



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
COIMBRA



---

***Fundamentos de Inteligência Artificial***  
**2022/2023 - 2º semestre**

---

**Trabalho Prático Nº2**  
**Rolling in the Hill Evolutionary Edition**

Trabalho realizado por:

Bruno Sequeira nº2020235721 brunosequeira@student.dei.uc.pt

Rui Santos nº 2020225542 rpsantos@student.dei.uc.pt

Tomás Dias nº 2020215701 tomasdias@student.dei.uc.pt

PL1

## META 1 - Modelação e desenvolvimento

### Código:

No enunciado era nos apresentado pseudo-código, este que passamos para os scripts para implementação do projeto.

### Realização de testes:

Após a implementação do código, testamos as funcionalidades do algoritmo evolucionário por forma a garantir o seu bom funcionamento. Testamos assim com o cenário **GapRoad**.

Na função de fitness existem várias opções que contribuem para o cálculo do valor. Visto estarmos num passo inicial, optamos por considerar a distância percorrida e o tempo necessário a que o veículo percorre a tal distância.

Na nossa opinião, a distância percorrida é mais importante que o tempo a que este percorre portanto optamos pela seguinte função:

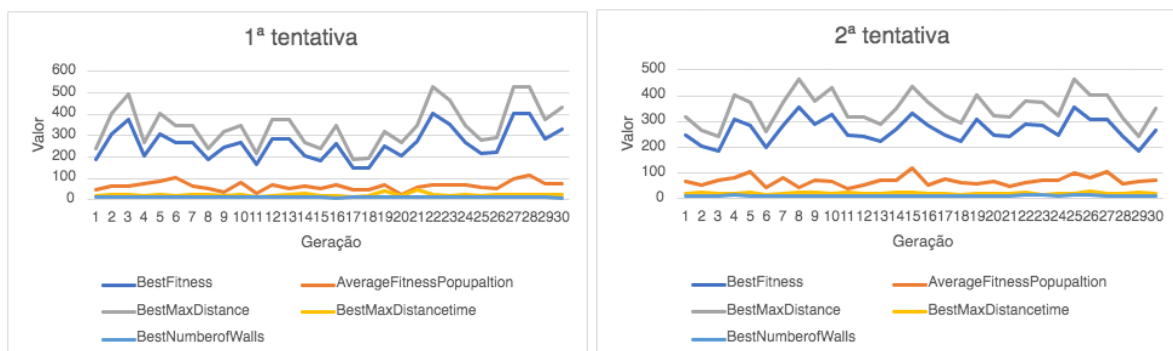
$$\text{fitness} = 0.75 * \text{distância} + 0.25 \text{ tempo}$$

Ou seja, a distância tem 75% de peso e o tempo 25%.

Os parâmetros que utilizamos nas experiências foram os seguintes:

	Mutação	Elitismo	Crossover	Número Gerações
Experiência 1	0.05	0	0.9	30
Experiência 2	0.2			
Experiência 3	0.05	2		
Experiência 4	0.2			
Experiência 5	0.05			

Apresentamos agora os resultados da experiência 1:



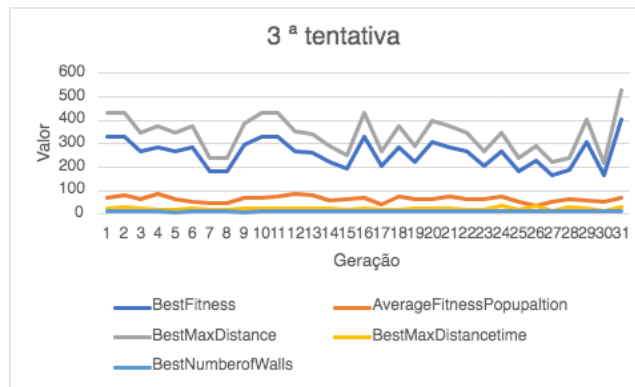


Figura 1,2,3 - Experiência 1 e respectivas tentativas.

Apresentamos os resultados da experiência 2:

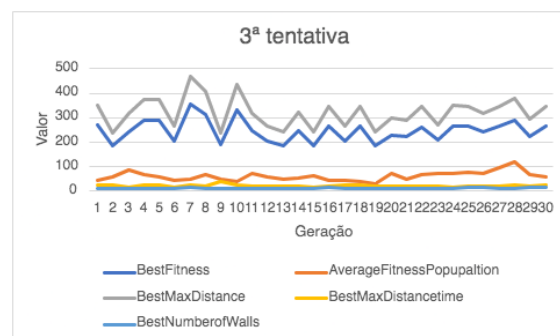
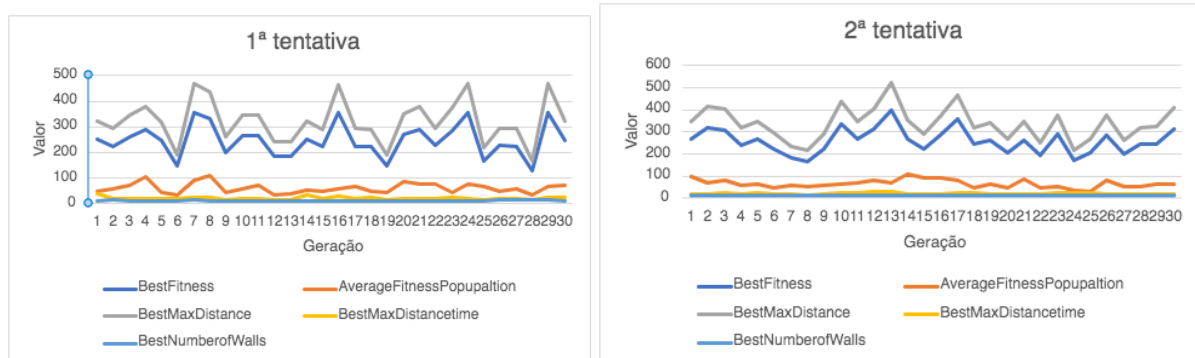
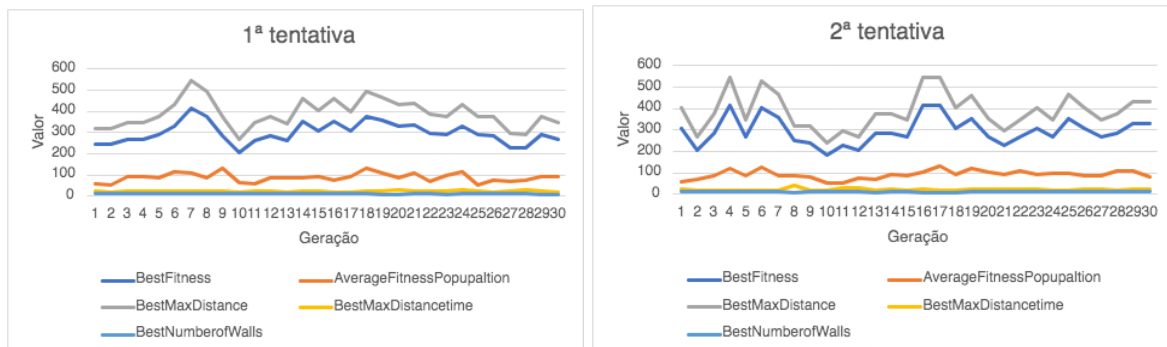


Figura 3,4,5 - Experiência 2 e respectivas tentativas.

Apresentamos os resultados da experiência 3:



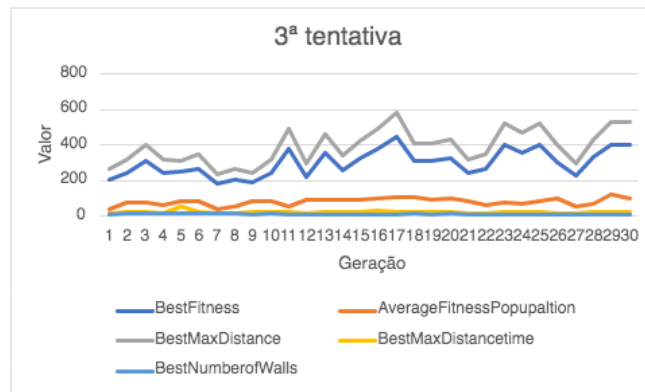


Figura 6,7,8 - Experiência 3 e respectivas tentativas.

Apresentamos os resultados da experiência 4:

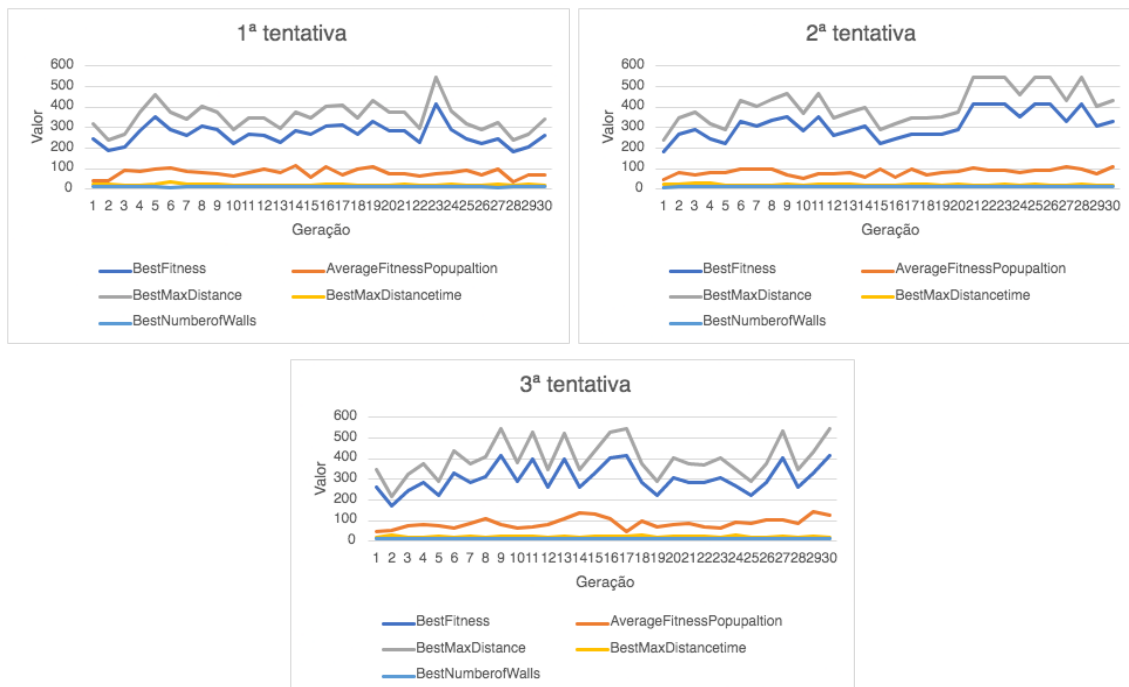


Figura 9,10,11 - Experiência 4 e respectivas tentativas.

## Notas e Conclusões:

A experiência 3 e 5 apresentam os mesmos parâmetros, logo apenas realizamos as 3 tentativas para a experiência 3.

Como previsto, a evolução do Best Fitness acompanha a da BestMaxDistance e é bastante parecida com a BestMaxDistanceTime, visto que esta é calculada por 75% da BestMaxDistance e 25% da BestMaxDistanceTime.

A Partir dos gráficos acima ilustrados, percebemos que existe um certo grau de aleatoriedade, visto que a AverageFitnessPopulation varia consoante a geração.

Visualizamos também que existindo um maior valor probabilidade de mutação do genótipo, as variações são mais bruscas. A explicação para essa ocorrência é o facto de que os veículos diferem mais dos veículos da geração anterior.

## Meta 2 - Experimentação e análise

Nesta meta tínhamos como objetivo implementar uma função de fitness que em conjunto com as configurações de mutação, elitismo e crossover os veículos conseguissem percorrer com sucesso as pistas GapRoad, HillRoad e ObstacleRoad.

Após análise do código desenvolvido na meta1 melhoramos o código do crossover visto que não tínhamos implementado o “.clone()”, permitindo assim um melhoramento no desenvolvimento do projeto.

Analisando o resultado da meta1 acerca do GapRoad apesar de se notar evoluções dos veículos a função de fitness não foi bem implementada, visto que nenhum veículo conseguiu completar a pista.

Antes de apresentarmos os resultados, temos certas conclusões acerca dos parâmetros acima apresentados(Configurações):

- Mutação é a percentagem de diferença que o veículo tem em relação ao melhor da geração anterior. Ou seja, se a mutação for de 0.05 implica que este muda 5% em relação ao melhor da geração anterior.
- CrossOver - um operador genético usado para variar a informação (“cromossomas”) de uma geração para a próxima. Sendo que são escolhidos com a roleta. Quanto maior este for mais a informação entre o pai e o filho aumenta.
- Elitismo é o número de veículos que iremos à geração anterior para a nova geração. Basicamente se o elitismo estiver a 2 faz com que este vá buscar os 2 melhores veículos da geração anterior para os colocar na nova, se for 0 não vai buscar nada, fazendo com que a próxima geração seja pior visto que os melhores da geração anterior já não estão presentes.
- Número de Gerações tal como o nome indica é o nº de gerações sendo que utilizamos o que pedem no enunciado, 30 gerações.

Após esta pequena introdução iremos apresentar as experiências que nos permitiram avançar no projeto, sendo o objetivo deste ver uma evolução entre gerações e ter veículos a chegar à meta, e se possível quando aparecer um veículo a chegar à meta as gerações seguintes também o conseguirem.

Nota:

Todas as implementações estão na pasta Results do projeto.

## GapRoad:

Este percurso corresponde uma pista que contém vários pedaços de “estrada” com buracos a meio e quanto mais pista o carro anda mais esses buracos ficam mais compridos.

No nosso ponto de vista, a mutação favorece se for baixa, visto que se a mutação “estragar” o desenvolvimento do veículo não será por muito. Uma desvantagem poderá ser o tempo que demora a ter uma geração que complete uma pista.

Mutação	Elitismo	CrossOver	Nº de Gerações
0.01	2	0.9	30

Tabela 1 - Configurações para a experiência 1 do GapRoad.

Após conversarmos concordamos que na função de fitness teríamos de utilizar os seguintes parâmetros:

- MaxDistance - a distância máxima que o veículo percorreu.
- MaxDistanceTime - o Tempo que demorou a percorrer a pista.
- IsRoadComplete - Completou a pista ou não.

Função de Fitness	$\text{Math.Pow}(\text{MaxDistance}, 3) + 400000 * \text{IsRoadComplete} + \text{Math.Pow}((\text{MaxDistance} / (\text{MaxDistanceTime} + 1)), 2);$
-------------------	--

Tabela 2 - Função de Fitness para a experiência 1 do GapRoad.

Optamos por maximizar a distância percorrida pelo carro, e a velocidade média deste.

A razão da distância percorrida é óbvia pelo facto de que quanto mais percorre, maior a probabilidade de este completar a pista. A velocidade média se fosse elevada permitia que o carro nunca “batesse” nos blocos da estrada e passasse por cima destes. Elevamos a distância a 3 para que este na roleta tivesse uma probabilidade de ser escolhido. Juntando assim com a velocidade média tornaria o carro mais forte em relação a todos os outros, permitindo assim notar-se uma diferença em relação aos outros. Se o carro conseguisse completar a pista recebia cerca de 400000 de bônus.

Apresentamos agora os resultados que obtemos acerca desta implementação:



Figura 12,13,14 - Experiências GapRoad.

## Notas e Conclusões:

Quando o BestFitness é superior a 1640000000 implica que o veículo completou.

A função de fitness resolveu o problema sendo que na experiência 1, a partir da 15ª geração existiu sempre um veículo que completou a pista, na geração 21 ocorre um pico visto que o carro saltou por cima da meta, ultrapassando assim os 547 metros, sendo que este no fim retorna “False” quando na realidade completou a pista com sucesso.

A experiência 2, existem bastantes True's, e tal como a experiência 1 existem Falses a meio mas foi pelo facto do carro ter saltado por cima da meta.

A experiência 3 também ocorre pela mesma razão.

Acerca do Average Fitness Population, ocorre uma evolução principalmente na Experiência 1 e 2. A experiência 3 no final decai um pouco os valores do Average Fitness provavelmente porque a mutação que ocorreu foi “demasiado má” para que os veículos conseguissem completar a meta apesar de pelo menos 1 ter conseguido ultrapassar a meta.

Concluindo este percurso foi completado com sucesso. Passaremos agora para o HillRoad.

## HillRoad:

Este percurso corresponde uma pista que apresenta várias subidas e descidas, quanto mais distância é percorrida mais as subidas/descidas serão mais acentuadas.

A nossa primeira implementação é a mesma que utilizamos no GapRoad, para não repetirmos a mesma informação, convidamos a sua leitura na página 6. Apresentamos agora os resultados obtidos:

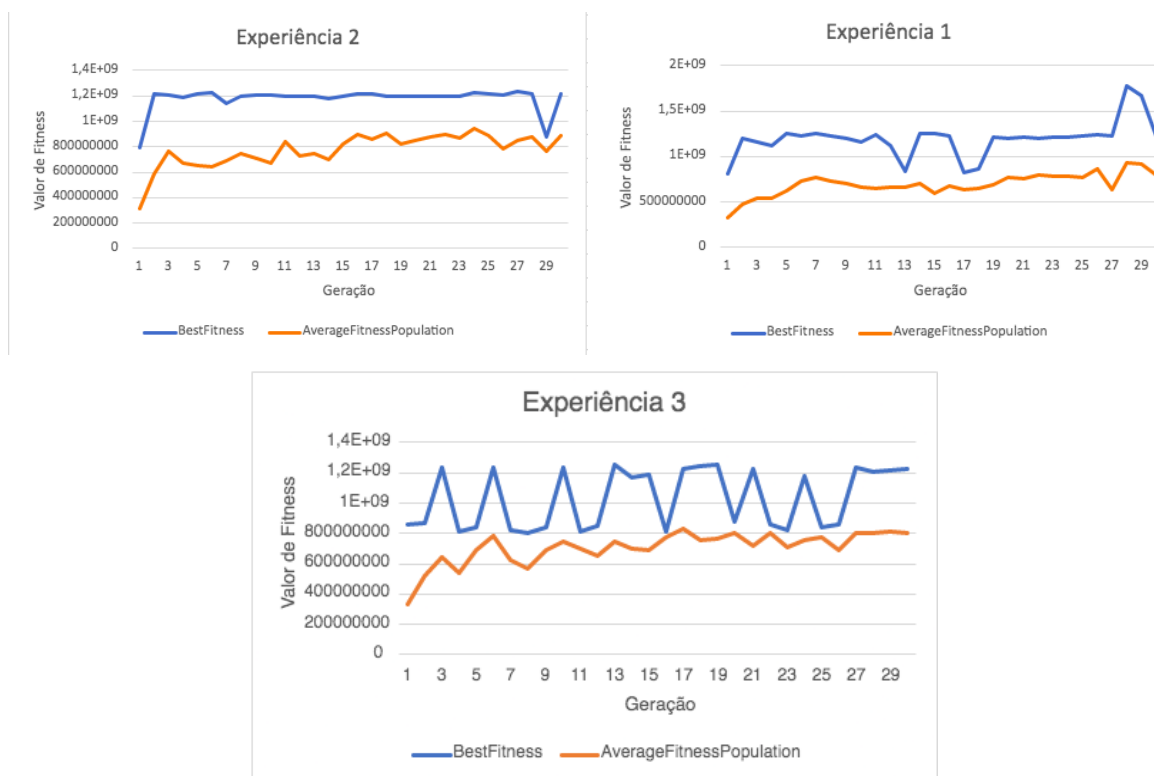


Figura 15,16,17 - Experiências HillRoad.

## Notas e Conclusões:

Antes de mais, esta implementação não conseguiu que um único veículo ultrapassasse a meta. Analisando o gráfico, pouco se vê de evolução, apenas da Average Fitness.

Após verificarmos no cenário Evaluation Hill, podemos reparar que o veículo não consegue passar dos 1200 metros aproximadamente, logo a nossa função de fitness não consegue resolver o problema, portanto iremos seguir para uma nova implementação com uma ideia diferente, usando novos parâmetros.



## 2ª implementação do cenário HillRoad:

Para as configurações utilizamos as seguintes:

Mutação	Elitismo	CrossOver	Nº de Gerações
0.01	2	0.9	30

Tabela 7 - Configurações para a experiência 2 do HillRoad.

Para função de fitness teríamos de utilizar os seguintes parâmetros:

- MaxDistance - a distância máxima que o veículo percorreu.
- CarMass - o peso do veículo.
- IsRoadComplete - Completou a pista ou não.

Função de Fitness	$\text{Math.Pow}(\text{MaxDistance}, 2) + 400000 * \text{IsRoadComplete} + (5000 / \text{CarMass}) * 200$
-------------------	---

Tabela 8 - Função de Fitness para a experiência 2 do HillRoad.

Optamos por maximizar a distância percorrida pelo carro, e minimizar o peso do veículo.

A razão da distância percorrida é óbvia pelo facto de que quanto mais percorre, maior a probabilidade de este completar a pista. Quanto menor for a carga do veículo mais o carro anda, ou seja, nas subidas se o veículo fosse pesado este poderia não conseguir subir. Se o carro fosse leve e conseguisse percorrer bastante este tinha uma enorme probabilidade de concluir o percurso. Se o carro conseguisse completar a pista recebia cerca de 400000 de bônus.

Optamos por dar mais valor à distância, pelo facto de que queríamos que o melhor carro fosse sempre o que conseguisse percorrer maior distância, e que o peso deste também influencia-se um pouco.

Nesta função de fitness em comparação com a da 1ª implementação, ignoramos o parâmetro da velocidade média e colocamos o peso que este tem. Isto porque se o carro for demasiado rápido podia “despistar-se” e não conseguiria subir bem as rampas.

Apresentamos agora os resultados que obtemos acerca desta implementação:



Figura 18,19,20 - Experiências HillRoad.

## Notas e Conclusões:

Antes de mais, esta implementação conseguiu que um único veículo ultrapassasse a meta. Isto na experiência 2, apesar de não ser True's seguidos. 7/30 Gerações que tiveram pelo menos um carro a chegar à meta. A experiência 1 não conseguiu que um único veículo chegasse à meta. Na experiência 3 das últimas 4 gerações tiveram pelo menos um carro a chegar à meta.

Após analisarmos, esta implementação não é a mais correta para esta situação, visto que nas 3 experiências apenas nas 2 consegue ter pelo menos um "True". Tanto a experiência 2 e 3 fizeram com que soubéssemos que a implementação estava quase feita. Um fato positivo é que a Average Fitness aumenta ao longo das gerações.

### Iremos apresentar a 3ª implementação do HillRoad:

Para as configurações utilizamos as seguintes:

Mutação	Elitismo	CrossOver	Nº de Gerações
0.01	2	0.9	30

Tabela 7 - Configurações para a experiência 3 do HillRoad.

Para função de fitness teríamos de utilizar os seguintes parâmetros:

- MaxDistance - a distância máxima que o veículo percorreu.
- CarMass - o peso do veículo.
- IsRoadComplete - Completou a pista ou não.

Função de Fitness	$\text{Math.Pow}(\text{MaxDistance}, 2) + 400000 * \text{IsRoadComplete} + \text{Math.Pow}(5000 / \text{CarMass}, 3)$
-------------------	---

Tabela 8 - Função de Fitness para a experiência 3 do HillRoad.

Optamos por maximizar a distância percorrida pelo carro, e minimizar o peso do veículo.

A razão da distância percorrida é óbvia pelo facto de que quanto mais percorre, maior a probabilidade de este completar a pista. Quanto menor for a carga do veículo mais o carro anda, ou seja, nas subidas se o veículo fosse pesado este poderia não conseguir subir. Se o carro fosse leve e conseguisse percorrer bastante este tinha uma enorme probabilidade de concluir o percurso. Se o carro conseguisse completar a pista recebia cerca de 400000 de bônus.

Nesta função de fitness em comparação com a da 2ª implementação elevamos a minimização do peso do veículo a 3, permitindo que na roleta o veículo mais leve conseguiria ter uma probabilidade maior de ser escolhido, pensamos que esta implementação poderá funcionar bem melhor do que a 2ª implementação por causa disso.

Apresentamos agora os resultados que obtemos acerca desta implementação:

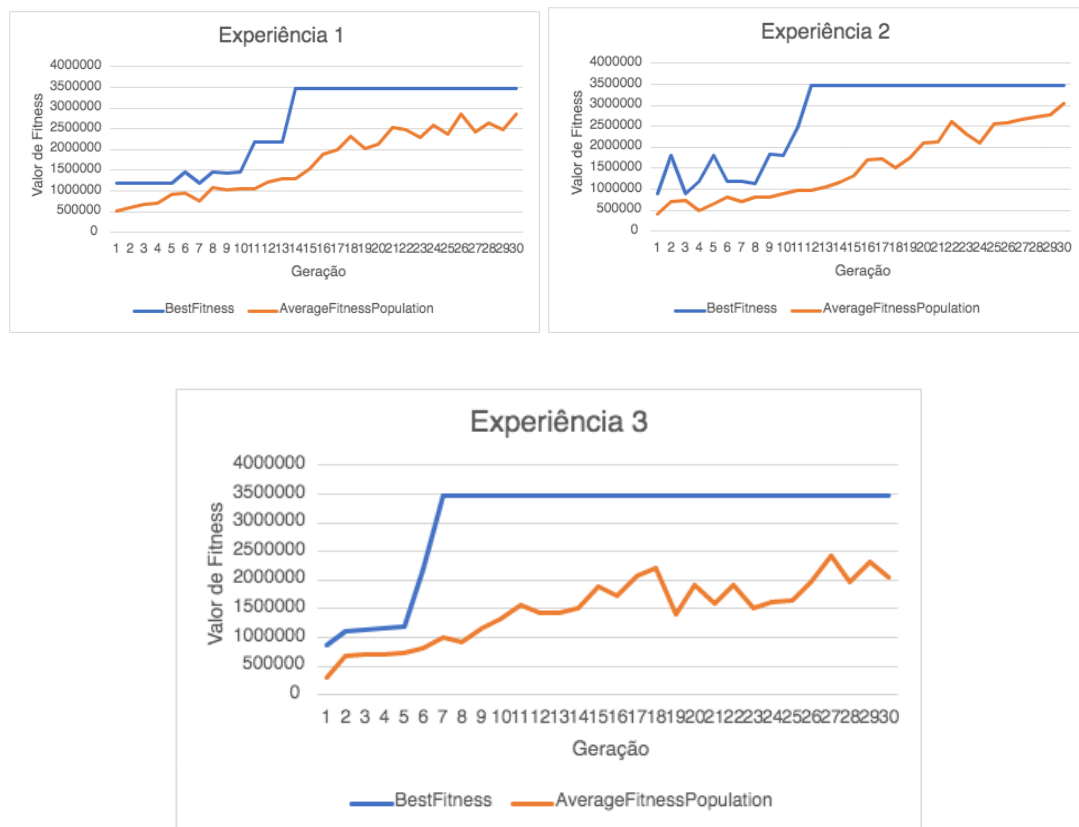


Figura 21,22,23 - Experiências HillRoad

## Notas e Conclusões:

Quando o BestFitness é semelhante a 3500000 implica que o veículo completou.

A função de fitness resolveu o problema sendo que na experiência 1, a partir da 14ª geração existiu sempre um veículo que completou a pista.

A experiência 2 a partir da geração 12 existiu sempre um veículo que completou a pista.

A experiência 3 a partir da geração 7 existiu sempre um veículo que completou a pista.

Tanto no BestFitness como no Average Fitness Population ocorre uma evolução, a experiência 3 encontrou mais facilmente um veículo a completar a meta em relação às outras duas experiências a razão pela qual isso acontece, é a probabilidade da mutação corresponder às expectativas e os veículos evoluírem logo em cada geração, no caso do veículo ficar mais frágil, este poderá passar para a geração seguinte as suas características que poderão ser mais fracas que as da geração anterior.

Concluindo após várias implementações conseguimos ter um resultado positivo tendo bastantes veículos a completar a pista.

Passaremos agora para o ObstacleRoad.

## ObstacleRoad:

Este percurso corresponde uma pista que contém vários blocos que ao longo do tempo ficam maiores e mais próximos uns dos outros.

No nosso ponto de vista, a mutação favorece se for um pouco alta, porque não achamos que com 30 gerações consigamos completar a meta.

Mas para comprovar iremos fazer um teste, este que será a 3ª e última implementação da pista Hill Road.

Apresentamos agora os resultados obtidos:



Figura 24,25,26 - Experiências ObstacleRoad

## Notas e Conclusões:

Antes de mais, esta implementação não conseguiu que um único veículo ultrapassasse a meta. Analisando o gráfico, nota-se uma evolução tanto no Fitness como na Average Fitness, mas insuficiente para realização com sucesso deste cenário.

Para este cenário, a mutação necessita de ser mais alta para que possam existir mais diferenças entre as gerações, permitindo que em menos gerações consigamos obter um veículo que tenha as características favoráveis para chegar à meta, portanto na próxima implementação iremos aumentar a mutação e claro, a função de fitness.

## 2ª implementação do cenário ObstacleRoad:

Para as configurações utilizamos as seguintes:

Mutação	Elitismo	CrossOver	Nº de Gerações
0.05	2	0.9	+/- 330

Tabela 7 - Configurações para a experiência 2 do ObstacleRoad.

Para função de fitness teríamos de utilizar os seguintes parâmetros:

- MaxDistance - a distância máxima que o veículo percorreu.
- CarMass - o peso do veículo.
- IsRoadComplete - Completou a pista ou não.
- MaxDistanceTime - Tempo máximo até acabar.

Função de Fitness	$((\text{float}) \text{Math.Pow}(\text{MaxDistance}, 4) + (\text{int}) \text{Math.Pow}((\text{MaxDistance} / (\text{MaxDistanceTime} + 1))^{5,3}) + 4000000 * \text{IsRoadComplete} + (\text{int}) \text{Math.Pow}((5000 / \text{CarMass})^{5,2}) / 100000;$
-------------------	--

Tabela 8 - Função de Fitness para a experiência 2 do ObstacleRoad.

Optamos por maximizar a distância percorrida pelo carro, e minimizar o peso do veículo.

A razão da distância percorrida é óbvia pelo facto de que quanto mais percorre, maior a probabilidade de este completar a pista. Quanto menor for a carga do veículo mais o carro anda, ou seja, conseguiria subir mais facilmente os blocos se este for mais leve. Quanto mais rápido for o veículo, mais será a probabilidade de este ultrapassar os blocos. Se o carro conseguisse completar a pista recebia cerca de 4000000 de bônus.

Optamos por dar mais valor à distância, pelo facto de que queríamos que o melhor carro fosse sempre o que conseguisse percorrer maior distância, e que o peso deste também influencia-se um pouco, como também a velocidade média.

Usamos também o Time Scale do Unity a 100.

Apresentamos agora os resultados que obtemos acerca desta implementação:

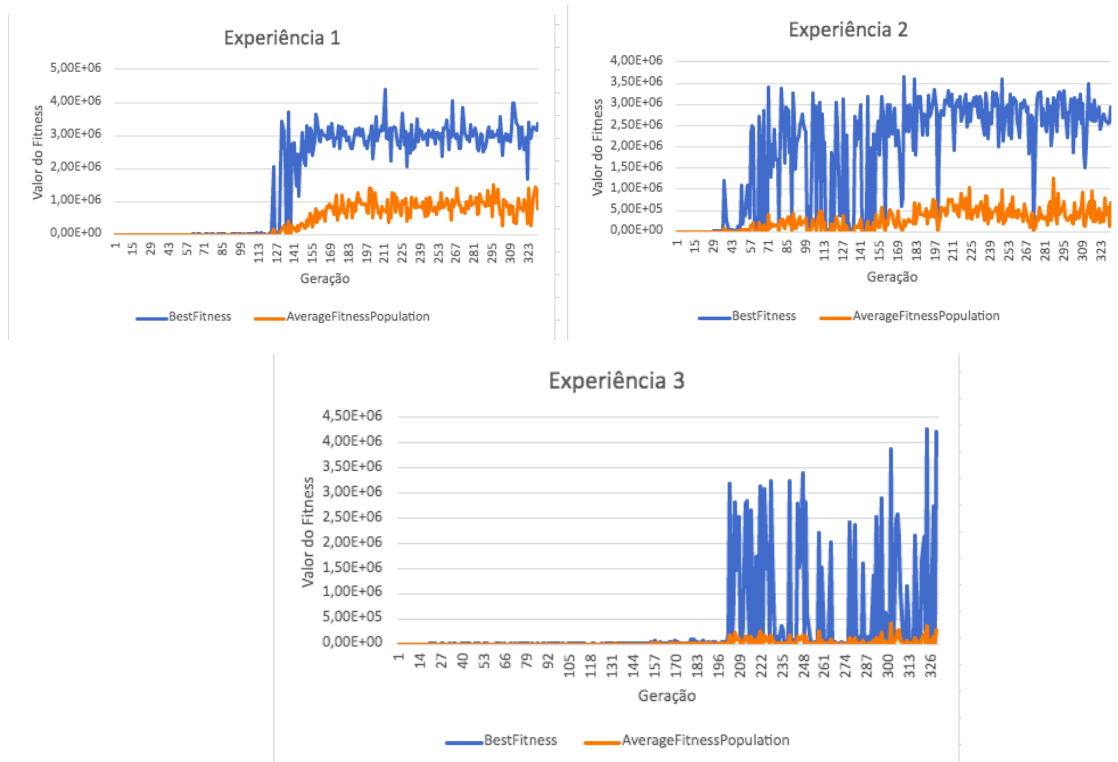


Figura 27,28,29 - Experiências ObstacleRoad

## Notas e Conclusões:

Apesar da função ter conseguido resolver o problema, não achamos que seja correto usar o Time Scale a 100, visto que influencia bastante no resultado, negativamente, porque se usarmos o valor deste mais baixo não conseguirá ultrapassar a meta.

Observando as tabelas, podemos ver que existe uma evolução do fitness, sendo que na experiência 1 a partir da geração 130 existiu sempre um veículo que completou a pista.

Na experiência 2 a partir da geração 130, e na experiência 3 completa a partir da 190 mais ou menos.

A Average Fitness Population ocorre uma evolução, principalmente na Experiência 1 e 2, sendo esse um dos nossos objetivos.

A implementação é melhor que a 1ª, sendo que a primeira apenas testamos para 30 gerações e esta para 330, achamos que faria mais sentido elevarmos o valor das gerações, pois após vários testes chegamos à conclusão que seria muito difícil o fazer com 30.

### 3ª implementação do cenário ObstacleRoad:

Para as configurações utilizamos as seguintes:

Mutação	Elitismo	CrossOver	Nº de Gerações
0.05	2	0.9	30

Tabela 7 - Configurações para a experiência 3 do ObstacleRoad.

Para função de fitness teríamos de utilizar os seguintes parâmetros:

- MaxDistance - a distância máxima que o veículo percorreu.
- CarMass - o peso do veículo.
- IsRoadComplete - Completou a pista ou não.
- NumberOfWheels - número de rodas.

Função de Fitness	$(\text{float}) \text{Math.Pow}((10000 / \text{CarMass}), 3 * \text{IsRoadComplete}) + (\text{float}) \text{Math.Pow}(\text{MaxDistance} * 10000, 3 * \text{IsRoadComplete}) + (\text{float}) \text{Math.Pow}(\text{NumberOfWheels} * 100, 3 * \text{IsRoadComplete})$
-------------------	--

Tabela 8 - Função de Fitness para a experiência 3 do ObstacleRoad.

Optamos por maximizar a distância percorrida pelo carro, minimizar o peso do veículo e maximizar o número de rodas que o veículo apresenta.

A razão da distância percorrida é óbvia pelo facto de que quanto mais percorre, maior a probabilidade de este completar a pista. Quanto menor for a carga do veículo mais o carro anda, ou seja, conseguiria subir mais facilmente os blocos se este for mais leve. Se o número de rodas fosse elevado a probabilidade de este nunca parar é elevada, optamos então por maximizar esse parâmetro. Se o carro conseguisse completar a pista as potências passavam de 3 para 4.

Usamos também o Time Scale do Unity a 10. O que permite que o carro passe com mais facilidade.



Apresentamos agora os resultados que obtemos acerca desta implementação:

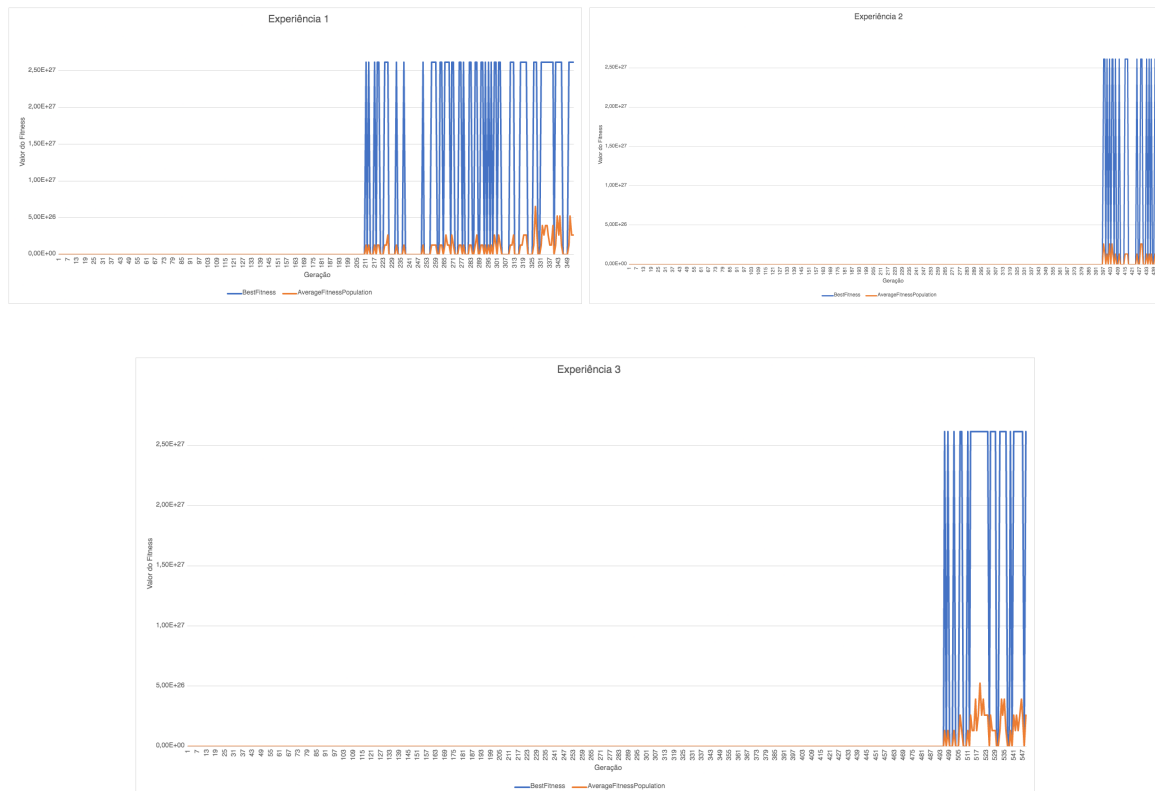


Figura 30,31,32 - Experiências ObstacleRoad

## Notas e Conclusões:

Analisando os resultados, conseguimos chegar a uma função de fitness que conseguisse resolver o cenário.

A Partir da Experiência 1, vemos que um veículo consegue bater na meta na geração 210 (aproximadamente), na experiência 2 consegue bater na meta na geração 397 (aproximadamente) e na experiência 3 consegue bater na meta na geração 490 (aproximadamente).

Um dado importante a formular, aumentamos a meta, nos prefabs pelo facto de esta ser demasiado baixa.

Concluimos que se uma função de fitness tiver umas características que maximizem o uso das rodas, da distância que percorre e minimize o peso obteremos um bom resultado tal como o esperado.

## Observações:

Na pasta Results do projeto é possível encontrar uma pasta “PastasRelatorio” sendo que dentro desta existem mais duas pastas:

- **Pastas\_Finais:** correspondem aos resultados das experiências que apresentamos ao longo do relatório, as 3 experiências para cada implementação.
- **Testes\_ :** corresponde a experiências que executamos, mas não guardamos as suas funções de fitness, pelo facto de que achávamos que como o seu resultado foi negativo, não necessitamos de guardar as pastas, sendo que as que estão nesta pasta são algumas das que realizamos na realidade, logo como estavam no “Lixo” não sabemos agora para que função de fitness e que configurações foi utilizada. Após conversarmos com o nosso professor das PL's, este pediu-nos para colocar numa pasta na mesma o que conseguimos e escrevermos no relatório.

## Conclusões:

Concluindo, o projeto foi realizado com sucesso, completamos os 3 cenários que nos foram propostos, sendo este o principal objetivo do projeto, adquirimos competências relacionadas com a análise, desenvolvimento, implementação e teste de agentes adaptativos.

Analisando as funções de fitness finais correspondem ao esperado, visto que com o gap road maximizamos a distância que este percorre, e maximizamos a velocidade média a que o veículo percorre. Hill road maximizamos a distância, minimizamos o peso do veículo permitindo a este conseguir subir as colinas e descer com segurança e com mais facilidade. ObstacleRoad optamos por maximizar a distância, e minimizar o peso, sendo que foi o cenário que nos demorou mais tempo, que realizamos mais testes mas que no fim foi compensado visto que conseguimos que veículos ultrapassem a meta.

Para todos as implementações finais de cada cenário tanto a BestFitness e a Average Fitness Population, evoluíram ao longo das gerações, sendo este mais um objetivo que queríamos alcançar e que foi alcançado.