

Aplicação de Redes Neurais e Algoritmos Genéticos em Jogos de Corrida

André Grahl Pereira Cesar Tadeu Pozzer Erick Baptista Passos*
Marcos Cordeiro d'Ornellas

UFSM, Departamento de Eletrônica e Computação, Brazil

*UFF, Instituto de Computação, Brazil

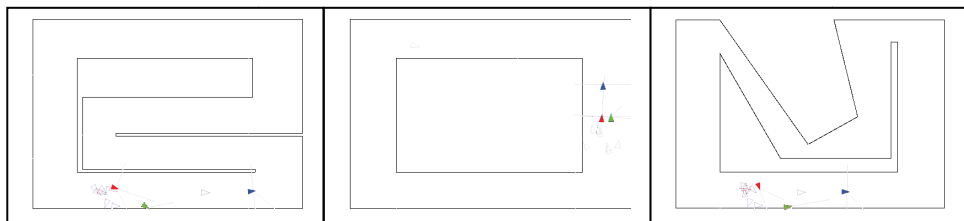


Figura 1: Variação de pistas, modelo de população e veículos utilizados nas simulações.

Abstract

Este artigo apresenta um modelo alternativo de inteligência artificial para jogos de corrida. O problema abordado é a criação de um agente para controle de um automóvel de corrida com comportamento semelhante ao de um ser humano, suscetível a erros e adaptável a situações emergentes no decorrer do jogo. O sistema proposto combina redes neurais artificiais, treinamento *off-line* por meio de algoritmos genéticos e adaptação em *real-time*.

Keywords: redes neurais, algoritmos genéticos, jogos de corrida, aprendizagem em tempo real

Authors' contact:

{grahl, pozzer, ornellas}@inf.ufsm.br
@epassos@ic.uff.br

1. Introdução

Jogos eletrônicos são uma indústria em expansão. A busca por melhorias é constante, seja no contexto de realismo físico e gráfico, como no caso de jogos como *Crysis* [Crysis 2007], ou na elaboração de histórias com grande imersão, como no caso de *Máfia 2*, que possui um roteiro de 700 páginas [Ekberg 2009].

Nessa busca por realismo nos jogos, o aperfeiçoamento gráfico e físico, bem como a complexidade das histórias e do game-design são abordagens importantes, mas outro foco de especial interesse atual é o desenvolvimento de inteligência artificial (IA) mais convincente.

Neste contexto, o uso de técnicas clássicas de inteligência artificial, como máquina de estados finitos, apesar de eficientes, demonstram pouco realismo [Buckland 2002]. O objetivo é desenvolver agentes mais realistas, que não sejam apenas eficientes e acertem sempre, e sim cometam erros,

preferencialmente de maneira semelhante a um ser humano. Técnicas adaptativas de IA são uma alternativa neste contexto. Redes neurais e algoritmos genéticos apresentam características desejáveis como: adaptação, aprendizagem e diversidade de reações em situações idênticas propiciada pela variação dos pesos da rede.

No caso específico de jogos de corrida, o desenvolvimento de NPC's (*Non-player characters*) é uma situação ainda mais complexa. Para que o NPC gerado seja um bom adversário, uma série de qualidades são necessárias tais como: velocidade de reação, previsão, criatividade, aprendizagem e adaptação [Togelius 2005].

A abordagem mais utilizada neste tipo de problema é uma combinação de linhas de corrida com máquina de estados finitos [Biasillo 2002]. A linha de corrida é utilizada como guia para o veículo, delimitando o caminho pelo qual ele deve percorrer a pista. Neste caso o NPC já conhece previamente a pista antes mesmo da corrida ser iniciada. Pode-se desenvolver uma linha de corrida perfeita criando um NPC invencível, o que foge dos objetivos de uma inteligência artificial realista.

Este cenário se mostra ideal para utilização de técnicas adaptativas de IA com aprendizagem em tempo real. No entanto, autores como Spronck [Spronck 2003] não as considera como candidatas para esta utilização, pois não satisfazem quatro dos requisitos apresentados pelo autor: rapidez, eficácia, robustez e eficiência. No entanto, outros autores tais como Yannakakis [Yannakakis 2005], que desenvolveu um jogo utilizando o modelo predador-presa utilizando um algoritmo evolutivo no aprendizado on-line de agentes controlados pelo computador através de uma rede neural, e Crocomo [Crocomo 2006] que elaborou um jogo em que as estratégias adotadas pelos agentes controlados pelo computador se adaptam de acordo com as estratégias utilizadas pelo jogador a cada vez

que este joga, demonstraram que é viável a utilização de aprendizagem *on-line* em jogos eletrônicos.

Neste trabalho é apresentado um modelo alternativo de inteligência artificial para jogos de corrida. O sistema tem como fundamentos a criação de agentes com comportamento semelhante ao de um ser humano, suscetível a erros e adaptável, para o controle de um carro de corrida em um jogo.

Para tanto, o sistema usa uma combinação de treinamento *off-line* e adaptação *on-line*. São usadas informações simuladas que são semelhantes às disponíveis para um jogador ou piloto, ou seja, não são necessários *waypoints*, como na maioria das abordagens anteriores para esse problema. O agente utiliza como entrada de dados apenas as informações obtidas por um campo de visão baseado em traçamento de raios.

Na seção 2 são apresentados trabalhos relacionados e diferenças em relação ao atual trabalho. Na seção 3 é apresentada uma breve descrição protótipo de jogo de corrida utilizado nos experimentos. Na seção 4 é discutido o sistema de aprendizagem e de adaptação. As seções 5 e 6 discutem aspectos relevantes referentes a rede neural e ao algoritmo genético. Na seção 7 são apresentados dados e comentários sobre os experimentos realizados. A seção 8 contém a conclusão e trabalhos futuros.

2. Trabalhos Relacionados

Em Togelius [Togelius 2005] também foi utilizada uma arquitetura de redes neurais multicamadas e algoritmos genéticos para a criação de um agente piloto de automóveis. Foram utilizadas três configurações de pistas diferentes com um veículo sem adversário e dois tipos de sensores: os de pistas e dos *waypoints* que recebiam distância e direções dos objetivos. O objetivo desse trabalho foi experimentar a capacidade de especificação da rede neural e dos sensores utilizados. Entretanto, dado o tipo de entrada de dados utilizada, que incluiu o uso de *waypoints*, observou-se que os parâmetros ganhavam configurações muito específicas para cada pista, sendo impraticável usar um agente treinado para uma pista em outra. O nosso sistema utiliza informações equivalentes à inferência visual de um piloto, sendo possível treinar o agente com uma pista e utilizá-lo em qualquer outra.

Stanley [Stanley 2005] desenvolveu um sistema que utiliza redes neurais e algoritmos genéticos como predictor de colisões. Os testes foram conduzidos em uma pista com o simulador RARS [RARS 2009], onde o sistema recebe como entradas a distância de objetos e das laterais da própria pista. Esses dados são inseridos na rede neural e se obtêm como saída um nível de alerta para uma possível colisão. O nosso sistema usa uma abordagem semelhante para a entrada de dados,

mas faz uso de um agente piloto completo para ser utilizado em uma corrida em tempo real.

Em Wloch [Wloch 2004], utilizou-se algoritmos genéticos para a otimização de parâmetros no simulador *Formula One Challenge '99-'02*. Foi demonstrado que este é um método eficiente na melhora no desempenho e conseqüentemente na obtenção de melhores tempos.

No trabalho de Vieira [Vieira 2006], foi apresentado um sistema que utiliza pontos guia para direcionar a trajetória do veículo em uma pista seccionada. Os pontos guia são obtidos através de um sistema de aprendizagem que utiliza algoritmos genéticos na busca por melhores referências.

3. Protótipo do Jogo de Corrida

Para os experimentos, foi desenvolvido em OpenGL um simulador 2D de jogos de corrida. A pista é delimitada por barreiras sólidas, que são também identificadas pelo sistema de visão, o qual é simulado pelo traçamento de raios. Um sistema simples de simulação física foi implementado, incluindo atrito e forças.

Os veículos são representados por formas geométricas simples e três pistas de configurações distintas foram criadas para possibilitar diversos experimentos, como mostrado na Figura 1.

4. Sistema de Aprendizagem

O sistema de aprendizagem (Figura 2) é utilizado em dois momentos distintos: na adaptação *on-line* e na aprendizagem *off-line*. Em ambos os momentos, sempre existe uma população de veículos participando da simulação. Dois grupos de veículos são utilizados: os corredores e os fantasmas.

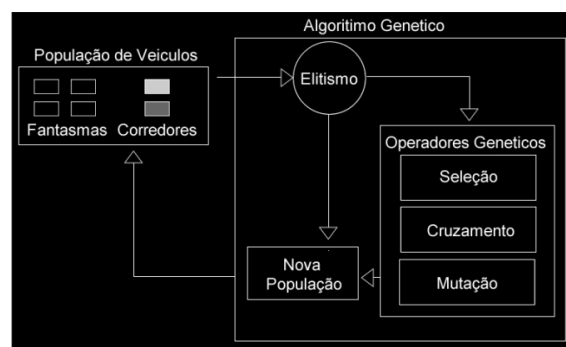


Figura 2: Modelo do sistema de aprendizagem

Os corredores são formados pelos indivíduos da população que obtiveram o melhor desempenho na geração anterior. Como as colisões entre esses são consideradas, um corredor afeta o comportamento dos demais. Esses são os indivíduos com os quais o jogador interage.

Os fantasmas são formados pelo restante da população. Entre esses não existe colisão, mas os mesmos levam em consideração os veículos do jogador e do grupo dos corredores para o seu próprio aprendizado e simulação.

4.1 Aprendizagem Off-line

Na aprendizagem *off-line*, inicialmente a população de veículos é criada aleatoriamente. Esta fase é dividida em dois momentos, antes e após a pista ser completamente percorrida.

Antes da pista ser percorrida, o valor da função de avaliação para o algoritmo genético é dado pela quantidade de *checkpoints* ultrapassados, o número de *checkpoints* varia de pista para pista sendo usados em pontos específicos como curvas ou em fim de obstáculos, pontos que teoricamente determinariam um aprendizado. Observa-se que os *checkpoints* não são usados pelo sistema *runtime* de direção, sendo apenas condição necessária para a escolha inicial de indivíduos que "progridem" na pista, do contrário a geração inicial aleatória poderia não conter indivíduos capazes de completar uma volta, dificultando a avaliação do algoritmo genético.

Quando um indivíduo é capaz de dar uma volta completa, sua avaliação é dada pelo tempo de duração de uma volta. Após um limite de tempo, ou após todos os indivíduos completarem uma volta na pista, o algoritmo genético é então ativado para evoluir a população.

Os pesos dos neurônios de cada indivíduo são o seu genoma, sendo que a aptidão de cada um é o desempenho obtido pelo método descrito acima. Os genomas com maior aptidão (grupo elite) são diretamente transferidos para a nova população. Com essa técnica de elitismo, esses corredores farão parte da próxima geração sem sofrer influência de operandos de permutação e mutação, garantindo assim um desempenho mínimo igual ao da presente geração.

O restante da população é utilizado normalmente com o uso da permutação e mutação, incluindo o grupo elite. Após algum tempo de treinamento, são obtidos corredores com habilidades mínimas que podem ser utilizados com o mecanismo de adaptação *on-line*.

4.2 Adaptação On-line

Na adaptação *on-line*, os corredores obtidos na aprendizagem *off-line* são utilizados para um maior refinamento. Isso garante um desempenho mínimo ao sistema perante o jogador, o que é uma característica importante para jogos casuais, onde o jogador normalmente não é um *expert*. Dessa forma, também são excluídos comportamentos aleatórios ou demasiadamente inesperados, mas garantindo a variabilidade necessária para que haja candidatos de

soluções suficientes para a adaptação em relação ao jogador.

5. Rede Neural

A rede neural é responsável por todo o comportamento do veículo, a cada instante recebe os dados do ambiente, processa de acordo com seu genoma e responde com um conjunto de saídas. Graças à variedade genética dos genomas, cada veículo responde de forma diferenciada a eventos idênticos. Sua configuração é de oito entradas (uma para cada sensor), uma camada oculta com 25 neurônios e quatro saídas. As quatro saídas são os comandos do veículo virar a direita, esquerda, acelerar e frear. O sensoramento da pista é realizado por 8 sensores de tamanho fixo presentes em cada veículo, como ilustrado na Figura 3.



Figura 3: Configuração dos Sensores dos veículos

Quando o sensor não está tocando em nenhum objeto assume o valor de entrada -1. Quando o sensor toca em algum objeto o valor de entrada é a distância, que é normalizada entre 1 e 0, valendo 0 quando o ponto externo do sensor toca o objeto e 1 quando o ponto interno do sensor toca o objeto.

6. Algoritmo Genético, Seleção, Cruzamentos e Mutação

O operador usado na seleção é o torneio, onde inicialmente um membro da população é escolhido aleatoriamente. Ele é então comparado com '*n*' outros membros escolhidos aleatoriamente. Como a comparação é feita com dois membros o que possuir melhor desempenho permanece e o outro é descartado. Ao final das '*n*' comparações o membro vencedor é escolhido para o cruzamento.

O cruzamento é feito através do operador de multipontos por partes, onde múltiplos pontos definidos aleatoriamente dos genomas escolhidos são trocados dentro de faixas validadas, ou seja, só podem ser trocados pesos de um mesmo neurônio (Figura 4).

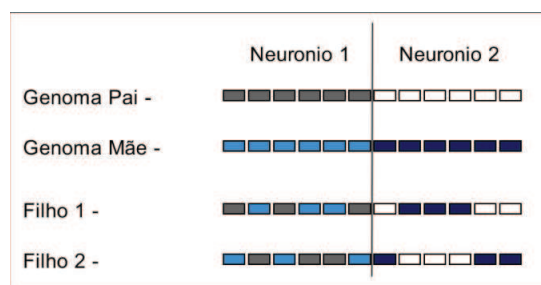


Figura 4: Cruzamento entre dois membros selecionados pelo operador mutação

O operador mutação é usado para multiplicar uma faixa aleatória da rede neural por valores aleatórios, gerando assim uma perturbação na rede neural, o que possibilita o surgimento de novas características (Figura 5).

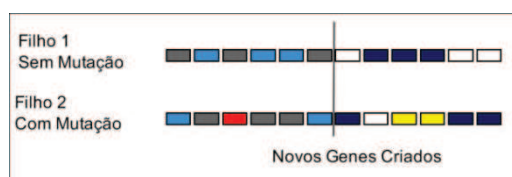


Figura 5: Mutação dos filhos gerados pelo operador cruzamento

7. Experimentos Realizados

Durante os experimentos *off-line* o sistema foi capaz de em pouco mais de nove gerações treinar indivíduos capazes de completar a pista de forma satisfatória, e em torno das 150 gerações o sistema alcançou a estabilidade (Figura 6).

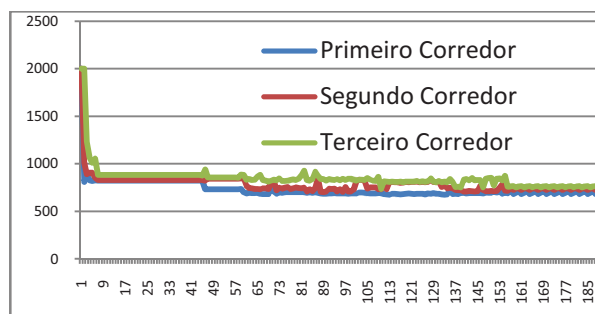


Figura 6: Desempenho dos corredores num experimento

Os corredores obtiveram comportamentos interessantes como o de buscar ultrapassagens por caminhos alternativos e o de impedir ultrapassagens fechando a frente de seus adversários. Também buscaram o caminho ótimo ao percorrer as curvas de forma fechada e fazendo as transições entre as retas da maneira mais breve possível, visando diminuir a distância total do trajeto.

Nos testes de adaptação *on-line*, resultados igualmente interessantes foram obtidos. Os corredores conseguiram evitar dificuldades impostas pelo jogador como em casos onde o jogador ficava colidindo constantemente contra os corredores. Nesta situação os corredores se adaptaram e usaram posturas evasivas, matendo distância dos adversários e assim evitando as colisões.

8. Conclusão e Trabalhos Futuros

Após a análise dos resultados obtidos, pôde-se concluir que o sistema proposto é capaz de trazer bons resultados para o problema de desenvolver um NPC que possua características como velocidade de reação, previsão, criatividade, aprendizagem e adaptação.

Para trabalhos futuros pretende-se testar esse sistema em um simulador 3D completo como um jogo de ocorrência real. Também pretende-se agregar mais complexidade à rede neural usando NEAT, que é uma técnica que permite a evolução da topologia da rede neural, juntamente com um sistema de treinamento mais robusto, além de elaborar um método que permita agregar conhecimento a todo sistema possibilitando o aprendizado efetivo por exemplo após completar uma pista.

9. Referências

- BIASILLO, G., 2002b. Racing AI Logic. RABIN, S. AI Game Programming Wisdom. New York: Charles Iwer Media. p.444-454.
- BUCKLAND, M., 2002. AI Techniques for Game Programming. 1 ed. Premier Press, 2002.
- CROCOMO, M. K., 2006. Desenvolvimento de um Jogo Adaptativo Utilizando um Algoritmo Evolutivo. *Dissertação (Graduação) – Universidade de São Paulo – Instituto de Ciências Matemática e Computação (USP-ICMC), São Carlos, SP.*
- CRYSISEU, 2007. *Crysis Graphics*. Available from: <http://crysiseu.com/content/view/19/43/> [Accessed 23 June 2009].
- EKBERG B., 2009. *Mafia II Impressions*. Available from: <http://www.gamespot.com/xbox360/action/mafia2/news.html?sid=6211649> [Accessed 23 June 2009].
- RARS, 2009. *Robot Auto Racing Simulator*. Available from: <http://rars.sourceforge.net/> [Accessed 24 June 2009].
- SPRONCK., 2003. Online Adaptation of Game Opponent AI in Simulation and IN Practice. In: *Proceedings of the 4th International Conference on Intelligent Games and Simulation*.
- STANLEY K, O., KOHL N., SHERONY R., MIIKKULAINEN R., 2005. Neuroevolution of an automobile crash warning system. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2005)*, 2005.
- TOGELIUS J., LUCAS S, M., 2006. Evolving robust and specialized car racing skills. In: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*.
- VIEIRA., 2006. Um controle autônomo de um carro de corrida. In: *Brazilian Symposium on Computer Games and Digital Entertainment*.
- WLOCH K, O., BENTLEY P, J., 2004. Optimising the performance of a formula one car using a genetic algorithm. In: *Proceedings of Eighth International Conference on Parallel Problem Solving From Nature*, 2004, pp. 702–711.
- YANNAKAKIS, J. N., 2005. A thesis submitted to University of Edinburgh in the subject of Artificial Intelligence for the degree of Doctor of Philosophy. In: *Computer Games: Generating Interesting Interactive Opponents by the use of Evolutionary Computation*.