

USANDO PROGRAMAÇÃO GENÉTICA PARA EVOLUIR AGENTES JOGADORES DE FUTEBOL DE ROBÔS

Luiz Carlos Maia Jr.¹, Reinaldo A. C. Bianchi

Departamento de Engenharia Elétrica
Faculdade de Engenharia Industrial - FEI
Av. Humberto de A. C. Branco, 3972
09850-901 São Bernardo do Campo – SP
rbianchi@cci.fei.br

RESUMO

Este artigo descreve o projeto e a implementação de agentes capazes de atuar no domínio do Futebol de Robôs, segundo uma abordagem baseada em Programação Genética. Para tanto, inicialmente é apresentado o domínio do Futebol de Robôs e, a seguir, são revistos conceitos básicos de Programação Genética (PG). Posteriormente, são descritos os agentes implementados em um ambiente simulado: primeiro, agentes capazes de seguir paredes - problema tradicional da área de Programação Genética. A seguir são descritos dois tipos de agentes capazes de seguir uma bola, um com o sistema de visão global, onde uma câmera é colocada sobre o campo de jogo, permitindo ao agente ter uma percepção completa do seu ambiente e o outro com um sistema de visão local, onde cada robô possui uma câmera embarcada, resultando em uma visão parcial da área de jogo. Finalmente, são apresentados resultados por meio de gráficos de evolução, exemplos de caminhos percorridos e exemplos de indivíduos. Esses resultados permitem concluir que a complexidade dos agentes com visão local sofreu um grande aumento em relação a visão global, mas que a performance dos dois agentes foi satisfatória.

ABSTRACT

This paper describes the design and the implementation of agents capable of acting in the Robotic Soccer domain, based on a Genetic Programming approach. To achieve this objective initially is presented the domain of Robotic Soccer and are reviewed basic concepts of Genetic Programming. Next are described the agents implemented in a simulated environment: first, agents capable of following walls – a traditional problem in the field of Genetic Programming. Two types of agents capable of chasing a ball are then presented: first one with a global vision system and later one with a local vision system. In a global vision system, a camera is placed over the game field, allowing the agent to have a complete perception of its environment and in a local vision each robot possesses an embarked camera, resulting in a partial vision of the game area. Finally the results are presented by means of evolution graphics, courses examples and individuals examples.

¹ Bolsista Fundação de Ciências Aplicadas – FCA/FEI

1. INTRODUÇÃO

Partidas de futebol entre robôs, além de serem extremamente motivantes para possibilitar o surgimento de um espírito de ciência e tecnologia nos futuros engenheiros, constituem uma atividade que possibilita a realização de experimentos reais para o desenvolvimento e testes de robôs que apresentam comportamento inteligente e que cooperam entre si para a execução de uma tarefa, formando um time.

Entre os desafios e problemas a serem estudados, Shen (1998) afirma que “Jogadores Robóticos precisam realizar processos de reconhecimento visual em tempo real, navegar em um espaço dinâmico, rastrear objetos em movimento, colaborar com outros robôs e ter controle para acertar a bola na direção correta”. Para atingir esses objetivos os robôs devem ser autônomos, eficientes, cooperativos, com capacidades de planejamento, raciocínio e aprendizado, além de atuarem sob as restrições de tempo real. (Veloso et al., 1998 e Tambe, 1998).

Este trabalho tem por objetivo a evolução de uma família de agentes robóticos que seja capaz de seguir uma bola em um campo simulado e que possa, futuramente, ser utilizada na construção de um time de jogadores de futebol de robôs para a liga F180 (de pequenos robôs) da RoboCup.

Para estudar a evolução de agentes robóticos, foi criado um ambiente de simulação para o ambiente Linux onde os robôs se movimentam livremente, respeitando apenas os limites das paredes e de obstáculos. Em seguida foi criada uma família de robôs que, inicialmente, teve o objetivo de seguir paredes – um problema tradicional da área de Programação Genética – e que foi aperfeiçoada para seguir uma bola utilizando um sistema de visão global– onde uma câmera é colocada sobre o campo de jogo permitindo ao agente ter uma visão completa do seu ambiente (Maia & Bianchi, 2000) – e que finalmente se transformou em uma família de agentes dotados de sistema de visão local – onde cada robô possui uma câmera embarcada resultando em uma visão parcial da área de jogo. (A figura 1 mostra imagens adquiridas destas duas maneiras.)

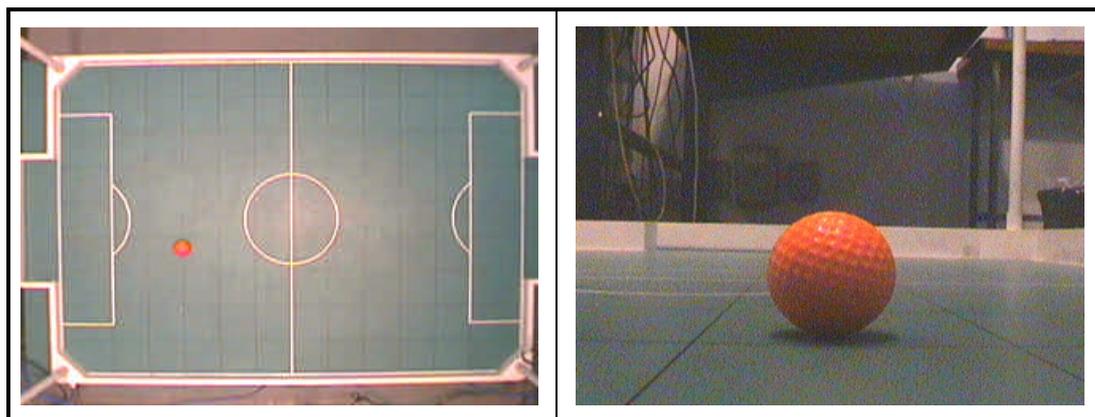


Figura 1 – Imagens de sistemas de visão Global x Local

Na próxima seção descrito o Domínio do Futebol de Robôs. A seção 3 apresenta de forma sucinta características básicas da Programação Genética. A seção 4 descreve o agente Seguidor de Paredes, a seção 5 o agente seguidor de bola com visão global e a seção 6 o agente com visão local. Os resultados são apresentados na seção 7, que também compara as influências dos dois tipos de sistema de visão computacional na evolução de agentes jogadores de futebol robótico. A última seção apresenta as conclusões deste trabalho.

2. DOMÍNIO DO FUTEBOL DE ROBÔS

Os livros textos modernos de Inteligência Artificial, como RUSSELL e NORVIG (1995), começam a apresentar esta disciplina a partir de uma visão integrada. Isto não só permite que se trate os problemas desta área de pesquisa a partir de diversas abordagens, como é o resultado da compreensão, por parte dos pesquisadores, que IA não deve ser vista como segmentada.

Seguindo esta tendência, os domínios de aplicação pesquisados também começaram a mudar. Na área dos jogos, clássica em IA, criar programas eficientes para jogos de tabuleiro deixou de ser um objetivo distante: no xadrez os computadores já conseguem vencer os campeões humanos. Novos domínios fizeram-se necessários.

O futebol de robôs foi proposto por diversos pesquisadores ([KITANO, 1997], [KIM, 1998], [SANDERSON, 1997]) para criar para IA um novo desafio a longo prazo. O desenvolvimento de times de robôs envolve muito mais que integração de técnicas de IA. Segundo KRAETZCHMAR et al. (1998), “Dispositivos mecatrônicos, hardware especializado para o controle de sensores e atuadores, teoria de controle, interpretação e fusão sensorial, redes neuronais, computação evolutiva, visão, e Sistemas Multi-Agentes são exemplos de campos envolvidos nesse desafio”. A figura abaixo apresenta o time de futebol de robôs da *Carnegie Mellon University* que venceu a copa mundial de 1997.

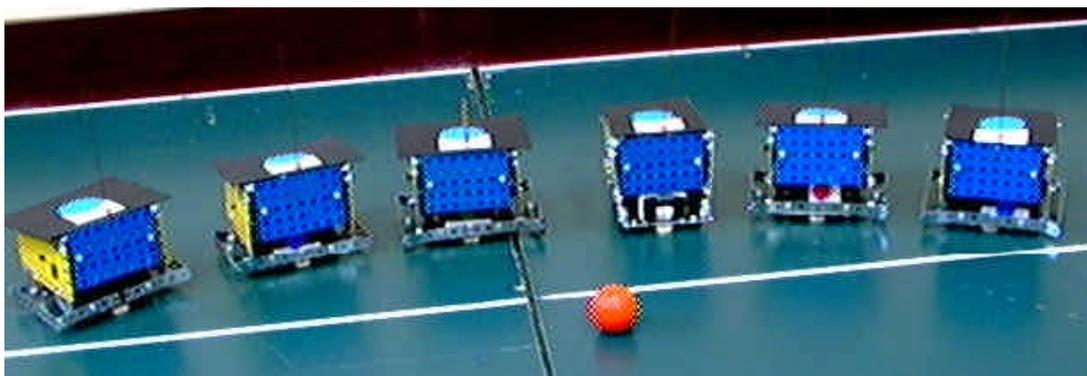


Figura 2 - O time CMUnited-97 vencedor da copa de 1997. [VELOSO, M.; STONE, P; HAN, K., 1998]

A organização chamada *Federation of International Robot-soccer Association* (FIRA) é responsável pelo estabelecimento e controle das diversas copas emergentes de futebol de robôs. Existem diversas modalidades de jogos entre robôs, com variações desde o número e o tamanho dos robôs até a capacidade computacional de cada robô. As principais modalidades de campeonatos existentes são:

- MIROSOT - *Micro Robot World Cup Soccer Tournament*: é a considerada a de mais fácil implementação (tem os robôs menores e de mais baixo custo). Fisicamente a plataforma deste projeto é constituída por um campo para o jogo, plano, com as dimensões de 150 x 90 cm (similar a de futebol de botão) e, para cada time, uma câmara de vídeo CCD e respectivo sistema de aquisição de imagens, um computador, sistema de transmissão de dados e os 3 robôs. Estes têm as dimensões máximas de 7.5x7.5x7.5 centímetros e são compostos por 2 motores controlados por um microprocessador, bateria própria e sistema de comunicação sem fio. A figura 3 apresenta o esquema básico do sistema. A bola utilizada é padronizada: uma bola de golfe laranja.

- *RoboCup Small League*: muito similar à Mirobot, tem times de 5 robôs de até 180 cm² e uma mesa similar a de tênis de mesa;
- *RoboCup Medium e Full Size*: nestas modalidades não é permitido o uso de um sistema de visão global único nem de um computador central. Cada robô deve ter sua própria câmera – visão local – e computador e trocam as mensagens via rádio. Nestas duas últimas modalidades é que se encontram as maiores contribuições até o momento.
- Liga de simulação *RoboCup*: modalidade simulada onde é utilizado um sistema cliente-servidor, onde o servidor gera um campo virtual e controla as ações dos jogadores e um cliente comanda um jogador no campo e controla suas ações. As comunicações entre o cliente e o servidor são feitas por UDP/IP, permitindo que os clientes sejam escritos em qualquer sistema de programação que tenha interface UDP/IP. O Servidor de Futebol Multiplataforma pode ser usado em sistemas como SunOS 4.1.x, Solaris 2, DEC OSF/1, NEWS-OS, Linux and Irix 5.

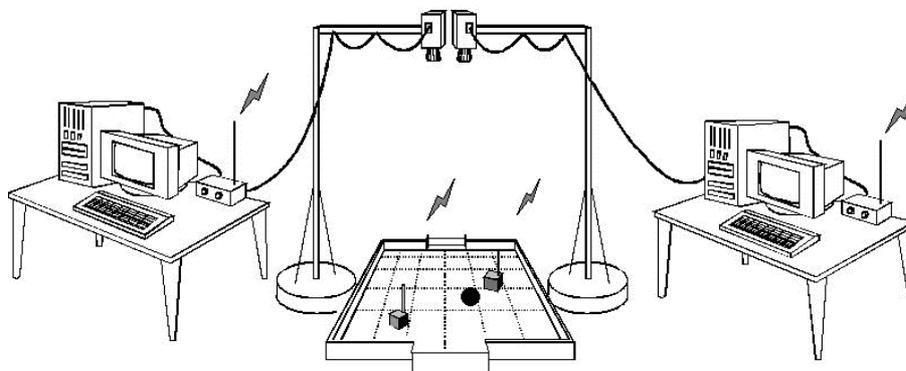


Figura 3 - O esquema básico de um jogo com apresentado em [FIRA, 1998].

O funcionamento de um time que utiliza visão global segue uma fórmula básica: realiza a aquisição da imagem por meio de uma única câmera localizada sobre o campo; processa a imagem usando técnicas de Visão Computacional para descobrir a posição de todos os robôs e da bola. Com esta imagem, um sistema de decisão define a melhor tática a ser aplicada e os movimentos instantâneos de cada robô. Já um time que utiliza visão local deve extrair dados da imagem localmente, para depois definir a tática de cada jogador. Este processo é repetido com a decisão tomada mudando conforme a configuração encontrada na imagem, em ciclos de 60 vezes por segundo.

Já existem algumas competições anuais, aliadas a importantes eventos das áreas de Inteligência Artificial e Robótica, onde são observados os desenvolvimentos neste tipo de aplicação. A cada 4 anos, as copas de robôs são realizadas juntamente com as copas mundiais de futebol, no país sede. Assim, em 1998, juntamente com a Copa Mundial de Futebol, da FIFA, foi realizada a Copa Mundial de Robôs, da FIRA, na França, patrocinada, entre outros, pela própria FIFA. No Brasil já foram realizadas duas copas nacionais (na Escola Politécnica da USP em 1998 e na UFRGS em 1999), contando com a participação de diversos times de universidades e institutos de pesquisas brasileiros.

A construção física dos robôs envolvem diversos aspectos importantes, como o estudo de dispositivos mecatrônicos, de hardware especializado para o controle de sensores e atuadores e teoria de controle.

O sistema robótico deve possuir capacidades visuais para perceber o mundo: reconhecer e localizar os robôs no jogo e a bola, descobrindo os movimentos na cena. A construção do

sistema sensorial envolve: visão computacional, interpretação e fusão sensorial, redes neurais e computação evolutiva.



Figura 4 – Jogada de falta na 1ª Copa Brasil de Futebol de Robôs (POLI x UFMG).

Uma das principais características de um sistema para o futebol de robôs não é sua habilidade para raciocinar, mas sua habilidade para escolher ações de maneira rápida e efetivas, sendo capaz de atuar em um ambiente incerto e variável. Esse domínio pode ser caracterizado como o de uma tarefa de planejamento reativo complexa, que envolve diversos aspectos a serem estudados: a geração e execução de planos complexos para resolver objetivos específicos, a alocação de recursos *on-line*, a capacidade de lidar com problemas que surgem em tempo real e a capacidade de raciocinar com informações incompletas (nem sempre o sistema sensorial consegue detectar todos os movimentos) e eventos imprevisíveis (um toque na bola pode sair como previsto ou não: envolve aspectos físicos como atrito e aderência).

Por ser um sistema com múltiplos robôs executando uma tarefa em cooperação, este domínio é ideal para o estudo de aspectos fundamentais dos Sistemas Multi-Agentes: a modelagem distribuída do ambiente (ou como cada agente representa o conhecimento que possui do problema), a organização das sociedades e a comunicação entre os agentes, além de arquiteturas de controle.

Finalmente VELOSO (1998) afirma que “o Futebol de Robôs é excitante devido a sua natureza de sensores/atuadores em tempo real. E isso pode ser atingido usando a mais simples das arquiteturas até as mais complexas.” Isto permite que equipes com as mais diversas abordagens possam competir e medir os resultados de suas abordagens de uma maneira objetiva: o placar e a classificação no campeonato. Neste domínio a natureza das restrições de tempo real ficam óbvias: um sistema deve ser rápido o suficiente pois se ele levar alguns segundos entre a percepção e a ação ele perde para o adversário.

O domínio do Futebol de Robôs tem adquirido importância cada vez maior na área de Inteligência Artificial pois possui todas as características encontradas em outros problemas reais complexos, desde sistemas de automação robóticos, que podem ser vistos como um grupo de robôs realizando uma tarefa de montagem, até missões espaciais com múltiplos robôs [TAMBE, 1997].

3. PROGRAMAÇÃO GENÉTICA (PG)

De acordo com a Teoria da Evolução de Darwin, os organismos mais adaptados ao mundo em que vivem são os que terão maiores chances de sobrevivência. Os organismos que existem hoje são consequência da evolução de outros organismos inferiores que se extinguiram e que sem eles, provavelmente, não existiriam. Cada criatura nessa cadeia é o produto de uma série de “acidentes” que têm acontecido continuamente sobre a pressão seletiva do ambiente.

Programação Genética [KOZA, 1992] é uma forma de computação baseada na simulação da evolução de indivíduos. Nela é aproveitado o fato de que a evolução na natureza é criativa, pois produz muitas vezes resultados inesperados, impensáveis e não-lineares, diferente do modo de programação usual. Algumas características importantes da Programação Genética são descritas a seguir.

Em PG o indivíduo que deve ser evoluído é um programa de computador definido como uma árvore de funções e terminais, que constituem suas características e define o comportamento no ambiente para o qual foi desenvolvido. Cada função é um ramo da árvore e cada terminal uma folha do indivíduo. Funções podem ser condições, sensores, operações aritméticas ou lógicas etc. e são usadas para captar informações sobre o ambiente e, convenientemente, para melhorar o desempenho de cada indivíduo. Já os terminais geralmente representam as ações que os robôs podem executar, como andar para frente, virar etc. Uma população contém um determinado número de indivíduos que é renovada a cada iteração, criando uma nova geração.

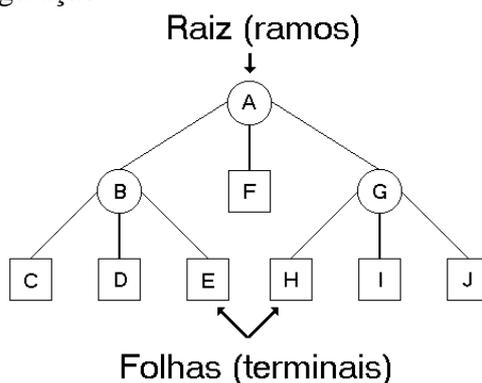


Figura 5 – Exemplo de Indivíduo

Fitness é um parâmetro que mede o “encaixe” da espécie em seu ambiente, sendo usado para descartar, a cada iteração, os piores indivíduos. Existem várias definições para *fitness*, sendo aqui utilizada a seguinte: “Quanto maior o *fitness* do indivíduo maior sua probabilidade de sobrevivência e adaptação ao ambiente”. Com este dado calculado, os indivíduos podem ser organizados de forma descendente, sendo então eliminados os piores e mantidos os melhores para serem reproduzidos ou cruzados.

A criação de uma nova geração pode ser feita de três maneiras distintas: a Reprodução, onde os melhores indivíduos da geração atual são simplesmente copiados para a nova geração, com o objetivo de não se descartar os melhores indivíduos; o Cruzamento, onde são escolhidos dois indivíduos (pais) bem adaptados, dos quais são criados dois descendentes que vão estar na próxima geração – o objetivo é um novo indivíduo seja criado usando o que há de melhor de uma geração, com alguma variação; finalmente, na Mutação um indivíduo é selecionado, depois um ponto de mutação, e a partir dele é criado um novo ramo – o objetivo é melhorar um indivíduo, mas com isso ele pode perder desempenho e ser eliminado. A figura 6 mostra um exemplo de cada uma destas maneiras descritas.

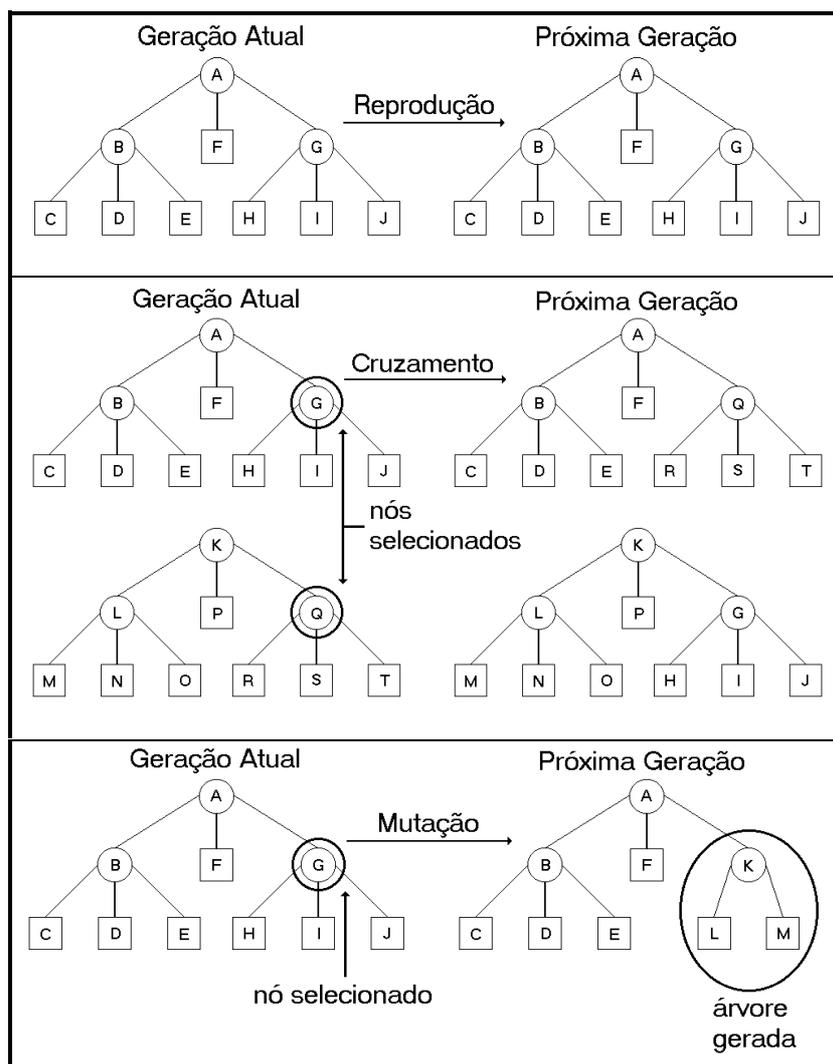


Figura 6 – Exemplo de Reprodução, Cruzamento e Mutação.

Usando os conceitos de Programação Genética, definiram-se funções e terminais, com os quais os indivíduos foram testados no ambiente e o seu fitness medido para determinar os que melhor executaram a tarefa designada. A partir dos melhores criou-se uma nova geração utilizando a reprodução e o cruzamento, essa nova geração foi então testada e a partir dela novas gerações foram criadas. Ao final do processo obteve-se indivíduos capazes de realizarem a tarefa destinada. A seguir é apresentado o agente mais básico que foi evoluído, o Seguidor de Paredes.

4. AGENTE SEGUIDOR DE PAREDES

A implementação de um agente Seguidor de Paredes é necessária por se tratar de um comportamento básico de movimentação de agentes inteligentes no domínio do Futebol de Robôs e por ser possível o aproveitamento das características desenvolvidas nesse agente para criar um time de futebol.

O objetivo principal desse comportamento é desenvolver um indivíduo que seja capaz de percorrer toda a lateral do ambiente de simulação no tempo estipulado.

Com as características da PG foi criado o programa do robô seguidor de paredes no qual o robô deve visitar o maior número de quadrados adjacentes à parede dentro de um ambiente de

simulação em um número máximo de passos. É considerado um passo cada vez que o robô executar um terminal (agir), e é considerada uma execução cada vez que a árvore completa do robô for percorrida uma vez, ou seja, existem vários passos dentro de uma execução e opta-se por não pará-la antes que termine, por isso têm-se robôs com mais passos, mas não com menos.

O indivíduo deve percorrer a sala sempre próximo à parede e completar sua volta em no máximo '200 passos'.

A cada simulação o indivíduo começa em posições e direções diferentes para evitar que ele percorra o caminho automaticamente sem verificar o ambiente em que está.

O *fitness* é calculado da seguinte forma:

$$\text{fitness} = \text{fit} - (\text{unfit} / 500)$$

Na qual 'fit' é o número de posições do caminho ideal que o robô visitou e 'unfit' é o número de vezes em que ele se desviou do caminho (penalização) ideal ou visitou uma célula mais de uma vez, por esse número ser "relativamente" grande ele é dividido por '500' para diminuir sua influência no *fitness* final. O *fitness* máximo é '780' e representa o número de células do caminho ideal, ou seja, se um robô conseguir percorrer o caminho ideal sem se desviar muito da sua trajetória, certamente terá um *fitness* próximo ao máximo.

As funções que permitem o robô desviar de paredes e as ações de movimentação (andar para frente, virar) são definidas como:

- PROGN3(3): executa três ramos em seqüência;
- PROGN2 (2): executa dois ramos em seqüência;
- IFWALL (I): executa seu ramo esquerdo se não for detectada parede pelo robô (a no máximo 1 passo de distância) e caso contrário executa seu ramo direito.

Os terminais (ações) definidos são:

- WALKFRONT (F): faz robô dar um passo à frente;
- WALKBACK (B): faz robô dar um passo para trás;
- RIGHT (R): faz robô virar à direita;
- LEFT (L): faz robô virar à esquerda.

Outros parâmetros usados na simulação incluem: População (M) = 500; Probabilidade de Cruzamento = 70%; Probabilidade de Reprodução = 30%; Matriz de simulação = 200 x 200, com o posicionamento do robô em números reais; Limite de complexidade (no sorteio) = 1000; Ângulo de virada = 5° (permite bastante precisão na movimentação buscando a bola); Número de execuções por indivíduo = 2. É importante que seja dito que o programa permite que o robô se vire em qualquer ângulo que seja definido, o ângulo de 30° foi adotado depois de executadas várias simulações.

5. AGENTE SEGUIDOR DE BOLA COM VISÃO GLOBAL

O agente Seguidor de Bola foi escolhido para ser o segundo passo da evolução dos agentes jogadores de futebol, pois é similar ao clássico Seguidor de Paredes, adicionando elementos simples como a bola e obstáculos. Por possuir visão global, qualquer indivíduo desta família sabe a cada momento onde se encontra a bola e ele próprio.

O objetivo dos indivíduos na simulação é alcançar a bola e tocá-la o maior número de vezes em um número determinado de passos. É considerado um passo cada vez que o robô executar um terminal (agir) e o indivíduo deve percorrer a sala buscando pela bola em no máximo 2000 passos. Esse número de passos foi definido para tornar a representação final mais clara, pois o excesso de caminhos em uma figura poderia torná-la ilegível. A cada simulação a posição da bola e a posição e orientação do agente são iniciadas aleatoriamente.

O *fitness* foi calculado através da seguinte fórmula:

$$\text{fitness} = \text{hits} * k - \sum(n / D_{ini})$$

Onde *hits* é o número de vezes que o agente atingiu a bola; *n* é o número de passos do agente e *D_{ini}* é a distância inicial medida antes do agente se movimentar. O valor *n/D_{ini}* (chamado de penalização) é recalculado a cada vez que o robô atinge a bola, por isso é usada a somatória desses valores no *fitness*; 'k' é uma constante a ser definida em cada caso e cuja variação é analisada nos resultados. O número de *hits* máximo é estimado em 15 com base na área de movimentação, no número de execuções e na distância máxima que a bola pode estar do robô.

Além das funções de desvio de paredes e das ações de movimentação (andar para frente, virar), que são similares a do agente seguidor de paredes, adicionou-se apenas uma ação que relaciona a movimentação do robô com a movimentação da bola. As funções definidas são:

- PROGN3(3): executa três ramos em seqüência;
- PROGN2 (2): executa dois ramos em seqüência;
- IFWALL (I): executa seu ramo esquerdo se não for detectada parede pelo robô (a no máximo 1 passo de distância) e caso contrário executa seu ramo direito.

Os terminais (ações) definidos são:

- WALKFRONT (F): faz robô dar um passo à frente;
- WALKBACK (B): faz robô dar um passo para trás;
- RIGHT (R): faz robô virar à direita;
- LEFT (L): faz robô virar à esquerda;
- ALIGN (A): direciona o robô para a bola.

Outros parâmetros usados na simulação incluem: População (M) = 500; Probabilidade de Cruzamento = 70%; Probabilidade de Reprodução = 30%; Matriz de simulação = 200 x 200, com o posicionamento do robô em números reais; Limite de complexidade (no sorteio) = 1000; Ângulo de virada = 5° (permite bastante precisão na movimentação buscando a bola); Número de execuções por indivíduo = 1.

6. AGENTE SEGUIDOR DE BOLA COM VISÃO LOCAL

Esta segunda família de agentes tem por objetivo, como no caso anterior, alcançar a bola e tocá-la o maior número de vezes possíveis dentro de seu tempo de execução. Os indivíduos criados nesta família são mais complexos, pois além dos obstáculos e da bola, o agente possui uma visão limitada e não sabe as condições de todo o ambiente (como acontecia na visão global). Sendo assim, deve estar preparado para uma situação na qual não vai enxergar a bola e terá que encontrá-la antes de poder tocá-la. O agente precisa buscar a bola da maneira mais rápida e desviando dos obstáculos que atrapalham sua visão.

Para simular a visão local de uma câmera sobre um robô móvel, foi adicionada ao sistema uma função que fornece ao agente a informação sobre a visibilidade da bola. Essa função retorna verdadeira se a bola estiver em frente ao agente (com um ângulo de visão de 30° para cada lado) e falsa caso contrário. A figura 7 mostra o campo visual de um agente com visão local.

O *fitness* nesta simulação foi calculado da mesma forma que na simulação anterior, visto que o objetivo da evolução é o mesmo (tocar a bola), apenas adicionando dificuldade na localização da bola.

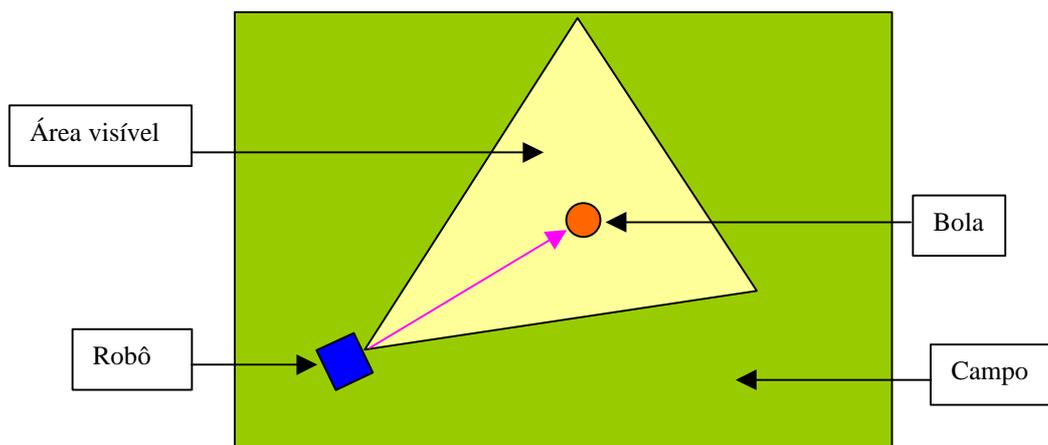


Figura 7 – Campo visual de um agente com visão local

As funções e ações definidas para esta simulação são as mesmas do agente com visão global, apenas com a adição da função IFBALL:

- IFBALL (C): executa o ramo esquerdo se enxergar bola e caso contrário executa o ramo direito.

Os terminais definidos são os mesmos do agente com visão global, apenas com uma diferença na função ALIGN, que só direciona o robô para a bola se ela estiver dentro do campo de visão do agente.

Os outros parâmetros da simulação (população, probabilidades, etc.) são exatamente iguais ao da visão global. Os agentes também são iniciados em posições aleatórias e o ambiente de simulação (paredes e obstáculos) continua igual.

7. RESULTADOS

Esta seção apresenta e compara três tipos de resultados para os três agentes: os gráficos evolutivos, exemplos de caminhos percorridos e os melhores indivíduos gerados.

7.1 - Gráficos Evolutivos

O primeiro gráfico apresentado é do fitness do agente seguidor de paredes em função da geração, que mostra a evolução dos indivíduos após '51' gerações (a geração '0' também é contada). Pode se observar como a média acompanha a reta dos maiores após as gerações e como é mais estável não apresentando variações bruscas. Por exemplo: na 'geração 9' o maior indivíduo possui fitness igual à '400' e na 'geração 11' o fitness é igual à '530' o que representa um grande salto, isso devido ao cruzamento.

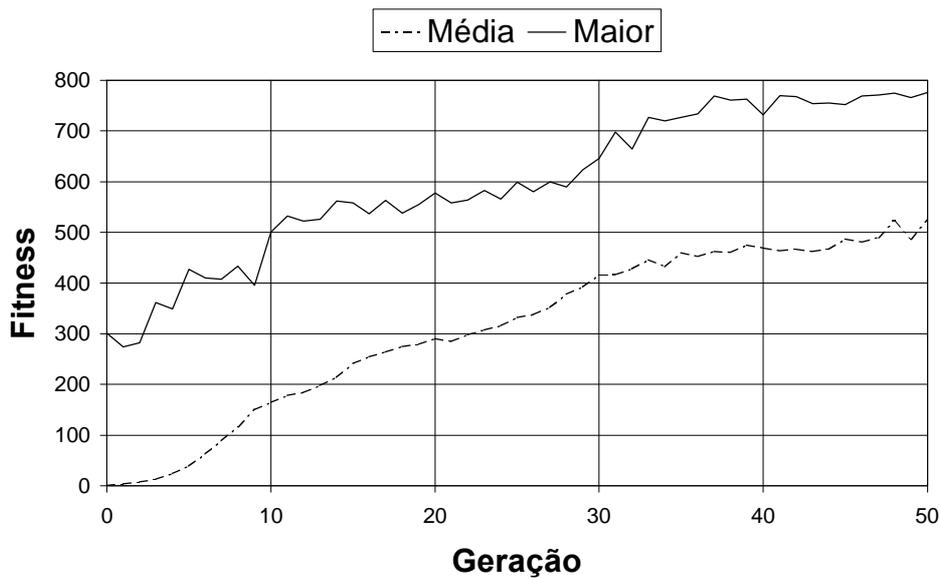


Figura 8 – Gráfico evolutivo para agentes seguidores de parede (51 gerações)

No próximo gráfico, em uma simulação de ‘42’ gerações do seguidor de paredes, nota-se claramente uma queda na média do fitness. Isso ocorre devido a uma tentativa de renovação dos indivíduos executada no meio do processo de evolução, essa renovação é chamada de “decimação”. Na decimação apenas ‘10%’ deles são aproveitados e os outros ‘90%’ são criados novamente. Apesar dessa tentativa e do fitness inicial ser próximo à ‘500’ no final da simulação não se conseguiu indivíduos tão bons como os da execução anterior. Isso demonstra como a PG é muito dependente dos seus primeiros indivíduos, mas também da variedade existente.

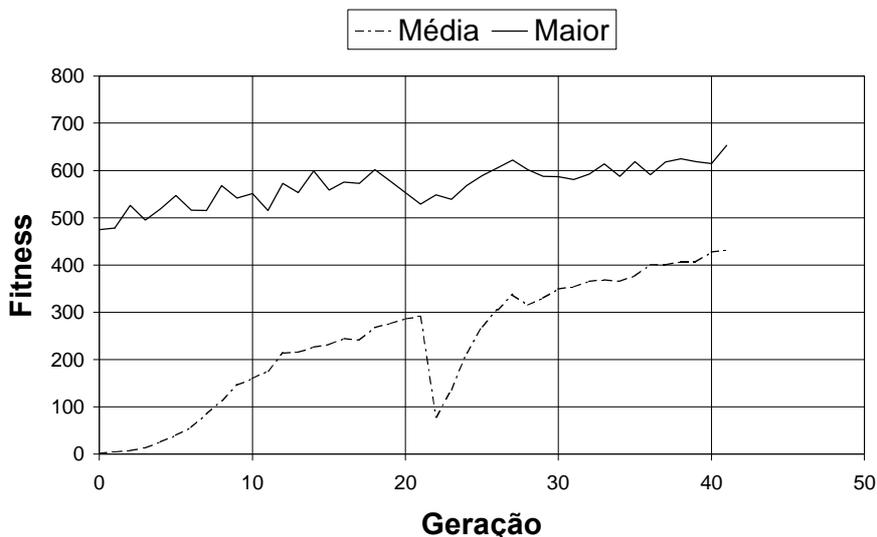


Figura 9 – Gráfico evolutivo para agentes seguidores de parede (42 gerações com decimação na 21ª geração).

O gráfico abaixo (figura 10) apresenta a evolução dos agentes com visão global, por meio de quatro seqüências. As duas mais altas mostram o resultado do *fitness* do melhor agente de cada geração, com a evolução usando $k = 1500$ (a mais alta) e $k = 1000$ (a segunda mais alta).

As duas seqüências mais baixas apresentam o *fitness* médio para cada geração, com $k = 1500$ (a terceira) e $k = 1000$ (a mais baixa).

A primeira observação que se pode fazer é que ocorre uma grande variação do *fitness* do melhor indivíduo (que não acontece em casos simples como o seguidor de paredes). Isso se deve ao fato do agente interagir com outro objeto que também se movimenta, que é mais rápido do que ele e que reflete em outros objetos.

Podemos comparar também os resultados com $k = 1500$ e $k = 1000$. O valor $k = 1000$ fez com que a somatória das penalizações dos indivíduos tivesse maior significância no *fitness* e com isso selecionou melhor aqueles que deveriam continuar, ocasionando uma variação menor nos melhores indivíduos. Já a utilização de $k = 1500$ aumentou a velocidade da evolução dos agentes (tanto os melhores quanto a média).

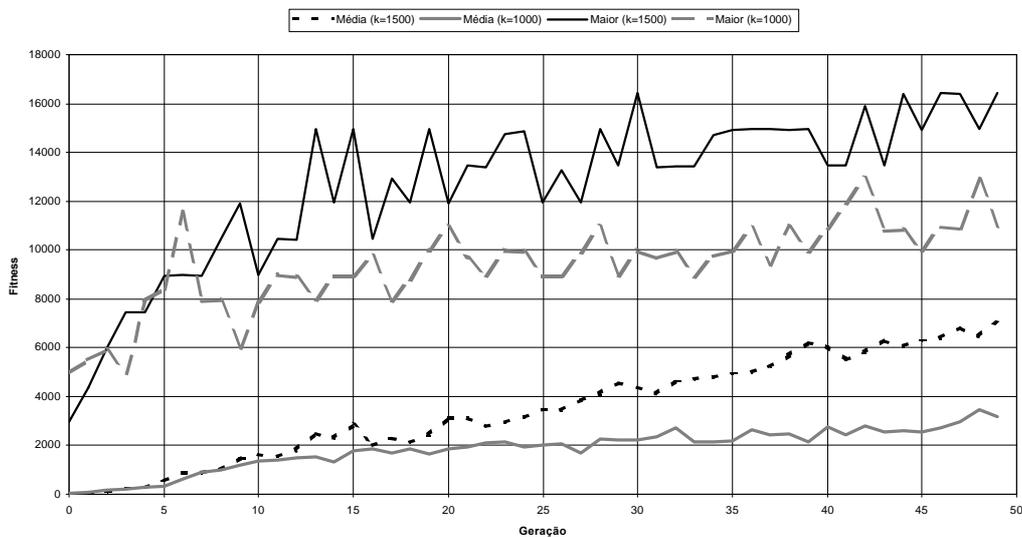


Figura 10 – Gráfico evolutivo para agentes com visão global (50 gerações)

O último gráfico (figura 11) apresenta a evolução dos agentes com visão local, por meio de duas seqüências. A mais alta mostra o resultado do *fitness* do melhor agente de cada geração, com a evolução usando $k = 1500$. A mais baixa apresenta o *fitness* médio para cada geração também com $k = 1500$.

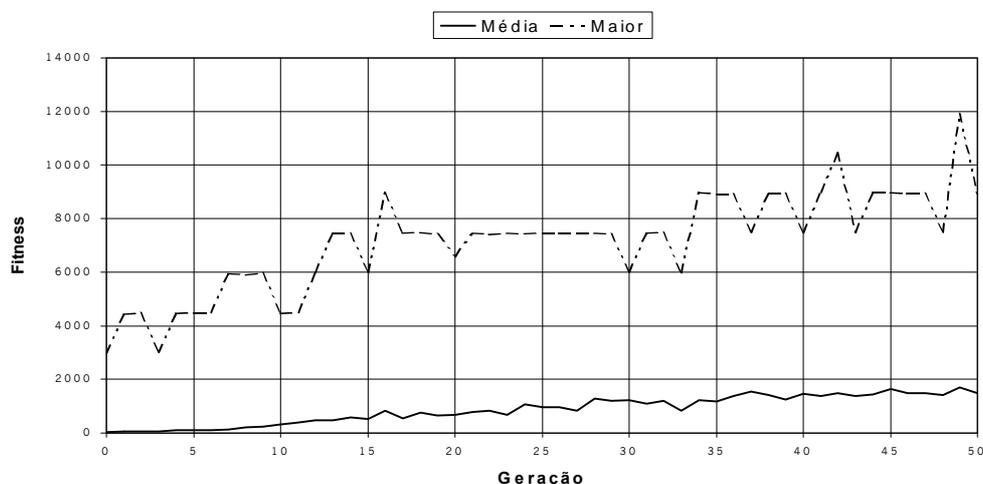


Figura 11– Gráfico evolutivo para agentes com visão local (50 gerações)

Neste gráfico, como já havia sido notado na simulação anterior, também existe uma grande variação no desempenho do melhor agente. Mais uma vez isso se deve à grande gama de possibilidades em que o ambiente pode se apresentar para o agente, antes que ele comece a buscar pela bola.

Um ponto marcante que pode ser percebido nessas simulações e que as diferenciam de outros sistemas usados geralmente em sistemas de PG é o baixo valor da média dos indivíduos, que nem chega a alcançar o desempenho do melhor indivíduo da primeira geração. Isso se deve principalmente ao alto grau de dificuldade dessa simulação. Mas como se trata de uma evolução espelhada na natureza, o que se espera é que se esses mesmos agentes fossem deixados evoluindo durante mais tempo, os melhores deveriam estabilizar-se em um valor próximo ao ótimo e a média não deveria ficar muito longe disso.

7.2 – Exemplos de Caminhos

A figura 12 apresenta os caminhos extraídos das simulações dos agentes seguidores de paredes. Vale ressaltar que a escolha dos exemplos de caminhos apresentados priorizou aqueles que exemplifiquem o desenvolvimento e sejam de fácil entendimento.

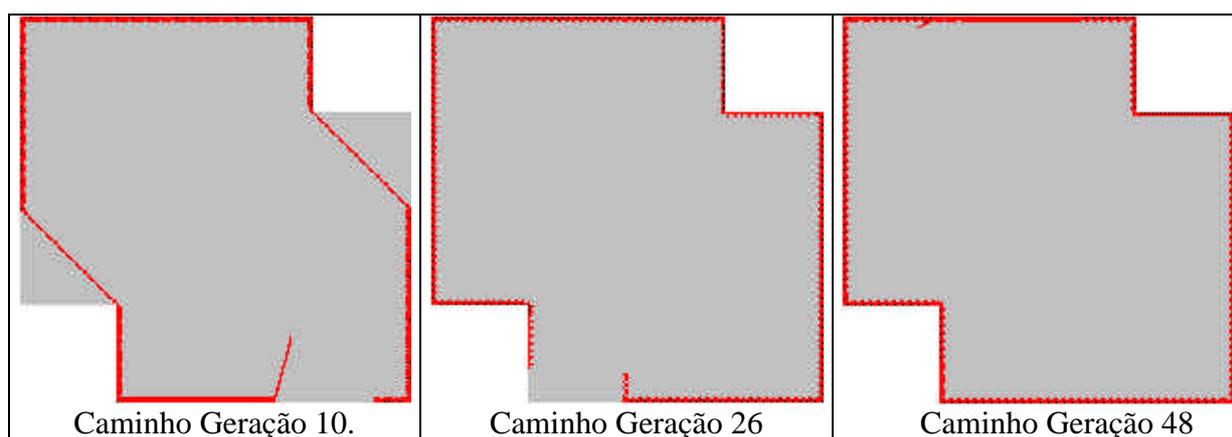


Figura 12 – Caminhos para agentes seguidores de paredes.

Analisando as figuras pode-se notar que na décima geração o indivíduo pode acompanhar a parede e detectar o obstáculo à sua frente, mas não percebe quinas, apenas caminha até encontrar algo que impeça seu movimento. Isso faz com que ele tenha algumas características que podem ser aproveitadas na próxima geração. Na geração número 26 o indivíduo percorre muito bem o caminho, mas por virar-se muitas vezes para encontrar quinas acaba não percorrendo o caminho todo. Ele demonstra o aperfeiçoamento que ocorreu desde a décima geração na detecção de quinas. Finalmente, na geração número 48 o indivíduo tem um resultado excelente, pois percorre todo o caminho dentro do número de movimentações estipulado, conseguindo detectar as quinas que foram problema na 10ª geração sem a perda de tempo que isto acarretava na 26ª geração.

A figura 13 apresenta caminhos extraídos da simulação de agentes com visão global, com k valendo 1500. Os quadrados brancos são obstáculos para o indivíduo e para a bola. O caminho feito pelo agente é representado pela cor vermelha (com o R amarelo marcando o início do caminho) e a cor verde mostra o caminho da bola (B no início).

Na figura 15 é mostrado o melhor indivíduo seguidor de paredes, a figura 16 o melhor seguidor de bolas com visão global enquanto a figura 17 apresenta um dos melhores indivíduos com visão local. Como o indivíduo é uma árvore, abaixo é apresentada uma travessia desta árvore que também é usada para guardar os agentes ao término da evolução. Cada letra representa uma função ou um terminal da árvore: PROG3 = 3; PROG2 = 2; IFWALL = I; WALKFRONT = F; WALKBACK = B; RIGHT = R; LEFT = L; ALIGN = A.

```
3IR2222FIF3B2II2FII2B2LI2B2B32F2LRR122BL3IBIFIFBFB323L3B3BFB2F32RFF23IIB3FLIFRLR2L
3IILLILFILLRIF23B2II32RLIFRBB2RIIBB33IR223I3IILL2I2I33BBRBRBIRRL3IB32LRBRRBI2BRLRF
R2L3222FRIILLIFFRIFLRIF3BIBRLRFFR2FRRL3IILBILFIFLRR3IB2I32LIRR22ILRBRILBBBBLFLFB
FBLR2IBRRRRRIILLIFFRFI33FLIFR2LIRBB3ILLFIR3BBR
```

Figura 15 – Exemplo de agente seguidor de paredes

```
3IIR2FF323I2ILIB32R3FRRR22L3RAF2BRFR323I2ILIBILLIAAIR32F2IF3FFRFFF33RB2RRF33FFRILI
23I2IRIB32R3FR32AR32F2IIAL2FRFAF22IALF2BRR22L3RF2IBF323IFIRR33RBII3FBBIRAFF33FF22
F2IIAL3F3RRFRFIRAI232FRR2B2BR33FBIRF3L2RF3FFR2RBA2BIR32FRFBAIRFII333F2LFFFBAI2
IBIIL3FF3FFR3FFRAF2BRFR323I2ILIB33BIAR22FF3A2IIAL3FFRLFF2FRIAIR32F2IF3FFRRFF3
3RF2RRF33FFRILR2BIR32FR3FFRBFLLI3FLBIRAF33FFRL32FRR2FIBFAIRF32RIIFRIF33AIL3A2FF
FFLFR2BIR32FR2FRBAIRFII3FBBIRFF33FFRL3AR2FIBFAIRFII3L33FA3AILIFRFFFIRAF2LRF
```

Figura 16 – Exemplo de agente seguidor de bolas com visão global

```
3F33FCA32LLACI23FRBFBIIFL3I232FIICAF22BIIRRLBR3CA23FLI2F2BLIBIL2IILR2BAF3I232FIBR
3CA23FLIICA32LL2I3B2IBCARR2LFRBCAIIFL3I232FIICAF22BIIR3RCA3R3ICARIR2BIBCAIRFI23F
CACA3FBBRRFALRLBR3CA23FLI2F2BLIBIL2IILR2BCARFIL22LBIL223CAIFB3IRBFBLL3FB2I33
LILRBFL2IIR2CA3R2FIICBB3LL2LBCAR32B2LLLFCAR3FRLLRF223L2FRRCABLCAICAFFFBILCA
2I3L3FFR3B33FLFBLBLBRLIRIL2IILR2BCARFIL22L22IFB23FCALB2LRR2F3RIF2RLI323BRRFL23
CACACAF2BI3F2RBB3BRFLLI3LCA3FIR3233FFF3B2ILRFBCA2B322LRFLRLACAR3BBBCA3R333I
LRLFBBIIR3IILB23BFFBLLRIR23CAFBB3FB2ICA2FFCALRF223L2FRRBLCAICAFFFBILCA2I3L3
FFR3BRBLLRLF22L3IB3CAFLLBFLFBILCA2I3L3FF2I32B3RCF3FB232I33CF2BR3B2RIBRRFIFFB
LI3B2LFILBFLB3RIRR33RIBCA3CA32FBR2L2CALBR2I2BLILBCA3RBB3CA2I2LR3CARRLLRLRI23
IR23LBCAI2L3LL2I22CAR3FI3ICALI23B3R3CAF2BI2LCA3FRII3I2F3RLRICA2BCAIIA2IRBCAF3
B2CABBCA3RABLLFRLRLBRRBFLBRFRFFBLBRBFBRRBFCALBLLRLFLBCAB
```

Figura 17 – Exemplo de agente seguidor de bolas com visão local

É muito clara a diferença na complexidade dos indivíduos seguidores de bola em relação aos indivíduos seguidores de paredes principalmente por se tratar de uma tarefa mais difícil de ser executada: os indivíduos do Seguidor de Bola possuem uma complexidade muito maior.

A tabela 1 apresenta uma comparação entre o *fitness* e os pontos de complexidade dos melhores agentes de cada tipo. Pode-se perceber que a complexidade dos agentes com visão local sofreu um grande aumento, já que esta tarefa é mais complexa e exigiu uma maior elaboração dos indivíduos. Quanto ao *fitness*, por já saber onde se encontra a bola, o agente com visão global apresenta um melhor resultado, enquanto o seguidor de paredes atinge 776 pontos de um máximo de 780 – sendo quase perfeitamente adaptado.

	<i>Fitness médio (Melhores agentes de cada simulação)</i>	<i>Complexidade média (melhores agentes)</i>
Seguidor de Paredes	776	290
Seguidor de Bolas com Visão global	16433	535
Seguidor de bolas com Visão local	8447	1479

Tabela 1 – comparação entre *fitness* e complexidade dos agentes

8. CONCLUSÃO

A partir dos resultados obtidos, conclui-se que, usando Programação Genética é possível criar indivíduos que apresentam os comportamentos básicos para atuar no domínio do Futebol de Robôs, como seguir paredes e bolas, tanto utilizando um sistema perceptual baseado em visão global quanto um baseado em visão local.

Verificou-se que a medida que a complexidade da tarefa exigida pelo ambiente se torna maior, aumenta também a média da complexidade dos indivíduos que são capazes de realizar as tarefas corretamente. Assim, a complexidade dos agentes seguidores de bola com visão local é aproximadamente três vezes maior que a dos agentes com visão global – uma vez que a tarefa mais complexa exigiu uma maior elaboração dos indivíduos.

Uma tarefa mais complexa também aumenta o gasto computacional: por exemplo, para serem geradas 50 gerações de um agente que apenas siga uma parede, o tempo médio de execução do programa é de 30 minutos; já o agente seguidor de bola com visão global leva 50 minutos e o seguidor de bola com visão local 1:40 h (todos no mesmo computador, modelo Celeron com S.O. Linux).

A função de fitness é essencial para o controle dos indivíduos que permanecerão nas próximas gerações. Ela controla a velocidade da “evolução”, o resultado final e complexidade dos indivíduos. A escolha de uma função de fitness que não selecione corretamente os indivíduos que permanecerão acarreta no maior tempo necessário para evoluir uma população e até na criação de populações que não atuam como desejado. Quanto ao fitness, os agentes seguidores de paredes apresentam o melhor resultado – fato esperado, já que não dependem da bola. Entre os agentes seguidores de bola, apesar do fitness dos agentes com sistema de visão local ser menor, a performance dos dois tipos de agentes foi satisfatória.

É importante destacar que nenhum tipo de reedição foi utilizada para eliminar os ramos desnecessários ou reduzir a complexidade dos indivíduos, sendo os indivíduos aqui apresentados e seus comportamentos resultados dos conceitos de Programação Genética implementados.

A Programação genética permite a criação de indivíduos que produzem os resultados desejados desde que as habilidades necessárias para realização das tarefas e a relação dos indivíduos com o ambiente sejam conhecidas. Desse modo, um aspecto muito importante na Programação Genética é a simulação do ambiente, que deve incluir as dificuldades e dados relevantes do ambiente, pois os indivíduos criados só estarão bem preparados para reagir a situações conhecidas.

Para a obtenção de indivíduos capazes de realizarem outras tarefas deve-se realizar novos treinamentos com base nos mesmos indivíduos (se puderem ser aproveitadas características adquiridas com o treinamento anterior) ou em novos (como foi feito para o seguidor da bola). Ainda, para que um mesmo indivíduo seja capaz de executar duas tarefas, é necessário treiná-lo nos dois ambientes onde atuará, pois suas características somente permanecem ou se perpetuam quando as mesmas são necessárias para o seu “encaixe” no ambiente. Mudando o ambiente de um indivíduo pode tornar muitas das características, antes essenciais para a sua sobrevivência, desnecessárias.

Entre os possíveis trabalhos futuros em estudo encontram-se a evolução de comportamentos mais complexos como: goleiro, defesa, ataque, etc.; a inserção de comunicação entre os agentes para a formação de times; a implementação destes agentes no

Simulador Oficial da RoboCup; e, finalmente, a utilização dos melhores agentes gerados como módulos de controle de robôs reais.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- FIRA - Federation of International Robot-Soccer Association. **The rules of MiroSot**. FIRA, 1998. (<http://www.fira.net/fira/98fira/rules.html>)
- KIM, J.H.; VADAKKEPAT, P.; VERNER, I.M. “FIRA Robot World Cup Initiative and Research Directions”. **Int. J. of Robotics and Automation Systems**, 1998.
- KITANO, H. et al. “RoboCup: A challenge Problem for AI”. **AI Magazine**, v. 18, n. 1, p. 73-85, Spring 1997.
- KOZA, J. **Genetic programming: on the programming of computers by means of natural selection**. Cambridge, MA, MIT Press, 1992.
- KRAETZCHMAR, G. et al. The ULM Sparrows: Research into Sensorimotor Integration, Agency, Learning, and Multiagent Cooperation. In: **ROBOCUP WORKSHOP**, 2, Paris, 1998. **Proceedings**. FIRA, 1998. p. 459- 465
- MAIA, L.; BIANCHI, R. A. C. Evoluindo Agentes Jogadores de Futebol. In: **WORKCOMP’2000**, 3, São José dos Campos, 2000. **Proceedings**. CTA/ITA, pp. 113 - 119, 2000.
- RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Englewood Cliffs, Prentice Hall, 1995.
- SANDERSON, A. “Micro-Robot World Cup Soccer Tournament (MiroSot)”. **IEEE Robotics and Automation Magazine**, pg.15, December 1997.
- SHEN, W. et al. Integrated Reactive Soccer Agents. In: **ROBOCUP WORKSHOP**, 2, Paris, 1998. **Proceedings**. FIRA, pp. 251-264, 1998.
- TAMBE, M. Implementing Agent Teams in Dynamic Multi-Agent Environments. **Applied Artificial Intelligence**, v.12, March 1998.
- VELOSO, M., STONE, P. e HAN, K. The CMUnited-97 Robotic Soccer Team: Perception and Multiagent Control. In: **INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS**, 2, Minneapolis, 1998. **Proceedings**. AAAI, 1998.