



UNIVERSIDADE D  
**COIMBRA**

Dário Félix, N° 2018275530, [dario@student.dei.uc.pt](mailto:dario@student.dei.uc.pt)  
João Calhau, N° 2016255704, [uc2016255704@student.uc.pt](mailto:uc2016255704@student.uc.pt)  
Tatiana Simões, N° 2018285812, [2018285812@student.uc.pt](mailto:2018285812@student.uc.pt)

**TP2 - D31, THE RISE OF THE BALLZ**  
META 1 (MODELAÇÃO E DESENVOLVIMENTO)  
META 2 (EXPERIMENTAÇÃO E ANÁLISE)

**Relatório no âmbito da cadeira de Fundamentos de Inteligência Artificial da Licenciatura em Engenharia Informática, orientado pelos Professores Doutores João Correia (PL3) e Fernando Jorge Machado (T/TP), do Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.**

Maio de 2021

# Índice

<b>Introdução.....</b>	<b>2</b>
Sumário .....	2
Palavras-chave.....	2
<b>Implementação (código) .....</b>	<b>3</b>
Funções de Recombinação, Mutação e Seleção .....	3
Alterações ao código dado.....	3
Função de Aptidão .....	3
Parametrização.....	4
<b>Metodologia Experimental.....</b>	<b>5</b>
Preâmbulo .....	5
Exp. 1: probabilidade de mutação e crossover.....	5
Exp. 2: número de gerações .....	6
Exp. 3: marcar golo e a influência do tempo de simulação .....	6
Exp. 4: controlar a bola .....	6
Exp. 5: defesa .....	7
Exp. 6: 1 versus 1.....	7
<b>Resultados Experimentais .....</b>	<b>8</b>
Exp. 1: probabilidade de mutação e crossover.....	8
Exp. 2: número de gerações .....	8
Exp. 3: marcar golo e a influência do tempo de simulação .....	9
Exp. 4: controlar a bola .....	9
Exp. 5: defesa .....	10
Exp. 6: 1 versus 1.....	10
<b>Discussão .....</b>	<b>11</b>
Exp. 1: probabilidade de mutação e crossover.....	11
Exp. 2: número de gerações .....	11
Exp. 3: marcar golo e a influência do tempo de simulação .....	11
Exp. 4: controlar a bola .....	11
Exp. 5: defesa .....	12
Exp. 6: 1 versus 1.....	12
<b>Conclusão .....</b>	<b>13</b>
<b>Agradecimentos.....</b>	<b>14</b>
<b>Referências.....</b>	<b>14</b>

# Introdução

## Sumário

Relatório no âmbito do Trabalho Prático 2, “D31, THE RISE OF THE BALLZ”, da cadeira de Fundamentos de Inteligência Artificial do ano letivo 2020/21 da Licenciatura em Engenharia Informática da Faculdade de Ciências e Tecnologias da Universidade de Coimbra.

O objetivo deste trabalho é capacitar o controlador do agente em fazer determinadas tarefas como controlar a bola, marcar golos e defender, seja em ambientes estáticos, seja também em ambientes dinâmicos.

Para isso, serão feitas várias experiências, começando por explorar e compreender o impacto da probabilidade da mutação e do crossover, do tempo de simulação, do tamanho da população, do número de gerações, e do tamanho do torneio.

Depois, serão feitos testes à função de aptidão, modificando-a caso se justifique e alterando os pesos que cada informação (input) tem conforme as características do ambiente a ser considerado.

Este documento abrange as metas 1 e 2, modelação e desenvolvimento, e experimentação e análise, respetivamente. No início será abordado a implementação do código, e só depois é listado as experiências feitas com a seguinte ordem: parâmetros utilizados, resultados experimentais obtidos e, por fim, a discussão sobre esses resultados.

## Palavras-chave

Inteligência artificial; agentes adaptativos; algoritmos genéticos; selecção; operador de recombinação (*crossover*); operador de mutação; função de avaliação; aptidão (*fitness*); descendentes; tamanho da população; modelo elitista; número de gerações; tamanho do torneio; função gaussiana; ambiente dinâmico; tempo de simulação; rede neural artificial; parametrização; recombinação de 1-ponto; problema de maximização;

## Implementação (código)

Nesta fase desenvolveu-se as componentes essenciais que faltavam, sejam para cumprir os requisitos mínimos do algoritmo genético, ora sejam para o melhorar.

### Funções de Recombinação, Mutação e Seleção

Implementou-se o algoritmo de mutação gaussiana, *gaussianMutation()* em *GeneticIndividuals.cs*, e o algoritmo de seleção por torneio, *tournamentSelection()* em *TournamentSelection.cs*, como constam no pseudocódigo do enunciado. Note-se que o algoritmo de seleção por torneio segue a norma de um problema de maximização.

Desenvolveu-se o algoritmo de recombinação, *crossover()* em *GeneticIndividuals.cs*, sendo que é um operador de recombinação de 1-ponto e altera ambos os indivíduos.

Não se desenvolveu, por opção, outros operadores alternativos de mutação e recombinação para experimentação, tal como sugerido no enunciado. Aplica-se também ao número de neurónios na camada escondida (ficando os 3).

### Alterações ao código dado

Alterou-se o ficheiro *MetaHeuristics.cs* de modo a guardar os resultados separados por ponto e vírgula, ao invés de vírgulas, no ficheiro CSV.

### Função de Aptidão

Desenvolveu-se ao todo e ao longo do trabalho, várias versões de funções de aptidão, consequência das atualizações necessárias após a interpretação dos resultados experimentais.

Apesar disso, a ideia passava por ter uma só função de aptidão e colocar na equação todas as 15 informações dadas (os *inputs*), numa espécie de somatório, e colocadas numa forma a que, quanto maior for o seu valor, mais *fitness* terá, e, portanto, mais próximo estará daquilo que é pretendido.

Comecemos por abordar as funções de aptidão que vão ao encontro da ideia que foi descrita em cima (aplicam-se às funções *GetFitnessV3()* e *GetFitnessV4()* e parcialmente às funções *GetFitnessV1()* e *GetFitnessV2()* em

*D31NeuralControler.cs*). Para se poder colocar todas as 15 informações na equação foram normalizadas todos os 15 inputs (apesar das suas diferentes tipologias) para um valor que variasse entre 0 e 100. No caso de os inputs do tipo *array*, utilizou-se ou a média ou o valor máximo ou o valor mínimo para representar essa informação/*input* na “grande equação”, conforme a necessidade e a pertinência.

O objetivo era tirar partido da parametrização. Para isso utilizou-se um conceito de “pesos”, com uma escala entre “0” a “5”, que seriam posteriormente multiplicados individualmente e discriminadamente aos 15 inputs. O “0” representa a anulação desse input, o “1” permite a influência desse input na equação, mas num estado bruto (com um fator de multiplicação igual a 1). Do “2” ao “5” é aplicado um fator de multiplicação superior a 1 (valorizar essa informação), e crescente ao longo dessa escala.

Como já referido, outras soluções foram desenvolvidas que não cumprem a premissa anteriormente abordada, mas foram descartadas, e por isso, não serão objeto de estudo neste documento.

## Parametrização

Para além dos parametros já existentes, criou-se e desenvolveu-se o ficheiro auxiliar *Param.cs* de modo a permitir a expansão da parametrização em múltiplas secções de ficheiros diferentes.

Com isso, conseguiu-se parametrizar o seguinte:

- *isLogActivo*, indica se deve ou não imprimir no Log o valor das variáveis usadas na função de aptidão;
- tempo máximo de simulação;
- *mean* e *stdev*, significam respetivamente, a média e o desvio padrão, usados na função *MutateGuassian()*;
- qual versão da função de aptidão a utilizar (escolher entre 1 e 4);
- o valor que cada um dos pesos tem;
- associação dos pesos aos diferentes inputs (pelo número entre 0 e 5);
- *isAdvAtivo*, indica se deve ou não considerar na equação os inputs relacionados ao adversário (para ignorar variáveis com lixo);
- *normalizarValores*, valor utilizado para limitar e normalizar os inputs (recomenda-se o valor 100).

## Metodologia Experimental

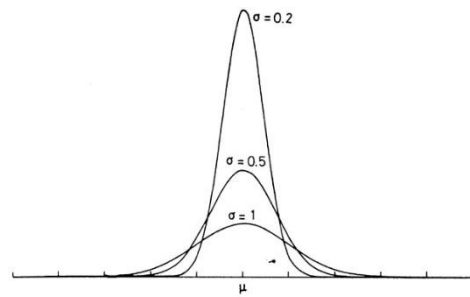
Procedeu-se à definição dos parâmetros e quais os parâmetros que iriam variar ao longo das experiências (*setup*).

Começou-se por explorar a variação da probabilidade da mutação e do crossover, do tempo de simulação e do número de gerações.

E só depois serão feitos testes à função de aptidão, modificando-a caso se justifique e alterando os pesos que cada informação (*input*) tem conforme as características do ambiente a ser considerado.

## Preâmbulo

A média e o desvio padrão, usados na função *MutateGuassian()*, permaneceram iguais ao longo de todas as experiências (*mean* = 0 e *stdev* = 0.5). É possível perceber o seu significado sem recorrer a testes: para o caso da média nula, quanto maior for o desvio padrão, também maior será a alteração (ver figura ao lado). [5]



O tamanho do torneio será sempre 2, uma vez que é o valor recomendado por vários artigos em consideração a experiências realizadas, apesar do autor consultado considerar que “por vezes pode não ser a melhor escolha e que pode depender do problema”. [2] [3]

Também se optou por não variar o tamanho da população, nem realizar experiências para perceber o seu impacto. O tamanho escolhido foi 42 (o mesmo número de simulações em simultâneo). Apesar de tudo verificou-se a sua relação com os parâmetros da probabilidade de mutação e crossover. [1]

Usou-se a opção elitista em todas as experiências.

Por fim, referir que se fez várias repetições da mesma experiência usando os mesmos parâmetros, mas com *seeds* diferentes e aleatórias, devido à natureza estocástica da abordagem.

## Exp. 1: probabilidade de mutação e crossover

Para esta experiência considerou-se o mapa *Evolving-ControlTheBallToAdversaryGoal*, utilizando a função de aptidão *GetFitnessV1()*, e optou-se por 100 gerações com um tempo de simulação de 7 segundos, apenas variando a probabilidade de mutação.

Teste A:

- Probabilidade de mutação = 0.5 (muito alto)

- Probabilidade de recombinação = 0.65

Teste B:

- Probabilidade de mutação = 0,005 (muito baixo)
- Probabilidade de recombinação = 0.65

## **Exp. 2: número de gerações**

Para esta experiência considerou-se o mapa *Evolving-ControlTheBallToAdversaryGoal*, utilizando a função de aptidão *GetFitnessV1()*, com 1000 gerações com um tempo de simulação de 20 segundos. Utilizou-se uma probabilidade de mutação de 0,03 e uma probabilidade de recombinação de 0,65.

## **Exp. 3: marcar golo e a influência do tempo de simulação**

Para esta experiência considerou-se privilegiar, e por esta ordem, as seguintes informações: quantos golos faz, quantos golos sofre (peso negativo), distância à bola (mínima), velocidade do agente (mínima), velocidade da bola (mínima), quantas vezes toca na bola, distância da bola à baliza do adversário (mínima) e a distância percorrida.

Configuração base:

- Mapa: *Evolving-ControlTheBallToAdversaryGoal*
- Função de aptidão: *GetFitnessV2()*
- Probabilidade de mutação: 0,03
- Probabilidade de crossover: 0,65

Teste A:

- Número de gerações: 300
- Tempo de simulação: 20 segundos

Teste B:

- Número de gerações: 100
- Tempo de simulação: 7 segundos

## **Exp. 4: controlar a bola**

Para esta experiência considerou-se privilegiar, e por esta ordem, as seguintes informações: quantos golos faz, quantos golos sofre (peso negativo), distância à bola (mínima), velocidade do agente (mínima), velocidade da bola (mínima), quantas vezes toca na bola, distância da bola à baliza do adversário (mínima) e a distância percorrida.

Configuração:

- Mapa: *Evolving-ControlTheBallToAdversaryGoalRandom*
- Função de aptidão: *GetFitnessV3()*
- Número de gerações: 200
- Probabilidade de mutação: 0,03
- Probabilidade de crossover: 0,60
- Tempo de simulação: 20 segundos

## Exp. 5: defesa

Para esta experiência considerou-se privilegiar, e por esta ordem, as seguintes informações: quantos golos faz, quantos golos sofre (peso negativo), distância à bola (mínima), velocidade do agente (mínima), quantas vezes toca na bola, distância da bola à baliza do adversário (mínima) e a distância percorrida. Também, através de cálculos das informações disponibilizadas, penalizar quando o agente fica entre a baliza adversária e a bola.

Configuração:

- Mapa: *Evolving-Defense*
- Função de aptidão: *GetFitnessV4()*
- Número de gerações: 500
- Probabilidade de mutação: 0,03
- Probabilidade de crossover: 0,60
- Tempo de simulação: 20 segundos

## Exp. 6: 1 versus 1

Para esta experiência considerou-se privilegiar, e por esta ordem, as seguintes informações: quantos golos faz, quantos golos sofre (peso negativo), distância à bola (mínima), velocidade do agente (mínima), velocidade da bola (mínima), quantas vezes toca na bola, distância da bola à baliza do adversário (mínima) e a distância percorrida. Também, através de cálculos das informações disponibilizadas, penalizar quando o agente fica entre a baliza adversária e a bola.

Configuração:

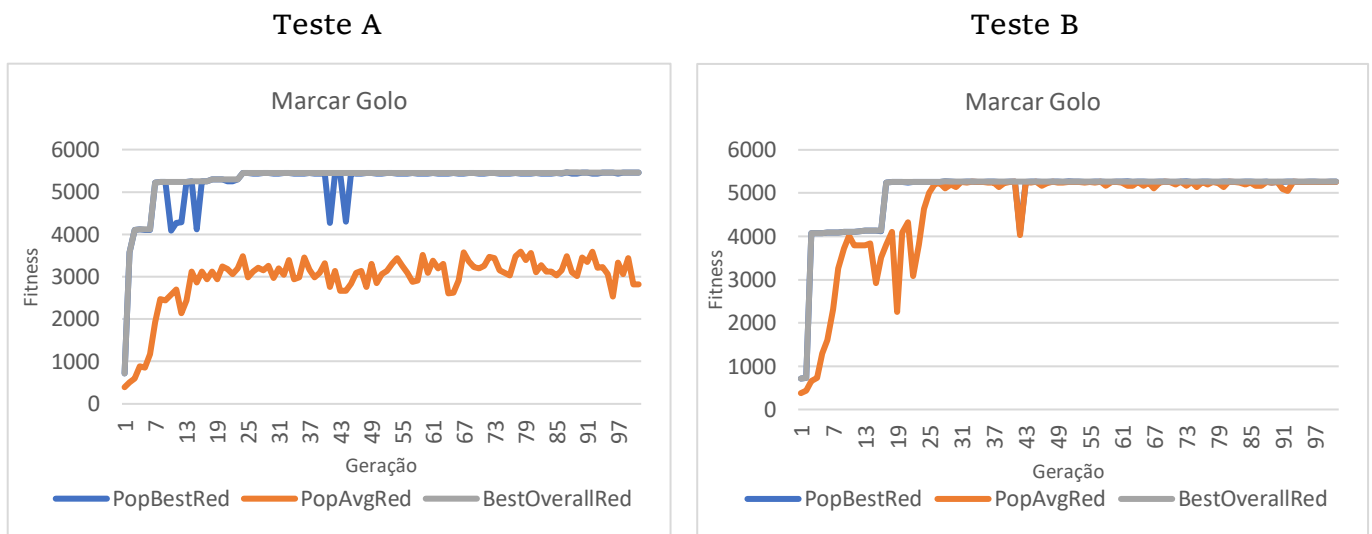
- Mapa: *Evolving-OnevsOne*
- Função de aptidão: *GetFitnessV4()*
- Número de gerações: 200
- Probabilidade de mutação: 0,03
- Probabilidade de crossover: 0,60
- Tempo de simulação: 20 segundos



## Resultados Experimentais

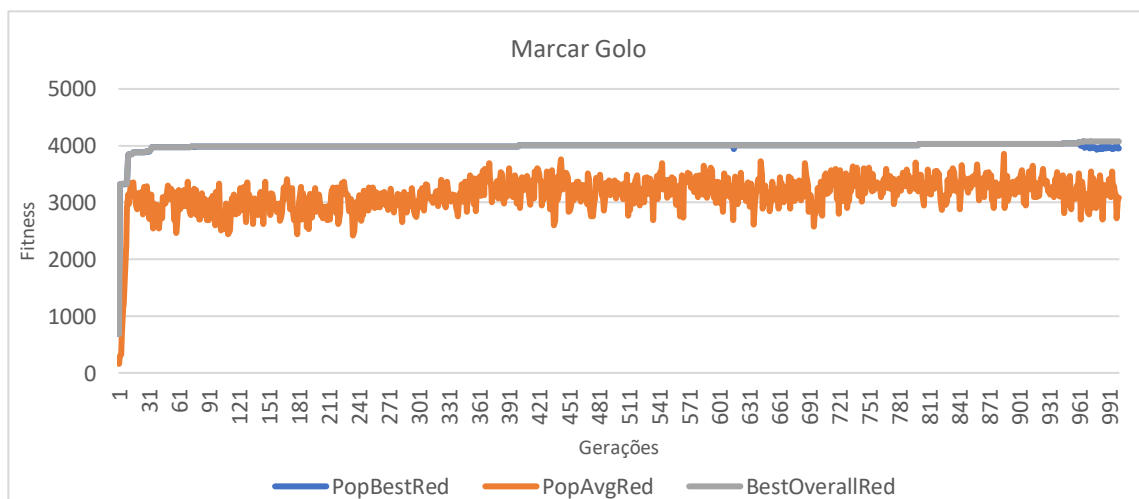
Em seguida, são apresentados os principais resultados obtidos em cada experiência.

### Exp. 1: probabilidade de mutação e crossover

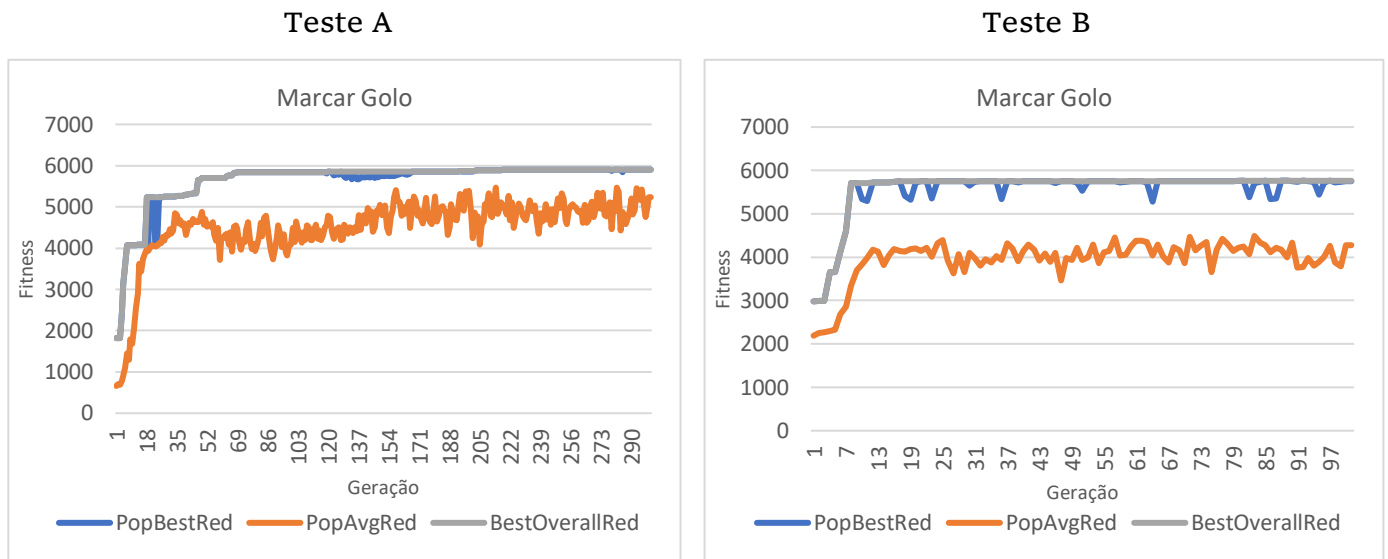


A principal diferença entre os dois testes é a distância do fitness entre a média da população e o melhor indivíduo dessa população. No Teste A é possível também verificar que em algumas gerações há perda dos indivíduos mais aptos.

### Exp. 2: número de gerações

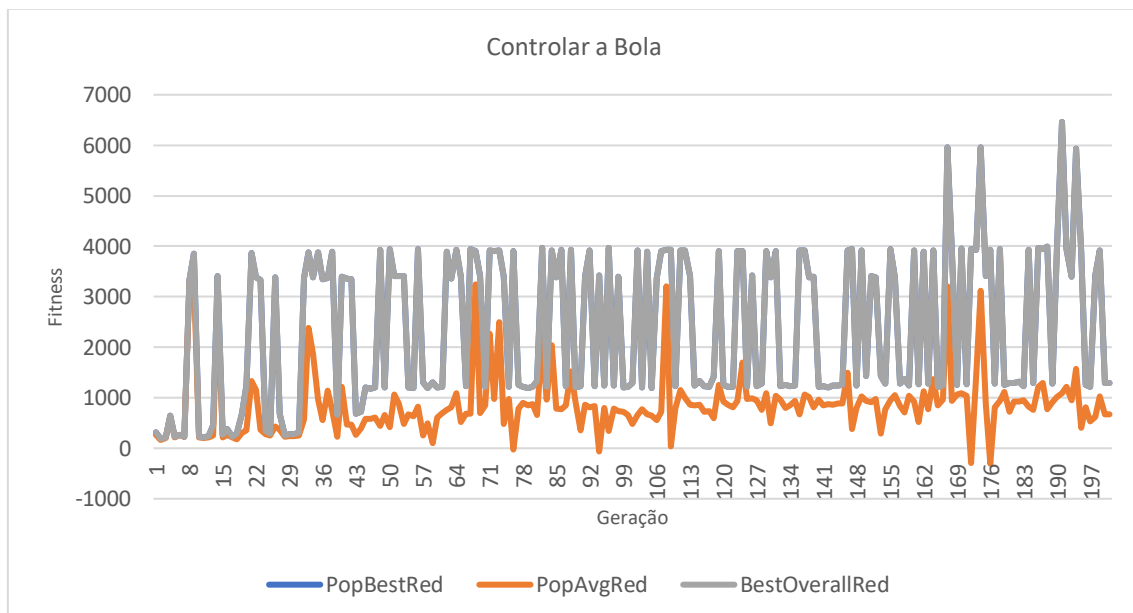


### Exp. 3: marcar golo e a influência do tempo de simulação



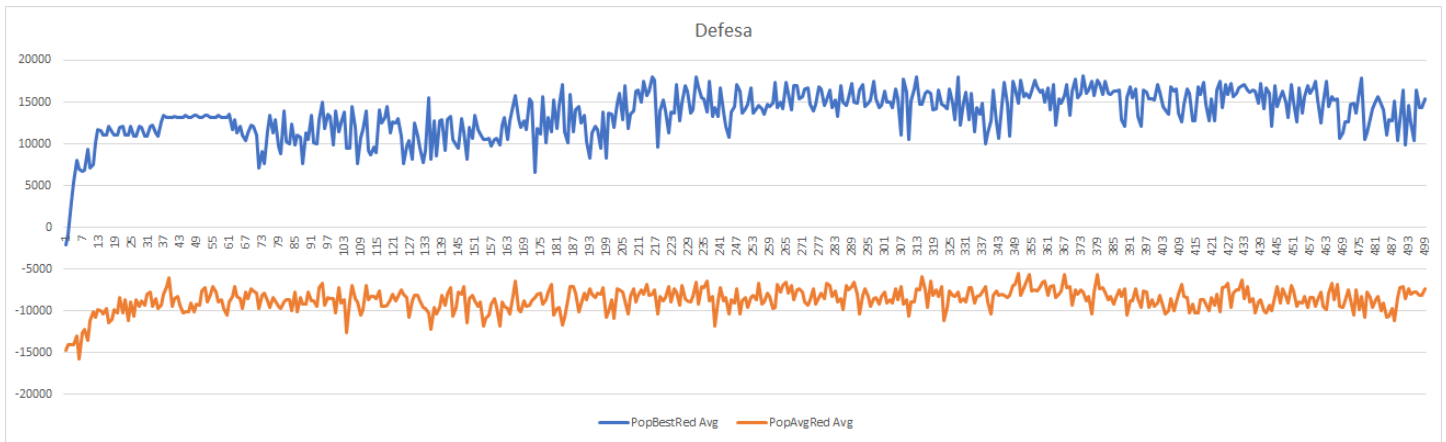
Além disto, no Teste A obtivemos um agente muito mais lento que o agente do teste B.

### Exp. 4: controlar a bola



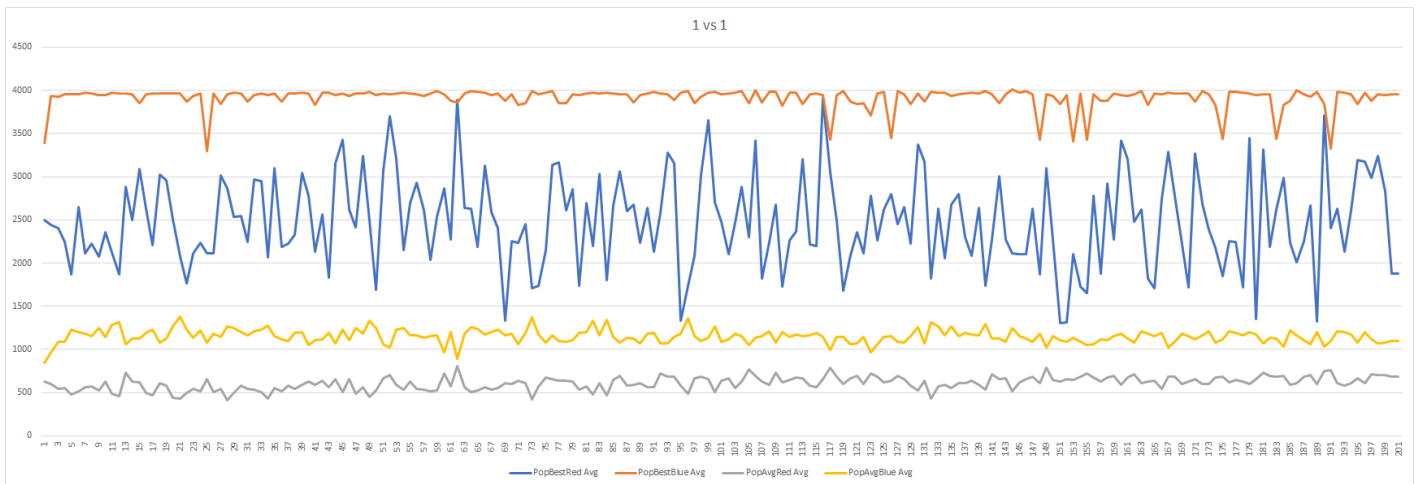
Verifica-se que o melhor numa determinada geração não é necessariamente o melhor da próxima geração (tendo em conta que a bola muda de posição).

## Exp. 5: defesa



Resultados semelhantes á experiência anterior.

## Exp. 6: 1 versus 1



É notável que o agente *Blue* foi quase sempre melhor que o agente *Red* (linha laranja versus linha azul).

## **Discussão**

Segue-se a análise e a discussão dos resultados obtidos.

### **Exp. 1: probabilidade de mutação e crossover**

Os resultados obtidos vão ao encontro daquilo que já sabíamos [1]: com uma probabilidade de mutação muito alta, e apesar de criar mais rapidamente variabilidade nos agentes, há uma destruição efetiva dos indivíduos que já são aptos. Com uma probabilidade de mutação muito baixa, não há muita variabilidade, e como consequência, a média da população fica próximo do best.

Daí se conclui que, para as próximas experiências, devemos usar valores na ordem dos 5%, 3% e 1%.

### **Exp. 2: número de gerações**

Verificamos que, a partir de uma determinada geração, o fitness estagna.

Nas próximas experiências não precisaremos de um número de gerações muito significativo.

### **Exp. 3: marcar golo e a influência do tempo de simulação**

O objetivo neste cenário foi facilmente cumprido. Reparamos que o tempo de simulação influencia na velocidade do agente em cumprir os objetivos deste cenário, uma vez que com tempos muito restritos, o algoritmo através do fitness privilegiará aqueles que conseguirem fazer mais rápido.

Por outro lado, com tempos muito restritos, por vezes, há uma perda dos indivíduos mais aptos.

### **Exp. 4: controlar a bola**

Os resultados obtidos não foram muito bons.

Apesar disso, o agente aprendeu a seguir e controlar a bola de modo a marcar golo. É possível verificar que o fitness varia bastante de geração para

geração, uma vez que, a bola ao mudar de posição, o agente adaptativo terá de interpretar de uma nova forma que será sempre diferente a todas as outras já ocorridas (de notar que a bola mudava de posição em todas as gerações).

### **Exp. 5: defesa**

Os resultados foram muito próximos ao da experiência anterior. A única diferença é que se obteve um valor muito mais negativo devido aos golos sofridos. Apesar de tudo, obteve-se indivíduos na população capazes de defender a baliza ao longo das gerações.

### **Exp. 6: 1 versus 1**

Os resultados obtidos nesta experiência foram bastante curiosos. Apesar de terem como fonte o mesmo ficheiro, o agente Blue foi sempre o melhor ao longo das gerações.

## **Conclusão**

Os objetivos foram globalmente cumpridos. Tivemos algumas dificuldades em conseguir agentes capazes de realizar as tarefas em ambientes mais dinâmicos (bola com posições aleatórias e em movimentação).

Para trabalho futuro, recomenda-se o aprofundamento das experiências nos ambientes mais dinâmicos, e o desenvolvimento de novas e melhores funções de aptidão.

## Agradecimentos

Agradecer particularmente ao Professor Doutor Fernando Jorge Machado por nos ter ensinado a construir um relatório minimamente decente.

E agradecer ao Doutor Frederico Varandas, ao Técnico Rúben Amorim, aos seus jogadores, e a outras ilustres personalidades e entidades que, apesar de não o saberem, ao conseguirem com que o Sporting Clube de Portugal fosse campeão 2020/21 de Portugal, contribuíram tanto para o sucesso deste trabalho como para o sucesso na cadeira. Obrigado.

## Referências

- [1] “The optimal crossover or mutation rates in genetic algorithm: a review”, acedido em maio de 2021, disponível em <https://www.cibtech.org/J-ENGINEERING-TECHNOLOGY/PUBLICATIONS/2015/VOL-5-NO-3/05-JET-006-PATIL-MUTATION.pdf> ;
- [2] “Experimental Analysis of the Tournament Size on Genetic Algorithms”, acedido em maio de 2021, disponível em [https://www.researchgate.net/publication/330478320\\_Experimental\\_Analysis\\_of\\_the\\_Tournament\\_Size\\_on\\_Genetic\\_Algorithms](https://www.researchgate.net/publication/330478320_Experimental_Analysis_of_the_Tournament_Size_on_Genetic_Algorithms) ;
- [3] “Tournament selection in genetic algorithm”, acedido em maio de 2021, disponível em <https://stackoverflow.com/questions/31933784/tournament-selection-in-genetic-algorithm> ;
- [4] “Agentes Adaptativos”, slides da cadeira FIA, acedido em maio de 2021;
- [5] Costa, E. e Simões, A. (2011). “Inteligência Artificial. Fundamentos e Aplicações”. 3ª edição, FCA – Editora de Informática. Lisboa.