



UNIVERSIDADE DE  
COIMBRA

FACULDADE  
DE CIÊNCIAS  
E TECNOLOGIA

## **Relatório do Trabalho Prático nº3: D31 The Rise of the Ballz**

Beatriz Isabel F. Santos  
*bisantos@student.dei.uc.pt*  
2017263066, PL6

José Veríssimo Lamas  
*lamas@student.dei.uc.pt*  
2017259895, PL6

Pedro José F. Marcelino  
*pjmarcelino@student.dei.uc.pt*  
2017277263, PL2

Departamento de Engenharia Informática  
Licenciatura em Engenharia Informática  
Introdução à Inteligência Artificial  
3º Ano, 2º Semestre  
2019/20

# Índice

|   |           |
|---|-----------|
| <b>1 Introdução.....</b>  | <b>3</b>  |
| <b>2 Implementação .....</b>                                    | <b>3</b>  |
| <b>3 Setup Experimental.....</b>                                | <b>4</b>  |
| <b>4 Resultados Experimentais e Análise .....</b>               | <b>5</b>  |
| 4.1 Mapa 1 – Evolving-ControlTheBallToAdversaryGoal .....       | 5         |
| 4.2 Mapa 2 – Evolving-ControlTheBallToAdversaryGoalRandom ..... | 6         |
| 4.3 Mapa 3 – Evolving-Defense.....                              | 7         |
| 4.4 Mapa 4 – Evolving-DefenseBallRandom .....                   | 8         |
| 4.5 Mapa 5 – Evolving-OnevsOne .....                            | 9         |
| <b>5 Conclusão .....</b>  | <b>12</b> |
| <b>6 Referências.....</b>                                       | <b>12</b> |
| <b>7 Anexos .....</b>   | <b>13</b> |

# 1 Introdução

No âmbito da cadeira de Introdução à Inteligência Artificial, foi-nos pedida a realização de várias experiências num simulador de treino virtual, para que o agente D31 ganhe a capacidade de realizar variadas tarefas, nomeadamente de defesa, controlo de bola, e jogar com um adversário.

Com estes testes pretende-se desmistificar redes neuronais e algoritmos genéticos usando agentes adaptativos. Vão ser utilizados vários mapas com diferentes ambientes, onde o agente irá ser treinado para adquirir diferentes aptidões e condicioná-lo de forma a que este obtenha o comportamento pretendido. As a rede neuronal irá evoluir tendo em conta os melhores resultados que forem obtidos ao longo do treino.

## 2 Implementação

Nesta fase foram implementados os algoritmos Algoritmo de Mutação Gaussiana, Algoritmo de Seleção por Torneio de acordo com o pseudocódigo fornecido no enunciado do trabalho prático. Foi também implementada uma função de recombinação (crossover), que combina a informação genética dos dois progenitores baseado numa probabilidade que vai sendo alterada ao longo da fase de experimentação.

É possível manipular os parâmetros da experiência a partir do Unity, tais como a probabilidade de crossover, probabilidade de mutação, número de gerações e neurónios de forma a facilitar a fase experimental.

A função de aptidão (fitness) foi alterada sempre que um comportamento precisava de ser encorajado ou penalizado. A implementação inicial da função foi aplicada tanto ao indivíduo da equipa azul como ao indivíduo da equipa vermelha e é a seguinte:

$$fitness = (distanceTravelled * 1000 + GoalsOnAdversaryGoal) - 1$$

### 3 Setup Experimental

Após a fase de implementação foram planeados todos os testes necessários para realizar a análise experimental. As condições em que as experiências foram realizadas foram as seguintes: Em cada um dos 5 mapas diferentes (*Evolving-ControlTheBallToAdversaryGoal*, *Evolving-ControlTheBallToAdversaryGoalRandom*, *Evolving-Defense*, *Evolving-DefenseBallRandom*, *Evolving-OnevsOne*), foi testado o algoritmo genético. O algoritmo foi testado com probabilidades de mutação de 0.15, 0.25, 0.5 e 0.9 e probabilidade de crossover de 0.9, 0.8, 0.6, 0.05. Em todos os mapas foram aplicados 3 neurónios com uma população de 50 indivíduos e com o tamanho do torneio a 50.

Os indivíduos foram testados ao longo de 200 gerações em cada experiência para que conseguissem evoluir mais facilmente.

Como os algoritmos usados têm uma componente estocástica, as experiências foram repetidas 3 vezes para cada mapa com as seguintes seeds aleatórias: 3021, 4022.

| <i>Mapa x</i>             |       |                  |                    |                 |                |           |                 |
|---------------------------|-------|------------------|--------------------|-----------------|----------------|-----------|-----------------|
|                           | Seeds | Prob. de Mutação | Prob. de Crossover | Nº de Neurónios | Nº de Gerações | População | Tam. Do Torneio |
| <i>Algoritmo Genético</i> | 3021  | 0.15             | 0.9                | 3               | 200            | 50        | 50              |
|                           | 4022  | 0.25             | 0.8                |                 |                |           |                 |
|                           |       | 0.5              | 0.6                |                 |                |           |                 |
|                           |       | 0.9              | 0.05               |                 |                |           |                 |
|                           |       |                  |                    |                 |                |           |                 |

Tab. 1 – Dados para as experiências efetuadas em cada mapa.

## 4 Resultados Experimentais e Análise

### 4.1 Mapa 1 – *Evolving-ControlTheBallToAdversaryGoal*

Neste primeiro mapa o objetivo foi treinar os agentes para que estes conseguissem ir ao encontro da bola e marcar na baliza adversária. A função de fitness inicial reforça a distância percorrida, os golos no adversário e bater na bola. Esta função foi testada várias vezes com probabilidade de crossover e mutação distintas. Foi possível observar que, de um modo geral, em todos os testes experimentais a população se aproximava muito das paredes do campo de jogo e alguns indivíduos conseguiram nas gerações iniciais ir ao encontro da bola. Das gerações 100 a 200 os indivíduos da população sincronizaram e já tentavam bater na bola e andar em direção da baliza adversária, marcando golos em alguns casos. Neste mapa só foi treinado o indivíduo vermelho. Após uma primeira análise verificou-se que em todas as repetições dos testes com diferentes *seeds* o comportamento do agente adaptativo foi semelhante. Neste mapa o indivíduo foi evoluindo ao longo das gerações tendo os melhores resultados com probabilidade de mutação de 0.15 e probabilidade de crossover de 0.9. O agente obteve melhor pontuação quando a probabilidade de crossover era mais elevada. No entanto este resultado deveu-se aos pesos elevados na função de fitness.

A função de aptidão inicial implementada apenas encorajava a distância viajada e os golos na baliza adversária o que mostrou não ser a melhor para o ambiente em causa, pois o agente ficava sempre encostado a um canto do campo de jogo sem evoluir muito ao longo das gerações. A função que permitiu alcançar o objetivo do mapa foi a seguinte:

$$\begin{aligned} \text{fitness} &= \text{distanceTravelled} + \text{GoalsOnAdversaryGoal} * 1000 + \text{hitTheBall} * 500 \\ \text{fitness} &= \text{fitness} - (\text{distanceToBall.Average}() * 3000) \end{aligned}$$

Esta função encorajou o agente a ir contra a bola e marcar golo na baliza adversária.

|                                   | <i>Melhor Pontuação</i> | <i>Pontuação Média</i> | <i>Melhor Global</i> |
|-----------------------------------|-------------------------|------------------------|----------------------|
| <b><i>PM = 0.15, PC = 0.9</i></b> | 536889.7                | 267626.01              | 538497.2             |
| <b><i>PM = 0.25, PC = 0.8</i></b> | 79812.9                 | 23007.3                | 80957.1              |
| <b><i>PM = 0.5, PC = 0.6</i></b>  | 8692.9                  | 2622.0                 | 13467.1              |
| <b><i>PM = 0.9, PC = 0.05</i></b> | 6074.1                  | 1594.9                 | 14333.5              |

Tab. 2 - Resultados médios ponderados ao longo de 200 Gerações. (PM – Probabilidade de Mutação, PC = Probabilidade de Crossover)

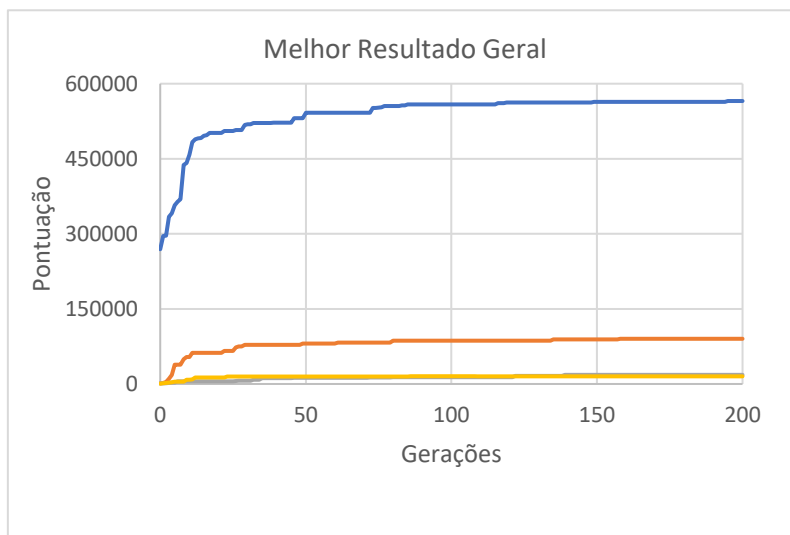


Gráfico 1 - Melhor resultado geral ao longo de 200 gerações.

**(Legenda: Azul-  
BestOverallRed  
(PM = 0.15, PC = 0.9),  
Laranja – BestOverallRed  
(PM = 0.25, PC = 0.8),  
Amarelo – BestOverallRed  
(PM = 0.5, PC = 0.6),  
Cizento – BestOverallRed  
(PM = 0.9, PC = 0.05)).**

## 4.2 Mapa 2 – *Evolving-ControlTheBallToAdversaryGoalRandom*

Neste mapa ao contrário do anterior, a bola aparece em lugares aleatórios no mapa de jogo o que torna treinar o agente bastante mais complicado.

Como os indivíduos se aproximavam dos limites do campo de jogo no mapa anterior, a função de aptidão foi alterada de forma a penalizar este comportamento, daí o decremento da variável - *hitTheWall*.

Após esta adaptação na função de fitness, foi possível observar que os indivíduos tentaram não se aproximar das paredes e focar-se em bater na bola e marcar na baliza adversária. Os comportamentos começaram a ficar cada vez mais rápidos ao longo das gerações.

Nas primeiras gerações (geração 60) como se usou a informação obtida do primeiro mapa, ao diminuir a *distance travelled* o indivíduo limitou-se a andar em frente e reduzir a velocidade.

Por volta da 100ª geração quase todos os indivíduos começaram a marcar golos e autogolos. Em média na 160ª geração os agentes tocavam na bola e andavam para trás enquanto a bola estava a andar na direção da baliza oposta. Alterações às funções de fitness não resultaram.

```
fitness = (distanceTravelled/100) + (GoalsOnAdversaryGoal*10000) + (hitTheBall * 5000) - hitTheWall;
fitness = fitness - (distanceToBall.Sum() * 3000) - (GoalsOnMyGoal*500)
```

Foi feita uma nova alteração à função de aptidão, mas o agente demora algumas gerações a agir de uma forma expectável, demorando sempre pelo menos 2 gerações a adaptar-se à bola numa nova posição. Após atingir a bola uma vez o agente não tenta atingi-la de novo com frequência, sendo que, em alguns casos bate diretamente numa parede e noutros acompanha a bola a uma distância algo constante, mas sem a tocar. Nesta função foi introduzido o peso *avgSpeed* de forma a fazer com que o robô se movimentasse mais rápido e atingisse a bola com força suficiente para marcar golo. Quando a bola se encontra numa boa posição (em frente ao agente ou um pouco para os lados) existem membros da população que conseguem marcar golos com alguma regularidade.

```
fitness = 0;
if(hitTheBall > 0){
    fitness = hitTheBall * 10;
}else{
    fitness = distanceTravelled - hitTheWall;
}
fitness = (fitness * avgSpeed) - (distanceToBall.Sum() * 3) - (GoalsOnMyGoal * 5) - hitTheWall
+ (GoalsOnAdversaryGoal * 10000);
```

|                                   | <i>Melhor Pontuação</i> | <i>Pontuação Média</i> | <i>Melhor Global</i> |
|-----------------------------------|-------------------------|------------------------|----------------------|
| <b><i>PM = 0.15, PC = 0.9</i></b> | 283035.7                | 18324.3                | 283035.7             |
| <b><i>PM = 0.25, PC = 0.8</i></b> | 330735.3                | 23410.6                | 330735.3             |
| <b><i>PM = 0.5, PC = 0.6</i></b>  | 149734.0                | 22507.7                | 149734.0             |
| <b><i>PM = 0.9, PC = 0.05</i></b> | 36533.9                 | 3147.7                 | 3533.9               |

Tab. 3 - Resultados médios ponderados ao longo de 200 Gerações. (PM – Probabilidade de Mutação, PC = Probabilidade de Crossover)

Foi feita uma última tentativa de alteração da função de fitness e o agente começa com movimentos aleatórios, acabando muitas vezes por ficar colado á parede. À medida que vai evoluindo vai passando a aproximar-se da bola, tocando nela com alguma frequência e marcando golos, embora acidentalmente. Às vezes o robô não toca na bola, acabando começando-se a afastar dela quando se aproxima demasiado (uso de Average() em vez de Sum()). Quando a bola muda de posição são sempre precisas algumas gerações para o agente se ajustar à nova posição, o que influenciou bastante o seu desenvolvimento e os resultados.

```
float fitness = 0;
if (hitTheBall == 0) {
    fitness = fitness - ((distanceToBall.Average()) * 2000) - (hitTheWall * 4000);
} else {
    fitness = distanceTravelled + GoalsOnAdversaryGoal * 10000 + hitTheBall * 8000; fitness =
    fitness - ((distanceToBall.Average() * 2000) - (hitTheWall * 4000) * (1 + GoalsOnMyGoal)) ;
}
```

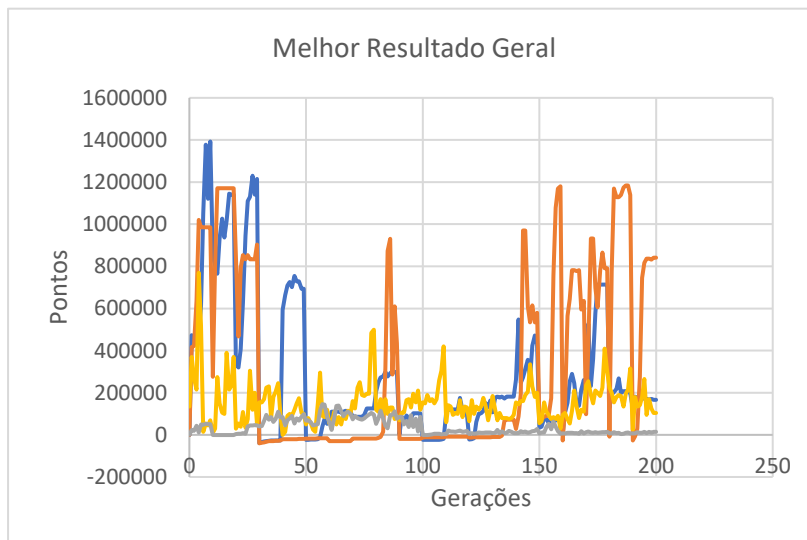


Gráfico 2 - Melhor resultado geral ao longo de 200 gerações.

**(Legenda:** Azul-  
BestOverallRed  
(**PM = 0.15, PC = 0.9**),  
Laranja – BestOverallRed  
(**PM = 0.25, PC = 0.8**),  
Amarelo – BestOverallRed  
(**PM = 0.5, PC = 0.6**),  
Cizento – BestOverallRed  
(**PM = 0.9, PC = 0.05**)).

### 4.3 Mapa 3 - *Evolving-Defense*

Neste mapa a bola encontra-se no centro e mexe-se em direção à baliza e o agente tem como objetivo defender de modo a que a bola não entre na sua própria. Nas gerações iniciais, o agente defendia a bola e empurrava-a para o seu campo, mesmo com a penalização na função de fitness. Nas gerações seguintes o agente já defendia a bola mais certamente e empurrava-a para o canto superior direito do campo de jogo. Em algumas simulações o agente ultrapassava-a a bola e tentava manter-se à frente dela.

De um modo geral foi possível observar que o agente cumpriu o objetivo pretendido, sendo mais eficaz quando a probabilidade de mutação era 0.9 e a probabilidade de recombinação 0.05. A função de fitness usada nesta fase de testes foi a seguinte:

```
float fitness = 0;
if(hitTheBall > 0){
    fitness = hitTheBall * 100;
}else{
    fitness = distanceTravelled/5 - hitTheWall;
}
fitness = fitness - (distanceToBall.Average() * 3) - (GoalsOnMyGoal*50) - hitTheWall + (GoalsOnAdversaryGoal*1_000_000);
```

É também possível observar que o agente foi evoluindo ao longo das gerações sendo o sucesso de aprendizagem dependente das probabilidades utilizadas.

|                                   | <i>Melhor Pontuação</i> | <i>Pontuação Média</i> | <i>Melhor Global</i> |
|-----------------------------------|-------------------------|------------------------|----------------------|
| <b><i>PM = 0.15, PC = 0.9</i></b> | 366.2                   | -34863.6               | 606.8                |
| <b><i>PM = 0.25, PC = 0.8</i></b> | 404.5                   | -39404.1               | 821.7                |
| <b><i>PM = 0.5, PC = 0.6</i></b>  | 7.7                     | -147.3                 | 7.87                 |
| <b><i>PM = 0.9, PC = 0.05</i></b> | 558                     | -198795.8              | 832.8                |

Tab. 4 - Resultados médios ponderados ao longo de 200 Gerações. (PM – Probabilidade de Mutação, PC = Probabilidade de Crossover)

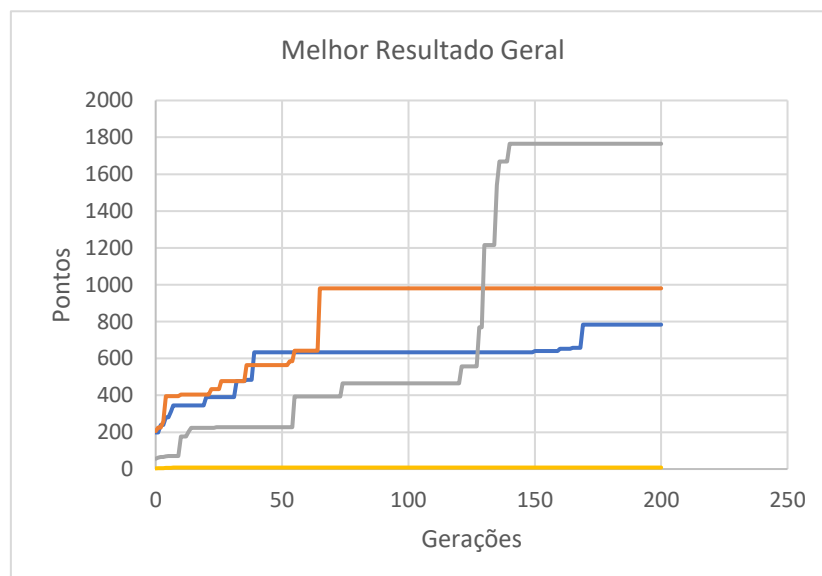


Gráfico 3 - Melhor resultado geral ao longo de 200 gerações.

**(Legenda:** Azul-  
PopAvgRed  
(***PM = 0.15, PC = 0.9***),  
Laranja – PopAvgRed  
(***PM = 0.25, PC = 0.8***),  
Amarelo – PopAvgRed  
(***PM = 0.5, PC = 0.6***),  
Cizento – PopAvgRed  
(***PM = 0.9, PC = 0.05***)).



#### 4.4 Mapa 4 - *Evolving-DefenseBallRandom*

Neste mapa a bola aparecia em diferentes posições e movimentava-se em direção à baliza vermelha. O agente à semelhança do mapa anterior tentava defender a bola, no entanto demorava a reagir. A função de aptidão foi alterada de forma a que o agente empurrasse a bola para longe da própria baliza e tentava-se manter próximo desta. O agente obteve valores negativos visto que foi bastante penalizado por sofrer golos e como o agente demorava algum tempo a adaptar-se acabou por não obter pontos positivos. No entanto cumpriu o objetivo deste mapa ainda que demorasse algum tempo a fazê-lo. A função de fitness usada nesta fase de testes foi a seguinte:

*float fitness = 0; f*

*fitness += GoalsOnAdversaryGoal\*10 - GoalsOnMyGoal\*100\_000 + distanceTravelled + distancefromBallToMyGoal.Average()\*100 + hitTheBall\*5;*

|                                   | <i>Melhor Pontuação</i> | <i>Pontuação Média</i> | <i>Melhor Global</i> |
|-----------------------------------|-------------------------|------------------------|----------------------|
| <b><i>PM = 0.15, PC = 0.9</i></b> | -2984788.7              | -13176078.7            | -2984788.7           |
| <b><i>PM = 0.25, PC = 0.8</i></b> | -4726107.7              | -16737272              | -4726107.7           |
| <b><i>PM = 0.5, PC = 0.6</i></b>  | -25146.3                | -216319                | -25146.3             |
| <b><i>PM = 0.9, PC = 0.05</i></b> | -144655.0               | -255357.4              | -144655.0            |

Tab. 5 - Resultados médios ponderados ao longo de 200 Gerações. (PM – Probabilidade de Mutação, PC = Probabilidade de Crossover)

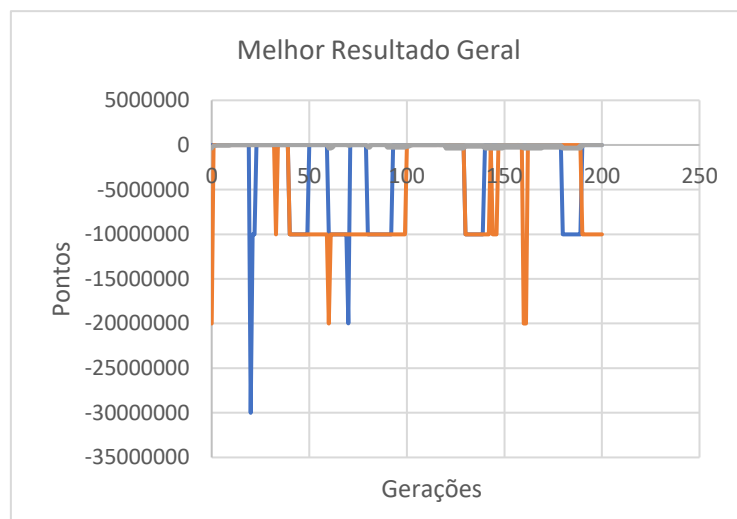


Gráfico 4 - Melhor resultado geral ao longo de 200 gerações.

(**Legenda:** Azul- *PopAvgRed* (***PM = 0.15, PC = 0.9***), Laranja – *PopAvgRed* (***PM = 0.25, PC = 0.8***), Amarelo – *PopAvgRed* (***PM = 0.5, PC = 0.6***), Cizento – *PopAvgRed* (***PM = 0.9, PC = 0.05***)).

## 4.5 Mapa 5 –*Evolving-OnevsOne*

Neste mapa existem dois jogadores onde a bola aparece sempre no centro e cada um dos jogadores aparece na sua baliza.

|                            | <i>Melhor Pontuação</i> | <i>Pontuação Média</i> | <i>Melhor Global</i> |
|----------------------------|-------------------------|------------------------|----------------------|
| <i>PM = 0.15, PC = 0.9</i> | 269207.2                | 43934.1                | 269207.2             |
| <i>PM = 0.25, PC = 0.8</i> | 471674.1                | -40803.4               | 471674.1             |

Tab. 6 - Resultados médios ponderados ao longo de 200 Gerações para o agente vermelho (PM – Probabilidade de Mutação, PC = Probabilidade de Crossover)

|                            | <i>Melhor Pontuação</i> | <i>Pontuação Média</i> | <i>Melhor Global</i> |
|----------------------------|-------------------------|------------------------|----------------------|
| <i>PM = 0.15, PC = 0.9</i> | 259281.0                | 40587.0                | 259281.0             |
| <i>PM = 0.25, PC = 0.8</i> | 517422.0                | 43970.9                | 517422.0             |

Tab. 7 - Resultados médios ponderados ao longo de 200 Gerações para o agente azul (PM – Probabilidade de Mutação, PC = Probabilidade de Crossover)

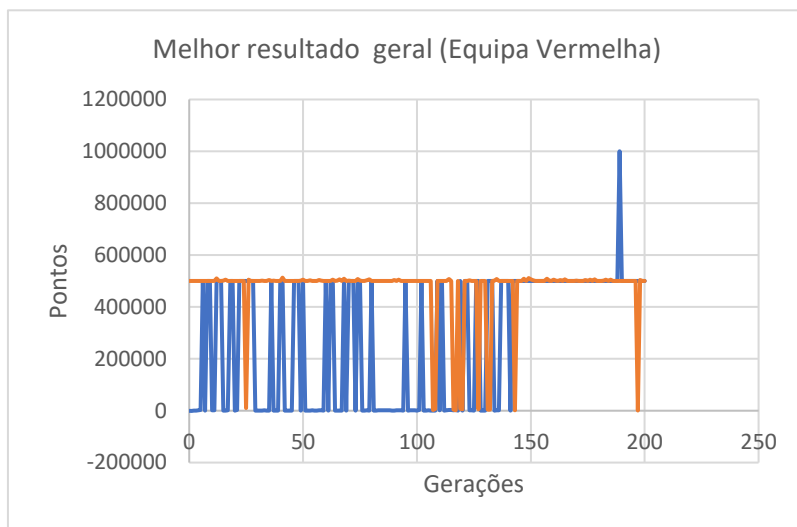


Gráfico 5 - Melhor resultado geral ao longo de 200 gerações.

(**Legenda:** Azul- *PopAvgRed* ( $PM = 0.15$ ,  $PC = 0.9$ ),  
Laranja – *PopAvgRed* ( $PM = 0.25$ ,  $PC = 0.8$ )).

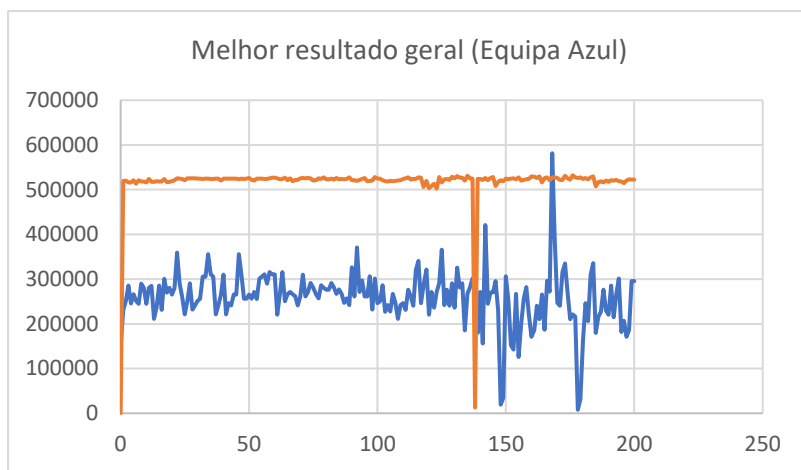


Gráfico 6 - Melhor resultado geral ao longo de 200 gerações.

(**Legenda:** Azul- *PopAvgRed* ( $PM = 0.15$ ,  $PC = 0.9$ ),  
Laranja – *PopAvgRed* ( $PM = 0.25$ ,  $PC = 0.8$ )).

## 5 Conclusão

Neste trabalho é possível observar através dos testes efetuados, que o jogador evolui de uma forma reativa, o que influencia de forma significativa o seu desenvolvimento em mapas com elementos aleatórios. Também podemos concluir que, as probabilidades de crossover e de mutação desempenham um papel essencial no desenvolvimento da inteligência artificial. Nomeadamente, uma probabilidade de mutação elevada permite ao robô adaptar-se mais facilmente a ambientes desconhecidos, pois está relacionado com a geração de comportamentos diferentes na população, comportamentos estes que podem ser mais eficazes do que o atual. Verifica-se também que uma probabilidade de crossover elevada é benéfica quando nos encontramos em ambientes com elementos maioritariamente estáticos, pois estes permitem que o robô melhore o seu comportamento atual (ao invés de gerar comportamentos novos), pegando nos melhores membros da população anterior e usando-os para gerar a nova geração

## 6 Referências

- **Inteligência Artificial: Fundamentos e Aplicações**  
Ernesto Costa, Anabela Simões
- **Artificial Intelligence: A Modern Approach**  
Stuart Russel, Peter Norvig

## **7 Anexos**

- Juntamento com este relatório, será enviado três ficheiros com resultados de todas as experiências efetuadas, os gráficos utilizados e o package do Unity onde foram efetuadas as experiências.
  - BestDados\_e\_Gráficos.xlsx
  - TestesIIA.pdf
  - Tp3\_final.unitypackage