

# MODELO NEURO-EVOLUTIVO COM INSPIRAÇÃO QUÂNTICA APLICADO AO PROBLEMA PRESA – PREDADOR

Eduardo D. M. Dias, Marley M. B. R. Vellasco, André V. Abs da Cruz

Departamento de Engenharia Elétrica – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)  
Marquês de São Vicente 255, Gávea, Rio de Janeiro – RJ

[eduardo.dias@ifrj.edu.br](mailto:eduardo.dias@ifrj.edu.br), [marley@ele.puc-rio.br](mailto:marley@ele.puc-rio.br), [andrev@ele.puc-rio.br](mailto:andrev@ele.puc-rio.br)

**Resumo** – Este artigo propõe o uso de um modelo neuro-evolutivo para a otimização do movimento do caçador para o *benchmark* presa/predador em um *grid* contínuo. Para a otimização dos pesos sinápticos da rede foi utilizado um algoritmo evolutivo com inspiração quântica e representação real (AEIQ-R). Os resultados obtidos foram comparados com modelos evolutivos tradicionais. Os resultados corroboram a robustez da metodologia com inspiração quântica para o *benchmark* utilizado.

**Palavras-chave** – Computação Evolutiva, Algoritmos Genéticos, Algoritmos com Inspiração Quântica

## 1 Introdução

Sistemas Multi-Agentes é uma área dentro da Inteligência Computacional que fornece ferramentas para a coordenação de sistemas que envolvem múltiplos agentes inteligentes. Um Sistema Multi-Agente pode ser definido como um grupo de agentes autônomos, interagindo entre si e compartilhando um mesmo ambiente, que é percebido através de sensores, e onde eles agem realizando ações [1]. Um exemplo de sistema multi-agente bastante estudado na literatura é o clássico jogo de perseguição presa/predador, o qual consiste em um *grid* normalmente discreto bidimensional, onde cinco agentes, sendo quatro predadores e uma presa, se movem pelo espaço, não podendo, obviamente, ocupar a mesma posição simultaneamente. O objetivo dos caçadores é capturar a presa e esta é considerada capturada quando é cercada pelos predadores, não tendo assim a possibilidade de se mover. Como exemplos de trabalhos que estudaram esta forma clássica do jogo pode-se citar [2] e [3].

Diversas variações desse jogo, como a alteração do número de agentes, podem ser encontradas na literatura. Brooks e Griffin, em [4], simulam uma perseguição entre automóveis onde se admite que o predador não possui informação perfeita sobre a posição da presa; [5] utiliza sistemas de controle não linear para simular a fuga/perseguição de aviões não tripulados; [6] faz uso de um algoritmo evolutivo para simular ecologicamente o comportamento de espécies distintas. Estudos em aplicações militares, como batalha naval [7] e lançamento de mísseis [8] também podem ser encontrados.

Em muitos problemas multi-agentes, é extremamente custoso para o programador decifrar e implantar as inter-relações entre os agentes e os objetivos desejados. Por conta disso, os chamados sistemas inteligentes estão sendo cada vez mais usados para tratar esse tipo de problema, pois, neste caso, os agentes possuem capacidade autônoma de aprendizado. Como exemplos de sistemas inteligentes pode-se citar a Lógica Fuzzy (LF) [9], Redes Neurais Artificiais (RNA) [10], algoritmos genéticos e programação genética [11],[12]. Porém, tais ferramentas de aprendizado possuem fortes limitações [13]: (i) usualmente assumem que os conhecimentos a serem adquiridos/manipulados e as informações disponíveis são corretos e devem estar completos em relação ao problema (o que dificilmente ocorre); (ii) são orientadas para manipular informações simbólicas (informações qualitativas), onde os valores numéricos e contínuos (informações quantitativas) representam um problema difícil de ser tratado.

Uma alternativa para melhorar o desempenho dos sistemas inteligentes é integrar dois desses sistemas numa modelagem híbrida com o objetivo de associar as vantagens de cada abordagem e minimizar as possíveis limitações e deficiências. Um exemplo clássico é o sistema neuro-fuzzy ANFIS [14], que é a integração entre o sistema *Fuzzy* e as Redes Neurais Artificiais. A principal vantagem deste sistema é associar a interpretabilidade dos sistemas *fuzzy* à capacidade de aprendizado das redes neurais, bem como a sua tolerância a falhas.

Há ainda a possibilidade de melhorias nos algoritmos de aprendizado, como por exemplo, os algoritmos genéticos com inspiração quântica. Estes algoritmos se baseiam em idéias inspiradas na física quântica, em particular no conceito de superposição de estados, apresentando melhor desempenho em diversos tipos de aplicações [15].

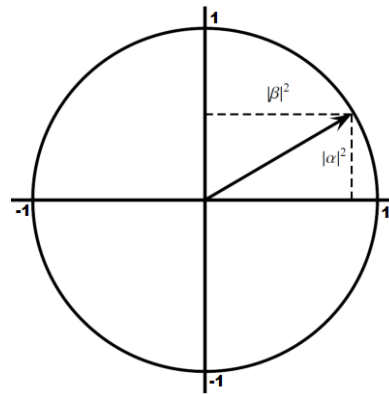
Um exemplo de problema onde os algoritmos evolutivos tradicionais não apresentam bom desempenho é na otimização de pesos para redes neurais. Os algoritmos evolutivos apresentam um bom desempenho para mapear as regiões mais promissoras do espaço de busca mas, em geral, não são tão eficientes quanto os algoritmos baseados em gradiente para encontrar o mínimo local dentro de uma região [16], [17]. Além disso, na área de redes neurais, normalmente necessita-se fazer diversas seqüências de aprendizado, devido ao fato de que a topologia ideal da rede neural não é, geralmente, conhecida à priori [10].

Este trabalho tem por objetivo utilizar os Algoritmos Genéticos com Inspiração Quântica visando otimizar os pesos de uma Rede Neural para fazer uma modelagem multi-agentes do problema *benchmark* de presa e predador em uma grade contínua, onde se procura a melhor estratégia a ser empregada pelos predadores com o objetivo de capturar a presa.

Além desta primeira parte introdutória, o presente trabalho é composto por mais cinco seções. A Seção 2 descreve os algoritmos evolutivos quânticos, enfatizando o algoritmo com representação real que é utilizado no trabalho. A Seção 3 introduz a área de neuro-evolução em geral e a Seção 4 apresenta a modelagem neuro-evolutiva particularmente empregada neste trabalho. Os resultados obtidos e as comparações com outros modelos são apresentados na Seção 5. Finalmente, na Seção 6 são feitas as considerações finais sobre o trabalho.

## 2 Algoritmos Evolutivos com Inspiração Quântica e Representação Real

Os algoritmos evolutivos com inspiração quântica usando representação binária foram propostos inicialmente em [18]. Neste modelo, o algoritmo usa uma representação especial que simula um cromossomo composto por  $q$ -bits, onde cada  $q$ -bit é formado por um par de números  $(\alpha, \beta)$ , onde  $|\alpha|^2 + |\beta|^2 = 1$ . O valor dado por  $|\alpha|^2$  indica a probabilidade do respectivo  $q$ -bit assumir o valor 0 quando for observado, enquanto que a probabilidade do mesmo assumir valor 1 é dada por  $|\beta|^2$ . Pode-se visualizar graficamente esta relação entre  $\alpha$  e  $\beta$  na Figura 1 a seguir.



**Figura 1** - Representação gráfica das probabilidades de se observar os valores 0 e 1 para um  $q$ -bit qualquer

Este algoritmo com representação binária foi usado com sucesso em problemas de otimização combinatorial [18] e detecção de faces [19], apresentando resultados superiores aos algoritmos genéticos convencionais em termos de convergência e qualidade de soluções encontradas. Entretanto, como acontece com outros tipos de algoritmos evolutivos (sendo os algoritmos genéticos o exemplo mais comum), a representação binária não é, necessariamente, a mais adequada para um problema de otimização numérica, por apresentar algumas particularidades que restringem a capacidade de otimização do algoritmo. Em particular, pode-se citar o fato de que a representação binária pode introduzir problemas de métrica relativas à distância entre soluções. Por exemplo, os números 31 e 32 na base decimal são representados em binário como “01111” e “10000”, ou seja, dois números próximos um do outro são representados por dois números binários completamente diferentes entre si.

Por conta disso, optou-se neste trabalho por um algoritmo evolutivo com inspiração quântica e representação real (AEIQ-R) [15][20][21]. Neste modelo, os indivíduos da população deixam de ser representados por um certo número de  $q$ -bits e passam a ser representados por um determinado número de genes, que são funções densidade de probabilidade como na equação 1 a seguir:

$$q_i \rightarrow [g_{i1} = p_{i1}(x), g_{i2} = p_{i2}(x), \dots, g_{iG} = p_{iG}(x),] \quad (1)$$

onde  $i = 1, 2, 3, \dots, N$ ,  $j = 1, 2, 3, \dots, G$  e as funções  $p_{ij}$  representam as funções densidade de probabilidade do gene  $j$  do  $i$ -ésimo indivíduo quântico.

Esta função densidade de probabilidade é usada pelo AEIQ-R para gerar os valores para os genes dos indivíduos clássicos. Em outras palavras, cada 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 respeitar a propriedade de normalização, ou seja, possuir a integral em seu domínio igual a 1.

Uma das funções mais simples que se pode usar como função densidade de probabilidade é o pulso quadrado. Esta função pode ser definida pela equação:

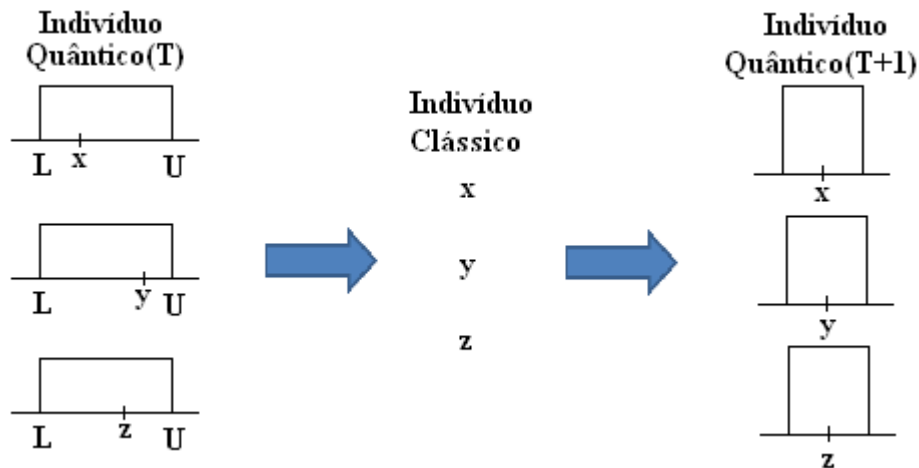
$$p_{ij}(x) = \begin{cases} \frac{1}{U_{ij} - L_{ij}}, & \text{se } x \in [L_{ij}, U_{ij}] \\ 0, & \text{caso contrário} \end{cases} \quad (2)$$

onde  $L_{ij}$  é o limite inferior e  $U_{ij}$  o limite superior do intervalo no qual o gene  $j$  do  $i$ -ésimo indivíduo quântico pode assumir valores (colapsar) quando observado. Esta equação respeita a propriedade de normalização mencionada no parágrafo anterior. Além disso, corresponde a uma distribuição de probabilidade que pode ser facilmente utilizada em algoritmos computacionais, já que a mesma corresponde a uma distribuição uniforme no intervalo  $[L_{ij}, U_{ij}]$ .

A partir de um indivíduo quântico podem ser gerados diversos indivíduos clássicos que, após avaliados e selecionados, são usados para atualizar a população quântica. Essa atualização, em geral, deve:

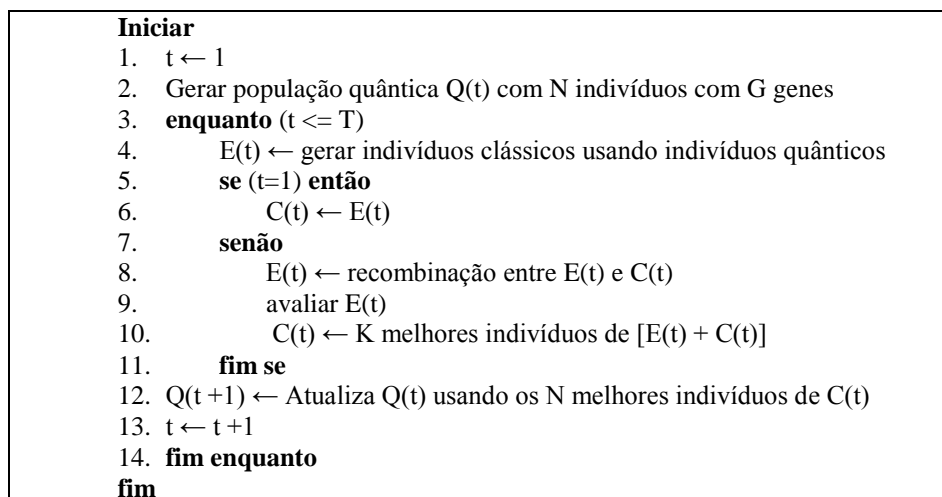
- Reduzir o espaço de busca da função que se deseja otimizar. No AEIQ-R isto é feito reduzindo-se o tamanho da região onde a função densidade de probabilidade (genes quânticos) tem probabilidade diferente de 0;
- Mapear as regiões mais promissoras do espaço de busca. Isto deve ser feito aumentando-se a probabilidade de se observar um determinado conjunto de valores para o gene clássico nas proximidades dos indivíduos mais bem sucedidos da população clássica.

Por simplificação, neste trabalho todos os genes partem da mesma distribuição de probabilidade e sofrem a mesma diminuição percentual do comprimento dos pulsos. A Figura 2 a seguir ilustra o processo em que os indivíduos quânticos são colapsados, ou seja, se transformam em indivíduos clássicos que podem ser avaliados e que, após selecionados, são utilizados para atualizar a população quântica.



**Figura 2** – Atualização da população quântica

O pseudo código utilizado para a modelagem quântica é apresentado na Figura 3.



**Figura 3** – Listagem completa do algoritmo evolutivo com inspiração quântica usando representação real

### 3 Neuro-Evolução

A neuro-evolução 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 neuro-evolução é buscar automaticamente a melhor configuração para uma rede neural usando algoritmos genéticos. Em outras palavras, a neuro-evolução 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.

Para a neuro-evolução funcionar não é necessário que o sistema satisfaça nenhuma restrição em particular: o mesmo pode ser contínuo e parcialmente observável. No que concernem os métodos de aprendizado por neuro-evolução, a única exigência é que os mesmos possam, de alguma forma, ter as suas soluções candidatas avaliadas relativamente umas às outras. Se o problema atende esta condição, estes métodos de aprendizado serão capazes de encontrar uma solução [15].

As abordagens possíveis para os sistemas neuro-evolutivos diferem umas das outras, basicamente, pelo modo como as mesmas codificam os pesos e a topologia das redes neurais nos cromossomos. Os cromossomos podem codificar qualquer informação relevante para a parametrização da rede neural, incluindo os pesos sinápticos, o número de camadas escondidas, a topologia da rede, a taxa de aprendizado, etc. A escolha do esquema de codificação tem um papel significativo na formação do espaço de buscas, no comportamento do algoritmo de busca e em como os genótipos devem ser transformados nos fenótipos (representação direta ou indireta).

O modelo descrito em [22] utiliza um algoritmo híbrido binário-real com inspiração quântica (AEIQ-BR) para otimizar parâmetros de uma rede neural como as variáveis a serem utilizadas na camada de entrada, o número de neurônios na camada escondida e o tipo da função de ativação em cada neurônio. [23] utiliza um modelo neuro-evolutivo composto por 3 redes neurais numa estrutura em cascata para prever o preço de energia elétrica. Em [24], é apresentada uma combinação entre o algoritmos de aprendizado por reforço SARSA( $\lambda$ ) e a neuro-evolução, com o objetivo de diminuir os espaço de estados e, conseqüentemente, melhorar o aprendizado da máquina.

No presente trabalho, a estrutura da rede está predefinida, sendo que só os pesos sinápticos e os *biases* da rede neural serão otimizados, conforme a metodologia descrita a seguir.

### 4 Metodologia

Como citado anteriormente, o *benchmark* utilizado foi o jogo da presa e caçador numa grade quadrada de tamanho 8 e contínua. O caçador deve caçar a presa um determinado número (“caçadas”) de vezes e as posições iniciais dos agentes são aleatórias e guardam uma distância maior que uma unidade.

A presa segue um movimento aleatório tanto na direção quanto na magnitude, limitada entre 0 e 1, e o movimento do caçador foi determinado pela saída da rede neural exibida na figura 4, onde  $x_p$  e  $y_p$  são as coordenadas x e y da presa,  $x_c$  e  $y_c$  são as coordenadas do caçador e  $dxc$  e  $dyc$  são os deslocamentos vertical e horizontal do caçador, respectivamente. Vale ressaltar que as entradas da rede foram normalizadas com o objetivo de deixar o modelo independente do tamanho da grade utilizado. A figura também contém a representação do cromossomo utilizado.

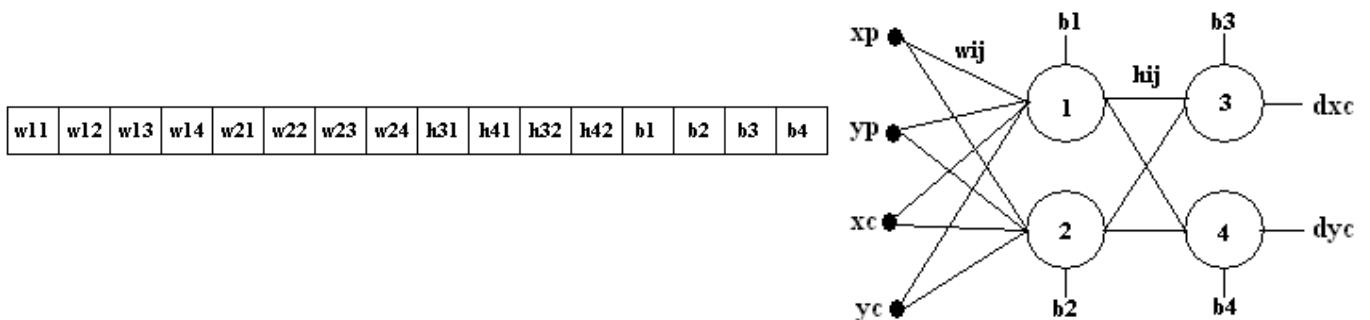


Figura 4 – Cromossomo e arquitetura da rede neural utilizada

A evolução dos cromossomos foi feita de 3 formas diferentes para efeito de comparação: i) um modelo aleatório, onde a atualização dos pesos da rede é feita gerando-se novos valores aleatoriamente; ii) um modelo simples, onde um algoritmo

genético tradicional atualiza os pesos sinápticos da rede neural; e iii) um modelo com inspiração quântica, onde os pesos da rede são atualizados através do algoritmo AEIQ-R. Todos os modelos utilizaram *steady-state*, os modelos simples e quântico utilizaram *crossover* aritmético enquanto que o modelo simples utilizou uma mutação *creep* que permite variações grandes no início e menores no final, conforme a equação 3. Mutações não são indicadas para algoritmos genéticos quânticos, dada a aleatoriedade já presente no modelo. O critério de seleção utilizado pelos algoritmos genéticos foi a seleção por roleta.

$$\Delta(y, t) = r \cdot y \cdot (1 - t/T)^B \quad (3)$$

A atualização dos pulsos quânticos seguiu a seguinte formulação: se a média das avaliações da população anterior fosse menor que o da população atual, diminui-se a largura do pulso quântico em  $\delta\%$ ; caso contrário, aumenta-se em  $\delta\%$ .

O desejado é que os caçadores aprendam a pegar a presa o mais rápido possível; portanto, ao atingirem o critério de parada, os caçadores mais bem avaliados são aqueles mais próximos da presa. Deste modo, a função de avaliação utilizada foi

$$Fitness = \sum_{k=1}^{caçadas} (passos_{ik} + dist.final_{ik})^{-1} \quad (4)$$

onde  $passos_{ik}$  é a quantidade de passos efetuados pelo caçador na k-ésima caçada e  $dist.final_{ik}$  é a distância final entre caçador e presa ao final da k-ésima caçada. A presa é considerada 'pega' pelo caçador se a distância entre eles for menor que 0,5 e o número máximo de passos que o predador pode dar para alcançar o objetivo é igual a 1000.

## 5 Análise dos Resultados e Discussão

Foram testadas diversas configurações para cada modelo e os cinco melhores resultados obtidos para cada um e os parâmetros utilizados podem ser vistos nas tabelas 1,2 e 3. Para efeito de comparação foram calculadas as avaliações médias dos indivíduos na última geração e foi exibida a melhor solução encontrada.

**Tabela 1 – Resultados do Modelo Aleatório**

População	Gerações	Caçadas	Gap-steady state	Média Final	Melhor solução
24	20	6	0,2	0,001181	0,002865
40	30	10	0,3	0,000675	0,001398
40	30	10	0,3	0,000675	0,001398
40	25	10	0,2	0,000605	0,001398
40	40	10	0,15	0,000558	0,001256
Média:				<b>0,000739</b>	<b>0,001663</b>

**Tabela 2 – Resultados do Algoritmo Genético Tradicional**

População	Gerações	Caçadas	Gap-	Crossover	Mutação	B- creep	Média	Melhor
24	20	6	0,2	0,7	0,05	1	0,001364	0,003906
24	20	6	0,2	0,6	0,02	1	0,001635	0,003434
40	30	10	0,3	0,7	0,05	1	0,000957	0,002304
40	30	10	0,3	0,7	0,05	1	0,000957	0,002304
56	30	10	0,2	0,6	0,05	2	0,000646	0,002158
Média:							<b>0,001112</b>	<b>0,002821</b>

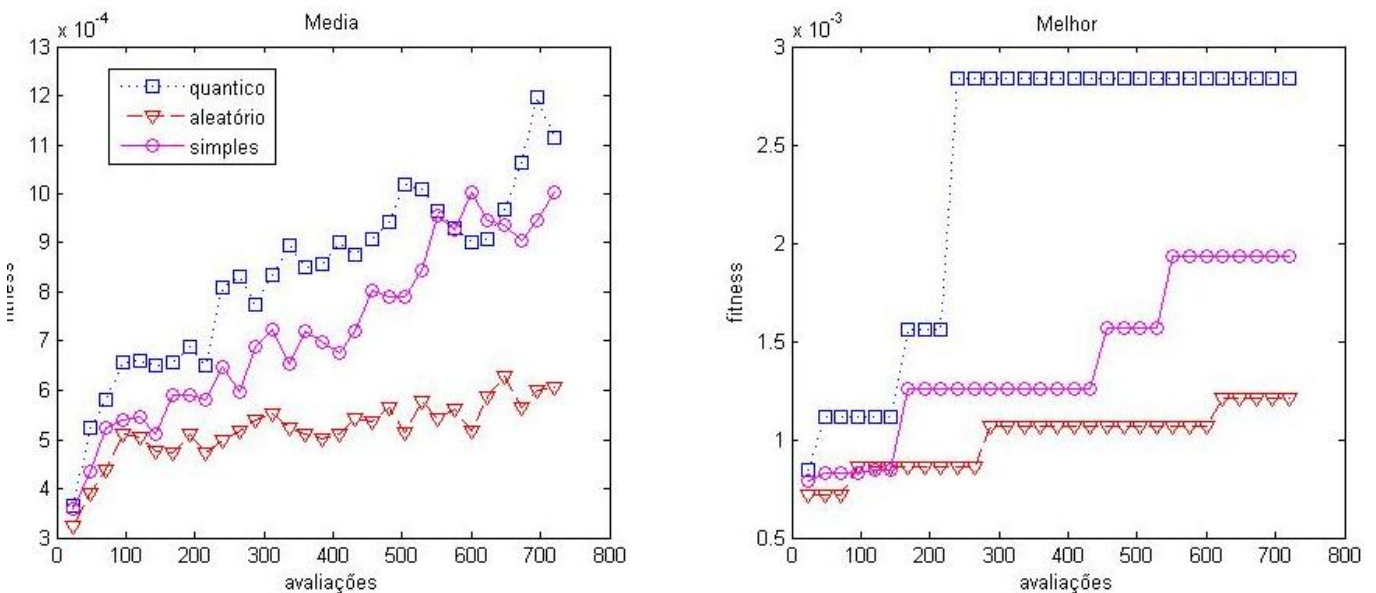
Comparando-se o algoritmo genético tradicional com o modelo de buscas aleatória, percebe-se um ganho médio de 50% em relação a aptidão média da população final. O ganho é ainda maior se a comparação for feita sobre as melhores soluções encontradas, onde o algoritmo genético teve uma média 70% maior.

**Tabela 3** – Resultados do Algoritmo Genético com Inspiração Quântica

População	Gerações	Caçadas	Gap-	Limite	Limite	Crossover	Ajuste dos	Média	Melhor
48	30	10	0,3	-2	2	0,8	0,25	0,001938	0,006355
24	20	6	0,2	-2	2	0,7	0,2	0,001636	0,004745
24	20	6	0,2	-1	1	0,6	0,3	0,001779	0,004657
40	25	10	0,2	-1	1	0,6	0,2	0,001041	0,00371
56	30	10	0,2	-2	2	0,6	0,2	0,001076	0,003665
Média:								<b>0,001494</b>	<b>0,004626</b>

Entretanto, os resultados do modelo de algoritmo genético com inspiração quântica superaram os resultados do modelo tradicional, apresentando uma melhora de 34% em relação à avaliação média final da população. Vale ressaltar também que as melhores soluções encontradas pelo algoritmo quântico são, em média, 64% melhor avaliadas, o que já justifica a metodologia proposta.

As comparações anteriores podem ser distorcidas pelas posições iniciais da presa e do caçador, pois para cada experimento, são geradas novas configurações iniciais. Outro problema é a diferença no número de avaliações que depende de parâmetros como o tamanho da população e o número de gerações. Com o objetivo de contornar esses problemas e visando uma melhor comparação entre os modelos, foram rodados, para um mesmo cenário inicial e o mesmo número de avaliações, os três modelos propostos e a evolução dos indivíduos pode ser vista na figura 5.



**Figura 5** – Comparação entre os modelos

Nota-se que, para um mesmo cenário e o mesmo número de avaliações, a modelagem com inspiração quântica continuou com um desempenho superior as modelagem tradicional e aleatória, tanto nas médias da avaliação da população, como na obtenção da melhor solução. Pode-se verificar que a solução obtida pelo modelo tradicional, foi avaliada em  $2 \times 10^{-3}$ , o que já é 53% superior ao modelo de busca aleatória, que obteve solução por volta de  $1,3 \times 10^{-3}$ . Entretanto, o modelo quântico obteve uma solução avaliada em  $2,8 \times 10^{-3}$ , 40% melhor que o modelo tradicional e 115% superior à solução encontrada pelo modelo aleatório.

## 6 Considerações Finais e Trabalhos Futuros

Este trabalho apresentou a proposta de um modelo neuro-evolutivo com inspiração quântica para o *benchmark* da presa e do caçador num *grid* contínuo e comparou os resultados com um modelo neuro-evolutivo que usa um algoritmo genético tradicional para a otimização dos cromossomos.

Verificou-se que com poucas avaliações o caçador aprendeu a caçar a presa, mesmo não sendo fornecidas estratégias sobre o comportamento esperado para o caçador, indicando que o trabalho cumpriu o objetivo de modelagem do problema presa-predador com inspiração quântica para um *grid* contínuo, diferentemente da maioria dos trabalhos anteriores. Mostrou-se

que a inspiração quântica pode fazer com que os sistemas inteligentes evoluam mais rapidamente, principalmente no que diz respeito ao tempo de convergência, conforme os resultados obtidos.

Este trabalho está sendo estendido para o estudo do *benchmark* da presa-caçador num cenário multi-agente, com o número de presas e caçadores variando. Também pode-se verificar como o algoritmo funcionaria para uma presa inteligente que evitaria o caçador, ao invés de se mover apenas aleatoriamente.

Pretende-se também estender o modelo proposto para a evolução da própria estrutura da rede neural, em termos de número de processadores e suas conexões.

## 7 Referências

- [1] Vlassis, N. “A concise introduction to multiagent systems and distributed AI”. Fac. Sci. Univ. Amsterdam, Amsterdam, The Netherlands, Tech. Rep. [Online]. Available: <http://www.science.uva.nl/~vlassis/cimasdai/cimasdai.pdf>, Sep, (2003).
- [2] Walid Chainbi, W., Hanachi C., Sibertin-Blanc, C. “The Multi-agent Prey/Predator problem: A Petri net solution” **Multiconference Computational Engineering in Systems Applications**, July (1996)
- [3] Corrêa, M.F., Vellasco, M., Figueiredo, K. Building Multi-Agent Systems with Reinforcement Hierarchical Neuro-Fuzzy Models, submetido ao **10<sup>th</sup> Brazilian Congress on Computational Intelligence (CBIC)**, (2011)
- [4] Brooks, R.R, Pang, J., Griffin, C. “Game and Information Theory Analysis of Electronic Countermeasures in Pursuit-Evasion Games” **IEEE Transactions On Systems, Man, And Cybernetics—A**: Vol. 38, No. 6, November, (2008)
- [5] Eklund, J.M., Sprinkle, J., Sastry, S.S. “Switched and Symmetric Pursuit/Evasion Games Using Online Model Predictive Control With Application to Autonomous Aircraft” **IEEE Transactions On Control Systems Technology** (2011)
- [6] Laumanns, M., Rudolph, G., Schwefel H. “A spatial predator-prey approach to multi-objective optimization: A preliminary study” **Lecture Notes in Computer Science Parallel Problem Solving from Nature** (2006)
- [7] Laporte, V., Nicolas J. M., Bernhard P., “About the resolution of discrete pursuit games and its applications to naval warfare.” **Lecture Notes in Control and Information Sciences Differential Games — Developments in Modelling and Computation** (2006)
- [8] Turetsky, V., Shinar, J. “Missile guidance laws based on pursuit-evasion game formulations.” **Automatica** Vol. 39 Issue 4, p607, abril (2003)
- [9] Mendel, J. M. “Fuzzy Logic Systems for Engineering: A Tutorial, **Proc. of the IEEE**”, Vol.83, n.3, (1995) 345-377.
- [10] Haykin, S. Neural Networks – A Comprehensive Foundation, **Macmillan College Publishing Company**, Inc. (1998)
- [11] Goldberg D. E. “Genetic Algorithms in Search, Optimization and Machine Learning.” **Addison-Wesley Publishing**. (1989)
- [12] Mitchell, Tom M., “Machine Learning”. **McGraw-Hill**. (1997)
- [13] Torres, G.ermano L. Silva, Alexandre P. A. da “Aplicação de Sistemas Inteligentes em Engenharia de Potência” **Revista SBA: Controle & Automação**. (2008)
- [14] Jang, J.R. “ANFIS: Adaptive-Network-Based Fuzzy Inference System.” **Electric Power Systems Research, IEEE Transactions on systems**, Vol. 23, No 3, (1993), 169-176
- [15] Abs da Cruz, A. V. Algoritmos evolutivos com inspiração quântica para problemas com representação numérica. **Tese de Doutorado, PUC-Rio**. (2007)
- [16] Yao, X. “Evolving artificial neural networks”. In: **Proceedings of the IEEE**, volumen 87, (1999) p. 1423–1447, 1.1
- [17] Ilonen, J.; Kamarainen, J.-K. ; Lampinen, J. “Differential evolution training algorithm for feed-forward neural networks. **Neural Processing Letters**, 17:93–105, 2003. 1.1(2003)
- [18] Han, K.; Kim, J. Genetic quantum algorithm and its application to combinatorial optimization problem. In: **Proceedings of the 2000 Congress on Evolutionary Computation**, volumen 2, 2.5(2000), 1354–1360
- [19] Jang, J.-S.; Han, K.-H. ; Kim, J.-H. “Face detection using quantum inspired evolutionary algorithm.” In: **Proceedings of Congress on Evolutionary Computation**, volumen 2, (2004), 2100–2106
- [20] Abs da Cruz, André V. Vellasco, Marley M.B.R. and Pacheco, M. “Quantum-Inspired Evolutionary Algorithm for Numerical Optimization”, **Book Series Studies in Computational Intelligence, Book Quantum Inspired Intelligent Systems**, Vol. 121, Nedjah, Nadia; Coelho, Leandro dos Santos; Mourelle, Luiza de Macedo (Eds.), Publisher Springer Berlin / Heidelberg, pp.115-132, 2008
- [21] Abs da Cruz, André V., Vellasco, Marley M.B.R. and Pacheco, M. Quantum-Inspired Evolutionary Algorithms Applied to Numerical Optimization Problems, **IEEE World Congress on Computational Intelligence (WCCI 2010), Congress on Evolutionary Computation**, pp. 3899- 3904, 2010
- [22] Vellasco, Marley M. B. R., Abs da Cruz, A. V., Pinho A. G. de. “Quantum-Inspired Evolutionary Algorithms Applied to Neural Modeling” **IEEE World Congress on Computational Intelligence (WCCI), International Joint Conference on Neural Networks (IJCNN2010)**, Plenary and Invited Lectures, IEEE Computational Intelligence Society, pp. 125-150, July 18-23, 2010.
- [23] Amjady, N. “Application of a new hybrid neuro-evolutionary system for day-ahead price forecasting of electricity markets”, **Applied Soft Computing** Vol. 10, n 3 p. 784-792, 2010
- [24] Wright, R. Gemelli, N. “Adaptive State Space Abstraction Using Neuroevolution”, **Communications in Computer and Information Science Agents and Artificial Intelligence Part 2**, 2010.