

## IMPLEMENTAÇÃO DE UM MECANISMO AUTOMÁTICO DE CONTROLE DE VIZINHANÇAS ENTRE AGENTES DE UMA POPULAÇÃO INSPIRADA EM ALCATÉIA

Luís F. A. Pessoa<sup>1</sup>, Fernando B. D. Lima Neto<sup>1</sup>, Amanda Leonel<sup>1</sup> e Luís C. S. Menezes<sup>1</sup>

<sup>1</sup>Departamento de Sistemas Computacionais, Universidade de Pernambuco, Rua Benfica, 455, 50750-470 Madalena, Recife

{lfap, fbln, aln, luis}@dsc.upe.br

**Resumo** – Através de técnicas de Inteligência Artificial, como Sistemas Multi-Agentes e Computação Social – inspirada no comportamento de populações naturais (*i.e.* uma alcatéia), busca-se propor uma nova ferramenta para a modelagem de contornos de uma dada nuvem de pontos de entrada (que podem representar uma função), de forma a diminuir a representatividade desse conjunto. Foram definidos e construídos o repertório de comportamentos dos lobos, assim como as características da dinâmica do ambiente, cuja saída será a lista de posicionamento e raio final para cada lobo, de modo que representem a nuvem de pontos. Como prova do conceito aqui apresentado, foram realizadas simulações de aproximação com quatro funções diferentes: duas polinomiais e duas trigonométricas. Essas simulações objetivaram determinar o grau de influência no resultado final dos parâmetros escolhidos, além de apresentar os bons resultados obtidos com o algoritmo desenvolvido. Baseado na boa qualidade dos resultados obtidos, este trabalho representa um passo importante para a viabilização de uma inovadora ferramenta inteligente de aproximação de funções baseada em inteligência coletiva.

**Palavras-chave** – Sistemas Multi-Agentes, Computação Natural, Computação Social, Aproximação de Funções.

### 1. Introdução

A aproximação de funções representa uma classe de problemas bastante importante, uma vez que funções podem representar problemas do mundo real ou descrever alguma realidade dada a partir de uma nuvem de pontos. Porém, problemas do mundo real geralmente possuem leis de formação – funções – complexas e, por isso, caras de serem calculadas computacionalmente através de técnicas como Redes Neurais Artificiais (RNA) ou Programação Linear.

As RNAs possuem um alto custo computacional (de memória e de processamento) devido as suas diversas fases de treinamento, tempo de execução excessivo, além da possibilidade de acontecer *overfitting* ou *underfitting* e a função não ser aproximada corretamente. Contudo, existem abordagens híbridas que utilizam Algoritmos Genéticos para determinar a arquitetura de uma RNA de alta ordem [1], minimizando as chances da rede não conseguir aproximar a função. Porém, com a inserção de um novo passo no processo de aproximação, o custo computacional aumenta.

Por sua vez, a Programação Linear com o uso do algoritmo SIMPLEX tem uma complexidade exponencial –  $O(2^n)$  – para o pior caso. Isso é ruim, pois há um grande consumo de recursos computacionais existentes, além de possuírem um alto tempo de execução que escala exponencialmente com o tamanho da entrada do problema de aproximação.

Dessa forma, é buscada uma abordagem alternativa e inteligente para a resolução do problema da aproximação de funções. Essa abordagem busca (i) modelar e diminuir a representatividade da nuvem de pontos de entrada e (ii) realizar operações matemáticas para determinar a função aproximada.

Então, este trabalho busca utilizar metáforas com o comportamento de caça de uma alcatéia na natureza para modelar a nuvem de pontos de entrada e diminuir a sua representatividade. Com isso, acredita-se que o modelo computacional se torne mais intuitivo e o sistema obtenha um menor custo computacional, aproveitando as vantagens da interação entre os agentes que compõem o ambiente, a saber: os lobos e as ovelhas.

O ambiente proposto é composto por árvores - que representam pontos da nuvem de dados de entrada; por ovelhas – que representam as presas dos lobos (ou pontos a serem “caçados”); e, por lobos –

que irão caçar as ovelhas. O que se imaginou foi que após a caça, os lobos estarão estrategicamente posicionados, de forma que as ovelhas estejam formando o contorno da nuvem de pontos de entrada. Assim, ela poderá ser representada pelos posicionamentos finais dos lobos e seus respectivos raios de atuação durante a caça.

A organização deste trabalho é apresentada a seguir. Primeiramente, foram abordados os conteúdos teóricos que dão sustentação para o desenvolvimento do trabalho: (i) Comportamento de uma alcatéia na natureza; e, (ii) Computação Social. Em seguida, toda a modelagem e implementação da contribuição é descrita e são mostrados e analisados os resultados obtidos nas simulações realizadas. Por fim, são apresentadas as conclusões finais e perspectivas futuras.

## 2. Comportamento de uma Alcatéia na Natureza

As alcatéias geralmente são formadas por grupos entre 8 a 15 lobos, dependendo do tamanho de seu território, da abundância de comida e por fatores sociais. Elas possuem uma hierarquia bem definida, assumidas através da postura do lobo, características comportamentais e lutas pela dominância do bando.

O casal de lobos dominantes é denominado de *alfa*, os quais são os primeiros a se alimentarem e os únicos que se reproduzem dentro do bando. Seguindo essa hierarquia, existem os lobos *beta* que exercem uma liderança logo abaixo dos *alfa*. Enquanto isso, na base da pirâmide, encontra-se o lobo *omega*, o qual não possui quase nenhum direito dentro do grupo.

Ao contrário do que muitos pensam, os lobos não se comunicam apenas através dos uivos, mas, também, através da sua postura corporal e dos odores [2]. Seus variados meios de comunicação permitem que eles possam demarcar territórios, manter a alcatéia junta em ambientes densos como florestas, chamar o bando para um local específico ou demonstrar medo, submissão, dominância, etc.

A caça é feita em grupo e todos os membros de uma alcatéia participam. Sua estratégia é perseguir as presas até elas se cansarem, buscando cercá-las. Em geral, visam ou o animal mais velho ou o mais novo ou aquele que esteja ferido ou doente. Dessa forma, o grupo gastará menos energia na perseguição.

Uma vez que a caça está abatida, procuram comê-la rapidamente para evitar o roubo ou disputa do alimento por outro animal. Cercar o alvo e operar em coordenação de ações para obtenção de objetivos comuns são aspectos centrais a serem incorporados no modelo artificial.

## 3. Computação Social

Computação Social é uma subárea de Sistemas Multi-Agentes cujo foco é a cooperação e colaboração entre os agentes que compõem o ambiente a partir da interação, comunicação e do comportamento coletivo. Dessa forma, as principais vantagens desse paradigma de Computação Inteligente são: (i) o fortalecimento do grupo e (ii) o conhecimento que emerge da interação entre seus membros.

Utilizando o comportamento de *populações sociais* como uma metáfora, a Computação Social é, também, uma subárea da Computação Natural a qual se propõe a:

- simular a dinâmica de sistemas reais e analisar aspectos específicos da interação entre os agentes que compõem o ambiente. Logo, é possível prever ou projetar o impacto sobre o grupo que determinadas modificações no ambiente podem ocasionar; e,
- resolver problemas complexos com um menor custo computacional e de forma mais intuitiva [3].

A expressão '*populações sociais*' é utilizada aqui para se referir a um conjunto de agentes que podem interagir. Qualquer conjunto de agentes com comportamentos coletivos (*i.e.* uma multidão, uma manada, uma alcatéia, etc.) são considerados populações sociais. Entretanto, não é obrigatória a movimentação desses agentes pelo ambiente, basta haver uma comunicação e interação entre eles.

Apesar da modelagem e implementação desse paradigma de computação serem feitas utilizando-se a técnica de Sistemas Multi-Agentes, várias aplicações incorporam algum mecanismo de Computação Evolucionária [4] para aperfeiçoar a adaptabilidade e mutação dos agentes dentro da população. Assim, é possível reproduzir melhor a dinâmica natural de sistemas reais no ambiente artificial.

Na modelagem desenvolvida, a cooperação entre os agentes (*i.e.* objetivos e intencionalidades não conflitantes) foi cuidadosamente planejada para que o custo computacional fosse baixo, além de possibilitar um modelo mais intuitivo, através da metáfora com uma alcatéia na natureza.

#### 4. Modelagem e Implementação do Sistema

O ambiente foi modelado e inspirado no comportamento de caça de uma alcatéia na Natureza. Imaginou-se que durante a caça, os lobos perseguem as presas (ajustando seu posicionamento) assim como ajustam seus raios de atuação, o qual determina as ovelhas que estão sendo caçadas pelos lobos atualmente.

Por sua vez, as ovelhas reagirão ao ataque dos lobos fugindo em direção à floresta toda vez que se encontrarem dentro do raio de atuação do lobo. Enquanto as árvores representam a nuvem de pontos de entrada a qual se deseja modelar.

O objetivo é que ao final da execução, os lobos estejam posicionados em locais estratégicos e com seus raios de atuação determinados. De forma que as ovelhas formem o contorno da floresta. A Figura 1 mostra o ambiente modelado e os elementos que o compõe, de acordo com a descrição anterior.

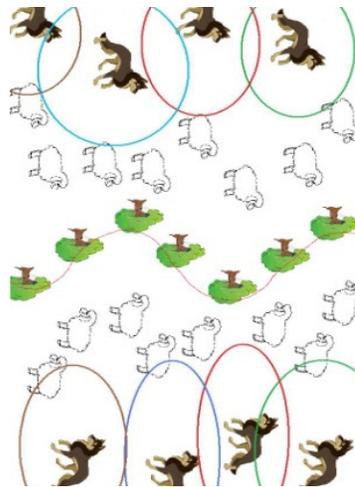


Figura 1 – Modelo conceitual do ambiente.

Ressalta-se, entretanto, que nesta modelagem nenhum protocolo explícito de comunicação entre os lobos foi desenvolvido. Porém, como será discutido posteriormente, os lobos se comunicam indiretamente através de suas ações e das ovelhas.

A partir da Figura 1 é possível observar que existem dois bandos no ambiente: um acima e outro abaixo da floresta. Essa modelagem foi realizada de forma que concavidades voltadas para cima e para baixo fossem melhores modeladas.

##### 4.1 Descrição dos Elementos do Ambiente

Os lobos são agentes orientados a objetivos, os quais são: (i) diminuir seu erro médio individual (*i.e.* média do somatório da distância das ovelhas dentro de seu raio de atuação – inclusive – para a floresta); e, (ii) diminuir seu desvio-padrão individual, baseado no seu erro médio.

Além disso, os lobos podem modificar o ambiente através dos operadores de (i) ajuste de seus posicionamentos em direção às ovelhas e (ii) ajuste de seus raios de atuação. O processo de escolha do operador a ser utilizado será detalhado adiante.

As ovelhas são agentes reativos simples. Elas reagirão a uma ação do lobo toda vez que se encontrarem dentro dos raios de atuação deles. Sua ação consiste em movimentar-se o mínimo possível em direção à floresta, de forma que estejam seguras. Ou seja, para a extremidade da área de atuação do lobo.

As árvores não são agentes dentro do ambiente, apenas modelam a nuvem de pontos de entrada.

## 4.2 Modelo Computacional

A Figura 2 mostra como o modelo conceitual foi implementado computacionalmente. Nela, é possível observar os lobos com seus respectivos raios de atuação; as ovelhas nas extremidades dos raios de atuação do lobo; e, a floresta (ao centro da imagem) que representa a nuvem de pontos de entrada.

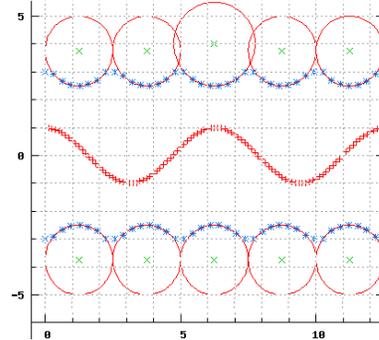


Figura 2 – Modelo computacional do ambiente.

Nesse modelo, o número de lobos e ovelhas acima da floresta é o mesmo do número de lobos e ovelhas abaixo da floresta, os quais são definidos por dois parâmetros distintos dentro do sistema. Além disso, para cada lobo abaixo da floresta, há um lobo posicionado na mesma abscissa acima da floresta. O mesmo ocorre para as ovelhas.

O ambiente implementado possui, ainda, as seguintes restrições que refletem o modelo conceitual: (i) as ovelhas não podem ultrapassar a floresta – pois ficará susceptível ao ataque do bando da região oposta; e, (ii) os lobos não poderão ultrapassar ovelhas, tendo em vista que essa é uma situação improvável durante a caça.

## 4.3 Dinâmica do Sistema

A execução do sistema é semelhante ao de uma lista circular. O primeiro lobo da lista é selecionado e delibera sobre suas ações. Em seguida, o segundo lobo é selecionado e delibera sobre suas ações, e assim por diante. Quando o último lobo da lista delibera sobre suas ações, o primeiro lobo volta a ser selecionado. Cada ciclo desse determina uma época de execução.

O lobo deliberará sobre a ação a ser executada, baseado no seu erro médio ( $E$ ) e no seu desvio-padrão ( $\sigma_L$ ). O erro médio do lobo é a média do somatório da distância das ovelhas ( $O$ ) compreendidas dentro de seu raio de atuação para a floresta ( $d_{O,F}$ )

$$E = \frac{1}{n} \sum_{i=1}^n d_{O_i,F} \quad (1)$$

onde,  $n$  é o número de ovelhas que estão dentro do raio de atuação do lobo,  $F$  é o conjunto da nuvem de pontos de entrada que representam a floresta e

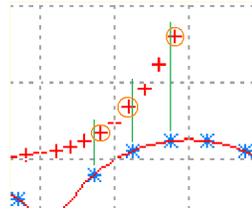
$$d_{O,F} \in \{d_{O,A} \mid A \in F \text{ and } \forall A': F, |A_x - O_x| \leq |A'_x - O_x|\} \quad (2)$$

é a distância euclidiana ( $d_{O,A}$ ) obtida entre a ovelha e a árvore ( $A$ ) mais próxima de sua abscissa.

Uma vez que o lobo tenha computado seu erro médio individual, é calculado, em seguida, o seu desvio-padrão, de acordo com a seguinte equação:

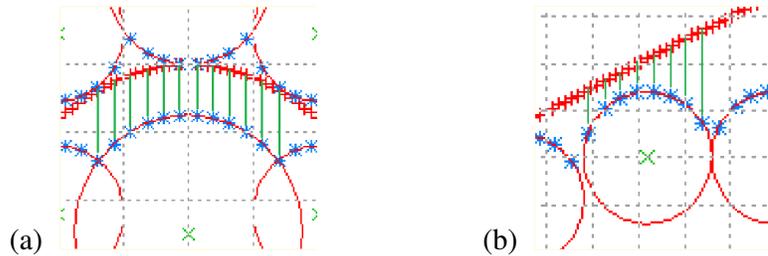
$$\sigma_L = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_{O_i,F} - E)^2} \quad (3)$$

A Figura 3 ilustra como é realizado o cálculo da distância de três ovelhas para a nuvem de pontos. A distância de cada uma delas para a floresta é igual a distância delas para as respectivas árvores destacadas e mais próximas de suas abscissas.



**Figura 3** – Distância das ovelhas para a floresta.

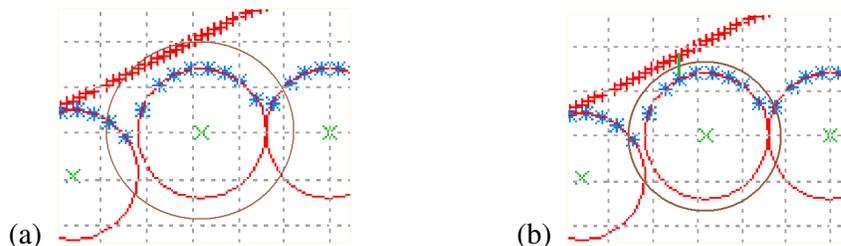
Uma vez calculados os valores do erro médio e do desvio-padrão individual, os lobos irão escolher o operador a ser utilizado: (i) *ajuste de posicionamento* – caso o seu erro médio e seu desvio-padrão sejam altos; ou, (ii) *ajuste de raio* – caso o seu erro médio seja alto e seu desvio-padrão seja baixo. Essas situações estão ilustradas na Figura 4.



**Figura 4** - Caso de ajuste de raio (a) e ajuste de posicionamento (b).

Após o lobo executar sua ação, o sistema verificará se alguma das condições de parada foi atingida e, assim, pare a execução; caso contrário, o lobo seguinte da lista irá deliberar sobre suas ações. As condições de parada do sistema são: o número máximo de épocas ou o erro médio global mínimo (*i.e.* a média do somatório da distância de todas as ovelhas para a floresta).

Por fim, ressalta-se que nem sempre um lobo poderá ajustar seu raio ou posicionamento do valor pré-determinado no sistema, tendo em vista que ovelhas poderiam ultrapassar a floresta (situação proibida no ambiente). Dessa forma, o novo valor de ajuste do raio ou do posicionamento será igual a 90% da menor distância das ovelhas compreendidas dentro de seu raio de atuação para a floresta. A Figura 5 mostra uma situação em que ocorre o explicado.



**Figura 5** - Ajuste de raio proibido (a) e novo valor de ajuste (b).

Ao final da execução do algoritmo, sua saída será o conjunto de posicionamentos dos lobos e seus respectivos raios de atuação, determinados de forma que modelem a nuvem de pontos de entrada.

## 5. Experimentos

Os experimentos foram planejados através da técnica de Fatorial  $2^k$  [5] com três fatores ( $k = 3$ ), obtendo-se 8 simulações para cada função. Essa foi a técnica escolhida, pois é simples de ser realizada e permite a compreensão e análise dos efeitos de cada fator sobre as variáveis de resposta. Outro objetivo dos experimentos foi verificar se o sistema funciona de acordo com o planejado, ou seja, se as ovelhas conseguem formar o contorno correto da floresta.

Os fatores utilizados para a realização dos experimentos foram: o número de lobos iniciais, o erro médio mínimo (EMM) e o desvio-padrão mínimo (DPM) do lobo. O EMM e o DPM são os limiares que

definem se o erro individual ou o desvio-padrão do lobo, respectivamente, está alto ou baixo. Alterando os valores desses dois fatores, será alterada a tendência do comportamento do lobo.

Os níveis utilizados para cada fator foram: número inicial de lobos (5% e 20% do número de árvores); EMM (0.5 e 0.15 unidades cartesianas); e, DPM (0.5 e 0.15 unidades cartesianas).

Foram escolhidas quatro configurações distintas da nuvem de pontos de entrada. Duas delas representam funções do tipo polinomial (uma linear e a outra quadrática), enquanto as outras duas são do tipo trigonométrica (a cosseno e a tangente hiperbólica). Essas funções foram escolhidas de forma que o algoritmo fosse testado para comportamentos distintos da floresta.

Nota-se que para todas as funções analisadas, o pior caso obtido (dentre as oito configurações dos fatores experimentadas) foi para: número de lobos = 5% das árvores; EMM = 0,5; e, DPM = 0,15. Enquanto, o melhor caso obtido foi para a configuração: número de lobos = 20% das árvores; EMM = 0,15; e, DPM = 0,15.

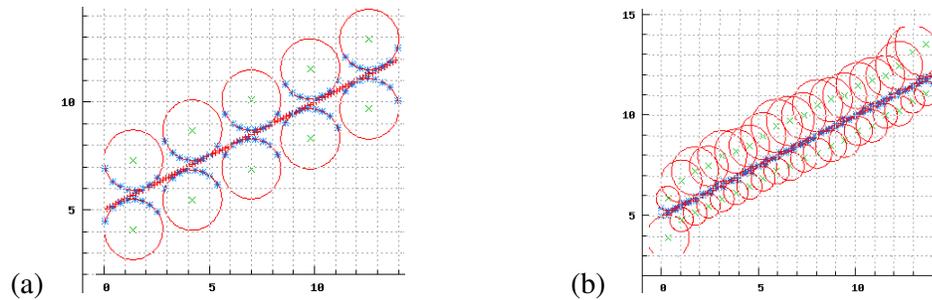
### 5.1 Função Linear

Para a função linear, cuja lei de formação é

$$y(x) = 0,5x + 5, \quad x \in [0, 14] \quad (4)$$

,o fator de maior relevância dentre os analisados foi o número de lobos inicial, seguido pelo EMM. Para o pior caso, o erro médio global obtido foi 0,47177, com desvio-padrão de 0,45831.

O melhor caso obtido dentre as experimentações realizadas obteve um erro médio global igual a 0,09772, com um desvio-padrão de 0,05912. Os estados finais para o melhor e o pior caso são mostrados na Figura 6.



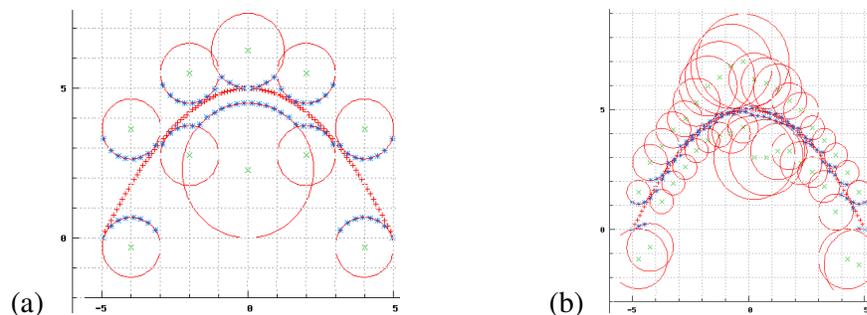
**Figura 6** - Pior (a) e melhor (b) resultados para a função linear.

### 5.2 Função Quadrática

A floresta para a função quadrática possui a seguinte lei de formação:

$$y(x) = -0,2x^2 + 5, \quad x \in [-5, 5] \quad (5)$$

Para essa nuvem de pontos, o fator mais relevante foi o número de lobos inicial, seguido pelo DPM. O pior caso obtido obteve um erro médio global de 0,47849 e um desvio-padrão de 0,40943. Enquanto, para o melhor caso, o erro médio global foi 0,11890, com um desvio-padrão igual a 0,09962. A Figura 7 mostra os estados finais para o melhor e o pior caso referente à função quadrática.



**Figura 7** - Pior (a) e melhor (b) caso para a função quadrática.

### 5.3 Função Cosseno

A função cosseno utilizada foi definida dentro do intervalo  $[0, 12,5]$ . Assim, foi possível obter concavidades voltadas para cima e outra para baixo.

Assim como para a função quadrática, o fator que resultou numa maior variação do resultado final foi o número inicial de lobos, seguido pelo DPM.

O pior resultado para essa floresta obteve um erro médio global de 0,47950 e um desvio-padrão de 0,4291. Enquanto, para a configuração que obteve um melhor resultado, o erro médio global foi 0,11100, com um desvio-padrão de 0,07625.

A Figura 8 mostra os estados finais para o pior e o melhor resultado obtido para essa floresta.

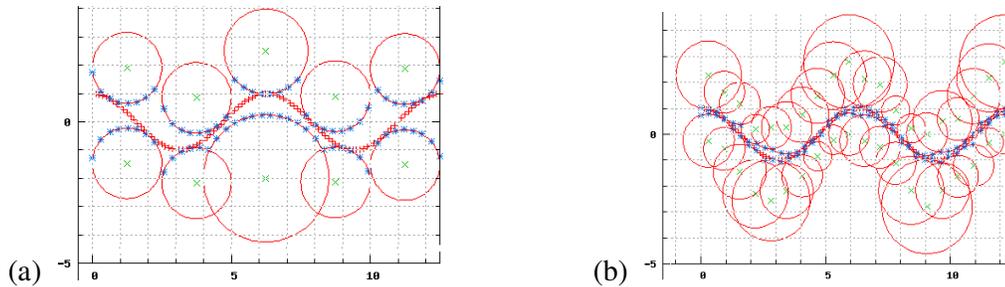


Figura 8 - Pior (a) e melhor (b) resultado obtido para o cosseno.

### 5.4 Função Tangente Hiperbólica

A seguir, apresenta-se a lei de formação utilizada para representar a floresta:

$$y(x) = 2 \tanh(x), \quad x \in [-10, 10] \quad (6)$$

Assim como para a função linear, o fator de maior relevância foi o número de lobos inicial, seguido do EMM.

O pior resultado para essa floresta obteve um erro médio global de 0,52966 e um desvio-padrão de 0,58227. Já para o melhor caso, o erro médio global foi 0,13242 com um desvio-padrão de 0,08611. A Figura 9 mostra os estados finais para o pior e o melhor resultado obtido.

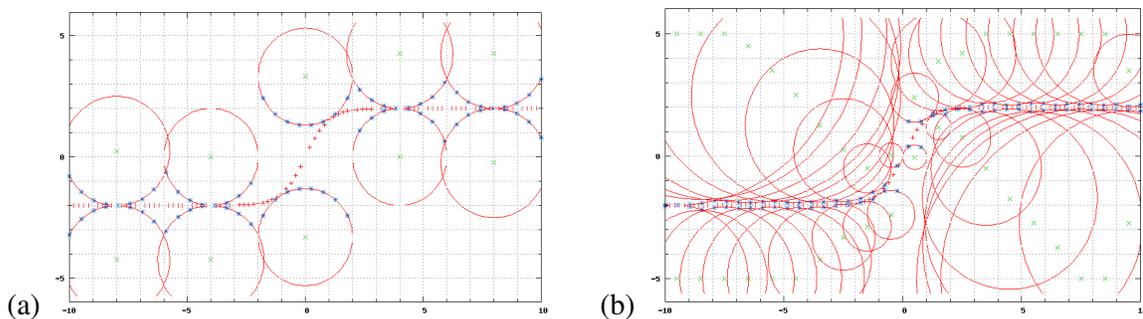


Figura 9 - Pior (a) e melhor (b) resultado obtido para a tangente hiperbólica.

## 6. Discussão

A partir dos resultados obtidos, percebe-se que o número de lobos no ambiente é fundamental para realizar um melhor contorno da floresta, tendo em vista que para as configurações com um menor número de lobo, as ovelhas não aderiram bem ao comportamento da floresta. Nota-se, ainda, que o EMM e o DPM possuem uma mesma relevância dentro do sistema.

Além disso, apesar de não haver um protocolo de comunicação explícito entre os lobos, ocorre uma comunicação indireta entre eles. Pois suas ações são baseadas nas ovelhas dentro de seu raio de atuação e, como há interseções entre os raios do lobo, uma ação do lobo poderá interferir na ação do seu vizinho.

Observa-se, ainda, que os resultados obtidos com a execução do algoritmo para uma mesma configuração de parâmetros e da floresta foram sempre os mesmos. Onde o pior resultado foi obtido para a tangente hiperbólica, enquanto o melhor foi o obtido para a função linear.

Dessa forma, este sistema proposto representa um passo importante para a viabilização de uma ferramenta inteligente para a modelagem do contorno de uma dada nuvem de pontos. Em longo prazo, ela poderá viabilizar uma alternativa computacional para resolver o problema da aproximação de funções e, espera-se, com um baixo consumo de recursos computacionais.

## 7. Trabalhos Futuros

Para tornar o sistema menos dependente do número de lobos inicial do sistema, deverá ser criado um novo operador para o lobo: o de recrutamento e descarte. Dessa forma, o número de lobos será adaptado durante a execução do sistema para que modele e represente melhor a nuvem de pontos de entrada.

Todavia, para tornar o sistema mais automático e robusto, deverá ser construído um mecanismo inteligente para a definição dos valores dos parâmetros do ambiente. Além disso, a elaboração de um protocolo e um mecanismo de comunicação entre os lobos possibilitará uma melhor coordenação de suas ações.

## 8. Referências Bibliográficas

- [1] G. A. Rovithakis, I. Chalkiadakis, M. E. Zervakis, High-order neural network structure selection for function approximation applications using genetic algorithms, **Systems, Man, and Cybernetics, Part B: Cybernetics**, 34(2004), 150 – 158.
- [2] Wolf Country, The Wolf Pack, Disponível em: <<http://www.wolfcountry.net/information/WolfPack.html>> Acesso em: 21 de setembro de 2009.
- [3] E. Bonabeau, M. Dorigo, T. Theraulaz, Swarm Intelligence: From Natural to Artificial Systems, **Oxford University Press**, (1999).
- [4] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, **The MIT Press**, (1992)
- [5] P. J. Freitas, Introdução à Modelagem e Simulação de Sistemas com Aplicações Arena, 2ª Edição, **Visual Books**, (2008).