

Desenvolvimento de um sistema de inteligência artificial para obter altas pontuações no Tetris utilizando algoritmo genético

Pedro Baldotto¹, Rafael Crevelari¹

¹Departamento de Informática – Universidade Federal de Viçosa (UFV)
Viçosa – MG – Brasil

pedro.baldotto@ufv.br, rafael.crevelari@ufv.br

Abstract. *Final project presented to the subject Artificial Intelligence I, code INF 420, taught by Professor Julio C. S. Reis, as a partial requirement for approval in the subject.*

Keywords: *Software development, Genetic Algorithm, Artificial intelligence, Tetris.*

Resumo. *Projeto final apresentado à disciplina de Inteligência Artificial I, de código INF 420, ministrada pelo professor Julio C. S. Reis, como requisito parcial para aprovação na disciplina.*

Palavras-chave: *Desenvolvimento de Software, Algoritmo Genético, Inteligência Artificial, Tetris.*

1. Introdução

O objetivo deste projeto foi desenvolver um sistema que possa jogar o videogame Tetris de forma autônoma, buscando obter bom desempenho e grandes pontuações no jogo. A implementação da inteligência artificial envolveu a utilização de Algoritmos Genéticos, o que permitiu a ocorrência de aprendizado a partir da interação de diversos indivíduos do algoritmo com o ambiente do jogo. Essa abordagem fez com que estratégias promissoras fossem perpetuadas em nossas soluções.

2. Metodologia

A técnica de IA explorada em nosso trabalho foi um Algoritmo Genético. De forma sucinta, Algoritmos Genéticos são uma classe de algoritmos de otimização que foram inspirados no processo de evolução natural, pois utilizam conceitos como seleção natural, cruzamento de indivíduos e mutação na construção de soluções.

2.1. Sobre o jogo

Em nossa implementação, o jogo consiste em um tabuleiro de dimensões 10x20 em que constantemente existe um bloco em queda livre. Ao aterrissar no fundo do tabuleiro, ou em cima de outro bloco, o bloco que estava caindo é travado em sua posição. Caso haja completude de espaços do tabuleiro preenchidos nas linhas horizontais, estas serão limpas e a pontuação aumentará (100 pontos para uma linha, 250 para duas, 400 para três e 1000 para quatro). Caso o topo do tabuleiro seja atingido, o jogo acaba. Existem sete tipos diferentes de blocos, que podem sofrer rotações de 90 graus no sentido horário.

2.2. Representação dos Indivíduos

Como mencionado anteriormente, utilizamos a técnica de algoritmo genético para abordar o problema em questão. Assim como em qualquer algoritmo genético, a primeira etapa crucial é criar um modelo inicial de um indivíduo da população, especificando quais características (genes) ele deve possuir.

Nesse contexto, identificamos as seguintes características que consideramos relevantes e fundamentais para a tomada de decisões de um indivíduo durante uma jogada, a fim de caracterizá-lo: peso para o número de buracos, peso para a rugosidade do tabuleiro, peso para as linhas limpas no topo, peso para as linhas limpas na base, peso para a altura e peso para as sombras.

Assim, cada indivíduo possui um valor que representa a "importância" atribuída a essas características, denominado de peso, definido em uma escala de 0 a 1. Para ilustrar, apresentamos um exemplo de um indivíduo gerado por nosso algoritmo genético:

- **Peso para o Número de Buracos:** 0.5764343247818454
- **Peso para a Rugosidade:** 0.09973954062855961
- **Peso para as Linhas Limpas no Topo:** 0.7934715772414589
- **Peso para as Linhas Limpas na Base:** 0.16871021668664132
- **Peso para a Altura:** 0.03334951194142333
- **Peso para as Sombras:** 0.951676843485697

2.3. Função de Avaliação

Tendo exposto as características, faz-se necessário explicar a tomada de decisão para realizar uma jogada de um indivíduo. Dada uma nova peça gerada aleatoriamente no tabuleiro, são testadas todas as jogadas simples (que consistem em um número de movimentos horizontais em uma mesma direção, seguidos da quantidade máxima de movimentos para baixo possível). Cada uma dessas jogadas resulta em um tabuleiro diferente que é passado juntamente com o indivíduo que está jogando no momento como parâmetros para a função de avaliação, que foi definida como:

$$\begin{aligned} F = & (Lt \cdot \text{indivíduo.pesoLinhasLimpasTopo}) + (Lb \cdot \text{indivíduo.pesoLinhasLimpasEmbaixo}) \\ & - (Br \cdot \text{indivíduo.pesoNumBuracos}) - (Rg \cdot \text{indivíduo.pesoRugosidade}) \\ & - (At \cdot \text{indivíduo.pesoAltura}) - (Sm \cdot \text{indivíduo.pesoSombras}) \end{aligned} \quad (1)$$

Sabendo do formato da função, é válido descrever os valores utilizados no cálculo obtidos a partir do tabuleiro passado como parâmetro.

- **Número de Buracos (Br):** é o número de posições no tabuleiro que não podem ser acessadas em uma próxima jogada, isto é, não estão preenchidas com nenhum bloco mas estão cercadas.
- **Rugosidade (Rg):** refere-se ao grau de irregularidade em que se encontra o tabuleiro. O cálculo é feito a partir da soma das diferenças de alturas entre as colunas do tabuleiro.
- **Linhas Limpas no Topo (Lt):** são todas as linhas que foram limpas na metade superior do tabuleiro pela jogada.

- **Linhas Limpas na Base (Le):** são todas as linhas que foram limpas na metade superior do tabuleiro pela jogada.
- **Altura (At):** é a altura máxima entre as 10 colunas do tabuleiro.
- **Sombras (Sm):** refere-se a posições no tabuleiro que estão vazias, mas estão bloqueadas por um segmento de bloco posicionado em algum lugar acima (os buracos não são incluídos na contagem).

Observação: Consideramos as características Sombras, Altura, Rugosidade e Número de Buracos como características negativas, que devem ser evitadas. Portanto, elas contribuem para a diminuição do valor da função de avaliação.

Após executar todas as jogadas de teste, será selecionada aquela que apresentar o maior resultado na função de avaliação.

2.4. Seleção de Indivíduos

Considerando uma população de P indivíduos gerados com genes totalmente aleatórios, é necessário estabelecer um número J de jogos a serem realizados por cada indivíduo durante sua geração.

Após os jogos de um determinado indivíduo, é calculada a média de pontos obtidos, onde S é a soma de pontos obtidos nos J jogos do indivíduo:

$$\text{Média} = \frac{S}{J} \quad (2)$$

Apenas P / 2 indivíduos com as melhores médias de pontos serão selecionados para o processo de reprodução, enquanto os demais serão descartados.

Será arquivado os genes do melhor indivíduo dessa geração, ou seja, o que obtiver maior média de pontos.

2.5. Reprodução de Indivíduos

Após a seleção dos melhores indivíduos, temos uma lista de indivíduos para realizar a reprodução. Cada indivíduo na lista irá se reproduzir com um outro indivíduo escolhido aleatoriamente da mesma lista, garantindo que um indivíduo não se reproduza consigo mesmo.

Durante o processo de reprodução, um novo filho é gerado. Para cada gene do filho, é feita uma escolha aleatória entre o gene do pai e o gene da mãe. Essa escolha determina qual característica será herdada pelo filho em relação a cada atributo específico. Portanto, os genes do filho são compostos por uma combinação aleatória dos genes dos pais.

Aqui está um exemplo de reprodução simulada para melhorar o entendimento:

- **Indivíduo 1:**
 - **Peso para o Número de Buracos:** 0.5764343247818454
 - **Peso para a Rugosidade:** 0.09973954062855961
 - **Peso para as Linhas Limpas no Topo:** 0.7934715772414589
 - **Peso para as Linhas Limpas na Base:** 0.16871021668664132

- **Peso para a Altura:** 0.03334951194142333
- **Peso para as Sombras:** 0.951676843485697
- **Indivíduo 2:**
 - **Peso para o Número de Buracos:** 0.7812490128283374
 - **Peso para a Rugosidade:** 0.2538444036364491
 - **Peso para as Linhas Limpas no Topo:** 0.6281433973694106
 - **Peso para as Linhas Limpas na Base:** 0.8329312635668014
 - **Peso para a Altura:** 0.03950461971270591
 - **Peso para as Sombras:** 0.68991344117441
- **Filho Gerado:**
 - **Peso para o Número de Buracos:** 0.7812490128283374 (*herdado do Indivíduo 2*)
 - **Peso para a Rugosidade:** 0.09973954062855961 (*herdado do Indivíduo 1*)
 - **Peso para as Linhas Limpas no Topo:** 0.7934715772414589 (*herdado do Indivíduo 1*)
 - **Peso para as Linhas Limpas na Base:** 0.8329312635668014 (*herdado do Indivíduo 2*)
 - **Peso para a Altura:** 0.03950461971270591 (*herdado do Indivíduo 2*)
 - **Peso para as Sombras:** 0.68991344117441 (*herdado do Indivíduo 2*)

Além disso, é importante mencionar que, após a reprodução, o filho ainda está sujeito a um processo de mutação.

2.6. Mutação de Indivíduos

A mutação introduz pequenas alterações aleatórias nos genes (alterando o valor do gene, para um novo número totalmente aleatório) do filho, ajudando a explorar novas soluções e evitar a estagnação da população. A taxa de mutação, que deve ser definida em uma população, determina a probabilidade de ocorrência dessas alterações genéticas.

Aqui está um exemplo de mutação simulada para melhorar o entendimento:

- **Filho Gerado:**
 - **Peso para o Número de Buracos:** 0.7812490128283374
 - **Peso para a Rugosidade:** 0.09973954062855961
 - **Peso para as Linhas Limpas no Topo:** 0.7934715772414589
 - **Peso para as Linhas Limpas na Base:** 0.8329312635668014
 - **Peso para a Altura:** 0.03950461971270591
 - **Peso para as Sombras:** 0.68991344117441
- **Filho com Mutação:**
 - **Peso para o Número de Buracos:** 0.7812490128283374
 - **Peso para a Rugosidade:** 0.09973954062855961
 - **Peso para as Linhas Limpas no Topo:** 0.03950461971270591 (*Gene com Mutação*)
 - **Peso para as Linhas Limpas na Base:** 0.8329312635668014
 - **Peso para a Altura:** 0.03950461971270591
 - **Peso para as Sombras:** 0.68991344117441

2.7. Nova População

Após gerar todos os filhos, teremos uma nova população formada pelos pais e seus respectivos filhos. Essa nova população será submetida novamente às etapas mencionadas anteriormente, ou seja, seleção, reprodução e mutação. Esse processo se repete até que um número de gerações definido pelo usuário seja alcançado.

Ao final do processo, teremos uma lista de melhores indivíduos e seus respectivos conjuntos de genes em cada geração. Isso permite acompanhar a evolução da população ao longo do tempo e identificar quais indivíduos apresentam características mais vantajosas para a solução do problema em questão.

3. Resultados

Após a realização de uma extensa bateria de testes para avaliar a eficiência do nosso método de inteligência artificial proposto, obtivemos resultados altamente satisfatórios e em conformidade com nossas expectativas. O principal objetivo do projeto era gerar, em cada geração, indivíduos capazes de alcançar pontuações cada vez mais elevadas, e com base nos resultados obtidos, ficamos satisfeitos em constatar que esse objetivo foi alcançado.

Para tornar a visualização dos resultados mais fácil e compreensível, optamos por apresentar os resultados mais significativos por meio de gráficos.

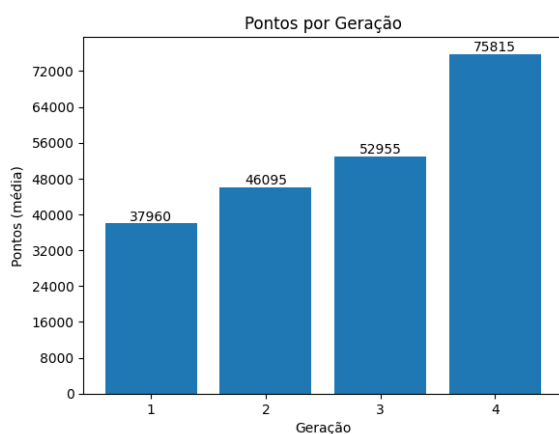


Figura 1

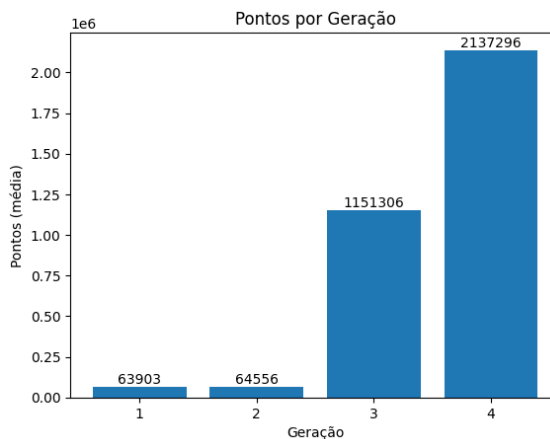


Figura 2

Conforme ilustrado na Figura 2, em uma das baterias de testes, identificamos um indivíduo que obteve uma média de 2 milhões de pontos. Essa conquista representa um desempenho excepcional, indicando que esse indivíduo possui uma combinação genética altamente favorável, capaz de gerar resultados de destaque.

Esses resultados reforçam a eficácia do nosso método de inteligência artificial e fornecem evidências sólidas de que estamos no caminho certo para alcançar nossos objetivos de maximizar as pontuações obtidas. Essa conquista nos motiva a continuar aprimorando nosso projeto e explorando novas possibilidades para impulsionar ainda mais o desempenho dos indivíduos gerados.

4. Implementação

O conteúdo utilizado no desenvolvimento deste projeto está disponível no repositório do Github: <https://github.com/rafazardo/TetrIA>.

O repositório contém a implementação do algoritmo genético aplicado ao jogo Tetris, juntamente com o visual do jogo. A interface visual foi fortemente baseada no tutorial "Creating Tetris in Python with pygame"[Nick 2021]. Você pode consultar esse tutorial para entender melhor o processo de criação da interface.

Além disso, o repositório contém informações detalhadas sobre como utilizar a implementação corretamente.

Certifique-se de visitar o repositório para obter acesso ao código-fonte, recursos e instruções de uso adequado.

5. Conclusão

Durante o desenvolvimento, diversas ideias de abordagem surgiram, nós tentamos implementar a inteligência artificial utilizando Q-Learning, consideramos utilizar redes neurais ou algoritmos de busca. Por fim, escolhemos o caminho dos algoritmos genéticos pois nos interessamos pela forma como funcionavam e parecia ser uma solução elegante para o problema que nos dispomos a resolver.

A tentativa de utilização de Q-Learning foi extremamente desafiadora, na medida em que o jogo de Tetris possui uma quantidade de estados imensa quando colocada em relação aos recursos computacionais existentes. Nós procuramos formas cada vez mais engenhosas de diminuir o tamanho da tabela Q gerada, mas não foi possível fazer com que as particularidades de cada estado real fossem traduzidas de forma efetiva para uma tabela de tamanho razoável sem perder qualidade na jogatina.

À parte dos detalhes de implementação do Algoritmo Genético previamente apresentados neste documento, podemos mencionar o que consideramos ser alguns de seus pontos fortes e fracos. Acreditamos que o processo de mutação que ocorre nos indivíduos seja interessante e muito importante para que escapemos de ótimos locais, ele cumpre o papel da exploração do espaço de busca, porém, como trata-se de um algoritmo estocástico, ficamos à mercê do acaso para que esse benefício seja alcançado. Observamos a execução de seleção nos indivíduos como um ponto positivo, pois essa parte do algoritmo cumpre o papel de exploitation.

Foi um trabalho interessante e nos ajudou a reforçar e nos aprofundar no estudo de temas de inteligência artificial.

Referências

Nick (2021). Creating tetris in python with pygame - beginner tutorial (oop), programming with nick.