

# **Algoritmo Evolutivo com Inspiração Quântica, Representação Mista e Decaimento de Pesos Aplicado a Neuroevolução**

Antonio Cesar de Oliveira Pitta Botelho

## **Internal Research Reports**

Number 38 | October 2014

# **Algoritmo Evolutivo com Inspiração Quântica, Representação Mista e Decaimento de Pesos Aplicado a Neuroevolução**

Antonio Cesar de Oliveira Pitta Botelho

### **CREDITS**

#### **Publisher:**

**MAXWELL / LAMBDA-DEE**

**Sistema Maxwell / Laboratório de Automação de Museus, Bibliotecas Digitais e Arquivos**

**<http://www.maxwell.vrac.puc-rio.br/>**

#### **Organizers:**

Alexandre Street de Aguiar  
Delberis Araújo Lima

#### **Cover:**

Ana Cristina Costa Ribeiro

# **Algoritmo Evolutivo com Inspiração Quântica, Representação Mista e Decaimento de Pesos Aplicado a Neuroevolução**

**Antonio Cesar de Oliveira Pitta Botelho <sup>1</sup>**

**Resumo:** Algoritmos evolutivos com inspiração quântica representam um dos mais recentes avanços na área de computação evolucionária. Eles têm sido utilizados para se evoluir redes neurais artificiais em contrapartida aos métodos tradicionais de treinamento baseados no gradiente decrescente e retropropagação do erro. O algoritmo utilizado neste trabalho foi o NeuroEvolutivo com Inspiração Quântica e representação mista Binária e Real (NEIQ-BR) desenvolvido para modelagem de redes neurais do tipo perceptron de múltiplas camadas e utilizado em problemas de classificação de padrões.

A técnica de regularização de complexidade denominada decaimento de pesos tem como objetivo fazer com que os pesos excessivos, ou seja, aqueles que têm pequena ou nenhuma influência sobre a rede, assumam valores próximos de zero, reduzindo com isto a complexidade da rede e melhorando sua capacidade de generalização.

Este trabalho introduz a técnica de regularização de complexidade de decaimento de pesos no algoritmo neuroevolutivo em referência e avalia o desempenho do algoritmo em três casos benchmark de classificação de padrões. Os resultados serão comparados aos obtidos com o algoritmo original.

**Palavras-chave:** Algoritmos genéticos. Algoritmos com inspiração quântica. Neuroevolução. Computação evolucionária. Redes neurais artificiais. Método de Decaimento de pesos. Classificação de padrões.

<sup>1</sup> Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO) – Departamento de Engenharia Elétrica – Área de Métodos de Apoio à Decisão – Rua Marques de São Vicente, 225, Gávea – Rio de Janeiro – CEP 22415-900 – Rio de Janeiro - RJ (Brasil) – Tel: (55) 21 3527-1001 – E-mail: acesarpitta@yahoo.com.br

# Quantum-Inspired Evolutionary Algorithm with Mixed Representation and Weight Decay Applied to Neuroevolution

Antonio Cesar de Oliveira Pitta Botelho <sup>1</sup>

**Abstract:** *Quantum-inspired evolutive algorithms represent one of the most recent advances in evolutionary computation. They have been used to evolve artificial neural networks in contrast to traditional training methods based on the decreasing gradient and backpropagation. The algorithm used in this work was to Quantum-Inspired Evolutionary algorithm for Neuro-evolution with binary-real representation (QIEN-BR) developed for modeling neural networks of multilayer perceptron and used to problems of pattern classification.*

*The regularization technique called weight decay aims to make the weights who have little or no influence on the network, take values close to zero, with this reducing network complexity and improving their ability to generalize.*

*This work implements weight decay in the algorithm QIEN-BR and to evaluates the performance of the new algorithm, we will use three benchmark cases of classification. The results will be compared with those obtained with the original algorithm.*

**Keywords:** *Genetic algorithms. Quantum-inspired algorithms. Neuro-evolution. Evolutionary computation. Artificial neural networks. Weight Decay Method. Pattern Classification.*

<sup>1</sup> Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO) – Departamento de Engenharia Elétrica – Área de Métodos de Apoio à Decisão – Rua Marques de São Vicente, 225, Gávea – Rio de Janeiro – CEP 22415-900 – Rio de Janeiro - RJ (Brasil) – Tel: (55) 21 3527-1001 – E-mail: acesarpitta@yahoo.com.br

# Índice

<b>1. Introdução.....</b>	<b>1</b>
1.1 Organização do Trabalho.....	2
<b>2. Conceitos.....</b>	<b>3</b>
2.1 Algoritmos com Inspiração Quântica .....	3
2.2 Neuroevolução.....	4
<b>3. Algoritmo Evolutivo com Inspiração Quântica e Representação Mista .....</b>	<b>5</b>
3.1 Listagem do Algoritmo .....	5
3.2 População Quântica.....	6
3.3 Observação dos Indivíduos Quânticos .....	8
3.4 Atualização do Indivíduo Quântico.....	10
3.5 Cruzamento de Indivíduos Clássicos .....	12
3.6 Algoritmo Neuroevolutivo com Inspiração Quântica e Representação Mista .....	13
3.6.1 Cromossoma .....	14
3.6.2 Parâmetros do Modelo .....	15
3.6.3 Funções de Avaliação .....	16
<b>4. Técnicas de Regularização de Complexidade.....</b>	<b>18</b>
<b>5. Estudos de Caso .....</b>	<b>19</b>
5.1 Australian Credit.....	19
5.2 Wisconsin Prognostic Breast Cancer .....	20
5.3 Iris Data.....	20
5.4 Experimentos.....	20
<b>6. Resultados .....</b>	<b>22</b>
6.1 Erro de Codificação Identificado no Algoritmo de (Pinho10) .....	22
6.2 Resultados do Algoritmo NEIQ-BR (Pinho10) .....	25
6.3 Resultados do Algoritmo NEIQ-BR com Taxa de Decaimento de 0,001 .....	26
6.4 Resultados do Algoritmo NEIQ-BR com Taxa de Decaimento de 0,005 .....	29
6.5 Resultados do Algoritmo NEIQ-BR com Taxa de Decaimento 0,01 .....	32
<b>7. Conclusão e Trabalhos Futuros.....</b>	<b>35</b>
<b>Referências Bibliográficas .....</b>	<b>37</b>

# 1. Introdução

Os algoritmos evolutivos com inspiração quântica representam um dos mais recentes avanços na área de computação evolutiva. Estes algoritmos se baseiam em idéias inspiradas na física quântica, em particular no conceito de superposição de estados (Cruz07). O principal objetivo em se buscar inspiração na física quântica é utilizar conceitos de múltiplos universos para melhoria do desempenho dos algoritmos genéticos (Pinho10).

Algoritmos evolutivos têm sido utilizados para se evoluir redes neurais artificiais em contrapartida aos métodos tradicionais de treinamento baseados no gradiente decrescente e retropropagação do erro (Pinho10).

O algoritmo evolutivo com inspiração quântica e representação mista binária-real (AEIQ-BR) desenvolvido por (Pinho10) herda as principais características de seus precursores, tais como:

1. Capacidade de otimização global de um problema;
2. Representação probabilística do espaço de busca; e
3. Necessidade de poucos indivíduos na população, o que reduz drasticamente o número de avaliações necessárias para a otimização.

Este algoritmo foi utilizado na evolução de redes neurais artificiais do tipo *perceptron* de múltiplas camadas e alimentação adiante em problemas associados à modelagem da rede tais como:

1. Otimização da topologia das camadas escondidas e de entrada;
2. Seleção dos padrões de entrada;
3. Obtenção dos pesos da camada de entrada;
4. Obtenção dos pesos das camadas escondidas;
5. Seleção das funções de avaliação;
6. Seleção das funções de ativação dos neurônios das camadas escondidas;
7. Seleção das funções de ativação dos neurônios da camada de saída.

No entanto, o modelo neuroevolutivo proposto por (Pinho10) não considera durante a evolução a possibilidade de desativar os neurônios que tem pouca influência no processo de classificação, ou seja, pesos próximos de zero.

Este trabalho consiste em implementar, no algoritmo NEIQ-BR, a técnica de decaimento de pesos e avaliar os resultados obtidos comparando-os aos do algoritmo original. A técnica de decaimento de pesos (*weight decay*), além de permitir reduzir a

complexidade do modelo por meio do decaimento dos pesos, auxilia também na identificação dos elementos que tem pouca influência no processo de classificação.

## **1.1 Organização do Trabalho**

O texto deste trabalho encontra-se organizado da seguinte forma:

- Um resumo dos fundamentos teóricos necessários para a compreensão deste trabalho é apresentado no capítulo 2;
- No capítulo 3 o modelo do algoritmo evolutivo com inspiração quântica e representação binária-real desenvolvido por (Pinho10) é apresentado em detalhes;
- O capítulo 4 apresenta um resumo dos métodos de regularização de complexidade de redes neurais com foco na técnica de decaimento de pesos;
- Os estudos de caso de classificação de padrões utilizados para avaliação do novo algoritmo são descritos no capítulo 5;
- O capítulo 6 apresenta os resultados obtidos com o novo algoritmo e compara com os do algoritmo original de (Pinho10); e
- No último capítulo, são deduzidas as conclusões do trabalho, seguido da bibliografia utilizada.

## 2. Conceitos

### 2.1 Algoritmos com Inspiração Quântica

Algoritmos com inspiração quântica são algoritmos clássicos (capazes, portanto, de serem executados em computadores clássicos) que tiram proveito dos paradigmas da física quântica tais como superposição de estados e paralelismo, de forma a melhorar o desempenho dos mesmos na resolução de problemas (Cruz07).

Exemplos de algoritmos com inspiração quântica são apresentados em (Moore95), um para determinar a solução para o problema do *15-puzzle* e outro para resolver o problema de ordenação numérica. Uma metodologia para a formulação de um algoritmo com inspiração quântica também foi definida por (Moore95), que consiste dos seguintes passos:

1. O problema deve ter uma representação numérica ou, caso não tenha, um método para sua conversão em representação numérica deve ser empregado;
2. A configuração inicial deve ser determinada;
3. Uma condição de parada deve ser definida;
4. O problema deve poder ser dividido em subproblemas menores;
5. O número de universos (ou estados de superposição) deve ser identificado (conceito de múltiplos universos);
6. Cada subproblema deve ser associado a um dos universos (cada universo existe em paralelo);
7. Os cálculos nos diferentes universos devem ocorrer de forma independente;
8. Iteração entre os múltiplos universos (atualizações na dinâmica de cada universo).
9. Verificação clássica final para averiguar se o resultado obtido é uma solução válida.

Em termos de computação evolutiva com inspiração na física quântica, exemplos de algoritmos são o AEIQ-B de (Han00), onde ao invés de uma representação binária convencional, este algoritmo usa uma representação que simula um cromossomo formado por *q-bits*, numa analogia a um bit quântico de um computador quântico e o AEIQ-R de (Cruz07), que usa uma representação baseada em números reais para permitir representar uma superposição de estados contínuos.



## 2.2 Neuroevolução

A neuroevolução é um modelo híbrido que explora a potencialidade de duas diferentes áreas inspiradas em processos biológicos: Redes Neurais Artificiais (RNA) e Algoritmos Genéticos (AG). A idéia básica da neuroevolução é buscar, automaticamente, a melhor configuração para uma rede neural usando algoritmos genéticos. Em outras palavras, a neuroevolução combina a capacidade de generalização e aproximação de funções das redes neurais artificiais com um método eficiente de busca paralela. O objetivo dos algoritmos genéticos é melhorar os algoritmos de aprendizado, automatizando, total ou parcialmente, o processo de configuração da rede neural, bem como o processo de treinamento e atualização dos pesos da mesma (Cruz07).

As abordagens possíveis para os sistemas neuroevolutivos diferem uma da outra, basicamente, pela forma de representação dos genes dos cromossomos de seus algoritmos genéticos e os objetivos a que estes são submetidos. Os cromossomos podem codificar qualquer informação relevante para a parametrização da rede neural, incluindo:

1. Seleção de padrões de entrada;
2. Topologia das camadas escondidas;
3. Topologia da camada de entrada;
4. Pesos sinápticos;
5. Funções de avaliação.

O próximo capítulo detalha o funcionamento do algoritmo evolutivo com inspiração quântica e representação mista desenvolvido por (Pinho10).

### **3. Algoritmo Evolutivo com Inspiração Quântica e Representação Mista**

O uso de algoritmos evolutivos com inspiração quântica em problemas de natureza numérica e combinatorial como, por exemplo, otimização de plantas industriais, controle, previsão, classificação e treinamento de redes neurais pode, potencialmente, oferecer inúmeras vantagens (Cruz07).

O Algoritmo AEIQ-BR desenvolvido por (Pinho10) é um modelo evolutivo com inspiração quântica que usa uma representação mista numérica e combinatória no cromossoma. Este algoritmo tem as seguintes vantagens em relação aos algoritmos genéticos tradicionais:

1. O conhecimento do problema que está sendo otimizado é armazenado diretamente nos cromossomos quânticos;
2. Populações com poucos indivíduos, que como consequência, irá reduzir o tempo de processamento;
3. Capacidade de convergir de forma eficiente mesmo com populações com poucos indivíduos (Cruz07).

As próximas seções deste capítulo apresentam uma descrição completa do algoritmo bem como cada um dos seus passos.

#### **3.1 Listagem do Algoritmo**

A figura 3.1 mostra a listagem completa do Algoritmo Evolutivo com Inspiração Quântica e representação Binária-Real.

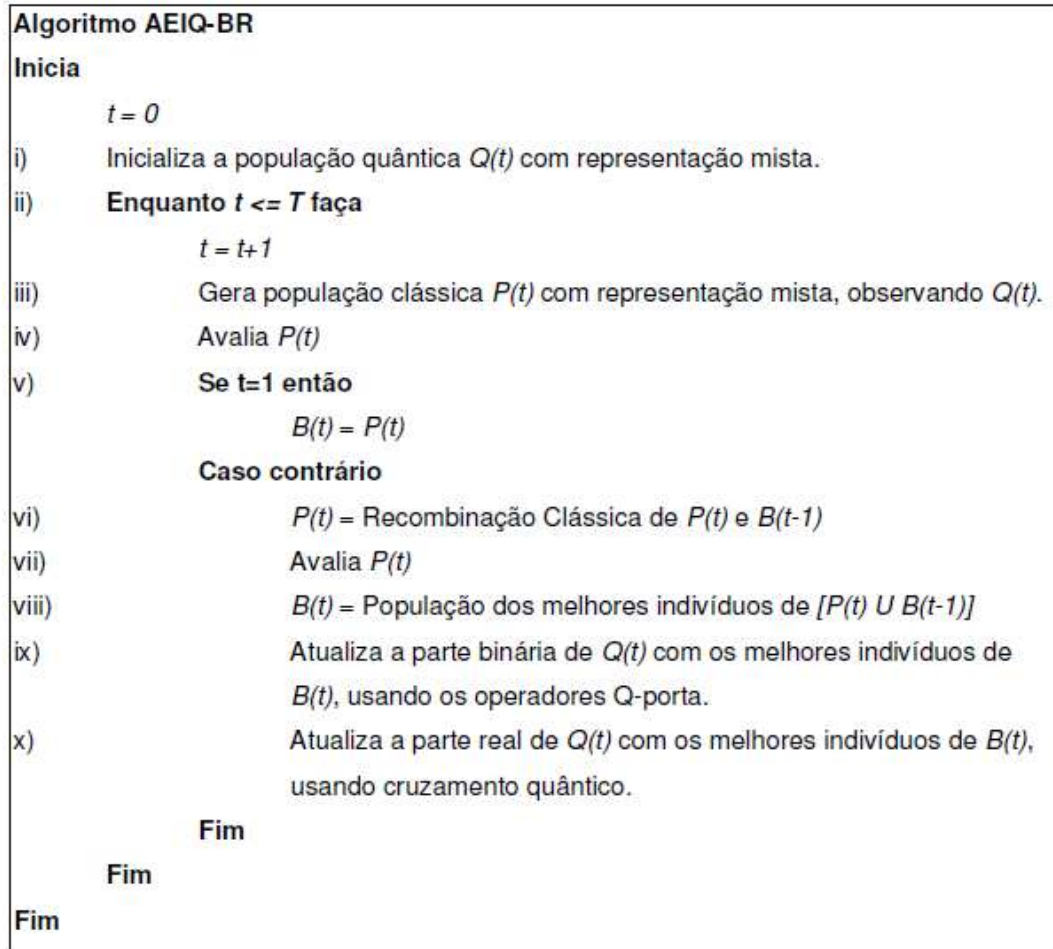


Figura 3.1 - Listagem completa do algoritmo AEIQ-BR (Pinho10).

Este algoritmo será explicado detalhadamente nas subseções a seguir.

### 3.2 População Quântica

O algoritmo AEIQ-BR necessita de uma população de indivíduos que representem a superposição dos possíveis estados que o indivíduo clássico pode assumir ao ser observado.

A população quântica  $Q(t)$ , em um instante  $t$  qualquer do processo evolutivo, é formada por um conjunto de  $N$  indivíduos quânticos  $q_i^t$  ( $i = 1, 2, 3, \dots, N$ ).

$$Q(t) = [q_1^t, q_2^t, q_3^t, \dots, q_N^t] \quad (3.1)$$

Cada indivíduo quântico  $q_i^t$  desta população é formado por  $L$  genes  $g_{ij}$  ( $j = 1, 2, 3, \dots, L$ ). Uma parte do indivíduo será representada através de genes quânticos binários (q-bit) e outra parte através de genes quânticos reais (q-real). Portanto, a representação de um indivíduo quântico  $i$  qualquer num instante de tempo  $t$  será dada por:

$$q_i^t = \left[ \left( q_i^t \right)_b, \left( q_i^t \right)_r \right] \quad (3.2)$$

onde o índice  $b$  representa a parte binária (q-bit) e o índice  $r$  representa a parte real (q-real).

A parte binária de um indivíduo quântico com  $M$  genes binários pode ser escrita da forma:

$$(q_i)_b = \left( g_{i1} = \begin{bmatrix} \alpha_{i1}^t \\ \beta_{i1}^t \end{bmatrix}, g_{i2} = \begin{bmatrix} \alpha_{i2}^t \\ \beta_{i2}^t \end{bmatrix}, \dots, g_{iM} = \begin{bmatrix} \alpha_{iM}^t \\ \beta_{iM}^t \end{bmatrix} \right)_b \quad (3.3)$$

onde  $\alpha_{ij}^t$  e  $\beta_{ij}^t$  são números complexos cuja soma dos quadrados de cada um dos elementos para os  $M$  genes é 1.

A parte real de um indivíduo quântico com  $G$  genes reais pode ser escrita da forma:

$$(q_i)_r = \left( g_{i1}^t = p_{i1}^t(x_{i1}^t), g_{i2}^t = p_{i2}^t(x_{i2}^t), g_{i3}^t = p_{i3}^t(x_{i3}^t), \dots, g_{iG}^t = p_{iG}^t(x_{iG}^t) \right)_r \quad (3.4)$$

onde as funções  $p_{ij}$  representam funções densidade de probabilidade (FDP). Esta função densidade de probabilidade é usada para gerar os valores para os genes dos indivíduos clássicos. Em outras palavras, a função  $p_{ij}(x)$  representa a densidade de probabilidade de se observar um determinado valor para o gene quântico quando a superposição do mesmo for colapsada. A função densidade de probabilidade deve ser integrável na região do domínio dentro do qual as variáveis que se quer otimizar podem assumir valores. Isto permite calcular a distribuição cumulativa de probabilidade e assim usar a distribuição de probabilidade para se gerar os valores para os genes clássicos (Cruz07).

A função densidade de probabilidade utilizada é o pulso quadrado, definida pela equação 3.5:

$$p_{ij}^t(x_{ij}^t) = f(x_{ij}^t; \mu_{ij}^t; \sigma_{ij}^t) = \begin{cases} \frac{1}{\sigma_{ij}^t}, & \text{se } (\mu_{ij}^t - \frac{\sigma_{ij}^t}{2}) \leq x_{ij}^t \leq (\mu_{ij}^t + \frac{\sigma_{ij}^t}{2}) \\ 0, & \text{caso contrário} \end{cases} \quad (3.5)$$

onde  $\mu$  é a posição do ponto central do pulso quadrado,  $\sigma$  a largura do mesmo,  $i=1,2,3,\dots,N$ , representando o  $i$ -ésimo indivíduo, e  $j=1,2,3,\dots,G$  representando o  $j$ -ésimo gene q-real. Portanto, pode-se representar a parte real do cromossoma como segue:

$$(q_i)_r = \left( g_{i1}^t = p_{i1}^t(x_{i1}^t), g_{i2}^t = p_{i2}^t(x_{i2}^t), g_{i3}^t = p_{i3}^t(x_{i3}^t), \dots, g_{iG}^t = p_{iG}^t(x_{iG}^t) \right)_r = \left( \begin{bmatrix} \mu_{i1}^t \\ \sigma_{i1}^t \end{bmatrix}, \begin{bmatrix} \mu_{i2}^t \\ \sigma_{i2}^t \end{bmatrix}, \begin{bmatrix} \mu_{i3}^t \\ \sigma_{i3}^t \end{bmatrix}, \dots, \begin{bmatrix} \mu_{iG}^t \\ \sigma_{iG}^t \end{bmatrix} \right)_r \quad (3.6)$$

Tem-se então que, um indivíduo quântico, pode ser descrito por:

$$q_i = \left[ (q_i)_b, (q_i)_r \right] = \left[ \begin{bmatrix} \alpha_{i1}^t & \alpha_{i2}^t & \dots & \alpha_{iM}^t \\ \beta_{i1}^t & \beta_{i2}^t & \dots & \beta_{iM}^t \end{bmatrix}_b, \begin{bmatrix} \mu_{i1}^t & \mu_{i2}^t & \mu_{i3}^t & \dots & \mu_{iG}^t \\ \sigma_{i1}^t & \sigma_{i2}^t & \sigma_{i3}^t & \dots & \sigma_{iG}^t \end{bmatrix}_r \right] \quad (3.7)$$

### 3.3 Observação dos Indivíduos Quânticos

Após a inicialização da população quântica o algoritmo entra no laço principal do processo evolutivo. Este laço será executado por um determinado número  $T$  de gerações. O passo (iii) é um dos mais importantes do algoritmo. É neste momento que se executa o processo de observação do estado quântico para se gerar indivíduos clássicos.

A observação do indivíduo quântico é feita através de um processo de seleção aleatória com iguais probabilidades de uma solução no espaço de busca (Pinho10). A execução desta observação é realizada em duas etapas: observação da parte binária e observação da parte real de um indivíduo quântico.

A geração de um gene clássico binário  $(x_{ij}^t)_b$  qualquer através da observação de um gene q-bit  $(g_{ij}^t)_b$  qualquer, é feita usando-se um gerador uniforme de números aleatórios no intervalo entre 0 e 1. Se o número aleatório gerado estiver entre 0 e a probabilidade de *crossover* clássico ( $C_c$ ), atribui-se o valor 0 ao gene clássico, caso contrário atribui-se o valor 1.

A figura 3.2 mostra a listagem do algoritmo de observação da parte binária de um indivíduo quântico.

```
Observação dos Q-bits  
Inicia  
  Para  $i=1,2,3,...,N_Q$ -número de indivíduos quânticos  
    Seleciona-se  $q_i^t$ , o  $i$ -ésimo indivíduo quântico em  $Q(t)$ .  
    Para  $j=1,2,3,...,M$ -número de q-bits presentes em  $q_i^t$   
      Calcula-se a probabilidade de se ter genes 0 ou 1 dado, respectivamente, por  
       $(\alpha_{ij}^t)^2$  e  $(\beta_{ij}^t)^2$ , em  $(g_{ij}^t)_b$ .  
      Gera-se um número aleatório entre 0 e 1.  
      Se aleatório entre 0 e uma probabilidade de crossover clássico  $C_c$ .  
        Atribui-se ao gene clássico  $(x_{ij}^t)_b$  o valor 0.  
      Caso contrário  
        Atribui-se ao gene clássico  $(x_{ij}^t)_b$  o valor 1.  
        (crossover tipo uniforme)  
      Fim Se  
    Fim Para  
  Fim Para  
Fim
```

Figura 3.2 – Listagem do algoritmo de observação da parte binária de um indivíduo quântico (Pinho10).

A geração de um gene clássico real  $(x_{ij}^t)_r$  qualquer através da observação de um gene q-real  $(g_{ij}^t)_r$  qualquer é feita através da inversa da distribuição acumulada de probabilidade de cada um dos genes quânticos. Para cada gene q-real é realizada uma observação através do seguinte procedimento:

1. Gera-se um número aleatório  $y_{ij}^t$  no intervalo entre 0 e 1;
2. Identifica-se o ponto tal que  $(x_{ij}^t)_r = P_{ij}^{t-1}(y_{ij}^t)$ , onde  $P_{ij}^{t-1}(y_{ij}^t)$  é a inversa da função distribuição cumulativa de probabilidade;
3.  $x$  é o valor observado para o gene  $j$  do indivíduo clássico  $i$ .

A figura 3.3 mostra a listagem do algoritmo de observação da parte real de um indivíduo quântico.

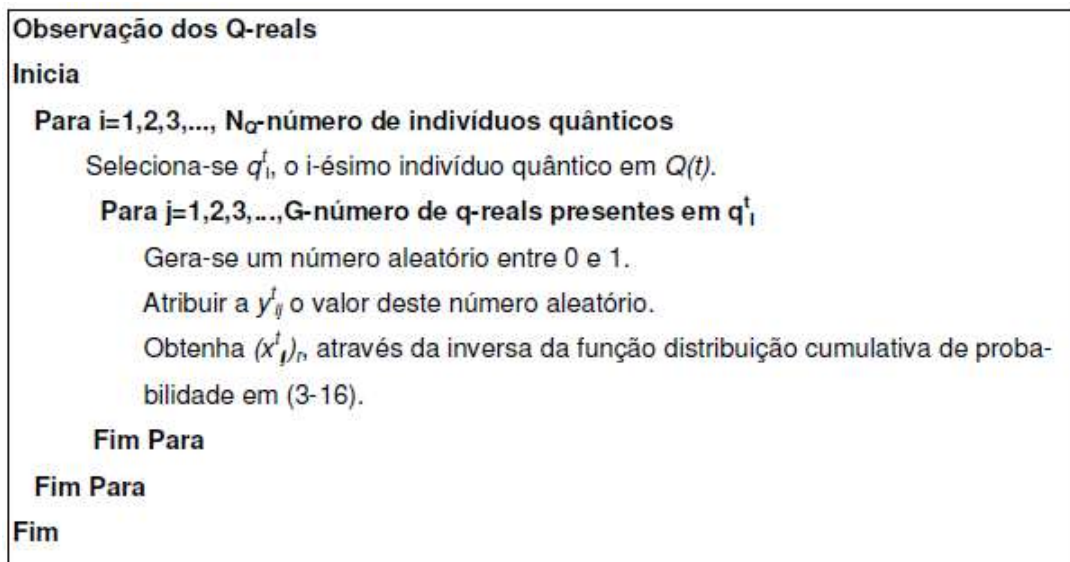


Figura 3.3 - Listagem do algoritmo de observação da parte real de um indivíduo quântico (Pinho10).

O número de indivíduos clássicos  $N_C$  pode ser igual ou maior ao número de indivíduos quânticos  $N_Q$ , bastando que, ao se gerar os indivíduos clássicos, não se dê preferência a certos indivíduos quânticos no processo de observação. Os indivíduos quânticos não são, a princípio, avaliados como os indivíduos clássicos e, portanto, não podem ser considerados melhores ou piores uns que os outros. Em outras palavras, deve-se garantir que nenhum indivíduo quântico seja preterido ou privilegiado em relação aos outros no que diz respeito ao número de vezes que o mesmo será usado para o processo de observação (Cruz07).

É importante ressaltar que, o número de indivíduos quânticos sendo maior que o número de indivíduos clássicos implicaria que alguns indivíduos quânticos não estariam sendo observados, comprometendo o processo de exploração do espaço de soluções.

Portanto é recomendado que  $N_C$  seja um múltiplo de  $N_Q$ , pois assim cada indivíduo quântico teria condições iguais de gerar, através da observação, a mesma quantidade de indivíduos clássicos.

### 3.4 Atualização do Indivíduo Quântico

Após a geração dos indivíduos clássicos, é necessário atualizar os indivíduos da população quântica com base na população dos melhores indivíduos clássicos (passos  $ix$  e  $x$  do algoritmo da figura 3.1). Os melhores indivíduos clássicos fornecerão as direções de atualização para que os indivíduos quânticos representem melhores soluções em gerações seguintes. Em geral, este processo de atualização dos indivíduos quânticos deve reduzir o espaço de busca da função que se quer otimizar e mapear as regiões mais promissoras deste espaço de busca (isto é obtido aumentando-se a probabilidade de se observar um determinado conjunto de valores para o gene clássico nas proximidades dos indivíduos mais bem sucedidos da população clássica) (Cruz07).

A seleção, sem repetição, dos indivíduos clássicos da população  $B(t)$ , é por ordem decrescente de aptidão. A probabilidade de cruzamento quântico  $C_Q$  é constante durante todas as gerações (Pinho10).

A atualização de qualquer q-bit  $(g_{ij}^t)_b$  de um indivíduo quântico  $q_i^t$  é executada no passo  $ix$  da figura 3.1 através do operador do tipo porta de Rotação  $U(\theta)$  dado por (Pinho10):

$$U(\theta) = \begin{bmatrix} \cos(\theta) & (+/-)\text{seno}(\theta) \\ (+ou-)\text{seno}(\theta) & \cos(\theta) \end{bmatrix} \quad (3.8)$$

onde  $\theta$  é um ângulo de rotação definido pelo usuário. Assim, a atualização de um gene quântico binário  $(g_{ij}^t)_b$  a partir do gene clássico correspondente  $(x_{ij}^t)_b$  de um indivíduo clássico qualquer, pertencente à população dos melhores indivíduos  $B(t)$ , é dado por (Pinho10):

$$(g_{ij}^{t+1})_b = \begin{bmatrix} \alpha_{ij}^{t+1} \\ \beta_{ij}^{t+1} \end{bmatrix}_b = \begin{cases} \begin{bmatrix} \cos(\theta) & \text{seno}(\theta) \\ -\text{seno}(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} \alpha_{ij}^t \\ \beta_{ij}^t \end{bmatrix}, & \text{se } (x_{ij}^t)_b = 0 \\ \begin{bmatrix} \cos(\theta) & -\text{seno}(\theta) \\ \text{seno}(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} \alpha_{ij}^t \\ \beta_{ij}^t \end{bmatrix}, & \text{se } (x_{ij}^t)_b = 1 \end{cases} \quad (3.9)$$

Já a atualização dos genes quânticos reais  $(g_{ij}^t)_r$  a partir do gene clássico correspondente  $(x_{ij}^t)_r$  de um indivíduo clássico qualquer, pertencente à população dos melhores indivíduos  $B(t)$ , consiste em (passo  $x$  do algoritmo da figura 3.1):



1. Modificação da posição dos pulsos quadráticos de modo que o ponto central dos mesmos ( $\mu$ ) coincida com o valor dos genes de um conjunto de indivíduos da população clássica. A atualização de  $\mu$  é definida por:

$$\mu_{ij}^{t+1} = \mu_{ij}^t + \lambda^t ((x_{ij}^t)_r - \mu_{ij}^t) \quad (3.10)$$

onde  $\lambda^t$  é um número aleatório entre 0 e 1 gerado a cada geração  $t$ ;

2. Modificação da largura dos pulsos quadráticos ( $\sigma$ ) de modo que o espaço de buscas seja reduzido. A atualização de  $\sigma$  é feita através de um decaimento exponencial e é definida pela equação 3.11.

$$\sigma_{ij}^{t+1} = \sigma_{ij}^t + \lambda^t \cdot ((\max((x_{ij}^t)_r) - \min((x_{ij}^t)_r)) - \sigma_{ij}^t) \quad (3.11)$$

onde:

- $\max((x_{ij}^t)_r)$ : máxima amplitude presente em  $(x_{ij}^t)_r$  entre todos os melhores indivíduos de  $B(t)$ ;
- $\min((x_{ij}^t)_r)$ : mínima amplitude presente em  $(x_{ij}^t)_r$  entre todos os melhores indivíduos de  $B(t)$ ;
- $\lambda^t$ : mesmo número aleatório entre 0 e 1 utilizado em 3.10.

A figura 3.4 mostra a listagem do algoritmo de atualização de um indivíduo quântico.

```

Atualização do Indivíduo Quântico
Inicia
  Para i=1,2,3,...,  $N_Q$ -número de indivíduos quânticos
    Seleciona-se  $q_i^t$ , o i-ésimo indivíduo quântico em  $Q(t)$ .
    Seleciona-se  $c_i^t$ , o i-ésimo indivíduo clássico em  $B(t)$ .
    Para j=1,2,3,...,  $M$ -número de q-bits presentes em  $q_i^t$ 
      Gera-se um número aleatório entre 0 e 1 para  $(g_{ij}^t)_b$ 
      Se aleatório entre 0 e uma probabilidade de crossover quântico  $C_Q$ 
        Atualiza  $(g_{ij}^t)_b$  com portão quântico  $U(\theta)$ , na direção de  $(x_{ij}^t)_b$ 
      Fim Se
    Fim Para
    Para h=1,2,3,...,  $G$ -número de q-reals presentes em  $q_i^t$ 
      Gera-se um número aleatório entre 0 e 1 para  $(g_{ih}^t)_r$ 
      Se aleatório entre 0 e uma probabilidade de crossover quântico  $C_Q$ 
        Atualiza  $(g_{ih}^t)_r$  com equações (3-19) e (3-20) na direção de  $(x_{ih}^t)_r$ 
      Fim Se
    Fim Para
  Fim

```

Figura 3.4 - Listagem do algoritmo de atualização de um indivíduo quântico (Pinho10).



### 3.5 Cruzamento de Indivíduos Clássicos

O espaço de melhores soluções  $B(t)$  funciona como um espaço de boas características, sendo refinado a cada geração, através da introdução de novas soluções ajustadas e/ou observadas. Este processo de refinamento acontece nos passos *vi*, *vii* e *viii* da figura 3.1 e consiste de (Pinho10):

1. Cruzamento clássico dos  $N$  indivíduos de  $B(t-1)$  com os observados no passo atual em  $P(t)$ , gerando somente um filho para cada par de indivíduos. Cada par de indivíduos é selecionado aleatoriamente em  $B(t-1)$  e  $P(t)$  sem reposição;
2. Substituição dos  $N$  filhos gerados na população  $P(t)$  atual;
3. Avaliação dos  $N$  filhos gerados em  $P(t)$  através de uma função de aptidão;
4. Atualização da população  $B(t)$  com os  $N$  melhores indivíduos da união de duas populações,  $B(t-1)$  e  $P(t)$ .

O processo de cruzamento é controlado por duas variáveis de probabilidade de cruzamento clássico: uma para a parte clássica binária,  $C_{Cb}$ , e outra para a parte clássica real,  $C_{Cr}$ , ambas definidas pelo usuário e constantes durante todas as gerações.

O tipo de cruzamento definido para a parte binária nos indivíduos clássicos é o uniforme. Por este tipo de operador, um gene binário  $(x_{ij}^t)_{bF}$  de um indivíduo filho  $(c_i^t)_F$  é definido da seguinte forma:

$$(x_{ij}^t)_{bF} = \begin{cases} (x_{ij}^t)_{bP}, & \text{se aleatório entre 0 e } C_{Cb} \\ (x_{ij}^t)_{bB}, & \text{caso contrário} \end{cases} \quad (3.12)$$

onde:

- “aleatório”: número aleatório entre 0 e 1 gerado para cada gene;
- $C_{Cb}$ : probabilidade de cruzamento clássico binário;
- B: indica que o gene real é de um indivíduo originário da população  $B(t)$ ;
- P: indica que o gene real é de um indivíduo originário da população  $P(t)$ .

Já para a parte real, o tipo de cruzamento definido é o aritmético. Por este tipo de operador, um gene real  $(x_{ij}^t)_{rF}$  de um indivíduo filho  $(c_i^t)_F$  é dado por:

$$(x_{ij}^t)_{rF} = \begin{cases} (x_{ij}^t)_{rB} + \lambda^t \left( (x_{ij}^t)_{rP} - (x_{ij}^t)_{rB} \right), & \text{se aleatório entre 0 e } C_{Cr} \\ (x_{ij}^t)_{rB}, & \text{caso contrário} \end{cases} \quad (3.13)$$

onde:

- “aleatório”: número aleatório entre 0 e 1 gerado para cada gene;
- $\lambda^t$ : velocidade de atualização do indivíduo quântico;
- $C_{cr}$ : probabilidade de cruzamento clássico real;
- B: indica que o gene real é de um indivíduo originário da população  $B(t)$ ;
- P: indica que o gene real é de um indivíduo originário da população  $P(t)$ .

A figura 3.5 apresenta a listagem do algoritmo de cruzamento entre indivíduos clássicos das populações  $P(t)$  e  $B(t)$  (passo vi da figura 3.1).

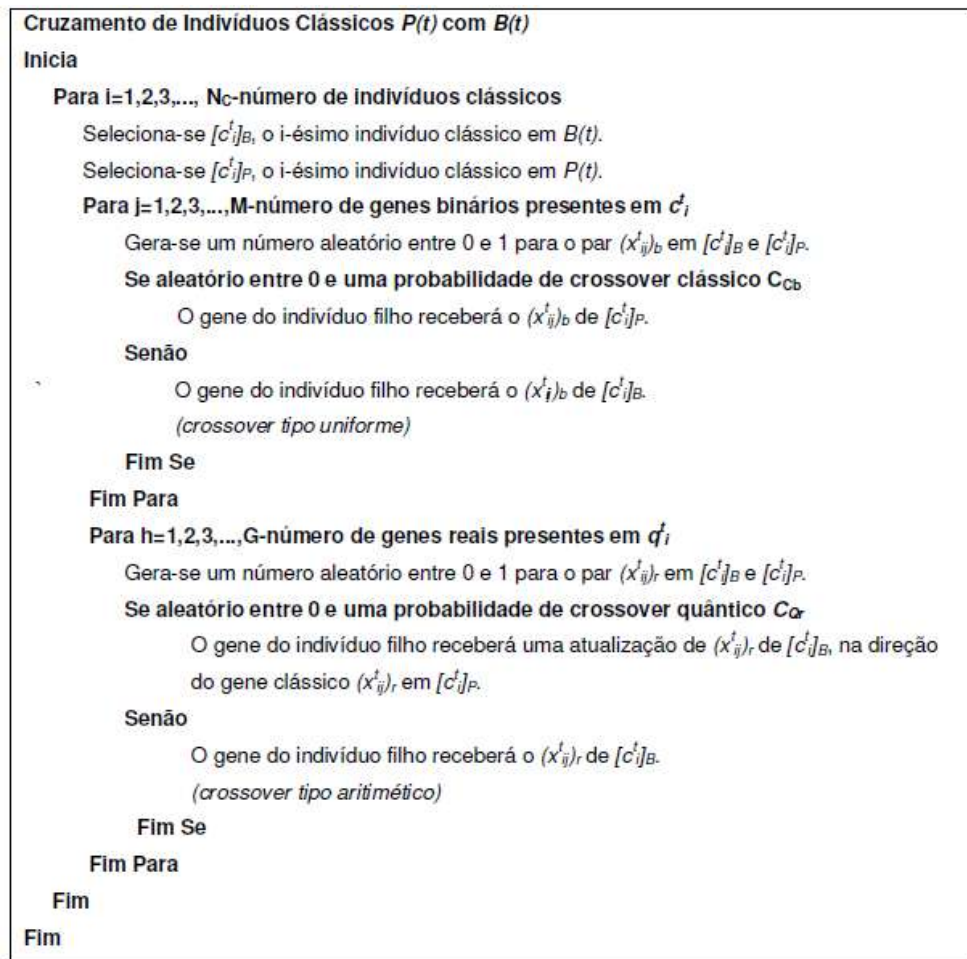


Figura 3.5 - Listagem do algoritmo de cruzamento de indivíduos clássicos em  $P(t)$  com  $B(t)$  (Pinho10).

### 3.6 Algoritmo Neuroevolutivo com Inspiração Quântica e Representação Mista

O algoritmo NeuroEvolutivo com Inspiração Quântica e representação mista Binária e Real – NEIQ-BR é uma extensão do algoritmo AEIQ-BR para evoluir uma rede neural artificial do tipo *perceptron* de múltiplas camadas. A utilização deste modelo neuroevolutivo está restrita ao tratamento de problemas de classificação (Pinho10).

### 3.6.1 Cromossoma

A informação que é codificada em cada um dos genes quânticos ou clássicos, binários ou reais de um cromossoma está definida na figura 3.6.

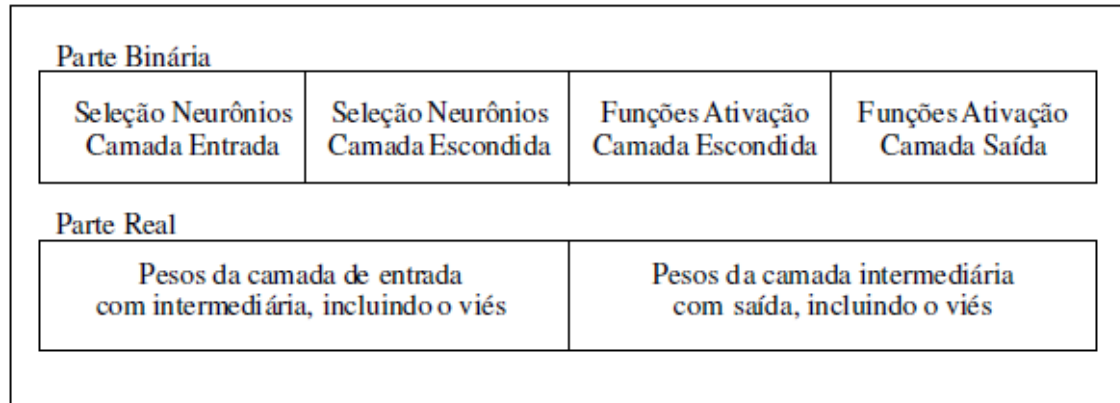


Figura 3.6 – Informação codificada em um cromossoma quântico ou clássico (Pinho10).

A decodificação de um cromossoma clássico, por sua vez, é definida conforme a seguir (Pinho10):

- **Camada de entrada:** códigos na parte binária do cromossoma determinam quais variáveis de entrada serão selecionadas, ou seja, estarão ativas na RNA, onde o código 1 significa presença e o código 0 significa ausência.
- **Camada intermediária:** códigos na parte binária do cromossoma determinam quais neurônios estarão ativos do total definido pelo usuário, onde o código 1 significa presença e o código 0 significa ausência. Cada neurônio ativo realizará a soma ponderada dos valores propagados da camada anterior. Cada um destes neurônios ativará esta soma ponderada com a função de ativação definida pela parte binária do cromossoma, onde 1 significa tangente hiperbólica e 0 sigmóide logística.
- **Camada de saída:** cada neurônio representará uma de um total de  $TO$  classes. Todos os neurônios realizarão a soma ponderada dos valores propagados da camada anterior, e genes na parte binária do cromossoma indicarão a função de ativação a ser utilizada onde, da mesma forma, o código 1 significa tangente hiperbólica e 0 sigmóide logística.

A predição acontecerá através da atribuição da classe ao padrão de entrada representada pelo neurônio da camada de saída cuja ativação é máxima entre todos os neurônios desta camada.

É importante frisar que a evolução de pesos e funções de ativação nos cromossomas quânticos e clássicos está condicionada ao fato do neurônio estar ativo ou não na parte binária correspondente. Ou seja, os genes representantes dos pesos e funções de ativação permanecerão inalteráveis por cruzamentos quânticos e clássicos, caso este neurônio esteja inativo.

Outro ponto importante é que o usuário poderá definir o número mínimo de gerações que um neurônio permanecerá ativo e, portanto sofrendo atualizações de seus pesos e função de ativação antes que se permita a sua desativação por um cruzamento clássico.

### 3.6.2 Parâmetros do Modelo

Os parâmetros do modelo proposto por (Pinho10) e configuráveis pelo usuário são:

- $NQ$ : número de indivíduos quânticos;
- $NC$ : número de indivíduos clássicos;
- $CQ$ : probabilidade de *crossover* quântico;
- $C_{Cb}$ : probabilidade de *crossover* clássico binário;
- $C_{Cr}$ : probabilidade de *crossover* clássico real;
- $T$ : número de gerações;
- $\theta$ : ângulo de rotação para velocidade de atualização dos genes clássicos binários;
- $minT$ : número mínimo de gerações que um neurônio permanecerá ativo antes que se permita a sua desativação por um cruzamento clássico (este parâmetro fará com que haja um contador individual para cada gene clássico binário responsável pela atividade, ou inatividade, de um neurônio na camada de entrada e na camada intermediária);
- $updateT$ : intervalo entre gerações até que os operadores quânticos sejam executados em todos os indivíduos da população quântica, atualizando os centros e amplitudes com base nos melhores indivíduos da população clássica;
- $TI$ : número máximo de neurônios na camada intermediária.

O modelo neuroevolutivo com decaimento de pesos, tema deste trabalho, apresenta além dos parâmetros definidos acima, dois novos parâmetros, também configuráveis pelo usuário:

- $\lambda$ : taxa de decaimento dos pesos da rede;
- $numexperimentoAG$ : número de experimentos do algoritmo genético.

### 3.6.3 Funções de Avaliação

As formas de avaliação de um cromossoma clássico definidas para o modelo NEIQ-BR, conforme passos *iv* e *vii* do algoritmo da figura 3.1, são (Pinho10):

1. Percentual total de acerto ponderado pela classe 1;
2. Percentual total de acerto não ponderado;
3. Área por debaixo da curva ROC.

Em problemas de classificação de padrões com duas classes, uma forma simples de avaliar os resultados e identificar se o método adotado de classificação é bom ou ruim é através da matriz de confusão. A matriz de confusão oferece uma medida efetiva do modelo de classificação sobre um dado conjunto de padrões, ao mostrar o número de classificações corretas *versus* as classificações preditas para cada classe.

Seja a matriz de confusão de um problema de classificação com duas classes conforme figura 3.7.

Classe	predita $C_+$ predita $C_-$		Taxa de Erro da Classe	Taxa de Erro Total
verdadeira $C_+$	$T_p$	$F_N$	$\frac{F_N}{T_p + F_N}$	$\frac{F_p + F_N}{n}$
verdadeira $C_-$	$F_p$	$T_N$	$\frac{F_p}{F_p + T_N}$	

Figura 3.7 – Matriz de confusão.

onde:

- $T_p$ : Verdadeiro positivo
- $F_N$ : Falso negativo
- $F_p$ : Falso positivo
- $T_N$ : Verdadeiro negativo
- $n$ : total de padrões

O número de acertos, para cada classe, se localiza na diagonal principal da matriz e os demais elementos da matriz representam erros na classificação.

O percentual total de acerto ponderado pela classe 1 é a razão entre o total de acertos e o total de padrões ambos ponderados pela classe 1. Este fator de ponderação representa o quanto a classe 1 é maior que cada uma das demais classes. Generalizando para mais de duas classes, a função de avaliação baseada neste indicador é definida por (Pinho10):

$$\frac{\left( \sum_{i=j=1}^n a_{ij} \cdot r_{c1/ci} \right)}{\left( \sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot r_{c1/ci} \right)} \quad (3.14)$$

onde  $a_{ij}$  ( $i=j$ ) representa o total de acertos da classe  $c_i$  e  $r_{c1/ci}$  é a razão entre o total de padrões da classe  $c_1$  e o total de padrões da classe  $c_i$ .

O percentual total de acerto não ponderado é a razão entre o total de acertos e o total de padrões. Generalizando para mais de duas classes, a função de avaliação baseada neste indicador é definida por (Pinho10):

$$\frac{\left( \sum_{i=j=1}^n a_{ij} \right)}{\left( \sum_{i=1}^n \sum_{j=1}^n a_{ij} \right)} \quad (3.15)$$

onde  $a_{ij}$  ( $i=j$ ) representa o total de acertos da classe  $c_i$ .

A análise ROC (*Receiver Operating Characteristic*) teve a sua origem na teoria de detecção de sinal. Ela é baseada em dois conceitos: sensibilidade, que é a fração de padrões classificados como verdadeiros positivos (FVP), e especificidade, que é a fração de padrões classificados como falsos positivos (FFP). Este tipo de análise se aplica a problemas de classificação com somente duas classes e por esta razão, quando a função de avaliação for a curva ROC, somente um neurônio será considerado na camada de saída do cromossoma (Pinho10). A função de avaliação baseada neste indicador é definida pela equação 3.16, que é a área sob a curva ROC pela regra do trapézio:

$$\sum_{i=1}^n \left\{ (1 - (1 + FVP_i) \cdot \Delta FFP) + \frac{1}{2} [\Delta FVP \cdot \Delta FFP] \right\} \quad (3.16)$$

onde:

- $n$ : número de variações do ponto de corte
- $\Delta FVP = FVP_i - FVP_{i-1}$
- $\Delta FFP = FFP_i - FFP_{i-1}$

## 4. Técnicas de Regularização de Complexidade

Uma questão prática que existe no contexto de redes neurais é a busca pela minimização do tamanho da rede mantendo-se o seu desempenho. A minimização do tamanho da rede implica na redução do número de seus parâmetros livres. Uma rede com muitos parâmetros tende a aprender também o ruído dos dados de treinamento o que pode ocasionar uma resposta incorreta aos padrões nunca vistos. Em uma rede muito complexa, a remoção de alguns pesos pode não aumentar o erro significativamente, indicando que estes pesos têm pouca ou nenhuma influência sobre a rede.

Dentre as técnicas de regularização de complexidade de redes neurais artificiais pode-se destacar (Haikin00):

1. Decaimento de pesos (*weight decay*);
2. Eliminação de pesos;
3. Suavizador aproximativo;
4. Análise de sensibilidade.

A técnica utilizada neste trabalho foi a de decaimento de pesos. Neste método, um termo de punição da complexidade correspondente a norma quadrática do vetor de pesos da rede é adicionado à função de erro (equação 4.1),

$$E(w) = E_0(w) + \frac{1}{2}\lambda \sum_i w_i^2 \quad (4.1)$$

onde  $\lambda$  é a taxa de decaimento dos pesos (parâmetro entre 0 e 1).

O termo de penalidade faz com que os pesos insignificantes assumam valores próximos de zero, enquanto permite que os outros pesos mantenham seus valores relativamente elevados. Os pesos que tem pequena ou baixa influência sobre a rede (pesos com valores próximos a zero) são ditos pesos excessivos. Na ausência de regularização da complexidade, estes pesos resultam em uma generalização pobre em virtude de causar um ajuste excessivo dos dados da rede.

No modelo neuroevolutivo com decaimento de pesos definiu-se o termo de punição de complexidade como o produto entre o vetor de pesos da rede ( $w$ ) e a constante de decaimento ( $\lambda$ ). Este termo é decrementado do vetor de pesos a cada treinamento da rede (equação 4.2). A cada geração, para cada um dos indivíduos clássicos, aplica-se a regra de decaimento e treina-se a rede neural definida por este indivíduo.

$$w = w - \lambda w \quad (4.2)$$

## 5. Estudos de Caso

Neste capítulo são descritos os casos de *benchmark* de problemas de classificação de padrões utilizados para medir o desempenho do algoritmo neuroevolutivo NEIQ-BR com decaimento de pesos. Os resultados obtidos pelo modelo NEIQ-BR com decaimento de pesos foram comparados aos obtidos por (Pinho10).

Os estudos de caso utilizados neste trabalho foram:

- *Australian Credit*
- *Wisconsin Prognostic Breast Cancer*
- *Iris Data*

Conforme descrito em (Pinho10), as bases de dados selecionadas sofreram um pré-processamento para tratamento de observações faltantes e normalização de variáveis numéricas.

No caso de tratamento de observações faltantes, para as variáveis numéricas foi utilizada a média enquanto que para as variáveis categóricas foi utilizada a moda.

Variáveis numéricas relacionadas a tempo decorrido, como por exemplo idade, foram normalizadas com base na regra de divisão pela faixa máxima de valores (equação 5.1).

$$x_{ij}^{normalizado} = \frac{x_{ij} - \min(x_i)}{\max(x_i) - \min(x_i)} \quad (5.1)$$

Para as demais variáveis numéricas utilizou-se a normalização padrão, com base na média e desvio padrão (equação 5.2).

$$x_{ij}^{normalizado} = \frac{x_{ij} - \bar{X}}{\sigma_X} \quad (5.2)$$

As variáveis discretas com duas categorias foram representadas por duas variáveis binárias enquanto que aquelas com mais de duas categorias foram transformadas conforme codificação 1 de N.

### 5.1 *Australian Credit*

Esta base de dados consiste de uma aplicação típica de tomada de crédito, que tem como objetivo prever bons e maus pagadores. Ela apresenta um total de 14 variáveis preditivas, sendo 8 categóricas e 6 numéricas e 690 registros, sendo 307 para bons pagadores, e 383 para maus pagadores. Em 37 dos 690 registros existe pelo menos um



valor faltante em uma das variáveis preditivas (Pinho10). A tabela 5.1 apresenta um resumo das principais características desta base de dados.

<i>Base de Dados</i>	<i>Amostras</i>	<i>Variáveis Binárias</i>	<i>Variáveis Numéricas</i>	<i>Número Classes</i>
<i>Australian Credit Approval</i>	690	32	30	2

Tabela 5.1 – Base de dados *Australian Credit Approval*.

## 5.2 Wisconsin Prognostic Breast Cancer

Esta base de dados tem como objetivo a identificação prévia da possibilidade de recorrência de um câncer de mama, ou seja, um prognóstico. Ela apresenta 33 variáveis numéricas preditivas, medidas através de imagens digitalizadas e 198 registros de pacientes. Destes, 47 são casos reincidentes e 151 são não reincidentes. Em 4 dos 198 registros existe pelo menos um valor faltante em uma das variáveis preditivas. A tabela 5.2 apresenta um resumo das principais características desta base de dados.

<i>Base de Dados</i>	<i>Amostras</i>	<i>Variáveis Binárias</i>	<i>Variáveis Numéricas</i>	<i>Número Classes</i>
<i>Prognostic Breast Cancer</i>	198	-	33	2

Tabela 5.2 – Base de dados *Prognostic Breast Cancer*.

## 5.3 Iris Data

Esta base de dados consiste de três tipos de planta Iris, cada tipo contendo 50 amostras. Ela apresenta 4 variáveis numéricas preditivas e nenhum valor faltante. A tabela 5.3 apresenta um resumo das principais características desta base de dados.

<i>Base de Dados</i>	<i>Amostras</i>	<i>Variáveis Binárias</i>	<i>Variáveis Numéricas</i>	<i>Número Classes</i>
<i>Iris Data</i>	150	-	4	3

Tabela 5.3 – Base de dados *Iris Data*.

## 5.4 Experimentos

Como o objetivo deste trabalho foi o de comparar o desempenho do algoritmo NEIQ-BR com decaimento de pesos com o algoritmo NEIQ-BR original, todas as bases de dados selecionadas foram avaliadas através do processo de validação cruzada particionado em 3, *3-fold cross-validation*, mesmo procedimento adotado por (Pinho10).

O processo de validação cruzada *3-fold cross-validation* consiste do particionamento das bases de dados em três partes, cada uma contendo 33% da amostra.

Em cada uma das validações, duas partições são utilizadas para treinamento, e o grupo remanescente utilizado para testar a generalização do modelo. Cada estratégia evolutiva foi avaliada através do desempenho médio obtido entre as 3 partições do grupo de teste considerando uma das seguintes funções de avaliação:

- Percentual total de acerto ponderado pela classe 1 (1);
- Percentual total de acerto (2); e
- Área abaixo da curva ROC (3).

Os três casos de *benchmark* descritos nas seções 5.1, 5.2 e 5.3 foram analisados considerando as 12 estratégias evolutivas definidas por (Pinho10) e cujas configurações constam da tabela 5.4.

Parâmetro	Estratégia										
	1	2	3	4	5	6	7	8	9	10	12
<i>TI</i>	30	20	40	30	20	40	30	20	40	30	40
<i>NQ</i>	30	15	10	30	15	5	30	15	10	30	5
<i>NC</i>	30	60	100	30	15	200	30	60	100	30	200
<i>T</i>	200	200	200	200	200	200	200	200	200	200	200
<i>C<sub>cb</sub></i>	0,92	0,95	0,98	0,99	0,95	0,98	0,92	0,95	0,98	0,99	0,98
<i>C<sub>cr</sub></i>	0,08	0,05	0,02	0,01	0,05	0,02	0,08	0,05	0,02	0,01	0,02
<i>θ</i>	0,05 0* <i>pi</i>	0,080* <i>pi</i>	0,020* <i>pi</i>	0,050* <i>pi</i>	0,080* <i>pi</i>	0,020* <i>pi</i>	0,050* <i>pi</i>	0,080* <i>pi</i>	0,020* <i>pi</i>	0,050* <i>pi</i>	0,020* <i>pi</i>
<i>C<sub>o</sub></i>	0,20	0,10	0,30	0,20	0,10	0,30	0,20	0,10	0,30	0,20	0,30
<i>minT</i>	2	2	2	2	2	2	2	2	2	2	2
<i>updateT</i>	4	4	4	4	4	4	4	4	4	4	4
<i>Função de avaliação</i>	(1)	(1)	(1)	(1)	(1)	(1)	(3)	(3)	(3)	(3)	(3)
<i>Experimento (AG)</i>	1	1	1	1	1	1	1	1	1	1	1

Tabela 5.4 – Estratégias evolutivas analisadas.

Para todas as estratégias evolutivas consideraram-se os seguintes valores para a taxa de decaimento dos pesos ( $\lambda$ ):

Parâmetro	Valor		
$\lambda$	0,001	0,005	0,01

Tabela 5.5 – Valores da taxa de decaimento dos pesos.

Os resultados dos diversos experimentos foram comparados em termos da média do percentual de padrões classificados erroneamente - MPPCE, e do desvio padrão de padrões classificados erroneamente - DPPCE, do grupo de teste (Pinho10).

Define-se percentual de padrões classificados erroneamente como a razão entre o total de padrões classificados erroneamente e o total de padrões. O MPPCE é a média dos percentuais de padrões classificados erroneamente das 3 partições do grupo de teste. Já o DPPCE é a diferença entre o maior e o menor percentual de padrões classificados erroneamente das 3 partições do grupo de teste.

## 6. Resultados

O código fonte dos arquivos abaixo relacionados foram alterados para implementação da rotina de decaimento de pesos e correção dos erros identificados no algoritmo original.

- *aieqNN.m*
- *outputs.m*
- *evoluiTESTEwpbc.m*
- *evoluiTESTEiris.m*
- *evoluiTESTEaustralian.m*
- *evoluiRESULTADOwpbc.m*
- *evoluiRESULTADOiris.m*
- *evoluiRESULTADOaustralian.m*

Os experimentos definidos no capítulo 5 foram executados e os resultados comparados aos obtidos por (Pinho10).

### 6.1 Erro de Codificação Identificado no Algoritmo de (Pinho10)

Analisando o código fonte do arquivo *outputs.m* identificou-se que o mesmo não totalizava corretamente o número de variáveis na camada de entrada, o número de neurônios na camada escondida, o número de neurônios com ativação tangente hiperbólica e o número de neurônios com ativação sigmóide.

A figura 6.1 apresenta parte do código fonte do arquivo *outputs.m* desenvolvido por (Pinho10). Os trechos assinalados em vermelho estão com erro em sua codificação.

```

% TESTANDO A REDE %

net = x(1:wsiz);
sol = x;

numtanhhide=0;
for i=1:nh
    if sol(wsize+ni+nh+i:wsize+ni+nh+i)==1 & sol(wsize+(ni-1)+i:wsize+(ni-1)+i)==1
        numtanhhide = 1+numtanhhide;
    end
end

numsighide=0;
for i=1:nh
    if sol(wsize+ni+nh+i:wsize+ni+nh+i)==0 & sol(wsize+(ni-1)+i:wsize+(ni-1)+i)==1
        numsighide = 1+numsighide;
    end
end

numtangout=0;
for i=1:no
    if sol(wsize+ni+nh+nh+i:wsize+ni+nh+nh+i)==1
        numtangout = sol(wsize+ni+nh+nh+i:wsize+ni+nh+nh+i)+numtangout;
    end
end
numsigout=no-numtangout;

disp(' ');
disp('Configuração da Rede:');
disp(['Número de variáveis na camada de entrada (sem o bias): ' num2str(sum(sol(wsize+1:wsize+(ni-1))))]);
disp(['Número de neurônios na camada escondida (sem o bias): ' num2str(sum(sol(wsize+(ni-1)+1:wsize+(ni-1)+nh))))]);
disp(['Neurônios com ativação "Tangente Hiperbólica" na escondida: ' num2str(numtanhhide)]);
disp(['Neurônios com ativação "Sigmóide" na escondida: ' num2str(numsighide)]);
disp(['Neurônios com ativação "Tangente Hiperbólica" na saída: ' num2str(numtangout)]);
disp(['Neurônios com ativação "Sigmóide" na saída: ' num2str(numsigout)]);

```

Figura 6.1 – Trecho do código fonte de *outputs.m* (Pinho10).

O novo código fonte é apresentado na figura 6.2. As alterações realizadas estão assinaladas em azul.

```

% TESTANDO A REDE %

net = x(1:wsiz);
sol = x;

numtanhhide=0;
for i=1:nh
    if sol(wsize+ni+nh+i:wsize+ni+nh+i)==1 & sol(wsize+ni+i:wsize+ni+i)==1
        numtanhhide = 1+numtanhhide;
    end
end

numsighide=0;
for i=1:nh
    if sol(wsize+ni+nh+i:wsize+ni+nh+i)==0 & sol(wsize+ni+i:wsize+ni+i)==1
        numsighide = 1+numsighide;
    end
end

numtangout=0;
for i=1:no
    if sol(wsize+ni+nh+nh+i:wsize+ni+nh+nh+i)==1
        numtangout = sol(wsize+ni+nh+nh+i:wsize+ni+nh+nh+i)+numtangout;
    end
end
numsigout=no-numtangout;

disp(' ');
disp('Configuração da Rede:');
disp(['Número de variáveis na camada de entrada (sem o bias): ' num2str(sum(sol(wsize+1:wsize+ni))))];
disp(['Número de neurônios na camada escondida (sem o bias): ' num2str(sum(sol(wsize+ni+1:wsize+ni+nh))))];
disp(['Neurônios com ativação "Tangente Hiperbólica" na escondida: ' num2str(numtanhhide)]);
disp(['Neurônios com ativação "Sigmóide" na escondida: ' num2str(numsighide)]);
disp(['Neurônios com ativação "Tangente Hiperbólica" na saída: ' num2str(numtangout)]);
disp(['Neurônios com ativação "Sigmóide" na saída: ' num2str(numsigout)]);

```

Figura 6.2 – Trecho do novo código fonte de *outputs.m*.

A título de exemplo, considere o problema de classificação *Prognostic Breast Cancer* e a estratégia de evolução 1. Os valores das variáveis *ni* (número de variáveis de entrada), *nh* (número de neurônios na camada escondida), *no* (número de classes) e *wsiz* (tamanho da parte real de um indivíduo clássico) constam da tabela 6.1.

<i>Variável</i>	<i>Valor</i>
<i>ni</i>	33
<i>nh</i>	30
<i>no</i>	2
<i>wsiz</i>	1.082

Tabela 6.1 – Variáveis *ni*, *nh*, *no* e *wsiz*.

Cada indivíduo clássico apresenta 1.177 genes dos quais, 1.082 são reais e 95 são binários.

A estrutura do cromossoma para o problema de classificação *Prognostic Breast Cancer* e estratégia de evolução 1 está definida na tabela 6.2.

<i>Gene</i>	<i>Tipo</i>	<i>Descrição</i>
1 até 1.082	Real	Pesos da RNA incluindo o viés.
1.083 até 1.115	Binário	Variáveis de entrada ativas ou não.
1.116 até 1.145	Binário	Neurônios da camada escondida ativos ou não.
1.146 até 1.175	Binário	Função de ativação de cada neurônio da camada escondida.
1.176 até 1.177	Binário	Função de ativação de cada neurônio da camada de saída.

Tabela 6.2 – Estrutura do cromossoma para o problema de classificação *Prognostic Breast Cancer* e estratégia de evolução 1.

A tabela 6.3 apresenta a definição do menor e maior índices das partes real e binária de um cromossoma em função das variáveis *ni*, *nh*, *no* e *wsiz*.

<i>Gene</i>	
1 até 1.082	1 até ( <i>wsiz</i> )
1.083 até 1.115	( <i>wsiz</i> + 1) até ( <i>wsiz</i> + <i>ni</i> )
1.116 até 1.145	( <i>wsiz</i> + <i>ni</i> + 1) até ( <i>wsiz</i> + <i>ni</i> + <i>nh</i> )
1.146 até 1.175	( <i>wsiz</i> + <i>ni</i> + <i>nh</i> + 1) até ( <i>wsiz</i> + <i>ni</i> + <i>nh</i> + <i>nnh</i> )
1.176 até 1.177	( <i>wsiz</i> + <i>ni</i> + <i>nh</i> + <i>nh</i> + 1) até ( <i>wsiz</i> + <i>ni</i> + <i>nh</i> + <i>nh</i> + <i>no</i> )

Tabela 6.3 – Definição do menor e maior índices das partes real e binária de um cromossoma.

Pelo código fonte do arquivo *outpus.m* de (Pinho10), verifica-se que os intervalos considerados no cálculo do número de variáveis ativas na camada de entrada e número de neurônios ativos na camada escondida estão errados. Os intervalos são definidos pelas expressões (*wsiz* + 1) : (*wsiz* + *ni* - 1) e (*wsiz* + *ni* - 1 + 1) : (*wsiz* + *ni* + 1 + *nh*), respectivamente, ou seja, de 1.083 até 1.114 e 1.115 até 1.146 quando o correto é o definido na tabela 6.3, ou seja, de 1.083 até 1.115 e 1.116 até 1.145. O gene 1.115 representa uma variável de entrada e, no entanto era contabilizado no cálculo do total de neurônios ativos da camada escondida. Observação similar se aplica ao gene 1146, que era considerado no cálculo do total de neurônios ativos da camada escondida quando na verdade representa o tipo de função de ativação de um neurônio da camada escondida.

O total de neurônios da camada escondida com ativação tangente hiperbólica e ativação sigmóide também não está correto no código implementado por (Pinho10). No

arquivo *outputs.m* a expressão  $(wsize + ni - 1 + i) : (wsize + ni - 1 + i)$ , quando  $i$  igual a 1, referencia o gene 1.115, que indica se a variável de entrada está ativa ou não quando o correto é referenciar um gene que indique se o neurônio está ativo ou não. Desta forma, ao invés de avaliar os conteúdos dos genes 1.146 (função de ativação de neurônio da camada escondida) e 1.116 (neurônio ativo ou não da camada escondida), 1.147 (função de ativação de neurônio da camada escondida) e 1.117 (neurônio ativo ou não da camada escondida), e assim sucessivamente, até 1.175 e 1.145, aos pares, são avaliados os conteúdos dos genes 1.146 (função de ativação de neurônio da camada escondida) com 1.115 (variável de entrada ativa ou não), 1.147 (função de ativação de neurônio da camada escondida) com 1.116 (neurônio ativo ou não da camada escondida), até 1.175 com 1.144, ignorando-se o gene 1.145 (neurônio ativo ou não da camada escondida).

## 6.2 Resultados do Algoritmo NEIQ-BR (Pinho10)

As tabelas 6.4 e 6.5 apresentam, respectivamente, o MPPCE e o DPPCE obtidos por (Pinho10) para os problemas de classificação *Australian Credit Approval*, *Prognostic Breast Cancer* e *Iris Data*.

A pior e melhor estratégias, por problema, constam da tabela 6.6. A pior estratégia é aquela com maior MPPCE enquanto que a melhor estratégia é aquela com menor MPPCE.

<b>Problema</b>	<b>Estratégia</b>											
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<i>Australian Credit Approval</i>	12,8	11,2	13,0	16,5	13,0	12,3	6,2	6,6	6,1	6,8	6,2	6,2
<i>Prognostic Breast Cancer</i>	18,2	22,7	21,7	22,2	18,2	15,7	20,7	17,7	15,6	17,2	21,2	16,2
<i>Iris Data</i>	3,3	2,7	1,3	2,0	2,0	2,0	4,7	3,3	2,7	3,3	2,0	4,7

Tabela 6.4 – Média do percentual total de padrões classificados erroneamente através de processo de validação cruzada em 3 (valores em percentual) (Pinho10).

<b>Problema</b>	<b>Estratégia</b>											
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<i>Australian Credit Approval</i>	3,5	1,1	2,3	3,9	1,2	1,3	1,9	0,6	0,1	1,1	2,0	0,7
<i>Prognostic Breast Cancer</i>	1,5	9,9	3,8	3,8	6,9	4,9	0,9	4,6	4,9	4,9	2,6	3,2
<i>Iris Data</i>	1,2	1,2	2,3	2,0	0,0	2,0	3,1	2,3	1,2	3,1	0,0	1,2

Tabela 6.5 – Desvio padrão do percentual total de padrões classificados erroneamente através de processo de validação cruzada em 3 (valores em percentual) (Pinho10).

<b>Problema</b>	<b>Estratégia</b>	
	<b>Melhor Desempenho</b>	<b>Pior Desempenho</b>
<i>Australian Credit Approval</i>	9	4
<i>Prognostic Breast Cancer</i>	9	2
<i>Iris Data</i>	3	12

Tabela 6.6 – Estratégias com pior e melhor desempenho por problema de classificação (Pinho10).

Nas tabelas 6.7 e 6.8 são apresentados, respectivamente:

1. Percentual de acerto obtido por problema, por melhor estratégia, por grupos de treino e teste, por partição de validação cruzada, por classe predita; e
2. Topologia ótima obtida dos melhores indivíduos, das melhores estratégias, por problema, por partição de validação cruzada.

obtidos por (Pinho10).

<b>Problema</b>	<b>Partição</b>	<b>Treino</b>				<b>Teste</b>			
		<b>Acerto Classe 1</b>	<b>Acerto Classe 2</b>	<b>Acerto Classe 3</b>	<b>Acerto Total</b>	<b>Acerto Classe 1</b>	<b>Acerto Classe 2</b>	<b>Acerto Classe 3</b>	<b>Acerto Total</b>
<i>Australian Credit Approval</i>	1	84,2	90,3	-	93,8	90,0	87,0	-	94,0
	2	86,5	91,0	-	94,9	83,7	91,6	-	93,8
	3	85,1	86,8	-	94,0	87,5	87,3	-	94,0
<i>Prognostic Breast Cancer</i>	1	80,6	88,5	-	86,4	89,1	81,8	-	87,9
	2	88,5	73,6	-	76,5	73,3	90,5	-	78,8
	3	76,5	86,7	-	84,1	84,9	92,3	-	86,4
<i>Iris Data</i>	1	100,0	93,9	100,0	98,0	100,0	94,1	91,7	96,0
	2	100,0	97,1	91,4	96,0	100,0	100,0	100,0	100,0
	3	100,0	94,4	94,4	96,0	100,0	100,0	100,0	100,0

Tabela 6.7 – Percentual de acerto por problema, melhor estratégia, partição de validação cruzada, grupos de treinamento e teste, classe (Pinho10).

<b>Problema</b>	<b>Partição</b>	<b>Número Variáveis Camada Entrada</b>	<b>Número Neurônios Camada Escondida</b>	<b>Neurônios Ativação Tangente Hiperbólica Camada Escondida</b>	<b>Neurônios Ativação Sigmóide Camada Escondida</b>	<b>Neurônios Ativação Tangente Hiperbólica Camada Saída</b>	<b>Neurônios Ativação Sigmóide Camada Saída</b>
<i>Australian Credit Approval</i>	1	15	21	10	11	0	1
	2	20	27	11	16	0	1
	3	14	24	10	14	0	1
<i>Prognostic Breast Cancer</i>	1	24	22	14	8	1	0
	2	19	19	10	9	1	0
	3	18	22	12	10	1	0
<i>Iris Data</i>	1	1	16	8	8	2	1
	2	1	21	11	10	2	1
	3	2	21	13	8	1	2

Tabela 6.8 – Topologia da rede por problema, melhor estratégia e partição de validação cruzada (Pinho10).

### 6.3 Resultados do Algoritmo NEIQ-BR com Taxa de Decaimento de 0,001

As tabelas 6.9 e 6.10 apresentam, respectivamente, o MPPCE e o DPPCE obtidos com o algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,001.

<b>Problema</b>	<b>Estratégia</b>											
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<i>Australian Credit Approval</i>	12,6	12,3	12,8	16,7	14,2	11,3	13,0	12,6	12,3	14,1	14,0	13,3
<i>Prognostic Breast Cancer</i>	26,8	18,2	19,2	24,7	26,8	19,7	20,2	18,2	16,2	16,7	16,7	17,7
<i>Iris Data</i>	4,0	3,3	2,7	3,3	4,0	1,3	0,7	2,7	4,0	6,7	6,0	1,3

Tabela 6.9 – Média do percentual total de padrões classificados erroneamente através de processo de validação cruzada em 3 (valores em percentual).

<b>Problema</b>	<b>Estratégia</b>											
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<i>Australian Credit Approval</i>	3,5	1,7	5,7	8,3	0,9	2,6	1,7	3,5	0,4	1,3	2,2	2,6
<i>Prognostic Breast Cancer</i>	10,6	4,6	4,5	24,2	3,0	7,6	12,1	4,5	10,6	12,1	10,6	12,2
<i>Iris Data</i>	0,0	8,0	2,0	4,0	4,0	4,0	2,0	2,0	6,0	6,0	4,0	2,0

Tabela 6.10 – Desvio padrão do percentual total de padrões classificados erroneamente através de processo de validação cruzada em 3 (valores em percentual).

A pior e melhor estratégias, por problema, do algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,001 constam da tabela 6.11.

<b>Problema</b>	<b>Estratégia</b>	
	<b>Melhor Desempenho</b>	<b>Pior Desempenho</b>
<i>Australian Credit Approval</i>	6	4
<i>Prognostic Breast Cancer</i>	9	1
<i>Iris Data</i>	7	10

Tabela 6.11 – Estratégias com pior e melhor desempenho por problema de classificação.

As tabelas 6.12 e 6.13 apresentam, respectivamente:

1. Percentual de acerto obtido por problema, por melhor estratégia, por grupos de treino e teste, por partição de validação cruzada, por classe predita; e
2. Topologia ótima obtida dos melhores indivíduos, das melhores estratégias, por problema, por partição de validação cruzada.

obtidos com o algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,001.

<b>Problema</b>	<b>Partição</b>	<b>Treino</b>				<b>Teste</b>			
		<b>Acerto Classe 1</b>	<b>Acerto Classe 2</b>	<b>Acerto Classe 3</b>	<b>Acerto Total</b>	<b>Acerto Classe 1</b>	<b>Acerto Classe 2</b>	<b>Acerto Classe 3</b>	<b>Acerto Total</b>
<i>Australian Credit Approval</i>	1	82,2	91,3	-	86,1	87,4	91,9	-	89,6
	2	87,1	93,2	-	89,6	83,9	89,8	-	87,0
	3	88,0	90,0	-	88,9	88,8	90,6	-	89,6
<i>Prognostic Breast Cancer</i>	1	47,2	93,7	-	75,0	63,6	93,18	-	83,3
	2	73,7	95,8	-	89,4	52,0	95,1	-	78,8
	3	52,1	92,9	-	78,0	69,6	100,0	-	89,4
<i>Iris Data</i>	1	100,0	93,3	100,0	98,0	100,0	95,0	100,0	98,0
	2	100,0	96,9	94,3	97,0	100,0	100,0	100,0	100,0
	3	96,9	100,0	93,9	97,0	100,0	100,0	100,0	100,0

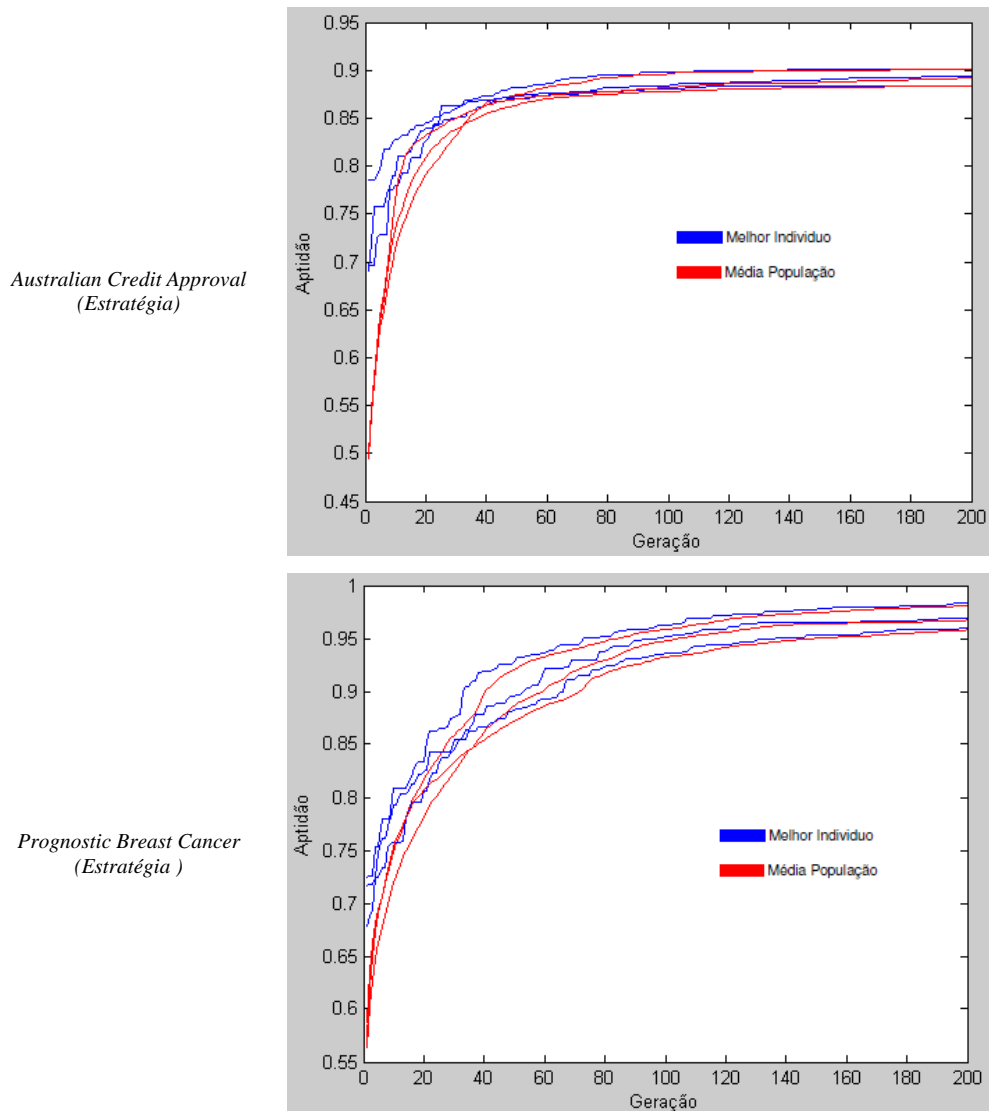
Tabela 6.12 – Percentual de acerto por problema, melhor estratégia, partição de validação cruzada, grupos de treinamento e teste, classe (NEIQ-BR com decaimento de pesos e taxa de decaimento de 0,001).



<i>Problema</i>	<i>Partição</i>	<i>Número Variáveis Camada Entrada</i>	<i>Número Neurônios Camada Escondida</i>	<i>Neurônios Ativação Tangente Hiperbólica Camada Escondida</i>	<i>Neurônios Ativação Sigmóide Camada Escondida</i>	<i>Neurônios Ativação Tangente Hiperbólica Camada Saída</i>	<i>Neurônios Ativação Sigmóide Camada Saída</i>
<i>Australian Credit Approval</i>	1	16	20	6	14	2	0
	2	17	16	7	9	0	2
	3	24	23	9	14	1	1
<i>Prognostic Breast Cancer</i>	1	19	27	13	14	0	1
	2	14	22	12	10	0	1
	3	11	20	7	13	1	0
<i>Iris Data</i>	1	2	16	12	4	0	3
	2	2	17	8	9	2	1
	3	3	19	9	10	0	3

Tabela 6.13 – Topologia da rede por problema, melhor estratégia e partição de validação cruzada (NEIQ-BR com decaimento de pesos e taxa de decaimento de 0,001).

A figura 6.3 apresenta a evolução de populações por melhor estratégia, por problema, por partição de validação cruzada e taxa de decaimento de 0,001.



*Iris Data*  
(Estratégia )

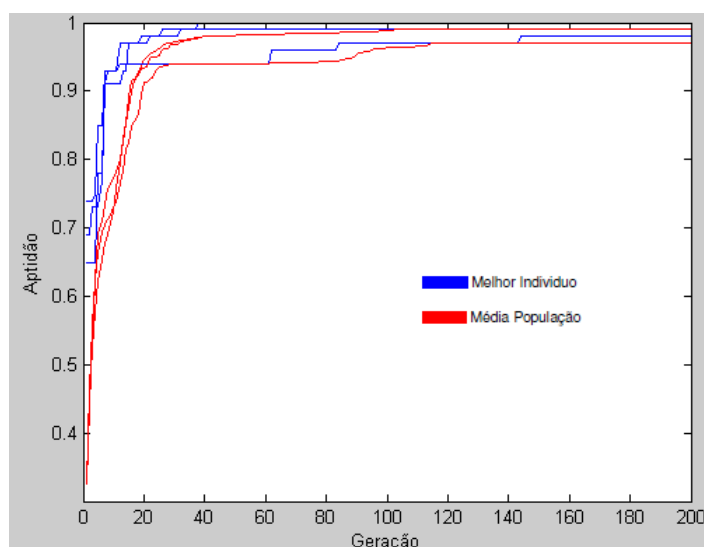


Figura 6.3 - Evolução de populações por melhor estratégia, por problema, por partição de validação cruzada e taxa de decaimento de 0,001.

## 6.4 Resultados do Algoritmo NEIQ-BR com Taxa de Decaimento de 0,005

As tabelas 6.14 e 6.15, por sua vez, apresentam, respectivamente, o MPPCE e o DPPCE obtidos com o algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,005.

<i>Problema</i>	<i>Estratégia</i>											
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
<i>Australian Credit Approval</i>	12,0	14,6	12,8	16,1	12,8	12,0	12,2	12,9	10,9	10,3	14,2	12,3
<i>Prognostic Breast Cancer</i>	21,2	18,2	20,2	36,4	24,7	20,2	20,2	11,6	15,2	21,7	16,2	20,2
<i>Íris Data</i>	1,3	3,3	2,0	4,7	4,7	2,0	3,3	3,3	2,0	2,7	2,0	3,3

Tabela 6.14 – Média do percentual total de padrões classificados erroneamente através de processo de validação cruzada em 3 (valores em percentual).

<i>Problema</i>	<i>Estratégia</i>											
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>	<i>11</i>	<i>12</i>
<i>Australian Credit Approval</i>	3,0	1,7	2,6	1,7	9,6	3,0	3,5	3,9	2,2	1,3	4,4	1,3
<i>Prognostic Breast Cancer</i>	15,2	10,6	4,5	15,2	10,6	12,1	10,6	6,1	6,1	10,6	4,6	13,6
<i>Íris Data</i>	2,0	6,0	6,0	2,0	6,0	0,0	2,0	2,0	0,0	2,0	6,0	4,0

Tabela 6.15 – Desvio padrão do percentual total de padrões classificados erroneamente através de processo de validação cruzada em 3 (valores em percentual).

A pior e melhor estratégias, por problema, do algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,005 constam da tabela 6.16.

<i>Problema</i>	<i>Estratégia</i>	
	<i>Melhor Desempenho</i>	<i>Pior Desempenho</i>
<i>Australian Credit Approval</i>	10	4
<i>Prognostic Breast Cancer</i>	8	4
<i>Íris Data</i>	1	4

Tabela 6.16 – Estratégias com pior e melhor desempenho por problema de classificação.

As tabelas 6.17 e 6.18, por sua vez, apresentam, respectivamente:

1. Percentual de acerto obtido por problema, por melhor estratégia, por grupos de treino e teste, por partição de validação cruzada, por classe predita; e
2. Topologia ótima obtida dos melhores indivíduos, das melhores estratégias, por problema, por partição de validação cruzada.

obtidos com o algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,005.

Problema	Partição	Treino				Teste			
		Acerto Classe 1	Acerto Classe 2	Acerto Classe 3	Acerto Total	Acerto Classe 1	Acerto Classe 2	Acerto Classe 3	Acerto Total
Australian Credit Approval	1	81,5	90,7	-	86,1	87,8	92,4	-	90,4
	2	88,2	89,9	-	89,1	86,1	92,6	-	89,6
	3	82,2	89,7	-	86,3	85,7	92,8	-	89,1
Prognostic Breast Cancer	1	72,2	95,8	-	89,4	73,3	88,2	-	84,8
	2	54,7	88,6	-	75,0	58,3	96,3	-	89,4
	3	49,0	92,6	-	75,8	75,0	97,8	-	90,9
Iris Data	1	96,7	93,8	94,7	95,0	100,0	100,0	100,0	100,0
	2	100,0	94,1	100,0	98,0	100,0	93,8	100,0	98,0
	3	100,0	93,8	96,7	97,0	100,0	100,0	95,0	98,0

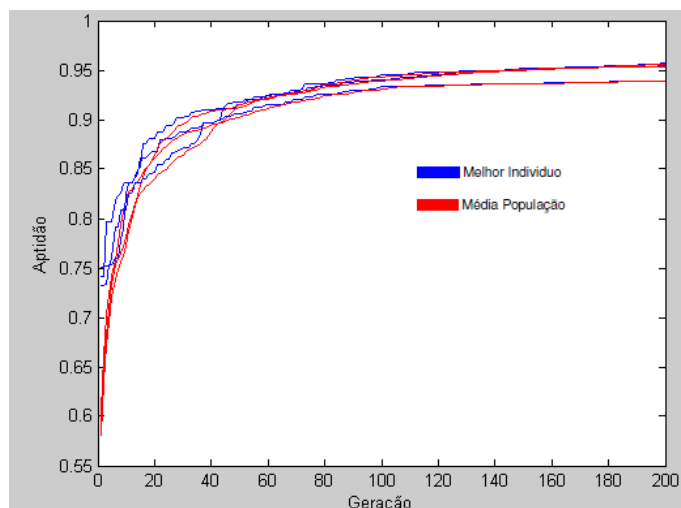
Tabela 6.17 – Percentual de acerto por problema, melhor estratégia, partição de validação cruzada, grupos de treinamento e teste, classe (NEIQ-BR com decaimento de pesos e taxa de decaimento de 0,005).

Problema	Partição	Número Variáveis Camada Entrada	Número Neurônios Camada Escondida	Neurônios Ativação Tangente Hiperbólica Camada Escondida	Neurônios Ativação Sigmóide Camada Escondida	Neurônios Ativação Tangente Hiperbólica Camada Saída	Neurônios Ativação Sigmóide Camada Saída
Australian Credit Approval	1	15	17	6	11	0	1
	2	18	11	8	3	0	1
	3	14	13	8	5	1	0
Prognostic Breast Cancer	1	13	12	8	4	1	0
	2	19	10	4	6	1	0
	3	18	14	8	6	0	1
Iris Data	1	4	16	8	8	1	2
	2	2	13	7	6	1	2
	3	3	14	7	7	2	1

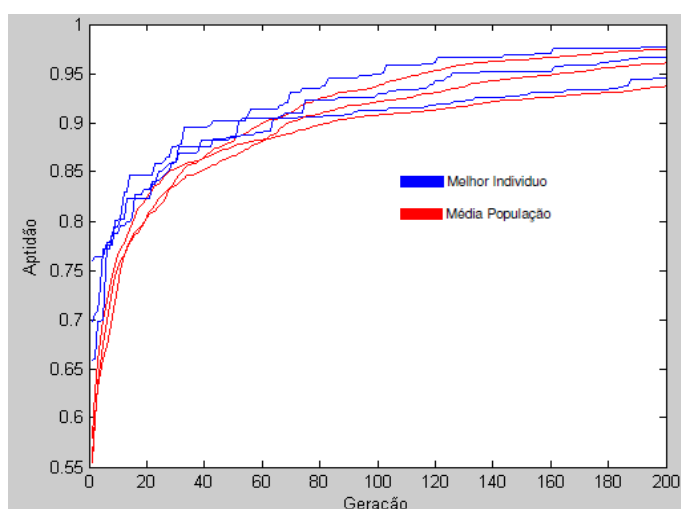
Tabela 6.18 – Topologia da rede por problema, melhor estratégia e partição de validação cruzada (NEIQ-BR com decaimento de pesos e taxa de decaimento de 0,005).

A figura 6.4 apresenta a evolução de populações por melhor estratégia, por problema, por partição de validação cruzada e taxa de decaimento de 0,005.

*Australian Credit Approval  
(Estratégia 10)*



*Prognostic Breast Cancer  
(Estratégia 8)*



*Iris Data  
(Estratégia 1)*

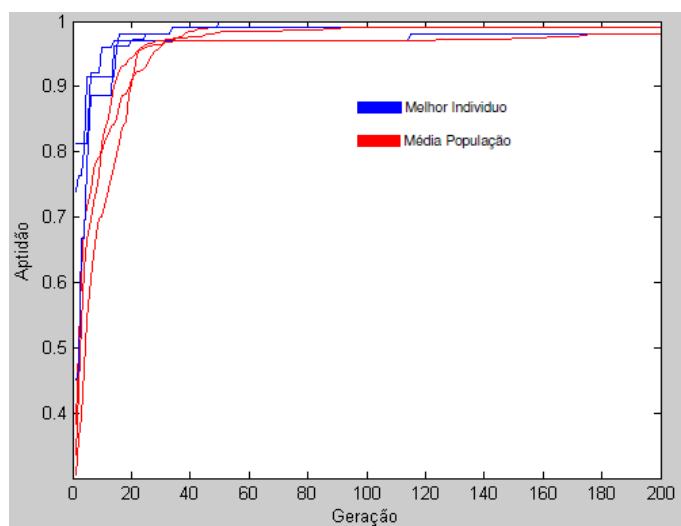


Figura 6.4 - Evolução de populações por melhor estratégia, por problema, por partição de validação cruzada e taxa de decaimento de 0,005.

## 6.5 Resultados do Algoritmo NEIQ-BR com Taxa de Decaimento 0,01

As tabelas 6.19 e 6.20 apresentam, respectivamente, o MPPCE e o DPPCE obtidos com o algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,01.

<b>Problema</b>	<b>Estratégia</b>											
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<i>Australian Credit Approval</i>	12,8	12,3	13,6	15,2	13,2	9,9	11,9	12,2	12,6	12,5	13,6	13,3
<i>Prognostic Breast Cancer</i>	17,2	15,7	15,2	34,8	20,7	21,2	21,7	21,2	14,6	18,2	22,8	22,7
<i>Íris Data</i>	3,3	2,7	2,7	5,3	2,0	3,3	4,7	1,3	2,7	1,3	3,3	8,7

Tabela 6.19 – Média do percentual total de padrões classificados erroneamente através de processo de validação cruzada em 3 (valores em percentual).

<b>Problema</b>	<b>Estratégia</b>											
	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>
<i>Australian Credit Approval</i>	3,5	5,2	6,1	7,0	1,3	0,4	4,7	3,0	2,6	2,6	1,3	5,6
<i>Prognostic Breast Cancer</i>	10,6	15,2	4,5	27,3	18,2	7,6	19,7	22,7	6,1	7,5	12,1	15,1
<i>Íris Data</i>	2,0	8,0	2,0	8,0	4,0	4,0	6,0	4,0	4,0	4,0	4,0	2,0

Tabela 6.20 – Desvio padrão do percentual total de padrões classificados erroneamente através de processo de validação cruzada em 3 (valores em percentual).

A pior e melhor estratégias, por problema, do algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,01 constam da tabela 6.21.

<b>Problema</b>	<b>Estratégia</b>	
	<b>Melhor Desempenho</b>	<b>Pior Desempenho</b>
<i>Australian Credit Approval</i>	6	4
<i>Prognostic Breast Cancer</i>	9	4
<i>Íris Data</i>	8	12

Tabela 6.21 – Estratégias com pior e melhor desempenho por problema de classificação.

As tabelas 6.22 e 6.23 apresentam, respectivamente:

1. Percentual de acerto obtido por problema, por melhor estratégia, por grupos de treino e teste, por partição de validação cruzada, por classe predita; e
2. Topologia ótima obtida dos melhores indivíduos, das melhores estratégias, por problema, por partição de validação cruzada.

obtidos com o algoritmo NEIQ-BR com decaimento de pesos para uma taxa de decaimento de 0,01.

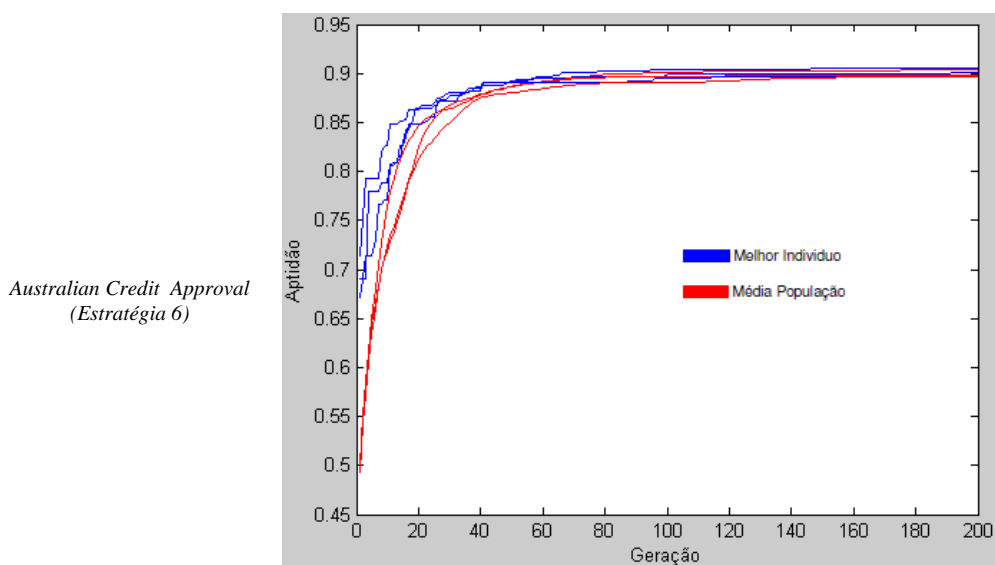
Problema	Partição	Treino				Teste			
		Acerto Classe 1	Acerto Classe 2	Acerto Classe 3	Acerto Total	Acerto Classe 1	Acerto Classe 2	Acerto Classe 3	Acerto Total
Australian Credit Approval	1	91,1	85,6	-	88,7	94,4	85,8	-	90,4
	2	82,1	91,9	-	86,3	87,6	92,7	-	90,0
	3	85,7	86,5	-	86,1	95,4	82,8	-	90,0
Prognostic Breast Cancer	1	70,6	92,9	-	87,1	75,0	92,0	-	87,9
	2	59,1	93,2	-	81,8	56,0	97,6	-	81,8
	3	65,7	95,9	-	87,9	76,2	91,1	-	86,4
Iris Data	1	100,0	100,0	92,9	98,0	100,0	100,0	100,0	100,0
	2	100,0	100,0	100,0	100,0	100,0	95,0	92,3	96,0
	3	100,0	88,6	100,0	96,0	100,0	100,0	100,0	100,0

Tabela 6.22 – Percentual de acerto por problema, melhor estratégia, partição de validação cruzada, grupos de treinamento e teste, classe (NEIQ-BR com decaimento de pesos e taxa de decaimento de 0,01).

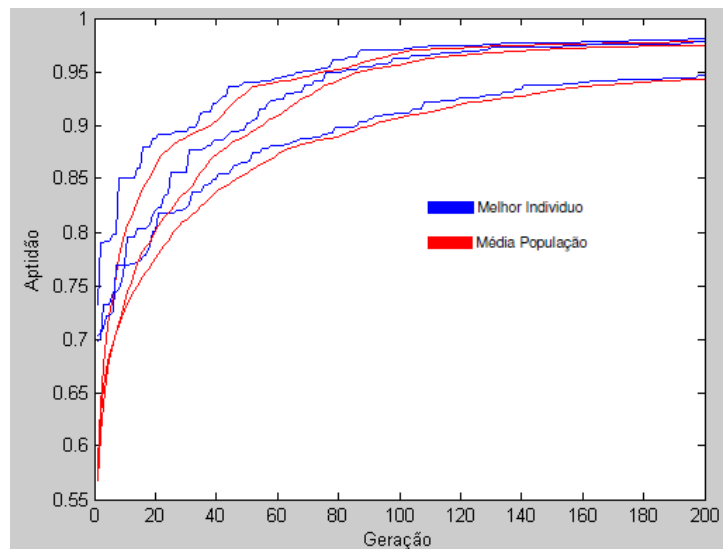
Problema	Partição	Número Variáveis Camada Entrada	Número Neurônios Camada Escondida	Neurônios Ativação Tangente Hiperbólica Camada Escondida	Neurônios Ativação Sigmóide Camada Escondida	Neurônios Ativação Tangente Hiperbólica Camada Saída	Neurônios Ativação Sigmóide Camada Saída
Australian Credit Approval	1	16	20	5	15	1	1
	2	16	23	11	12	1	1
	3	18	16	11	5	1	1
Prognostic Breast Cancer	1	16	21	13	8	1	0
	2	16	22	17	5	1	0
	3	17	19	6	13	1	0
Iris Data	1	4	12	9	3	0	3
	2	3	10	5	5	2	1
	3	2	11	4	7	2	1

Tabela 6.23 – Topologia da rede por problema, melhor estratégia e partição de validação cruzada (NEIQ-BR com decaimento de pesos e taxa de decaimento de 0,01).

A figura 6.5 apresenta a evolução de populações por melhor estratégia, por problema, por partição de validação cruzada e taxa de decaimento de 0,01 (conjunto de teste).



*Prognostic Breast Cancer*  
(Estratégia 9)



*Iris Data*  
(Estratégia 8)

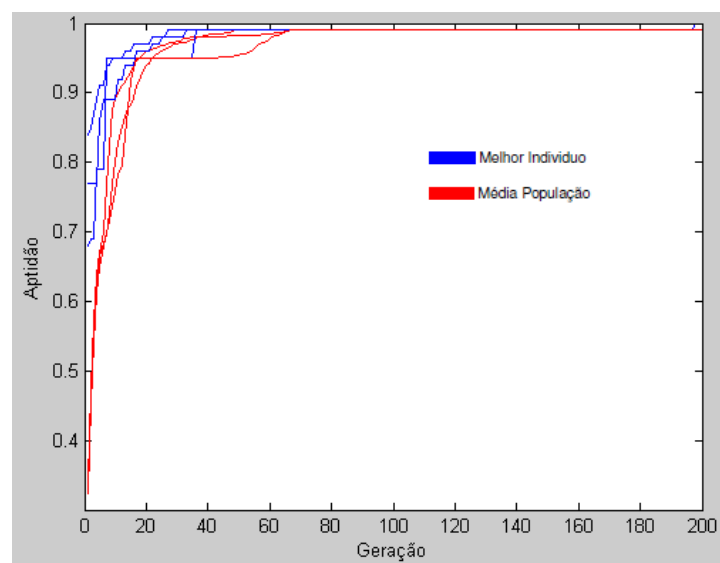


Figura 6.5 - Evolução de populações por melhor estratégia, por problema, por partição de validação cruzada e taxa de decaimento de 0,01.

## 7. Conclusão e Trabalhos Futuros

Comparando-se os resultados obtidos com os dois algoritmos (com e sem taxa de decaimento) e com diferentes taxas de decaimento conclui-se que:

1. A estratégia com pior desempenho em um maior número dos problemas analisados e para diferentes valores de taxa de decaimento foi a 4 corroborando o resultado obtido em (Pinho10), onde as estratégias com pior desempenho em um maior número de problemas foram a 4 e 10, que apresentam um baixo número de indivíduos clássicos (30) e um alto número de indivíduos quânticos (30).
2. O valor da taxa de decaimento de pesos a ser utilizada depende do problema em análise e sua escolha é obtida por experimentação. Como exemplo, para o estudo de caso *Prognostic Breast Cancer*, o melhor resultado foi obtido com uma taxa de decaimento de 0,005. Considerando o resultado da melhor estratégia em cada caso (algoritmo sem taxa de decaimento e com taxas de decaimento de 0,005; 0,01 e 0,001), verifica-se que, com uma taxa de decaimento de 0,005, houve um aumento no percentual médio de acerto total no grupo de teste e uma redução na complexidade da rede (redução do número de parâmetros livres da rede). Já para o problema *Iris Data*, o melhor resultado foi obtido com uma taxa de decaimento de 0,001 (aumento do percentual médio de acerto total no grupo de teste e redução do MPPCE). No entanto, para o estudo de caso *Australian Credit*, o melhor resultado obtido por (Pinho10) foi superior ao melhor resultado obtido com quaisquer uns dos valores utilizados para a taxa de decaimento. Porém, cabe ressaltar que com uma taxa de decaimento de 0,005 a rede neural obtida apresenta menor complexidade que a do algoritmo sem decaimento de pesos (menor número de parâmetros livres) tendo-se uma redução de apenas 3% no percentual médio de acerto total no grupo de teste.

Como trabalho futuro sugere-se aplicar o algoritmo com decaimento de pesos para os demais problemas de classificação propostos por (Pinho10) com diferentes taxas de decaimento.

Outra sugestão é avaliar se o desempenho do algoritmo seria melhorado se fosse definido, para todos os pesos da rede, um valor mínimo abaixo do qual o peso é descartado. Este novo parâmetro do algoritmo neuroevolutivo seria definido pelo



próprio usuário e esta verificação seria aplicada após a aplicação da técnica de decaimento dos pesos.

## Referências Bibliográficas

1. [Cruz07] CRUZ, A. V. A. *Algoritmos Evolutivos com Inspiração Quântica para Problemas com Representação Numérica*. Dissertação de Doutorado, Departamento de Engenharia Elétrica, PUC-Rio, Rio de Janeiro, RJ, Brasil, Mar. 2007.
2. [Haikin00] HAIKIN, S. S. *Redes Neurais – Princípios e Prática*. 2ª. edição, Bookman Companhia Editora, 2000.
3. [Han00] HAN, K.; KIM, J. *Genetic quantum algorithm and its application to combinatorial optimization problem*. In: PROCEEDINGS OF THE 2000 REFERÊNCIAS BIBLIOGRÁFICAS 88 CONGRESS ON EVOLUTIONARY COMPUTATION, volumen 2, p. 1354–1360, 2000. 2.5, 2.5, 2.5
4. [Moore95] MOORE, M.; NARAYANAN, A. *Quantum-inspired computing*, 1995. 2.4, 2.4, 2.4, 2.4
5. [Pinho10] PINHO, A. G. *Algoritmo Evolucionário com Inspiração Quântica e Representação Mista Aplicado a Neuroevolução*. Dissertação de Mestrado, Departamento de Engenharia Elétrica, PUC-Rio, RJ, Brasil, Out. 2010.