

**Fundamentos de Inteligência Artificial 2023/24**

**Licenciatura em Engenharia Informática**

**Trabalho Prático Nº2**



**Rolling in the Hill Evolutionary Edition**

**Discente:**

**-Cíntia Dalila Luís Cumbane**

**Número:**

**202024460**

**Coimbra, aos 20 de Maio de 2024**



## Índice

<b>1.Introdução</b> .....	2
<b>2.Meta 1 - Modelação e desenvolvimento do Algoritmo Genético</b> .....	2
<b>2.1 Recombinação</b> .....	2
<b>2.2 Seleção de Progenitores</b> .....	3
<b>2.3 Mutação</b> .....	4
<b>2.4 Aptidão</b> .....	4
<b>2.6. Parametrização</b> .....	5
<b>3. Experimentação e análise – GapRoad</b> .....	5
<b>3.1. Métodos de Avaliação</b> .....	5
<b>3.2. Primeira Iteração</b> .....	6
<b>3.3. Segunda Iteração</b> .....	9
<b>3.4. Terceira Iteração</b> .....	11
<b>4. Experimentação e análise - HillRoad</b> .....	13
<b>4.1. Primeira Iteração</b> .....	13
<b>4.2. Segunda Iteração</b> .....	14
<b>5. Experimentação e análise - RockyHillRoad</b> .....	15
<b>5.1. Primeira Iteração</b> .....	15
<b>5.2. Segunda Iteração</b> .....	16
<b>Conclusão</b> .....	18
<b>Referências</b> .....	19

## 1.Introdução

Este projeto tem como finalidade o desenvolvimento de um algoritmo evolucionário capaz de criar veículos motorizados através de um simulador virtual. O objetivo é avaliar o desempenho desses veículos em diferentes cenários. Este trabalho prático visa principalmente adquirir competências de análise, desenvolvimento, implementação e teste de agentes adaptativos.

Assim, pretende-se elaborar um Algoritmo Evolucionário (AE) que possibilite a geração de veículos capazes de percorrer diversos trajetos. O AE irá evoluir um conjunto de parâmetros relacionados com a estrutura dos veículos (por exemplo, número de rodas, posição das rodas, raio, etc.), de modo a criar veículos com base nesta dinâmica evolutiva.

Ao longo deste trabalho, serão exploradas diferentes abordagens para otimizar o desempenho dos veículos. Por meio da análise dos resultados obtidos, será possível compreender melhor as estratégias de adaptação e evolução dos agentes no contexto da criação de veículos motorizados.

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

Neste capítulo, são detalhadas todas as etapas envolvidas na elaboração do Algoritmo Genético, tais como:

- Recombinação
- Seleção de progenitores
- Mutação
- Aptidão

### 2.1 Recombinação

A etapa de recombinação é fundamental no processo de evolução dos veículos motorizados. A abordagem de recombinação adotada é o método de crossover de ponto único. Nesse método, um ponto de corte é escolhido aleatoriamente ao longo dos cromossomos parentais. Os genes à esquerda desse ponto de corte são herdados de um dos pais, enquanto os genes à direita são herdados do outro pai. Isso cria dois cromossomos descendentes, combinando características de ambos os pais[1].

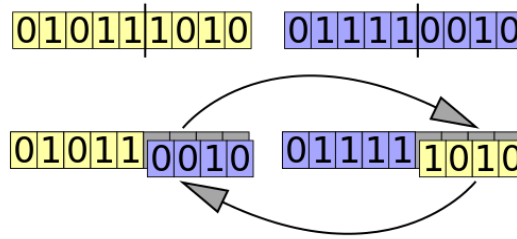


Figura 1: ilustração de recombinação de ponto único

A escolha desse método se deve à sua simplicidade e eficácia em explorar o espaço de busca de soluções. Além disso, o crossover de ponto único ajuda a preservar partes promissoras dos cromossomos parentais, contribuindo para a convergência em direção a soluções melhores ao longo das gerações.

## 2.2 Seleção de Progenitores

A seleção de progenitores é o processo de escolha dos indivíduos que irão se reproduzir na próxima geração. Para isso, foi implementado o método de seleção por torneio. Nesse método, seleciona-se aleatoriamente um número fixo de indivíduos da população e escolhe-se o melhor deles com base em sua aptidão. Esse indivíduo selecionado é então escolhido como progenitor para a próxima geração.

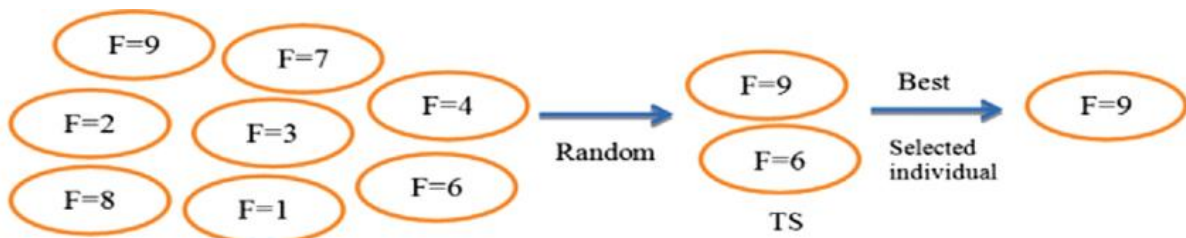


Figura 2: ilustração da seleção de progenitores por torneio

A seleção por torneio oferece uma maneira simples e eficaz de manter a diversidade genética na população, ao mesmo tempo em que favorece a reprodução dos indivíduos mais aptos. Isso ajuda a evitar a convergência prematura para mínimos locais e promove a exploração do espaço de busca de soluções de forma mais eficiente.

## 2.3 Mutação

A mutação é um processo genético essencial em algoritmos evolucionários, funcionando de forma semelhante à mutação biológica. Ele desempenha um papel crucial na introdução de diversidade genética em uma população de indivíduos, o que é fundamental para a evolução e melhoria contínua do desempenho do algoritmo. Ao introduzir variações nos genes dos indivíduos, a mutação ajuda a equilibrar a relação entre as mutações e a herança genética, promovendo assim a convergência do algoritmo evolucionário em direção a soluções ótimas ou próximas a elas [2]. Foi implementada uma mutação que opera em cada gene do cromossomo com uma determinada probabilidade. Para isso, utilizou-se uma abordagem baseada em uma distribuição gaussiana para gerar novos valores para os genes mutados, permitindo a exploração eficiente do espaço de busca.

Essa mutação é crucial para a adaptação dos veículos a diferentes ambientes e desafios, garantindo que a população não fique presa em ótimos locais subótimos. Ao introduzir aleatoriedade controlada no processo evolutivo, a mutação permite que o algoritmo genético explore uma variedade de soluções e encontre as mais adequadas para os diferentes cenários.

## 2.4 Aptidão

A avaliação da aptidão dos veículos motorizados é crucial para direcionar o processo evolutivo em direção a soluções melhores. A aptidão de um veículo é determinada pela sua capacidade de percorrer um determinado trajeto e atingir os objetivos definidos. Para isso, são considerados diversos critérios, como a distância percorrida, o tempo necessário para completar o percurso, a estabilidade do veículo durante o trajeto e sua eficiência energética.

Esses critérios são combinados em uma função de aptidão, que atribui um valor numérico a cada veículo com base em seu desempenho. Quanto melhor o desempenho do veículo, maior será sua pontuação de aptidão. Essa função de aptidão é fundamental para orientar a seleção de indivíduos na população e guiar o processo evolutivo em direção a soluções mais eficazes.

A escolha dos critérios de avaliação e a formulação da função de aptidão são aspectos críticos do projeto, pois influenciam diretamente na qualidade das soluções encontradas pelo algoritmo genético. Portanto, é importante realizar uma análise cuidadosa dos requisitos do problema e dos objetivos do projeto para definir uma função de aptidão adequada, que seja capaz de capturar os aspectos mais importantes do desempenho dos veículos motorizados nos cenários simulados.

Por essa razão, farei ajustes na função de aptidão de acordo as necessidades específicas, destacando a importância dessa etapa no processo. Embora não tenha a função de aptidão final definida, farei os ajustes necessários para garantir resultados satisfatórios.

## 2.6. Parametrização

Na parametrização do algoritmo, consideram-se os seguintes parâmetros:

- 1. Probabilidade de Crossover:** Determina a chance de dois cromossomos se combinarem durante a recombinação.
- 2. Probabilidade de Mutação:** Especifica a probabilidade de cada gene em um cromossomo sofrer mutação.
- 3. Número Máximo de Gerações:** Define o limite de gerações que o algoritmo executará antes de encerrar.
- 4. Tamanho da Elite:** Determina quantos dos melhores indivíduos serão mantidos intactos na população.
- 5. Tamanho do Torneio:** Define o número de indivíduos que participarão do torneio de seleção de pais.
- 6. Valores Mínimo e Máximo do Gene:** Estabelece os limites dos valores dos genes nos cromossomos.

Esses parâmetros são fundamentais para ajustar o comportamento e o desempenho do algoritmo durante a evolução dos veículos motorizados.

## 3. Experimentação e análise – GapRoad

### 3.1. Métodos de Avaliação

Na fase de experimentação e análise do cenário GapRoad, adotaram-se métodos de avaliação que nos permitem compreender o progresso e a variabilidade da aptidão dos indivíduos ao longo das gerações. De forma a avaliar os resultados, extraíram-se dos dados a média e desvio padrão ao longo das gerações, isto porque a média permite-nos perceber o crescimento ou decrescimento da aptidão dos indivíduos e o desvio padrão permite perceber a variação da aptidão entre gerações.

## 3.2. Primeira Iteração

### 1. Distância sobre Tempo (Linear):

Nesta abordagem a fórmula usada para calcular a aptidão (fitness) de cada indivíduo (carro) em uma população dentro do seu algoritmo genético para melhorar a eficiência dos carros de forma a terminar o percurso.

#### 1. Cálculo da Média de Velocidade:

- Calcula-se a média das velocidades alcançadas pelos carros durante o percurso. Isso é feito dividindo a soma total das velocidades pelo número de velocidades registradas. Essa média representa uma medida geral de desempenho em termos de velocidade dos carros.

#### 2. Cálculo da Fitness:

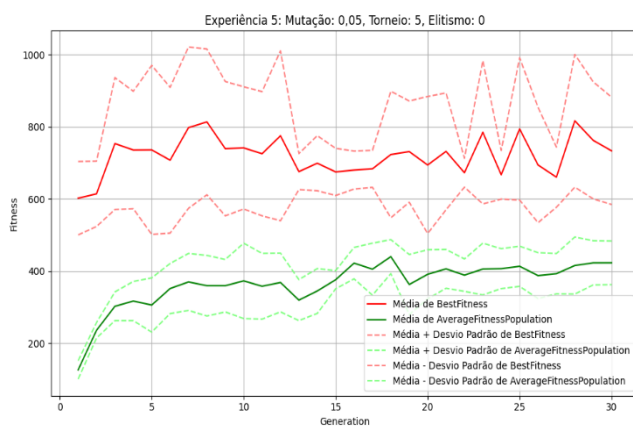
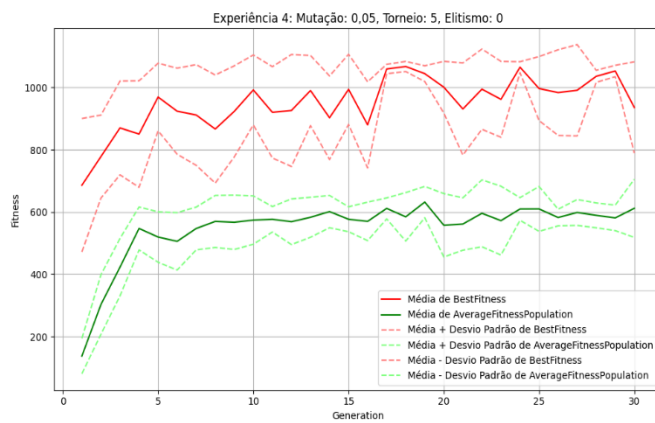
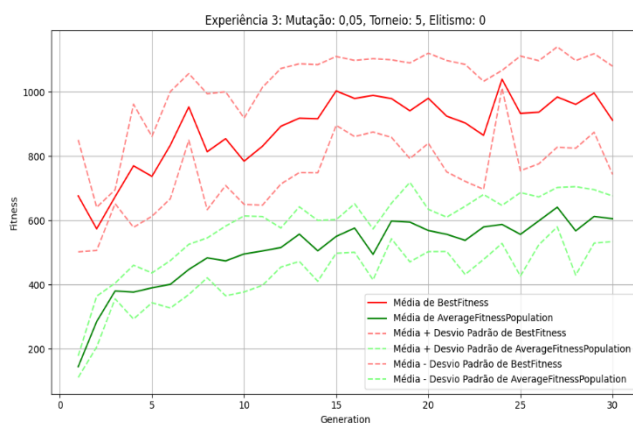
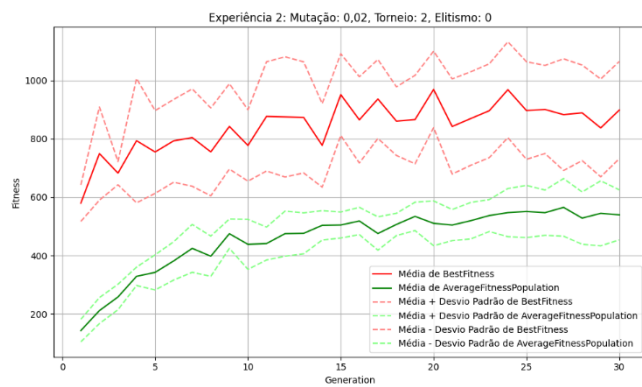
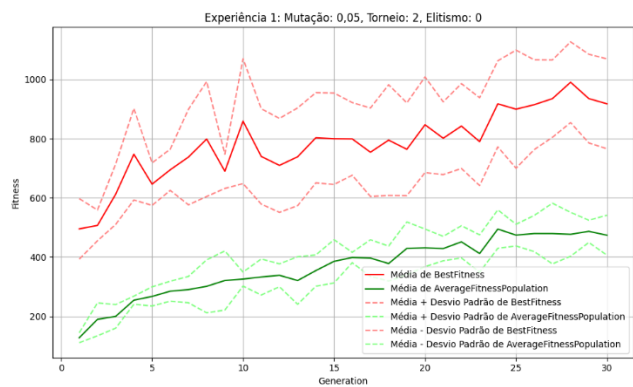
- A aptidão (fitness) de cada carro é então calculada com base em três componentes:
  - Média de Velocidade Multiplicada por 7: A média de velocidade é ponderada e multiplicada por 7. Isso destaca a importância da velocidade média no desempenho geral do carro.
  - Distância Percorrida Dividida por 5: A distância percorrida pelo carro é dividida por 5 e adicionada ao fitness. Isso incentiva os carros a percorrerem maiores distâncias.
  - Uma vez que o objetivo é terminar o percurso Completar o Percurso, adiciona-se um bônus de 200 ao fitness. Isso incentiva os carros a alcançarem o objetivo final.

#### 3. Atribuição da Fitness ao Carro:

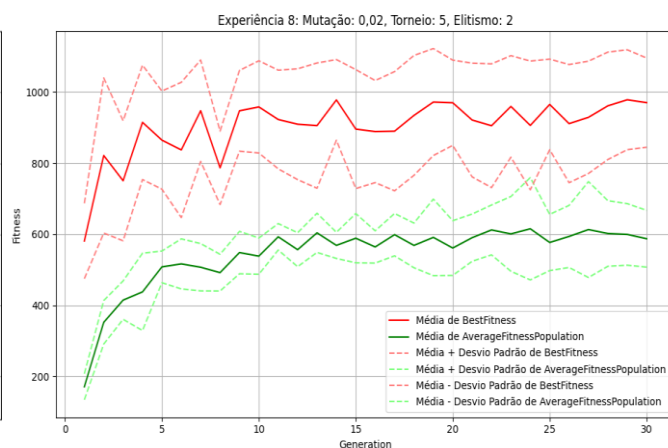
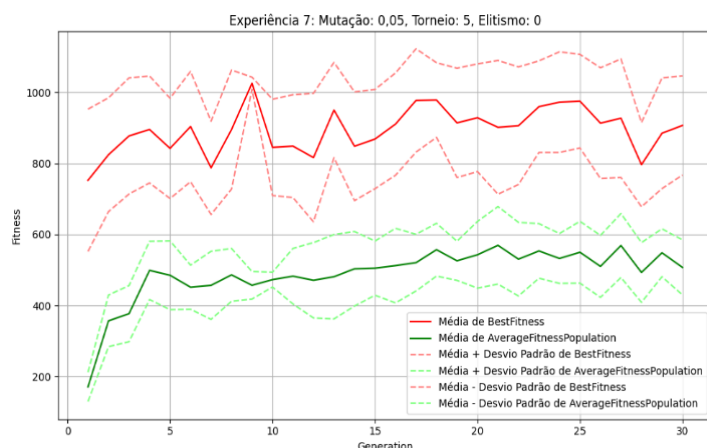
- O valor de aptidão calculado é então atribuído ao carro, o que o ajudará a ser selecionado como pai para a próxima geração no processo de reprodução do algoritmo genético.

$$fitness = mediumVelocity * 7 + distance / 5 + (roadCompleted ? 200 : 0);$$

Resumidamente, essa fórmula de fitness leva em consideração a velocidade média, a distância percorrida e se o carro conseguiu completar o percurso. Isso orienta a evolução da população de carros autônomos em direção a melhores desempenhos no percurso.







1. Com os resultados obtidos podemos ver que experimentos com uma alta taxa de mutação não mostram uma melhoria significativa, enquanto aqueles com uma baixa taxa de mutação exibem uma progressão mais consistente. Por exemplo, na experiência 1, mesmo que um carro tenha um desempenho excelente em uma geração, a próxima geração pode ter um desempenho significativamente pior. Essa tendência persiste ao longo dos experimentos, com uma evolução lenta. No entanto, na experiência 6, observou-se uma melhoria gradual, embora mais lenta do que na experiência 5, onde apenas a taxa de mutação difere. Isso ocorre porque, quando um indivíduo com bom desempenho é selecionado como progenitor, seus descendentes podem perder suas características favoráveis devido à aleatoriedade do algoritmo.

2. Por outro lado, experimentos que incorporam elitismo mostram uma progressão consistente ao longo do tempo. Na experiência 1, onde não há elitismo, a falta de preservação das soluções de alto desempenho resulta em uma falta de evolução. Já na experiência 5, onde o elitismo está presente, as boas soluções são preservadas e melhoradas por meio de mutações e recombinações entre progenitores. Este é o cenário mais promissor, pois sem elitismo, o algoritmo pode ficar preso em mínimos locais sub-ótimos. Quando as melhores soluções não são protegidas e mantidas, a busca pode se concentrar em áreas do espaço de solução que não são ideais, resultando em soluções de menor qualidade.

3. Portanto, podemos concluir que:

- Taxas de mutação mais altas tendem a destruir soluções promissoras.
- Taxas de mutação mais baixas permitem um refinamento gradual das soluções e facilitam a busca por soluções ótimas localmente.

- A presença de elitismo preserva e aprimora as soluções de maneira mais eficaz, levando a uma convergência mais rápida.

### 3.3. Segunda Iteração

Para melhorar a aptidão (fitness) com o objetivo de que os carros completem o percurso, considerou uma abordagem semelhante a fórmula anterior pois mostrou bons resultados, portanto, fatores relevantes, como a média de velocidade e um bônus significativo para os carros que completem o percurso. A ideia é criar uma fórmula que valorize a eficiência do percurso (alta velocidade e menor tempo) e recompense aqueles que conseguem completar o percurso

#### 1. Eficiência do Percurso:

- **distance \* 100 / (elapsedTime + 1)**: Essa parte da fórmula continua a valorizar a distância percorrida em relação ao tempo decorrido.

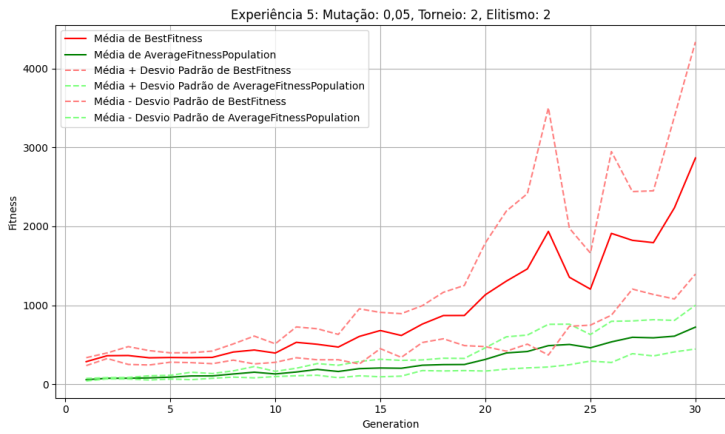
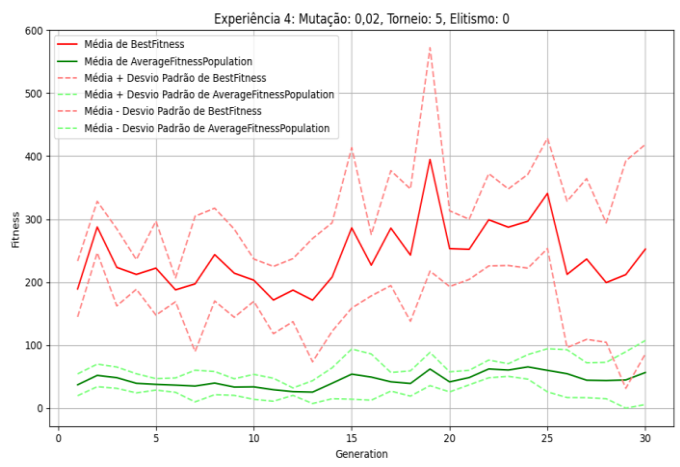
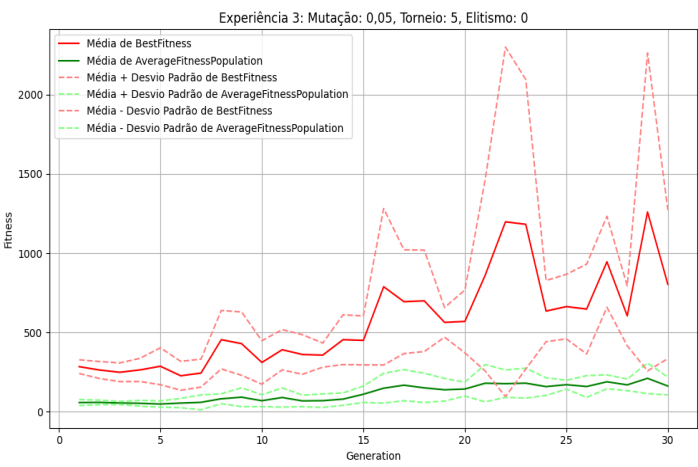
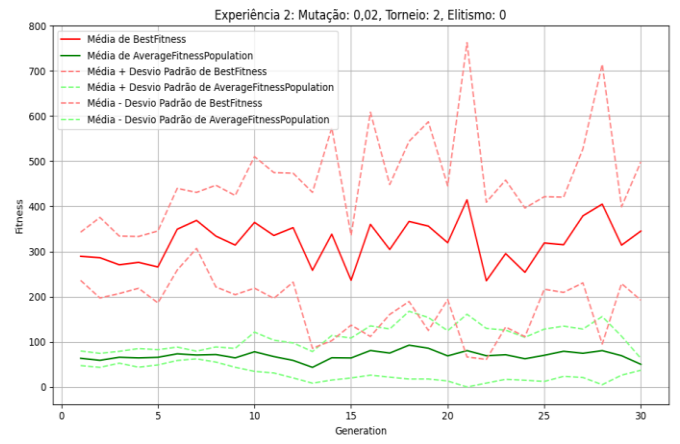
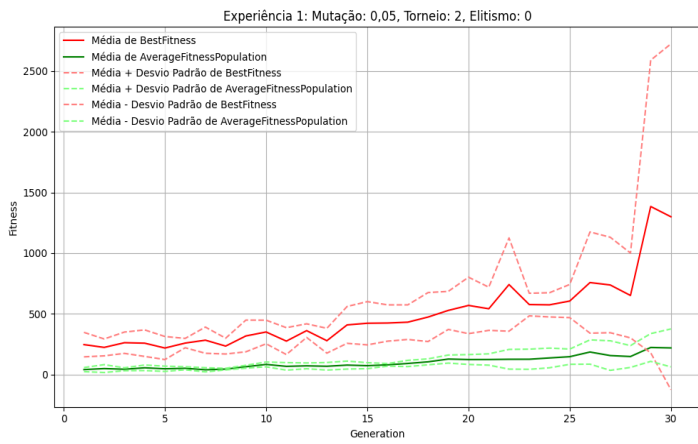
#### 2. Velocidade Média:

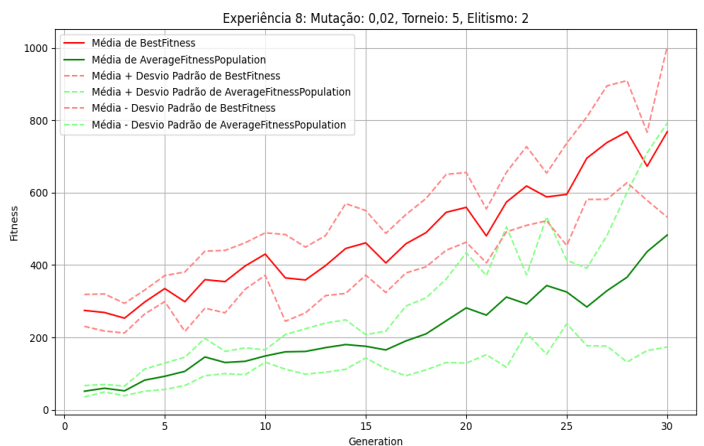
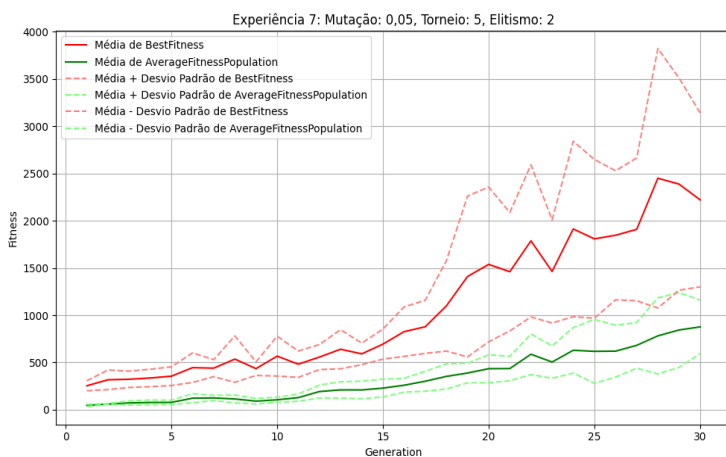
- **mediumVelocity \* 10**: Multiplicar a velocidade média por 10 dá peso significativo à média de velocidade do carro durante o percurso, incentivando carros mais rápidos.

#### 3. Bônus por Completar o Percurso:

- **(roadCompleted ? 500: 0)**: Se o carro completar o percurso, adiciona-se um bônus de 500 ao fitness. Esse bônus substancial incentiva fortemente a conclusão do percurso.

*float fitness = (distance \* 100 / (elapsedTime + 1)) + (mediumVelocity \* 10) + (roadCompleted ? 500 : 0)*





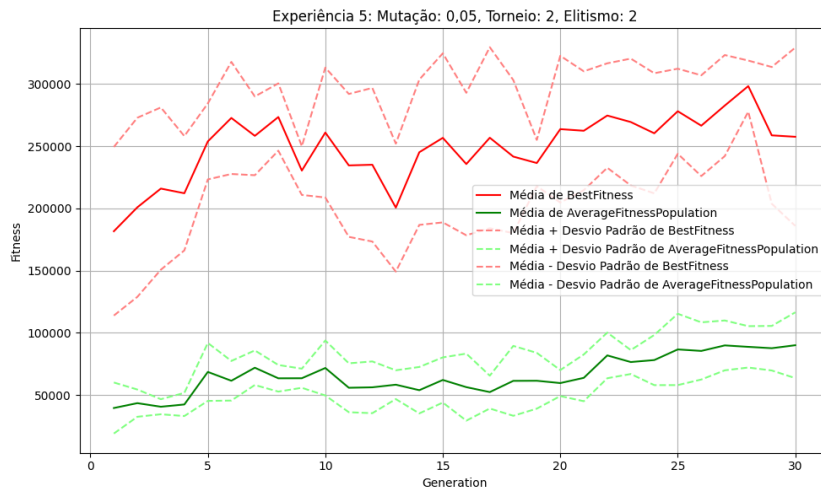
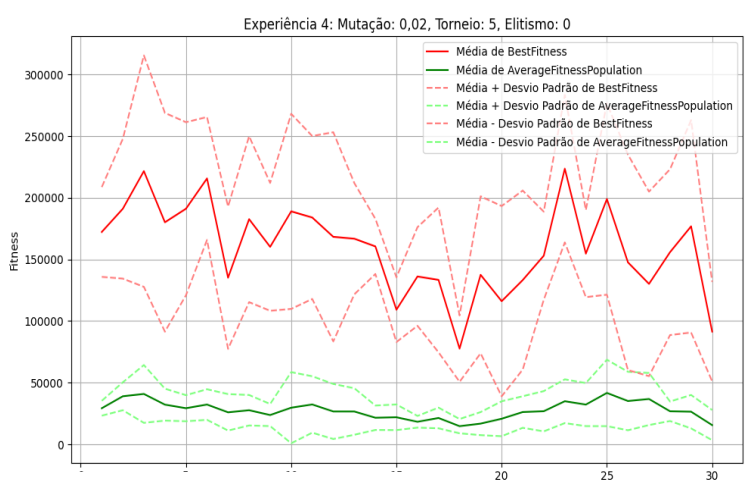
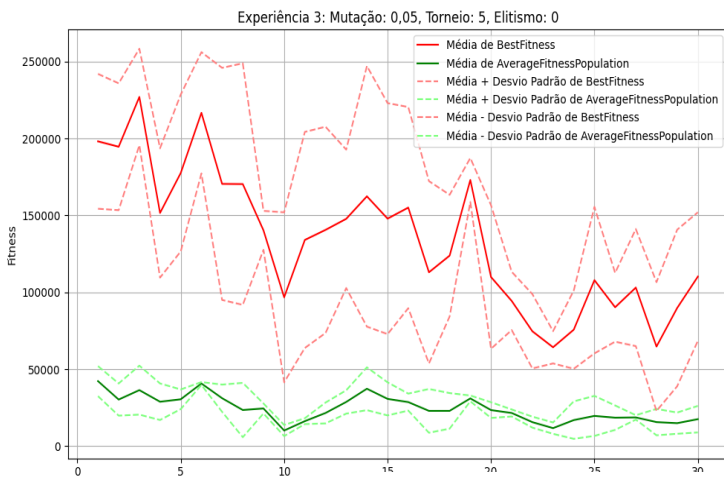
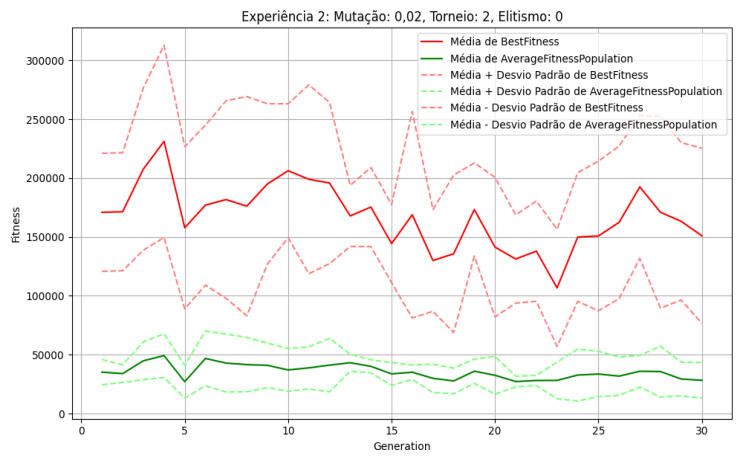
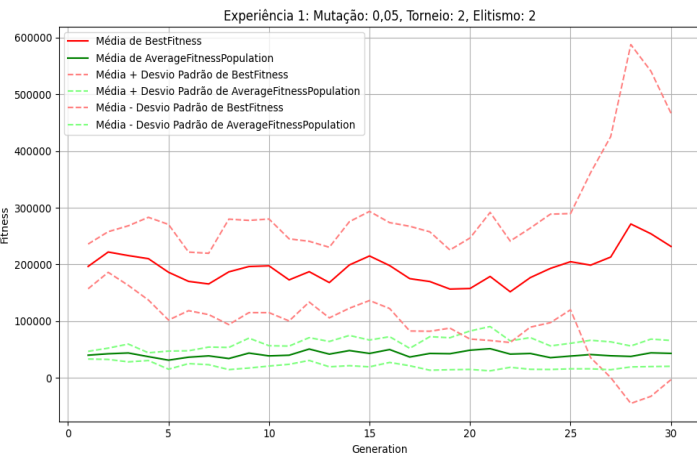
Ao longo de 30 gerações, como era de se esperar houve uma melhoria contínua na aptidão dos indivíduos. Inicialmente, haverá uma ampla exploração de soluções devido à alta diversidade genética. Conforme as gerações avançam, a seleção por torneio e o elitismo garantiram que os melhores indivíduos fossem preservados e reproduzidos, acelerando a convergência para soluções ótimas, um caso interessante foi que, na **vigésima sétima** geração da **segunda experiencia** todos os indivíduos chegaram ao fim.

### 3.4. Terceira Iteração

#### 3. Distância e Tempo ponderados pela Massa:

Na terceira abordagem, a fórmula de aptidão leva em consideração não apenas a relação entre distância e tempo, mas também a influência da massa do veículo. Ao adicionar a distância percorrida pelo tempo decorrido e dividir pelo valor da massa do veículo, essa fórmula prioriza veículos que conseguem percorrer distâncias maiores em menos tempo, levando em conta a eficiência energética e a agilidade. Veículos mais leves que alcançam distâncias maiores em menos tempo serão favorecidos, incentivando a busca por soluções que otimizem tanto a velocidade quanto a eficiência.

$$fitness = elapsedTime * 9 + distance / CarMass$$



Os valores de melhor aptidão, representados pelos picos no gráfico, aumentam ao longo das gerações. Isso sugere que o algoritmo está sendo eficaz na otimização da velocidade dos veículos. Em outras palavras, ao longo do tempo, os indivíduos mais aptos, ou seja, aqueles que podem

percorrer maiores distâncias em menos tempo, estão a ser preferencialmente selecionados e reproduzidos na população. Isso indica uma melhoria progressiva na performance dos veículos ao longo das iterações do algoritmo.

Ao longo das gerações, não apenas os melhores indivíduos estão a tornar-se mais eficientes, mas também há uma melhoria geral na qualidade da população como um todo. Isso pode ser resultado da seleção, onde os indivíduos mais aptos têm uma maior probabilidade de reprodução, levando a uma melhoria geral nas características da população.

No entanto, houve algumas flutuações nos valores da melhor aptidão ao longo das gerações. Essas flutuações podem ser atribuídas à aleatoriedade introduzida pela taxa de mutação e pelo processo de seleção, especialmente o tamanho do torneio. A taxa de mutação determina a probabilidade de ocorrerem mudanças aleatórias nos genes dos indivíduos, introduzindo assim variabilidade na população. Da mesma forma, o tamanho do torneio afeta a seleção dos pais para a reprodução, podendo resultar em uma variação nos descendentes e, consequentemente, nos valores de aptidão.

Essas flutuações são esperadas em algoritmos evolucionários devido à natureza estocástica do processo. No entanto, apesar das flutuações, a tendência geral de melhoria nos valores de aptidão ao longo das gerações indica que o algoritmo está a progredir na direção desejada, otimizando a performance dos veículos em relação ao critério de aptidão especificado.

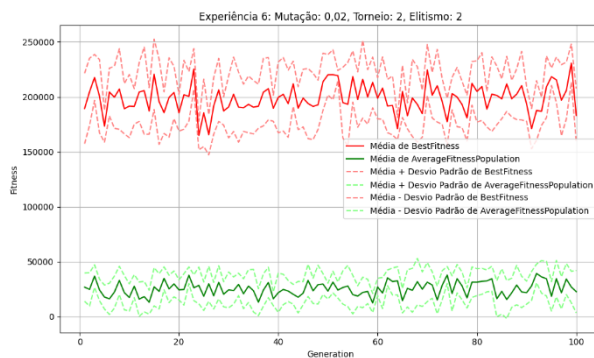
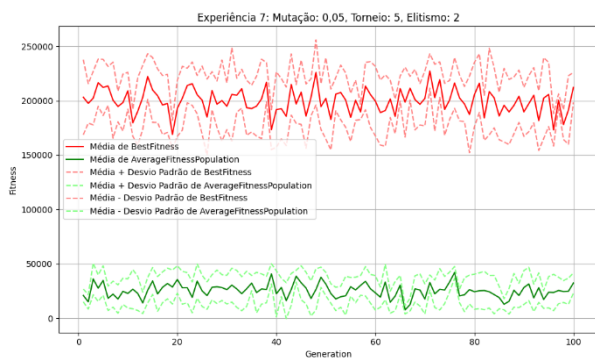
## 4. Experimentação e análise - HillRoad

### 4.1. Primeira Iteração

Após a análise e discussão das experiências realizadas no cenário GapRoad, começa-se a experimentação no cenário HillRoad. Decidiu-se inicialmente fazer testes a partir de uma função de aptidão que obteve bons resultados no cenário anterior, sendo esta a seguinte:

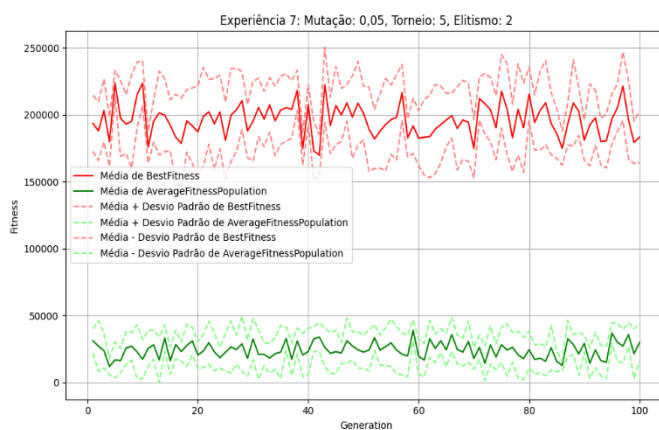
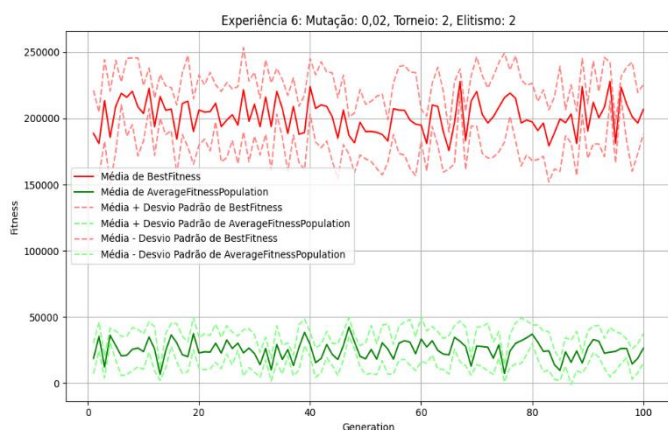
$$fitness = mediumVelocity * 7 + distance / 5 + (roadCompleted ? 200 : 0);$$

Em termos de parâmetros do algoritmo evolucionário utilizados para as experiências, foram utilizados os parâmetros iguais à experiência 6 e 7 do enunciado.



## 4.2. Segunda Iteração

uma abordagem de otimização para permitir que o algoritmo convirja para a solução ótima lentamente, de forma que aproveite o máximo de todas as soluções que possam ser ótimas decidiu-se fazer um teste com 100 gerações



Podendo analisar assim a minha intuição, se progredirmos lentamente consegue-se convergir para uma solução ótima. Algumas experiências precisavam de mais algumas gerações, tais como: experiência 6 e experiência 7, mas elas já estavam todas na zona de convergência. Ambas as experiências apresentam bons resultados, mas a que se destaca um pouco é a 6, ela consegue ter o seu average fitness da população a crescer em média.



## 5. Experimentação e análise - RockyHillRoad

### 5.1. Primeira Iteração

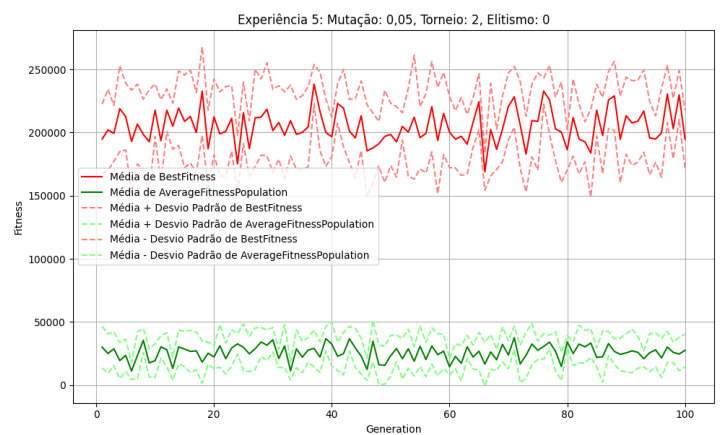
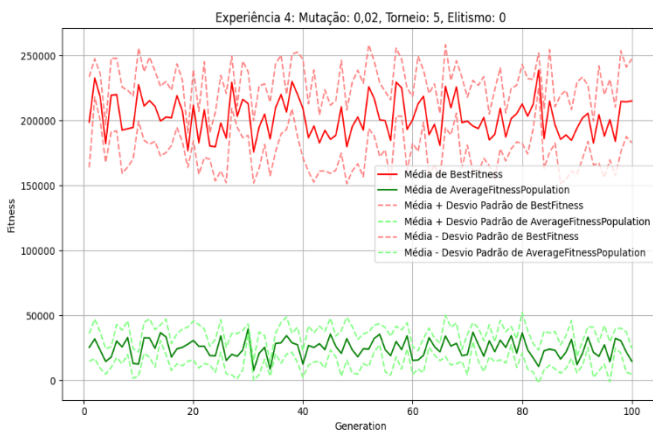
Depois de ter obtido bons resultados na experimentação do cenário **HillRoad**, decidiu-se começar a explorar o cenário **RockyHillRoad**.

Neste cenário, os carros enfrentam um desafio complicado, com um percurso cheio de obstáculos para ultrapassar. Para avaliar o desempenho destes agentes, considerou-se o seguinte:

$$fitness = distance / (elapsedTime + 1);$$

Esta fórmula considera a distância percorrida como o principal fator de avaliação, pois cada metro adicional percorrido contribui positivamente para a pontuação de aptidão. Além disso, o tempo é utilizado como um fator de normalização, de modo que agentes que percorram a mesma distância em menos tempo sejam considerados mais eficientes em relação aos obstáculos do percurso.

Esta fórmula reflete os critérios estabelecidos, priorizando tanto a distância percorrida quanto a eficiência em relação ao tempo.



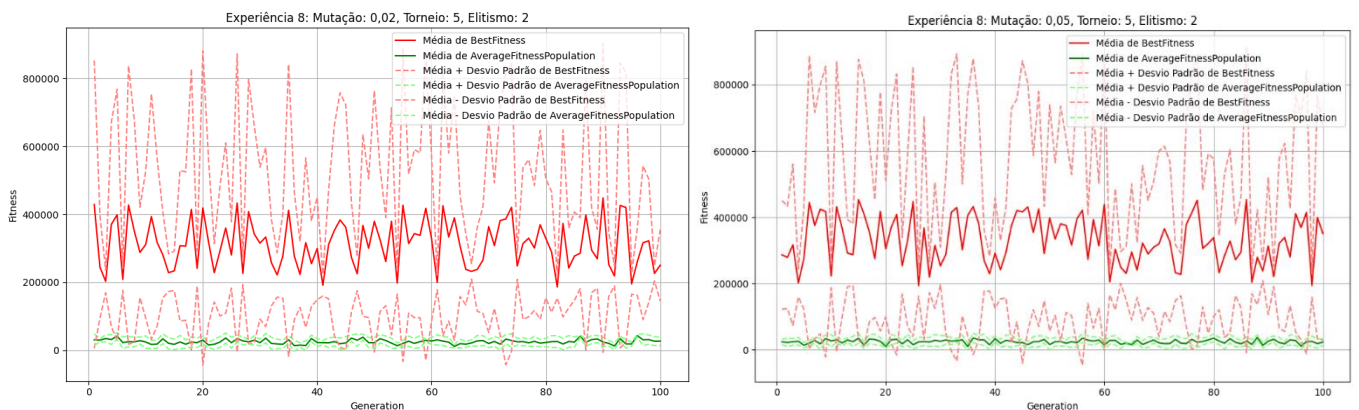


## 5.2. Segunda Iteração

Na segunda iteração, observou-se que os agentes que conseguiram chegar ao fim do percurso compartilhavam algumas características comuns. Esses carros eram geralmente baixos, rápidos e com um grande número de rodas, principalmente na parte da frente, o que os ajudava a superar os obstáculos de forma mais eficaz. Embora tivessem menos rodas no centro e na parte de trás, essas ainda desempenhavam um papel importante no equilíbrio do carro.

$$fitness = 6 \times distance + 2 \times sumVelocities + numberOfWheels$$

Com base nessas observações, exploraram-se várias funções de aptidão para avaliar o desempenho dos agentes. Entre elas, a que obteve os melhores resultados foi uma que priorizava a distância máxima percorrida pelos agentes, mas também atribuía peso significativo à velocidade máxima alcançada e ao número de rodas que os carros possuíam. Quanto mais rápido o carro conseguia ir e quantas mais rodas tinha, melhor era sua pontuação de aptidão, em linha com as características de um carro ideal discutidas anteriormente.



Nas experi ncias realizadas, ao longo de 100 gera  es, foi poss vel obter carros a chegar ao fim em 2 delas (experi ncias 8 e 5), n o tendo verificado nenhum carro a chegar na primeira experi ncia. Atrav s dos resultados obtidos percebeu-se uma evolu  o lenta em rela  o a dist ncia que os agentes conseguiam atingir ao longo das gera  es, tornando-se evidente que atrav s desta express o n o seria poss vel obter carros a chegar   meta nas 30 gera  es como aconteceu.

Depois de v rias tentativas, dos testes realizados com diferentes fun  es de aptid o, seguem-se as que n o resultaram em melhorias significativas em rela  o  s itera  es anteriores. Mesmo ap s milhares de gera  es, n o foi poss vel desenvolver agentes  timos para o cen rio em

questão. A evolução observada foi mínima ou quase inexistente, o que me levou a concluir que as funções de aptidão testadas não são pertinentes para esta aplicação.

<i><b>FintnessFuntions</b></i>
$\text{fitness} = ((\text{distance} / \text{elapsedTime}) * (1 / \text{sumForces}))$ $\text{fitness} = (\text{distance} * \text{elapsedTime}) / \text{carMass}$ $\text{fitness} = (\text{sumVelocities} / \text{elapsedTime}) * (\text{sumAccelerations} / \text{elapsedTime})$

*Tabela 1: Exemplos de funções que não obtiveram bons resultados*

<b>Experiência</b>	<b>(BestRoadCompleted)</b>	<b>(BestSumVelocities)</b>	<b>BestelapsedTime</b>
<b>5</b>	60%	595.33	173.25
<b>6</b>	70%	733.78	142.14

*Tabela 2: GapRoad*

<b>Experiência</b>	<b>(BestRoadCompleted)</b>	<b>(BestSumVelocities)</b>	<b>BestelapsedTime</b>
<b>6</b>	80%	612.33	133.31
<b>7</b>	85%	833.78	125.45

*Tabela 3: HillRoad*

<b>Experiência</b>	<b>(BestRoadCompleted)</b>	<b>(BestSumVelocities)</b>	<b>BestelapsedTime</b>

<b>4</b>	60%	789.33	132.22
<b>8</b>	83%	822.18	121.45

*Tabela 4: RockyHillRoad*

## Conclusão

Pode-se concluir que a taxa de mutação desempenha um papel crucial na evolução das soluções, no cenário "HillRoad " por exemplo. Quando a taxa de mutação é alta, observa-se que isso pode levar à destruição prematura de soluções promissoras. Isso ocorre porque as mutações excessivas podem introduzir mudanças drásticas nas soluções, levando a uma perda da informação útil acumulada ao longo das gerações.

Por outro lado, quando a taxa de mutação é baixa, permite-se um refinamento gradual das soluções. Isso significa que as mutações introduzidas são mais sutis e direcionadas, permitindo que o algoritmo explore o espaço de soluções de forma mais eficiente, buscando por soluções ótimas.

O elitismo, por sua vez, desempenha um papel importante na preservação de boas soluções. Ao permitir que as melhores soluções de cada geração sejam mantidas na próxima geração, o elitismo evita a perda prematura de soluções de alta qualidade e permite uma convergência mais rápida para soluções ótimas.

Uma particularidade que surge quando não há elitismo é a maior variabilidade genética entre as gerações sucessivas. Sem a preservação das melhores soluções, cada geração começa do zero, o que pode levar a uma maior diversidade de soluções, mas também pode resultar em uma busca menos eficiente pelo espaço de soluções. Por outro lado, quando há elitismo, as soluções de alta qualidade são mantidas, garantindo uma base sólida para a evolução contínua das soluções.

Através da aplicação destes algoritmos, é possível resolver uma ampla gama de problemas. Permitiu explorar soluções em um espaço de busca vasto e multidimensional, para além de se adaptarem a mudanças no ambiente e evoluírem gradualmente em direção a soluções ótimas.

## Referências

- [1] Recombinação (computação evolutiva) em: [Recombinação \(computação evolutiva\) – Wikipédia, a enciclopédia livre \(wikipedia.org\)](https://pt.wikipedia.org/wiki/Recombina%C3%A7%C3%A3o_(computa%C3%A7%C3%A3o_evolutiva))
- [2] Natsuki Higashi e Hitoshi Iba. «Particle swarm optimization with Gaussian mutation». Em: Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706). IEEE. 2003, pp. 72–79.