Um Modelo Neuro-Evolutivo de Coordenação Adaptativa em Ambientes Dinâmicos

Deise Côrtes Luis Otávio Campos Alvares

{deisesc,alvares}@inf.ufrgs.br
Universidade Federal do Rio Grande do Sul - UFRGS
Instituto de Informática
Av. Bento Gonçalves, 9500 - Campus do Vale - Bloco IV
Bairro Agronomia - Porto Alegre - RS -Brasil
CEP 91501-970 Caixa Postal: 15064

Resumo

Em ambientes dinâmicos e complexos, a política ótima de coordenação não pode ser derivada analiticamente, mas deve ser aprendida através da interação direta com o ambiente. Geralmente, utiliza-se aprendizado por reforço para prover coordenação em tais ambientes. Atualmente, neuro-evolução é um dos métodos de aprendizado por reforço mais proeminentes. Neste trabalho, é proposto um modelo de coordenação baseado em neuro-evolução. Foi desenvolvida uma extensão do método neuro-evolutivo conhecido como *Enforced Subpopulations* (ESP). Na extensão desenvolvida, a rede neural que define o comportamento de cada agente é totalmente conectada. Adicionalmente, é permitido que o algoritmo encontre, em tempo de treinamento, a quantidade de neurônios que deve estar presente na camada oculta da rede neural de cada agente. Esta alteração além de oferecer flexibilidade na definição da topologia da rede de cada agente e diminuir o tempo necessário para treinamento, permite também a constituição de grupos de agentes heterogêneos. Os experimentos realizados mostraram que os agentes treinados com o modelo proposto possuem capacidade de se adaptar a alterações no ambiente em tempo de execução. O modelo foi aplicado no domínio das tarefas de perseguição-evasão.

Palavras-Chave: Sistemas Multiagentes, Coordenação, Neuro-evolução, Redes Neurais, Algoritmos Genéticos.

1. Introdução

No contexto de sistemas multiagentes, coordenação é o processo pelo qual um agente raciocina sobre suas ações locais e as ações dos outros para tentar garantir que a comunidade se comporte de forma coerente. Em ambientes dinâmicos e complexos a política ótima de coordenação não pode ser derivada analiticamente, mas deve ser aprendida através da interação direta com o ambiente. Aprendizado por reforço é a técnica geralmente utilizada para prover a coordenação de agentes em tais ambientes (HAYNES 1995, HAYNES 1996, MORIARTY 1997, YONG 2001). Na aprendizagem por reforço, os agentes aprendem através de sinais que fornecem alguma medida de desempenho após a realização de uma seqüência de ações.

Existem duas estratégias básicas para a solução de problemas de aprendizagem por reforço. A primeira é a busca no espaço de comportamentos, procurando por um comportamento que

desempenhe bem no ambiente. Esta abordagem vem sendo geralmente implementada com algoritmos genéticos e programação genética. A segunda estratégia mais tradicional é o uso de técnicas estatísticas e métodos de programação dinâmica que estimam a utilidade de determinadas ações no espaço de estados do ambiente (mundo).

Segundo Kaelbling (1996) não é claro qual conjunto de abordagens é melhor e em que circunstâncias. Contudo, os métodos tradicionais de aprendizado por reforço que utilizam programação dinâmica, se apóiam num mapeamento dos estados e ações do mundo, tornando-os menos extensíveis. Assim, quanto mais o espaço de estados e ações do mundo crescer, mais lento será o processo de aprendizagem. Alguns pesquisadores (YONG 2001, MORIARTY 1997) constataram soluções mais rápidas e eficientes ao utilizarem neuro-evolução, um dos métodos disponíveis para aprendizado por reforço em ambientes dinâmicos.

Moriarty (1997) mostra que métodos evolucionários têm vantagem sobre os métodos tradicionais de aprendizado por reforço por terem um mecanismo de atribuição de crédito mais robusto. Outros pesquisadores (BELEW 1993, NOLFI 1994) descobriram que a neuro-evolução, ou evolução simulada de redes neurais, é uma estratégia efetiva para resolver problemas de aprendizado por reforço, mesmo quando o sinal de esforço é esparso. Portanto, mesmo quando o agente só é avaliado após a execução de uma seqüência de ações e não para cada ação tomada individualmente.

Neste trabalho, propõe-se um modelo de coordenação baseado em neuro-evolução. Estendemos o *Enforced Subpopulations*, método de neuro-evolução proposto por Gomez (1997), permitindo maior flexibilidade na definição da topologia da rede neural de cada agente.

Na próxima seção, é fornecida uma breve descrição de neuro-evolução. Na seção 3, é abordada a neuro-evolução. Na seção 4, apresentamos o modelo proposto. Na seção 5, mostramos alguns dos experimentos realizados. Por fim, na seção 6, este artigo é concluído.

2. Neuro-Evolução

Neuro-evolução é um método de aprendizado por reforço, no qual a busca pelo comportamento ideal é implementada através do aprendizado de redes neurais por algoritmos genéticos. Na neuro-evolução, cromossomos representam parâmetros das redes neurais, que podem ser, por exemplo, pesos, limites e conectividade. Estes cromossomos são recombinados baseados no princípio de seleção natural com o objetivo de se encontrar uma rede neural satisfatória para um dado problema.

Embora a pesquisa que vem sendo realizada na área de Redes Neurais tenha levado à descoberta de vários resultados teóricos e empíricos e ao desenvolvimento de várias aplicações práticas nas últimas décadas, o projeto de redes neurais para aplicações específicas ainda é um processo de tentativa e erro, onde, geralmente se leva em consideração a experiência passada em aplicações similares.

O desempenho de redes neurais em problemas particulares é criticamente dependente, entre outras coisas, do número de neurônios, da arquitetura e do algoritmo de aprendizagem utilizado (MORIARTY 1997). Estes fatores tornam o processo de projeto de redes neurais uma tarefa difícil. A falta de bons princípios para projeto constitui o maior obstáculo no desenvolvimento de sistemas de redes neurais em larga escala para uma variedade de problemas práticos.

Algoritmos evolucionários, por sua vez, oferecem uma abordagem aleatória, relativamente eficiente, para busca por soluções quase ótimas em uma variedade de domínios de problemas. O projeto de redes neurais eficientes para classes específicas de problemas é, portanto, um candidato natural para a aplicação de algoritmos evolucionários.

Os processos chaves na abordagem evolucionária para o projeto de arquiteturas neurais são ilustrados na Figura 1:

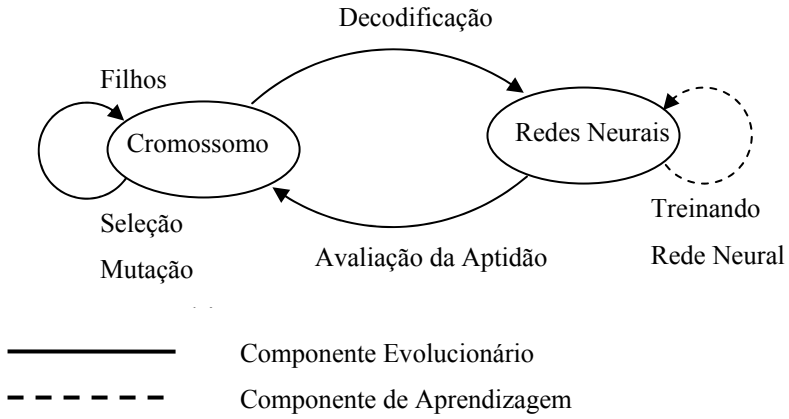

Figura 1: Processos da abordagem evolucionária.

O procedimento neuro-evolucionário trabalha em uma população de cromossomos, preferencialmente selecionando e reproduzindo aqueles que codificam redes neurais com melhores aptidões. Operadores genéticos, tais como mutação, recombinação, inversão, etc..., são usados para introduzir diversidade na população. Logo, após várias gerações, a população evolui gradualmente para cromossomos que correspondem às melhores redes neurais.

O *Enforced Sub-Populations* (ESP) é um método neuro-evolutivo proposto por Gomez (1997). No ESP, a população consiste de neurônios individuais em vez de redes inteiras, e a rede é formada por um subconjunto de neurônios. Existe uma população de neurônios para cada um dos **u** neurônios presentes na camada oculta da rede, e um neurônio pode ser recombinado somente com neurônios de sua própria sub-população. Vide Figura 2.

Cada população de neurônios tende a convergir para um papel que apresenta as melhores aptidões quando a rede é avaliada. Desta forma, ESP decompõe o problema de encontrar uma rede satisfatória em vários problemas menores. Segundo Yong (2001), esta abordagem resulta em uma evolução mais eficiente.

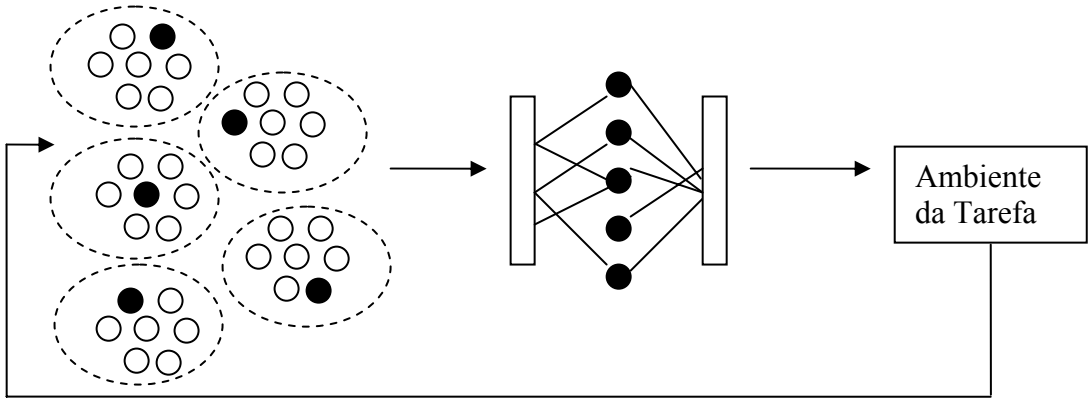

Figura 2: Neuro-evolução em ESP.

3. Modelo Neuro-Evolutivo

A base do modelo proposto de coordenação adaptativa é uma extensão do ESP, descrito na seção anterior. Na nossa extensão, cada neurônio da camada oculta se conecta com todos os neurônios da camada de entrada e todos os neurônios da camada de saída, ou seja, a rede que descreve o comportamento de cada agente é uma *2-layer-feedforward* totalmente conectada.

Contudo, a diferença mais importante do presente modelo é que a topologia da rede não é fixa. Embora seja possível definir o tamanho máximo de neurônios na camada oculta, é de responsabilidade do algoritmo buscar uma quantidade de neurônios ocultos que possibilite aos agentes executarem bem a tarefa. Desta forma, reduz-se a complexidade da topologia da rede. Adicionalmente, alguns experimentos, descritos na seção 5, comprovaram que o tempo de treinamento é reduzido significantemente a medida que diminuímos a quantidade de neurônios na camada oculta.

No passo 2 do algoritmo evolucionário proposto, descrito na Figura 3, uma nova rede neural experimental é construída para cada agente. No sub-passo **b**, é definido, aleatoriamente, quantos **s** neurônios devem existir na camada oculta. Posteriormente, no passo **c**, um neurônio é escolhido aleatoriamente da população correspondente, isto é, o neurônio **i** é escolhido aleatoriamente da população de neurônios **i**, onde i varia de 1 a **s**.

1. Crie **p** populações de neurônios ocultos para a rede neural de cada agente, onde **p** é o tamanho máximo da camada oculta.
2. Para cada agente
    a. Zere os valores de aptidão de cada neurônio
    b. Gere um inteiro s Є [0,p]
    c. Crie uma rede neural com **s** neurônios ocultos obtidos aleatoriamente da respectiva população
3. Avalie os agentes de acordo com uma função de aptidão apropriada para a tarefa.
4. Para cada agente
    a. Adicione a aptidão para a variável de aptidão de cada neurônio que participou da rede
5. Repita os passos de 2 a 4 um número suficiente de vezes
6. Calcule a aptidão média de cada neurônio, dividindo seu valor total de aptidão pelo número de redes em que ele participou.
7. Execute operações de recombinação na população baseada no valor de aptidão média de cada neurônio.

Figura 3: Algoritmo básico de treinamento.

De acordo com a complexidade da tarefa, algumas vezes não é possível obter um comportamento coerente do agente com evolução direta. Gomez (1997) propõe uma abordagem em que comportamentos complexos são obtidos através de um aprendizado incremental, isto é, os agentes são inicialmente treinados em tarefas mais simples e a medida que seus comportamentos convergem, o grau de dificuldade da tarefa é aumentado, tornando-a cada vez mais complexa.

O algoritmo de *Delta-Coding* foi utilizado para implementar as transições para tarefas cada vez mais complexas. A idéia do *Delta-Coding* (WHITLEY 1991) é buscar modificações ótimas para a melhor solução corrente. Quando a população de soluções candidatas converge, *Delta-Coding* salva a melhor solução corrente e inicializa uma população de cromossomos chamadas Δ–cromossomos. Os Δ–cromossomos possuem a mesma quantidade de genes que os cromossomos da melhor solução corrente, e eles consistem de Δ–valores que representam a diferença da melhor solução. Uma nova população é evoluída, selecionando-se Δ–cromossomos, adicionando seus Δ–valores a melhor solução corrente e avaliando o resultado. Os Δ–cromossomos que melhoram o resultado são selecionados para reprodução. Logo, *Delta-Coding* explora a vizinhança da melhor solução. *Delta-*

Coding pode ser aplicada várias vezes, com Δ–populações sucessivas representando as diferenças da melhor solução anterior.Em cada transição de tarefa, a melhor solução é salva, uma população de Δ-cromossomo é inicializada e esta população é evoluída para que o agente se adapte a nova tarefa. Quando a melhor solução da tarefa $\mathbf{t_i}$ evolui para a tarefa $\mathbf{t_{i+1}}$, os Δ-valores são adicionados ao melhor resultado do agente, formando um novo resultado melhor. A Figura 4, ilustra o algoritmo de *Delta-Coding*.

1. Crie **p** população de Δ-valores, onde **p** é o tamanho máximo da camada oculta.
2. Para cada agente
   a. Zere os valores de aptidão de cada Δ-valor
   b. Altere o conjunto de Δ-valores que serão adicionados aos pesos da rede
   c. Crie uma rede neural, somando os Δ-valores selecionados acima ao melhor resultado corrente.
3. Avalie os agentes de acordo com uma função de aptidão apropriada para a tarefa.
4. Para cada agente
   a. Adicione a aptidão para a variável de aptidão de cada Δ-valor somado aos pesos da rede
   b. Se o desempenho do agente melhorou com estes Δ-valores, some estes valores aos pesos do agente.
5. Repita os passos de 2 a 4 um número suficiente de vezes
6. Calcule a aptidão média de cada Δ-valor, dividindo seu valor total de aptidão pelo número de redes em que ele participou.
7. Desempenhe operações de recombinação na população baseada no valor de aptidão média de cada Δ-valor.

Figura 4: Algoritmo de *Delta-Coding*.

No modelo proposto, cada cromossomo corresponde a um neurônio da camada oculta e armazena os pesos das conexões que o neurônio representado tem com os demais das camadas de entrada e saída. Cada neurônio da rede é representado como uma cadeia de números reais. Considerou-se redes completamente conectadas, de forma que se uma rede possui, por exemplo, dois neurônios na camada de entrada e quatro na camada de saída, o neurônio oculto armazena um total de seis pesos, conforme Figura 5.

Para a recombinação dos cromossomos, foi utilizado o operador de recombinação aritmético, onde o gene de cada filho é obtido pela combinação linear dos genes correspondentes nos pais.

$$geneFilho_1 = \alpha * (genePai_1)+(1- \alpha)*genePai_2;\quad geneFilho_2 = (1- \alpha)*genePai_1+ \alpha *genePai_2;$$

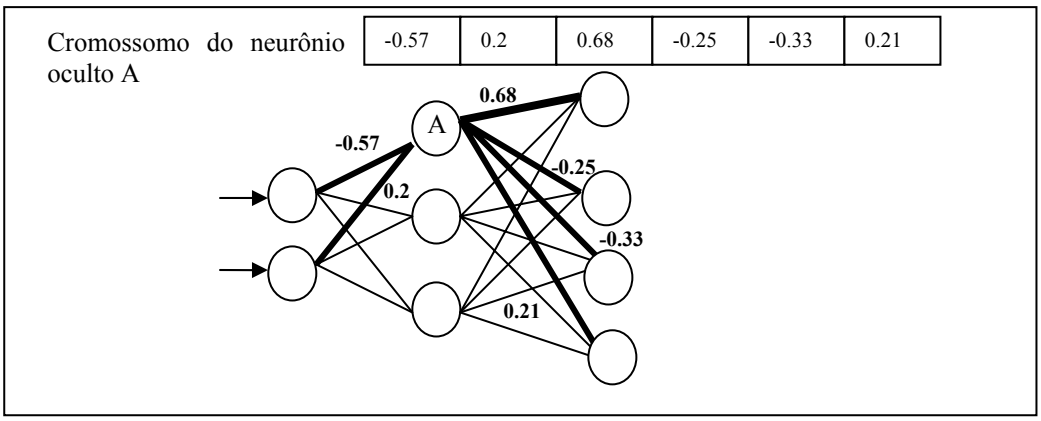

Figura 5: Codificação do neurônio no modelo proposto.

O operador de mutação é aplicado com dada probabilidade em cada um dos filhos gerado. Essa probabilidade é parametrizada, mas recomenda-se que seja um valor baixo, como 0.2, por exemplo. Foi utilizada a mutação aritmética, onde um alelo é substituído por um número real gerado aleatoriamente dentro do intervalo [-1,+1].

4. Estudo de Caso: Domínio da Presa-Predador

A tarefa de captura de uma presa, um caso especial de problemas de perseguição-evasão, foi utilizado, como estudo de caso, para validar o modelo proposto. Tarefas do domínio presa-predador consistem de um ambiente com uma ou mais presas e um ou mais predadores. Os predadores possuem o objetivo de capturar as presas, enquanto, as presas tentam escapar dos predadores. Tarefas desta natureza são interessantes porque existem no mundo real, oferecem um objetivo claro que requer coordenação complexa no que diz respeito ao ambiente, outros agentes com o mesmo objetivo e agentes adversários. Eles são verdadeiros desafios mesmo para os melhores sistemas de aprendizagem, permitindo medição precisa, análise e visualização das estratégias evoluídas.

O mundo considerado neste trabalho é constituído por uma presa e até quatro predadores. A presa é controlada por um dos seguintes algoritmos: se locomover para uma posição adjacente aleatória, caminhar sempre em uma mesma direção ou fugir do predador mais próximo. Os predadores são controlados por redes neurais. O objetivo é evoluir as redes neurais dos predadores para formar um time eficiente na tarefa de capturar a presa. A rede neural de cada predador recebe a percepção do mundo e fornece a direção para a qual o predador deve se locomover.

O ambiente é bidimensional e pode ser configurado para se comportar de forma toroidal ou limitada. Em qualquer das situações, não existem obstáculos. Todos os agentes podem se mover em quatro direções: **N** (Norte), **S** (Sul), **L** (Leste), **O** (Oeste). A velocidade e a posição inicial da presa podem ser configuradas de forma a aumentar ou diminuir a complexidade do mundo.

1. Para cada agente
    a. Zere os valores de aptidão de cada neurônio
    b. Gere um inteiro s Є [0,Qtde máxima de neurônios ocultos]
    c. Crie uma rede neural com s neurônios ocultos
    d. Posicione o agente no mundo bidimensional
2. Posicione a presa de acordo com a posição inicial especificada
3. Para os cenários de 1 à k
    a. Posicione os predadores em um extremo aleatório
    b. Para os passos de 1 à n
        i. Para cada agente, execute o próximo movimento
        ii. Movimente a presa de acordo com a velocidade especificada
    c. Calcule a distância média final d_f entre os predadores e a presa
    d. Avalie o time de acordo com a função abaixo:
        aptidao += 100/ d_f;
4. Calcule a aptidão média do grupo da seguinte maneira:
    aptidaoMedia = aptidão/k;
5. Para cada agente
    a. Adicione a aptidão para a variável de aptidão de cada neurônio que participou da rede
6. Repita os passos de 2 à 5 um número suficiente de vezes
7. Calcule a aptidão média de cada neurônio, dividindo seu valor total de aptidão pelo número de redes em que ele participou.
8. Desempenhe operações de recombinação na população baseada no valor de aptidão média de cada neurônio.

Figura 6: Ciclo evolucionário para a captura da presa.

A tarefa de captura de uma presa não-estacionária é uma tarefa de decisão seqüencial. Só é possível avaliar se o time de agentes possui uma boa política de decisão após a execução de uma seqüência de passos. Considere, por exemplo, que o ambiente possui a seguinte configuração: mundo toroidal, onde a presa se locomove na mesma velocidade que os predadores, e fugindo sempre do predador mais próximo. A primeira vista, poderia se pensar que uma boa estratégia seria que a cada passo de execução, os agentes de aproximassem da presa. Contudo, como o mundo considerado é toroidal, os predadores nascem todos em um mesmo extremo e como a presa se locomove na mesma velocidade que os predadores, a presa nunca seria capturada. Na Figura 6, é ilustrado o algoritmo evolucionário resultante para o problema de captura da presa.

Os agentes precisam "aprender" uma estratégia mais inteligente, onde alguns precisam impedir o avanço da presa em alguma direção enquanto os outros tentam captura-la.

Em cada ciclo evolucionário, para avaliar se a estratégia do grupo de predadores é satisfatória, foram criados diferentes cenários. Cada cenário consiste na alteração das posições iniciais dos predadores e para cada cenário criado, é permitido que os agentes se locomovam por **n** passos antes de seus comportamentos serem avaliados.

5. Experimentos

Foram realizados experimentos que avaliaram a adaptabilidade do comportamento aprendido pelos predadores em face da troca de estratégia da presa; e possibilitaram a descoberta de valores satisfatórios para os parâmetros do modelo proposto no estudo de caso utilizado.

Em cada experimento descrito nesta seção, o treinamento foi realizado de forma incremental. Os predadores eram inicialmente treinados com a presa estacionária. Em cada fase subseqüente do treinamento, a velocidade da presa era incrementada de 0,1 até que atingisse a velocidade de 1,0. Em cada troca de velocidade da presa, *Delta-Coding* era aplicado.

5.1 Experimentos Comparativos

Yong(2001) utilizou ESP/*Delta-Coding* para evoluir o comportamento dos predadores, em uma tarefa de perseguição-evasão, onde, a presa era considerada capturada quando um dos predadores tocassem nela.

A estratégia padrão utilizada pela presa nos experimentos de Yong(2001) era fugir do predador mais próximo. Os predadores da abordagem de Yong(2001), assim como os predadores que tiveram seus comportamentos evoluídos com o modelo proposto no presente trabalho, são capazes de capturar sempre a presa desde que o comportamento utilizado por ela, em tempo de execução, seja o mesmo usado no treinamento.

Yong(2001) realizou experimentos a fim de avaliar a adaptabilidade dos comportamentos dos predadores à mudança de estratégia da presa em tempo de execução. Nos experimentos reportados, os predadores foram treinados para capturar uma presa cuja estratégia era fugir do predador mais próximo. Em tempo de execução, a estratégia da presa era alterada para se movimentar sempre à direita.

Neste trabalho, foi realizado um experimento similar. Utilizou-se uma configuração para treinamento parecida com a que foi adotada por Yong(2001) no treinamento de seus agentes predadores. A comparação das configurações adotadas pode ser visualizada na Tabela 1. O tempo gasto para a realização de um experimento completo (treinamento inicial com presa estacionária e dez treinamentos com *Delta-Coding* incrementando-se a velocidade da presa a cada novo treinamento) durou cerca de 10 horas. A máquina utilizada para o treinamento possui um processador AMD Athlon™ Xp 2200+ 1.80GHz e 1.00GB de memória RAM.

Tabela 1: Comparação das configurações adotadas para treinamento dos agentes.

	Configuração Utilizada em Yong (2001)	Configuração do Modelo proposto
Tamanho de cada População de Neurônios	100	100
Elitismo	50	20
Tamanho (Máximo) da Camada Oculta	10	9
Quantidade de ciclos evolucionários	400	400
Quantidade de avaliações/cenários por ciclo	6	6
Quantidade de Experimentações (redes formadas por ciclo evolucionário)	1000	1000
Dimensão do Mundo	(100,100)	(100,100)

As configurações adotadas no presente trabalho variaram sutilmente em dois parâmetros: o elitismo e a quantidade de neurônios presentes na camada oculta de cada agente. No algoritmo evolucionário utilizado por Yong(2001), em cada ciclo de evolução, ele substituía apenas 50% dos piores neurônios de cada população, usando, desta forma, uma alta taxa de elitismo. No experimento realizado no presente trabalho, optou-se por uma taxa de elitismo de 20%.

Na abordagem de Yong(2001), a quantidade de neurônios presente na camada oculta era fixa e igual a 10. Nos comportamentos evoluídos neste experimento, utilizando o modelo proposto, existiam nove neurônios presentes na camada oculta de cada predador.

A Tabela 2 compara os resultados do modelo proposto com o modelo utilizado por Yong(2001). Executou-se 10 simulações para os comportamentos evoluídos. Em cada simulação, a presa iniciava em uma posição diferente no mundo, sempre com uma distância de, no mínimo, dez casas do predador mais próximo.

Tabela 2: Comparação de resultados com o ESP.

	Taxa de captura quando a presa troca de comportamento em tempo de execução
ESP	14,5 %
Modelo Proposto	80 %

5.2 Experimentos Variando Diversos Parâmetros

Na abordagem neuro-evolucionária, muitos parâmetros precisam ser configurados. Realizou-se uma série de experimentos a fim de avaliar quais os melhores parâmetros para a tarefa proposta, tratados a seguir.

5.2.1. Entrada do Agente

A percepção que o agente tem do mundo pode ser categorizada como parcial ou total. Na percepção parcial, o agente recebe informações relativas a ele e a presa. Na percepção total, o agente recebe informações relativas a ele e ao resto do grupo de predadores:

Foram avaliados seis tipos de entrada para os predadores, sendo que três forneciam ao agente uma percepção parcial do mundo, enquanto as outras três forneciam uma percepção total do mundo:

Entradas de percepção parcial:

- Sinal do *offset* [x,y] entre o agente atual e a presa,
- *Offset* [x,y] entre o agente e a presa,
- Coordenada absolutas [x_a,y_a,x_p,y_p] do agente atual e da presa.

Entradas de percepção total:

- Sinal do *offset* [x,y] entre o agente atual e a presa e o agente atual e os outros n-1 predadores,
- *Offset* [x,y] entre o agente atual e a presa e o agente atual e os outros n-1 predadores,
- Coordenada absolutas [x_{a1},y_{a2},...,x_{an},y_{an},x_p,y_p] de todos os agentes do sistema .

Os agentes só se mostraram capazes de aprender como capturar a presa quando as entradas correspondiam ao sinal do *offset*. Informações envolvendo distâncias e posições absolutas são entradas mais complexas e dificultam o aprendizado da tarefa.

Para este experimento, foi utilizado somente o treinamento inicial com a presa estacionária. O time com a percepção parcial do mundo, conseguia alcançar a presa com uma quantidade menor de passos. Observe o resultado de dez simulações consecutivas, cada uma constituída de um novo treinamento e uma nova execução, conforme ilustra a Tabela 3:

Tabela 3: Passos para captura da presa.

Execução	**Time com Percepção Total do Mundo**	**Time com percepção parcial do Mundo**
1	35	29
2	33	29
3	43	29
4	53	31
5	37	33
6	35	31
7	34	31
8	41	31
9	55	29
10	47	31
Média	**41.3**	**30.4**

5.2.2. Quantidade de Neurônios na Camada Oculta

Foi avaliada a taxa de captura da presa em função da quantidade máxima de neurônios na camada oculta. Foram realizadas dez execuções para cada quantidade máxima de neurônios na camada oculta. O resultado destas execuções pode ser visualizado na Tabela 4. Os predadores se mostraram capazes de capturar a presa, na maior parte das vezes, mesmo quando tinham, no máximo, três neurônios na camada oculta da rede.

Tabela 4: Taxa de captura por tamanho da camada oculta.

Quantidade de Neurônios	Taxa de Captura (%)
3	80
4	90
5	100
6	100
7	90
8	100
9	90
10	90

Avaliou-se também o tempo necessário para o treinamento em função do tamanho da camada oculta. Para este experimento foi considerado apenas o treinamento inicial com a presa estacionária, sem *Delta-Coding*. Na Tabela 5, mostra-se o tempo de treinamento por quantidade de neurônios na camada oculta. A camada oculta com menor quantidade de neurônios (3) possui um tempo de treinamento 20% menor que o da camada com a maior quantidade de neurônios (10).

Tabela 5: Tempo de treinamento por tamanho da camada oculta.

Quantidade de Neurônios	Tempo Médio de Treinamento (ms)
3	56367
4	59649
5	60955
6	59917
7	60528
8	62840
9	67503
10	70568

5.2.3. Quantidade de ciclos evolucionários

Neste conjunto de experimentos, buscou-se identificar a quantidade adequada de ciclos evolucionários. Foi observado que as aptidões dos neurônios se estabilizaram a partir do ciclo 300, conforme pode ser visualizado na Figura 7.

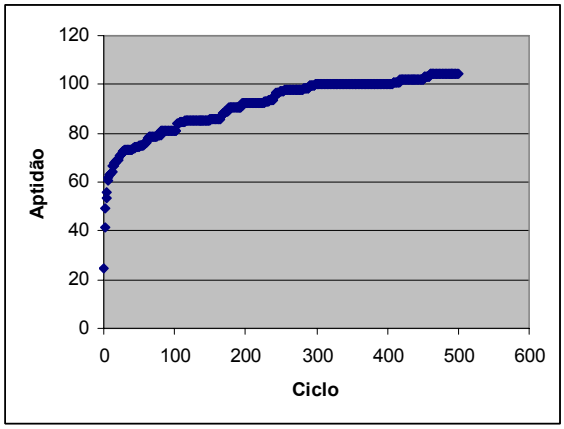

Figura 7: Média das melhores aptidões dos agentes por ciclo evolucionário. Time constituído por 3 agentes.

6. Conclusões e Trabalhos Futuros

Em ambientes dinâmicos e complexos, a política ótima de coordenação não pode ser derivada analiticamente, mas deve ser aprendida através da interação direta com o ambiente. Neste trabalho, propomos um modelo de coordenação baseado em neuro-evolução, que, de acordo com vários pesquisadores (MORIARTY 1997, MCQUESTEN 2002, YONG 2001, STANLEY 2002) é um dos métodos mais promissores para aprendizagem em ambientes estocásticos.

Para testar o modelo, utilizou-se a tarefa da presa-predador, um caso especial de uma classe de problemas conhecidos como perseguição-evasão. Nesta tarefa, uma presa tem por objetivo fugir de um ou mais predadores, enquanto que os predadores precisam se coordenar de forma a capturá-la.

A tarefa da presa-predador é praticamente um *benchmark* para modelos de coordenação, por basicamente duas razões: primeiro por simularem situações que ocorrem no mundo real e segundo porque, a depender da configuração do ambiente e da estratégia de fuga da presa, exige elevado grau de coordenação entre os predadores.

Foi realizado um estudo sobre os modelos de neuro-evolução existentes na literatura, e proposto uma extensão do método neuro-evolutivo conhecido como ESP. Ao contrário do ESP, que trabalha com uma topologia fixa para as redes neurais dos agentes, o nosso modelo permite que o algoritmo evolucionário encontre, em tempo de treinamento, a topologia da rede neural de cada agente.

Além de propiciar flexibilidade, o nosso modelo possibilita uma otimização do tempo necessário para o treinamento. Nossos experimentos mostraram que a quantidade de neurônios presentes na camada oculta de cada agente, no final do treinamento, é, freqüentemente, inferior à quantidade máxima especificada. Adicionalmente, os experimentos também mostraram que o tempo de treinamento aumenta proporcionalmente com o tamanho da camada oculta. Nos experimentos foi constatado que o tempo de treinamento quando o agente possui três neurônios na camada oculta é 20% menor que o tempo necessário para o treinamento com 10 neurônios na camada oculta. Este último resultado já era esperado, uma vez que uma quantidade maior de neurônios implica em maior complexidade computacional.

A variação na topologia da rede dos agentes, em tempo de treinamento, levou a um outro resultado: a construção de um time heterogêneo de agentes, no que diz respeito à topologia da rede neural de cada agente, o que é mais coerente com o papel diferenciado que cada agente possui no grupo. Considere, por exemplo, a tarefa mais difícil para a qual obteve-se sucesso com o modelo proposto: um mundo toroidal, onde a presa se movimenta na mesma velocidade que os predadores, fugindo

sempre do predador mais próximo. Se todos os agentes possuíssem o papel de correr atrás da presa, eles nunca iriam conseguir capturá-la, pois, todos os agentes (presa e predadores) ficariam girando na mesma direção ao redor do mundo. Desta forma, os agentes precisam evoluir comportamentos diferenciados, isto é, parte dos agentes persegue a presa enquanto outra parte impede que a presa avance em determinada direção.

Os melhores comportamentos foram obtidos através de um processo de aprendizagem incremental, isto é, começando com uma tarefa simples: capturar uma presa estacionária e finalizando com uma tarefa mais complexa, por exemplo, capturar uma presa que se movimenta na mesma velocidade que os predadores, fugindo sempre do predador mais próximo.

Nos experimentos apresentados na seção anterior, mostrou-se que o modelo proposto tem boa capacidade de adaptação em função da alteração no comportamento da presa. Em trabalhos futuros, pretendemos desenvolver um modelo em que os agentes sejam capazes de compreender possíveis alterações no mundo e alterar seu comportamento, em tempo de execução, de forma a suportar estas alterações sem a necessidade de um novo treinamento. Uma forma possível de realizar tal procedimento seria fazer com que os agentes armazenassem as várias estratégias aprendidas para as diversas variações do mundo e em tempo de execução, identificassem qual estratégia melhor se aplica à percepção atual do mundo. Essa identificação poderia ser realizada por uma rede neural a parte.

Referências

BELEW, R. K. **Interposing an ontogenic model between genetic algorithms and neural networks.** In Advances in Neural Information Processing Systems (NIPS),S.J. Han- son and J.D. Cowan and C.L.Giles,Morgan Kauffman:San Mate,1993.

GOMEZ, F. & MIIKKULAIEN, R. **Incremental evolution of complex general behavior.** Adaptive Behavior, vol. 5, pp. 317-342, 1997.

HAYNES, THOMAS & SEN, S. **Evolving Behavioral Strategies in Predators and Prey.** Adaptation and Learning in Multiagent Systems, Lectures Notes in Artificial Intelligence, Spring Verlag, Berlim, Spring 1996.

HAYNES, T., WAINWRIGHT, R. & SEN S. **Evolving Cooperation Strategies.** In Proceedings of the First International Conference on Multiagent Systems, ed., Lesser, V., pp. 450, MIT Press, San Francisco, CA., 1995.

KAELBLING, L., LITTMAN, M. AND MOORE, A. **Reinforcement Learning: A Survey.** Journal of Artificial Intelligence Research, 4:237-285, May, 1996.

MORIARTY, DAVID E. **Symbiotic Evolution Of Neural Networks In Sequential Decision Tasks. Ph.D. Dissertation**; Technical Report AI-97-257, January 1997.

NOLFI, S., AND PARISI, D. **Desired answers do not correspond ecological neural networks.** Neural Processing Letters, (2):1–4, 1994.

YONG, CHERN & MIIKKULAIEN, RISTO**. Cooperative Coevolution of Multi-Agent Systems.** Technical Report AI01-287, Department of Computer Sciences, The University of Texas at Austin, 2001.

WHITLEY, D.; MATHIAS, K.; FITZHORN, P. **Delta-coding: An iterative search strategy for genetic algortihms**. In Proceedings of the Fourth International Conference on Genetic Algorithms. Los Altos, CA: Morgan Kaufmann, 1991.