



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Introdução à Inteligência Artificial
Fundamentos de Inteligência Artificial
2022/2023

Trabalho Prático nº 2:
Rolling in the Hill Evolutionary Edition

João António Grilo Alves, nº 2020220790, uc2020220790@student.uc.pt

Carlos Miguel Matos Soares, nº 2020230124,
uc2020230124@student.uc.pt

Conteúdo

Meta 1 – Modelação e Desenvolvimento do Algoritmo Genético.....	3
Objetivos	3
Procedimentos	3
Testes	5
Meta 2 – Experimentação e Análise.....	5
Objetivos	5
Procedimentos	5
Experiências – Configurações Utilizadas	6
Gap Road	7
Fórmula 1	7
Experiência 1	7
Experiência 2	8
Experiência 3	9
Experiência 4	10
Fórmula 2	11
Experiência 1	11
Experiência 2	12
Experiência 3	13
Experiência 4	14
Conclusão	14
Hill Road	15
Resultados	15
Conclusão	15
Obstacle Road	16
Resultados	16
Conclusão	16
Conclusão	17

Meta 1 – Modelação e Desenvolvimento do Algoritmo Genético

Objetivos

Implementar os algoritmos de Recombinação Uniforme, Mutação Gaussiana, Roleta e Elitismo.

Procedimentos

Tendo como base os algoritmos apresentados no enunciado, a adaptação dos mesmos para C# foi relativamente fácil de fazer.

Para o algoritmo de Crossover Uniforme, alteramos o ficheiro “Meta1/UniformCrossver.cs”, sendo o código alterado o seguinte:

```
public UniformCrossover(float crossoverProbability) : this(2, 2, 2, true)
{
    this.crossoverProbability = crossoverProbability;
}

public UniformCrossover(int parentsNumber, int offSpringNumber, int minChromosomeLength, bool isOrdered)
{
    ParentsNumber = parentsNumber;
    ChildrenNumber = offSpringNumber;
    MinChromosomeLength = minChromosomeLength;
    IsOrdered = isOrdered;
}

public IList<IChromosome> Cross(IList<IChromosome> parents)
{
    IChromosome parent1 = parents[0];
    IChromosome parent2 = parents[1];
    IChromosome offspring1 = parent1.Clone();
    IChromosome offspring2 = parent2.Clone();

    double[] mask = new double[parent1.Length];
    for (int i = 0; i < parent1.Length; i++)
    {
        mask[i] = RandomizationProvider.Current.GetDouble(0, 1);
    }

    if (RandomizationProvider.Current.GetDouble() <= crossoverProbability)
    {
        for (int i = 0; i < parent1.Length; i++)
        {
            if (mask[i] >= 0.5)
            {
                offspring1.ReplaceGene(i, parent2.GetGene(i));
                offspring2.ReplaceGene(i, parent1.GetGene(i));
            }
        }
    }

    return new List<IChromosome> { offspring1, offspring2 };
}
```

Rolling in the Hill Evolutionary Edition

Para o algoritmo de Mutação, alteramos o ficheiro “Meta1/GaussianMutation.cs”, sendo o código alterado o seguinte:

```
public class GaussianMutation : IMutation
{
    public bool IsOrdered { get; private set; } // indicating whether the operator is ordered (if can keep the chromosome order).
    private double std = 1;

    public GaussianMutation(double std)
    {
        IsOrdered = true;
        this.std = std;
    }

    public GaussianMutation()
    {
        IsOrdered = true;
    }

    public void Mutate(ICHromosome chromosome, float probability)
    {
        //YOUR CODE HERE
        for (int i = 0; i < chromosome.Length; i++)
        {
            if (RandomizationProvider.Current.GetDouble() <= probability)
            {
                double geneValue = SampleGaussian((double)chromosome.GetGene(i).Value, this.std);
                chromosome.ReplaceGene(i, new Gene(geneValue));
            }
        }
    }

    protected double SampleGaussian(double mean, double std)
    {
        //Generates a random number based on a normal distribution using the Box-Muller method.
        double x1 = RandomizationProvider.Current.GetDouble(0, 1);
        double x2 = RandomizationProvider.Current.GetDouble(0, 1);
        double y1 = Math.Sqrt(-2.0 * Math.Log(x1)) * Math.Cos(2.0 * Math.PI * x2);
        return y1 * std + mean;
    }
}
```

Para o algoritmo de Roleta, alteramos o ficheiro “Meta1/Roulette.cs”, sendo o código alterado o seguinte:

```
public class Roulette : SelectionBase
{
    public Roulette() : base(2)
    {
    }

    protected override IList<ICHromosome> PerformSelectChromosomes(int number, Generation generation)
    {
        IList<CarChromosome> population = generation.Chromosomes.Cast<CarChromosome>().ToList();
        IList<ICHromosome> parents = new List<ICHromosome>();

        //YOUR CODE HERE
        double sumFitness = 0.0;
        int i = 0;

        for (; i < population.Count; i++)
        {
            sumFitness += population[i].Fitness;
        }

        while (parents.Count < number)
        {
            double pointer = RandomizationProvider.Current.GetDouble();
            double partial = 0.0;
            int index = 0;
            while (partial <= pointer)
            {
                partial += population[index].Fitness / sumFitness;
                index++;
            }
            parents.Add(population[index - 1]); // we want the individual of the previous index
        }

        return parents;
    }
}
```

Para o algoritmo de Elitismo, alteramos o ficheiro “Meta1/Elitism.cs”, sendo o código alterado o seguinte:

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using GeneticSharp.Domain.Chromosomes;
using GeneticSharp.Domain.Populations;
using GeneticSharp.Domain.Reinsertions;

public class Elitism : ReinsertionBase
{
    protected int eliteSize = 0;
    public Elitism(int eliteSize) : base(false, false)
    {
        this.eliteSize = eliteSize;
    }

    protected override IList<IChromosome> PerformSelectChromosomes(IPopulation population, IList<IChromosome> offspring, IList<IChromosome> parents)
    {
        // YOUR CODE HERE
        var bestParents = population.CurrentGeneration.Chromosomes.OrderByDescending(p => p.Fitness).Take(eliteSize).ToList();
        for (int i = 0; i < bestParents.Count; i++)
        {
            offspring[i] = bestParents[i];
        }
        return offspring;
    }
}
```

Testes

Após a implementação do código, ao tentar correr o projeto uma vez, os carros foram gerados até às 30 gerações, o que prova que o código se encontra 100% funcional, podendo assim prosseguir para a próxima meta.

Meta 2 – Experimentação e Análise

Objetivos

Estudar a evolução dos veículos ao longo das 30 gerações em cada experiência.

Criar um veículo que consiga passar em todos os cenários de avaliação (“Evaluation”).

Procedimentos

Nesta fase, inicialmente, começamos por estudar diversas funções de fitness, nomeadamente:

- Fitness = MaxDistance;
- Fitness = MaxDistance – MaxDistanceTime;
- Fitness = MaxDistance + MaxVelocity;
- Fitness = MaxDistance + MaxVelocity + CarMass.

No entanto, aquela que prevaleceu (daqui em diante será tratada por Fórmula 1) foi:

- $$\text{Fitness} = 75 * \text{MaxDistance} / 547 + 10 * \text{IsRoadComplete} + 15 * \text{helpMaxDistanceTime}.$$

Deste modo, os valores iriam todos cair entre 0 e 100. O MaxDistance é dividido 547 pois 547 é a distância máxima do GapRoad. helpMaxDistanceTime é obtido da seguinte forma:

Rolling in the Hill Evolutionary Edition

```
// Se demorar 5 segundos ou menos, ou mais de 60 segundos, vai receber 0 pontos pelo helpMaxDistanceTime
if (MaxDistanceTime < 5 || MaxDistanceTime > 60) helpMaxDistanceTime = 0;

else if (MaxDistanceTime < 15) helpMaxDistanceTime = (float)(0.1 * MaxDistanceTime - 0.5);

else helpMaxDistanceTime = -MaxDistanceTime/45 + 4/3;
```

Outra consideração que tivemos foi a de que havia carros a “voar” por cima da meta, algo que vimos acontecer pelo menos 6 vezes. Decidimos então que se um carro sobrevoasse a meta:

```
if (MaxDistance > 547)
{
    MaxDistance = 547;
    IsRoadComplete = 1;
}
```

Após obter todos os valores das experiências com esta fórmula, criamos uma segunda versão que aparentava ser melhor, sendo que o objetivo da mesma é beneficiar quanto mais longe forem e, para diferenciar os carros que acabem o percurso uns dos outros, penalizá-los pelo tempo gasto, sendo esta a “Fórmula 2”:

Float t = MaxDistanceTime > 100 ? 100 : MaxDistanceTime;

Fitness = MaxDistance + IsRoadComplete * (100 – t);

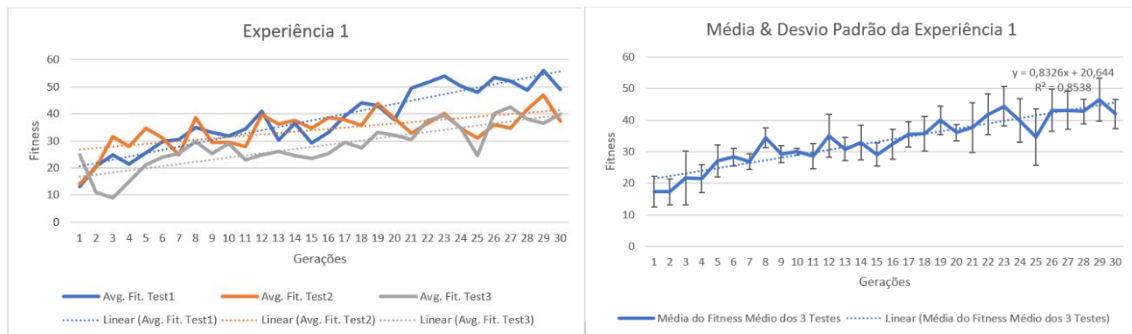
Experiências – Configurações Utilizadas

	Mutação	Elitismo	Crossover	Número de Gerações
Experiência 1	0,05	0	0,9	30
Experiência 2	0,2			
Experiência 3	0,05	2		
Experiência 4	0,2			

Gap Road

Fórmula 1

Experiência 1



Gráficos 1.1.1 e 1.1.2: Evolução do fitness médio da população ao longo das gerações para os valores da experiência 1 em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	93,48959	547	19,53125	13	409,9167	True
2	93,02663	547	20,9201	13	352,8132	True
3	74,39759	496,4052	25,99585	12	343,9671	False
Média	86,97127	530,1350667	22,14907	12,6666667	368,899	True

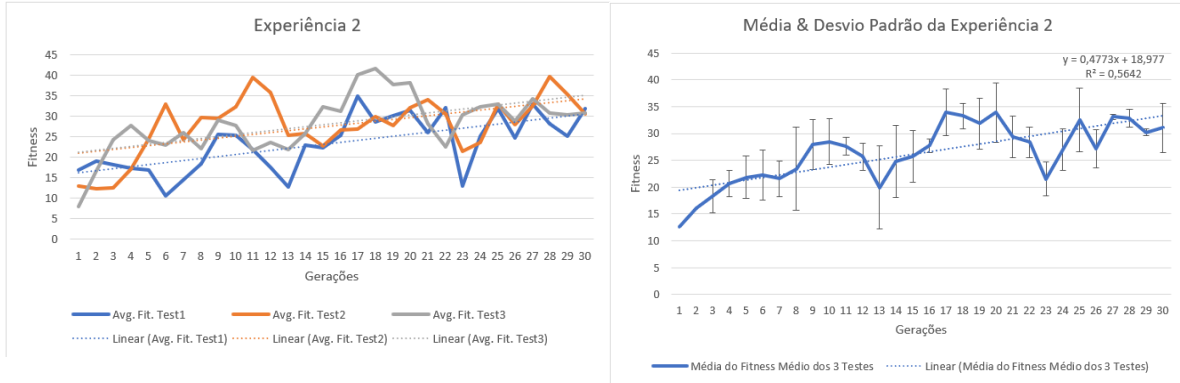
Tabela 1.1: Melhor carro de cada teste na 1ª experiência com a fórmula 1.



Figura 1.1: Melhores Carros (Fitness Maior) de Cada Teste para a Experiência 1, ordenados pela tabela 1.1.

Analisando o gráfico de dispersões da experiência, verificamos o efeito do parâmetro de “Elitismo” a 0, ou seja, a não utilização de Elitismo provoca alguma indefinição na evolução genética do fenótipo. Verificamos também, que durante a execução, alguns veículos conseguiram terminar o cenário, no entanto, apesar de alguns dos seus genes continuarem na geração seguinte, por efeito de mutação, não mantém as propriedades.

Experiência 2



Gráficos 1.2.1 e 1.2.2: Evolução do fitness médio da população ao longo das gerações para os valores da experiência 2 em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	93,07061768	547	20,78815	13	417,9718	True
2	91,9642334	547	24,1073	14	323,3241	True
3	45,43288422	265,9619	18,10065	11	343,4568	False
Média	76,82257843	453,3206333	20,9987	12,66666667	361,58423	True

Tabela 1.2: Melhor carro de cada teste na 2ª experiência com a fórmula 1.

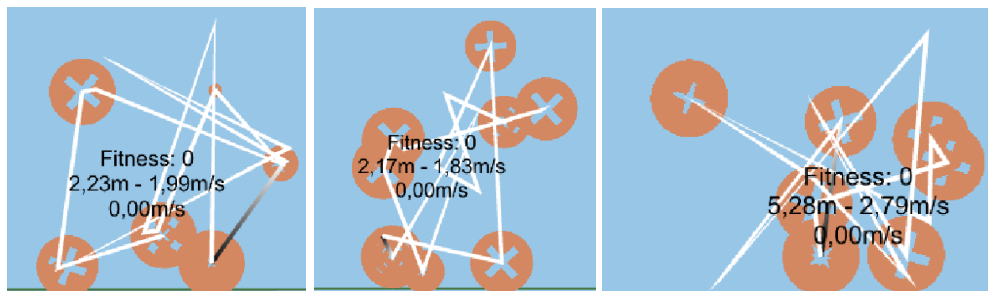


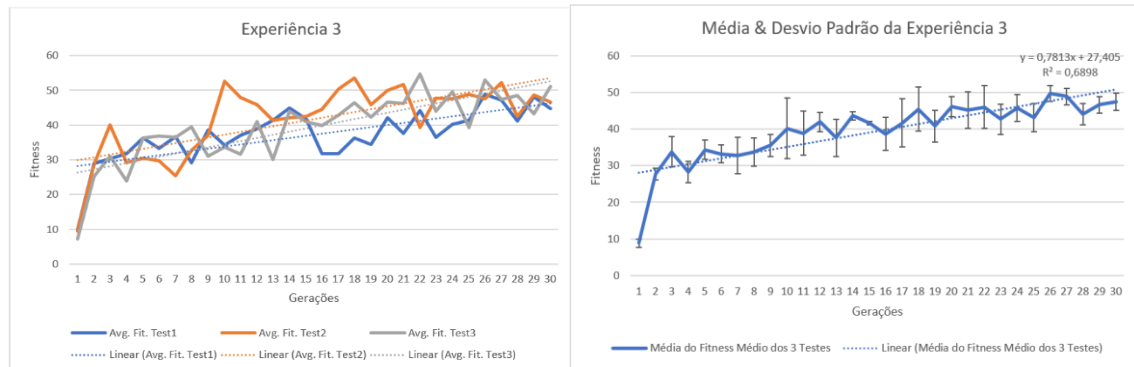
Figura 1.2: Melhores Carros (Fitness Maior) de Cada Teste para a Experiência 2, ordenados pela tabela 1.2.

De forma semelhante à Experiência 1, e analisando o gráfico de dispersões da experiência, verificamos novamente o efeito de pouca linearidade do parâmetro de “Elitismo”.

Analisando as diferenças visuais entre os gráficos da experiência 1 e 2, onde o único parâmetro que se altera é a probabilidade de mutação, sendo mais elevada na experiência 2, verificamos uma maior diferença entre veículos de gerações sucessivas na Experiência 2, tal como seria expectável, pois os carros encontram-se mais propensos a mutações neste caso. Matematicamente, a taxa de crescimento do fitness por geração é superior na experiência 1 (0.8326) relativamente à experiência 2 (0.4773).

Logo, é mais provável existirem carros capazes de acabar o trajeto na experiência 1 do que na 2.

Experiência 3



Gráficos 1.3.1 e 1.3.2: Evolução do fitness médio da população ao longo das gerações para os valores da experiência 3 em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	91,7494125	547	24,75177	13	432,9418	True
2	92,5798339	547	22,2605	11	319,5267	True
3	93,2145614	547	20,35632	13	270,5515	True
Média	92,5146026	547	22,45619	12,3333333	341,00666	True

Tabela 1.3: Melhor carro de cada teste na 3ª experiência com a fórmula 1.

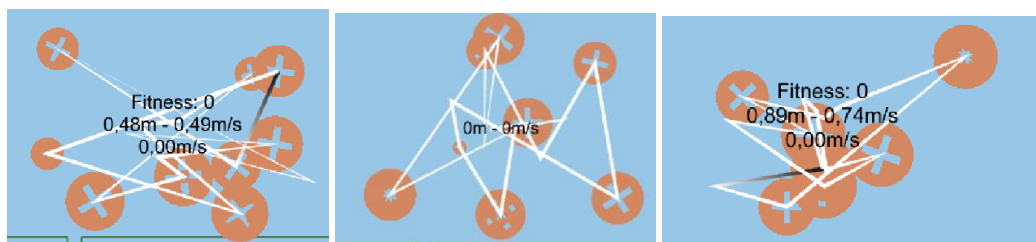
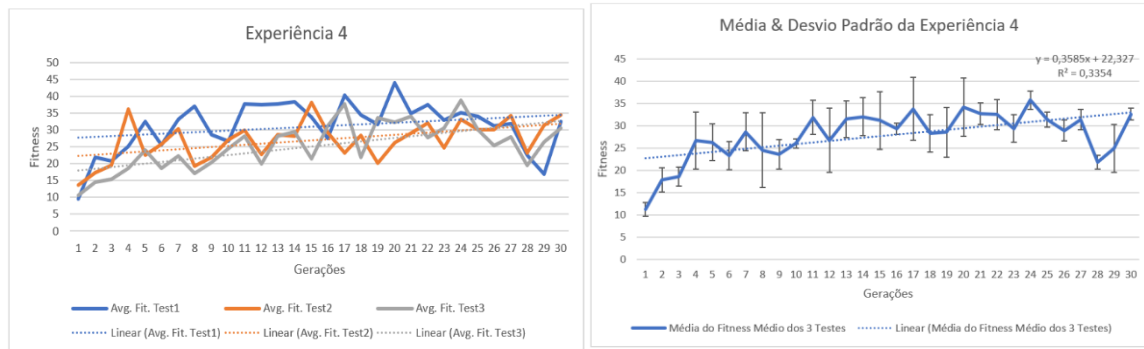


Figura 1.3: Melhores Carros (Fitness Maior) de Cada Teste para a Experiência 3, ordenados pela tabela 1.3.

Analisando o gráfico da Experiência 3 conseguimos verificar a tese apresentada nas experiências anteriores sobre o Elitismo: verificamos que os resultados das diferentes gerações apresentam uma tendência crescente, a uma taxa superior a qualquer uma das anteriormente apresentadas (cresce, em média, 0.7813 de fitness por geração)

Experiência 4



Gráficos 1.4.1 e 1.4.2: Evolução do fitness médio da população ao longo das gerações para os valores da experiência 4 em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	92,9166870	547	21,2499	12	369,8724	True
2	79,7879104	526,51	22,2080	12	293,6404	False
3	92,7460556	547	21,7618	11	377,2651	True
Média	88,483551	540,17	21,7399	11,66666667	346,925966	True

Tabela 1.4: Melhor carro de cada teste na 4ª experiência com a fórmula 1.

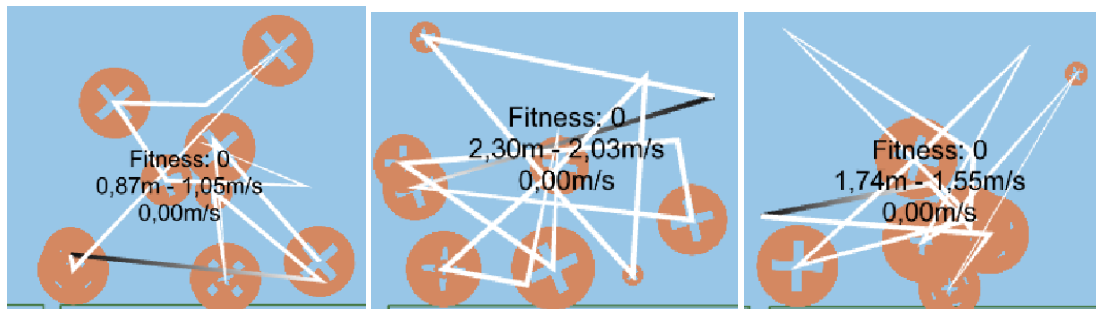
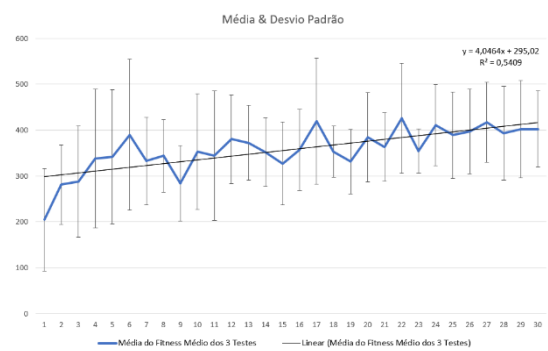
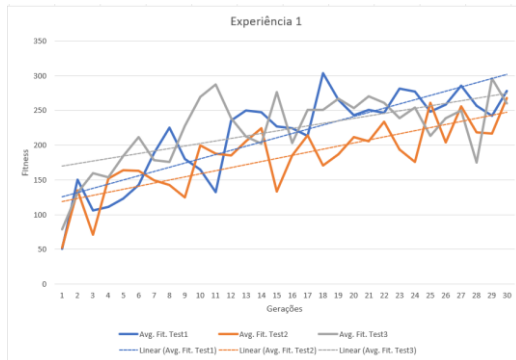


Figura 1.4: Melhores Carros (Fitness Maior) de Cada Teste para a Experiência 4, ordenados pela tabela 1.4.

À semelhança da experiência 1 e 2, da experiência 3 para a 4 apenas varia o grau de mutação genética. Portanto seria expectável obter também resultados piores na experiência 4 do que para a experiência 3, e foi isso que se verificou.

Fórmula 2

Experiência 1



Gráficos 2.1.1 e 2.1.2: Evolução do fitness médio da população ao longo das gerações para os valores da experiência 1 em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	532,640380	532,6404	27,02899	13	326,3532	False
2	527,803955	527,804	29,13153	12	356,03	False
3	530,230896	530,2309	24,15387	13	336,5626	False
Média	530,22507	530,2251	26,77146	12,66666667	339,6486	False

Tabela 2.1: Melhor carro de cada teste na 1ª experiência com a fórmula 2.

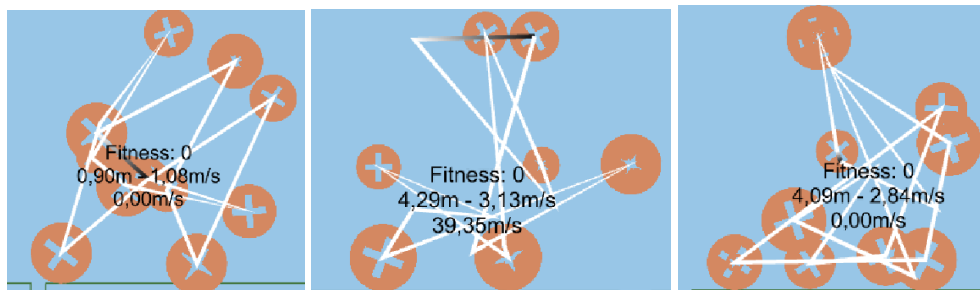


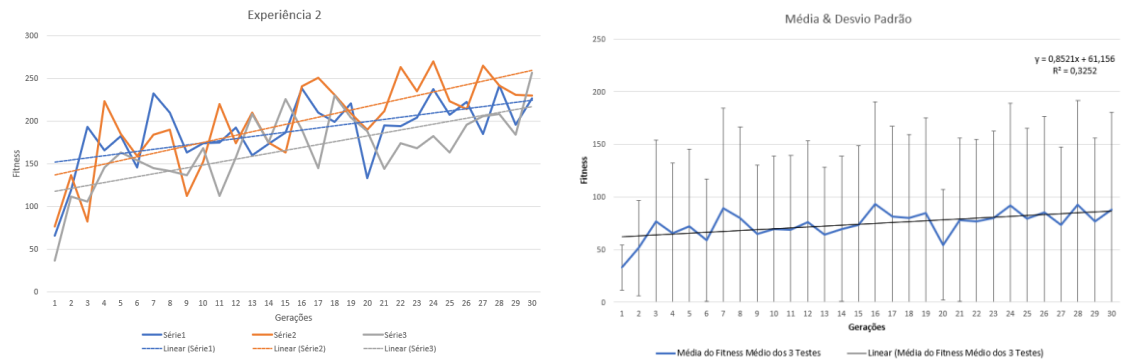
Figura 2.1: Melhores Carros (Fitness Maior) de Cada Teste para a Experiência 1, ordenados pela tabela 2.1.

Análise

Apesar de nenhum carro ter concluído o trajeto, esta fórmula revelou-se melhor do que a anterior pois apresenta um melhor crescimento entre gerações (o incremento é cada vez maior). A análise do gráfico permite retirar as mesmas conclusões da análise aos gráficos 1.1.1 e 1.1.2.

As 3 experiências que se seguem não terão análise detalhada, uma vez que as mesmas permitem chegar às mesmas conclusões das análises retiradas com a experiência 1.

Experiência 2



Gráficos 2.2.1 e 2.2.2: Evolução do fitness médio da população ao longo das gerações para os valores da experiência 2 em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	568,88836	547	25,28559	11	322,3743	True
2	567,82708	547	22,73456	14	354,8825	True
3	560,59643	547	28,04846	13	329,3895	True
Média	565,770623	547	25,35620	12,666667	335,548767	True

Tabela 2.2: Melhor carro de cada teste na 2ª experiência com a fórmula 2.

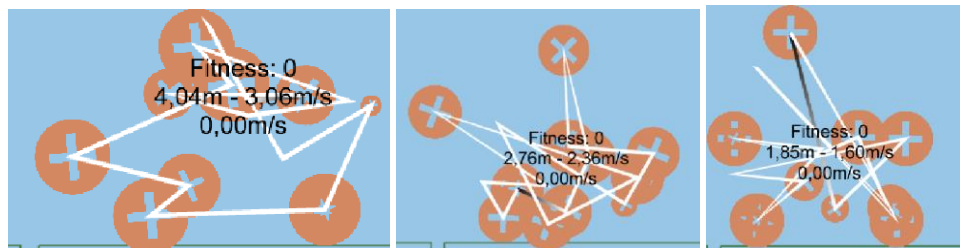
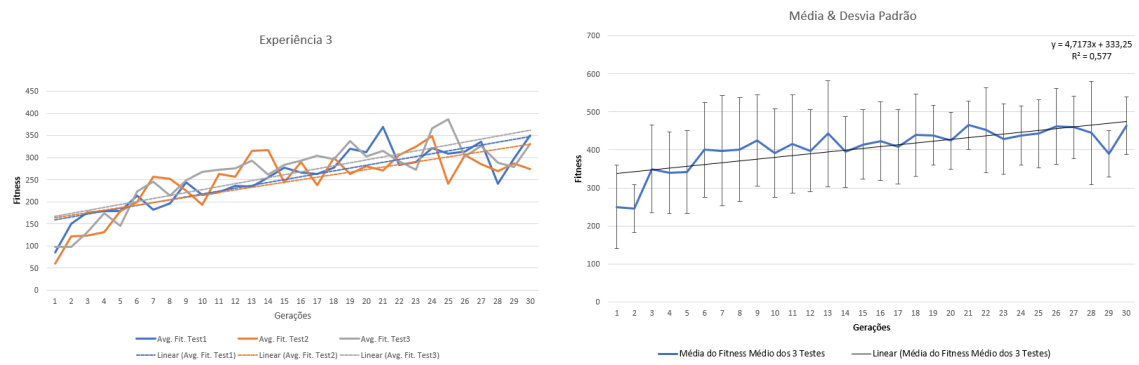


Figura 2.2: Melhores Carros (Fitness Maior) de Cada Teste para a Experiência 2, ordenados pela tabela 2.2.

Experiência 3



Gráficos 2.3.1 e 2.3.2: Evolução do fitness médio da população ao longo das gerações para os valores da experiência 3 em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	561,678466	547	24,97101	11	318,2791	True
2	563,613281	547	25,43356	14	352,7689	True
3	565,824035	547	30,93567	14	386,4385	True
Média	563,705261	547	27,11341	13	352,4955	True

Tabela 2.3: Melhor carro de cada teste na 3ª experiência com a fórmula 2.

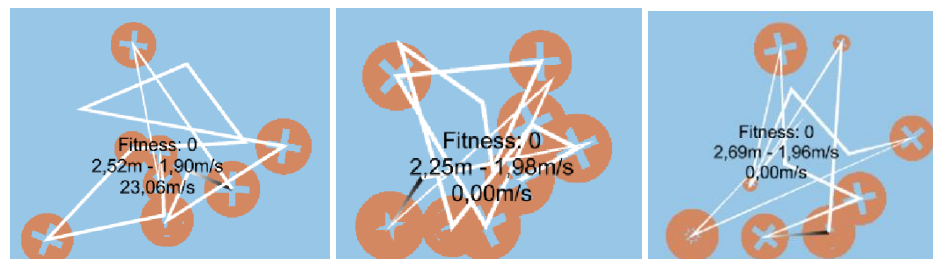
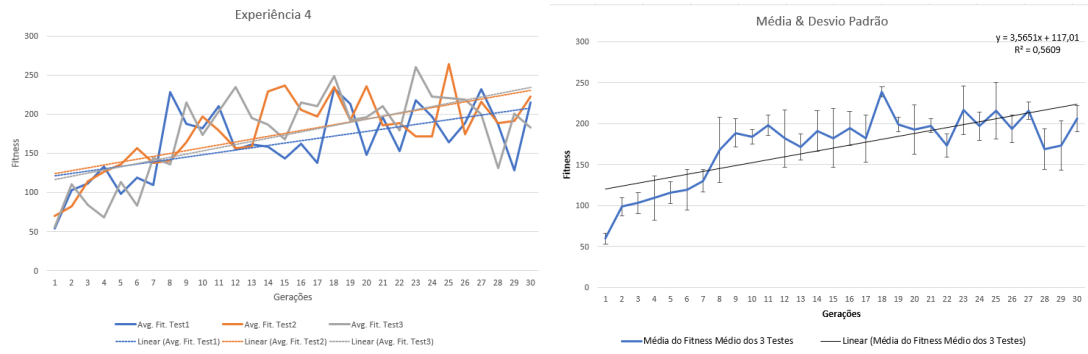


Figura 2.3: Melhores Carros (Fitness Maior) de Cada Teste para a Experiência 2, ordenados pela tabela 2.3.

Rolling in the Hill Evolutionary Edition

Experiência 4



Gráficos 2.4.1 e 2.4.2: Evolução do fitness médio da população ao longo das gerações para os valores da experiência 4 em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	569,728454	547	27,28598	13	327,2394	True
2	557,621887	547	29,48953	13	309,5735	True
3	561,468566	547	27,15543	13	300,6604	True
Média	562,939636	547	27,97698	13	312,4911	True

Tabela 2.4: Melhor carro de cada teste na 4ª experiência com a fórmula 2.

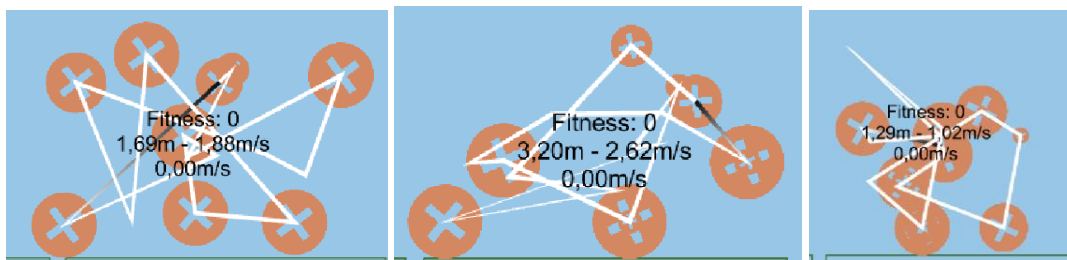


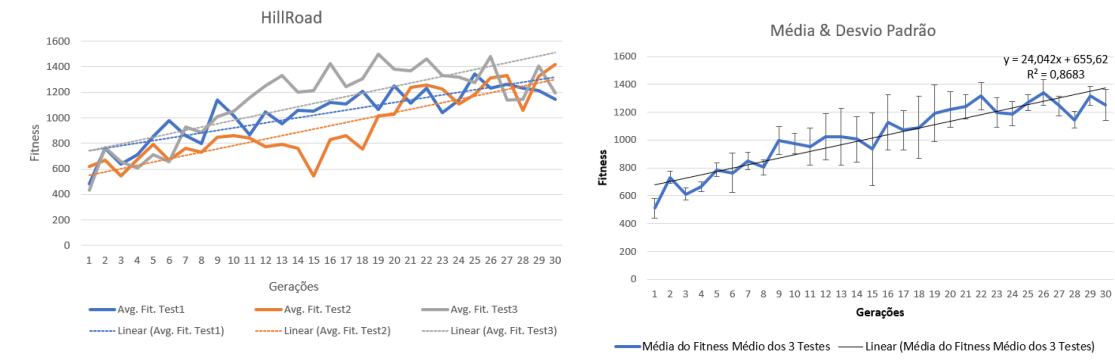
Figura 2.4: Melhores Carros (Fitness Maior) de Cada Teste para a Experiência 2, ordenados pela tabela 2.4.

Conclusão

A experiência que apresentou melhores resultados foi a número 3, quer para uma fórmula, quer para a outra, sendo aquela cujo declive da regressão linear é maior (ou seja, é aquela no qual o crescimento do fitness entre gerações é superior). Isto acontece poi, aproveita sempre os dois melhores exemplares de cada geração, sendo a probabilidade de mutação bastante baixa. Como tal, para os cenários do HillRoad e ObstacleRoad, iremos considerar os parâmetros da experiência 3, com o elitismo a 2 e a probabilidade de mutação a 0.05. Para a função de fitness, usaremos a fórmula 2.

Hill Road

Resultados



Gráficos 3.1 e 3.2: Evolução do fitness médio da população ao longo das gerações para os valores do Hill Road em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	1766,85986	1750	83,14014	7	172,2884	True
2	1771,44482	1750	78,55518	7	148,6558	True
3	1765,93359	1750	84,06641	6	153,2819	True
Média	1768,07942	1750	81,9205	6,66666667	158,075366	True

Tabela 3.1: Melhor carro de cada teste no cenário Hill Road.



Figura 3.1: Melhores Carros (Fitness Maior) de Cada Teste para o cenário Hill Road, usando a fórmula 2, ordenados pela tabela 3.1.

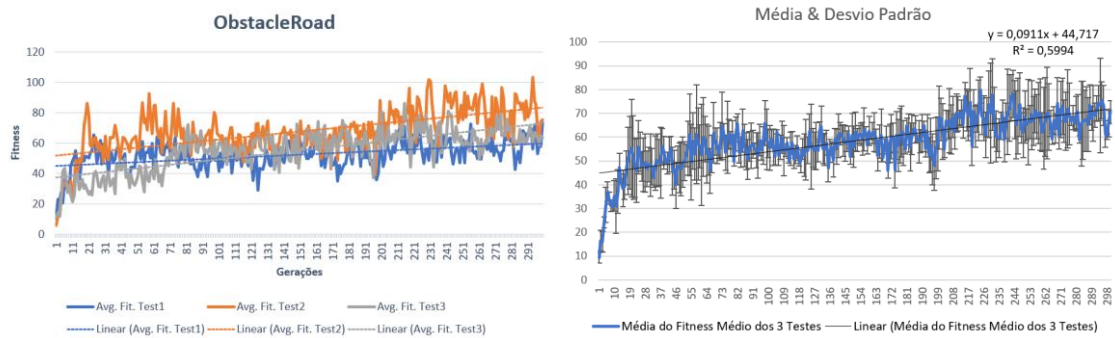
Conclusão

Após experimentação, o melhor carro para este cenário provou ser um carro baixo e com duas rodas grandes, uma em cada extremidade.

Usando os valores da Experiência 3, os carros evoluíram a um excelente ritmo, sendo que a partir da 17ª geração, houve sempre pelo menos um carro a atingir a meta, o que levou a uma melhor evolução dos carrinhos.

Obstacle Road

Resultados



Gráficos 4.1 e 4.2: Evolução do fitness médio da população ao longo de 300 gerações para os valores do Obstacle Road em 3 testes diferentes, bem como a evolução da sua média.

Teste	Fitness	MaxDistance	Time	Wheels	Mass	Road Complete
1	181,095413	181,0954	23,98047	10	311,346	False
2	172,501297	172,5013	42,1543	12	188,4198	False
3	202,160751	202,1608	24,60156	10	286,0574	False
Média	185,25249	185,2525	30,24544	10,6666667	261,941067	False

Tabela 4.1: Melhor carro de cada teste no cenário Obstacle Road.



Figura 4.1: Melhores Carros (Fitness Maior) de Cada Teste para o cenário Obstacle Road, usando a fórmula 2, ordenados pela tabela 4.1.

Conclusão

Inicialmente testámos com apenas 30 gerações. No entanto, nenhum dos veículos conseguia superar fitness 110 (ultrapassar os 110 metros). Decidimos então alargar o número de gerações para 300. Apesar de não termos conseguido chegar a nenhum carro que chegasse ao final do Obstacle Road, qualquer um dos melhores carros de cada teste conseguiu superar o Gap Road e o Hill Road com relativa facilidade.

Provavelmente, mesmo fazendo a filtragem dos carros que chegam mais longe, seria muito difícil atingir a meta antes das 500 gerações (de acordo com a reta de regressão linear).

Conclusão

Este projeto teve como objetivo o desenvolvimento de um algoritmo evolucionário (AE) para o desenho de um veículo motorizado. Inspirado na Teoria da Seleção Natural de Charles Darwin e nos princípios de herança genética de Gregor Mendel, o AE utiliza um simulador virtual para avaliar o desempenho do veículo em diferentes cenários.

Através da modelação e implementação do AE, foram desenvolvidas as componentes essenciais, como a representação dos cromossomas, os operadores genéticos (recombinação e mutação), os mecanismos de seleção e a função de aptidão. O objetivo principal era melhorar a performance do veículo ao longo do tempo, permitindo que ele percorresse a maior distância possível nos percursos propostos.

Foram realizadas experimentações e análises utilizando o cenário GapRoad, onde foram testadas diferentes configurações do AE. Foram exploradas diferentes funções de aptidão e ajustados os parâmetros do algoritmo, como a taxa de mutação, elitismo e número de gerações. As experiências realizadas visavam encontrar soluções que permitissem aos veículos completar o percurso com sucesso.

No final, o projeto demonstrou a aplicação prática dos algoritmos evolucionários no desenvolvimento de soluções adaptativas. Os resultados obtidos mostraram a capacidade do AE em evoluir os veículos, melhorando a sua performance ao longo das gerações. Através deste processo evolutivo, os veículos foram capazes de se adaptar aos desafios dos cenários propostos, demonstrando a eficácia e versatilidade desta abordagem inspirada na natureza.

Em suma, este projeto proporcionou uma experiência de aprendizagem significativa no campo da análise, desenvolvimento e implementação de agentes adaptativos utilizando algoritmos evolucionários. Os conhecimentos adquiridos neste trabalho podem ser aplicados em diversas áreas que envolvam otimização e resolução de problemas complexos, contribuindo para avanços na ciência da computação e engenharia.

Concluindo, com este trabalho, o grupo ficou a perceber mais como funcionam os Algoritmos Evolucionários e o quanto eles podem ser explorados para diferentes finalidades. Apesar de ser um trabalho que requeria muito tempo, principalmente em testes, foi uma experiência positiva e o grupo em reflexão pensa que correu muito bem.