

## EVOLUINDO AGENTES JOGADORES DE FUTEBOL

Luiz Carlos Maia Jr.<sup>1</sup>, 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 alcançar este objetivo, inicialmente é apresentado o domínio do Futebol de Robôs e são revistos conceitos básicos de Programação Genética. A seguir são descritos os agentes implementados: primeiro, agentes capazes de seguir paredes - problema tradicional da área de Programação Genética – e a seguir, agentes capazes de seguir uma bola, ambos em um ambiente simulado. Finalmente são apresentados resultados por meio de gráficos de evolução, exemplos de caminhos percorridos e exemplos de indivíduo.*

### ABSTRACT

*This paper describes the project and the implementation of agents capable of acting in the domain of Robotic Soccer, 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 implemented agents: first, agents capable of following walls – a traditional problem in the field of Genetic Programming – and then, agents capable of chasing a ball, both in a simulated environment. Finally are presented the results by means of evolution graphics, courses examples and individuals examples.*

## 1- INTRODUÇÃO

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 transformada em um time de jogadores de futebol de robôs.

Para isso, foi criado um ambiente de simulação onde os robôs se movimentam livremente, respeitando apenas os limites das paredes. Também foi criada uma família de robôs que, inicialmente, teve o objetivo de seguir paredes e que foi aperfeiçoada para seguir uma bola.

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

O domínio do futebol de robôs, liga de simulação, foi escolhido, pois permite testes dos indivíduos de maneira simulada, permitindo a realização de experimentos para o desenvolvimento e testes que apresentam comportamento inteligente. O futebol de robôs constitui um sistema bastante complexo por possuir diversos agentes inteligentes com objetivos diferentes dentro do campo e um objetivo em comum.

Na seção 2 está descrito o Domínio do Futebol de Robôs. A seção 3 é apresentada de forma sucinta, características da Programação Genética. Na seção 4 o Agente Seguidor de Paredes e seus resultados são apresentados. Na seção 6 tem-se o Agente Seguidor de Bola e seus resultados.

## 2- DOMÍNIO DO FUTEBOL DE ROBÔS

Partidas de futebol entre robôs, além de serem extremamente motivantes para possibilitar o surgimento de um espírito de ciência e tecnologia nas jovens gerações, 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.

Copas mundiais de Futebol de Robôs têm sido realizadas desde 1997, sendo que 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.

O funcionamento de cada time segue uma mesma fórmula básica: realiza a aquisição da imagem através da sua câmera, 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. Todo processamento é realizado em um único computador. Com a decisão de movimentação tomada, um sistema de comunicação por rádio envia para os robôs uma mensagem com o movimento a ser realizado.

Atualmente na Robocup (campeonato de futebol de robôs mundial), existe a liga de simulação, onde é utilizado um sistema cliente-servidor que gera um campo virtual e controla as ações dos jogadores. Um cliente comanda um jogador no campo e controla suas ações. As comunicações entre o cliente e o servidor são feitas por UDP/IP. Os clientes podem ser 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.

Entre os desafios e problemas a serem estudados, Shen (1998) afirma que “Jogadores Robóticos precisam realizar processos de reconhecimento visuais 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 estes 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).

<sup>1</sup> Bolsista Fundação de Ciências Aplicadas – FCA/FEI

### 3- PROGRAMAÇÃO GENÉTICA (PG)

#### 3.1 Evolução e Seleção

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 se tem hoje são consequência da evolução de outros organismos inferiores que se extinguíram 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, ou seja, é fruto do ambiente em que vive e não o contrário. Por muitas gerações, a variação aleatória de características e a seleção natural formaram o comportamento de indivíduos e espécies para que estes suprissem as demandas do seu ambiente.

Segundo Koza (1992), o processo evolutivo ocorre na natureza quando quatro condições básicas são satisfeitas:

- “Um indivíduo tem a habilidade de se reproduzir;
- Existe uma população desses indivíduos;
- Existe alguma variedade entre esses indivíduos;
- Alguma diferença em habilidade de sobrevivência está associada com a variedade.”

Em PG será 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. Este fato refere-se às formas de evolução utilizadas em PG como Reprodução, Cruzamento e Mutação.

#### 3.2 Fitness

*Fitness* é um parâmetro que mede o “encaixe” da espécie em seu ambiente. Existem várias formas de *fitness*, a usada é: “Quanto maior o *fitness* do indivíduo maior sua probabilidade de sobrevivência e adaptação ao ambiente”. Com este dado obtido, os indivíduos podem ser organizados de forma descendente, então são eliminados os piores e os melhores continuam para serem reproduzidos ou cruzados.

O *fitness* na Programação Genética é usado para que os piores indivíduos sejam descartados.

#### 3.3 Indivíduo

Em PG o indivíduo é definido como uma árvore (representações de árvores serão mostradas a seguir) de funções e terminais, que constituem suas características e comportamento no ambiente para o qual foi desenvolvido, cada função é um ramo e cada terminal uma folha do indivíduo.

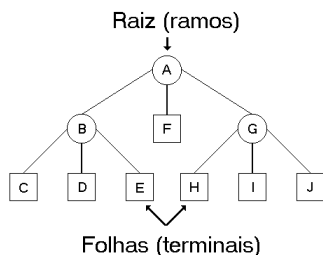


Figura 1 – Exemplo de Indivíduo

Funções: podem ser condições, sensores, operações aritméticas, booleanas, etc. São usadas funções para captar informações sobre o ambiente (mundo) e, convenientemente, para melhorar o desempenho de cada indivíduo.

Terminais: geralmente são variáveis ou constantes. Os terminais serão usados para fazer o indivíduo agir no ambiente.

#### 3.4 Reprodução

Os melhores indivíduos da geração atual são simplesmente copiados para a nova geração. O objetivo é que os melhores indivíduos não sejam perdidos.

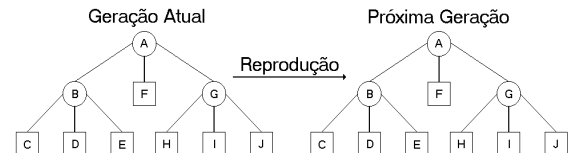


Figura 2 – Exemplo de Reprodução

#### 3.5 Cruzamento

São escolhidos dois indivíduos (pais) a partir do *fitness*, selecionando pontos de cruzamento, e deles criados dois descendentes que vão estar na próxima geração. O objetivo é que algo de novo seja criado usando o que há de melhor, os pais também são preservados para a próxima geração.

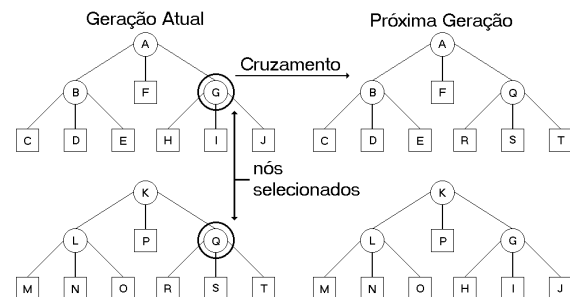


Figura 3 – Exemplo de Cruzamento

#### 3.6 Mutação

Um indivíduo é selecionado, depois um ponto de mutação, e a partir dele é criado um novo ramo. O objetivo é óbvio: melhorar um indivíduo já bom, mas com isso ele pode perder desempenho e ser eliminado. Observação: isso não impede que o indivíduo seja salvo antes de ser aplicado à mutação, o que normalmente acontece.

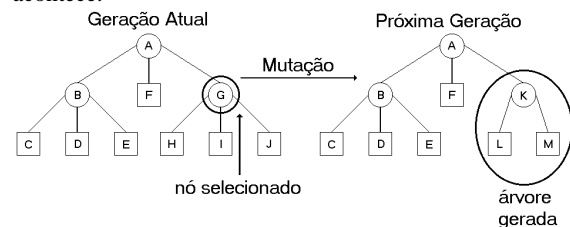


Figura 4 – Exemplo de Mutação

A mutação é pouco utilizada porque não melhora, relevantemente, o desempenho normal de uma execução.

#### 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.

##### 4.1 Características

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 '5000 passos' devido à grande área a ser percorrida.

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.

##### 4.1.1 Fitness

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.

##### 4.1.2 Funções

- 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.

##### 4.1.3 Terminais

- 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.

##### 4.1.4 Outros Parâmetros

- População (M) = 500;
- Probabilidade de Cruzamento = 70%;

- Probabilidade de Reprodução = 30%;
- Matriz de simulação = 200 x 200;
- Posicionamento do robô em números reais;
- Limite de complexidade (no sorteio) = 1000;
- Ângulo de virada = 30°;
- 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.

#### 4.2 Resultados

##### 4.2.1 Gráficos Evolutivos

O primeiro gráfico, do *fitness* em função da geração, 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.

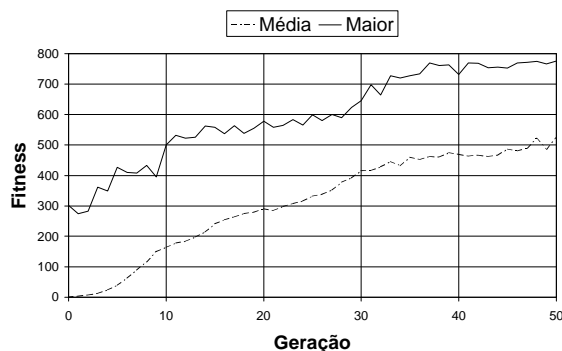


Figura 5 – Gráfico Evolutivo '51' gerações'

No próximo gráfico, em uma simulação de '42' gerações, 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.

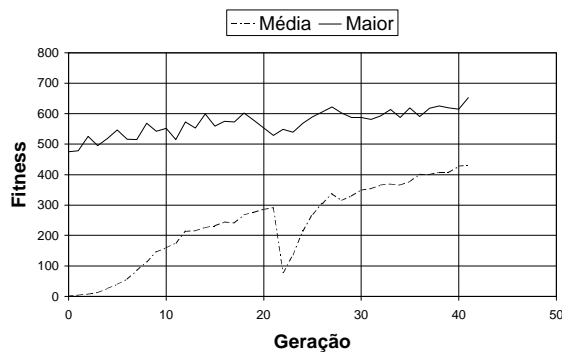


Figura 6 – Gráfico Evolutivo '42' gerações' com decimação na 21ª geração

#### 4.2.2 Exemplos de Caminhos

'Geração 10': o indivíduo dessa geração apesar de poder acompanhar a parede e detectar o obstáculo à sua frente, 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.

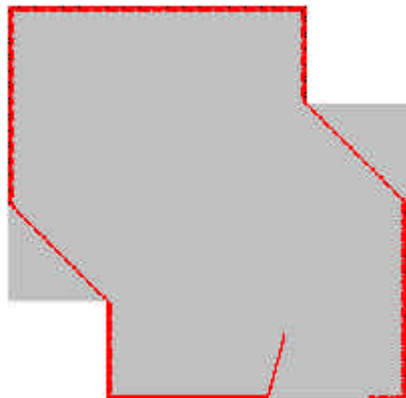


Figura 7 – Caminho Geração 10.

'Geração 26': esse 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.

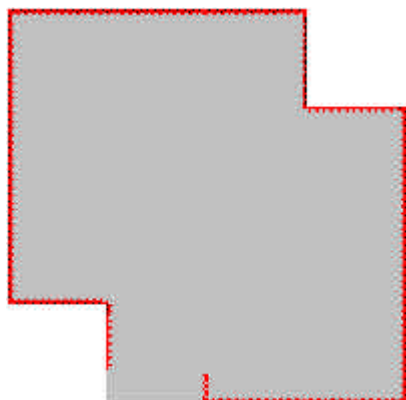


Figura 8 – Caminho Geração 26

'Geração 48': esse é o indivíduo excelente dessa execução, pois percorre todo o caminho dentro do número de movimentações estipulado. Consegue detectar as quinas que foram problema na 10ª geração e não perde tempo com isso como na 26ª geração.

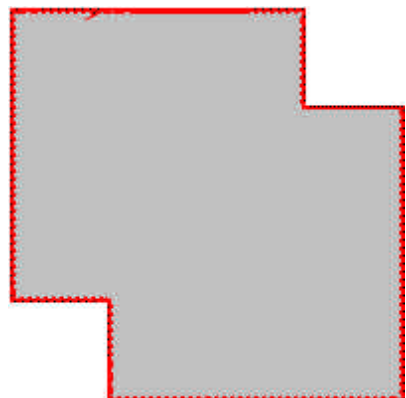


Figura 9 – Caminho Geração 48

#### 4.2.3 Exemplos de Indivíduos

Abaixo é mostrado o melhor indivíduo dessa simulação na forma em que ele é escrito em arquivo depois de ter sido evoluído, e os dados do segundo e terceiro melhores. Escrito dessa forma pode reutilizá-lo em novos testes:

##### Indivíduo 1

```
3IR22222FIF3B2II2FII2B2LI2B2B32F2LRR122BL3IBI
FIFBFB323L3B3BFB2F32RFF23IIB3FLIFRLR2L3IILL
ILFILLRIF23B2II32RLIFRBB2RIIBB33IR223I3IILL2I
2I33BBRBRBIRRL3IB32LRBRRBI2BRLRFR2L3222F
RIILLIFFRIFLRIF3BIBRLRFFR2FRRL3IILBILFIFLR
R3IB2I32LIRR2ILRBRILBBBBLFLFBFBLR2IBRRR
RIILLIFFRFI33FLIFR2LIRBB3ILLFIR3BBR
```

PONTOS DE COMPLEXIDADE = 290  
FITNESS = 776

##### Indivíduo 2

PONTOS DE COMPLEXIDADE = 110  
FITNESS = 766

##### Indivíduo 3

PONTOS DE COMPLEXIDADE = 44  
FITNESS = 761

Podem ser notadas as diferenças na complexidade dos três melhores indivíduos. O primeiro é o maior de todos, mas que também é o que possui maior *fitness*. O segundo mesmo tendo *fitness* menor não apresenta tanta inferioridade e possui menos da metade de pontos de complexidade do primeiro, o terceiro por sua vez também não perde muito em desempenho e possui complexidade inferior à metade do segundo. Deve ser lembrado também que todos possuem *fitness* muito próximos ao máximo, mesmo tendo sido descontados desvios, portanto cabe a nós escolher uma das soluções que nos foram apresentadas, se quiser um indivíduo que é apenas suficiente para suprir as nossas necessidades atuais a escolha mais óbvia seria o terceiro que apresentará um tempo de execução menor, mas se o caso for um indivíduo que venha a ter bom comportamento em outros ambientes primeiramente ele deverá ser testado nesse novo ambiente.

#### 5- AGENTE SEGUIDOR DE BOLA

O agente Seguidor de Bola foi escolhido para ser o segundo passo da evolução dos Agentes Jogadores de Futebol, pois apresenta maior dificuldade que o Seguidor de Paredes e adiciona elementos como a bola e obstáculos.

Além das funções de desvio de paredes e de movimentação, que foram reaproveitadas, adicionou-se apenas uma função que relaciona a movimentação do robô com a movimentação da bola, alterando o ambiente e a maneira como se mede o *fitness*.

##### 5.1 Características

Com as características da PG foi criado o programa do robô seguidor de bolas aonde o robô deve alcançar a bola e tocá-la o maior número de vezes possíveis dentro

da sua execução. O ambiente de simulação continua do mesmo tamanho, apesar de ter sido adicionados obstáculos, os passos são contados da mesma forma. A execução da árvore também não é parada enquanto não terminar.

O indivíduo deve percorrer a sala buscando pela bola em no máximo '2000 passos'. Os passos são reduzidos em relação ao caso anterior especificamente porque o robô não precisa se movimentar tanto para alcançar a bola, e para tornarem a representação final mais clara.

A cada simulação o robô e a bola começam em posições aleatórias.

### 5.1.1 Fitness

O *fitness* foi calculado de diversas maneiras. Abaixo, algumas das melhores maneiras e os motivos pelos quais são considerados melhores são apresentadas.

A fórmula que serviu de base para as outras é:

$$\text{fitness} = \text{hits} * k - \Sigma(n / \text{Dini});$$

Aonde 'hits' é o número de vezes que o robô atingiu a bola; n é o número de passos do robô e Dini é a distância inicial medida antes do robô se movimentar. O valor 'n / Dini' (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 será mostrado como ela influi no resultado.

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 ter do robô.

### 5.1.2 Funções

- 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.

Observação: são as mesmas do caso anterior.

### 5.1.3 Terminais

- 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.

### 5.1.4 Outros Parâmetros

- População (M) = 500;
- Probabilidade de Cruzamento = 70%;
- Probabilidade de Reprodução = 30%;
- Matriz de simulação = 200 x 200;
- Posicionamento do robô em números reais;
- Limite de complexidade (no sorteio) = 1000;
- Ângulo de virada = 5°;
- Número de execuções por indivíduo = 1.

O ângulo de 5° foi adotado por se tratar um valor suficiente e que permite bastante precisão na movimentação buscando a bola.

## 5.2 Resultados

### 5.2.1 Gráficos Evolutivos

No primeiro gráfico (50 gerações) nota-se uma grande diferença em relação ao Seguidor de Paredes: a variação exagerada do *fitness* do melhor indivíduo, isso se deve ao fato do robô interagir-se com outro objeto que também se movimenta, que é mais rápido que ele e que reflete em outros objetos, ou seja, executa-se apenas uma simulação de cada robô e a posição inicial aleatória influi no resultado. De qualquer forma o *fitness* nesse caso é mais útil para informar quantas vezes o robô consegue tocar a bola dentro da execução estipulada.

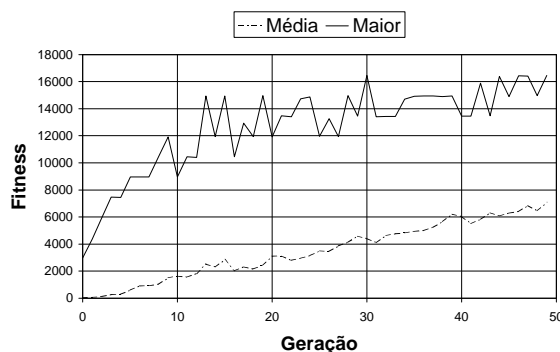


Figura 10 – Gráfico Evolutivo '50 gerações' k = 1500

No segundo gráfico (100 gerações com decimação) foi executada a decimação já citada, na '50ª geração'. Como no caso anterior tem-se a variação do *fitness* do melhor indivíduo e também é percebido que essa variação após algum tempo permanece dentro da mesma faixa, com poucas exceções. Essa simulação foi feita para observar a convergência final dos dados e a eventualidade de ter indivíduos por vezes tão bons, e como estes não poderiam ter perdido suas características de uma geração para outra. Até a centésima geração a média de *fitness* continua aumentando e a faixa em torno de onde varia o indivíduo com *fitness* máximo, também. Algumas vezes a posição inicial, sorteada de certa forma, pode favorecer o desempenho final e isso não foi retirado, pois a sorte também influi na evolução.

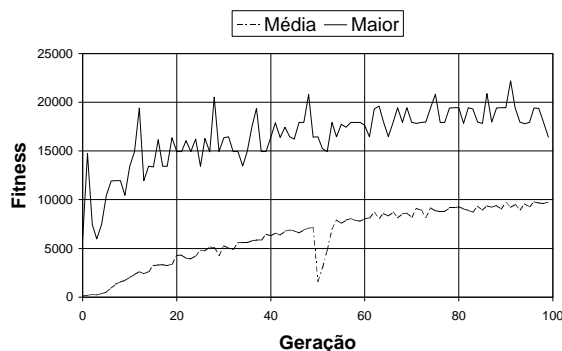


Figura 11 – Gráfico Evolutivo '100 gerações' com decimação - k = 1500

O último gráfico (50 gerações) mostra uma simulação onde 'k = 1000'. Isso 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 um pouco menor nos melhores. Pode-se notar indivíduos com *fitness* muito próximo à '14000' o que indica que atingiram a bola '14' vezes. No caso anterior com 'k =

1500' teve-se indivíduos com *fitness* próximo à '23000' o que indica '15' hits. Ou seja, obteve-se uma melhora em um aspecto (maior estabilidade dos dados finais) e em termos de *fitness* não se perdeu, pois a execução com 'k = 1500' foi feita duas vezes e somente na segunda execução (100 gerações) que se obteve indivíduos tão bons.

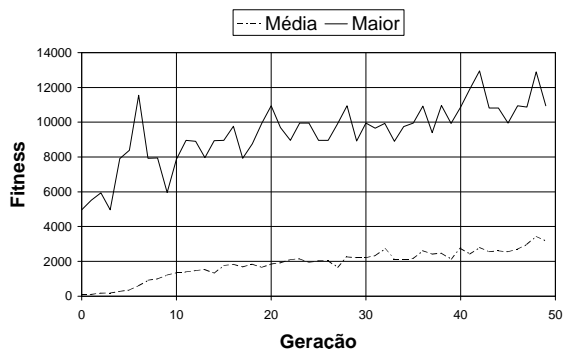


Figura 12 – Gráfico Evolutivo '50 gerações'  
k = 1000

### 5.2.2 Exemplos de Caminhos

As figuras a seguir foram extraídas da simulação com 'k = 1500'. Os quadrados brancos são obstáculos para o indivíduo e para a bola.

Nessas figuras o caminho feito pelo robô é representado pela cor vermelha. O amarelo representa o princípio do caminho, e o verde a posições da bola.

'Geração 1': Esse indivíduo toca poucas vezes na bola, mas pode perceber como ele já a persegue, apesar de não ter tido problemas com os obstáculos.

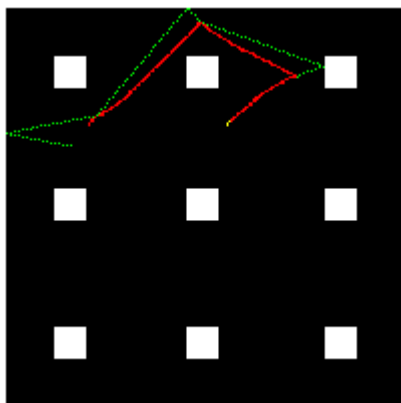


Figura 13 – Caminho Geração 1

'Geração 8': Exemplo de um caminho onde o indivíduo persegue muito bem a bola, por partir de uma distância maior que o exemplo anterior e mesmo assim possui *fitness* maior, formando uma parábola com o seu caminho.

'Geração 13': Indivíduo que inicia a busca pela bola já com um obstáculo entre ele e seu destino e utiliza suas funções e treinamento para desviar-se.

Geração 41': Indivíduo bastante evoluído que consegue por várias vezes atingir a bola já que o *fitness* o estimula a fazer isso. Nota-se também que este indivíduo não perde tempo ao seguir a bola e sempre tenta a melhor trajetória. Pouco antes de a execução terminar ele ainda buscava atingir a bola mais uma vez e fez uma pequena curva dentro da trajetória da bola.

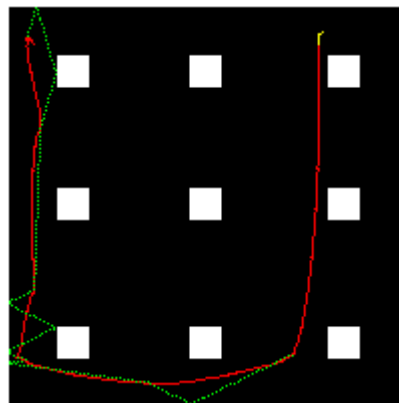


Figura 14 – Caminho Geração 8

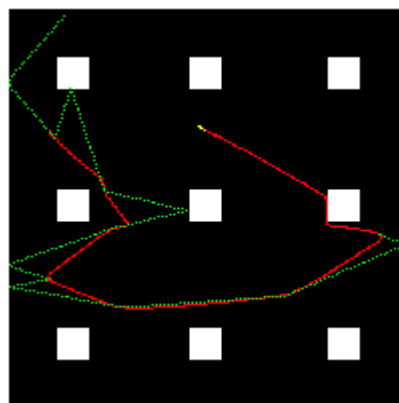


Figura 15 – Caminho Geração 13

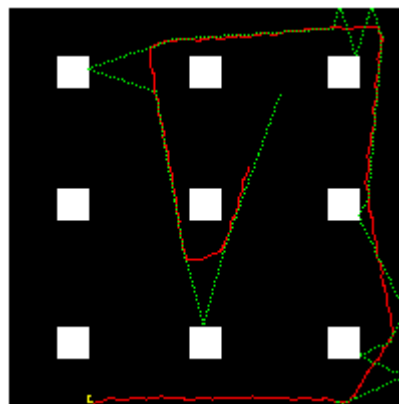


Figura 16 – Caminho Geração 41

### 5.2.3 Exemplos de Indivíduos

Abaixo é mostrado o melhor indivíduo da simulação com 'k = 1500' na forma em que ele é escrito em arquivo depois de ter sido evoluído, e os dados do segundo e terceiro melhores. Escrito dessa forma pode reutilizá-los em novos testes:

#### Indivíduo 1

```
3IIR2FF323I2ILIB32R3FRRR22L3RAF2BRFR323I2ILI
BILLIAAIR32F2IF3FFRFFF33RB2RRF33FFRILI23I2I
RIB32R3FR32AR32F2IIAL2FRFAF22IALF2BRR22L3
RF2IBF323IFIRR33RBII3FBBIRAFF33FF22F2IIAL3F
3RRFRFIRAI232FRR2B2BR33FBIRF3L2RF3FFR2RB
A2BIR32FRFBIRFII333F2LFFFIRBAI2IBIIAL3FF3F
FR3FFRAFL2BRFR323I2ILIB33BIAR22FF3A2IIAL3F
```

FRLFF2FRIAAIR32F2IF3FFRRFF33RF2RRF33FFRIL  
R2BIR32FR3FFRBFLI3FLBIRAF33FFRL32FRR2FIB  
FAIRF32RIIFRIF33AIL3A2FFFFLFR2BIR32FR2FRB  
AIRFII3FBBIRFF33FFRL3AR2FIBFAIRFII3L33FA3AI  
LIFRFFFFRIRAFRLR

PONTOS DE COMPLEXIDADE = 477  
FITNESS = 16436

#### Indivíduo 2

PONTOS DE COMPLEXIDADE = 696  
FITNESS = 16433

#### Indivíduo 3

PONTOS DE COMPLEXIDADE = 432  
FITNESS = 16431

É muito clara a diferença na complexidade dos indivíduos do Seguidor de Bola em relação aos indivíduos do Seguidor 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 complexidade do indivíduo não pode ser diretamente relacionada com o seu desempenho no ambiente. Os três indivíduos acima têm *fitness* muito próximos e o segundo se destaca por ter uma complexidade consideravelmente maior.

## 6- CONCLUSÃO

A partir dos resultados obtidos conclui-se que, usando Programação Genética, pode-se criar indivíduos que apresentam os comportamentos básicos para atuar no domínio do Futebol de Robôs, como seguir paredes e bolas.

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. Assim, 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.

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

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.

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.
- E finalmente, a implementação destes agentes no Simulador Oficial da RoboCup.

## REFERÊNCIAS BIBLIOGRÁFICAS

- KOZA, J. **Genetic programming: on the programming of computers by means of natural selection.** Cambridge, MA, MIT Press, 1992.
- 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.