

# RELATORIO PROGRAMAÇÃO GENÉTICA

**GRUPO: Matheus Ribas, Marco Decco, Lucas Picanço, Rafael Portugal**

## **Introdução**

Nesse trabalho de Programação Genética, nosso objetivo foi desenvolver e aprimorar um robô que consegue se locomover de forma autônoma em um ambiente aleatório 2D cheio de obstáculos, recursos e uma meta final. A proposta era fazer com que o robô aprendesse sozinho, através de um processo de evolução, a melhor maneira de se movimentar, desviar dos obstáculos, coletar os recursos e, só então, ir até a meta.

Durante os testes iniciais, percebemos que o robô tinha dificuldades em se locomover corretamente. Ele ficava preso em obstáculos, girando no mesmo lugar sem conseguir se desvencilhar. A partir disso, começamos a implementar melhorias no algoritmo genético, fazendo com que ele entendesse que ganhar recompensas por coletar recursos e receber penalidades quando não cumpria corretamente as tarefas era essencial para seu aprendizado. Com isso, o robô começou a evoluir, tomando decisões mais inteligentes a cada geração.

## **O que foi feito**

A primeira etapa foi entender como o código base funcionava. A partir daí, percebemos que o robô estava tomando decisões muito simples e, muitas vezes, não conseguia se comportar de forma inteligente no ambiente. Então começamos a aplicar várias melhorias na parte do algoritmo genético, que é responsável por fazer a evolução das estratégias de controle do robô, adicionamos logicas e verificamos como o robô se comportava diante de variações de números novas, adicionamos verificações constantes para ver se ajudava ou atrapalhava o robô a achar os caminhos dos recursos e atingir a meta, penalizamos ele de forma mais rude para aprender o que não deve ser feito e adicionamos bônus para quando fizesse as coisas de maneira correta, nosso grupo também acrescentou ilhas para poder criarmos um sistema onde a população se dividisse de forma separada para aprender coisas novas da população inicial, adicionamos função para injetar periodicamente a cada 3 gerações diversidades na população para sempre estar atingindo uma ótima resolução

## O que foi alterado

- Adicionamos novos operadores matemáticos, como seno (sin), cosseno (cos) e logaritmo seguro (log\_safe). Isso foi feito porque percebemos que o robô precisava tomar decisões mais suaves e que dependiam de comportamentos cíclicos, como girar de maneira mais precisa ou procurar um recurso que estivesse mais acessível.
- Implementamos um sistema de ilhas, onde a população é dividida em pequenos grupos que evoluem separadamente. Isso ajuda a evitar que todo mundo aprenda a mesma coisa e fique preso em uma solução ruim.
- Também criamos um sistema de migração, onde de tempos em tempos os melhores indivíduos de uma ilha vão para outra, ajudando a espalhar boas soluções.
- A função de fitness, que é basicamente o que diz se um robô é bom ou ruim, foi completamente reformulada. Agora ela leva em conta se o robô coletou todos os recursos antes de ir para a meta, se bateu em obstáculos, se gastou muita energia, se ficou parado e quanto ele andou no mapa, isso nos ajudou entender o que podia ser alterado para fazer o robô ter decisões melhores em cima de cada ambiente e obstáculo.
- Aumentamos a profundidade das árvores de decisão, que são as estruturas que definem como o robô pensa. Antes, elas eram muito rasas e limitadas, incluímos a profundidade 5 para fazer o robô tomar decisões mais complexas para resolver desafios, mas também pensando em evitar overfitting (sobrecarga de decisões)
- Aumentamos também a taxa de mutação, porque percebemos que isso ajuda o robô a explorar mais possibilidades diferentes durante a evolução e evitando que ele ache uma solução boa e fique apenas por ela mesmo.

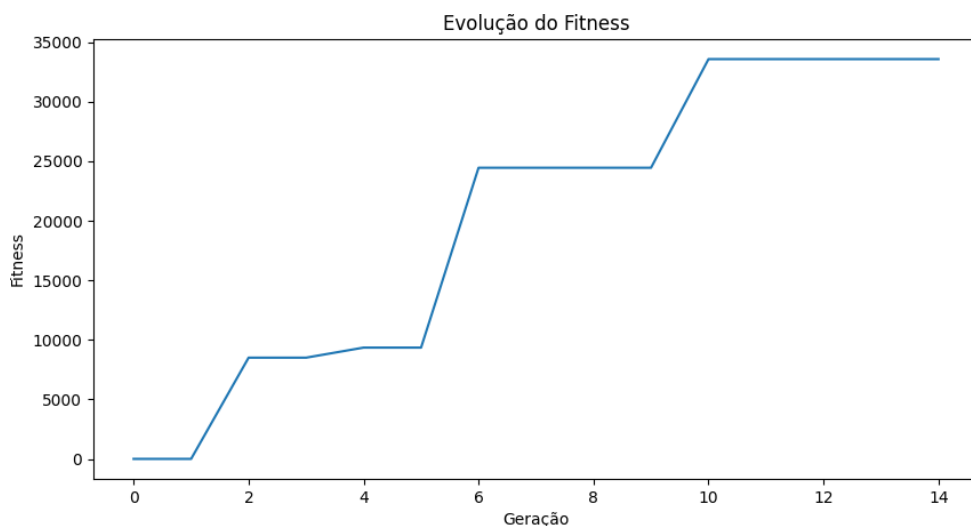
## Por que foi alterado

Essas mudanças foram necessárias porque, durante os testes iniciais, o robô frequentemente ficava preso, girando no mesmo lugar ou tentando ir direto para a meta sem pegar os recursos, o que não era o comportamento que queríamos. Também percebemos que, às vezes, a população inteira aprendia um comportamento ruim e ficava difícil sair dele. Foi aí que surgiu a ideia de usar ilhas e forçar mais diversidade com mais mutações, durante o processo fizemos o robô entender que ele não podia ficar batendo no mesmo obstáculo sem achar formas de sair do mesmo, o robô conseguiu compreender tal ato e melhorou suas decisões em desafios complexos que eram gerados de forma aleatória.

No fim, as melhorias fizeram muita diferença. O robô passou a conseguir pegar os recursos de forma mais eficiente, desviar melhor dos obstáculos e só então ir até a meta, tudo isso com menos colisões e gastando menos energia.

## Resultados

O gráfico abaixo mostra como o fitness foi evoluindo geração após geração. Dá para perceber claramente que, conforme as gerações foram passando, os indivíduos foram ficando cada vez melhores, até chegarem em um ponto de estabilidade onde o comportamento já estava bem otimizado, utilizamos 15 gerações pois 5 era muito limitado e não dava tempo ao robô aprender adequadamente a tomar novas decisões.



## Links Importantes

**Vídeo do robô funcionando:**

<https://youtu.be/xEIEjIOH38E>

**Repositório no GitHub:**

[https://github.com/MarcoDecco/Robo\\_Programacao\\_Genetica](https://github.com/MarcoDecco/Robo_Programacao_Genetica)

**Obs: Todos commits por parte do grupo foi feito em Branch, somente o melhor código foi commitado na master.**

## Conclusão

Esse trabalho foi uma ótima experiência para entender na prática como algoritmos evolutivos podem ser aplicados para resolver problemas complexos. Foi muito interessante ver como, a partir de soluções completamente aleatórias, o robô foi aprendendo comportamentos cada vez mais inteligentes. O uso de estratégias como ilhas, migração e uma boa definição de fitness foi essencial para conseguir alcançar bons resultados, conseguimos entender e botar em

pratica o conteúdo de programação genética, verificando em cada novo código um novo aprendizado tanto para o robô e para os integrantes do grupo, foi um trabalho bastante recompensador tanto para aprendizado mas também em questão de colaboração ,onde todos integrantes estiverem presentes e executaram testes frequentes para chegarmos em uma melhor resposta juntos.