



Universidade Federal de Ouro Preto
DECSI — Departamento de Computação e Sistemas



Ibsiany Dias Godinho - 20.2.8097
Lincoln dos Santos Rebouças - 20.1.8068
Rafael Caetano Teixeira - 20.1.8105

TRABALHO I - INTELIGÊNCIA ARTIFICIAL

Trabalho I

João Monlevade, 2023

Ibsiany Dias Godinho - 20.2.8097
Lincoln dos Santos Rebouças - 20.1.8068
Rafael Caetano Teixeira - 20.1.8105

TRABALHO I - INTELIGÊNCIA ARTIFICIAL

Relatório teórico do trabalho de Inteligência Artificial para o curso de Engenharia de Computação e Sistemas de Informação, como requisito parcial para nota na disciplina.

Universidade Federal de Ouro Preto

João Monlevade

2023

Sumário

1	INTRODUÇÃO	3
2	ALGORITMO DE PODA ALFA-BETA	4
3	ALGORITMO DE AVALIAÇÃO (AVAL)	7
4	JUNÇÃO DO ALGORITMO DE PODA ALFA-BETA E AVALIA- ÇÃO DE DESEMPENHO	9
5	CONCLUSÃO	11
	Bibliografia	12

1 Introdução

O jogo Connect4 é um jogo popular de estratégia em que dois jogadores competem para formar uma sequência de quatro fichas em uma grade vertical, horizontal ou diagonal. O algoritmo Minimax é comumente utilizado para desenvolver uma inteligência artificial capaz de jogar o Connect4 de forma eficiente e desafiadora.

O algoritmo Minimax, pode ser computacionalmente caro quando aplicado a jogos de tabuleiro complexos, como o Connect4. Para lidar com esse problema, iremos utilizar uma técnica chamada poda alfa-beta para melhorar o desempenho do algoritmo Minimax, que será implementado em combinação o algoritmo poda alfa-beta e Minimax.

Além disso, a avaliação de desempenho será uma parte fundamental do trabalho. Implementaremos uma função de avaliação que possa atribuir um valor numérico a uma posição no tabuleiro, refletindo a vantagem de um jogador em relação ao outro. Essa função de avaliação será essencial para que o algoritmo Minimax tome decisões informadas durante a busca.

Ao combinar o algoritmo Minimax com a poda alfa-beta e a avaliação de desempenho, esperamos criar uma inteligência artificial para o Connect4 que possa tomar decisões estratégicas de forma rápida e eficiente, tornando o jogo mais desafiador e envolvente para os jogadores humanos.

2 Algoritmo de poda Alfa-Beta

O algoritmo de poda alfa-beta é uma técnica utilizada para melhorar a eficiência da busca em árvores de jogos. Ele foi projetado para reduzir o número de nós que precisam ser avaliados durante a busca, sem comprometer a qualidade da solução encontrada. A figura 1 mostra um pseudo código para o código de poda Alfa-Beta que foi readaptado para nossa necessidade no algoritmo MiniMax do jogo Connect4.

```
1  função BUSCA-ALFA-BETA (estado) retorna uma ação
2  entradas: estado, estado corrente no jogo
3  v ← VALOR-MAX(estado, -∞, +∞)
4  retornar a ação em SUCESSORES(estado) com valor v
5  função VALOR-MAX (estado, α, β) retorna um valor de utilidade
6  entradas: estado, estado corrente no jogo
7  α, valor da melhor alternativa para MAX ao longo do caminho até estado
8  β, valor da melhor alternativa para MIN ao longo do caminho até estado
9  se TESTE-TERMINAL(estado) então retornar UTILIDADE(estado)
10 v ← -∞
11 para cada s em SUCESSORES(estado) faça
12 v ← MAX(v, VALOR-MIN(s, α, β))
13 se v ≥ β então retornar v
14 α ← MAX(α, v)
15 retornar v
16 função VALOR-MIN (estado, α, β) retorna um valor de utilidade
17 entradas: estado, estado corrente no jogo
18 α, valor da melhor alternativa para MAX ao longo do caminho até estado
19 β, valor da melhor alternativa para MIN ao longo do caminho até estado
20 se TESTE-TERMINAL(estado) então retornar UTILIDADE(estado)
21 v ← +∞
22 para cada s em SUCESSORES(estado) faça
23 v ← MIN(v, VALOR-MAX(s, α, β))
24 se v ≤ α então retornar v
25 β ← MIN(β, v)
26 retornar v
```

Figura 1 – Pseudo código do algoritmo de poda Alfa-Beta. Fonte: (ENGEL, s.d., p. 20).

Foi realizada uma rodada teste de um dos nossos integrantes do grupo contra a Inteligência Artificial utilizando esse código a profundidade escolhida foi 4 e deixado a inteligência artificial ganhar nos 4 primeiros movimentos(Movimentos realizados pelo jogador 1 em ambas as técnicas: 1-2-5-6) para ser possível realizar um gráfico de comparação, a Figura 2 mostra o resultado sem a implementação do algoritmo poda alfa-beta.

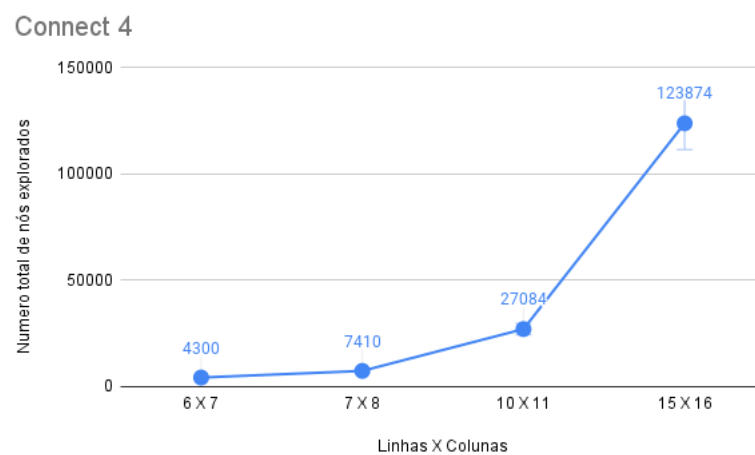


Figura 2 – Gráfico de comparação com tabuleiro de diversos tamanhos

A figura 3 mostra o resultado com a implementação do algoritmo poda alfa-beta.

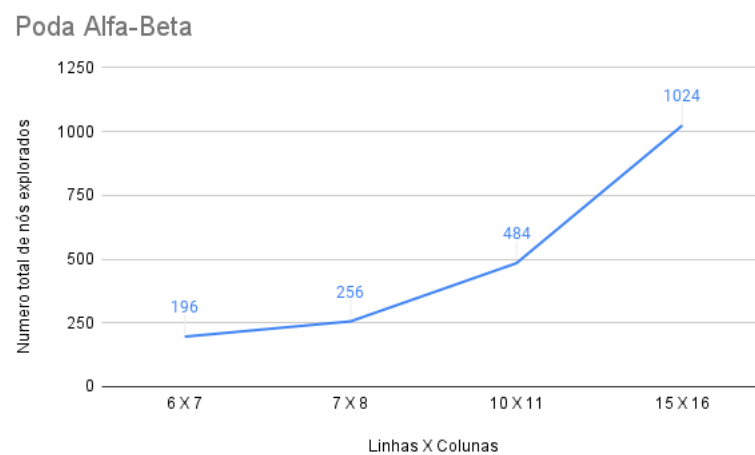


Figura 3 – Gráfico de comparação com tabuleiro de diversos tamanhos

Já a figura 4 mostra a comparação entre o minimax sem poda alfa-beta e o minimax com poda alfa-beta.

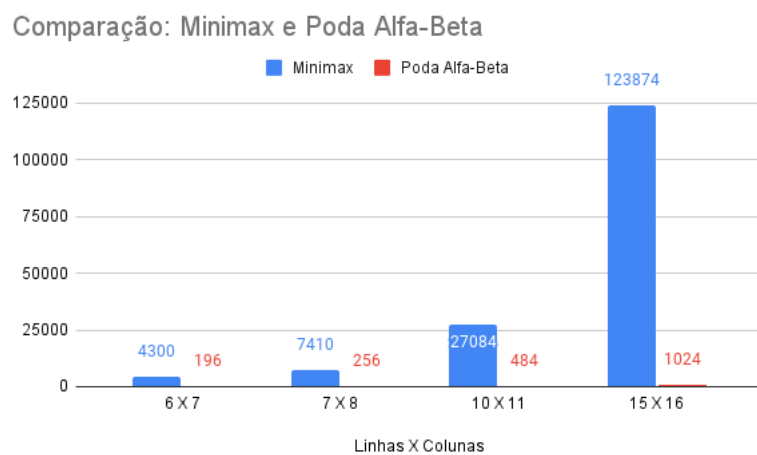


Figura 4 – Gráfico de comparação entre os algoritmos Minimax e Poda Alfa-Beta

É possível notar que o número de nós explorados durante a execução do algoritmo poda alfa-beta foi significativamente menor que o minimax sem a implementação da técnica informada, comprovando haver melhoria no desempenho.

3 Algoritmo de avaliação (AVAL)

O algoritmo Minimax com avaliação de desempenho é uma técnica utilizada em jogos de estratégia para encontrar a melhor jogada possível, considerando as ações dos jogadores adversários. Ele é baseado no conceito de que cada jogador maximizará sua própria pontuação e minimizar a pontuação oponente. A figura 3 mostra um pseudo código para o código de avaliação de desempenho readaptado para nossa necessidade no algoritmo MiniMax do jogo Connect4.

```
1  Algoritmo Minimax (com avaliação)
2  função DECISÃO-MINIMAX(estado, profundidade) retorna uma ação
3  entradas: estado, estado corrente no jogo; profundidade corrente
4  v ← VALOR-MAX(estado)
5  retornar a ação em SUCESSORES(estado) com valor v
6  função VALOR-MAX(estado) retorna um valor de avaliação
7  se TESTE-PARADA(estado, profundidade) então retornar AVAL(estado)
8  v ← ∞
9  para cada s em SUCESSORES(estado) faça
10 v ← MAX(v, VALOR-MIN(s))
11 retornar v
12 função VALOR-MIN(estado) retorna um valor de avaliação
13 se TESTE-PARADA(estado, profundidade) então retornar AVAL(estado)
14 v ← ∞
15 para cada s em SUCESSORES(estado) faça
16 v ← MIN(v, VALOR-MAX(s))
17 retornar v
18 TESTE-TERMINAL(estado) ← TESTE-PARADA(estado, profundidade)
19 UTILIDADE(estado) ← AVAL[estado]
```

Figura 5 – Pseudo código do algoritmo de Avaliação. Fonte: (ENGEL, s.d., p. 9).

Foi realizada uma rodada teste de um dos nossos integrantes do grupo contra a Inteligência Artificial utilizando o algoritmo de avaliação. A profundidade escolhida foi 4 e foi realizada uma série de movimentos pelo jogador 1 até que a IA ou o Jogador 1 vencesse (Movimentos realizados pelo jogador 1 em ambas as técnicas: 1-2-5-6). O algoritmo de avaliação teve mais nós explorados por encontrar a melhor solução para a jogada, como pode-se observar pela Figura 6 a quantidade de nós explorados é explicitamente maior do que o gráfico da Figura 2 a respeito do jogo apenas com o minimax.

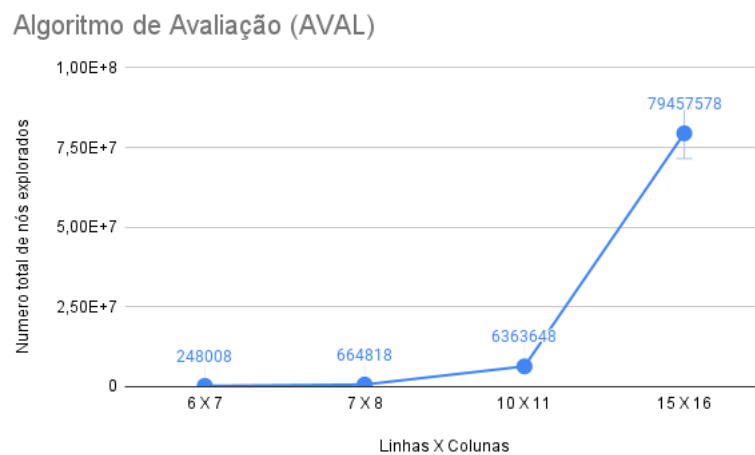


Figura 6 – Algoritmo Minimax com Avaliação

Já na Figura 7 é possível observar melhor a diferença entre os nós explorados dos dois algoritmos.

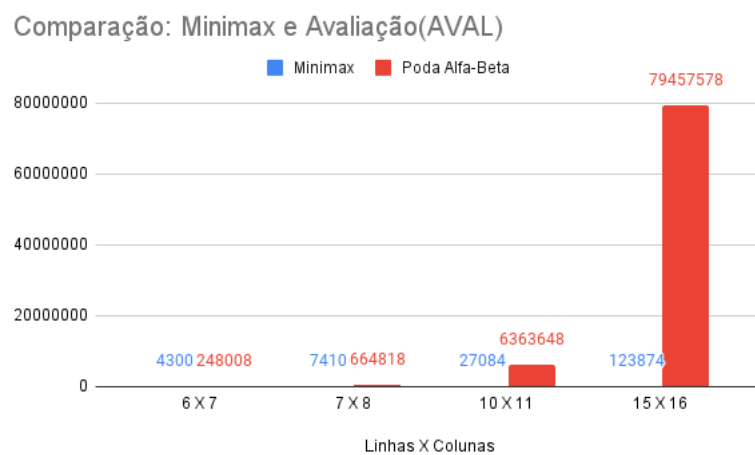


Figura 7 – Comparação do Algoritmo Minimax e Minimax com Avaliação

4 Junção do algoritmo de poda Alfa-Beta e avaliação de desempenho

O algoritmo Minimax com poda alfa-beta e função de avaliação (AVAL) é uma abordagem eficiente para encontrar a melhor jogada em um jogo de adversários. Ele utiliza uma árvore de busca para examinar todas as jogadas possíveis, atribui valores de avaliação às posições do jogo e utiliza a poda alfa-beta para reduzir o número de nós a serem explorados. Ao considerar as avaliações e aplicar a poda, o algoritmo consegue determinar a jogada ótima para o jogador maximizador.

Foi realizada uma rodada teste de um dos nossos integrantes do grupo contra a Inteligência Artificial utilizando o algoritmo de poda alfa-beta com a função de avaliação. A profundidade escolhida foi 4 e os movimentos realizados foram os mesmos realizados nos algoritmos anteriores. A figura 8 mostra os nós explorados do algoritmo de junção em 4 tipos de tabuleiro conforme ilustra a imagem.

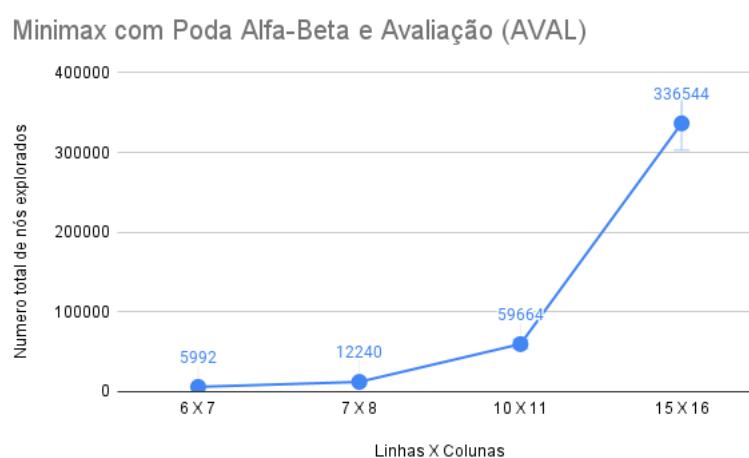


Figura 8 – Minimax com Poda Alfa-Beta e Avaliação(AVAL)

Na figura 9 é possível perceber a diferença entre os 4 algoritmos. Comparando minimax sem nenhuma alteração com o algoritmo de poda alfa-beta é perceptível que a exploração de nós do poda alfa-beta é extremamente menor, devido a poda que é realizada no algoritmo evitando de explorar nós inúteis. O algoritmo minimax sem alteração com o algoritmo de avaliação é possível perceber a quantidade elevada de nós explorados, tendo em conta que o algoritmo de avaliação irá explorar todas as possibilidades procurando a melhor solução portanto tendo um custo e um tempo maior. Já o algoritmo de junção onde contém o poda alfa-beta e o algoritmo de avaliação não tem uma exploração de nós muito elevada quanto o algoritmo de avaliação, mas também não tem uma exploração mínima

como o algoritmo de poda, o algoritmo de junção tem uma solução ótima comparado ao algoritmo minimax sem alteração nenhuma e se comparado aos outros algoritmos também, levando em conta o desempenho e a eficiência do algoritmo. Pode-se concluir que o algoritmo de junção traz uma solução ótima e uma exploração de nós aceitável para a complexidade do algoritmo.

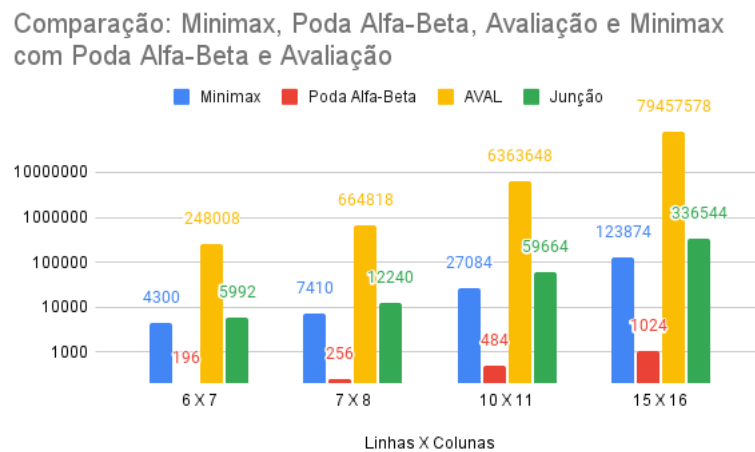


Figura 9 – Comparação entre os algoritmos Minimax, Poda Alfa-Beta, Avaliação e Minimax com Poda Alfa-Beta e Avaliação

5 Conclusão

É possível concluir que os algoritmos de Poda Alfa-Beta e Avaliação(AVAL) conseguiu melhorar significativamente o jogo Connect 4 e sua eficiência. O algoritmo de poda Alfa-Beta implementado em conjunto com o Minimax melhorou a eficiência do código equiparando ao número de estados explorados, já o algoritmo minimax com avaliação melhorou a eficiência no qual a Inteligência Artificial jogasse, ou seja, fez com que ela ficasse mais inteligente analisando e trazendo melhores soluções para ganhar o jogo com soluções ótimas. Em suma, o algoritmo Minimax com poda alfa-beta e AVAL demonstra uma eficiência notável no jogo Connect 4. Através da estratégia de Minimax, o algoritmo consegue considerar todas as possíveis jogadas e encontrar a melhor sequência de movimentos para maximizar as chances de vitória, enquanto a poda alfa-beta permite reduzir drasticamente o espaço de busca, eliminando ramos desnecessários e acelerando o processo de tomada de decisão. A combinação dessas técnicas resulta em um algoritmo eficiente para o Connect 4. A capacidade de examinar várias jogadas à frente, sem considerar todas as possibilidades, torna o algoritmo rápido o suficiente para jogar em tempo real.

Bibliografia

AGUIRRE, L. A. Introdução à Identificação de Sistemas, Técnicas Lineares e Não lineares Aplicadas a Sistemas Reais. Belo Horizonte, Brasil, EDUFMG. 2004.

ENGEL, Paulo. Inteligência Artificial. Algoritmo Minimax, p. 8, 9 e 20, 2023. Disponível em: <https://www.inf.ufrgs.br/engel/data/media/file/inf01048/jogos.pdf>.