
TRABALHO 1

Marcelo Gonçalves Pedro Junior Helena Beatriz Gomes Silva
cc20730@g.unicamp.br cc22135@g.unicamp.br
Elisa Chen Huang
cc23126@g.unicamp.br

1 Resumo

Este relatório tem por objetivo apresentar e explicar o funcionamento do código feito na implementação de uma Inteligência Artificial para o jogo de lógica "Nim". O algoritmo de busca utilizado para o desenvolvimento da IA é o Alpha-Beta Pruning, altamente prático para aplicações em tempo real, como IA de jogos, onde velocidade e eficiência são indispensáveis. O Alpha-Beta Pruning utiliza uma estrutura conhecida como árvore de busca, que pode ser usada para representar e organizar todas as possíveis decisões que podem ser feitas em um jogo ou problema. O relatório discute o funcionamento do código, a lógica por trás do algoritmo e os testes realizados para avaliar o desempenho da IA no jogo.

2 Introdução

Um dos algoritmos mais utilizados em Inteligência Artificial é o Alpha-Beta Pruning, que se baseia em uma técnica de otimização do algoritmo de busca Minimax. O algoritmo Minimax é um processo comumente usado em jogos de dois jogadores, em que um jogador busca maximizar sua pontuação enquanto o outro busca minimizá-la. Entretanto, conforme o jogo flui e a complexidade dos movimentos aumenta, o número de possibilidades de jogadas cresce exponencialmente, levando a elevados custos computacionais.

A solução para isso é justamente o algoritmo Alpha-Beta Pruning, já que ele reduz o número de possibilidades que o algoritmo Minimax precisa avaliar para "podar"(eliminar) os ramos que não influenciam na decisão final. Ou seja, a ideia-chave por trás do Alpha-Beta Pruning é evitar analisar ramos da árvore de possibilidades que não interferem no final do jogo com base nos valores já descobertos durante a busca. Como resultado, ele simplifica o processo de tomada de decisão, permitindo avaliações mais rápidas e eficientes para a jogada.

A partir disso, foi utilizado o algoritmo de busca competitiva Alpha-Beta Pruning para o desenvolvimento de uma Inteligência Artificial que determina e realiza as melhores jogadas no jogo "Nim".

3 Objetivos

O principal objetivo deste relatório é discorrer sobre o desenvolvimento do projeto e explicar a lógica por trás dos códigos feitos para a implementação do algoritmo.

4 Metodologia

Para o desenvolvimento da Inteligência Artificial, foi utilizado o algoritmo de busca competitiva Alpha-Beta Pruning. A IA foi implantada no jogo "Nim", um jogo onde dois jogadores alternam turnos para remover objetos de pilhas e o jogador que tirar o último objeto perde. Cada pilha possui uma determinada quantidade de objetos, no caso deste trabalho, são 4 pilhas e essas pilhas possuem, respectivamente: 1, 3, 5 e 7 objetos. A implementação do Alpha-Beta Pruning foi feita na linguagem Python, com as seguintes funções: `choose_action`(escolher uma ação), `is_terminal`(se terminou o jogo), `evaluate`(avaliar o estado do jogo), `result`(calcular se realmente é o melhor resultado do jogo) e `play`(fazer uma ação).

A função de escolher ação realiza o método de busca recursiva que explora todas as possíveis jogadas que um jogador pode fazer. Se a profundidade máxima da busca atingir zero ou se o estado for terminal (ou seja, o jogo acabou), a função retorna o valor da avaliação do estado. Caso contrário, a função percorre as ações possíveis e, recursivamente, avalia o valor de cada uma, eliminando possibilidades que não afetam o resultado final.

A função `"is_terminal"` verifica se todas as pilhas do jogo estão vazias. Se sim, significa que o jogo terminou.

Agora a função de avaliação foi projetada para avaliar o valor de um estado, retornando um valor positivo se o estado for vantajoso para a IA e negativo se for vantajoso para o oponente.

Já a função `"result"` remove uma quantidade de objetos de uma pilha do estado atual e retorna o novo estado do jogo após a remoção dos objetos. A ação é composta por uma tupla (`pile`, `count`), onde `"pile"` é o indicador da pilha e `"count"` é o número de objetos a serem removidos dessa pilha.

Por fim, a função principal "play" gerencia o fluxo do jogo entre a IA e o jogador humano, alternando entre os turnos dos jogadores. O jogo termina quando um dos jogadores vence, ou seja, quando todas as pilhas estão vazias.

5 Conclusão

A Inteligência Artificial desenvolvida para o jogo demonstrou ser eficiente e eficaz. A IA foi testada em diferentes cenários, com variações no número de pilhas e objetos, para avaliar sua capacidade de tomada de decisão e eficiência computacional. Os resultados obtidos demonstraram que, em partidas com um número moderado de pilhas e objetos, a IA foi capaz de tomar decisões rápidas e estratégicas, apresentando uma alta taxa de sucesso.

Em síntese, conclui-se que o uso do algoritmo de busca competitiva Alpha-Beta Pruning é extremamente efetivo para a elaboração de uma IA competente, capaz de realizar jogadas rápidas, tomar decisões inteligentes e estratégicas adequadas a situação.

6 References

- MY GREAT LEARNING. Alpha Beta Pruning in AI. Disponível em: <https://www.mygreatlearning.com/blog/alpha-beta-pruning-in-ai/>. Acesso em: 22 de março 2025.
- ROBERTS, Eric. Alpha-Beta Pruning. Stanford University. Disponível em: <https://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/intelligent-search/alphabeta.html>. Acesso em: 29 de março 2025.