

# UMA ABORDAGEM EVOLUCIONÁRIA PARA A OTIMIZAÇÃO DE REDES RBF

RODRIGO G. F. SOARES, KELLY P. SILVA, FRANCISCO A. T. CARVALHO, TERESA B. LUDERMIR

*Centro de Informática, Universidade Federal de Pernambuco  
Caixa Postal 7851, CEP 50732-970 Recife (PE), Brazil  
{rgfs, kps, fatc, tbl}@cin.ufpe.br*

**Abstract**— One of the main obstacles to obtain an artificial neural network with reasonable performance is the parameter setting. This work proposes a methodology to the automatic definition of RBF (Radial Basis Function) networks with an appropriate configuration for the selected classification problems. We propose the use of a Memetic Algorithm in order to perform the search for networks with minimum architecture and error rate. A set of experiments was made with three data sets and we were able to show the effectiveness of the method.

**Keywords**— Evolutionary Algorithms, Neural Networks Optimization, Radial Basis Functions Networks.

**Resumo**— Um dos principais obstáculos para a obtenção de uma Rede Neural Artificial com boa performance é o ajuste de seus parâmetros. Este trabalho propõe uma metodologia para a definição automática de redes RBF (*Radial Basis Function*) com configurações adequadas aos problemas de classificação abordados. Propomos o uso de um Algoritmo Memético para realizar a busca por redes com arquiteturas e erros mínimos. Realizamos experimentos com três bases de dados e mostramos a eficácia do método.

**Palavras-chave**— Algoritmos Evolucionários, Otimização de Redes Neurais, Redes de Funções de Base Radial.

## 1 Introdução

Redes Neurais Artificiais (RNA) despertam grande interesse da comunidade científica devido às suas capacidades de mapear qualquer função com um grau arbitrário de acurácia, tal característica foi provada para redes neurais de Funções de Bases Radiais (RBF, do inglês *Radial Basis Function*) em [9]. As redes RBF são consideradas como uma alternativa melhorada às redes Perceptron Multi-Camadas (MLP, do inglês *Multi-Layer Perceptron*) no sentido de capacidade de generalização, eficiência computacional e plausibilidade biológica [11]. As RBF's são empregadas em diversas áreas: reconhecimento de padrões, previsão de séries temporais, controle e diversas aplicações matemáticas [9].

A definição de uma rede RBF consiste na escolha do número de unidades de processamento (com seu campo receptivo local), funções de ativação e largura do campo receptivo. A escolha do número de unidades de processamento é, geralmente, feita por tentativa e erro, pois, na maioria das vezes, não se sabe de antemão o número de centros ideal para o problema. Também não há um método efetivo para se encontrar a melhor disposição das várias possíveis funções de ativação dentre as unidades de processamento escondidas, sendo assim, essa definição baseada, também, em tentativa e erro. Existem várias heurísticas para a determinação da largura do campo receptivo, no entanto, na maioria das vezes, não se sabe a maneira mais adequada para o problema abordado. Tais escolhas dependem de um conhecimento *a priori* do problema, sem esse conhecimento, a definição de uma arquitetura ideal para a rede fica com-

prometida, baseando-se em tentativas e na experiência do projetista.

A fim de se encontrar um método automático para a geração de redes RBF com o melhor desempenho e com a menor complexidade (número de unidades de processamento empregado) possíveis, este trabalho propõe o uso de um Algoritmo Memético (AM) adaptado ao problema de otimização de redes RBF. Utilizamos, como procedimento de busca local, o algoritmo *k*-médias. O algoritmo proposto é detalhado nas próximas seções.

Os Algoritmos Meméticos representam um tipo de meta-heurística derivado dos Algoritmos Genéticos (AG) [8] - o primeiro possui um mecanismo de busca local em cada indivíduo da população [13]. Nessa estratégia, indivíduos podem evoluir autonomamente acrescentando unidades de informação cultural (*memes*) [3]. Várias aplicações de AM, em diversas áreas, têm mostrado que, ao custo de um maior esforço computacional despendido pelos operadores de busca local, a adoção de tais operadores fornece resultados melhores quando comparados àqueles produzidos com os AGs [6,7,13,15]. O desempenho encontrado em todos estes trabalhos se deve a um bom procedimento de busca local, geralmente específico para o problema que se quer resolver, a uma representação adequada para o cromossomo e ao operador de recombinação utilizado.

Buscamos, neste trabalho, obter um compromisso entre o desempenho da rede e a sua complexidade. Propomos o uso de uma média harmônica entre essas grandezas como a função objetivo (função de *fitness*) do AM. Deste modo, pretendemos, ao final do processo, obter uma rede RBF treinada e com arquitetura adequada ao problema atacado.

Este trabalho está organizado da seguinte forma:

a próxima seção discorre sobre trabalhos acerca deste tema presentes na literatura; a seção III descreve as redes RBF; a seção IV apresenta o Algoritmo Memético Proposto e a seção V apresenta os resultados obtidos em experimentos realizados com três problemas diferentes e as comparações com os resultados da literatura.

## 2 Abordagens Relacionadas

Encontramos, na literatura, diversos empregos de sistemas híbridos para aperfeiçoar redes neurais artificiais. Apresentamos aqui alguns trabalhos neste campo de pesquisa.

Ferentinos 2005 [5] abordou o problema de otimização de redes MLP através do uso de Algoritmo Genético. Ele usou uma codificação binária para representar a topologia da rede (número de unidades de processamento distribuídas nas camadas escondidas) completamente conectada, a função de ativação usada nas unidades e o algoritmo de minimização. Essa abordagem gerava a melhor rede MLP a partir de um conjunto predefinido de arquiteturas. Ele não incorporava a complexidade da rede na função objetivo do AG. O trabalho obteve bons resultados para problemas de engenharia biológica.

Yamazaki 2004 [18] propôs uma metodologia híbrida que une as características das meta-heurísticas *Simulated Annealing*, Busca Tabu e Algoritmo Genético para otimizar pesos e arquitetura de redes MLP. Como função objetivo, ele utilizou uma média aritmética entre o erro de classificação para o conjunto de treinamento e a porcentagem de conexões da rede. O desempenho de sua abordagem foi comparado com os de abordagens que apenas utilizavam uma dessas heurísticas, obtendo resultados melhores. A função objetivo idealizada não denota o compromisso entre desempenho e complexidade, apenas tenta maximizar tal função sem se preocupar com a variação dos seus parâmetros.

Zanchettin 2006 [19] elaborou uma estratégia que otimiza redes MLP através de um algoritmo que usa idéias de várias meta-heurísticas. O algoritmo melhora a complexidade das redes inserindo novas unidades de processamento e removendo-as através da poda. Como Yamazaki 2004 [18], ele usa uma média aritmética entre desempenho e complexidade da rede, o que não agrega a informação sobre o compromisso entre tais grandezas. Seus resultados foram comparados com os resultados obtidos com uma MLP treinada com *backpropagation*. A abordagem obteve melhor desempenho do que a rede MLP nos vários problemas de classificação.

Barreto 2002 [1] propôs uma metodologia para projeto e treinamento de redes RBF usando AG. Ele introduz um novo operador de cruzamento, uma inicialização minimalista que faz com que o algoritmo busque a simplicidade das redes e uma reformulação da Validação Cruzada Generalizada (GCV, do inglês *Generalized Cross-Validation*) que torna essa

medida mais rigorosa para redes complexas. O método tenta otimizar o número de unidades de processamento e seus parâmetros. Os pesos da camada de saída são ajustados com uso da matriz pseudo-inversa. A principal desvantagem deste método é o tempo computacional gasto.

A abordagem de Lacerda 2000 [12] também otimiza redes RBF com o uso de AG. O trabalho utiliza uma única largura do campo receptivo para todos os centros e tenta otimizar o número de centros, bem como suas localizações. Para o ajuste dos pesos da camada de saída, usou-se a computação da matriz pseudo-inversa. Essa abordagem particiona o espaço de entrada em hipercubos a fim de que haja centros em todas as regiões desse espaço. Esse trabalho utiliza duas funções objetivo: a GCV e a soma dos erros quadráticos. O trabalho conseguiu resultados tão bons quanto os de trabalhos anteriores. Conseguiu ainda mostrar que AG obtém redes com boa generalização, enquanto minimiza o número de unidades de processamento.

Todas essas abordagens utilizaram AG, o que proporciona um resultado global bom, mas não há uma busca local para os centros e larguras dos campos receptivos - o que pode tornar o desempenho inferior a um método que realize uma otimização local juntamente com a global. Nosso trabalho faz uso de um Algoritmo Memético, que realiza tanto a busca global quanto a local.

## 3 Redes de Função de Base Radial

Redes de Funções de Bases Radiais são redes multicamadas em que a ativação dos nodos intermediários é função da distância entre o vetor de entrada e os pesos do nodo [2].

Uma rede RBF consiste basicamente de uma camada de entrada, uma camada oculta e uma camada de saída [10]. Cada camada de uma rede RBF desempenha um papel específico. A camada de entrada apenas transfere o vetor de entrada para a camada oculta. A camada oculta da rede, cujos nodos utilizam funções de base radiais, transforma um conjunto de padrões de entrada não-linearmente separáveis em um conjunto de saídas linearmente separáveis. A terceira camada, a camada de saída, geralmente executa uma combinação linear das saídas da camada oculta.

A saída de uma RBF é um vetor  $m$ -dimensional cujo  $l$ -ésimo elemento é dado por:

$$y_l(x) = \sum_{j=1}^m w_{lj} z_j(x) \quad (1)$$

Onde  $x \in \mathbb{R}^n$ ,  $\{w_{lj}; j = 1, \dots, m\}$  denota os pesos e  $z_j(x)$  representa a função de ativação, conhecida como função de base radial definida pela equação a seguir:

$$z_j = \phi\left(\frac{\|x - \mu_j\|}{\sigma_j}\right) \quad (2)$$

Onde  $\|\cdot\|$  é a norma Euclidiana,  $\mu_j = [\mu_{j1}, \mu_{j2}, \dots, \mu_{jn}]$  é o centro do nodo  $j$ ,  $\sigma_j$  é a largura do campo receptivo do nodo e  $\phi(\cdot)$  é uma função de base radial.

Funções de Base Radiais representam uma classe especial de funções não-lineares cujo valor aumenta ou diminui de acordo com a distância a um ponto central [2]. Diferentes funções de base radiais têm sido utilizadas em redes RBF. As funções de base radiais mais comuns são:

- Função de Base Gaussiana

$$z_j(x) = \exp\left(-\frac{\|x - \mu_j\|^2}{2\sigma_j^2}\right) \quad (3)$$

- Função de Base Logística

$$z_j(x) = \left[1 + \exp\left(\frac{\|x - \mu_j\|}{\sigma_j} - \theta_j\right)\right]^{-1} \quad (4)$$

Onde  $\theta_j$  é um *bias* ajustável.

- Função Multiquadrática Inversa

$$z_j(x) = \frac{1}{\left(\sqrt{\|x - \mu_j\|^2 + \sigma_j^2}\right)} \quad (5)$$

- Função Thin-Plate-Spline

$$z_j(x) = \|x - \mu_j\|^2 \times \ln\left(\|x - \mu_j\|\right) \quad (6)$$

O treinamento de uma rede RBF com os nodos de saídas com funções de ativação lineares é realizado em dois estágios. O primeiro estágio é não-supervisionado e consiste na determinação do centro e da largura do campo receptivo de cada nodo da camada oculta [2]. O método mais utilizado para a determinação dos centros é o algoritmo de  $k$ -médias [4]. O segundo estágio do treinamento ajusta os pesos dos nodos de saída. Estes pesos podem ser determinados por modelos lineares, como a regra delta.

#### 4 O Algoritmo Proposto

O algoritmo proposto neste trabalho é inspirado em um algoritmo evolucionário conhecido na literatura como Algoritmo Memético (AM). O principal aspecto a se considerar em um sistema como esse é a codificação de um fenótipo em um genótipo específico. Em nosso caso, um fenótipo consiste de uma rede RBF, composta por um conjunto de nodos ocultos, onde cada nodo possui um centro, a largura de seu campo receptivo e a função de ativação utilizada. Um genótipo é um vetor que possui valores binários e valores reais. A representação será melhor esplanada na subseção seguinte.

A representação utilizada incorpora informações sobre todos os nodos ocultos da rede RBF. Cada uni-

dade de processamento é representada como na Figura 1.

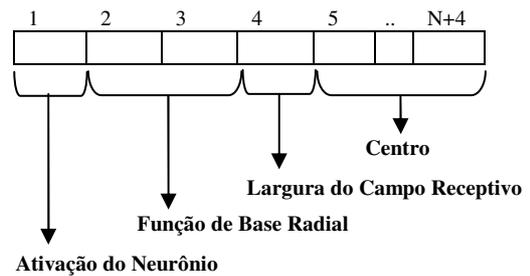


Figura 1 - Representação de um nodo oculto da RBF

Cada cromossomo será constituído por um vetor de estruturas como a apresentada na Figura 1, este vetor possui tamanho máximo determinado previamente.

A primeira posição da estrutura indica se o nodo está ativo (1) ou inativo (0). O número de nodos ocultos da RBF é a totalização dos nodos ativos do vetor.

A função de ativação do nodo é representada com duas entradas binárias (nas posições 2 e 3). As funções são mapeadas da seguinte maneira:

- 00: Função de Base Gaussiana
- 01: Função de Base Logística
- 10: Função Multiquadrática Inversa
- 11: Função Thin-Plate-Spline

A próxima posição da estrutura contém um valor real que representa a largura do campo receptivo do nodo. O centro, de dimensão  $N$ , será representado por valores reais nas  $N$  posições seguintes.

O algoritmo desenvolvido consiste de três partes principais: A parte “Nível do usuário”, a parte do “Algoritmo Memético” e a parte do “Treinamento da RBF”. A primeira parte lida com os processos de entrada/saída. O algoritmo é mostrado esquematicamente na Figura 2.

Inicialmente, o usuário fornece os parâmetros do algoritmo, que são: Conjunto de Treinamento da rede RBF ( $C_t$ ), Conjunto de Teste ( $C_n$ ), Conjunto de Validação ( $C_v$ ), um conjunto de parâmetros para o treinamento da RBF (*param*), o tamanho da população do AM ( $M$ ), o número de gerações do AM ( $G$ ), a probabilidade de cruzamento do AM ( $P_c$ ) e a probabilidade de mutação do AM ( $P_m$ ). A população inicial ( $X_{inicial}$ ) é gerada aleatoriamente. Todas essas informações são passadas para a função AM. A população inicial é então passada para o “Decodificador” que decodifica as informações dos indivíduos em seus conjuntos de raios (largura do campo receptivo), centros e funções de ativação. Estas informações são então passadas para a função “Busca Local”.

A busca local executa uma função de vizinhança, mais especificamente o algoritmo  $k$ -médias. O algoritmo de  $k$ -médias encontra, a partir dos centros gerados aleatoriamente, os centros mais apropriados para cada nodo e envia os raios, os novos centros e as funções de ativação para a função “Treinamento”.

O treinamento da camada de saída da RBF utiliza a regra delta [2]. Um dos critérios de parada do

treinamento é o erro obtido no conjunto de validação. Quando o erro de validação aumenta durante certo número de iterações, o treinamento é parado e assim, evita-se o *overfitting*. Um outro critério de parada para o treinamento é o número máximo de iterações definido. Após a realização do treinamento, a função de treinamento envia o tamanho da rede e seu desempenho no conjunto de teste para a função de *fitness*, onde o *fitness* de cada indivíduo é calculado.

O *fitness* é o valor que o AM tenta maximizar. Para manter um compromisso entre o desempenho da rede e sua complexidade, representada pelo número de nodos ocultos da RBF, utilizamos, para o cálculo do *fitness*, uma função baseada nas propriedades da média harmônica apresentada em [14]. O *fitness* é dado pela fórmula:

$$fitness(d, t) = (c + 1) \times d \times \left( \frac{t}{d + c \times t} \right) \quad (7)$$

onde  $d$  é a taxa de acerto da rede e  $t = 1 - (\text{tamanho} / \text{tamanho\_máximo})$ , onde *tamanho* é o número de nodos ativos da rede e *tamanho\_máximo* é o número máximo de nodos da rede. A constante de compromisso  $c$  foi adicionada à fórmula para permitir que a acurácia tenha um peso maior que a complexidade da rede. Desse modo, quanto maior o valor de  $c$ , maior será a importância da acurácia no cálculo do *fitness*. Se  $0 < c < 1$ , o tamanho será mais importante que a acurácia.

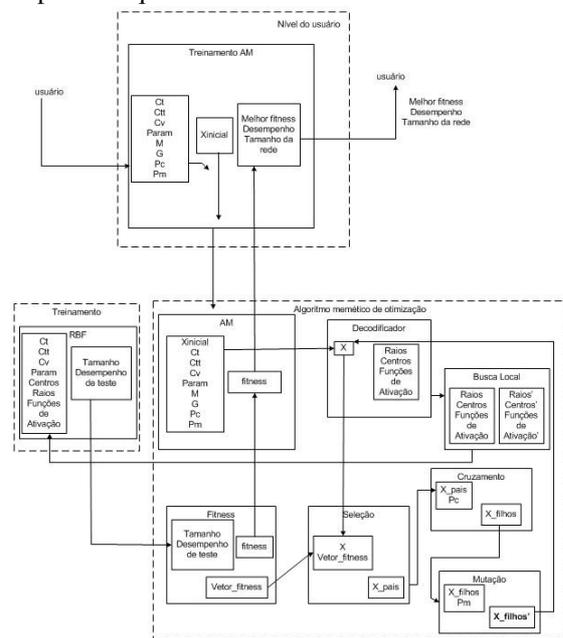


Figura 2. Representação esquemática do algoritmo memético para redes RBF.

Após o cálculo do *fitness*, é executado o processo de seleção de pais. Existem diversos tipos de seleção: roleta, elitismo, torneio, dentre outros. Neste trabalho, utilizamos o método da roleta com elitismo. No método da roleta, cada indivíduo tem sua aptidão denotada por um percentual da roleta. Um valor aleatório é gerado e o cromossomo correspondente

na roleta é selecionado para participar do cruzamento. O número de indivíduos selecionados é igual ao tamanho original da população e estes constituem os pais da próxima geração ( $X_{\text{pais}}$ ).

A função de cruzamento recebe o conjunto de indivíduos selecionados e a probabilidade de cruzamento ( $P_c$ ). O cruzamento entre o indivíduo  $X$  e o  $Y$  ocorre, com a probabilidade  $P_c$ , da seguinte forma: se o gene em questão for um dos bits relativos ao estado de um neurônio (ativo ou inativo), o filho  $F_1$  herdará o bit de  $X$  e o filho  $F_2$ , de  $Y$ . Para o par de bits relativo à representação da função de ativação de um neurônio, o filho  $F_1$  terá o primeiro bit herdado de  $X$  e o segundo, de  $Y$ ; para o filho  $F_2$ , ocorre o contrário. Se o gene em questão for relativo ao raio ou a um dos componentes do centro do neurônio, o filho  $F_1$  terá seu gene formado por uma porcentagem aleatória  $P_a$  do gene de  $X$  e  $(1 - P_a)$  do gene de  $Y$ ; para o filho  $F_2$ , ocorre o inverso.

A função de mutação recebe os indivíduos gerados pelo cruzamento ( $X_{\text{filhos}}$ ) e a probabilidade de mutação  $P_m$ . De acordo com a probabilidade  $P_m$ , o indivíduo sofrerá ou não mutação. Nesse procedimento, escolhemos aleatoriamente um dos neurônios, que sofrerá alterações no seu bit de ativação, na sua função de ativação, na largura do campo receptivo e nas componentes do centro.

Após os operadores de variação, o procedimento de busca local é executado sobre os indivíduos da prole, gerando centros dos campos receptivos localmente ótimos.

A seleção de sobreviventes segue um esquema *steady-state* [5], no qual o melhor indivíduo da população é mantido enquanto os outros são substituídos pela prole. A escolha desse esquema se deve à necessidade de se manter boas soluções na população e, ao mesmo tempo, sair de mínimos locais através dessa reposição [5].

Este processo, que envolve seleção, cruzamento, mutação e avaliação, se repete até o número máximo de gerações ( $G$ ) ser atingido. O algoritmo pode ser descrito pelos seguintes passos:

1. Inicialize a população:
  - a. Gere a população inicial aleatoriamente;
  - b. Decodifique as informações dos indivíduos em seus conjuntos de raios (largura do campo receptivo), centros e funções de ativação;
  - c. Execute a busca local;
  - d. Treine a camada de saída das redes RBFs;
  - e. Calcule o *fitness* dos indivíduos através da equação 7.
2. Selecione os pais através do método da roleta;
3. Realize o cruzamento, com probabilidade  $P_c$ , seguindo as restrições do esquema de codificação utilizado;

4. Execute a mutação nos indivíduos da prole de acordo com a probabilidade  $P_m$ , seguindo as restrições da codificação;
5. Avalie os indivíduos gerados:
  - a. Decodifique as informações;
  - b. Treine a camada de saída das redes RBFs;
  - c. Calcule o *fitness* dos indivíduos através da equação 7.
6. Execute a busca local na prole;
7. Selecione os sobreviventes com elitismo – apenas o indivíduo mais apto é mantido, os demais são substituído pela prole;

Repita os passos 2-7 até um critério de parada ser atingido.

#### 4 Experimentos

As bases de dados utilizadas nos experimentos foram obtidas no Repositório UCI, disponível em [17]. As bases de dados pertencem à Proben1 [16], uma coleção de problemas para treinamento de redes neurais. Utilizamos as bases Câncer, Heart e Diabetes.

O algoritmo memético foi executado com os seguintes parâmetros:  $M=100$ ;  $G=50$ ;  $P_c = 0.8$ ;  $P_m=0.1$ .

O método de validação utilizado para avaliar os classificadores foi o *Hold-Out*. O conjunto de exemplos foi dividido da seguinte maneira: 50% dos exemplos foram utilizados para treinamento, 25% para teste e 25% para formação do conjunto de validação. A constante de compromisso  $c$  foi fixada em 4.

A fim de realizarmos uma comparação com os métodos tradicionais de treinamento de redes RBF, com uma fase não supervisionada e outra supervisionada, executamos o treinamento de redes RBF com essas bases ajustando seus parâmetros (número de centros e funções de ativação) baseando-se na tentativa e erro. Em cada rede RBF simples, utilizamos a função gaussiana como função de ativação em todas as unidades de processamento escondidas. A Tabela 1 apresenta os resultados do treinamento tradicional de redes RBF.

**Tabela 1 - Resultados obtidos com ajuste de parâmetros de treinamento baseado na tentativa e erro**

Base de Dados	Classificações corretas no conjunto de Teste (%)	Tamanho da Rede (Número de Nós ocultos)
Câncer	91,954	12
Heart	83,4783	21
Diabetes	77,0833	17

Agora, mostramos os resultados obtidos através da otimização evolucionária de redes RBF. A Tabela 2 apresenta tais performances, bem como a melhor função de avaliação obtida.

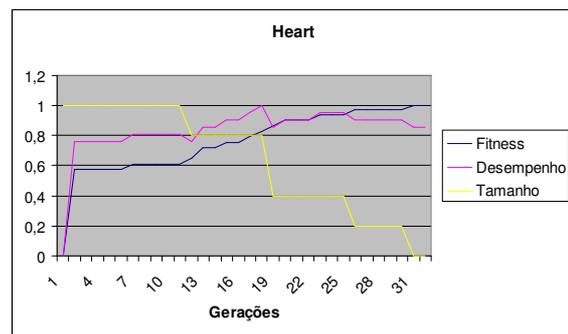
Podemos verificar que o método proposto obteve redes com capacidades de generalização melhores do que o método usual de tentativa e erro. A busca por um número bom de unidades de processamento, suas funções de ativações e larguras dos campos recepti-

vos correspondentes através da tentativa e erro é uma tarefa que exige tempo e nem sempre obtém os melhores resultados. A abordagem proposta conseguiu obter, de modo automático, parâmetros mais adequados a cada problema, minimizando a complexidade da rede neural.

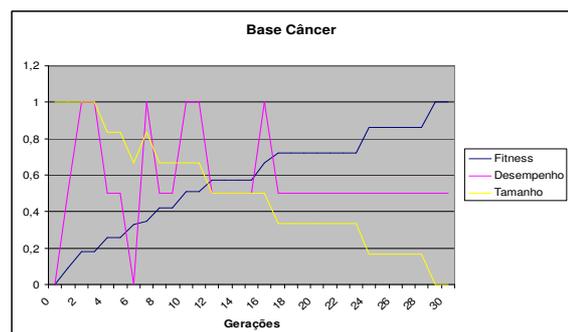
**Tabela 2 - Resultados obtidos com a otimização de redes RBF através do método proposto**

Base de Dados	Fitness Obtido	Classificações corretas no conjunto de Teste (%)	Tamanho da Rede (Número de Nós ocultos)
Câncer	0,907678	95,4023	12
Heart	0,813468	84,7826	15
Diabetes	0,80298	84,8958	17

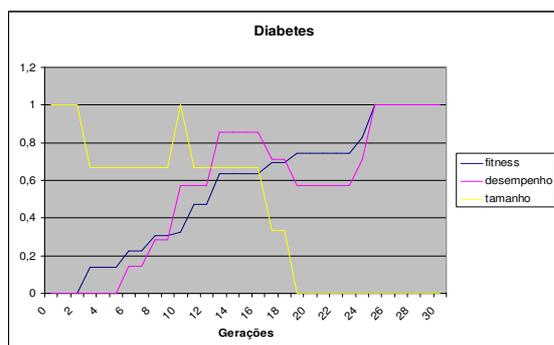
Para demonstrar a evolução das redes durante o processo evolucionário, apresentamos três gráficos que mostram os melhores valores da função de avaliação, do tamanho da rede e das porcentagens de classificações corretas no conjunto de teste a cada geração (iteração do AM). As Figuras 2, 3 e 4 representam tais gráficos para as bases Heart, Câncer e Diabetes, respectivamente. Para uma melhor visualização, os gráficos foram construídos com os resultados normalizados.



**Figura 2 - Gráfico com a evolução do desempenho e tamanho das redes geradas com o método proposto para a base Heart.**



**Figura 3 - Gráfico com a evolução do desempenho e tamanho das redes geradas com o método proposto para a base Câncer.**



**Figura 4 - Gráfico com a evolução do desempenho e tamanho das redes geradas para a base Diabetes.**

A estratégia proposta busca alcançar, com parcimônia ao longo das gerações, uma boa capacidade de generalização juntamente com a menor complexidade possível em uma rede RBF. A função objetivo tenta incorporar esse compromisso: não deixar que haja redes com uma complexidade alta e um desempenho alto (o que ocorre em outros trabalhos, sendo tratado como *outliers*) e o contrário, uma rede pequena com um desempenho baixo. É possível obter valores da função de avaliação que levem em consideração, também, a diferença entre o desempenho e o seu tamanho e não só uma média desses valores.

## 5 Conclusões

Apresentamos neste trabalho, uma abordagem evolucionária para a otimização da arquitetura de redes RBF. Levamos em consideração o número de unidades de processamento escondidas, suas funções de ativação e larguras do campo receptivo na busca por uma rede adequada para um dado problema. Usamos um Algoritmo Memético para evoluir as soluções, pois este lança mão de uma busca local para refinar as soluções obtidas e assim melhorando o resultado global. A função objetivo proposta tem a capacidade de retratar bem casos em que se encontram redes com bons desempenhos, mas complexas; bem como o inverso. Além disso, sua principal contribuição é penalizar redes que têm essas características divergentes, ou seja, ela incorpora um compromisso entre complexidade e generalização.

Realizamos experimentos com três bases de dados bastante exploradas na literatura usando o método de tentativa e erro para ajustar os parâmetros e com a estratégia. Comparamos os resultados e verificamos uma melhora significativa no desempenho da rede obtida e no esforço humano empregado para se gerar uma rede adequada.

Deste modo, este trabalho oferece uma ferramenta que consegue obter boas redes RBF ajustadas para um dado problema. Um ponto negativo deste método é o tempo computacional gasto para sua execução.

Como trabalhos futuros, pode-se investigar o uso de outros sistemas híbridos para lidar com a otimização de redes RBF.

## Referências Bibliográficas

- [1] Barreto, A. M. S., Barbosa, H. J. C.; Ebecken, N. F. F. Growing Compact RBF Networks Using a Genetic Algorithm. In: Proceedings of the SBRN'02, p. 61-66, 2002.
- [2] Braga, A. P., Carvalho, A.C P. L. F.; Ludermir, T. B. Redes Neurais Artificiais: Teoria e Aplicações. LTC. Rio de Janeiro, 2000.
- [3] Buriol, L.S., França, P.M.; Moscato, P.. A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem. Journal of Heuristics, Hingham, v. 10, n. 5, p. 483-506, 2004.
- [4] Duda, R. O., Hart, P. E.; Stork, D. G. Pattern Classification. 2ª Edição.
- [5] Ferentinos, K. P. Biological engineering applications of feedforward neural networks designed and parameterized by genetic algorithms. Neural Networks, v. 18, págs. 934-950, 2005.
- [6] França, P.M., Gupta, J.D., Mendes, A.S., Moscato, P., Veltink, K.J., Metaheuristic Approaches for the Pure Flowshop Manufacturing Cell Problem, 7th International Workshop On Project Management and Scheduling, Osnabrück, Germany, 2000.
- [7] França, P.M., Mendes, A. S., Moscato, P., A Memetic Algorithm for the total tardiness Single Machine Sheduling Problem, European Journal of Operational Research, 132-1, pp. 224-242, 2000.
- [8] Goldberg, D. E.. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison, 1989.
- [9] Harpham, C. W. Dawson e M. R. Brown. A review of genetic algorithms applied to training radial basis function networks. Neural Computation and Application, v. 13, p. 193-201, 2004
- [10] Haykin, S. Neural Networks: a comprehensive foundation. 2ª Edição. Prentice-Hall. New Jersey, 1999.
- [11] Kuncheva, Ludmila I.. Initializing of an RBF network by a genetic algorithm. Neurocomputing, p. 273-288, 1996.
- [12] Lacerda, E. G. M. de; Carvalho, A. de e Ludermir, T. B.. Evolutionary Optimization of RBF Networks. International Journal Of Neural Systems, v. 11, n. 3, p. 287-294, 2001.
- [13] Moscato, P., On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms, Technical Report, Caltech Concurrent Computation Program, C3P Report 826, 1989.
- [14] Murolo A. C. et al.. ESTATÍSTICA: Para os Cursos de Economia, Administração e Ciências Contábeis, Editora Atlas, V. 1., 3ª ed, 1999.
- [15] Park J, Sandberg IW . Universal approximation using radial basis function networks. Neur Comput, v. 3, n. 2, p. 246-257, 1991.
- [16] Prechelt, Lutz. Proben 1 - A set of Neural Network Benchmark Problems and Benchmarking Rules. Relatório Técnico. Alemanha. Universidade de Karlsruhe. 1994.
- [17] UCI Machine Learning Repository. Disponível em: <http://www.ics.uci.edu/~mlern/MLRepository.html>. Último acesso em 28/08/2006.
- [18] Ludermir, T. B.; Yamazaki, A. e Zanchettin, C.. An optimization methodology for neural network weights and architectures. IEEE Transactions on Neural Networks, v. 17, n. 5, p. 1127-1137, 2006.
- [19] Zanchettin, C. e Ludermir, T. B.. A methodology to train and improve artificial neural networks weights and connections. In Int. Joint Conf. on Neural Networks, p. 5267 - 5274. IEEE Computer Society, 2006.