

ALGORITMO GENÉTICO PARA RESOLUÇÃO DE UM JOGO DE LABIRINTO

Cintia Zago, Guilherme Machado, Rodrigo Vidal

cintiazago@gmail.com / 26guilhermemachado@gmail.com / rodrigo.vidal@fbsbrasil.com.br

FIAP – Faculdade de Informática e Administração Paulista
São Paulo, Brasil

Resumo — este documento apresenta a implementação e simulação de um algoritmo genético na resolução de um jogo de labirinto utilizando Python. Para isso, foram implementados diversos métodos que caracterizam o algoritmo genético, tais como a função de fitness, de cruzamento, mutação e seleção. Para simular os resultados obtidos, foi utilizada a biblioteca Pygame para visualizar o caminho percorrido pelo jogador com maior pontuação após determinado número de gerações.

Keywords— *python; algoritmos genéticos; inteligência artificial;*

I. INTRODUÇÃO

Os algoritmos genéticos são métodos de otimização inspirados no processo de seleção natural e evolução das espécies. Baseados em princípios como a seleção dos mais aptos, cruzamento e mutação, esses algoritmos, de forma iterativa, evoluem uma população de soluções candidatas buscando melhorar a cada geração.

Neste artigo, exploraremos a aplicação de algoritmos genéticos na resolução de um problema de labirinto, utilizando a biblioteca *Pygame* para a simulação. O algoritmo foi capaz de evoluir jogadores virtuais que encontraram a saída do labirinto após uma série de gerações. Este breve estudo não só ilustra a eficácia dos algoritmos genéticos em ambiente de navegação complexa, mas também abre caminho para futuras aplicações em áreas como robótica, jogos e otimização de rotas.

II. DESCRIÇÃO DO PROBLEMA E METODOLOGIA

O problema abordado neste artigo consiste em guiar o jogador virtual até a saída de um labirinto, representado em uma matriz bidimensional, onde cada elemento pode ser um caminho livre, uma parede ou a saída. O objetivo é evoluir uma sequência de comandos que permita ao jogador, partindo de uma determinada posição inicial, alcançar a saída. Para isso, utilizamos a biblioteca *Pygame* para simular o ambiente e as movimentações do jogador.

A metodologia segue os padrões da definição do algoritmo genético. Primeiro, inicia-se uma população inicial de indivíduos (jogadores), onde cada qual é atribuído uma série inicial de comandos aleatórios. Cada jogador é avaliado com base em uma função *fitness* que considera a distância do jogador até a saída e penalidades para colisões com as paredes. O algoritmo evolui a população ao longo de gerações,

aplicando os operadores de seleção, cruzamento e mutação de forma iterativa

1. População Inicial

Cada jogador é uma instância da classe *Player*, contendo uma lista de comandos aleatórios (esquerda, direita, cima e baixo).

2. Seleção

Os jogadores são ordenados por sua pontuação de *fitness*, e apenas os melhores são selecionados para reprodução.

3. Cruzamento

Dois pais são selecionados aleatoriamente entre os melhores jogadores, e um ponto de cruzamento é escolhido para gerar dois novos jogadores (filhos), combinando os comandos dos pais.

4. Mutação

Cada novo jogador pode sofrer uma mutação com uma probabilidade pré-definida, onde um comando aleatório da lista de comandos pode ser alterado para outra direção aleatória ou ainda adicionar um novo comando.

5. Fitness

A função de fitness avalia a trajetória do jogador com base em quão próximo ele chega da saída do labirinto, penalizando colisões com as paredes e recompensando a proximidade com a distância da saída. Essa distância é calculada utilizando um algoritmo de busca em largura (BFS).

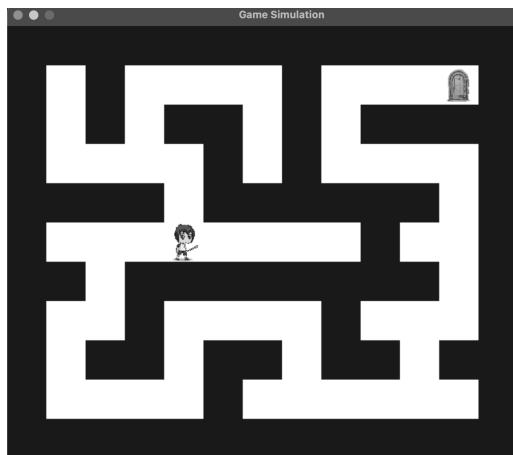
III. RESULTADOS E DISCUSSÃO

Ao final de várias gerações, o algoritmo convergiu para uma solução que guia o jogador até o final do labirinto, demonstrando a eficácia dos algoritmos genéticos na resolução de problemas complexos de navegação.

Para fins de demonstração, foi realizado um teste com população inicial de 100 jogadores e uma taxa de mutação de 99%, utilizando o labirinto mais complexo. Após cerca de 690 gerações, o algoritmo convergiu para uma solução que levava o jogador para a saída.

A evolução da população ao longo das gerações mostrou um aumento gradual na pontuação média de *fitness*, indicando que os jogadores estavam ficando progressivamente mais próximos da saída do labirinto. Inicialmente, muitos destes jogadores colidiram com as paredes e ficavam presos em loops, mas com a aplicação dos operadores de seleção, cruzamento e mutação, as trajetórias dos jogadores foram melhorando significativamente.

Além de tudo, observou-se que a alta taxa de mutação foi crucial para evitar o problema de convergência que ocorria muito rápido e para ampliar o espaço de soluções possíveis. Porém, isto resultou em uma maior variação nas trajetórias dos jogadores e exigiu mais gerações para alcançar o objetivo final que era a rota mais otimizada. No entanto, segundo *Goldergerg (1989)*⁴, estas taxas de mutação elevadas podem ajudar, mas devem ser equilibradas para evitar flutuações excessivas, pois mesmo melhorando a diversidade genética da população evitando que se tornem muito semelhantes, isso pode levar à destruição de soluções válidas, havendo o risco de se modificar resultados que já são bons destruindo estas soluções e tornando o processo de busca menos eficiente.



IV. APLICAÇÕES E FUTURAS EXTENSÕES

Os resultados obtidos demonstram que algoritmos genéticos são uma ferramenta poderosa para ajudar a resolver problemas de navegação em ambientes complexos, como labirintos. Esta abordagem pode ser aplicada também em diversas outras áreas, como:

1. Mapeamento de Rotas

Nesta área, algoritmos genéticos podem ser utilizados para encontrar rotas mais eficientes, por exemplo, em sistemas de transporte público, logística de entregas e navegação de veículos autônomos (*Holland, 2022*)¹.

2. NPC's de Jogos

Em jogos eletrônicos, NPCs (personagens não jogáveis) podem usar estes algoritmos para melhorar suas habilidades tanto de navegação quanto de comportamento, o que pode torná-los mais desafiadoras e realistas, como pode ser analisado no estudo de *Mitchell (1998)*² sobre algoritmos

genéticos para desenvolver estratégias para o dilema do prisioneiro.

3. Navegação de Robôs

Robôs móveis podem utilizar algoritmos genéticos para explorar e mapear os ambientes, encontrar caminhos otimizados e ainda evitar obstáculos de forma autônoma.

4. Otimização de Rotas

Operações industriais e armazéns pode se beneficiar dos algoritmos genéticos para otimizar as rotas para coleta e entrega de produtos, reduzindo, assim, custos operacionais e proporcionando mais agilidade para seus clientes, que está relacionado ao problema do Caixeiro Viajante descrito por *Michalewicz (1996)*³.

Futuras extensões deste trabalho, pode-se considerar a implementação de técnicas híbridas que combinam algoritmos genéticos com outros métodos de inteligência artificial, e melhorar ainda mais os resultados obtidos. Além disso, é possível explorar diferentes representações de indivíduos e operadores genéticos para descobrir novas resoluções de problemas complexos de navegação.

V. CONCLUSÃO

Este estudo apresentou a aplicação de um algoritmo genético na resolução de um jogo de labirinto, utilizando a biblioteca Pygame para visualização. Os resultados mostraram que o algoritmo genético foi eficaz em encontrar uma solução para guiar o jogador até a saída do labirinto, mesmo em um ambiente de navegação complexa. A alta taxa de mutação utilizada no experimento foi crucial para evitar a convergência prematura e para explorar melhor o espaço de soluções possíveis.

A análise dos resultados demonstrou que, ao longo de várias gerações, a população de jogadores evoluiu significativamente, melhorando suas trajetórias e aumentando suas pontuações de *fitness*. Este trabalho ilustra o potencial dos algoritmos genéticos não apenas para resolver problemas de labirinto, mas também para aplicações mais amplas em mapeamento de rotas, jogos eletrônicos, navegação de robôs e otimização de rotas industriais.

Futuras extensões deste trabalho podem incluir a integração de técnicas híbridas de inteligência artificial, como aprendizado por reforço, para aumentar ainda mais a eficiência e eficácia das soluções. Além disso, a exploração de diferentes representações de indivíduos e operadores genéticos pode proporcionar novas abordagens para resolver problemas de navegação complexa.

Assim, este estudo contribui para uma compreensão das capacidades e limitações dos algoritmos genéticos em ambientes de navegação e abre caminho para novas pesquisas práticas.

REFERÊNCIAS

- [1] Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. MIT Press.
- [2] Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. MIT Press.
- [3] Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer. Chapter 10.
- [4] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley. Chapter 4.