



Universidade de Coimbra  
Faculdade de Ciências e Tecnologia Universidade de Coimbra  
Licenciatura em Engenharia Informática

# Fundamentos de Inteligência Artificial

## The Slow & The Calm: Darwin's Edition

**Projeto realizado por:**

Alexandre van Velze – 2019216618

João Carlos Silva – 2019216753

Sofia Alves – 2019227240

**Professor orientador:**

Nuno Lourenço

## 1. Introdução

No âmbito da cadeira de Fundamentos de Inteligência Artificial, este projeto visa estudar algoritmos evolucionários, com a construção de um veículo motorizado capaz de completar certos percursos. O algoritmo irá construir o veículo com base num conjunto de parâmetros (número de rodas, tamanho das rodas, etc). Para isso, foi utilizado um simulador com recurso ao software *Unity*. É através deste simulador que podemos testar a performance do veículo em vários cenários.

## 2. Meta 1 - Modelação e desenvolvimento do Algoritmo Genético

Para esta meta, foi requisitado a implementação de funcionalidades básicas do algoritmo evolucionário, sendo estas **parametrização, recombinação, mutação, seleção de pais, seleção de sobreviventes e aptidão**. Este algoritmo evolucionário tem como objetivo gerar agentes adaptativos (carros) que completem um dado percurso (*GapRoad* e *HillRoad*) constituídos por vários obstáculos (fendas e colinas, respetivamente). De seguida iremos descrever cada uma destas funcionalidades.

### Parametrização

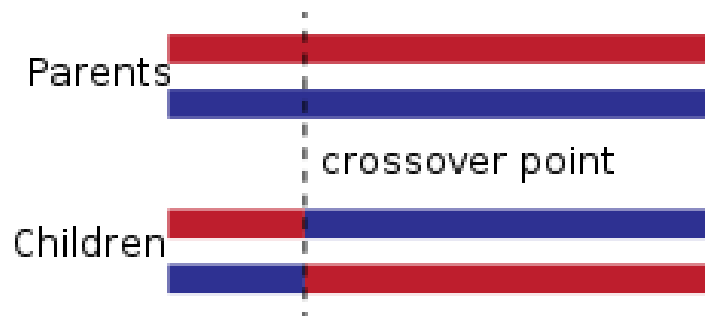
Neste componente são especificados os parâmetros do algoritmo evolucionário. Para este projeto foram utilizados os seguintes parâmetros:

- Probabilidade de Mutação
- Tamanho de Elitismo
- Tamanho do Torneio
- Probabilidade de Crossover
- Número de Gerações

Estes parâmetros visam gerir a forma como o algoritmo evolucionário se irá comportar.

### Recombinação

Nesta funcionalidade teremos a recombinação dos genes de dois dos pais (consoante uma probabilidade) que tenham sido obtidos através da “Seleção de Pais” obtendo assim 2 “filhos”. Visto que esta funcionalidade atende a um *Crossover Single-Point*, teremos apenas um ponto, obtido aleatoriamente, onde a recombinação acontece, isto é, até esse certo ponto teremos genes do 1º pai e a partir deste ponto teremos genes que pertenceram ao 2º pai para um dos “filhos”, o outro será precisamente o contrário, genes do 2º pai até certo ponto e genes do 1º pai a partir desse mesmo ponto. Isto resultará numa mistura de genes dos dois pais criando assim dois novos agentes resultante dos “pais”.



*Figura 1 – Representação da recombinação genética*

## Mutação

A mutação irá ser aplicada à população resultante do processo de recombinação. Esta consiste em operações de *bit-flip* (alterar bits a 1 para 0 e vice-versa) sobre os bits dos cromossomos, dependendo de uma dada probabilidade e de um valor aleatório associado a cada um deles (*RandomizationProvider*) (Figura 2). Este método denomina-se por single point mutation [1].



*Figura 2 – Representação da Mutação*

## Seleção de Pais

Para a seleção de pais, é selecionado, de forma aleatória, um conjunto de elementos de um dado tamanho (*tournamentSize*, definido pelo utilizador, que irão ser posteriormente comparados, com o objetivo de encontrar o melhor agente dentro deste grupo. Para tal, faz-se uma comparação da aptidão de cada um destes agentes, sendo selecionado aquele com maior *fitness*. Estes irão ser adicionados a uma lista de pais, que será depois utilizada para encontrar o par de pais mais apto para a próxima geração (elitismo). O parâmetro *tournamentSize* é de grande relevância, uma vez que quanto maior o tamanho do conjunto de agentes escolhidos maior a probabilidade de achar um indivíduo com um *fitness* favorável, garantindo assim a escolha de pais cada vez mais aptos para a próxima geração.

## Seleção de Sobreviventes

Nesta etapa serão selecionados os agentes que passam para a próxima geração. Para tal, começa-se por organizar a atual população por ordem decrescente de *fitness*. De seguida seleciona-se o número de agentes especificados na **Parametrização**, ou seja, o *eliteSize*.

## Aptidão

O objetivo da função de aptidão (fitness) é representar os requisitos que a população deve seguir. É a parte mais importante de qualquer algoritmo genético. Através desta função vamos avaliar cada indivíduo da população de forma a saber a sua posição em relação à solução desejada.

No momento da avaliação de um indivíduo, obtemos um valor que corresponde à qualidade do mesmo. Este valor será maior quanto melhor for a performance do indivíduo.

É fundamental que esta função esteja bem definida de forma a evitar excluir indivíduos que representem uma boa solução, o que irá posteriormente influenciar a evolução do algoritmo. De entre as funções testadas para esta meta, a que demonstrou melhores resultados foi:

$$fitness = 0.3 * MaxDistance + 100 * IsRoadComplete$$

Os pesos atribuídos aos fatores escolhidos irão dar uma maior importância ao fator *IsRoadComplete* uma vez que o objetivo da experiência consiste na geração de agentes que terminem o percurso. Como este fator apenas poderá ter os valores de 0 ou 1, concluiu-se, depois de vários testes, que seria mais favorável este ter um peso de 100. O *MaxDistance* permite, numa primeira fase, encontrar agentes que mais se aproximem da solução, caso nenhum consiga completar o percurso, uma vez que aqueles que quanto mais distância percorrerem, mais perto estão de cumprir o objetivo.

Deixámos de fora os parâmetros: *MaxDistanceTime*, *MaxVelocity*, *NumberOfWheels* e *CarMass*.

- *MaxDistanceTime*: achamos que o tempo que o agente demora a percorrer o percurso (*GapRoad*) não influencia em nada a sua performance ao longo do percurso dele.
- *MaxVelocity*: a velocidade irá tornar-se mais importante no cenário *HillRoad* (Meta 2). Para este cenário, apenas seria relevante a velocidade média.
- *NumberOfWheels* e *CarMass*: em parte estes dois parâmetros estão relacionados, porque quanto mais rodas o agente tiver maior será o seu peso, mas também não vimos grande utilidade em distinguir os agentes segundo estes elementos.

### 3. Meta 2 – Experimentação e análise

Neste capítulo foi realizado um conjunto de experiências de modo a testar várias funções de fitness e entender qual seria o melhor conjunto de parâmetros para a mesma, de modo a conseguir solucionar os percursos inicialmente propostos. Para fazer esta análise foi utilizado principalmente o cenário *GapRoad*. Posteriormente, após a avaliação dos parâmetros do algoritmo genético, foram realizadas experiências no cenário *HillRoad*.

#### Parâmetros do Algoritmo Evolucionário

Como referido, foi efetuado um conjunto de experiências sobre o cenário *GapRoad*, que consistiu na atribuição de diferentes valores a cada um dos parâmetros do algoritmo evolucionário. Os parâmetros utilizados em cada experiência estão descritos na seguinte tabela:

	Mutação	Elitismo	Torneio	Crossover	Número de gerações
Experiência 1	0.05	0	5	0.9	30
Experiência 2	0.2				
Experiência 3	0.05	2			
Experiência 4	0.2				
Experiência 5	0.05		2		

*Figura 3 – Parâmetros para cada experiência*

Dada a natureza estocástica dos algoritmos evolucionários, cada experiência foi executada 3 vezes de modo a se conseguir tirar conclusões de forma mais precisa e determinar assim qual o melhor conjunto de parâmetros para futuramente ser utilizado na procura de um veículo capaz de completar os cenários.

#### Análise teórica das experiências

Antes da realização de qualquer experiência, iremos formular que resultados serão esperados em cada uma delas.

Para a **experiência 1**, ao não possuir qualquer valor para o elitismo, não serão selecionados quaisquer agentes da geração atual que sejam mais aptos para a próxima geração. Deste modo, a evolução dos indivíduos irá ser prejudicada, pelo que não poderão ser gerados novos veículos com base nos melhores agentes. A probabilidade de mutação ao ser reduzida poderá ser, de certa forma, benéfica, visto que se forem encontrados carros que terminem um dado percurso, a probabilidade de surgirem filhos que fujam ao progenitor será também pequena. No entanto, se não encontrarmos indivíduos que consigam terminar o percurso, esta reduzida probabilidade de mutação também irá dificultar o processo de evolução, já que os indivíduos das próximas gerações irão ser gerados através dos mesmos. Deste modo, o efeito da probabilidade da mutação irá depender sempre muito do tipo de genótipos inicialmente criados, podendo ser benéfico que esta seja pequena caso o genótipo seja favorável, ou alta, caso contrário.

Quanto à **experiência 2**, possuímos também um valor nulo para o elitismo, no entanto, a probabilidade de mutação é mais elevada, 0.2, o que irá originar uma maior alteração nos genótipos, propiciando gerações mais diversificadas. Isto poderá fazer com que, caso sejam encontrados indivíduos que terminem o percurso, os indivíduos da próxima geração sofram demasiadas mutações e percam as características que potenciaram a conclusão do percurso pelos seus pais. Deste modo, espera-se que a experiência 1 origine melhores resultados que a experiência 2.

Na **experiência 3**, possuímos a mesma probabilidade de mutação usada na experiência 1, mas desta vez o número de agentes eleito para a próxima geração irá ser 2. Estes agentes irão então potenciar as gerações seguintes, já que este par é eleito consoante os melhores indivíduos de um dado conjunto, permitindo a geração de outros agentes com base nestes dois pais, herdando assim as suas características.

De seguida, temos a **experiência 4**. Nesta experiência temos outra vez o número de pais eleito de 2 e uma mutação de 0.2 (sendo que o torneio mantém-se igual a 5). Com o aumento da probabilidade de mutação, a probabilidade de alterar progenitores benéficos para a experiência aumentará também, mas, por outro lado, a probabilidade de alterar progenitores com características menos favoráveis irá também ser superior, tal como referido anteriormente. Com base nestas observações, espera-se que as experiências 3 e 4 sejam as que possuam melhores resultados.

Por último, temos a **experiência 5**. Comparando com a experiência anterior, é alterada a probabilidade de mutação, que passa a ser de 0.05, e o tamanho do torneio, que passa a ser 2. Aqui o fator que mais influenciará em relação às outras experiências será, naturalmente, o tamanho do torneio. Com tamanho de torneio 2 a probabilidade de achar um indivíduo mais apto irá diminuir abruptamente, visto que iremos estar a analisar e a comparar um número muito reduzido de agentes. Entre os 16 veículos de cada geração, estamos a escolher de forma aleatória apenas 2 deles para poder realizar o elitismo, sendo que os pais da próxima geração irão ser escolhidos de forma aleatória. Deste modo, iremos ter um processo de evolução dos indivíduos pouco eficiente, já que estes irão evoluir sempre de agentes escolhidos aleatoriamente.

## **Análise após a realização das experiências**

Iremos de seguida proceder à análise de cada experiência após a sua testagem no cenário *GapRoad* e tentar perceber se os resultados obtidos estão de acordo com os esperados.

### **Experiência 1:**

Com base nos resultados obtidos, chegamos a um consenso entre aquilo que era esperado e o que aconteceu para esta experiência. A falta de elitismo é de facto notória, como referido anteriormente não teremos pontos de referência para gerações futuras e isso prejudicará o algoritmo a manter estabilidade na procura de agentes que terminem os percursos, isto é, haver algum tipo de evolução de geração para geração.

De seguida apresentamos os gráficos dos resultados obtidos e que comprovam o que foi acima descrito:

### EXPERIÊNCIA 1 - BEST FITNESS

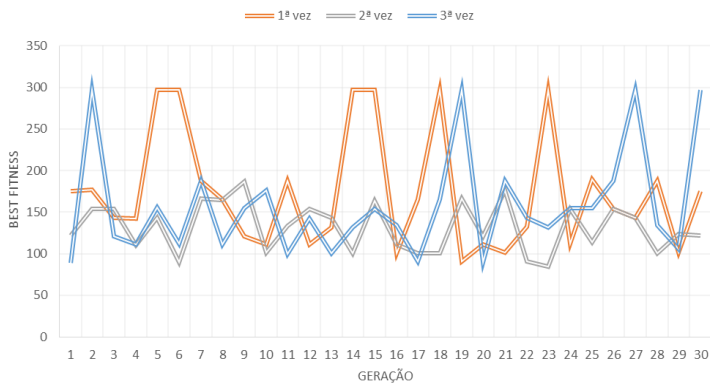


Figura 4 – Best Fitness para experiência 1

### EXPERIÊNCIA 1 - POP AVERAGE FITNESS

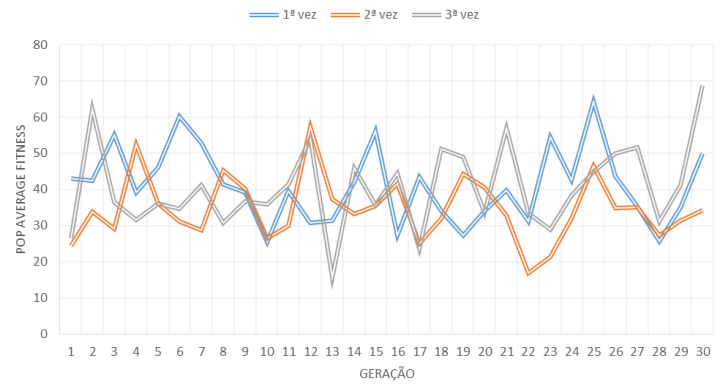


Figura 5 – Média de fitness para experiência 1

Temos então uma grande instabilidade entre os melhores indivíduos de cada geração onde a evolução será de certo modo aleatória. De notar também as baixas médias de fitness, comprovando assim que esta experiência não seria de todo a mais aconselhada para o objetivo pretendido.

### Experiência 2:

Nesta experiência vemos também uma grande instabilidade nos resultados. No entanto, comparando com a experiência anterior, obtemos um menor número de carros capazes de chegar ao fim do percurso. Tal poderá dever-se ao facto de termos uma maior probabilidade de mutação, o que faz com que os carros se alterem mais, não herdando tantas características dos seus pais. Para além disso, podemos observar uma menor média de fitness. Tal como esperado, a experiência 1 é melhor que a 2.

### EXPERIÊNCIA 2 - BEST FITNESS

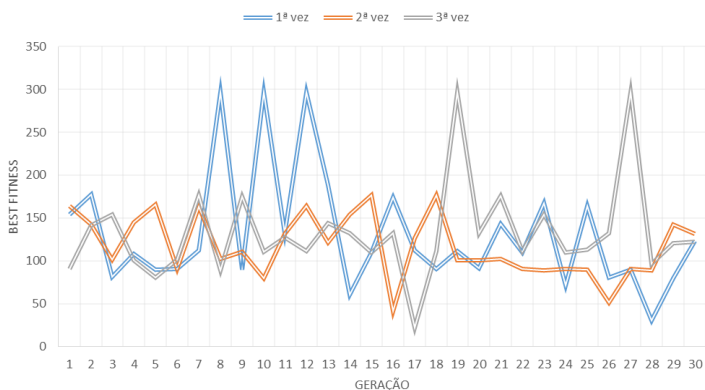


Figura 6 – Best Fitness para experiência 2

### EXPERIÊNCIA 2 - POP AVERAGE FITNESS

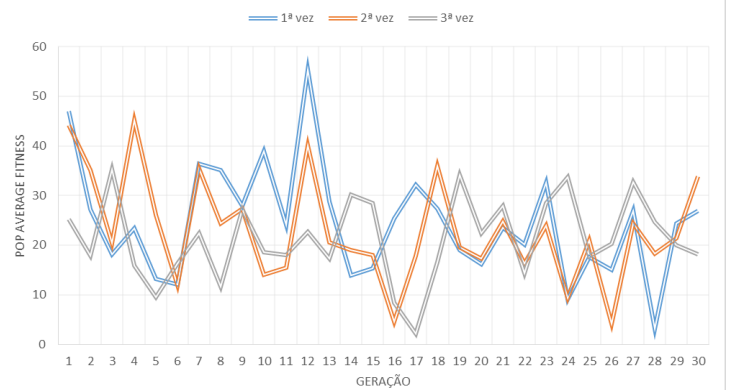
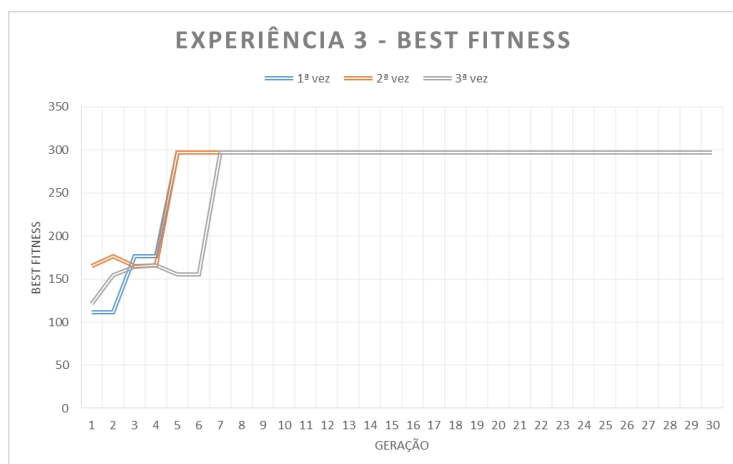


Figura 7 – Média de fitness para experiência 2

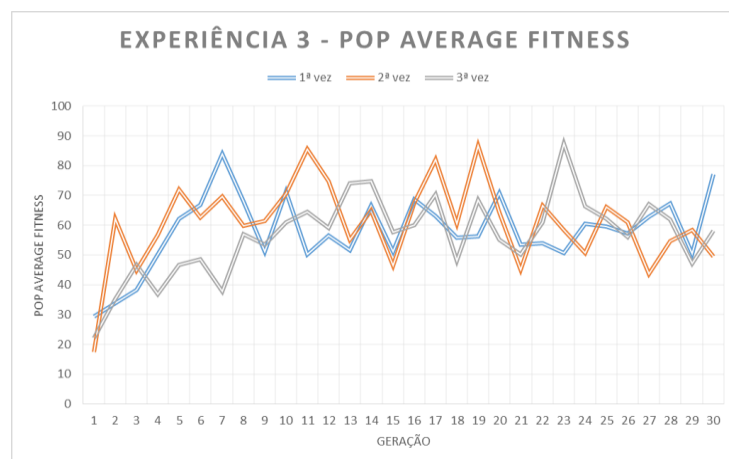
### Experiência 3:

Na experiência 3 vemos uma estabilidade muito maior na evolução dos veículos, sendo que, numa fase inicial, está-se à procura de um veículo que consiga terminar o percurso, melhorando gradualmente o seu fitness ao longo das gerações. Esta melhoria deve-se ao facto de possuímos elitismo, comprovando que, ao possibilitarmos a escolha dos pais mais aptos, conseguimos gerar gerações cada vez melhores. A partir do momento que um veículo consegue acabar a corrida, todas as gerações seguintes também terminam.

Esta estabilidade poderá dever-se à probabilidade de mutação que, ao ser baixa, não irá alterar muito os veículos provenientes dos pais eleitos na geração anterior, garantindo assim, de certa forma, a sua conclusão e claro que também ao facto de ser usado elitismo. Com esta experiência também podemos observar que os veículos conseguem terminar o percurso relativamente cedo, começando a acabar na geração 6 para a primeira e segunda execução da experiência, e na geração 7, para a terceira execução. Para além disso, conseguimos obter médias de fitness ainda mais elevadas do que as duas experiências anteriores.



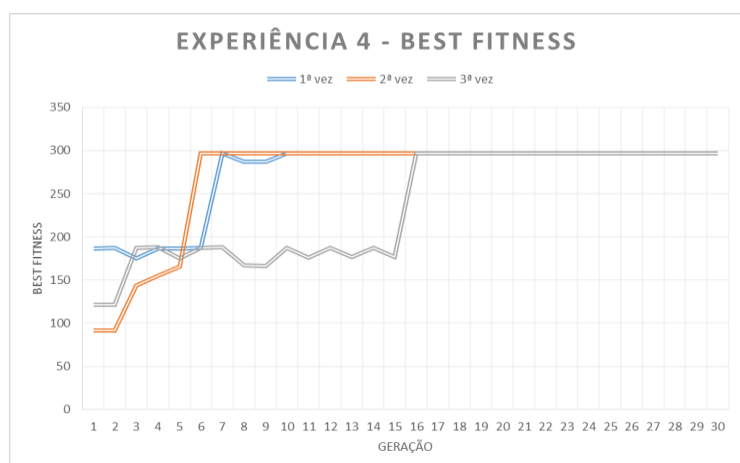
**Figura 8 – Best Fitness para experiência 3**



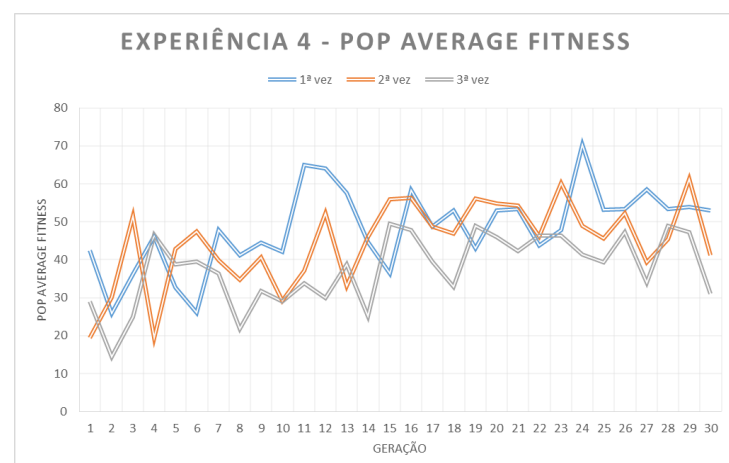
**Figura 9 – Média de fitness para experiência 3**

#### Experiência 4:

Na experiência 4 alteramos a probabilidade de mutação de 0.05 para 0.2 (em relação à experiência 3), fazendo com que as alterações aos offsprings sejam mais significativas. Ao usar uma probabilidade de mutação mais elevada vamos prevenir o algoritmo genético de convergir para uma solução mais rapidamente. Idealmente, queremos que o algoritmo chegue a uma solução o mais depressa possível. Assim, com este aumento na probabilidade observamos um aumento no número de gerações necessárias para alcançar uma solução. Além disso, podemos constatar que a média da fitness da população é mais inconsistente, causado pelo aumento de alterações nos descendentes.



**Figura 10 – Best Fitness para experiência 4**



**Figura 11 – Média de fitness para experiência 4**



### Experiência 5:

Finalmente falamos da **experiência 5**. Foi anteriormente referido que a dificuldade para encontrar pais que beneficiassem as gerações futuras aumentaria em relação às outras experiências, devido à diminuição do tamanho do torneio, algo que se torna notório principalmente na 3ª vez que a experiência foi executada. Em contradição observa-se que as médias de fitness na 1ª e 2ª execução são mais altas quando comparadas com algumas das experiências feitas anteriormente. Esta contradição surge devido à forma como este algoritmo funciona, ou seja, devido à forma como atende às probabilidades que estão “em jogo”.

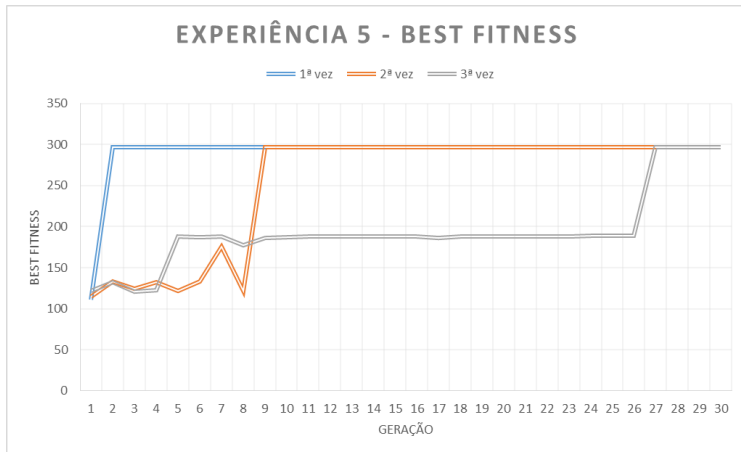


Figura 12 – Best Fitness para experiência 5

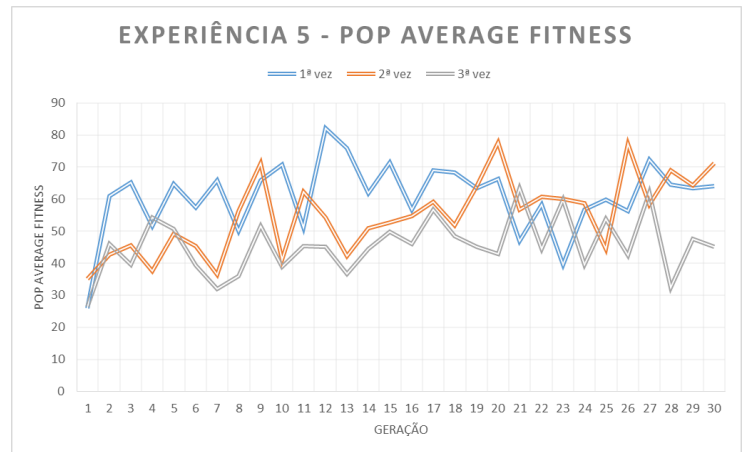


Figura 13 – Média de fitness para experiência 5

Após a avaliação de todas as experiências, concluímos que aquela que no geral apresentou os melhores resultados foi a experiência 3. Portanto será esta experiência a que possui o melhor conjunto de parâmetros e aquela que será usada para testar os cenários *GapRoad* e *HillRoad*.

**A nível teórico:** esta experiência apresentou o melhor valor de torneio de entre os disponíveis (tamanho 5), pois é o valor que implica maior probabilidade de gerar agentes benéficos para evolução do algoritmo; usa o elitismo (2), o que ajudará a potenciar gerações futuras; faz proveito de um bom valor de mutação (0.05) para fazer uma evolução mais progressiva e controlada, não deixando que os agentes divirjam muito das gerações anteriores.

**A nível experimental:** como já referido anteriormente é facilmente observado que esta experiência é a que mais rapidamente encontra a solução para o percurso testado; foi a que produziu das melhores médias de fitness por geração;

Por estas razões os parâmetros do nosso algoritmo evolucionário basear-se-ão nos da experiência 3.

## Aplicação dos resultados obtidos no *GapRoad*

A função de fitness mencionada na meta 1 é capaz de resolver o cenário. No entanto, é bastante trivial, sujeita a várias melhorias. Deste modo, fizemos uma análise dos parâmetros disponíveis e acabamos por chegar à seguinte função:

$$\begin{aligned} \text{fitness} = & 0.3 * \text{MaxDistance} \\ & + 100 * \text{IsRoadComplete} \\ & + 0.2 * \text{MaxDistance}/(\text{MaxDistanceTime} + 1) \end{aligned}$$

A diferença desta expressão para a anterior é a inclusão da velocidade média. Como o veículo irá passar por vários gaps na pista, esta componente serve para garantir que os veículos com uma maior velocidade acabem por obter uma fitness maior.

Pesos atribuídos:

- Damos uma maior importância naturalmente ao facto de o **agente terminar o percurso**, que é o objetivo do nosso algoritmo.
- Entre distância máxima percorrida e velocidade média, resolvemos dar um maior peso à **distância** visto que é sempre mais importante ter agentes que vão o mais longe possível e, portanto, tentamos premiá-los mais.

Deixámos de fora os restantes parâmetros pelas razões já descritas no capítulo da meta 1.

## Análise dos resultados para o cenário *GapRoad*

De seguida, voltámos a testar a nova expressão no cenário *GapRoad* e tentámos analisar o desenvolvimento do algoritmo ao longo deste cenário, o nosso foco foi nos parâmetros utilizados na expressão de fitness.

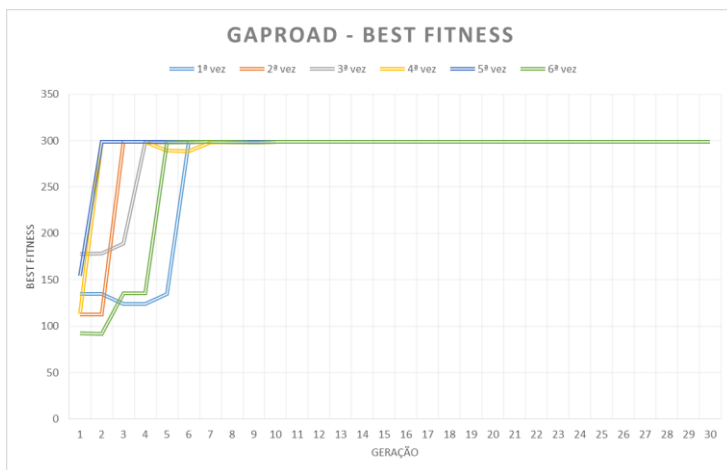


Figura 14 – Best Fitness para o cenário *GapRoad*

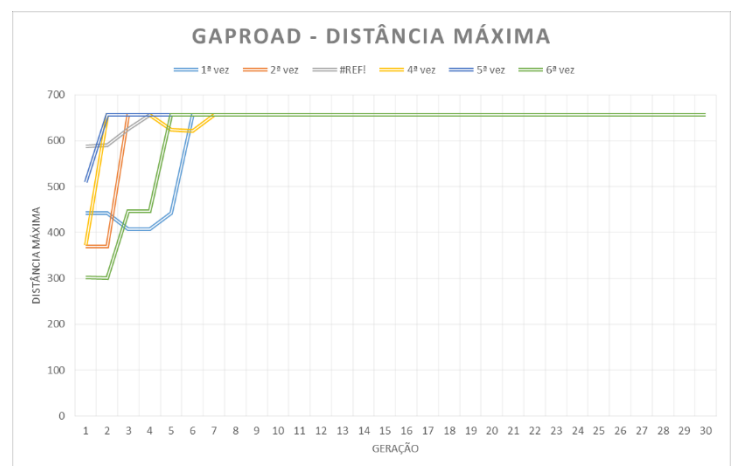
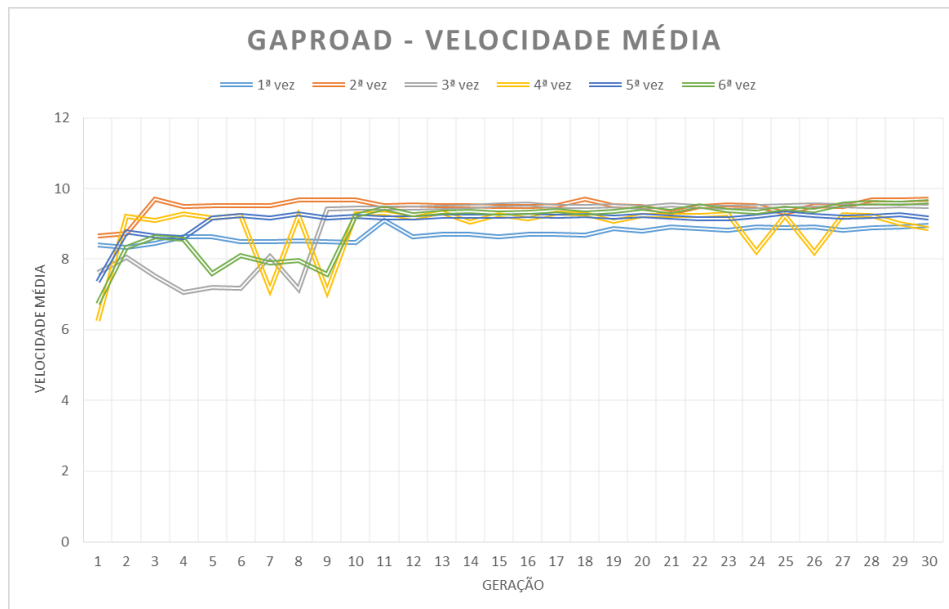


Figura 15 – Distância máxima para o cenário *GapRoad*

Com base nestes dois primeiros gráficos, rapidamente notamos que o algoritmo na grande maioria das vezes conseguiria descobrir uma solução rapidamente, algo que é benéfico, pois ajudaria a encontrar mais agentes capazes de resolver o cenário do *GapRoad* e por sua vez significaria que a evolução de geração para geração seria eficiente.

Falando da velocidade é de notar que estas mantêm-se superiores a 8 m/s na grande maioria dos casos o que daria alguns indícios para as velocidades ideais para os agentes resolverem o percurso.

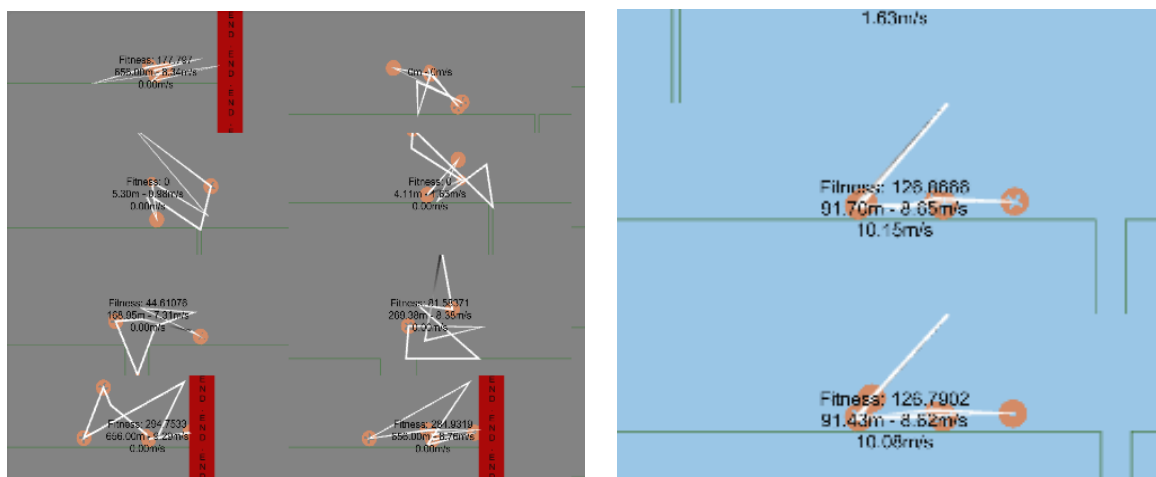


**Figura 16 – Velocidade média para o cenário GapRoad**

Com base nos resultados obtidos tentámos analisar qual o melhor tipo de veículo para completar este cenário, em termos de formato e características próprias. Acima podemos observar que grande parte dos agentes a terminarem o percurso de *GapRoad* terão uma velocidade superior a 8.7 / 8.8 m/s. Fisicamente estes casos apresentaram algumas características interessantes:

- O facto de serem normalmente longos, para não acabarem por cair ou ficarem presos em gaps maiores.
- Só serem longos não ajuda totalmente e, portanto, estes agentes foram evoluindo de forma a que a parte da frente fique ligeiramente levantada. Deste modo, a probabilidade de ficarem presos num gap torna-se muito mais pequena.
- Aliado a este último facto é costume ter-se uma espécie de “contrapeso” na parte de trás do agente para este se equilibrar após a passagem de um gap
- Possuem poucas rodas de apoio, normalmente 2/3 rodas, sendo até possível completar o percurso com apenas uma.

Estas são as características de um indivíduo perfeito para a conclusão do percurso *GapRoad*. De seguida, apresentam-se alguns exemplos:



**Figura 17 e 18 – Exemplos de carros que terminem o cenário GapRoad**

## Aplicação dos resultados obtidos no *HillRoad*

Após a avaliação das componentes do algoritmo genético e a análise das experiências, começamos por testar este cenário utilizando a melhor expressão de fitness encontrada no *GapRoad* e os parâmetros da experiência 3. Os resultados obtidos estão representados no gráfico e esquema seguintes:

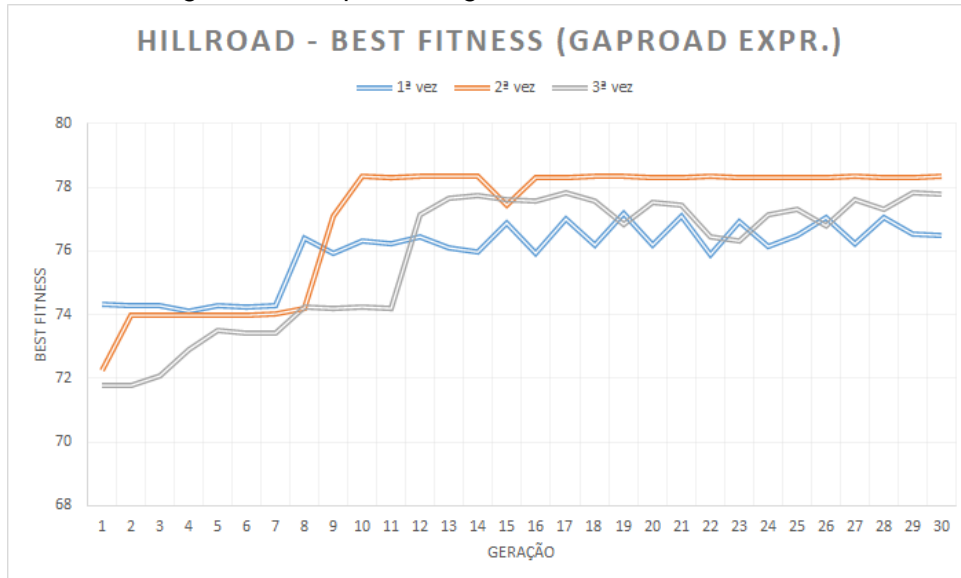


Figura 19 – Best Fitness para o cenário *HillRoad* (usando a expressão do cenário *GapRoad*)

	Geração																													
1ª vez																														
2ª vez																														
3ª vez																														

Figura 20 – Vista geral de quantos carros terminam o cenário *HillRoad* nesta condições

Rapidamente concluímos que esta expressão de fitness não iria ser suficiente para encontrar um veículo capaz de terminar o cenário. Percebe-se o porquê destes resultados, tal como no cenário de *GapRoad* era importante usar certas características, as mesmas poderão não constar nesta expressão de fitness visto que os cenários são completamente diferentes.

Assim, foi necessário pensar que parâmetros seriam os mais importantes para chegar ao fim.

Ter em conta a **distância máxima percorrida** e **IsRoadComplete** será sempre um ponto de partida, pois queremos sempre que o veículo termine o percurso, portanto é sempre útil olhar para os agentes que mais longe chegam neste caminho.

A **massa do carro** também é importante, devendo ser a mais pequena possível de forma a que o veículo necessite de uma velocidade inferior para conseguir subir - veículos com massa elevada implicam uma força maior para o seu deslocamento. Assim faz-se uma pequena penalização, de certa forma, a agentes com pesos maiores.

A **velocidade máxima** é também outro fator importante, pois um agente deverá tomar partido da aceleração que ganha na descida da primeira colina para conseguir ultrapassar a próxima. A altura em que é atingido um "vale" será então o ponto onde a velocidade é máxima, o que baterá certo pois é onde o veículo terá de exercer maior força para

começar a subir uma colina (consequentemente será o ponto onde a energia cinética também será maior, o que fornece maior movimento ao agente).

O **número de rodas** também deverá ser o menor possível para tentar manter a estabilidade do agente aquando a subida das colinas e de certa forma evitar que este tenha uma grande massa.

Portanto numa primeira análise a expressão de fitness utilizada foi a seguinte:

$$\begin{aligned} fitness &= 10000 * IsRoadComplete \\ &+ 40 * MaxDistance \\ &- 50 * CarMass \\ &- 100 * NumberOfWheels \\ &+ 200 * MaxVelocity \end{aligned}$$

Após alguns pequenos testes, chegámos à conclusão que o facto de estar a ser exigida uma penalização tão grande na massa e número de rodas do agente não estava a ser favorável. Assim, decidimos usar o inverso de ambos (com pesos menores na expressão) para diminuir a penalização e acrescentar a velocidade média.

Dado que o cenário é composto por várias subidas de diferentes declives, o veículo necessitará de uma certa velocidade mínima para poder conseguir alcançar o topo da colina. Ao incorporar a **velocidade média** na expressão de fitness, garantimos que os veículos com a velocidade necessária para alcançar futuras colinas são recompensados.

Assim a expressão fitness obtida foi a seguinte:

$$\begin{aligned} fitness &= 10000 * IsRoadComplete \\ &+ 40 * MaxDistance \\ &+ 5 * (1/CarMass) \\ &+ 15 * (1/NumberOfWheels) \\ &+ 200 * MaxVelocity \\ &+ 150 * MaxDistance/(MaxDistanceTime + 1) \end{aligned}$$

Pesos atribuídos:

- À semelhança do primeiro cenário voltamos a dar uma maior importância ao facto de o **agente conseguir terminar o cenário**.
- De seguida achámos melhor dar um peso de 200 à **velocidade máxima**, pois esta será um dos pontos fulcrais para que os nossos agentes tenham força suficiente para conseguir ultrapassar sempre as próximas colinas, sendo que a velocidade que o veículo toma na primeira colinha será deveras importante para manter o ritmo.
- Depois será **velocidade média** a ter um maior peso pois, como referido anteriormente, é importante que estes agentes consigam manter uma boa velocidade ao longo do percurso, assim aqueles que tiverem maiores valores neste quesito serão recompensados.
- Para a **distância máxima** o peso não será tão grande quanto os pesos atribuídos às velocidades pois os valores desta variam entre 0 e 10 m/s que multiplicados com 200 ou 150 nunca chegarão a ultrapassar os valores da MaxDistance multiplicada por 40. Deste modo a esta característica será sempre mais recompensada que as velocidades, pois um agente que percorreu uma maior distância terá indícios de estar mais perto da solução.

- Finalmente como já referido diminuámos a penalização para massas e números de rodas elevadas e achamos que de entre as características usadas estas deveriam um peso menor.

## Análise dos resultados para o cenário *HillRoad*

Utilizando esta expressão de fitness e os parâmetros da experiência 3, foi possível obter um veículo que terminou o cenário após 3 gerações. No entanto, ao realizar múltiplas execuções conclui-se que o algoritmo nem sempre encontra uma solução tão rapidamente (Figura 21), necessitando ainda de um número elevado de gerações.

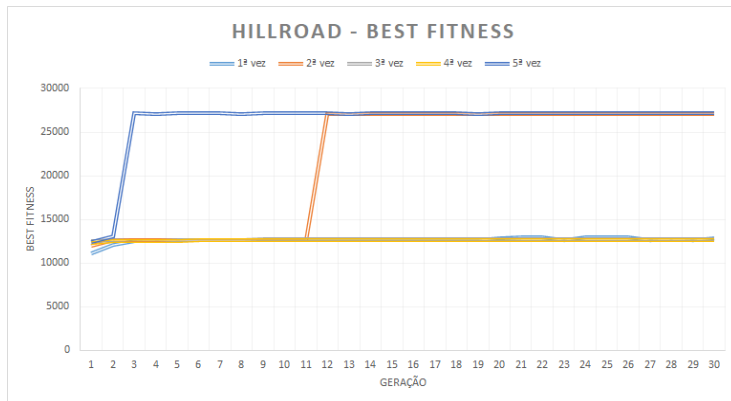


Figura 21 – Best Fitness para o cenário *HillRoad*

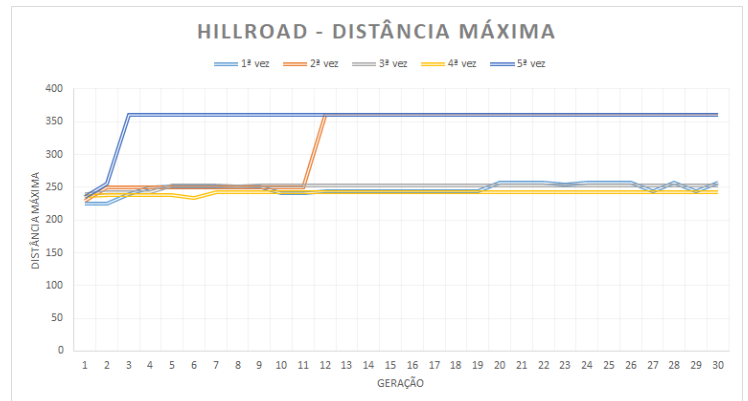


Figura 22 – Distância máxima para o cenário *HillRoad*

No gráfico da melhor fitness conseguimos observar experiências com resultados diversos. Conseguimos obter execuções que terminam o cenário dentro das 30 gerações, como a quinta, que conseguiu terminar na geração 3, e a segunda, que terminou após 12 gerações. Por outro lado, obtemos também várias execuções que não conseguiram terminar o percurso dentro das 30 execuções, tais como as restantes apresentadas no gráfico.

Observando o gráfico da distância máxima percorrida pelos veículos em cada geração (Figura 22), conseguimos concluir que este se assemelha ao gráfico do *Best Fitness*. Tal deve-se ao facto de este possuir um grande peso na expressão de fitness.

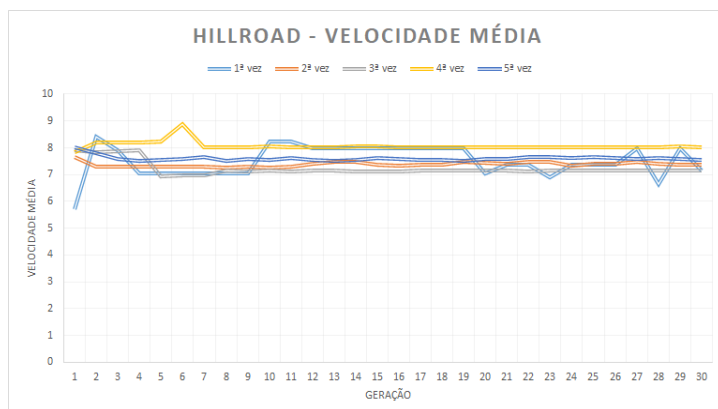


Figura 23 – Velocidade Média para o cenário *HillRoad*

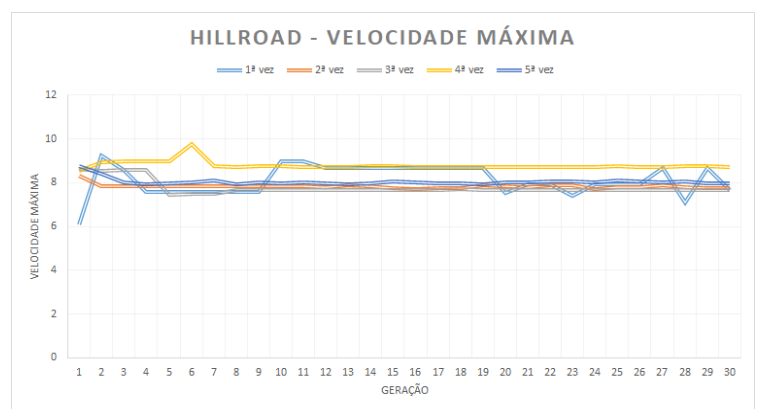


Figura 24 – Velocidade Máxima para o cenário *HillRoad*

A partir dos gráficos das velocidades média e máxima podemos observar a semelhança entre os mesmos, já que a velocidade máxima irá influenciar os valores da velocidade média. Para além disso, conseguimos ainda inferir que as experiências em que os veículos conseguem terminar o percurso possuem uma variação das velocidades mínima, permanecendo-se quase constantes ao longo das gerações. Isto poderá dever-se ao facto de, ao conseguir terminar um percurso, o carro com melhor fitness irá passar para as gerações seguintes devido ao elitismo, explicando assim o porquê de a velocidade não oscilar tanto nestas execuções.

De seguida mostramos uma vista geral das execuções usadas como análise:

	Geração																													
	1	2	3	4	5	6	7	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
1ª vez																														
2ª vez																														
3ª vez																														
4ª vez																														
5ª vez																														

**Figura 25 – Execuções usadas para análise**

No geral, podemos dizer que este cenário necessita de mais gerações para alcançar uma solução em comparação ao *GapRoad*. Apresentamos algumas das execuções onde isto verificado.

Geração													
1	2	3	4	5	6	...	167	168	169	170	...	199	200

Geração													
1	2	3	4	5	...	157	158	159	160	...	198	199	200

Geração																		
1	2	3	4	5	6	7	8	...	279	280	281	282	283	284	285	...	299	300

**Figura 26 – Execuções onde a solução demorou a ser encontrada**

Para finalizar este capítulo, passaremos para a análise dos agentes que normalmente terminam este cenário. Com base nos dados já previamente analisados reparamos que a velocidades tanto média como máxima variam entre os 7 a 9 m/s. Quanto à massa, esperar-se-ia que esta fosse mais determinante, mas conseguimos verificar que tanto agentes com valores entre 100 a 140 ou até mesmo 170 foram capazes de terminar o percurso. Passando para uma descrição mais física:

- À semelhança do que foi verificado no cenário anterior, os carros voltam a ter uma forma mais alongada o que os ajudará a conseguirem prender-se no topo da colina.
- Notamos que o número de rodas de apoio mantém-se entre 3 a 4 rodas como número ideal. Para juntar a este facto, é importante que estes agentes apresentem uma roda central para que estes não fiquem totalmente presos na colina e, assim, consigam transitar pelo seu topo.

Esta será a descrição para agentes que terminem o cenário HillRoad. Por fim apresentamos alguns exemplos:



*Figura 27 – Agente que termina após 3 gerações*



*Figura 28 – Agente que termina após 12 gerações*

## 4. Conclusão

Ao longo do desenvolvimento deste trabalho, foram adquiridos conhecimentos acerca de como um algoritmo evolucionário funciona, nomeadamente os seus parâmetros (mutação, crossover, elitismo, ...). Através deste algoritmo desenvolvemos competências em como formular uma função de fitness: atribuição de pesos para as diferentes características utilizadas, quais características escolher e como estas influenciam a evolução dos agentes.

Apercebemo-nos também da natureza estocástica deste tipo de algoritmo, pelo que foi necessária a realização de várias experiências para chegar aos melhores parâmetros.

Finalmente, já na realização de experiências nos cenários propostos (*GapRoad* e *HillRoad*) notámos que por vezes para um agente terminar estes mesmos percursos dependerá mais da sua fisionomia do que das suas características (velocidade, massa, número de rodas, ...), algo que infelizmente com o nosso algoritmo evolucionário foi complicado de gerir.

## 5. Bibliografia

- [1] [https://en.wikipedia.org/wiki/Mutation\\_\(genetic\\_algorithm\)](https://en.wikipedia.org/wiki/Mutation_(genetic_algorithm))
- [2] [https://en.wikipedia.org/wiki/Crossover\\_\(genetic\\_algorithm\)](https://en.wikipedia.org/wiki/Crossover_(genetic_algorithm))
- [3] S. Russell, P. Norvig : Artificial Intelligence - A Modern Approach
- [4] A.E. Eiben, J.E. Smith : Introduction to Evolutionary Computing