

RELATÓRIO

Inteligência Artificial



Connect Four

André Cirne - 201505860

José Sousa - 201503443

Índice

Introdução	3
Algoritmo MiniMax	3
Algoritmo AlfaBeta	3
Connect Four	3
MiniMax e AlfaBeta aplicados ao Connect Four	4
Conclusão	5
Referências	6

Introdução

Os jogos entre dois jogadores, podem ser interpretados como uma estratégia de diferentes jogadas, com o objetivo de chegar a uma posição ganhadora (1). Estes tipos de interações são estudados pela teoria de jogos.

Do ponto de vista da teoria de jogos e inteligência artificial o interesse é gigante, já que num ambiente jogo para cada estado, existe um grande número de jogadas que pode ser tomado, sendo desafiante conseguir escolher a melhor jogada, o mais rapidamente possível. Como resposta a este desafio existem vários algoritmos. Neste trabalho abordamos dois deles, o MiniMax e o AlfaBeta. Os dois permitem-nos ganhar sempre ou na pior das situações levam nos ao empate.

Algoritmo MiniMax

O algoritmo MiniMax consegue encontrar a melhor jogada em cada instância do jogo, graças a uma função de avaliação e escolhendo a cada nível o máximo ou o mínimo da avaliação dos seus nós filhos.

A utilização de restrições do tipo mínimo ou máximo em cada nível da árvore deve-se à existência de dois jogadores. Nos níveis do jogador que está a utilizar o MiniMax, vamos escolher o máximo, porque somos nós a efetuar a decisão onde queremos jogar. Assim maximizamos as nossas possibilidades de ganhar. Nas posições do adversário escolhemos o mínimo, o que significa que será a pior jogada que ele pode fazer para nós.

Os valores que falamos anteriormente devem-se à função de utilidade, que é aplicada a cada nó terminal, e que é passado para os nós pais, usando as funções de mínimo e máximo conjugados com a recursão. (2)

Algoritmo AlfaBeta

O funcionamento do algoritmo AlfaBeta é semelhante ao do MiniMax. No entanto, este vai proceder à poda de alguns ramos da árvores, que são desnecessários aumentando assim a rapidez de resposta. Para implementação desta poda não vamos ter só o valor corrente do nó, mas também duas variáveis, um alfa e um beta, que respetivamente, guardam o máximo e o mínimo do nível. (3)

No nível máximo, escolhe-se o maior valor entre os sucessores do nó, e atualiza-se o valor de α e se o valor de $\alpha \geq \beta$, efetua-se o corte. No caso de um nível mínimo, escolhe-se o menor valor entre os sucessores de nó, e atualiza-se o valor de β e se o valor de $\beta \leq \alpha$, efetua-se o corte. (4)

Connect Four

O jogo dos 4 em linha, é um jogo no qual existem dois adversários com o objetivo de conseguir 4 posições em linha, nas seguintes orientações, vertical, horizontal e nas diagonais. O campo de jogo é constituído por um tabuleiro 6 por 7, onde as posições só podem ser preenchidas de baixo para cima.

Existe em cada instância do jogo, um número limitado de jogadas. Umas que proporcionam a evolução para uma posição mais ganhadoras e outras menos. Para conseguirmos fazer essa avaliação, utilizamos os algoritmos MiniMax e AlfaBeta com uma função de avaliação. A função utilizada faz a seguinte avaliação, dá dezasseis pontos se a jogada é feita por parte da

máquina e menos dezasseis no caso de ser humano. De seguida vai avaliar cada conjunto de quatro elementos, onde todas as possibilidades são consideradas.

A avaliação no caso da jogada pertencer à máquina é avaliada da seguinte forma zero se não existirem posições da máquina ou posições misturadas, um se só existir uma posição pertencente à máquina e nenhuma adversária, dez se existirem duas posições da máquina e nenhuma adversária, cinquenta na existência de três posições da máquina e nenhuma adversária e por fim se for uma sequência ganhadora quinhentos e doze. Para o caso contrário, a jogada do humano, os valores são os mesmos, mas multiplicados por menos um. (5)

MiniMax e AlfaBeta aplicados ao Connect Four

Para a resolução deste problema utilizamos os algoritmos MiniMax e AlfaBeta, utilizando a função de avaliação anteriormente referida. A razão pela qual implementamos os dois algoritmos foi de conseguir comparar a performance entre AlfaBeta e MiniMax.

Em termos de linguagem de implementação, utilizamos java devido à orientação ao objeto. Para cada algoritmo implementamos uma classe, os quais tinham 3 funções diferentes, uma para os níveis mínimos, outra máximos e uma para pesquisa geral.

Não utilizamos grandes profundidades principalmente no MinMax já que torna o tempo de resposta da máquina muito grande, tornando o jogo in jogável. Além disso para aumentar a rapidez do código, quando corremos pela segunda vez utilizamos princípios de programação dinâmica, já que com toda certeza existe parte da árvore que pode ser sempre reaproveitada.

Tabela 1- Registo de tempos e nos gerados de cada algoritmo em relação à profundidade.

Profundidade	Tempo(s)	Nos gerados	Tempo(s)	Nos gerados
	MinMax		AlfaBeta	
4	0,004	399	0,003	243
5	0,016	2800	0,011	1384
6	0,145	19607	0,028	5519
7	0,364	137256	0,095	37066
8	1,699	944167	0,189	82249
9	-	-	1,516	564910
10	-	-	4,546	1541902

Nas observações apresentadas, é necessário realçar que a partir da profundidade máxima nove, o MiniMax não conseguiu obter qualquer tipo de resposta, já que deu origem a um erro do tipo *heap overflow*.

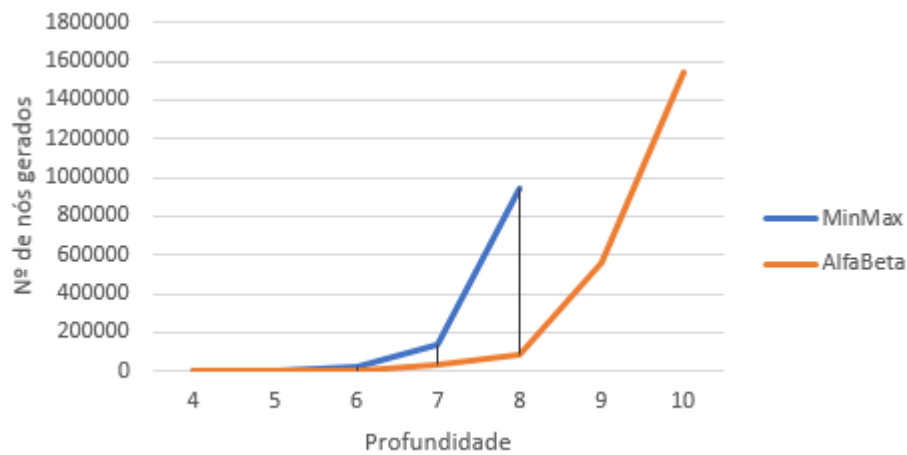


Gráfico 1 - Relação entre nós gerados e profundidade

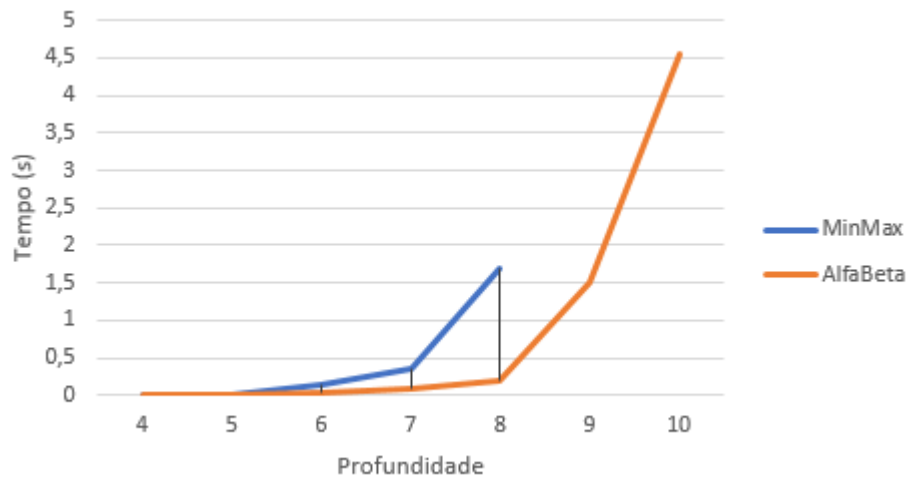


Gráfico 2-Relação entre tempos até à resposta e nível de profundidade

Conclusão

Como esperado, o algoritmo AlfaBeta conseguiu ter um melhor comportamento em termos de nós gerados e menor tempo de execução sem comprometer o resultado do jogo.

Inicialmente tivemos alguns problemas com os algoritmos em geral já que em determinadas situações, o algoritmo perante uma situação onde poderia ganhar com uma única só jogada, preferia defender a jogada do adversário quando este não estava numa posição de conseguir alcançar os quatro em linha. Este problema foi resolvido com um melhor tratamento das situações de empate da função de avaliação. No caso de duas avaliações iguais é escolhida a de menor profundidade, para garantir o caminho ótimo.

Referências

1. Wikipedia. [Online].; 2017. Available from: https://pt.wikipedia.org/wiki/Teoria_dos_jogos.
2. Aradhya. GeeksforGeeks. [Online].; 2017. Available from: <http://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/>.
3. Wkipedia. Alpha–beta pruning. [Online].; 2017 [cited 2017 Março 22. Available from: https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning.
4. Fred A. [Document].; 2002 [cited 2017 Março 22. Available from: <http://comp.ist.utl.pt/ec-aed/PDFs/AB.pdf>.
5. Universidade de Lisboa. [Document]. [cited 2017 Março 22. Available from: <https://fenix.tecnico.ulisboa.pt/downloadFile/3779573246185/cap6-jogos.pdf>.