento.

O fato do treinamento de uma RNA não ser uma boa tarefa para AG é justificado por ser um “Competing Conventions Problem”. Já Korning [87] o denomina de Problema de representação multi-simétrico (Multiple Symmetric Representations problem – MSRP). Estas denominações referem-se a característica de que um dado neurônio da camada intermediária bem como seus pesos de entrada e saída, podem ser trocados de lugar com qualquer outro neurônio na mesma camada sem mudanças na funcionalidade da RNA. Isto significa que a superfície de erro é extremamente multimodal. Radcliffe

[85,86] denominou este fato de problema de permutações (Permutations Problem).

Radcliffe sugeriu [86] que os neurônios da camada intermediária deveriam ser analisados segundo a sua conectividade. Assim, esta informação seria utilizada como guia para o cruzamento. Hancock (1992) [88] implementou estas idéias bem como estudou outras formas de identificar a similaridade entre os neurônios da camada intermediária. Contudo, concluiu que o problema do cruzamento não era tão crítico quanto sugerido.

Na verdade o que estes trabalhos tinham em comum era a tentativa de reduzir o espaço de busca eliminando regiões não promissoras que tendem a crescer à medida que é exigida maior capacidade da rede (maior número de neurônios).

Korning em 1994 [87] afirma, sensatamente, que ainda era cedo para descartar o AG como substituto do algoritmo de RP. Partindo das afirmações de Whitley [15] e Radcliffe [86] o autor aborda uma nova técnica de codificar os neurônios para reduzir o espaço de busca. Além da alteração na forma de codificar os pesos no cromossomo, o autor teve cuidados especiais com a inicialização dos pesos, técnica de cruzamento, e a escolha da função custo (empregou a função de entropia relativa). Como domínio da aplicação foram utilizados 5 problemas além do XOR.

A maioria dos trabalhos atuais que utilizam o AG, o fazem mas não segundo a técnica convencional e sim através de algum artifício tal como a codificação através de gramática⁸ [77] entre outras [70,89,90].

Como leitura complementar cita-se o artigo "The design and evolution of modular neural network architectures" de Happel et al [91] que apresenta uma série de considerações sobre o projeto de RNA evolutivas, onde segundo os autores "Um resultado potencialmente importante é que uma arquitetura definida inicialmente por processos genéticos não somente aumenta o aprendizado e desempenho do reconhecimento, mas também induz o sistema a uma melhor generalização de seu aprendizado para

⁸ Gramática é um conjunto de regras que pode ser aplicada para produzir um conjunto de estruturas (Ex.: sentenças em uma linguagem natural, programas em uma linguagem de programação, etc.) [105]

instâncias nunca antes vistas”. E conclui: isto deve explicar o porquê de muitas tarefas de treinamento vital em organismos necessitarem somente um mínimo de exposição a estímulos relevantes.

Este ponto de vista não é comum nos artigos que analisam o desempenho do AG híbrido com o algoritmo de RP no treinamento de RNA. A princípio, o objetivo principal geralmente encontrado era a velocidade de convergência e não a qualidade do treinamento, conforme objetivaram Happel et al em suas considerações acima.

Bäck et al [92] apresentam uma análise comparativa dos algoritmos evolutivos (AE) na otimização de parâmetros. É um bom material para consolidar a leitura e diferenciar a forma de atuação das técnicas envolvidas.

Hochman [93], em sua dissertação, apresenta uma análise interessante quanto a RNA evolutivas, sob o prisma da engenharia de software.

Dentre outros trabalhos que mesclam AG e o algoritmo de RP para a otimização do treinamento de RMD, citam-se: Antonius et al [34], Murray [94], Yi Shang and Wah [95], Chalmers [96], entre outros.

Alguns pesquisadores tentam buscar no aprendizado uma solução para os seus problemas [97]. Contudo, considera-se que a aplicação do aprendizado após a atuação do AG (como comumente é aplicado) deve ter um papel mais próximo do ajuste fino, o qual é realizado pelo algoritmo de RP. Nesta tese, o aprendizado é aplicado como uma forma de descobrir regiões promissoras para que o AG atue. Assim, indiretamente o espaço de busca estaria sendo reduzido. A descoberta de regiões promissoras é uma tarefa difícil. Uma técnica comumente empregada em análise de circuitos é a análise de sensibilidade (AS), a qual pode ser utilizada para identificar o que é e o que não é importante em cada momento do treinamento.

Svarer [98] em sua tese, discutiu a otimização da arquitetura de RNA visando aumentar a capacidade de generalização, velocidade de aprendizado e a carga computacional. A técnica de otimização adotada consiste em analisar a “saliência” de cada peso da rede, onde os menos salientes são removidos. Após a análise de todos os pesos e as cabíveis

remoções, a rede é treinada novamente até a proximidade de um mínimo, onde o processo se repete até que uma remoção eleve o erro obtido em relação à condição anterior. Neste caso assume-se que a arquitetura anterior é mínima e o objetivo foi alcançado. A “saliência” pode ser obtida através do cálculo da sensibilidade do erro da rede em relação a cada peso, de acordo com a sensibilidade (importância) o peso pode ser eliminado ou não.

2.4 SENSIBILIDADE

A sensibilidade é uma grandeza que permite avaliar o comportamento de uma dada função quando um ou mais parâmetros desta função são alterados.

No caso do treinamento de uma RNA, pode-se considerar a função como sendo o erro da época e os parâmetros como sendo os pesos que compõe a rede.

O interesse na aplicação da AS é saber qual a fração do erro da época que mudará quando houver uma fração de mudança em um dado peso da rede.

A sensibilidade da função erro da época (E) à variação dos pesos (W_{ij}) é definida conforme mostra a equação 23. Onde a função sensibilidade ($S_{W_{ij}}^E$) relaciona a variação percentual de E com a variação percentual de W_{ij} , sendo também denominada de sensibilidade normalizada.

$$S_{W_{ij}}^E \equiv \frac{\partial E/E}{\partial W_{ij}/W_{ij}} = \frac{W_{ij}}{E} \cdot \frac{\partial E}{\partial W_{ij}} \quad (23)$$

Além da equação 23, são utilizadas outras definições de sensibilidade denominadas de não normalizadas:

$$S_{W_{ij}}^E \equiv \frac{\partial E}{\partial W_{ij} \cdot W_{ij}} \quad (24)$$

Ou simplesmente:

$$S_{W_{ij}}^E \equiv \frac{\partial E}{\partial W_{ij}} \quad (25)$$

Neste trabalho foi empregada a equação 25, mesmo porque o próprio algoritmo de RP utiliza esta equação, como será visto a seguir.

2.4.1 Gradiente Descendente (GD)

O fato da sensibilidade operar como guia para a correção dos pesos de uma RNA não é novidade para o RP, uma vez que o cálculo da sensibilidade do erro a cada peso (SEP) é utilizado pelo RP não somente como guia, mas também na correção dos pesos na fase de retropropagação dos erros. No RP o fator de sensibilidade é dado pelo gradiente, derivada parcial do erro (E(t)) em relação a derivada parcial do peso ($W_{ij}(t)$), no instante t , equação 26.

$$\Delta W_{ij}(t+1) \propto -\partial E(t) / \partial W_{ij}(t) \quad (26)$$

Portanto, o algoritmo de RP através da informação da qualidade do conjunto dos pesos (erro da época), obtém uma aproximação da relevância de cada peso para o erro naquele instante do processo. Uma distorção que pode ocorrer é que pode-se obter um valor pequeno para o gradiente, como se o peso fosse pouco relevante quando o que pode estar acontecendo é uma saturação na saída da função de ativação do neurônio. Isto levaria o gradiente a valores tão menores quanto mais saturada estiver a função.

Uma forma bastante simples de obter a SEP é utilizar o cálculo do gradiente já incorporado no RP caso este venha a ser utilizado para fazer a aproximação final do treinamento.

2.4.2 Variação dos Pesos (VP)

Outra forma de obter a SEP sem utilizar o gradiente consiste em gerar um conjunto de pesos aleatoriamente para a rede (semente) a qual será treinada e obter o erro da época para este conjunto (ε_0). Para cada peso, além da semente (valor original) geram-se mais dois valores; um 10% acima do valor da semente e o outro 10% abaixo do mesmo valor⁹. Cada um dos dois pesos substituirá o peso original da rede e para cada um deverá ser memorizado o erro da época ($\varepsilon_1, \varepsilon_2$). Os três valores de erro da época serão utilizados para a obtenção do desvio padrão associado ao peso, que no caso de ser o primeiro peso do conjunto será representado por σ_1 , conforme mostra a figura 2.11. Este processo deve ser repetido para cada peso do conjunto; se a rede for composta de 9 pesos (XOR 2-2-1) haverá 19 cálculos do erro da época, 1 para o conjunto original mais 2 para cada um dos 9 pesos. No final a ordenação do desvio padrão dos nove pesos fornecerá a ordem de sensibilidade aos pesos da rede. A observação dos valores do desvio padrão e a comparação dos valores de ε_1 e ε_2 com ε_0 , fornecerá várias informações úteis para direcionar a atuação do AG. A amplitude dos valores obtidos para o desvio padrão permite identificar os pesos que apesar da variação ($\Delta W_{ji}(t)$) de +/- 10% no valor não apresentaram uma correspondente variação no erro da época. Sugere-se que os pesos cuja SEP estejam muito distantes dos mais significativos não sejam processados pelo AG, evitando desta forma perda de tempo computacional. Quanto à sensibilidade, esta também pode ser observada pela amplitude do valor do desvio padrão que mostra a capacidade de variação do erro da época em relação a um determinado peso. Contudo, na prática o que ocorre, com certa frequência, é que a variação no erro da época é para maior, ou seja, os erros ε_1 e/ou ε_2 apresentam-se maiores do que o erro da semente, ε_0 . A sugestão neste caso é que a ordenação da sensibilidade priorize os pesos que apresentarem uma tendência de decrescer o erro da época em relação aos

⁹ Esta variação está relacionada com a qualidade da superfície de erro.

que apresentarem maior desvio padrão. Não faz sentido trabalhar com pesos que só contribuem para aumentar o erro global do conjunto de pesos.

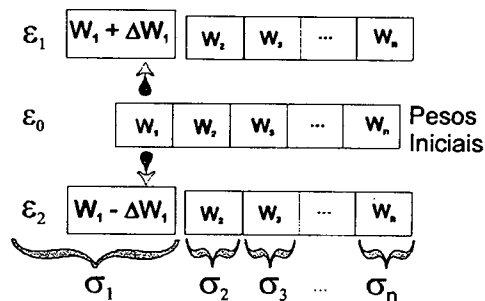


Figura 2.11 – Variação dos pesos originais

Através da análise de sensibilidade dos pesos no começo de cada geração é possível evitar o processamento sobre alguns pesos bem como ordenar a seqüência de atuação do AG sobre os demais. Em virtude do sistema ser interdependente, a alteração da sensibilidade de um dos pesos pode acarretar perda da validade das sensibilidades obtidas para os demais pesos da rede. Esta condição pode ser amenizada se forem processados somente os pesos que apresentarem grande sensibilidade. Como no início de cada geração é executada a análise da sensibilidade, é grande a probabilidade de que o AG esteja na maior parte do tempo processando os pesos mais significativos daquela época. Reduzindo o erro da época e com o passar de algumas gerações espera-se também uma certa homogeneização em torno da sensibilidade média do conjunto. A partir deste momento pode-se adotar a técnica convencional ou incluir mais pesos do conjunto no subconjunto que é processado pelo AG. Este procedimento deve ser executado até o momento da transição para a técnica baseada em gradiente descendente, a qual promoverá o ajuste fino da convergência.

3 METODOLOGIA DA PROPOSTA

Nos capítulos anteriores foram apresentadas as principais críticas à utilização do AG como técnica de treinamento de RNA, bem como ao desempenho do treinamento híbrido com o algoritmo de RP. Estas críticas partem do princípio de que o treinamento de RNA não é uma boa tarefa para o AG. Contudo, neste trabalho considera-se que os resultados obtidos nas pesquisas realizadas até o momento não são suficientes para afirmar que o problema resida na ferramenta adotada. O que se pode afirmar é que a aplicação do AG na forma convencional é fortemente afetada pela quantidade de pesos da rede, ou seja, pelo tamanho da rede.

3.1 INTRODUÇÃO

O treinamento de uma RNA consiste em resolver um sistema multivariável interdependente (SMI). A dificuldade na manipulação deste tipo de sistema é que, por ser interdependente, a alteração de uma única variável (peso) pode alterar significativamente a importância dos demais pesos do sistema em relação ao erro causado pelo conjunto de pesos. No caso do AG, quando implementado segundo a técnica convencional (todos os pesos formam uma única "string"), a única informação disponível é a aptidão, geralmente adotada como sendo o inverso do erro da época (erro médio quadrático produzido pelo conjunto de padrões de treinamento). Em se tratando de SMI a aptidão só é capaz de quantificar a qualidade do conjunto, sem no entanto mostrar onde e quanto poderia ser alterado. Este quantificador, no mínimo, não é representativo o bastante para problemas do tipo SMI. E à medida que a rede cresce, aumenta o número de pesos, tornando-se ainda mais crítica a representatividade da função custo. Isto ocorre tendo em vista que a função custo não externa (individualiza) a relação entre o erro do conjunto dos

padrões de treinamento e cada peso que compõe a rede. Na técnica convencional, o AG é guiado pela seleção dentro da população dos indivíduos que tem sido feita de forma aleatória dentro do indivíduo pelos OGs. O indivíduo é representado por uma "string" composta de várias variáveis. Logo, esta forma aleatória de abordá-lo é pouco representativa, porque não há uma informação que realce o comportamento individual de cada peso dentro do indivíduo. Com isso, a probabilidade de que os OGs manipulem uma região promissora decai à medida que aumenta o número de pesos da rede. Portanto, há necessidade de alguma técnica que auxilie o AG a evitar o processamento de regiões não promissoras. Baseado no exposto acima, poder-se-ia afirmar que não é o AG que não é adequado a esta tarefa como foi afirmado por Kitano[13,15] e sim a forma pela qual é aplicada a função custo ou até mesmo a própria função custo é que seja inadequada.

A avaliação destas afirmações inicialmente passou pela implementação da técnica convencional. Após, a realização de alguns testes sobre esta técnica, implementaram-se outros algoritmos que são apresentados a seguir, na mesma seqüência em que as observações e achados foram evoluindo.

O objetivo deste trabalho não é a obter uma técnica eficaz para o XOR, (domínio do problema adotado neste trabalho) e sim gerar conhecimento válidos no domínio XOR e que poderão ser testados em outros domínios de aplicação. Em todas as implementações a serem mostradas a seguir, o AG foi o responsável pela aproximação inicial do conjunto de pesos de uma região onde provavelmente localiza-se o mínimo global. Cabe ao algoritmo de RP a aproximação final. As alterações realizadas sobre a técnica convencional mostradas a seguir foram efetuadas somente sobre o AG. Nesta fase de testes não foi alterado nem o algoritmo de RP nem sua forma de aplicação. Portanto, resolveu-se ater-se a exposição sobre o AG, o qual teve a forma de atuação alterada de diversas maneiras. No transcorrer deste capítulo, serão apresentados os detalhes, as justificativas e os resultados obtidos nos algoritmos implementados.

A técnica que serviu de base para os testes foi a técnica convencional (descrita no item 1.3), a qual consiste em gerar de forma aleatória uma

população de indivíduos candidatos a solução da rede 2-2-1 do XOR que é composta por 9 pesos (6 na camada de pesos da entrada e mais 3 na camada de saída), conforme figura 3.1. Cada indivíduo da população agrupa os 9 pesos da rede e são processados pelos OAGs durante um certo número de gerações. Ao final, o indivíduo que apresentar o menor erro quando submetido à rede será repassado ao algoritmo de RP para a convergência final. O importante a salientar neste momento é que todos os 9 pesos da rede formam o indivíduo e que o OG não fará distinção entre os pesos, ou seja, o indivíduo é processado ignorando-se a sua composição, o que não será o caso nas demais implementações mostradas a seguir.

Vale ressaltar que o AG, nesta pesquisa, realiza um pré-processamento para estimar o conjunto de pesos mais promissores para a inicialização do treinamento da RMD por RP.

3.2 Implementações da Busca Unimodal

O resultado da análise realizada com a técnica convencional, como já era esperado, foi semelhante ao obtido por Kitano [13]. Porém, o acompanhamento (com o auxílio do ambiente de estudos – Anexo A) efetuado durante a execução deste algoritmo mostrou que nem todas as alterações realizadas pelo AG em certas partes do genótipo acarretam uma modificação na aptidão do indivíduo. Esta observação levou à implementação de uma forma de busca unimodal, mesmo sabendo da relação não linear entre os pesos da rede. Este teste visa aproximar mais a função custo, não só do indivíduo mas também de cada peso deste. Além de possibilitar a observação da relação genótipo versus fenótipo sob o ponto de vista da relevância de cada peso em determinado momento do treinamento.

As implementações apresentadas a seguir são: a) Busca Unimodal Direta; b) Busca Unimodal Reversa; c) Busca Unimodal com Sensibilidade Global; d) Busca Unimodal com Sensibilidade Parcial; e e) Busca Unimodal com Sensibilidade Parcial por Padrão.

a) Busca Unimodal Direta

Assim, foi implementado um novo algoritmo de treinamento cuja alteração recaiu somente sobre o AG. A técnica de busca adotada foi a unimodal, onde um indivíduo é um conjunto de 9 pesos, sendo cada um dos 9 pesos do indivíduo treinados individualmente enquanto os demais são mantidos fixos. A seqüência de codificação do indivíduo (a partir da RMD) será a mesma adotada pelo processamento, da esquerda para a direita e de cima para baixo (figura 3.1). O algoritmo foi denominado de algoritmo de **Treinamento_Unimodal_Direto**, numa alusão ao passo direto de propagação da informação pela rede.

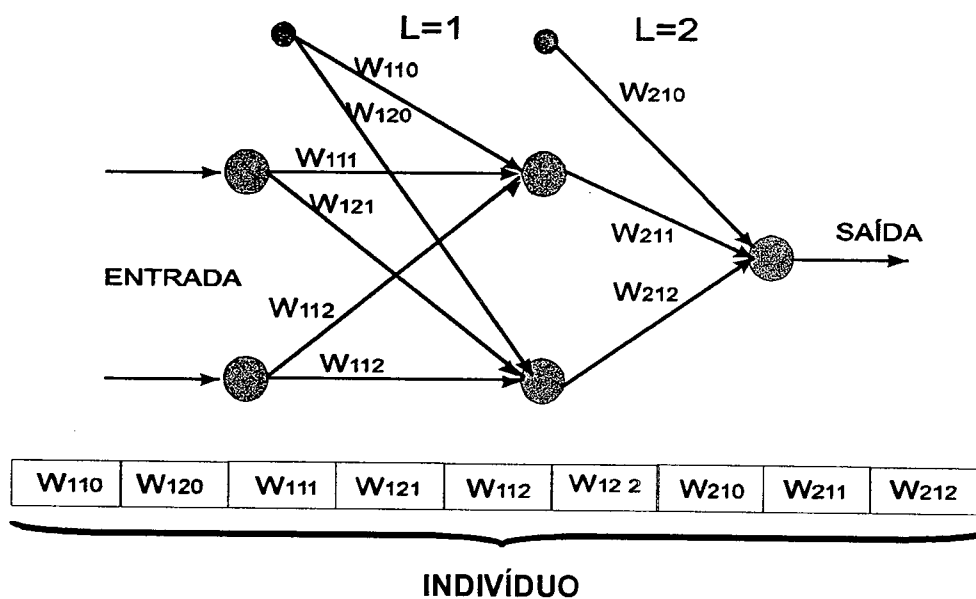


Figura 3.1 Codificação do cromossomo conforme seqüência dos pesos

Inicialmente, para um indivíduo são gerados aleatoriamente os 9 pesos, segundo uma distribuição uniforme dentro do intervalo previamente definido (parâmetros iniciais) para os pesos iniciais. Conforme ilustra a figura 3.1, cada peso do indivíduo é repassado ao AG que na primeira geração, têm os demais elementos da população gerados aleatoriamente. Durante o processamento genético desta população, a avaliação de cada elemento é obtida através do erro da época. Para que isso seja possível, cada elemento da população é devolvido ao indivíduo e a este são apresentados os 4

padrões de treinamento do XOR, de onde obtêm-se o erro da época. Portanto, cada peso é processado geneticamente mas a aptidão é obtida com os demais pesos do indivíduo. O processamento genético da população ocorre até um determinado número de gerações pré-fixadas ou por erro mínimo (erro de época).

Ao final do processamento genético da população para um determinado peso, o melhor elemento encontrado em todas as gerações é passado e mantido fixo no indivíduo. O processo se repete para o próximo peso e assim sucessivamente, até que todos tenham sido processados pelo AG, o que constitui uma etapa. Portanto, uma única etapa consiste no processamento genético de todos os pesos individualmente e durante um número fixo de gerações (ou erro mínimo, conforme critério de convergência). No caso dos 9 pesos da rede 2-2-1 para o XOR, resumidamente poder-se-ia dizer que durante uma etapa haverá a aplicação de 9 algoritmos genéticos; um de cada vez, um para cada peso e segundo a seqüência dos pesos do indivíduo. A melhor proposta de peso encontrado em um certo número de gerações é passado e mantido no indivíduo original. Desta forma, ao final da primeira etapa todos os pesos inicialmente aleatórios, são trocados pelos melhores obtidos por processamento genético.

Neste experimento, cabe ao AG a aproximação da região onde provavelmente localiza-se o mínimo global. Logo, dificilmente haverá convergência durante a fase de processamento genético. Assim, geralmente o processamento genético de cada peso do indivíduo é finalizado pelo número de gerações e a nível de indivíduo, a finalização ocorre por número de etapas. A convergência por erro mínimo é deixada a cargo do algoritmo baseado na técnica do gradiente descendente que neste caso é o algoritmo de RP.

A observação de inúmeras execuções do algoritmo Treinamento_Unimodal_Direto surtiu o efeito esperado. Isso porque durante a atuação do AG sobre cada peso, notou-se que apesar da diversidade dos indivíduos da população presente no AG, o erro da época não se alterava para a maioria dos pesos. Observou-se também que a princípio, dentre os pesos que de alguma forma alteravam o erro da época, aparentemente os da

camada de saída mostraram-se mais significativos. Justamente por estarem em menor número (3 pesos) em relação à camada de entrada, 6 pesos.

b) Busca Unimodal Reversa

Visando analisar o comportamento do erro da época em relação aos pesos da camada de saída, foi implementado um novo algoritmo de treinamento. Enquanto o anterior processava os pesos da entrada para a saída, este faz o inverso; processa os pesos da saída para a entrada. O novo algoritmo denomina-se de **Treinamento_Unimodal_Reverso**. Neste, a única mudança realizada foi o sentido de atuação do algoritmo genético, mantendo-se inalteradas as demais condições, tais como: número de etapas, número de gerações e demais parâmetros de controle.

O comportamento durante a execução do processamento genético pelo novo algoritmo reforçou as observações realizadas em relação ao algoritmo anterior. Isto é, os pesos que quando manipulados produzem alguma alteração no erro da época, parecem situar-se na maioria das vezes na camada de saída. Durante o processamento por várias etapas, notava-se que o peso que mais decrementava o erro da época não era o mesmo. No transcorrer de várias etapas, o peso mais significativo mudava de localização, mas ainda sugerindo que a localização preferencial era a camada de saída dos pesos. O fato do peso mais influente sobre o erro da época mudar de localização é uma condição inerente a um SMI. À medida que se altera a importância de determinado peso, a dos demais também pode alterar-se.

c) Busca Unimodal com Sensibilidade Global

Através da inclusão da AS, é possível determinar a importância de cada peso para o erro da época, num dado momento do treinamento. Assim, implementou-se o algoritmo **Treinamento_Sensibilidade_Global**, o qual avalia a sensibilidade do erro da época em relação a cada peso. Após isso, todos os pesos são processados individualmente pelo AG, conforme a ordenação dada pela AS.

Durante a avaliação com o auxílio da AS, notou-se que o primeiro peso (dado pela AS) geralmente conduz a uma alteração significativa no erro da época. Contudo, o mesmo não pode ser afirmado quanto aos demais pesos. Novamente, o fato do conjunto de pesos da rede comportar-se como um SMI dificulta a avaliação feita de forma global, ou seja, para todos os pesos. Também notou-se que para praticamente metade dos pesos, o processamento genético não promove alterações significativas sobre o erro da época, comportando-se como se estivesse paralisado, logo, confirmam-se as observações anteriores sobre a presença de regiões não promissoras no espaço de busca.

O fato de no mínimo o primeiro peso obtido segundo a AS conduzir a uma alteração significativa no erro da época comprova a viabilidade da aplicação desta técnica. No entanto, ainda persiste o processamento dos pesos que não apresentam viabilidade de redução do erro, naquele momento do treinamento. A diferença, agora, é que sabe-se quais são os pesos menos promissores em um determinado momento. A redução do número de pesos a serem processados em cada etapa implica na redução do espaço de busca; ou seja, limita-se a busca às regiões promissoras. Este é o ponto fundamental da inoperância da técnica convencional do AG sobre redes com grande número de pesos.

d) Busca Unimodal Parcial com Sensibilidade

Já havia sido observado desde o início que parte dos pesos, em determinados momentos do treinamento, não contribuíam para a redução do erro da época. Contudo, a preocupação era de como evitar que estes pesos fossem processados, reduzindo a carga computacional bem como o tempo de treinamento. Assim, implementou-se o algoritmo **Treinamento_Sensibilidade_Parcial**. Este algoritmo é semelhante ao anterior, com exceção de que neste, somente alguns pesos do indivíduo são processados em cada etapa. No início de cada época, é efetuada a AS, ordenando-se os pesos segundo esta. Destes, somente os que

apresentarem maior sensibilidade é que serão processados, ou seja, haverá um treinamento parcial dos pesos.

Na fase de avaliação, esta rotina mostrou-se interessante apesar de ainda depender do usuário para definir o número de pesos a serem processados segundo a seqüência dada pela sensibilidade. A redução no erro é similar à apresentada pelo algoritmo anterior, corroborando as informações da AS só que num tempo menor, já que nem todos os pesos são processados.

Mesmo com os resultados obtidos até o momento, considera-se importante a continuidade das avaliações, tendo em vista que foi possível verificar em alguns momentos a redução na velocidade de convergência, comportamento semelhante ao do algoritmo de RP quando encontra regiões planas ou ponto de sela.

e) Busca Unimodal Parcial com Sensibilidade por Padrão

Antes de apresentar a implementação utilizada nas avaliações, é importante lembrar a base matemática que apóia a técnica a ser mostrada. Assim, Sprinkhuizen-Kuyper e Boers[27,28], conforme descrição realizada no item 1.6 – Superfície de Erro do XOR, afirmam que somente o mínimo global é que apresenta gradiente nulo para cada padrão do conjunto de treinamento. No caso de mínimos locais e ponto de sela, o gradiente apresentado pelo conjunto de treinamento pode ser nulo, mas pelo menos um dos padrões apresenta gradiente diferente de zero.

A aptidão de cada elemento candidato, até o momento, baseia-se exclusivamente no erro da época. Esta forma de cálculo é semelhante à correção em lote do algoritmo de RP, e portanto, é susceptível aos mínimos locais e ponto de sela. Optou-se por estudar uma alternativa que evite esses problemas, tão freqüentes no treinamento de RNAs. Desta forma, alterou-se o algoritmo anterior, dando origem a um novo algoritmo, denominado de **Treinamento_Sensibilidade_Padrão**.

Nesse algoritmo implementaram-se as sugestões para trabalhar com superfícies de erro que apresentem mínimos locais e pontos de sela. Assim,

antes de calcular a sensibilidade dos pesos, o erro de cada padrão é obtido e comparado ao erro da época. Para o padrão que apresentar erro predominante, calcula-se a sensibilidade do erro deste padrão (e não mais do erro da época) em relação a cada um dos 9 pesos.

Apesar da ordenação dos pesos ser realizada segundo o erro do padrão predominante, a aptidão deve ser obtida tanto pelo erro da época quanto pelo erro do padrão em questão. Se a aptidão focar apenas o erro da época, retorna-se ao problema dos mínimos locais e pontos de sela, comentado anteriormente. Porém, se for focado somente o erro do padrão que foi utilizado para ordenar os pesos, então, embora a redução do erro deste padrão durante o treinamento possa ser grande, o erro da época, ao invés de reduzir, pode elevar-se. Isso ocorre porque o erro de um outro padrão pode tornar-se predominante e fazer com que o erro da época, ao invés de diminuir, aumente. A alternativa adotada foi obter a aptidão de cada elemento da população em relação ao erro do padrão predominante e também em relação ao erro da época. Caso o erro da época seja maior do que o inicial (anterior), o elemento é eliminado da próxima população, independente da redução causada no erro do padrão em questão. Assim, apesar da aptidão ser obtida em função de um padrão, a idéia de que o importante é o desempenho perante a época é preservada. As partes podem evoluir enquanto não comprometerem o desempenho do indivíduo, ou melhor ainda, enquanto a evolução das partes contribui para a evolução do indivíduo.

Além dos algoritmos apresentados, vários outros foram implementados. O objetivo destas implementações, dentre outros, era a de definir estratégias, buscando a otimização da técnica de inclusão da AS como forma de guia para a atuação do AG sobre um SMI. Na continuidade deste trabalho, serão apresentados vários outros dados que mostram os resultados obtidos bem como os detalhes observados.

3.3 INSPIRAÇÃO DA PROPOSTA DE BUSCA UNIMODAL

a) Inspiração Biológica

As implementações algorítmicas que visam o treinamento de uma RMD, dividem-se basicamente, em duas partes bem caracterizadas. A primeira é a base desta proposta e fundamenta-se na aplicação do AG. Já a segunda consiste na aplicação de um algoritmo de treinamento, baseado na técnica de gradiente descendente, como o de RP.

No tocante ao AG há plausibilidade biológica, à medida em que a individualidade de cada característica (peso) é garantida, sem que haja perda da informação do desempenho global (indivíduo). Isso é conseguido através do processamento de cada peso separadamente enquanto a aptidão é obtida em função do desempenho do indivíduo, cuja otimização de seu desempenho é o objetivo final.

O cromossomo (indivíduo - conjunto de pesos) neste trabalho, tem o seu menor elemento como sendo uma "string" ao invés de um único "bit". Desta forma, a "string" (peso) pode representar de forma mais adequada determinada característica de um ser vivo. Assim, a abordagem realizada através do AG assemelha-se à diversidade dos tecidos de um ser multicelular. Podendo-se imaginar um coração evoluindo separadamente de um rim, de um fígado, etc., da mesma forma que cada peso da rede deve evoluir individualmente e a seu tempo (relevância).

Exemplificando, se o cromossomo apresentado à direita na figura 3.2 representasse um ser qualquer, cada "string"(peso) seria uma característica deste. Caso a "string" selecionada representasse a cor dos olhos e esta possuísse 8 "bits", ter-se-ia a capacidade de representação de 256 cores diferentes ao invés de ter ou não olhos, caso a representação fosse realizada por apenas um "bit".

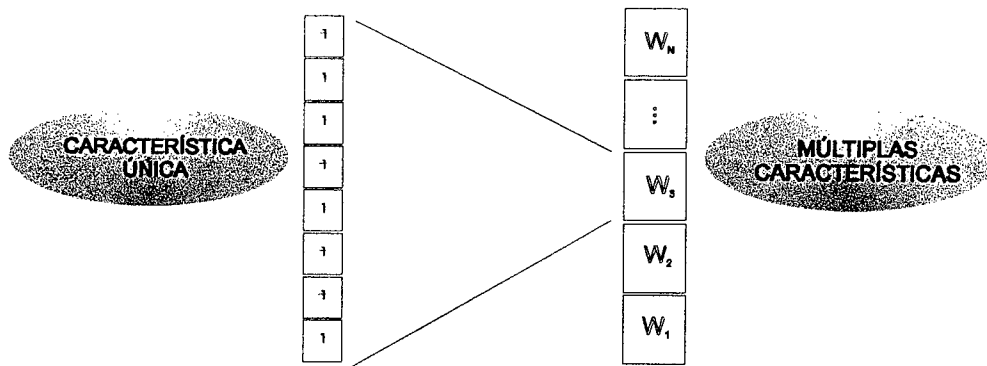


Figura 3.2 Representação do cromossomo e características.

O processo biológico pelo qual passa um indivíduo durante uma vida viável pode a grosso modo, ser dividido em três etapas.

A primeira etapa é o nascimento, onde o futuro indivíduo sofre a primeira grande avaliação. Se as exigências do genótipo em relação ao meio e vice-versa puderem ser atendidas, então o fenótipo é viabilizado.

A partir desse momento, inicia-se a segunda etapa que consiste na maturação do indivíduo, ou seja, é a transição entre o nascimento e o estágio adulto. Durante esta fase ocorre uma intensa interação do indivíduo com o meio (evolução e aprendizado). O desenvolvimento do indivíduo ocorre de forma não homogênea. Cada órgão (por exemplo) se desenvolve a seu tempo e de acordo com as necessidades do indivíduo, naquele instante ou num tempo próximo. Enquanto uns desenvolvem-se muito no princípio, outros, somente algum tempo e após certas condições. No entanto, todos visam aumentar a aptidão do indivíduo. Portanto, não podem buscar o seu próprio ótimo (ótimo local) e sim o ótimo do indivíduo (ótimo global). Similarmente também se comportam os pesos de uma RNA.

Nesta fase, a aplicação da AS destina-se a avaliar o indivíduo em torno dele; esta informação não é repassada ao código genético. Portanto, a AS funciona como o aprendizado Baldwiniano, onde a informação da capacidade que o indivíduo tem de melhorar é que é utilizada para melhorar a sua aptidão.

A terceira etapa consiste na fase adulta do indivíduo. Este será submetido ao processo de seleção e, dependendo de sua aptidão, poderá ocorrer o cruzamento e portanto a manutenção do seu patrimônio genético.

O processo descrito acima é o mesmo adotado no algoritmo `Treinamento_Sensibilidade_Padrão`, onde na fase de maturação cada peso evolui de forma relativamente independente. Assim, a aplicação da AS visa descobrir quais dos pesos devem ser desenvolvidos.

b) Inspiração Matemática

A busca unimodal também mostra-se como uma alternativa matematicamente plausível, se a AS for calculada para o padrão que apresentar o gradiente diferente de zero. No caso de haver mais de um padrão com gradiente não nulo, deve-se escolher o que apresentar maior contribuição para o erro da época.

A avaliação de um peso deve considerar a redução do erro do padrão segundo o qual o peso foi ordenado pela AS, bem como o erro da época. Conforme descrito anteriormente, caso a avaliação seja restrita ao erro da época, poderá ocorrer o mesmo problema de paralisia apresentado pelo algoritmo de RP. Já o motivo para não ignorar o erro da época consiste em que há pesos que levam à redução do erro de um padrão específico, mas a partir de um certo ponto, elevam o erro global (da época). Isso ocorre porque um peso não está relacionado somente a um padrão e são interdependentes. Logo, espera-se que o conhecimento esteja distribuído na rede. Esta característica da RNA denomina-se de tolerância a falhas. Resumindo, tem de haver um compromisso entre ambos, o erro do padrão e o erro da época.

A justificativa matemática para esta técnica tem suporte principalmente nos resultados das pesquisas de [24,26,27], apresentadas no item 1.6 – Superfície de Erro do XOR. Conforme estes trabalhos, o mínimo local bem como o ponto de sela são pontos estacionários instáveis e portanto, apresentam gradiente igual a zero para o erro da época. Contudo, no mínimo para um dos padrões apresentam o gradiente diferente de zero.

Outro ponto que matematicamente torna interessante a busca unimodal guiada pela AS é a redução do espaço de busca. O tamanho do espaço de busca, aliado à falta de informação da relevância das regiões deste espaço são os maiores problemas da técnica convencional. Há uma degradação do desempenho à medida que cresce o número de pesos da rede. O acréscimo no número de pesos significa um aumento do espaço de busca. Portanto, a aplicação da AS para identificar as regiões promissoras, deve levar a uma redução deste espaço.

4 RESULTADOS

Neste capítulo apresenta-se uma série de observações e experimentos realizados visando aumentar a compreensão dos problemas realmente existentes no treinamento híbrido de RNA, através de um algoritmo baseado na técnica de GD e do AG.

A exposição a seguir, na medida do possível, seguirá a seqüência apresentada nos objetivos específicos do presente trabalho, item 1.2.2.

4.1 REPRESENTATIVIDADE DA FUNÇÃO CUSTO

A base desta proposta, conforme apresentado no capítulo anterior, se pauta na baixa representatividade da função custo comumente adotada pela técnica convencional. Desta forma, afirma-se que quanto mais complexa a RNA (principalmente no número de conexões), menor será a representatividade da função custo.

Sugere-se como motivo para isso o fato de que a função custo é representativa do indivíduo, mas não de cada parte deste, ou seja, a função mede o desempenho do indivíduo, mas processa regiões deste sem analisar sua viabilidade. Com isso, o AG irá operar quase as cegas, sem um guia, permitindo desta forma que elementos pouco representativos sejam processados, ao invés de ajustar os pesos que mais contribuem para o erro do conjunto de treinamento. Este processamento pode elevar o tempo de treinamento, conforme o número de pesos que compõe o indivíduo. Assim, ter-se-á uma atuação tão menos otimizada quanto maior for o número de pesos da rede em treinamento.

Na fase de atuação do algoritmo baseado no GD, deve-se tomar cuidado com a relação entre a função de ativação e a função custo adotada. Por exemplo, se for utilizada a função de ativação tangente hiperbólica,

recomenda-se o emprego da função custo da entropia (equação 27), a qual faz com que a superfície de erros aproxime-se mais de uma parábola. Este fato facilita a operação dos algoritmos baseados em GD. A utilização da função custo quadrática (equação 3) torna a superfície de erro quase plana, o que dificulta muito o trabalho dos algoritmos baseados em GD, principalmente do algoritmo de RP. O desenvolvimento e comentários sobre o algoritmo de RP com função de ativação da tangente hiperbólica e função custo da entropia pode ser encontrado em Svarer[98].

$$\frac{1}{2} \sum_{i=1}^{N_0} \left(\frac{1}{2} (1 + d_i^p) \cdot \log \frac{1 + d_i^p}{1 + o_i^p} + \frac{1}{2} (1 - d_i^p) \cdot \log \frac{1 - d_i^p}{1 - o_i^p} \right) \quad (27)$$

Onde: E_p é o erro da rede para o padrão p ; d_i^p é a saída desejada no neurônio i para o padrão p ; o_i^p é a saída obtida no neurônio i para o padrão p .

Portanto, o desempenho do AG é sensível à função custo e a técnica baseada em GD é sensível à combinação entre a função de ativação e a função custo adotada.

4.2 A ANÁLISE DE SENSIBILIDADE E A DISTRIBUIÇÃO DO CONHECIMENTO SOBRE OS PESOS DA REDE

A seguir mostram-se as avaliações realizadas sobre a distribuição do conhecimento nos pesos da rede neural, tendo os seguintes objetivos básicos:

- a) Avaliar a importância da AS para melhorar a compreensão da distribuição de conhecimento sob a RNA;
- b) Avaliar o desempenho da VP em relação ao GD;
- c) Verificar se há alguma tendência de localização dos pesos que apresentam maior SEP;
- d) Comparar a seqüência de sensibilidade obtida em função do padrão que apresentar maior contribuição de erro com a seqüência obtida em relação ao erro da época.

A aplicação da AS clássica (item “a”) mostrará a viabilidade e o potencial desta técnica quando aplicado a RNA. O resultado obtido no item “a” servirá de base para a avaliação e validação da proposta alternativa de obtenção da AS através da variação dos pesos (item “b”). O planejamento deste experimento foi realizado de tal forma que a AS seja aplicada sobre cada peso da rede segundo o erro da época bem como sobre o erro do padrão predominante naquele momento do cálculo. Desta forma, busca-se identificar e avaliar se há alguma tendência na localização da SEP sobre os pesos da rede (item “c”). Uma comparação do resultado das duas formas de obtenção da SEP mostrará se o erro do padrão mais significativo é representativo ou não para a rede (item “d”).

4.2.1 Delineamento do Experimento

A fase de treinamento da RNA foi também a fase observacional onde percebeu-se, qualitativamente, a relação entre pesos e camadas.

Nesta fase, conforme comentado anteriormente, percebeu-se que no caso do processamento dos pesos por AG, alguns pesos apesar de estarem sendo variados pelo AG, não resultavam em correspondente variação no erro da época. Também foi possível observar que os pesos que mais contribuíam para a variação do erro da época aparentemente localizavam-se em certas regiões da rede, que, no caso do XOR, pareciam ser a camada de saída dos pesos.

Para avaliar objetivamente esta hipótese, delineou-se um experimento com as seguintes características:

- a) A resposta do sistema foi definida como sendo o erro da época (erro médio quadrático) da RNA;
- b) Os fatores foram definidos como sendo o gradiente descendente (GD) e a variação dos pesos (VP);
- c) Os pesos iniciais foram gerados aleatoriamente segundo uma distribuição uniforme [-5; 5], utilizando a função rnd do Visual Basic 3.0.

Na análise da sensibilidade foi analisada a distribuição dos resíduos.

4.2.2 Amostra

Adotando-se um nível de confiança de 95 % para os parâmetros do modelo, definiu-se uma amostra de 50 casos de treinamento. Cada caso é formado por um conjunto de 9 pesos gerados aleatoriamente, segundo uma distribuição uniforme no intervalo de -5 a $+5$.

Cada conjunto de pesos é processado tanto pelo GD quanto pela VP. No caso do GD, a sensibilidade de cada peso é obtida em função do erro da época. Já para a VP, a sensibilidade de cada peso é obtida em função do erro do padrão que apresentar a maior contribuição de erro no conjunto de treinamento. Portanto, antes de aplicar a VP, deve-se apresentar os quatro padrões de treinamento do XOR e determinar qual destes apresenta a maior contribuição para o erro. A sensibilidade de cada peso é obtida em função deste padrão com maior contribuição.

A iniciativa de propor a VP como forma de obter a sensibilidade, deve-se à existência de aplicações onde não é possível a aplicação da AS formal (equações 23 a 25). Isso pode ocorrer onde a função de ativação não é derivável ou o cálculo da derivada eleve a carga computacional a níveis inviáveis. O único cuidado a ser tomado é que a variação em torno do valor original é função da superfície de erro. Assim, se a superfície for muito convoluída, esta variação deverá ser menor do que os 10% utilizados aqui. Outra vantagem da VP é sua facilidade de implementação.

A motivação em comparar a AS obtida em função do erro da época, com a AS obtida em função do padrão mais significativo para o erro da época é de analisar a representatividade desse padrão mais significativo. Se não houver um padrão predominante em termos de erro, a AS com o auxílio da VP deve ser obtida em função do erro da época.

Estes dados foram dispostos em uma base de dados e posteriormente processados com o auxílio do software Statistica 5.0¹⁰.

¹⁰ Marca registrada da Statsoft.

O histograma da figura 4.1 apresenta a distribuição da sensibilidade dos pesos com o objetivo de mostrar que esta distribuição é fortemente assimétrica, o que levou à transformação logarítmica dos pesos para a análise estatística dos dados.

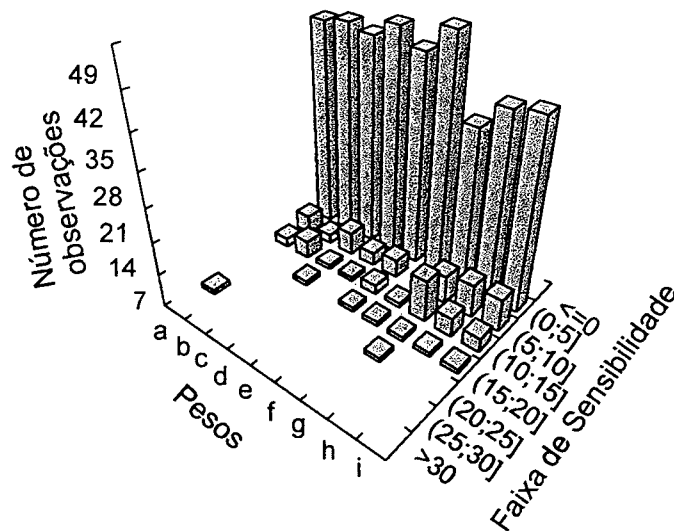


Figura 4.1 Histograma da distribuição dos pesos

4.2.3 Resultados

A tabela 7 mostra os valores de média e desvio padrão dos pesos. Onde se evidencia que os pesos da camada de saída tendem a ser mais significativos para o treinamento do que os pesos da camada de entrada. Na figura 4.2a e 4.2b, apresenta-se o intervalo de confiança para a média dos pesos, sendo que as ligações de a a f são relativas à camada de entrada e as ligações g, h, i definem a camada de saída. Pode-se concluir que os pesos da camada de saída são, em média, mais significativos do que os pesos da camada de entrada, para um nível de confiança de 95 %. Tanto a AS obtida por GD quanto a por VP apresentaram a mesma tendência, conforme pode ser visto nas figuras 4.2a e 4.2b.

Tabela 7. Valores de média e desvio padrão para o gradiente descendente e a variação dos pesos

Camada de Pesos	Peso	GD		VP	
		Média Da Sensibilidade	Desvio Padrão da Sensibilidade	Média da Sensibilidade	Desvio Padrão da Sensibilidade
E N T R A D A	a	-.444846	2.106669	.923165	2.045117
	b	-.938356	2.203141	.991512	3.011761
	c	-.661910	2.272157	.733574	1.456438
	d	-.981247	2.151082	.812519	1.969768
	e	-.438740	2.002066	1.246394	2.837439
	f	-1.28863	2.402584	.990088	2.316545
S A Í D A	g	.739219	1.488601	2.958941	6.313424
	h	.473403	1.526604	2.539462	4.037394
	i	.154026	1.667432	1.880912	3.837216

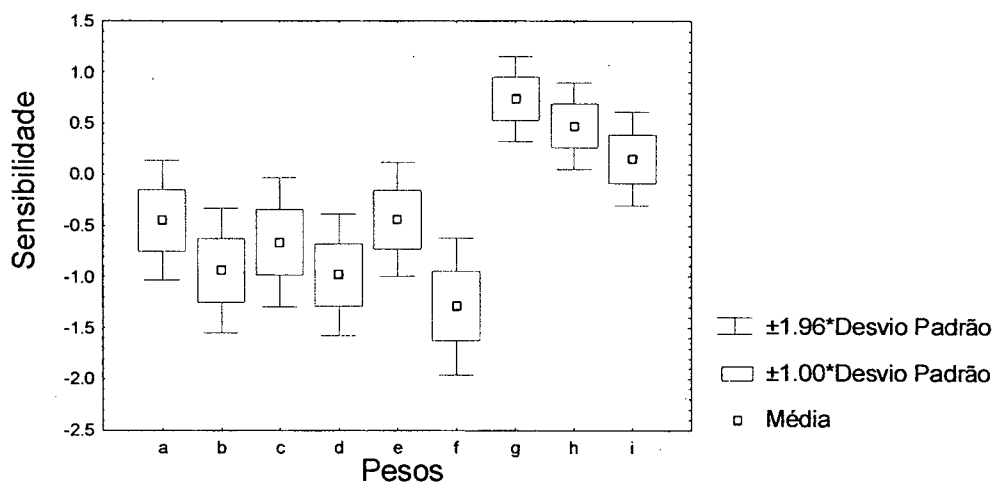


Figura 4.2a Intervalo de confiança para a sensibilidade dos pesos segundo o Gradiente Descendente

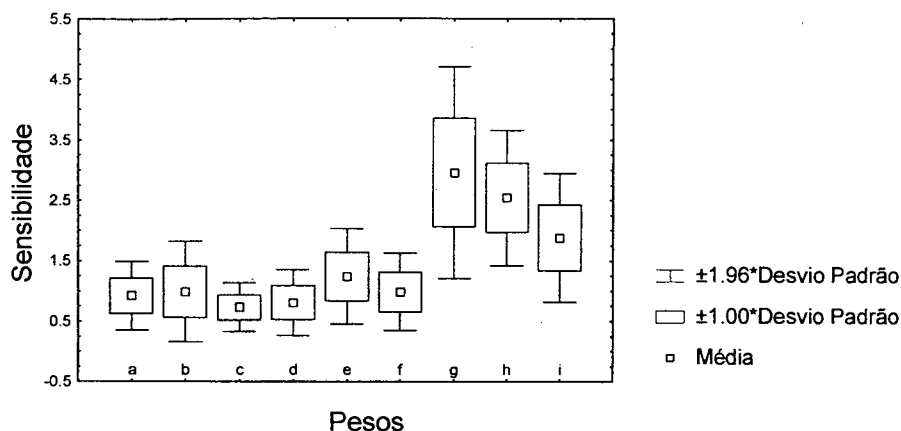


Figura 4.2b Intervalo de confiança para a sensibilidade dos pesos segundo a Variação dos Pesos

4.2.4 Conclusões do experimento

Este experimento foi realizado sobre os pesos obtidos de forma aleatória para evitar as influências do treinamento sobre a distribuição da sensibilidade.

De uma forma geral, os objetivos estabelecidos para este experimento (itens de “a” a “d”) foram alcançados e confirmaram as suspeitas levantadas durante a fase observacional. A AS mostrou ser uma ferramenta altamente importante na análise de RNA tendo em vista a sua capacidade de colocar em evidência aspectos que encontram-se escondidos na complexidade de sua estrutura.

Os resultados obtidos não podem ser diretamente generalizados para outro domínio que não seja o XOR. No entanto, há viabilidade do emprego da AS como ferramenta de análise de RNAs, o que geralmente é feito somente pelo conjunto de padrões de teste.

4.3 TRANSIÇÃO ENTRE O ALGORITMO GENÉTICO E O ALGORITMO BASEADO NO GRADIENTE DESCENDENTE

Um ponto relevante quando se tem um método híbrido é quando deve ocorrer a transição de uma técnica para outra. Baseado no fato do AG ser o encarregado da aproximação da região onde provavelmente localiza-se o mínimo global e não da finalização (ajuste fino), presume-se que a transição

geralmente não ocorrerá pelo critério do erro mínimo. Aliado a este fato, tem-se também a necessidade de uma exploração eficiente do espaço de busca, em um curto espaço de tempo.

Provavelmente, a região de transição do AG para o algoritmo baseado no GD mudará de um domínio de aplicação para outro. Contudo, dentro de um mesmo domínio, a variabilidade do momento da transição, nas avaliações efetuadas, mostrou-se pequena. Empiricamente, nota-se uma região com maior potencial de transição, a ponto de permitir a concepção de uma metodologia que determine pelo menos, uma região onde a transição traria mais benefícios do que transtornos. A vantagem da determinação consiste na redução do tempo de treinamento. Se a transição for realizada antes do tempo, o GD pode não contar com a necessária contribuição do AG. Já se a transição ocorrer muito depois do ponto ideal, haverá perda de tempo computacional, tendo em vista que o GD é mais rápido do que o AG, principalmente na fase final de convergência.

O momento da transição entre o AG e o algoritmo baseado no GD, para um mesmo domínio de aplicação, mostrou um comportamento sistemático, considerando um conjunto de parâmetros de controle constante nas várias execuções observadas. A metodologia proposta a seguir requer uma elevada carga computacional. Portanto, a aplicação desta metodologia será restrita, devido à viabilidade, aos casos onde houver necessidade de treinar várias RMD com a mesma topologia e domínio de aplicação.

O primeiro passo é a inicialização dos parâmetros de controle tanto para o AG quanto para o algoritmo baseado na técnica de GD a ser utilizado. Estes parâmetros serão mantidos constantes em todas as execuções.

A proposta de metodologia apresentada aqui consiste na obtenção de uma amostra de n casos, sendo cada caso composto por tantos indivíduos quantos forem as gerações necessárias ao AG, para levar o conjunto de pesos aleatórios até as proximidades da convergência final. O melhor indivíduo gerado na população inicial, é memorizado como sendo o indivíduo 0 (zero) do respectivo caso. Após isso, o AG irá processar esta população até uma região próxima da convergência (erro final aceitável). Este

processamento acontece sem a participação de qualquer outro algoritmo de treinamento. Ao final de cada geração, o melhor indivíduo (conjunto de pesos), é memorizado junto com a sua aptidão, gerando os casos de “1” a “m”, conforme figura 4.3.

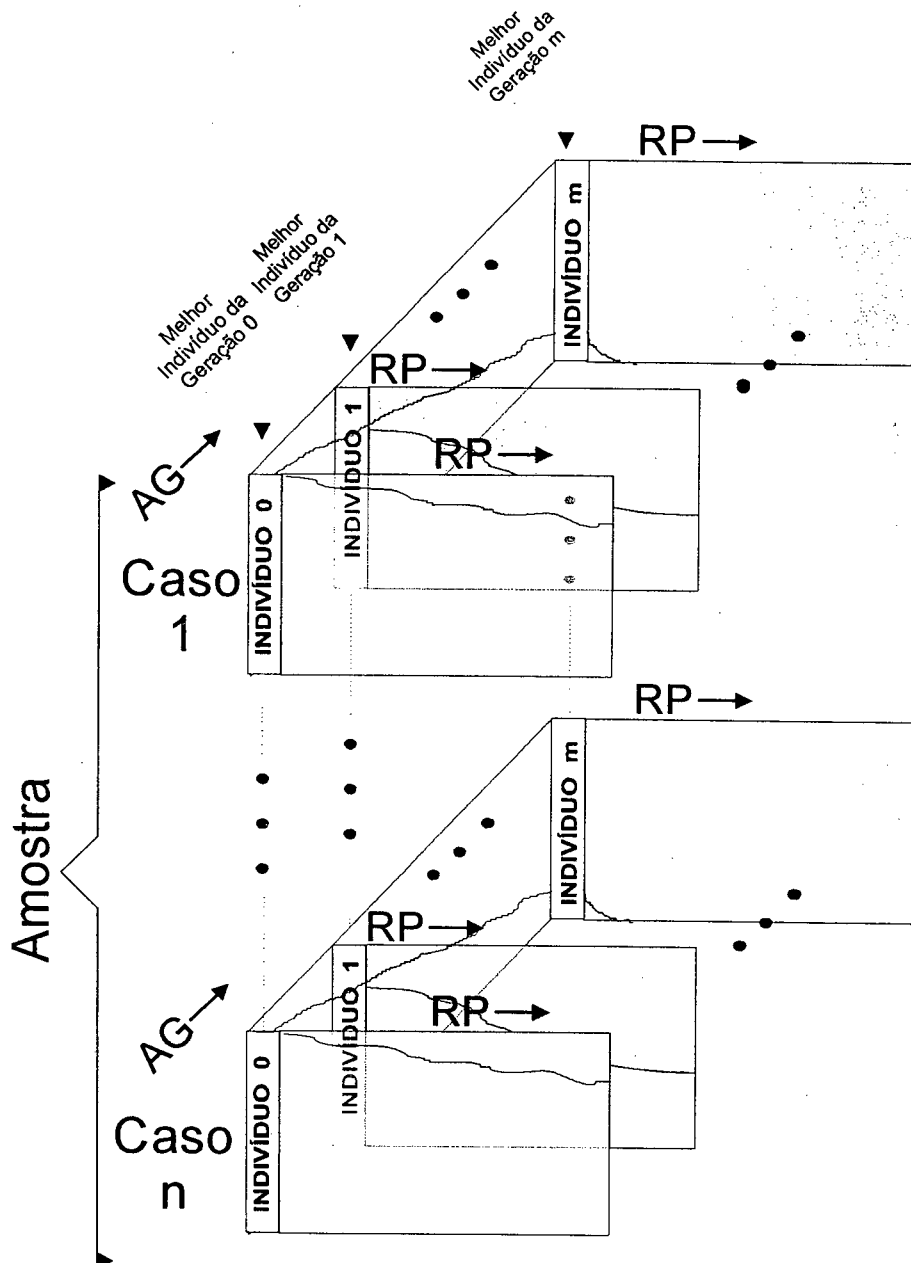


Figura 4.3 Detalhamento da metodologia de transição do Algoritmo Genético para o Algoritmo de Retropropagação

Finalizado a obtenção da amostra de casos, inicia-se o processamento pelo algoritmo baseado no GD. Este deve processar cada indivíduo memorizado até a convergência. Ao final, deve aplicar a base de teste para avaliar a qualidade do treinamento obtido. A capacidade de generalização é um dos parâmetros mais expressivos neste momento; se não for aceitável, o indivíduo é descartado da amostra.

Tendo realizado o treinamento com o algoritmo baseado no GD para cada conjunto de pesos memorizado, pode-se analisar não só o resultado final, mas a facilidade com que o GD chegou a este. Resumindo, a amostra é composta por um conjunto de casos. Cada caso é composto pelo melhor indivíduo de cada geração do AG que processará uma população, gerada aleatoriamente, até próximo da convergência. Ao final de cada geração, o melhor indivíduo irá compor um caso.

Na metodologia apresentada, há vários pontos importantes a serem observados:

- a) As chances de paralisia dos algoritmos baseados em GD, quando processando os conjuntos de pesos situados na região inicial, são grandes. Isso pode ocorrer devido aos valores elevados dos pesos gerados de forma aleatória para o AG. Portanto, sugere-se incluir no algoritmo baseado no GD, uma finalização por número de épocas que será a saída se não houver convergência ou se houver estagnação da convergência;
- b) Em dois ou mais casos consecutivos, pode haver duplicação de indivíduos. Isso pode ocorrer quando em gerações consecutivas, não for encontrado um indivíduo melhor do que o anterior e como a população é elitista, o melhor da geração anterior também será o melhor da atual. Portanto, processá-lo novamente com o algoritmo baseado no GD é perda de tempo;
- c) Sugere-se, para efeito de redução da carga computacional, que nem todos os indivíduos de cada caso sejam processados pelo algoritmo baseado em GD. O acréscimo de informação advindo do processamento de todos os indivíduos de cada caso, pode ser tão pequeno que não justifique tal

atitude. Assim, uma sugestão mais geral, seria a de que na primeira execução do algoritmo baseado no GD, este fosse realizado com um espaçamento grande entre os indivíduos. Após, executa-se outro processamento na região mais promissora, reduzindo o espaçamento para adquirir detalhes mais específicos, se este for o caso.

4.4 CRUZAMENTO UNIFORME E MUTAÇÃO: O QUE HÁ DE COMUM?

Se a preocupação deste trabalho fosse resolver um problema específico, caberia uma comparação de desempenho entre os métodos de cruzamento. No entanto, o objetivo maior são os levantamentos de heurísticas e informações comportamentais sobre os métodos. Além do mais, há vários artigos retratando o desempenho comparativo do cruzamento uniforme com o cruzamento de 1 e 2 partições, e destes com a mutação [60,64,67].

O trabalho de Spears e Anand [64] apresenta uma análise interessante para o presente trabalho. A abordagem realizada abrangeu tanto uma população considerada pequena (20 indivíduos), quanto uma população considerada grande (100 indivíduos). Nas figuras 4.4 e 4.5 apresentam-se as conclusões obtidas para o domínio da aplicação do caixeiro viajante. Note que, para uma população pequena, o desempenho da mutação aproximou-se do cruzamento (figura 4.4). Já para uma população maior, o desempenho do cruzamento foi dominante sobre o da mutação, figura 4.5. Conforme Spears e De Jong [60], populações pequenas tendem a convergir mais rapidamente para níveis de homogeneidade, o que reduz a produtividade do cruzamento e tende a levar a convergência para mínimos locais. Logo, concluem os autores, para pequenas populações uma forma de cruzamento mais disruptivo como o uniforme deve obter melhores resultados.

Torna-se importante lembrar que a diferença básica entre o operador de mutação e o de cruzamento é que o cruzamento trabalha com o material genético existente no patrimônio genético da população, enquanto que a mutação pode repor o material perdido ou introduzir material novo.

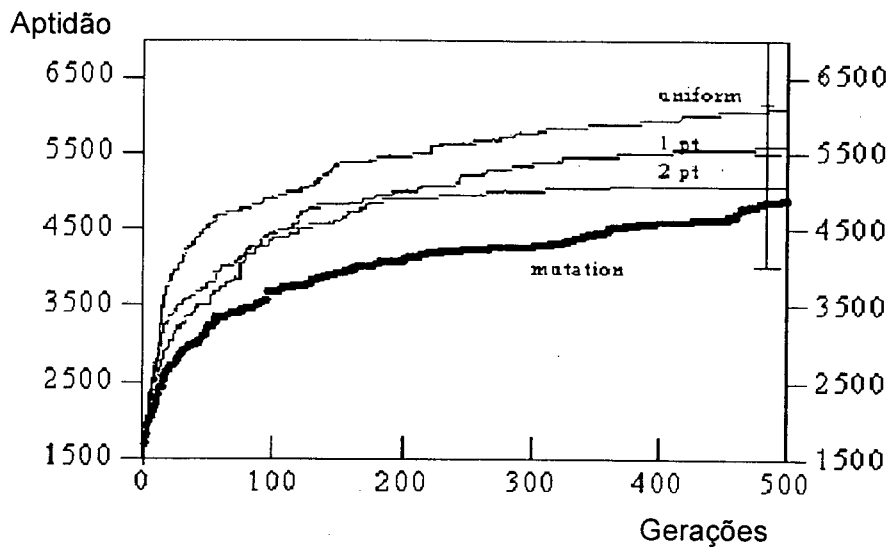


Figura 4.4 - Desempenho comparativo do cruzamento e mutação para uma população de 20 indivíduos [64]

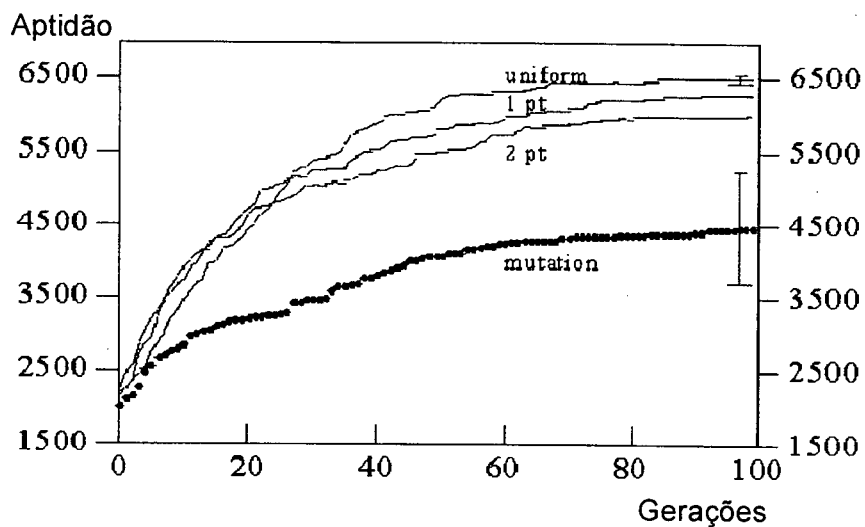


Figura 4.5 - Desempenho comparativo do cruzamento e mutação para uma população de 100 indivíduos [64]

A princípio, neste trabalho, propõe-se que em termos de resultado evolutivo, o cruzamento uniforme é um caso particular da mutação, sob o ponto de vista do grau de adaptação que a população pode atingir em um determinado tempo (número de gerações). Se para um casal selecionado na

população, for aplicado a mutação em um deles mas a troca de material genético for permitida somente se o novo material existir no outro indivíduo, ter-se-á o cruzamento uniforme. Portanto, extrapolando para o nível da população, o cruzamento uniforme é tanto mais disruptivo quanto maior for a diversidade da população.

Uma forma de comparar o desempenho do cruzamento uniforme com o da mutação seria duplicar uma população e enquanto uma das populações resultantes seria processada por cruzamento uniforme, a outra o seria por mutação. Nenhum outro operador deveria atuar em cada população (com exceção da seleção) para poder comparãr os resultados finais. A única modificação seria que na população processada pela mutação, não seria permitido ao operador de mutação repor material genético, ou seja, antes de aceitar a troca, deve-se procurar na população para ver se este material já existe. Se o material não existir, deve-se ignorar a troca. Assim, a mutação deverá apresentar a mesma capacidade de geração ou de destruição do patrimônio genético. A divergência encontrada pode ser atribuída à probabilidade adotada para cada operador. Um segundo teste poderia ser feito visando encontrar a probabilidade segundo a qual a aplicação da mutação, com as restrições acima, apresentariam o mesmo desempenho do cruzamento uniforme.

4.5 ASPECTOS GERAIS DA TÉCNICA HÍBRIDA

O sucesso do AG, independente da forma como for utilizado, depende, dentre outros fatores, do tamanho e diversidade da população. Quanto ao tamanho da população, sob o ponto de vista da busca, vale o “quanto maior melhor”. No entanto, sob o ponto de vista da carga computacional é justamente o inverso. Portanto, existe uma relação entre o tamanho do espaço de busca e o tamanho da população.

A técnica convencional de aplicação do AG, por atuar sobre todos os pesos ao mesmo tempo, apresenta um espaço de busca amplo e geralmente complexo. Já a aplicação da técnica unimodal, limita o espaço de busca

momentaneamente ao espaço de busca do peso que estiver sendo processado pelo AG naquele momento.

Portanto, para um mesmo domínio da aplicação a técnica convencional necessita de uma população maior do que a técnica unimodal. Já a diversidade, apesar de necessária em ambas, é mais crítica na técnica unimodal. Neste caso, o processo de busca deve ser o mais abrangente possível, permitindo com isso uma busca rápida e efetiva de uma solução que satisfaça as necessidades da otimização. Chama-se a atenção para a utilização da RoIP como técnica de seleção, ou de uma outra técnica que não force uma perda excessiva da diversidade. Uma leitura mais aprofundada sobre outras técnicas de seleção pode ser vista em [99-102], com ênfase especial no trabalho de Blicke [103] que apresenta as principais técnicas.

Caso não haja uma boa diversidade da população, não haverá grande contribuição a cada geração. Enquanto na técnica convencional o usuário pode avaliar o desempenho do treinamento no final de uma geração, na técnica unimodal isso se dá ao final de uma etapa.

Na técnica convencional, devido ao grande espaço de busca, recomenda-se que a população geralmente não seja inferior a 30 indivíduos[49,52,70]. Já para as implementações efetuadas, optou-se por uma população bem menor do que a convencional. A opção por uma população pequena, aparentemente se contrapondo a uma busca efetiva, deve-se à necessidade de repetir o treinamento de cada peso por várias etapas¹¹. As etapas são necessárias, tendo em vista que o conjunto de pesos é interdependente. Caso o tamanho da população fosse grande, a necessidade de execução do treinamento por várias etapas poderia inviabilizar o método, devido ao tempo de processamento. Nos testes executados, para uma população acima de 10 indivíduos, não foi perceptível melhora no desempenho final do AG. Isso é válido para uma superfície de erro razoavelmente bem comportada, pouco convoluída. Caso a superfície

¹¹ Uma etapa consiste do treinamento de todos os pesos isoladamente, similar à época no RP.

de erro seja altamente convoluída, torna-se necessário aumentar o tamanho da população, ou para um mesmo tamanho de população, aumentar o número de gerações e manter uma boa diversidade. Valores pequenos para a população e com alta diversidade (mantida pelos OAGs) são possíveis, mas é necessário utilizar a população elitista. A população elitista garante que o melhor indivíduo seja mantido na população seguinte, já que há poucas gerações disponíveis para avaliar a região de interesse.

Visando aproveitar ainda mais a população elitista, inicialmente pensava-se que a eficácia seria maior se a reposição dos pesos fosse realizada (segundo alguma distribuição) em torno do melhor indivíduo obtido na etapa anterior. Assim, testaram-se várias distribuições, tais como: distribuição uniforme na faixa pré-definida para os pesos iniciais; distribuição gaussiana em torno do peso existente (melhor da etapa anterior) e também segundo uma distribuição uniforme em torno do peso existente.

Nos testes efetuados, não se obteve bons resultados com esta estratégia de reposição. Havia uma tendência em tornar os valores dos pesos crescentes a cada etapa. Logo, quando o algoritmo de RP recebia estes valores, geralmente a convergência paralisava devido à saturação da função de ativação dos neurônios.

Assim, resolveu-se abandonar este método e continuou-se a utilizar a reposição aleatória com distribuição uniforme no intervalo definido inicialmente.

Quanto aos operadores genéticos, foi considerada a taxa de cruzamento com 1-partição e a mutação, tendo em vista a composição elitista da população. Desta forma, o melhor a fazer é aumentar a taxa de mutação e manter a de cruzamento, isso em relação à comumente adotada. Assim, geralmente a taxa de cruzamento ficou em 70 % e a de mutação variou de 4 a 10 %. Os resultados apresentados aqui foram baseados no operador genético de cruzamento com 1-partição. Contudo, os testes efetuados com o método de cruzamento do tipo uniforme apresentaram bom desempenho, comprovando o que alguns pesquisadores já haviam relatado sobre o

desempenho deste com baixo número de indivíduos na população, que é o presente caso [61,64,65].

Um dos principais parâmetros a ser observado nas implementações é o número de etapas segundo a qual o indivíduo terá seus pesos processados individualmente. Quanto maior o número de etapas, maior será o tempo de processamento, mas, em contrapartida, maior será a abrangência da busca. Geralmente, há necessidade de no mínimo duas etapas, sendo que o número de etapas depende muito do número de pesos que serão processados. Como o número de pesos promissores para o domínio da aplicação do XOR mostrou ser inferior à metade dos pesos do indivíduo, foi utilizado o processamento para 2 ou 3 pesos durante cada etapa. O número de etapas para esta condição foi de 3 ou 4. Estes valores são fortemente dependentes do domínio da aplicação.

O intervalo no qual foram gerados aleatoriamente os pesos iniciais é de grande importância. Principalmente porque se contrapõe às heurísticas adotadas pelos treinamentos convencionais, como o algoritmo de RP. Neste caso, a literatura técnica recomenda iniciar os pesos com valores pequenos, levando-se em consideração os valores de entrada dos padrões de treinamento ($\pm 0,1$, por exemplo) [21]. Em contraposição, no caso do AG recomenda-se iniciar com valores altos. Os “valores altos” são uma função do problema. Entretanto, valores no intervalo de $[-4; +4]$ são uma opção adequada, quando não se tem conhecimento do comportamento final dos pesos [13]. O fato pelo qual recomenda-se usar valores iniciais baixos no algoritmo de RP é a possibilidade de saturação da função de ativação, problema este que é fortemente atenuado no AG. Isto porque no AG a informação do passo da mudança é dado pela aptidão (“rank”), e não pela derivada como no gradiente descendente.

4.6 ALGORITMO DE RETROPROPAGAÇÃO E QUICKPROP

A preocupação deste trabalho é de estudar a proposta híbrida principalmente no tocante a aplicação do AG o qual, conforme descrito, considera-se o ponto frágil da referida proposta. Contudo, nos estudos efetuados, o algoritmo de RP mostrou-se não ser o mais adequado para operar em

conjunto com o AG. Isso se deve a dificuldade que o algoritmo de RP apresenta em trabalhar na região próxima da saturação da função de ativação. Esse problema é agravado pela facilidade com que o AG pode gerar e passar pesos com valores elevados ao algoritmo de RP.

O AG tem facilidade em extrapolar o intervalo inicial dos pesos no caso de um sistema com uma única variável. Esta característica permite a adoção de um intervalo inicial pequeno ($[-1; +1]$ por exemplo), tendo em vista a capacidade do AG em buscar outras regiões à medida que sejam necessárias. Já no caso de sistemas multivariáveis, o AG apresenta dificuldades em extrapolar esse intervalo. Isso implica na necessidade de que o intervalo inicial dos pesos, preferencialmente, englobe os valores desejados dos pesos ou que no mínimo esteja o mais próximo deste o possível. Uma forma prática para avaliar o intervalo inicial dos pesos é treinar a rede por qualquer processo até a convergência e observar o intervalo dos pesos obtidos no treinamento. Em seguida, esta informação é utilizada para definir o intervalo nos quais os pesos são gerados aleatoriamente para o AG. Talvez tenha sido esta observação que levou Kitano [13] em 1990 a propor o intervalo inicial de -4 a $+4$ para treinamento de RNA com AG. Desta forma, o AG já inicia a busca com valores¹² que por si só já levariam o algoritmo de RP à paralisia por saturação. Além disso, observou-se também que o AG geralmente passa ao algoritmo de RP pesos com valores elevados, considerando a sua capacidade de processamento de valores superiores a 3 (devido a derivada da função de ativação).

Nos artigos consultados e que abordam a técnica híbrida, não foi encontrado nenhuma análise quanto aos problemas de paralisia do algoritmo de RP, devido a qualidade dos pesos passados pelo AG. Caso não haja preocupação com a qualidade dos pesos no momento da transição, pode-se estar trocando a fonte da paralisia do algoritmo de RP. Ao invés da paralisia devido à superfície de erro, haveria paralisia causada pelos pesos passados pelo AG. Desta forma, a junção das duas técnicas não se apresenta como uma ferramenta otimizada.

Assim, ao invés de alterar ainda mais a atuação do AG, buscou-se uma alternativa mais prática, a de avaliar outros algoritmos baseados no gradiente descendente. Inicialmente, a avaliação restringiu-se à viabilidade prática. Conforme descrito no item 2.2.4 – Algoritmo de Treinamento, os algoritmos de segunda ordem, por apresentarem alta complexidade de implementação, não foram considerados, apesar de atenuarem o problema em questão. A solução adotada foi a troca do algoritmo de RP pelo Quickprop [14,36]. O algoritmo Quickprop é uma aproximação dos algoritmos de segunda ordem e é tão fácil de implementar quanto o algoritmo de RP.

Como já era esperado, o Quickprop apresentou-se bem mais robusto no processamento de pesos com valores elevados do que o algoritmo de RP, isso sem levar em conta o desempenho superior em velocidade. Atualmente, já é comum o emprego do Quickprop em detrimento do algoritmo de RP, mas não nas técnicas híbridas onde espera-se que também haja mudanças. O desempenho superior, apesar dos valores altos dos pesos, pode ser creditado ao denominador do segundo termo presente na equação 10. Neste, há uma subtração entre a alteração anterior e a atual. Uma pequena diferença nesta subtração promove um aumento do passo. Portanto, ao invés de paralisar a convergência próximo de ocorrer uma saturação, este termo força um passo maior, na expectativa de fugir da situação apresentada no momento.

¹² Considerando valores de entrada binários.

5 CONCLUSÕES

Este capítulo tem por objetivo apresentar as principais conclusões obtidas bem como uma lista de sugestões para futuros trabalhos que de alguma forma ampliariam o conhecimento aqui gerado.

5.1 CONCLUSÕES

Neste trabalho, foi abordado um número elevado de pontos relevantes ao treinamento híbrido de RNA com um algoritmo baseado no gradiente descendente e algoritmo genético. A opção por esta gama de pontos estudados, em parte, deve-se à interligação existente entre eles.

Apesar da literatura técnica demonstrar preocupação com o desempenho global da técnica híbrida, o que se pode concluir é que este desempenho é fortemente dependente de uma série de “detalhes”. No transcorrer deste trabalho mostrou-se vários destes pontos que certamente fragilizam esta técnica caso não sejam observados.

Os testes realizados demonstraram a viabilidade da sistemática de abordagem de cada peso do conjunto individualmente pelo AG. Os resultados experimentais foram suportados pela plausibilidade biológica e matemática da proposta, o que lhe confere maior consistência e amplia a suas possibilidades futuras de emprego em outras áreas que não somente a engenharia, tais como a modelagem de seres vivos e sua relação com o ambiente.

Através do ambiente desenvolvido, realizaram-se inúmeros testes, alterando parâmetros, acrescentando outros, cogitando o por quê dos fatos, enfim, avaliando o desempenho e propondo alterações. Assim, foi possível levantar algumas heurísticas durante a execução dos testes preliminares. Estas heurísticas funcionaram como guia na busca e definição dos experimentos realizados com o ambiente de estudos, desenvolvido no

contexto deste trabalho (Anexo A). Além do desenvolvimento do ambiente, também foi necessário interfaceá-lo com o Excel¹³, conferindo maior flexibilidade ao ambiente de estudos.

Dentre os principais pontos originais deste trabalho foram destacados os seguintes:

- a) O emprego da análise de sensibilidade como guia para a definição e redução do espaço de busca do algoritmo genético, abre uma frente promissora de trabalhos. Isto devido a qualidade da informação que esta apresenta sobre a RNA;
- b) Proposta e implementação da variação dos pesos como alternativa ao cálculo da análise de sensibilidade clássica;
- c) O predomínio da maior sensibilidade do erro em relação a camada de pesos da saída do que em relação a camada de pesos da entrada, de uma rede 2-2-1 para o domínio da aplicação do XOR, foi um dos achados relevantes deste trabalho. Este fato viabiliza uma redução do espaço de busca, seja através de uma busca unimodal como apresentada aqui ou através de uma outra técnica que permita o uso desta informação para selecionar a região mais promissora. O fato de haver uma localização preferencial para os pesos mais influentes, principalmente no conjunto inicial gerado aleatoriamente, não é uma condição tida como esperada e conhecida da comunidade de IA. Portanto, esta é uma informação altamente relevante, principalmente quando o objetivo é reduzir o espaço de busca sem alterar as propriedades fundamentais de operação das técnicas envolvidas;
- d) Uma nova visão da inspiração biológica aplicada ao algoritmo genético;
- e) A abordagem do algoritmo genético em cada peso do conjunto de forma individual é viável e abre uma grande área de pesquisa e aplicações, embora ainda necessite de estudos sobre a capacidade e limitações, envolvendo o próprio XOR bem como outros domínios de aplicação.

¹³ Marca registrada da Microsoft.

Além dos pontos relatados anteriormente, há uma quantidade razoável de heurísticas e conclusões levantadas durante o transcorrer dos trabalhos. A seguir reúne-se algumas selecionadas como importantes dentro do contexto do domínio da aplicação escolhido, o XOR. No entanto, pela experiência adquirida acredita-se que as informações, quanto a aplicação da AS, permaneçam válidas em outros domínios, necessitando adaptações de acordo com o domínio da aplicação em questão:

- a) A função custo para o algoritmo genético é fator preponderante para uma boa operação, conforme afirmação inicial deste trabalho. Aparentemente, a representatividade da função custo decresce à medida que aumenta o número de variáveis interdependentes;
- b) No caso dos algoritmos baseados em GD a preocupação com a função de ativação e a relação desta com a função erro, também é relevante, podendo interferir significativamente na velocidade de treinamento dependendo da forma da superfície de erros;
- c) O número de etapas na busca genética unimodal é no mínimo tão importante quanto o número de gerações para cada peso. Cuidado adicional deve ser tomado na escolha do número de etapas quando o número de variáveis dependentes entre si é grande. Nos experimentos realizados houve indícios de que há um compromisso entre o número de variáveis e o número de etapas. Nesta afirmação há uma certa lógica, uma vez que apesar do problema ser multivariável o treinamento é executado de forma unimodal;
- d) Os valores iniciais dos pesos para o algoritmo genético são de fundamental importância para o bom desempenho do treinamento. Valores muito pequenos ($\pm 0,1$ ou ± 1), dependendo da aplicação, geralmente inviabilizam a aplicação, assim como valores muito elevados podem saturar os neurônios (função de ativação), levando o treinamento a paralisia;
- e) O algoritmo genético é muito sensível aos seus parâmetros (tamanho da população, taxa de cruzamento e mutação, composição da população, etc.);

- f) Uma avaliação comparativa da AS obtida pelo GD (figura 4.2a) e pela VP (figura 4.2b), mostra que as duas técnicas apresentaram comportamento semelhante. Isso demonstra a viabilidade do emprego da VP. Adicionalmente, como a VP foi obtida em função do padrão com maior contribuição para o erro da época, conclui-se também que é viável utilizar a AS através deste padrão de treinamento. Contudo, isto é válido se o erro deste padrão predominar significativamente sobre os demais padrões. Caso contrário, a sensibilidade obtida em função do erro da época poderá divergir em relação a sensibilidade obtida através do padrão com maior erro. A informação sobre a presença de um padrão predominante é importante em aplicações que tenham conjunto grande de padrões de treinamento. Assim, se for possível trabalhar com somente um padrão de treinamento, a carga computacional será reduzida significativamente, mesmo que a cada momento do treinamento possa haver alteração da distribuição da predominância destes padrões;
- g) Enfatiza-se também o fato deste trabalho contribuir para amenizar erros de terminologia quanto ao AG, principalmente quanto à origem e forma de atuação de cada operador, bem como a distinção entre um operador genético de um operador do algoritmo genético (item 2.2.1).

5.2 TRABALHOS FUTUROS

Durante a execução dos estudos e experimentos relatados neste trabalho surgiram vários pontos que seriam de grande interesse da comunidade científica por resolver algum problema ou por aumentar o conhecimento sobre os temas em questão. Assim, sugere-se os seguintes trabalhos futuros:

- a) Estudar e implementar a metodologia proposta para a transição entre o algoritmo de RP e o algoritmo genético. O enfoque principal deveria ser a redução do tempo de execução e ajuste dos parâmetros da proposta;
- b) Estudar a aplicação da AS para auxiliar na escolha de RNA já treinadas e com mesmo erro. Uma rede mais tolerante a falhas deve apresentar o conhecimento distribuído de forma mais homogênea;

- c) Estudar e testar a análise de sensibilidade clássica e a VP em função da “rugosidade” da superfície de erro;
- d) Organizar um conjunto de benchmarks para RNA e AG de tal forma que haja condições de validação e teste de novas propostas bem como a comparação de desempenho. Para isso seria importante criar uma base de dados onde outros pesquisadores pudessem coletar e submeter um domínio da aplicação, os resultados obtidos e em que condições foram obtidos;
- e) Desenvolvimento de um ambiente de AG e outras ferramentas de IA na WEB para processamento local e envio do resultado e demais arquivos por e-mail. Desta forma o usuário poderia acessar o computador da universidade e deixar o seu problema processando e ao final o próprio sistema enviaria o resultado por e-mail para o usuário;
- f) Utilizar a Teoria de Sistemas para construir uma base formal para AG e aprendizado;
- g) Estudar o desempenho de uma RNA treinada com GD e com AG sob o ponto de vista da capacidade de generalização, bem como se há alguma relação entre a capacidade de generalização e a distribuição do conhecimento na rede.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WIDROW, B.; LEHR, Michael A., 30 Years of adaptative neural networks: Perceptron, Madaline and Back-propagation 78, no. 9, pp. 1415-1442, 1990. IEEE Press. Proceedings of the IEEE.
- [2] BARRETO, Jorge M., Conexionismo e a resolução de problemas 1996. Universidade Federal de Santa Catarina. Concurso Público para Professor Titular. Tese.
- [3] BARRETO, J. M., Neural network learning: new programing paradigm? *Int.Conf.Trend and Direction in Expert System (ACM)*, vol. 1990. Orlando, Florida.
- [4] ANDERSON, James A., Organization of neural networks: Structures and models Verlagsgesellschaft Weinheim, 1988. Germany.
- [5] NARENDRA, Kumpati S.; PARTHASARATHY, Kannan, Identification and control of dynamical systems using neural networks *IEEE Transaction on Neural Networks*, vol. 1, no. 1, pp. 4-27, 1990.
- [6] COIMBRA, A. J. F.; D'ANGELO, G. G.; MARINO-NETO, J.; AZEVEDO, F. M. de; BARRETO, J. M., Use of neural networks in brain state analysis ed. Eric de Bodt & Michel Verleysen. 1994. Louvain-la-Neuve.
- [7] RODRIGUES, R. Glauco de S.; SILVA FILHO, A. C. R. da; PELÁ, Carlos A., Redes neurais artificiais para reconstrução de imagens tomográficas 1995. II Congresso Brasileiro de Redes Neurais. Curitiba, Brasil.
- [8] COIMBRA, A. J. F.; MARINO-NETO, J.; AZEVEDO, F. M. de; FREITAS, C. G.; BARRETO, J. M., Brain electrographic state detection using combined unsupervised and supervised neural networks *Artificial Intelligence in Medicine*, vol. 6, pp. 76-79, 1994.
- [9] BARRETO, J. M.; AZEVEDO, F. M. de, Connectionist expert systems as medical decision aid *Artificial Intelligence in Medicine*, vol. 5, pp. 515-523, 1993. Elsevier Science Publishers. Netherlands.

- [10] NIEVOLA, Julio Cesar, Sistema Inteligente para auxílio ao ensino em traumatologia crânio-encefálica 1995. Pós-Graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina. Tese de Doutorado.
- [11] LAWRENCE, Jeannette, Untangling neural nets *Dr.Dobb's*, vol. pp. 38-44, 1990.
- [12] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J., Learning representations by back-propagation errors *Nature*, vol. 323, no. 9, pp. 533-536, 1986.
- [13] KITANO, Hiroaki, Empirical studies on the speed of convergence of neural networks training using genetic algorithms no. Proceedings of the National Conference of the American Association of Artificial Intelligence (AAAI), pp. 789-795, 1990.
- [14] FAHLMAN, S. E., An empirical study of learning speed in back-propagation networks - Technical Report CMU-CS-88-162, 1988. Carnegie Melon University. Pittsburgh, PA.
- [15] WHITLEY, Darrel. Genetic algorithms and neural networks. In: *Genetic Algorithms in Engineering and Computer Science*, ed. J.Periaux and G.Winter. John Wiley & Sons, 1995.
- [16] DAYHOFF, Judith E. *Neural network architectures: an introduction*, New York: Van Nostrand Reinhold, 1989.
- [17] HAYKIN, Simon. *Neural networks: A comprehensive foundation*, IEEE Press, 1994.
- [18] KRÖSE, Bem J. A. & P. Patrick Van der Smagt. *An introduction to neural networks*, Amsterdam, Holand: University of Amsterdam, 1993.
- [19] DARYANANI, G. Principles of active network synthesis and design, John Wiley & Sons. New York, 1998.
- [20] KARAYIANNIS, Nicolaos B.; VENETSANOPOULOS, Anastasios N., Fast learning algorithms for neural networks *IEEE Transactions on Circuits and Systems - Analog and Digital Signal Processing*, vol. 39, no. 7, pp. 453-474, 1992.
- [21] FRANCONI, P.; GORI, M.; TESI, A., Successes and failures of back-propagation: a theoretical investigation, Dipartimento di Sistemi e Informatica da Università di Firenze. Firenze, Italy.
- [22] GIBB, Jondarr, Back-propagation family album - C/TR96-05, 1996. Department of Computing, Macquarie University.

- [23] HAMEY, Leonard G. C., Analysis of the Error Surface of the XOR Network with Two Hidden Nodes ed. M.U. Department of Computing. 95-167C, 1995. Australia.
- [24] HAMEY, Leonard G. C., Results on Weight Configurations that are not Local Minima in Feed-Forward Neural Networks ed. Seventh Australian Conf.Artificial Neural Networks. pp. 173-178, 1996. Australia.
- [25] MINSKY, M.; PAPPERT, S. *Perceptrons*, Cambridge, MA: MIT Press, 1969.
- [26] SPRINKHUIZEN-KUYPER, I. G.; BOERS, E. J. W., Classification of all stationary points on a neural network error surface.
- [27] SPRINKHUIZEN-KUYPER, I. G.; BOERS, E. J. W., The error surface of the simplest XOR network has no local minima ed. D.o.C.S. Leiden University. 94-21, 1994. The Netherlands.
- [28] SPRINKHUIZEN-KUYPER, I. G.; BOERS, E. J. W., The shape of the error surface of some simple neural networks NAIC'95: Proceedings of the Seventh Dutch Conference on Artificial Intelligence, pp. 275-284, 1995. In: J.C.Bioch and Y.-H.Tan. Rotterdam.
- [29] ROISENBERG, M.; BARRETO, Jorge M.; AZEVEDO, F. M. de; BRASIL, L. M., On a formal concept of autonomous agents no. Congresso da Alemanha, 1998.
- [30] Neural network glossary. <http://www.boltz.cs.cmu.edu/glossary.html> . 1998.
- [31] BARRETO, J. M. *Inteligência artificial: No limiar do século XXI*, Florianópolis, SC, J.M. Barreto, 1997.
- [32] MCCULLOCH, W. S.; PITTS, W. H. *A logical calculus of ideas immanent in nervous activity*, Bull. of Mathematical Biophysics, 1943. pp. 115-133.
- [33] RICH, Elaine. *Inteligência Artificial*, São Paulo: Makron Books, 1993.
- [34] KNEGT, Antonius H. M.; ARAÚJO, Evandro de O.; MEDEIROS, Gutemberg de S., Um acelerador genético para redes neurais no. 1º Congresso Brasileiro de Redes Neurais, 1994. Itajubá, Minas Gerais.
- [35] MAREN, Alianna J.; MARSTON, Craig T.; PAP, Robert M. *Handbook of Neural computing applications*, San Diego, California: Academic Press, 1990.

- [36] FAHLMAN, S. E., Faster-learning variation on back-propagation: an empirical study ed. Morgan Kaufmann. 1988. Connectionist Models Summer School.
- [37] SCHIFFMANN, W.; JOOST, M.; WERNER, R., Optimization of the back-propagation algorithm for training multilayer perceptrons Institute for Physics, University of Koblenz, 1992.
- [38] PEARLMUTTER, Barak. Gradient descent: Second order momentum and saturating error. In: eds. J.E. Moody, S.J. Hanson, and R.P. Lippmann. San Mateo, CA: Morgan Kaufmann, 1992.pp. 887-894.
- [39] WATROUS, Raymond L.; SHASTRI, Lokendra, Learning phonetic features using connectionist networks 4, pp. 381-388, 1987. IEEE First International Conference on Neural Networks.
- [40] JOHANSSON, E. M.; DOWLA, F. U.; GOODMAN, D. M., Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method - UCRL-JC-104850, 1990. Lawrence Livermore National Laboratory.
- [41] TENG, Chin-Chi, Mixed-Mode supervised learning algorithms for multilayer feed-forward 1993. Graduate College of the University of Illinois at Urbana-Champaign. Master of Science in Electric Engineering.
- [42] WASSERMAN, Philip D. *Neural computing: theory and practice*, New York: Van Nostrand Reinhold, 1989.
- [43] FOGEL, David B. *Evolutionary computation - Toward a new philosophy of machine intelligence*, Piscataway, NJ: IEEE Press, 1995.
- [44] SPEARS, William M.; DE JONG, K; BÄCK, T.; FOGEL, David B.; GARIS, H. de, An overview of evolutionary computation ed. Springer-Verlag. ed. P.Brazil. 667, pp. 442-459, 1993. European Conference on Machine Learning. Berlin.
- [45] FOGEL, L. J.; OWENS, A. J.; WALSH, M. J., Artificial intelligence through simulated evolution 1966. John Wiley, New York.
- [46] HOLLAND, J. H. *Adaptation in natural and artificial systems*, USA: Univ. of Michigan Press, 1975.
- [47] DARWIN, Charles Robert. *On the origin of species by means of natural selection*, Londres: Murray, 1859.
- [48] CROW, James F. *Fundamentos de Genética*, Rio de Janeiro: Livros Técnicos e Científicos Editora S/A, 1978.

- [49] SIRIVAS, M.; PATNAIK, Lalit M., Genetic algorithms: a survey *IEEE Computer*, vol. 1994.
- [50] CHUA, Leon O.; KOZEK, T.; ROSKA, T., Genetic algorithm for CNN template learning *IEEE Transactions on Circuits and Systems*, vol. 40, no. 6, pp. 392-402, 1993.
- [51] SILVA, A. C. da; LUIZ, C. C.; COELHO, A. A. R., Projeto do controlador PID genético: algoritmo e aplicação no II Congresso Brasileiro de Redes Neurais, 1995.
- [52] TANOMARU, J., Motivação, fundamentos e aplicações de algoritmos genéticos 1995. II Congresso Brasileiro de Redes Neurais e III Escola de Redes Neurais. Curitiba, Brasil.
- [53] BLANC, Marcel. *Os herdeiros de Darwin*, São Paulo: Scritta, 1994.
- [54] BALDWIN, J. M., A new factor in evolution *American Naturalist*, no. 30, pp. 441-451, 1896.
- [55] TURNEY, Peter, Myths and Legends of the Baldwin Effect pp. 135-142, 1996. Bari, Italy.
- [56] SOUTH, M. C.; WETHERILL, G. B.; THAM, M. T., Hitch-Hiker's guide to genetic algorithm, *Journal of Applied Statistics*, vol. 20, pp. 153-175, 1993.
- [57] GOLDBERG, David E.; RUDNICK, Mike, Genetic algorithms and the variance of fitness - IlliGAL 91001, 1991. Illinois Genetic Algorithms Laboratory (IlliGAL).
- [58] BEASLEY, David; BULL, David R.; MARTIN, Ralph R., An overview of genetic algorithms: Part 1, Fundamentals, 1993. Univ. of Cardiff. Cardiff, UK.
- [59] RIBEIRO, J. L.; TRELEAVEN, C., Genetic algorithm programming environments *IEEE Computer*, vol. pp. 28-43, 1994.
- [60] SPEARS, William M.; DE JONG, K, An analysis of multi-point crossover pp. 301-315, 1990. Proceedings of the Foundations of Genetic Algorithms Workshop.
- [61] SPEARS, William M.; DE JONG, K, On the virtues of uniform crossover eds. R.Belew and L.Booker. pp. 230-236, 1991. Fourth International Conference on Genetic Algorithms. San Mateo, CA.
- [62] GOLDBERG, D. E.; DEB, K.; CLARK, J. H., Genetic algorithms, noise and the sizing of populations *Complex Systems*, vol. no. 6, pp. 333-362, 1992.

- [63] HARIK, Georges; CANTÚ-PAZ, Erik; GOLDBERG, David E.; MILER, Brad L., The Gambler's ruin problem, genetic algorithms and the sizing of populations, 96004, 1996. Illinois Genetic Algorithms Laboratory (Illegal).
- [64] SPEARS, William M.; ANAND, Vic, A study of crossover operators in genetic programming pp. 409-418, 1991. In proceedings of the Sixth Int'l Symposium on Methodologies for Intelligent Systems. Washington.
- [65] DE JONG, K; SPEARS, William M., An analysis of the interacting roles of population size and crossover in genetic algorithms ed. Springer-Verlag. eds. H.P. Schwefel and R. Männer. 1990. Proceedings of the international Conference on Parallel Problem Solving from Nature. Berlin.
- [66] DE JONG, K; SPEARS, William M., A formal analysis of the role of multi-point crossover in genetic algorithms *Annals of Mathematics and Artificial Intelligence Journal*, vol. 5, pp. 1-26, 1992.
- [67] SPEARS, William M., Adapting crossover in evolutionary algorithms pp. 367-386, 1995. Fourth Evolutionary Programming Conference. San Diego, CA.
- [68] SPEARS, William M., Crossover or mutation? pp. 221-237, 1992. Foundations of Genetic Algorithms Workshop.
- [69] GOLDBERG, D. E. *Genetic algorithms in search, optimization and machine learning*, New York: Addison-Wesley Publishing Company, Inc., 1989.
- [70] MANIEZZO, Vittorio, Genetic evolution of the topology and weight distribution of neural networks *IEEE Transaction on Neural Networks*, vol. 5, no. 1, pp. 39-53, 1994.
- [71] PARISI, Domenico; NOLFI, Stefano, The influence of learning on evolution ed. R.K.M.M. Belew. Plastic Individuals in Evolving Populations, 1994. Addison-Wesley.
- [72] HINTON, G. E.; NOLAN, S. J., How learning can guide evolution *Complex Systems*, vol. 1, pp. 495-502, 1987.
- [73] MAYLEY, Giles, Landscape, Learning Costs and Genetic Assimilation ed. D.W.a.R.A. P.Turney. Evolutionary Computation, Evolution, Learning, and Instinct: 100 Years of the Baldwin Effect, 1996.
- [74] MAYLEY, Giles, The Evolutionary Cost of Learning ed. P.M.M.M.J.P.J.&W.S. In Maes. Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, 1996. From Animals to Animats 4. MIT Press.

- [75] WHITLEY, Darrel; GORDON, V. Scott; MATHIAS, Keith, Lamarckian Evolution, The Baldwin Effect and Function Optimization *Parallel Problem Solving from Nature. Lecture Notes in Computer Science*, vol. 866, pp. 6-15, 1994. Y Davidor, H. P. Schwefel and Männer. Berlin: Springer - Verlag.
- [76] WHITLEY, Darrell. Modeling Hybrid Genetic Algorithms.
- [77] GRUAU, Frédéric; WHITLEY, Darrell, Adding learning to the cellular development of neural networks: Evolution and the Baldwin effect *Evolutionary Computation*, vol. 1, no. 3, pp. 213-233, 1993.
- [78] SCHIFFMANN, W.; JOOST, M.; WERNER, R., Synthesis and performance analysis of multilayer neural network architectures ed. U.o.K. Institute for Physics. 16/1992, 1992. Germany.
- [79] GRUAU, Frédéric, Genetic synthesis of modular neural networks pp. 318-325, 1993. Fifth Int. Conf. Genetic Algorithms.
- [80] PALMER, Charles C.; KERSHENBAUM, Aaron, Representing trees in genetic algorithms Anonymous 1993. IBM T. J. Watson Research Center. Yorktown Heights, NY.
- [81] MILLER, G.; M., Todd P.; U., Hegde S., Designing neural networks using genetic algorithms pp. 379-384, 1989. Third International Conference on Genetic Algorithms (ICGA). San Mateo, CA.
- [82] GORI, M.; TESI, A., On the problem of local minima in back-propagation *IEEE Transactions on PAMI*, vol. 14, no. 1, pp. 76-86, 1992.
- [83] MONTANA, D. J.; DAVIS, L. D., Training feedforward neural networks using genetic algorithms ed. Morgan Kaufmann. 1989. Proceedings of the International Joint Conference on Artificial Intelligence.
- [84] BRANKE, Jürgen, Evolutionary algorithms for neural network design and training 1995. Proceedings of the 1st Nordic Workshop on Genetic Algorithms and its Applications. Vaasa, Finland.
- [85] RADCLIFFE, N. J., Genetic neural networks on MIMD computers 1990. University of Edinburgh, Edinburgh, Scotland. Doctoral dissertation.
- [86] RADCLIFFE, N. J., Genetic set recombination and its application to neural network topology optimization - EPCC-TR-91-21, 1991. University of Edinburgh. Edinburgh, Scotland.

- [87] KORNING, Peter G., Training neural networks by means of genetic algorithms working on very long chromosomes - Technical Report, 1994. Computer Science Department, Aarhus University. Denmark.
- [88] HANCOCK, P. J. B., Genetic algorithm and permutation problems: a comparison of recombination operators for neural structure specification *IEEE Computer Society*, vol. Combinations of Genetic Algorithms and Neural Networks, 1992. D. Whitley and J. D. Schaffer.
- [89] SPEARS, William M., Using neural networks and genetic algorithms as heuristics for NP-complete problems pp. 118-121, 1990. In Proceedings of the Int'l Joint Conference on Neural Networks.
- [90] BALA, J.; HUANG, J.; VAFAIE, H.; DE JONG, K; WECHSLER, H., Hybrid learning using genetic algorithms and decision trees for pattern classification IJCAI Conference, 1995. Montreal, Canada.
- [91] HAPPEL, Bart L. M.; MURRE, Jacob M. J., The design and evolution of modular neural network architectures *Neural Networks*, vol. 7, pp. 985-1004, 1994.
- [92] BÄCK, Thomas; SCHWEFEL, Hans-paul, An overview of evolutionary algorithms for parameter optimization *Evolutionary Computation*, vol. 1, no. 1, pp. 1-23, 1993.
- [93] HOCHMAN, Robert, Software reliability engineering: an evolutionary neural network approach 1997. Florida Atlantic University. Master of Science.
- [94] MURRAY, Dan, Tuning neural networks with genetic algorithms *AI Expert*, vol. 1994.
- [95] SHANG, YI; W., Wah Benjamin, Global optimization for neural network training *IEEE Computer*, vol. 29, no. 3, 1996.
- [96] CHALMERS, David J., The evolution of learning: an experiment in genetic connectionism ed. Morgan Kaufmann. 1990. Proceedings of the Connectionist Models Summer School. San Mateo, CA.
- [97] NOLFI, Stefano; ELMAN, Jeffrey L.; PARISI, Domenico, Learning and evolution in neural networks 3, no. Adaptive Behavior, pp. 5-28, 1994.
- [98] SVARER, Claus, Neural networks for signal processing 1994. Electronics Institute Technical University of Denmark. Ph. D. Thesis.

- [99] JANAKIRAMAN, Jayathi; HONAVAR, Vasant, Adaptive learning rate selection for backpropagation networks 1993. SPIE'93 Conference. Orlando, Florida.
- [100] MAHFOUD, Samir W., Crowding and preselection revisited *Parallel Problem Solving from Nature*, vol. 2, pp. 27-36, 1992. Elsevier Science Publishers. Amsterdam, Holland. IlliGAL 92004.
- [101] MILLER, Brad L.; GOLDBERG, D. E., Genetic algorithms, tournament selection, and the effects of noise - IlliGAL 95006, 1995. Department of General Engineering, University of Illinois at Urbana-Champaign.
- [102] OEI, Christopher K.; GOLDBERG, D. E.; CHANG, Shau-Jin, Tournament selection, niching, and the preservation of diversity - IlliGAL 91011, 1991. University of Illinois at Urbana-Champaign.
- [103] BLICHLE, Tobias, Selection Techniques - TIK-Report 11, Dec, 1995. Zürich.
- [104] CHARNIAC, E.; MCDERMOTT, D. *Introduction to artificial intelligence*, Massachusetts: Adilson-Wesley, 1985.
- [105] MITCHELL, Melaine. *An introduction to genetic algorithms*, London, England: The MIT Press, 1996.

ANEXO A - AMBIENTE DE ESTUDO

1.1 Introdução

A viabilização deste trabalho passa pela necessidade de um bom ambiente de estudos para avaliação e testes das técnicas e heurísticas envolvidas. Optou-se pelo desenvolvimento de um ambiente próprio; para atender as funções citadas, bem como para aplicação no ensino dos paradigmas e busca de novas heurísticas.

A motivação para o desenvolvimento do próprio ambiente ao invés da utilização do MATLAB¹, disponível no GPEB², deve-se mais à facilidade de implementação da interface homem máquina e a velocidade de execução do que à flexibilidade, onde, dependendo da necessidade de implementação o MATLAB pode levar vantagem. No entanto, a filosofia adotada não é de excluir outras ferramentas e nem de duplicar recursos.

Seguindo esta mesma filosofia, o ambiente permite dentro de certas limitações a exportação de dados para o Excel³ e Statística⁴. Durante o desenvolvimento dos trabalhos, notou-se que observações e busca de achados estatísticos poderiam inviabilizar o desenvolvimento do ambiente. Tendo isto em mente, optou-se por incluir no ambiente as avaliações estatísticas comumente empregadas na área de CE para comparar desempenhos, tais como: média, desvio padrão, valores mínimos e máximos. As demais funções estatísticas, apesar de não serem tão freqüentes, são essenciais à consistência e qualidade deste e dos futuros trabalhos e

¹ Marca registrada da MathWorks

² Grupo de Pesquisas em Engenharia Biomédica – EEL - UFSC

³ Marca registrada da Microsoft

⁴ Marca registrada da StatSoft

portanto, têm de ser consideradas. Isso levou a incluir no ambiente algumas rotinas que viabilizam a criação de arquivos no padrão do Excel e deste pode ser exportado sem maiores problemas para o Statistica, por exemplo.

O fato das análises realizadas pelo Excel ou Statistica não serem “on-line”, não limita e nem afeta o desempenho do sistema. Haja visto que o processamento estatístico essencial está implementado no ambiente e que o sistema na sua fase final de implementação automaticamente gravará uma ou mais amostras. Estas amostras serão descartadas ao final, caso o usuário não queira mantê-las em arquivo. Assim, a qualquer momento, o usuário poderá retornar a sua avaliação de acordo com a sua necessidade.

Conforme já mencionado, o desenvolvimento do ambiente foi dividido em 3 fases, sendo que as duas iniciais visam atender em primeiro plano aos objetivos do presente trabalho. A última deverá ser implementada após a finalização deste trabalho e de forma a incorporar as sugestões e necessidades levantadas durante a fase final do trabalho.

A primeira fase foi implementada com a finalidade de viabilizar os testes necessários ao exame de qualificação. Portanto, tendo em vista as incertezas serem maiores que as certezas, o ambiente foi implementado tendo como premissa fundamental a flexibilidade. Esta fase também foi decisiva na execução da análise de requisitos do ambiente a ser implementado na segunda fase. Para tanto foi levado em consideração: os dados obtidos a partir de uma análise crítica e detalhada da primeira fase, definição formal da tese aprovada no exame de qualificação e as necessidades que ocasionalmente surgirão durante o transcorrer do trabalho. A segunda fase consiste na implementação de um ambiente que viabilize a conclusão do presente trabalho bem como a análise de requisitos para a terceira fase.

A versão atual disponível, foi implementada utilizando-se uma ferramenta de desenvolvimento rápido de aplicações (“Rapid Application Development – RAD”), o Visual Basic 3.0. A implementação foi realizada nesta linguagem tanto para a interface homem máquina quanto para as demais rotinas de processamento propriamente dito. Durante a primeira fase acreditava-se que talvez esta linguagem não pudesse suportar de forma

eficiente rotinas com processamento matemático intenso. Assim, pretendia-se utilizar o Visual Basic somente para tarefas tais como a interface homem máquina. As rotinas que necessitassem de maior desempenho (velocidade de processamento, por exemplo), seriam implementadas na linguagem C e incorporadas ao ambiente. Contudo, no transcorrer da segunda fase a limitação de velocidade de processamento não foi constatada e portanto optou-se por finalizar a segunda fase utilizando-se somente a linguagem Visual Basic. Entretanto, isso não deve ser mantido para o ambiente a ser implementado na terceira fase, tendo em vista que outros domínios de aplicação mais complexos certamente exigirão maior capacidade de processamento. Neste ponto, torna-se importante um esclarecimento do porque ainda persistirmos com a versão 3.0 enquanto a mesma linguagem já disponibiliza a versão 5.0, muito superior à atualmente utilizada. O principal motivo é de que não há compatibilidade total entre a versão 3.0 e 5.0. Segundo, não há certeza de que esta linguagem vá ser a utilizada na terceira fase.

A seguir apresenta-se o ambiente atualmente implementado. A descrição realizada não é completa porque fugiria do objetivo deste capítulo. As informações apresentadas aqui são as essenciais à compreensão do ambiente bem como à sua utilização.

1.2 Detalhamento do Ambiente Atual

Na implementação do ambiente, considerou-se que inicialmente deveria permitir no mínimo a implementação e testes do treinamento híbrido de RMD.

O treinamento híbrido é executado pelo AG e uma técnica de treinamento baseado no gradiente descendente, como é o caso do algoritmo de RP. Assim, já que haveria necessidade da implementação destas duas técnicas, buscou-se disponibilizá-las de tal forma que possam ser executadas de forma independente. Já na fase inicial o ambiente começava dispondo de 3 ferramentas: o AG, o algoritmo de RP e o híbrido entre o AG e o do algoritmo de RP. Durante o desenvolvimento deste trabalho o número de implementações cresceram. Mas neste momento, o ambiente será

descrito como se houvesse apenas as 3 já citadas. No capítulo 3, consta as demais implementações bem como a justificativa para cada uma.

Tendo em vista os objetivos deste ambiente, optou-se por uma tela bastante intuitiva e que as informações gráficas, parâmetros de entrada e de acompanhamento estejam disponível na mesma tela. Caso o volume de informações torne-se crítico, o ambiente permite que a informação seja trazida para a tela sem que haja prejuízo ao treinamento em execução. Seguindo esta premissa, o ambiente consiste de uma tela principal e duas telas secundárias, além da tela de abertura (institucional).

Ao executar o software, primeiramente aparece uma tela institucional que após algum tempo ou o pressionar do mouse ou qualquer tecla, é substituída pela tela principal (figura 1). Esta tela, permanecerá até o fim do treinamento (tanto na fase de entrada dos parâmetros quanto na fase de treinamento). O que mudará será a superposição de parte da tela principal por outras telas específicas a cada momento. Contudo, a barra de menu existente na parte superior permanecerá, enquanto que a parte logo abaixo do menu é utilizada pelas telas secundárias.

A tela principal apresenta na parte superior um menu que ficará ativo durante todo o tempo em que o software estiver em execução. Neste, aparece da esquerda para a direita: o menu Parâmetros, destinado a entrada dos parâmetros para uma determinada ferramenta. O menu Arquivo, destinado a abertura e gravação de arquivos necessários ao treinamento e resultado deste para posterior análise. O menu Inicialização, o qual destina-se a reinicializar os pesos ou indivíduos caso os valores atuais devam ser descartados. O menu Executa, destina-se a execução do treinamento propriamente dito. Contudo, como foram implementadas várias ferramentas, o usuário deverá escolher dentro deste menu a aplicação desejada. Assim que uma delas for selecionada a correspondente execução dará início. O menu Sair, como o próprio nome já diz, ao ser pressionado cessa a execução e não mais é possível retornar de onde antes estava. E por último o menu Estatística que destina-se a gerar um arquivo no formato Excel para posterior processamento estatístico. A criação de uma função específica para a

geração de uma base de dados para fins de análise estatística, deve-se a grande carga computacional existente o que tornaria o sistema muito lento.

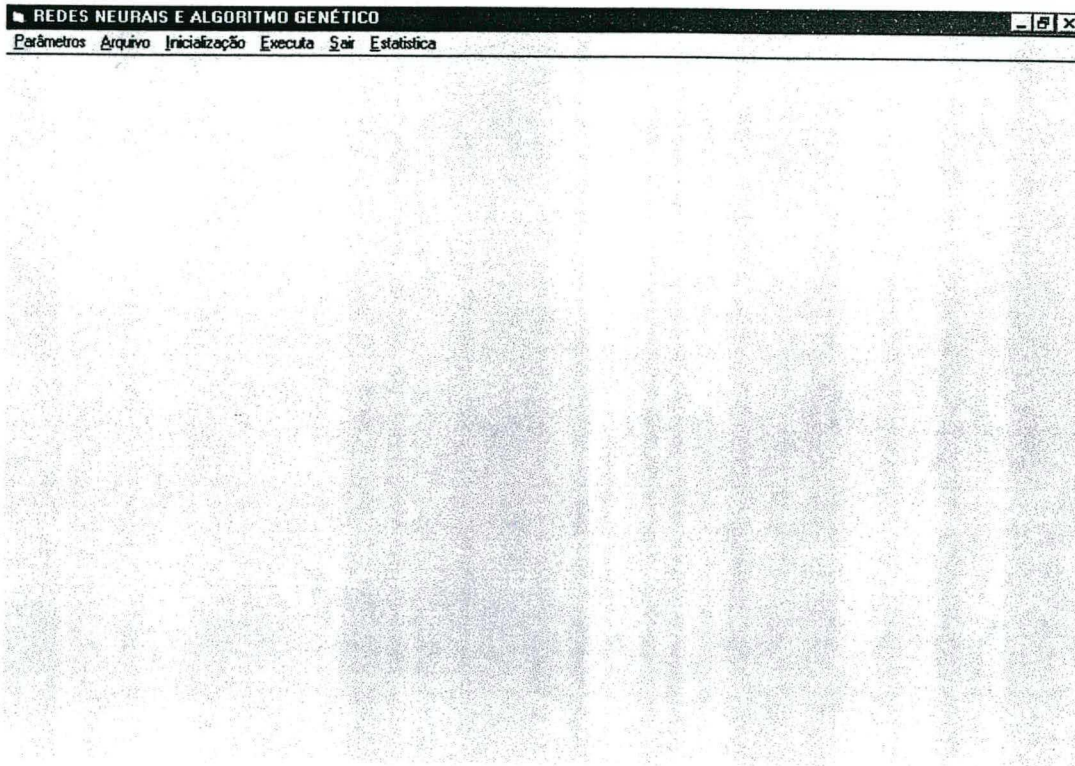


Figura 1 Tela principal do ambiente de estudos

Basicamente há duas telas secundárias que aparecerão a seu tempo. A primeira é a de entrada dos parâmetros e a segunda a da execução do treinamento. A tela de entrada dos parâmetros é específica para cada ferramenta a ser executada. A escolha se dá através do menu Parâmetro (barra de menu da tela principal), o qual apresenta a lista de todas as ferramentas implementadas no ambiente. A seleção de uma ferramenta automaticamente acarreta na apresentação da correspondente tela para a entrada dos parâmetros pertinentes à aquela ferramenta escolhida. Assim, se o usuário escolher o treinamento híbrido (AG+RP) aparecerá a primeira tela secundária, a de Parâmetros de Entrada do AG e do algoritmo de RP (figura 2). Esta tela será superposta a parte da tela principal. O menu da tela principal ainda continua visível e ativo na parte superior da tela.

REDES NEURAIS E ALGORITMO GENÉTICO

Parâmetros Arquivo Inicialização Executa Sair Estatística

PARÂMETROS DE ENTRADA

ALGORITMO GENÉTICO

População:

Qtos Bits:

Casas Decimais:

Desvio Padrão:

Prob.

Prob. Mutação:

Faixa Superior:

Faixa Inferior:

Geração Máxima:

Número Etapas:

Núm. Etapas Sal.:

Desvio Melhor %:

Pesos Salientes:

Default

REDE NEURAL

Erro min. desej.

Coef. trein.

Coef. momento

Total de épocas

Arquivo Teste

Arquivo Trein.

Default

NUMERO DE CAMADAS

OK

Figura 2 Tela de Entrada de Parâmetros superposta a tela principal

Na figura 2, a tela referente a entrada de parâmetros do AG e do algoritmo de RP encontra-se dividida em duas partes: à esquerda estão os parâmetros do AG e à direita os referente ao algoritmo de RP. As teclas *default* destinam-se a entrada de parâmetros pré-selecionados sendo que o usuário pode alterar estes parâmetros individualmente como desejar. Após as alterações, o usuário deve carregar o arquivo dos padrões de treinamento e testes. Isto é realizado através do menu “Arquivo” da barra de menu da tela principal. Neste basta escolher Abrir e depois, Padrão de Treinamento. Após a escolha do arquivo, o processo deve se repetir para o arquivo dos padrões de teste. O nome dos arquivos selecionados aparecerá nos campos correspondentes na tela “Parâmetros de Entrada”.

Finalizada a entrada de parâmetros, deve-se acionar a tecla “Ok”. O acionamento desta tecla provoca várias ações:

- a) Carrega os parâmetros da tela para o sistema;
- b) Retira a tela de “Entrada de Parâmetros” que está superposta a tela principal;

c) Carrega a tela secundária “Execução do Treinamento” sob a tela principal (figura 3).

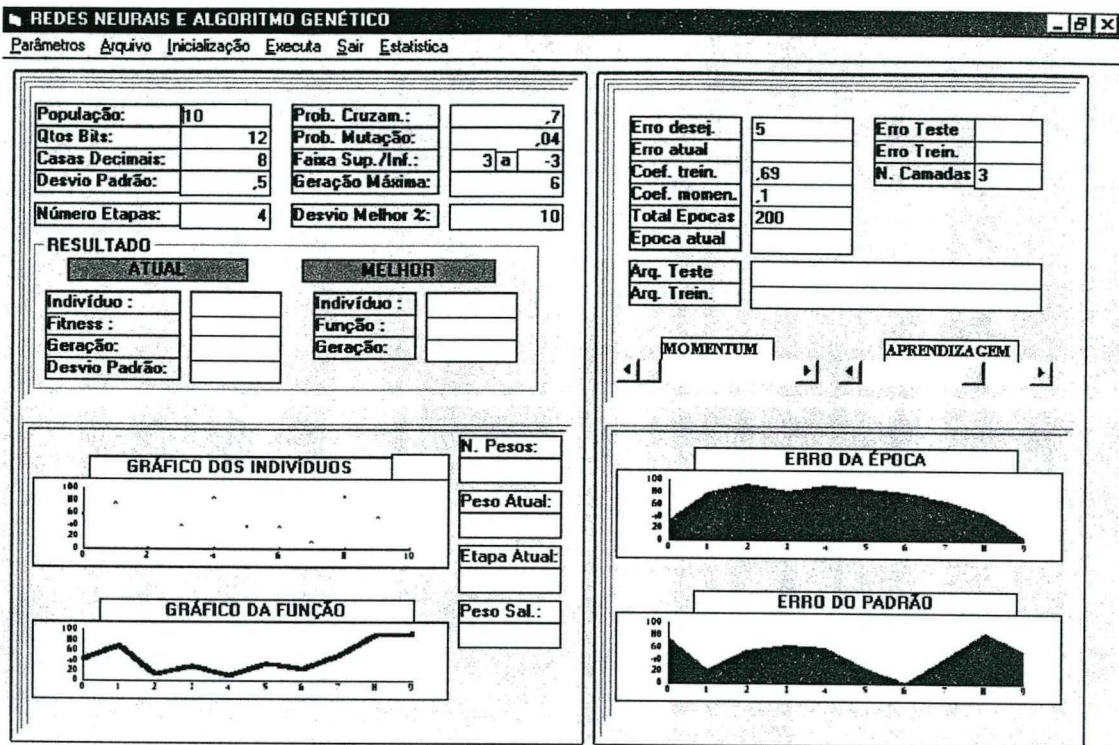


Figura 3 Tela de acompanhamento do treinamento

Na figura 3, pode-se observar que o menu da tela principal continua acessível, apesar da tela secundária ter sido trocada. A tela secundária de acompanhamento do treinamento, a exemplo da de entrada dos parâmetros, encontra-se dividida em duas partes. À esquerda o AG e à direita a Rede Neural.

No caso do AG, basicamente a tela apresenta-se funcionalmente dividida em 3 partes. Na parte superior consta uma cópia de todos os parâmetros de entrada escolhidos pelo usuário com o auxílio da tela anterior. Todos os parâmetros podem ser alterados antes do início do treinamento e alguns podem ser alterados a qualquer momento, inclusive durante a execução do treinamento. A segunda parte da tela, destina-se aos parâmetros de acompanhamento do treinamento. Nestes encontram-se

informações tais como: melhor indivíduo já encontrado, geração que isso ocorreu, bem como o que está acontecendo na presente geração. Para completar a exposição/acompanhamento do treinamento, a terceira parte da tela, apresenta dois gráficos sendo o superior destinado aos indivíduos da população e o inferior à aptidão dos indivíduos.

A apresentação dada a rede neural, segue a mesma filosofia da adotada no AG. Assim, na parte superior apresentam-se os parâmetros de entrada e também os parâmetros numéricos de acompanhamento do treinamento. Na parte inferior apresentam-se dois gráficos, um destinado a mostrar o erro da época ao longo do tempo e outro o erro dos padrões em cada época.

Finalizando o processamento, o sistema avisa qual o tipo de convergência que ocorreu, por erro mínimo ou por número de gerações ou épocas. Sendo que o usuário pode optar pela gravação em arquivos para posterior processamento, os seguintes dados: parâmetros de treinamento escolhidos, topologia da rede e os respectivos pesos obtidos no treinamento.

O conjunto de gráficos e parâmetros numéricos utilizados para acompanhar o treinamento foi muito útil na determinação e desenvolvimento deste trabalho, conforme será visto mais adiante. Contudo, na apresentação dos requisitos para a próxima versão, sugere-se vários acréscimos.

1.3 Análise de Requisitos

Como já mencionado antes, a presente análise de requisitos é produto de duas outras bem como da experiência adquirida durante a fase final do presente trabalho. Entretanto, não tem a intenção de ser completa e sim de servir de guia para uma análise mais rigorosa e que leve em consideração um levantamento apurado de outros sistemas com objetivos similares ou próximos. Assim, não serão discutido aqui os requisitos da linguagem e sim os requisitos básicos do sistema, visando principalmente um elevado desempenho e flexibilidade de adaptação aos problemas e necessidades de informação do usuário.

Para uma maior flexibilidade, tanto os dados como os parâmetros de configuração, deverão entrar no sistema pelo teclado ou por intermédio de

arquivos, havendo necessidade de crítica destes. Por exemplo, se uma determinada rede é especificada, não faz sentido aceitar os dados de uma diferente da especificada (conflitante). Os dados de entrada e saída poderão ser visualizados no vídeo ou gravados em arquivos de preferência com terminações intuitivas:

- Pesos iniciais: *.pin;
- Pesos finais (após o treinamento): *.pfi;
- Parâmetros de treinamento da rede: *.tre;
- Definição da rede: *.red;
- Padrões de treinamento: *.ptr;
- Padrões de teste: *.pte

É interessante que os arquivos explicitados acima possam ser visualizados em uma tela suplementar e que esta possa ser deslocada sobre a do treinamento. Nesta, também poderia constar a qualidade dos dados, por exemplo o desvio padrão, valores máximo, mínimo e valor médio. Assim, em caso de dúvida pode-se visualizar se o processamento em execução partiu de algum ponto já tendencioso ou não.

Já os parâmetros necessários para o início do processamento deverão ser feito por “*default*”, via teclado ou por arquivo. Sendo que os mesmos poderão ser alterados manualmente/automaticamente durante o transcorrer do processamento. A necessidade do “*default*” destina-se a auxiliar os usuários iniciantes. Assim, o tempo gasto no domínio do ambiente é drasticamente reduzido durante o primeiro contato com as técnicas e principalmente com o ambiente. Ao usuário, que utilizar-se das características “*default*”, caberá apenas os padrões de treinamento e testes bem como a respectiva topologia da rede a ser treinada (número de camadas intermediárias e o respectivo número de neurônios). A informação quanto ao número de neurônios da camada de entrada e saída é obtida automaticamente através do arquivo correspondente ao conjunto de padrões de treinamento. Caso o usuário queira inserir o número de neurônios destas

camadas, o *software* automaticamente fará uma crítica de consistência entre a topologia e o conjunto de padrões de treinamento e testes.

Após a inserção dos parâmetros necessários e/ou uso dos parâmetros “*default*”, o usuário pode iniciar a execução e durante a mesma, pode alterar os parâmetros iniciais (inclusive os “*default*”).

No caso específico do AG deverá ser possível a escolha dos principais tipos de cruzamento e mutação, bem como uma composição flexível da nova população. Além dos parâmetros tais como:

- Tamanho da população;
- Quantos bits de resolução são necessários ao indivíduo. Este item permitirá utilizar a resolução do indivíduo como fator limitador do espaço de busca quando o objetivo for apenas uma aproximação;
- Quantas casas decimais serão utilizadas;
- Desvio padrão da população aceitável como tendo ocorrido a convergência;
- Taxa de cruzamento bem como a escolha do tipo de cruzamento;
- Taxa de mutação bem como a escolha do tipo de mutação;
- Valor superior e inferior do intervalo segundo o qual será gerada a população inicial;
- Geração máxima limite para processamento genético caso não ocorra convergência pelo critério do desvio padrão;
- entre outros.

Para a RMD, deve-se permitir a alteração da função de ativação (uma linear e outras não lineares como a sigmóide e a tangente hiperbólica), escolha dos parâmetros fixos durante o treinamento ou variável manual ou automaticamente e principalmente de outras técnicas baseadas no gradiente descendente. Além das opções acima, deve-se prever no mínimo os seguintes parâmetros:

- Erro mínimo desejado, segundo o qual será aceito como tendo finalizado o treinamento;
- Coeficiente de treinamento ou taxa de treinamento;

- Coeficiente de momento;
- Total de épocas limite para o treinamento caso não ocorra a convergência segundo o critério do erro mínimo;
- Arquivo de teste;
- Número de neurônios em cada camada da RMD, não incluído o "bias" que será incluído de forma automática;
- Número de camadas da rede, incluindo a de entrada;
- entre outros.

Durante o processamento, é conveniente que na mesma tela seja apresentado os parâmetros previamente definidos, bem como os valores para acompanhamento do processo. O acompanhamento do processo será em tempo real, através de informações e gráficos. Permitindo desta forma analisar o comportamento atual com o anterior e a partir disso levantar heurísticas mais adaptadas ao problema em questão. Quanto ao gráfico é interessante dotar o sistema de um sensor de posição do mouse. Assim, quando este estiver sobre o gráfico o sistema informará o valor que define o ponto sob o qual o mouse se situa naquele momento. Esta mesma técnica permitirá que ao deslocar o mouse sobre um determinado parâmetro apareça uma janela descrevendo o que é este parâmetro, qual o valor default e demais informações necessárias ao treinamento de um usuário iniciante. Desta forma será como se um tutor estivesse presente e sensível ao contexto.

A presente análise tem por objetivo servir de base para uma especificação mais aprofundada, a ser realizado por profissionais da área em conjunto com uma equipe de interessados em CE, principalmente AG. Portanto, cabe mais ao presente trabalho o fato de ser a semente de um sistema mais amplo e genérico do que propriamente completo.