



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Trabalho Prático n.º 2

The Slow & The Calm: Darwin's Edition

Fundamentos de Inteligência Artificial
Licenciatura em Engenharia Informática
2021/2022

Duarte Emanuel Ramos Meneses – 2019216949 – duartemeneses@student.dei.uc.pt - PL3

Patrícia Beatriz Silva Costa – 2019213995 – patriciacosta@student.dei.uc.pt - PL3

Índice

Introdução	3
Modelação e desenvolvimento	4
Explicação do código	4
Experimentação e análise	4
Testagem	4
GapRoad	5
Resultados	5
Análise de Resultados – Testagem em GapRoad	7
HillRoad	8
Análise de Resultados – HillRoad	15
Conclusão	17
Referências	17

Introdução

Hoje em dia a Inteligência Artificial está presente nas nossas vidas, mesmo sem darmos conta. Uma das matérias dessa área são os agentes evolucionários e é nisso que este trabalho prático assenta.

A partir de um algoritmo desenvolvido para o efeito, a nossa missão passou por fazer evoluir veículos para chegar ao final de dois cenários: um em que a estrada ia tendo buracos cada vez maiores e outro em que a estrada era composta por colinas.

Neste relatório fazemos uma breve introdução ao código e documentamos o nosso processo evolutivo de experimentação de funções de fitness capazes de resolver as duas cenas propostas.

A partir da análise de resultados é sempre possível saber se estamos próximos da função certa ou não e é isso que tentamos transmitir ao longo deste documento.

Modelação e desenvolvimento

Explicação do código

Para começar, optamos por implementar o pseudo-código fornecido nos locais devidos. Deste modo, desenvolvemos os *scripts* originais para que estes contivessem as funções necessárias ao correto funcionamento do código.

Ao correr a nossa solução, após cada geração, o código vai seleccionar os pais no *script Meta1/Tournament.cs*. Aqui, são analisadas porções aleatórias de população já testada, e, para cada porção, escolhe-se o melhor indivíduo de acordo com o valor proveniente da função de fitness por nós desenvolvida (explicado mais à frente). Deste modo, obtemos uma lista dos melhores indivíduos em cada porção, o que nos permite extrapolar para os melhores da última geração.

Tendo os pais seleccionados, necessitamos de recombinar os seus genótipos de forma a formar os filhos. Isto acontece no *script Meta1/SinglePointCrossover.cs*. Com os filhos criados, e para estes não ficarem exatamente como sendo uma combinação dos seus pais, o algoritmo altera aleatoriamente alguns valores do genótipo no *script Meta1/SinglePointMutation.cs*.

Sendo que nem só de filhos se constrói uma nova população, em *Meta1/Elitism.cs* são escolhidos alguns dos melhores casos testados nesta geração para estes serem testados na próxima.

Tal como referido acima, para calcularmos o valor de fitness (de assertividade) de um veículo, necessitamos de uma função para o efeito. Isso é definido em *CarFitness.cs*.

Experimentação e análise

Testagem

Foram-nos fornecidos alguns parâmetros a testar. Para cada conjunto de parâmetros, realizamos três experiências para evitar possíveis *outliers* e ajudar a identificar um padrão. Em todas elas o crossover foi de 0.9 e o número de gerações 30.

A nossa primeira tarefa foi descobrir quais os melhores parâmetros para, a partir daí, desenvolver várias funções de fitness. Para tal, optamos por utilizar o cenário GapRoad, visto ser o mais fácil de completar. Tendo os resultados das experiências para cada conjunto de parâmetros, conseguimos escolher qual o melhor para utilizar com as várias funções de fitness.

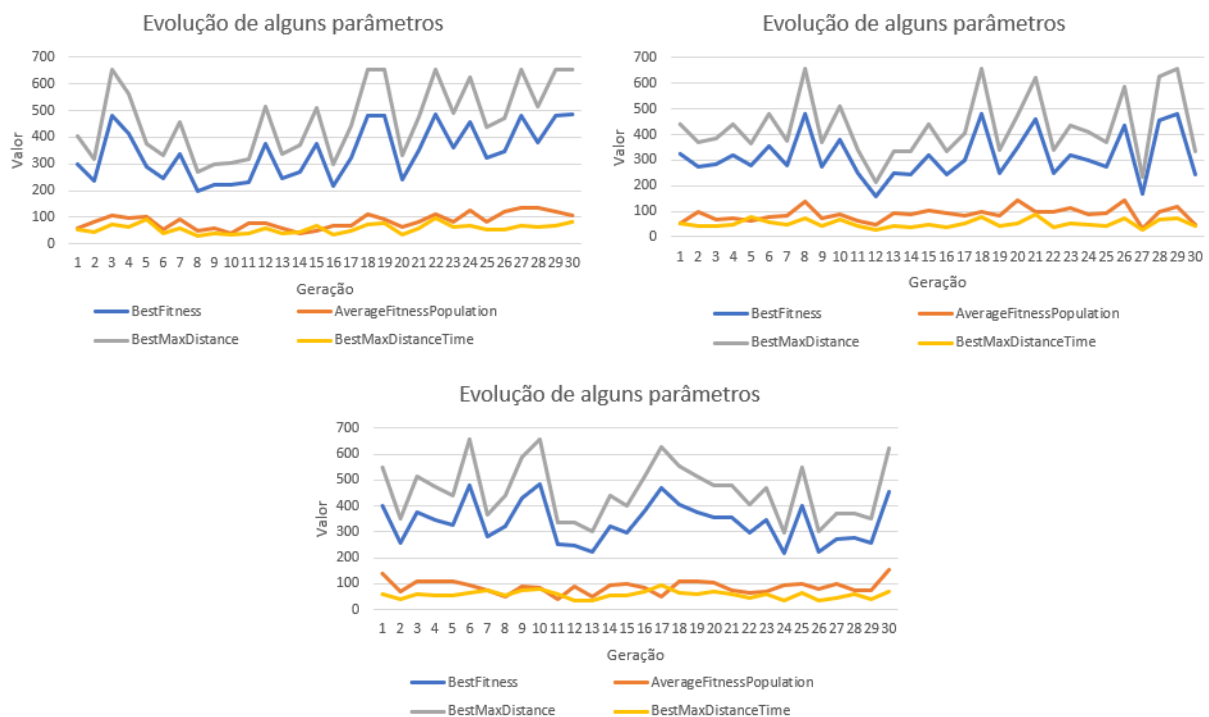
Numa função de fitness existem vários fatores que podem contribuir para o cálculo do valor necessário. Optamos, num momento inicial, por considerar a distância percorrida e o tempo necessário a percorrer tal distância. Decidimos isto pois, sendo o objetivo das cenas que os veículos cheguem ao fim, escolhendo os que percorrem maior distância, estamos mais próximos de estes ao evoluírem conseguirem atingir o objetivo. Para o fazerem no menor tempo possível, decidimos juntar estes dois atributos. Como a distância percorrida é, no nosso entender, mais importante, atribuímos-lhe um peso superior (0.7) ao invés dos 0.3 atribuído ao tempo.

GapRoad

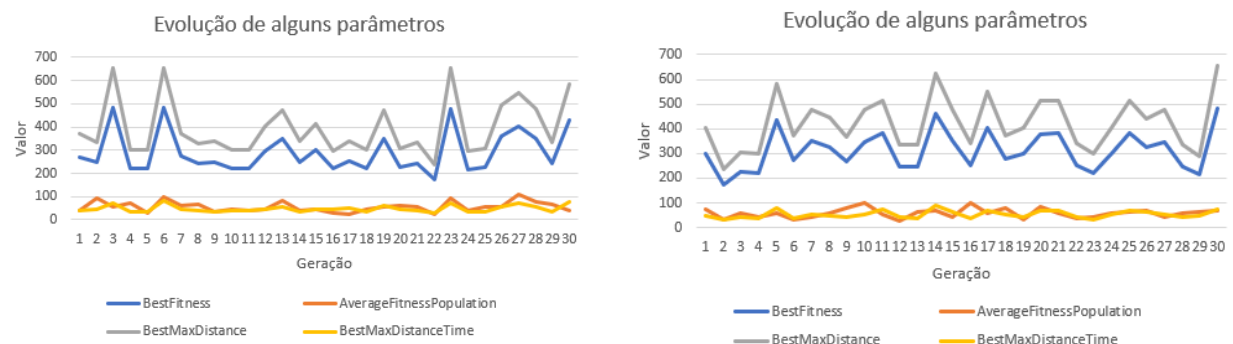
Este cenário consiste num plano raso que tem aberturas que vão aumentando de tamanho gradualmente. O objetivo é evoluir o veículo de modo a este conseguir atingir o final do percurso sem cair nem ficar preso nas aberturas.

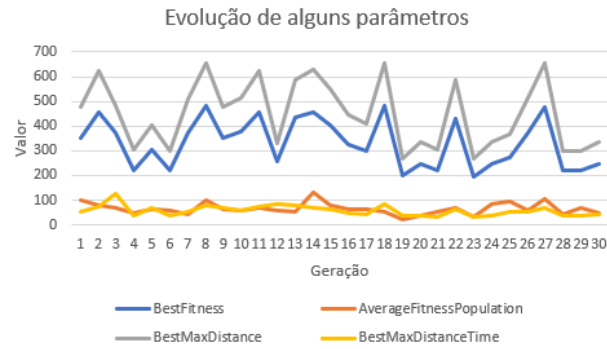
Resultados

Para uma probabilidade de mutação de 0.05, elitismo 0, torneio 5, os resultados foram os seguintes:

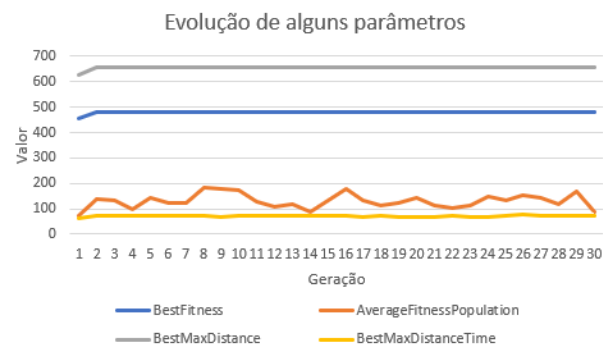
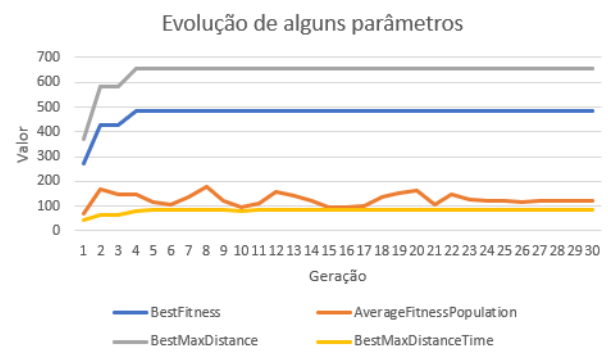
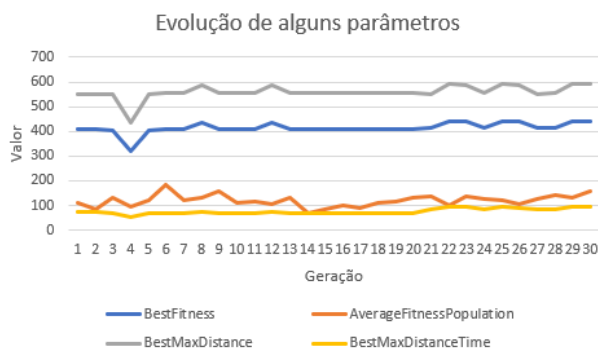


Para uma probabilidade de mutação de 0.2, elitismo 0, torneio 5, os resultados foram os seguintes:

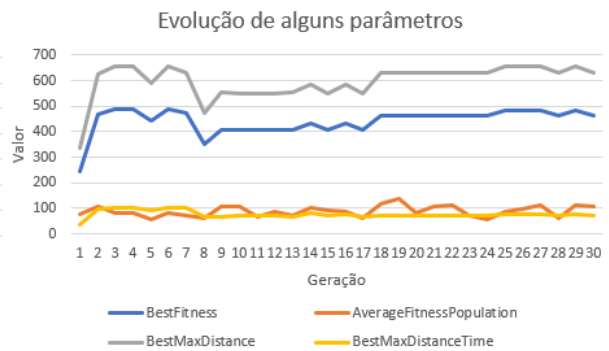
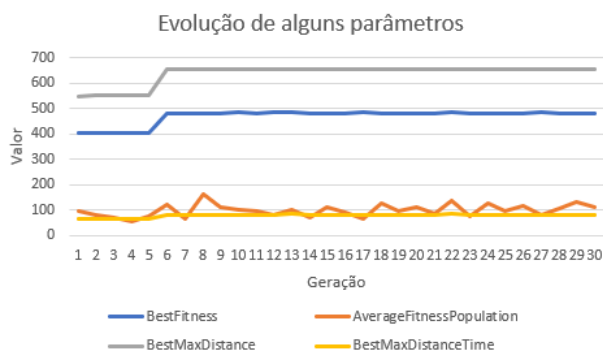


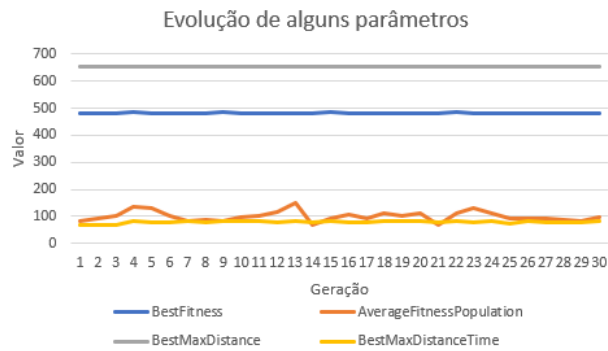


Para uma probabilidade de mutação de 0.05, elitismo 2, torneio 5, os resultados foram os seguintes:

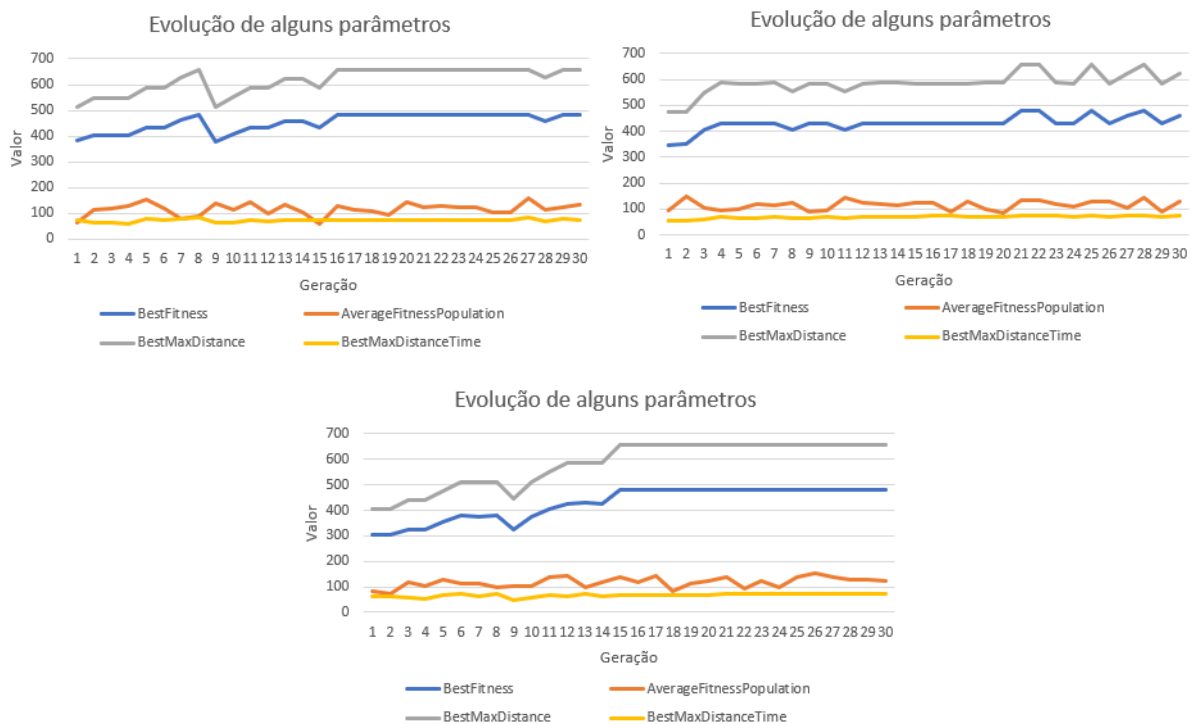


Para uma probabilidade de mutação de 0.2, elitismo 2, torneio 5, os resultados foram os seguintes:





Para uma probabilidade de mutação de 0.05, elitismo 2, torneio 2, os resultados foram os seguintes:



	Quantidade de veículos que chegaram ao fim			Média
Experiência 1	7	3	2	4
Experiência 2	3	1	3	2,334
Experiência 3	0	27	29	18,667
Experiência 4	25	11	30	22
Experiência 5	20	5	16	13,667

Análise de Resultados – Testagem em GapRoad

Tal como era espectável, a evolução do BestFitness acompanha a da BestMaxDistance e da BestMaxDistanceTime. Dizemos isto, uma vez que, tendo a função de fitness em conta estes dois parâmetros, era lógico que o resultado iria acompanhar a evolução destes. Como, presumivelmente, quanto mais distância percorrida pelo veículo mais tempo este demora no seu trajeto, a evolução destes dois valores é muito semelhante. Sendo que na função de fitness

por nós implementada, a distância percorrida tem um peso superior relativamente ao tempo, o BestFitness aproxima-se mais do valor da distância.

Em qualquer uma das experiências percebemos que existe um certo grau de aleatoriedade, uma vez que a AverageFitnessPopulation varia consoante a geração.

Dos resultados acima, conseguimos ainda extrair que, não tendo elitismo (os melhores da geração anterior não transitam para a nova), os resultados variam mais e não existe uma constância no número de veículos que chega ao fim.

Ainda na linha de raciocínio anterior, quanto menor o torneio, menor o número de melhores pais a gerar novos veículos. Deste modo, a probabilidade de acertar num que chegue ao fim reduz-se, pelo que a evolução é mais lenta.

Visualizando os gráficos acima percebemos também que existindo um maior valor de probabilidade de mutação do genótipo, as variações não são tão suaves. Isto acontece, pois, os veículos diferem mais dos da geração anterior.

Sendo o objetivo deste trabalho prático fazer com que os veículos cheguem ao fim do percurso, assumimos pelos resultados presentes na tabela acima que os melhores parâmetros de configuração são os da experiência 4. Deste modo, vamos utilizar estes para testar as próximas funções de fitness.

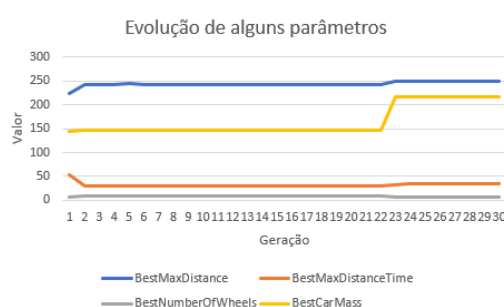
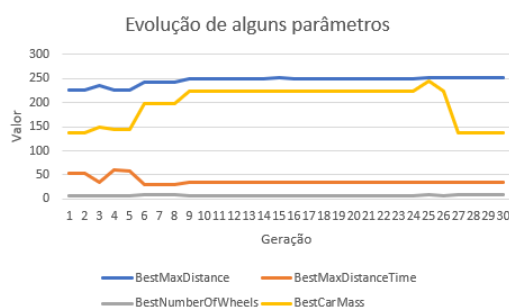
HillRoad

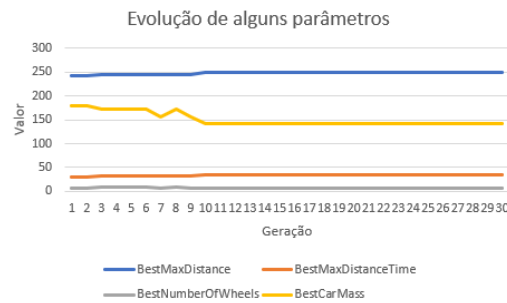
Pelos resultados visualizados anteriormente, ficou evidente que a função de fitness escolhida por nós resolveu o problema da cena Gap Road, já que conseguimos chegar ao final do percurso.

Deste modo, decidimos avançar para a resolução do problema presente na cena Hill Road, que consiste em chegar ao fim de um percurso com colinas. Decidimos, numa fase inicial, utilizar a função que nos tinha permitido chegar ao final do cenário anterior.

Os resultados foram os seguintes:

Nº Tentativa	Máxima Distância	Terminou?
1	249,478	Não
2	251,9311	Não
3	249,4924	Não



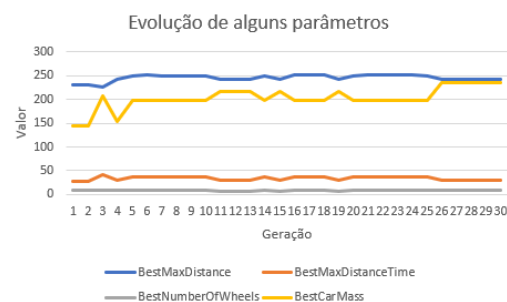
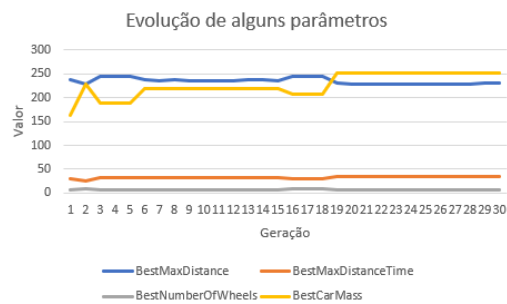
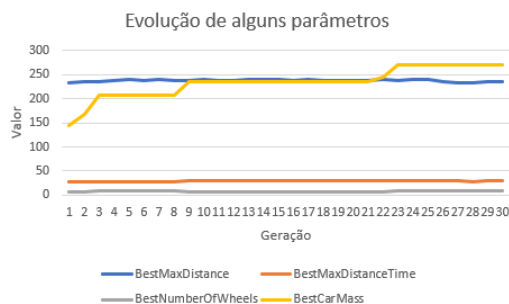


Com esta experiência, percebemos que esta função de fitness não serve para resolver tal cenário. Apercebemo-nos que ao subir uma das colinas, o veículo não chegava ao topo. Desconfiamos que isto acontecia uma vez que a massa não o permitia subir até ao fim. Deste modo, experimentamos utilizar uma função que envolvesse massa para ver se o veículo evoluía de modo a chegar ao fim do percurso. A função foi a seguinte:

$$\text{Fitness} = 0,7 * \text{distância percorrida} + 0,3 * \text{melhor massa}$$

Os resultados obtidos apresentam-se na tabela abaixo:

Nº Tentativa	Máxima Distância	Terminou?
1	241,4441	Não
2	245,194	Não
3	252,1479	Não



Com esta função, os resultados não foram muito distintos dos da experiência anterior. Mas... Por que razão isto aconteceu? Rapidamente percebemos que na variável melhor massa ficava a maior. Deste modo, não faz sentido adicionarmos (beneficiarmos) quem tem maior massa, mas sim penalizar (subtrair), visto que pretendemos que o veículo evolua e fique cada vez com menos massa para poder subir a colina.

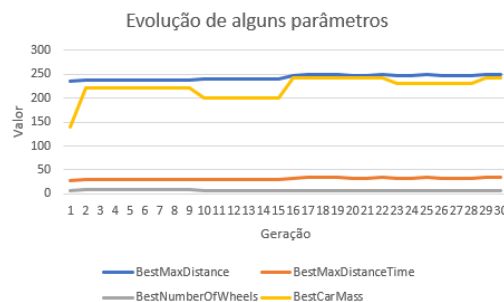
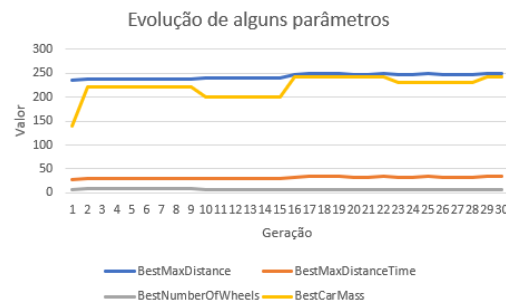
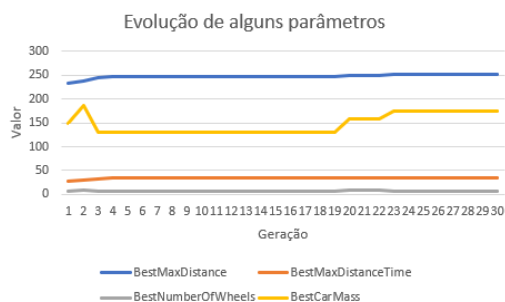
Antes de passarmos a uma nova função, decidimos testar outra teoria que tínhamos. Pensamos que ele poderia não estar a chegar ao topo da colina por chegar ao seu início com

velocidade baixa, o que, ao reduzir ainda mais na subida, o fazia não ter velocidade suficiente para chegar ao topo. Com isto, passamos à próxima função de fitness que inclui também a velocidade máxima atingida pelo veículo, sendo ela:

$$\text{Fitness} = 0,5 * \text{distância percorrida} + 0,5 * \text{velocidade máxima}$$

Os resultados obtidos apresentam-se na tabela abaixo:

Nº Tentativa	Máxima Distância	Terminou?
1	252,95	Não
2	250,6847	Não
3	247,856	Não



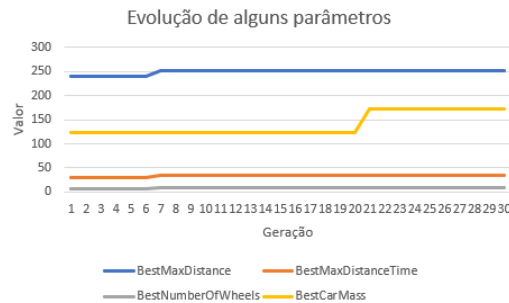
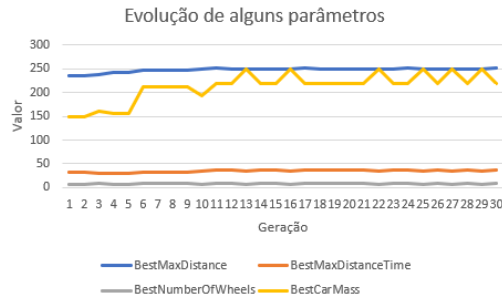
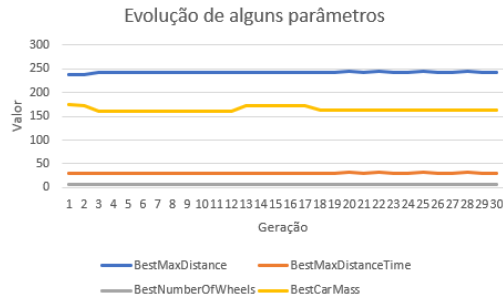
Uma vez mais, não conseguimos atingir o objetivo. Nesta fase das experiências pensamos que o problema poderia não estar apenas na velocidade máxima, mas sim na velocidade média. Tivemos este raciocínio uma vez que um veículo até pode atingir uma boa velocidade ao descer (por ter mais massa) mas, depois, ao subir, vai reduzir proporcionalmente ao que aumentou na descida. Ora, ao utilizar apenas a velocidade máxima, iríamos estar a beneficiar os veículos de maior massa que, depois nas subidas, iriam ficar para trás.

Com isto, passamos à nossa próxima função de fitness que já inclui essa questão da velocidade média. Para tal, dividimos a distância percorrida pelo tempo necessário a percorrer tal distância. Como muitos veículos nem chegam a arrancar, o tempo nestes casos é 0. Deste modo, para evitar divisões por 0, adicionamos 0.01 ao valor do tempo, ficando a função do seguinte modo:

$$\text{Fitness} = \text{distância percorrida} + \frac{\text{distância percorrida}}{\text{tempo demorado} + 0.01}$$

Os resultados foram:

Nº Tentativa	Máxima Distância	Terminou?
1	244,5756	Não
2	251,315	Não
3	253,0395	Não



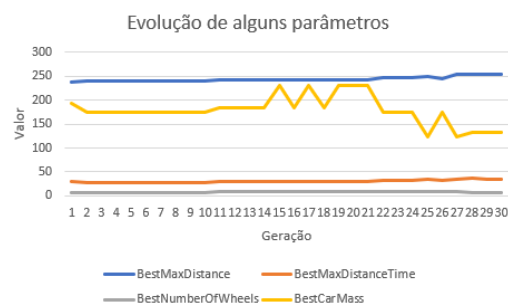
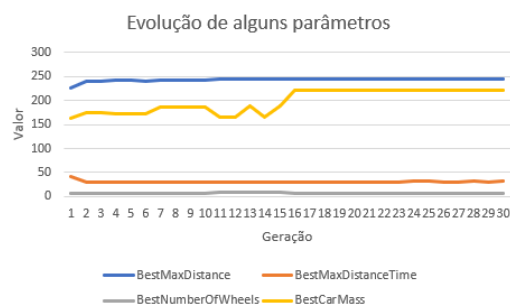
Ora, com estes resultados, embora o algoritmo utilizado seja estocástico, percebemos que houve uma melhoria na distância atingida.

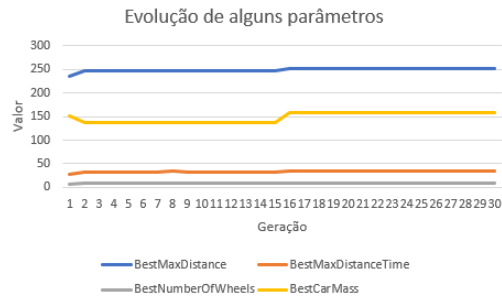
Com base nisto, decidimos voltar a experimentar esta função de fitness, agora atribuindo mais importância à distância percorrida, já que o objetivo é completar o percurso. A função por nós utilizada neste caso foi a seguinte:

$$\text{Fitness} = 3 * \text{distância percorrida} + \frac{\text{distância percorrida}}{\text{tempo demorado} + 0.01}$$

Os resultados os seguintes:

Nº Tentativa	Máxima Distância	Terminou?
1	245,9328	Não
2	254,6421	Não
3	252,1393	Não





Com esta mudança em relação à função anterior, os resultados não sofreram grande melhoria. Porém, continuou a ser das que apresentaram melhores resultados.

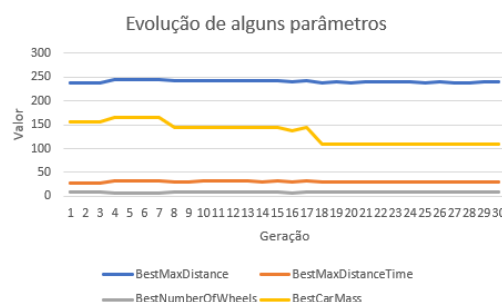
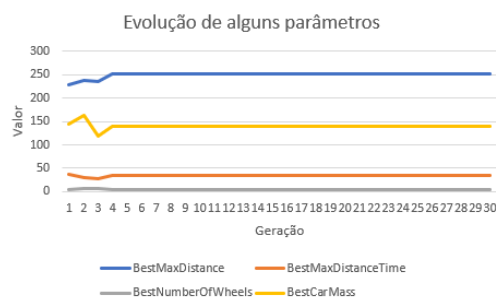
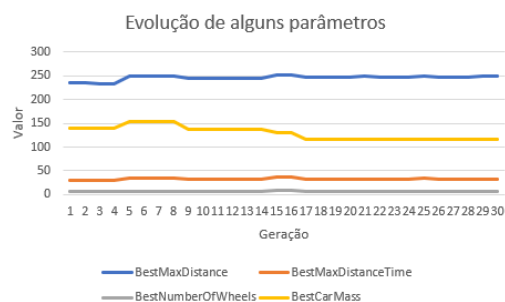
Tendo essa noção como base, passamos à próxima fase das nossas experiências. Juntamos agora os conhecimentos que obtivemos até aqui com as experiências passadas e decidimos experimentar a função anterior penalizando os veículos de maior massa. Deste modo, beneficiamos os que chegam mais longe com uma velocidade média superior (que já vimos resultar) e penalizamos os mais pesados (que, teoricamente, são piores a subir colinas).

Sendo que o objetivo é chegar ao final do trajeto, pensamos em dar mais importância (peso) à distância percorrida, pelo que função de fitness por nós utilizada foi a seguinte:

$$\text{Fitness} = 10 * \text{distância percorrida} + 5 * \frac{\text{distância percorrida}}{\text{tempo demorado} + 0.01} - 5 * \text{melhor massa}$$

Os resultados encontram-se abaixo:

Nº Tentativa	Máxima Distância	Terminou?
1	251,5947	Não
2	251,6801	Não
3	244,3386	Não



Analisando rapidamente os dados da tabela acima, fica evidente que esta foi a função de fitness que melhor evoluiu os veículos de modo a estes chegarem mais longe. Com isto,

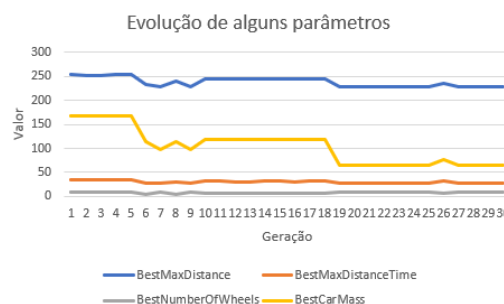
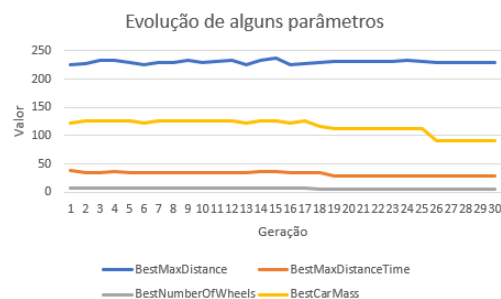
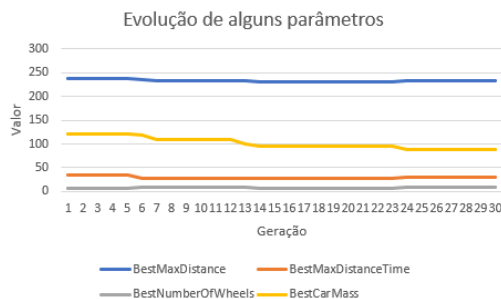
pensamos estar no caminho certo para encontrar a solução deste cenário. Como tal, optamos por, daqui em diante, utilizar funções de fitness muito semelhantes a esta, mudando apenas os pesos atribuídos a cada característica.

Sendo que, isoladamente, a função de fitness envolvendo a velocidade média obteve melhores resultados que a função envolvendo massa, decidimos aumentar o peso da velocidade, passando a função de fitness a ser:

$$\text{Fitness} = 15 * \text{distância percorrida} + 7 * \frac{\text{distância percorrida}}{\text{tempo demorado} + 0.01} - 5 * \text{melhor massa}$$

Com esta função, obtivemos:

Nº Tentativa	Máxima Distância	Terminou?
1	239,1055	Não
2	236,9385	Não
3	253,61	Não



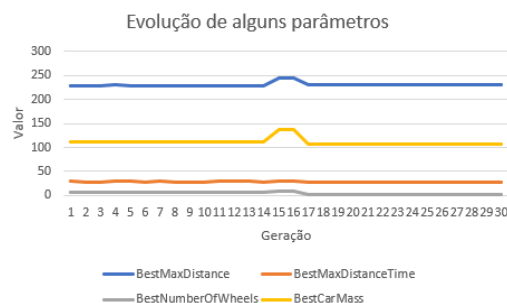
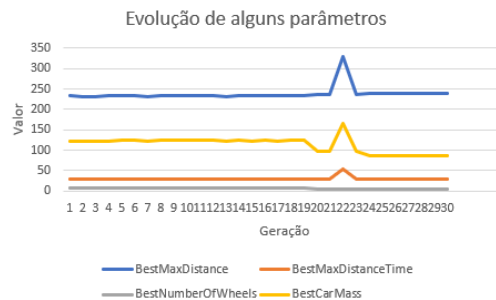
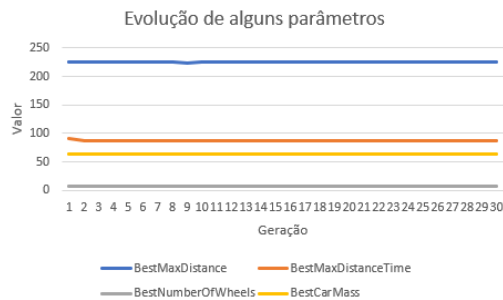
Com estes resultados percebemos que, apesar de na última tentativa o veículo ter chegado relativamente longe, esta função não é muito boa pois nas duas primeiras tentativas a distância máxima atingida ficou relativamente aquém das expectativas.

Deste modo, concluímos que ao alterar os pesos da forma que alteramos não contribui em nada para a resolução do problema, piorando inclusive, em média, os resultados anteriores. Com isto, voltamos a tentar, desta vez, com outros pesos. Se atribuir mais peso à velocidade média em relação à massa piorou os resultados, decidimos atribuir mais peso à massa. A função passou a ser então a seguinte:

$$\text{Fitness} = 15 * \text{distância percorrida} + 5 * \frac{\text{distância percorrida}}{\text{tempo demorado} + 0.01} - 8 * \text{melhor massa}$$

Os resultados foram os seguintes:

Nº Tentativa	Máxima Distância	Terminou?
1	224,8172	Não
2	329,5961	Não
3	244,2297	Não



Fica evidente que esta função foi a que nos permitiu até agora atingir uma maior distância. Porém, esta não foi suficiente para chegar ao fim do percurso. Uma questão que consideramos relevante salientar é o facto de essa distância máxima ter sido obtida na 22.ª geração e, a partir daí, os veículos não mais terem ultrapassado os 240 sequer. Isto leva-nos a crer que esta função não é ótima pois não escolheu um veículo que atingiu claramente uma distância maior que os demais.

Algo que fomos reparando à medida que realizamos as experiências anteriores foi que, ao contrário do espectável, os veículos mais longos e rasos (raso para compensar a massa ganha ao ser mais comprido), eram os que conseguiam atingir uma maior distância.

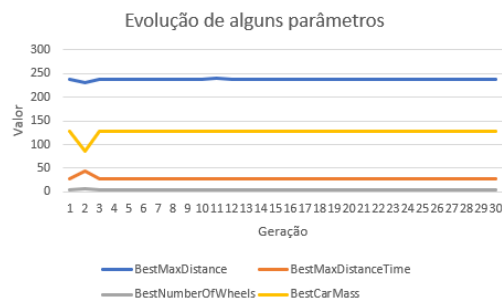
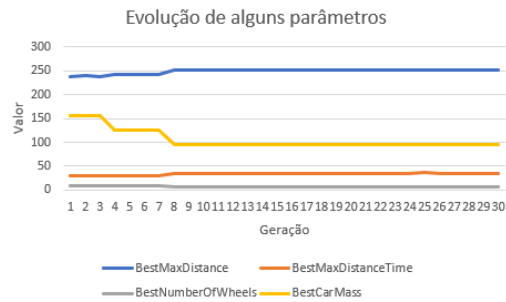
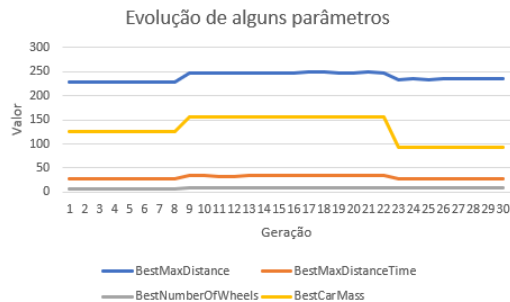
A partir dessa informação, optamos por testar uma função de fitness que envolvesse a distância percorrida, a velocidade média, o comprimento do veículo e a sua massa. Tomamos esta decisão uma vez que entendemos pelas experiências anteriores que, a distância percorrida e a velocidade média, fazem sentido estar nas funções de fitness a beneficiar quem tem um maior valor. Já quanto à massa, também chegamos à conclusão que deveríamos penalizar quem tem um maior valor. No que diz respeito ao comprimento, já que quanto comprido, mais longe chega, decidimos beneficiar os veículos maiores. Para evitar ficarmos com veículos compridos e altos (apenas queremos os compridos e rasos), ao prejudicar os de maior massa, resolvemos esse problema já que os altos têm, teoricamente, mais massa que os rasos.

Posto isto, a função de fitness que testamos foi:

$$\text{Fitness} = 3 * \text{dist. percorrida} + \frac{\text{distância percorrida}}{\text{tempo demorado} + 0.01} - \text{melhor massa} + \text{comprimento}$$

Os resultados foram os seguintes:

Nº Tentativa	Máxima Distância	Terminou?
1	248,673	Não
2	253,1236	Não
3	239,2975	Não



Fica evidente que esta função, apesar de ter mais uma característica em causa, não teve grandes melhorias em relação às anteriores.

Análise de Resultados – HillRoad

Fica evidente que, apesar das várias tentativas, não conseguimos encontrar a função de fitness ideal para atingir o final do percurso.

Em muitos casos o veículo atingiu o cume da colina mais elevada, mas, pela sua forma, ficou preso no topo já que as rodas deixaram de ter contacto com a superfície. Esta foi sem dúvida uma questão que tentamos ao longo das experiências resolver, mas que não obtivemos sucesso. Fomos sempre avançando mais um pouco, mas, à exceção de um caso, nunca passamos esse ponto.

No nosso ponto de vista, o veículo ideal devia ser comprido, raso e com três rodas (uma em cada ponta e outra no meio). Conseguimos desenvolver veículos muito semelhantes a esse, mas sem a roda do meio. Sem esta, o veículo ao ser comprido bate com uma aresta no cume da colina e as rodas, ficando no ar, deixam de fazer o veículo andar. Com a roda no meio, o veículo nunca ficaria preso nesse cume, podendo seguir livremente caminho.

Em suma:

Função Fitness	Máxima Distância	Terminou?
1	251,9311	Não

2	252,1479	Não
3	252,95	Não
4	253,0395	Não
5	254,6421	Não
6	251,6801	Não
7	253,61	Não
8	329,5961	Não
9	253,1236	Não

Com isto, concluímos que, obtivemos melhores resultados com a oitava função de fitness experimentada. Esse valor, sendo tão diferente dos demais e sendo que foi atingido numa geração relativamente cedo e não se manteve nas restantes, podemos especular que foi um caso de sorte e a função não é assim tão boa para desenvolver um veículo ao ponto de chegar àquela posição.

No entanto, perante os resultados, a nossa melhor função de fitness (a que obteve melhores resultados) foi:

$$\text{Fitness} = 15 * \text{distância percorrida} + 5 * \frac{\text{distância percorrida}}{\text{tempo demorado} + 0.01} - 8 * \text{melhor massa}$$

Conclusão

Este trabalho prático tinha como objetivo a evolução de veículos de modo a conseguir atingir o final dos cenários propostos.

Começamos por descobrir quais os parâmetros ideais para testar as várias funções de fitness por nós desenvolvidas. Para tal, utilizamos o cenário Gap Road como sugerido, uma vez que era o mais fácil de completar logo, de retirar conclusões.

Tendo esses parâmetros, e tendo conseguido completar o cenário acima referido, passamos para a cena Hill Road. Aqui testamos várias funções, algumas com mais sucesso que outras.

Sendo o algoritmo por nós implementado estocástico, existe sempre a probabilidade de a função resultar ou não, dependendo da sorte. Corremos cada função de fitness três vezes para conseguir obter algum padrão e reduzir os efeitos da aleatoriedade. Porém, essas três tentativas podem não ter sido suficientes para atingir o final do cenário Hill Road. Talvez com as funções por nós apresentadas neste relatório, correndo novamente, se tenha a sorte de atingir o final do percurso.

Com este trabalho, aprendemos o mecanismo de evoluir agentes através de funções de fitness: beneficiar quem tem as características mais necessárias e prejudicar quem não as tem. Isto ajudou a perceber melhor a matéria lecionada nas aulas teóricas da cadeira e a trabalhar numa matéria tão importante hoje em dia como são os agentes evolucionários em Inteligência Artificial.

Referências

- Material fornecido pelos docentes [acedido em 21/05/2022]
- <https://diegogiacomelli.com.br/GeneticSharp-Car2D/> [acedido em 11/05/2022]