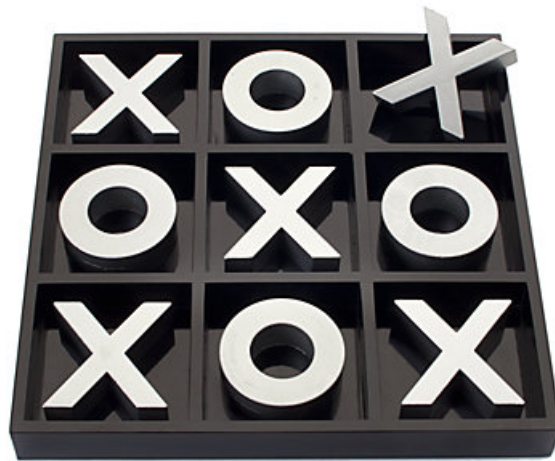


Jogo do Galo

Inteligência Artificial - Trabalho 2

Raul Ferreira {up200905641} e José Castro {up201007482},
DCC - FCUP

Março e Abril 2016



1 Introdução

Um conjunto de jogos com oponentes são os jogos de Soma-Zero (Zero-Sum Games) nos quais se inclui o Jogo do Galo.

Este trabalho consiste na implementação dos algoritmos Min-Max e Alfa-Beta aplicados ao Jogo do Galo, com o objectivo de comparar a eficiência em tempo e espaço de cada um deles e garantir que a sua implementação leva a que o computador nunca perca o jogo.

2 Algoritmos de decisão

2.1 Algoritmo Min-Max

O algoritmo Min-Max é um algoritmo recursivo utilizado em jogos com oponentes para escolher a próxima jogada. Consiste em tentar minimizar o valor do adversário e maximizar o valor do computador, assim o computador deve fazer o movimento que o deixa com maiores possibilidades de ganhar dentro do leque de movimentos que dão as menores possibilidades de vencer ao seu adversário.

Dada uma árvore de jogo, a estratégia óptima pode ser determinada a partir do valor minimax de cada nó. A função minimax retorna um valor heurístico para os nós folha e os restantes nós herdam o valor dos seus descendentes.

O algoritmo tem complexidade temporal $O(b^m)$ e espacial $O(b * m)$, sendo b o numero de ramificações (se estas forem constantes) ou a média do numero de ramificações (se estas forem variáveis), e a profundidade maxima da arvore de pesquisa.

2.2 Algoritmo Alfa-Beta

Alfa-beta é um algoritmo de busca que procura diminuir o número de nós que são avaliados pelo algoritmo minimax na sua árvore de pesquisa.

O benefício de alfa-beta reside no facto de ramos da árvore de pesquisa poderem ser eliminados. Desta forma, a pesquisa pode ser limitada à sub-árvore 'mais promissora'. A optimização reduz a profundidade efectiva para pouco mais de metade do algoritmo minimax.

O Alfa é o valor da melhor escolha encontrado até então para qualquer ponto de escolha do Max. Se um valor associado a um nó for menor do que alfa, então o Max não percorrerá o caminho em causa, ou seja, poda-se (faz-se *pruning*) a esse ramo da Árvore. O Beta é definida de maneira análoga para os pontos de escolha do Min.

O algoritmo tem complexidade temporal de aproximadamente $O(b^{m/2})$ e espacial $O(b*m)$, sendo b o numero de ramificações (se estas forem constantes) ou a média do numero de ramificações (se estas forem variáveis), e a profundidade maxima da arvore de pesquisa.

3 Jogo do Galo

O jogo consiste num tabuleiro com três linhas e três colunas, onde dois jogadores, de forma alternada, marcam um espaço que esteja vazio. Tradicionalmente, cada jogador escolhe um símbolo ("O" ou "X") para utilizar em todas as suas jogadas, sendo o objectivo final do jogo conseguir preencher, com o seu símbolo, três quadrados em linha(seja na horizontal, na vertical ou na diagonal). Existe uma variante em que o jogador deve também impedir que o adversário ganhe na jogada seguinte.

4 Min-max e Alfa-beta aplicados ao Jogo do Galo

Visto que o jogo do galo é um jogo com oponentes, os algoritmos Min-max e Alfa-Beta aplicam-se perfeitamente a ele. O algoritmo Alfa-Beta no entanto é consideravelmente mais rápido pois passa muito mais rapidamente para nós irmãos em vez de percorrer nós filhos que já sabe que não vale a pena serem considerados. É possível ainda melhorar o tempo da primeira jogada do computador fazendo-o jogar sempre no meio (tanto num algoritmo como no outro), uma vez que nesse caso ainda não há qualquer incerteza relacionada com a possível jogada do adversário. (O algoritmo retornaria sempre esse jogada). Esta optimização torna primeira jogada muito mais rápida (em ambos os algoritmos).

Para este trabalho não usamos nenhuma estrutura de dados em particular para além de uma classe para definir o nó, Arrays para nós sucessores, e matrizes para definir o tabuleiro de Jogo.

Com base nos resultados obtidos conseguimos concluir o que prevíamos, que o algoritmo alpha beta é mais rápido, pois não expande todos os nós (Consultar os gráficos em anexo).

A nossa função utilidade simplesmente calcula o número de possibilidades de o computador vencer nas linhas, colunas e diagonais, subtraindo depois pelo número de possibilidades de o jogador vencer nas linhas, colunas e diagonais. Não temos limite de profundidade, indo sempre até uma configuração de vitória, empate ou derrota, mas construímos o código de maneira a ser facilmente implementado o limite. (todos os gráficos encontram-se em anexo)

5 Comentários Finais e Conclusões

Tal como era esperado, pudemos concluir que o desempenho do algoritmo alfa-beta é vastamente superior ao min-max durante as primeiras jogadas do Jogo do Galo, devido à quantidade de nós a menos que tem de explorar (consultar gráficos em anexo). Essa vantagem tornar-se-ia ainda mais visível num jogo com oponentes onde a árvore de procura fosse mais profunda do que os 9 níveis(0 a 8) do Jogo do Galo.

Bibliografia

- [1] Inês Dutra. Jogos com oponentes. <http://www.dcc.fc.up.pt/~ines/aulas/1516/IA/jogos.pdf>, 2016.
- [2] Wikipedia. Alpha-beta pruning. https://en.wikipedia.org/wiki/Alphabeta_pruning, 2016.
- [3] Wikipedia. Minimax. <https://en.wikipedia.org/wiki/Minimax>, 2016.

6 Anexos

6.1 Min-Max VS Alfa-Beta (Y=Segundos, X=Nível da Árvore)

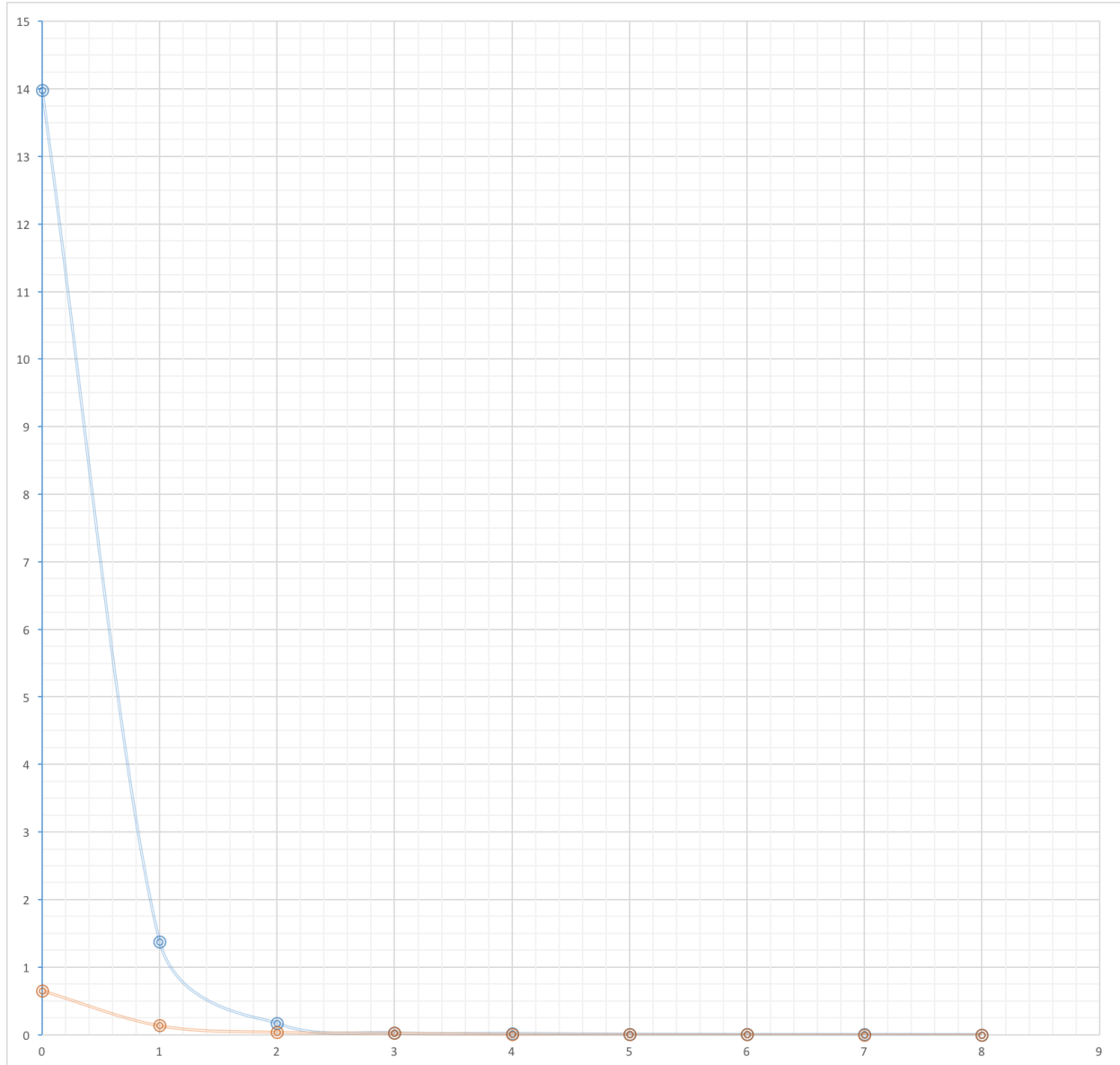


Figura 1: Min-Max (Azul), Alfa-Beta (Vermelho)

6.2 Min-Max VS Alfa-Beta (Y=Nós, X=Nível da Árvore)

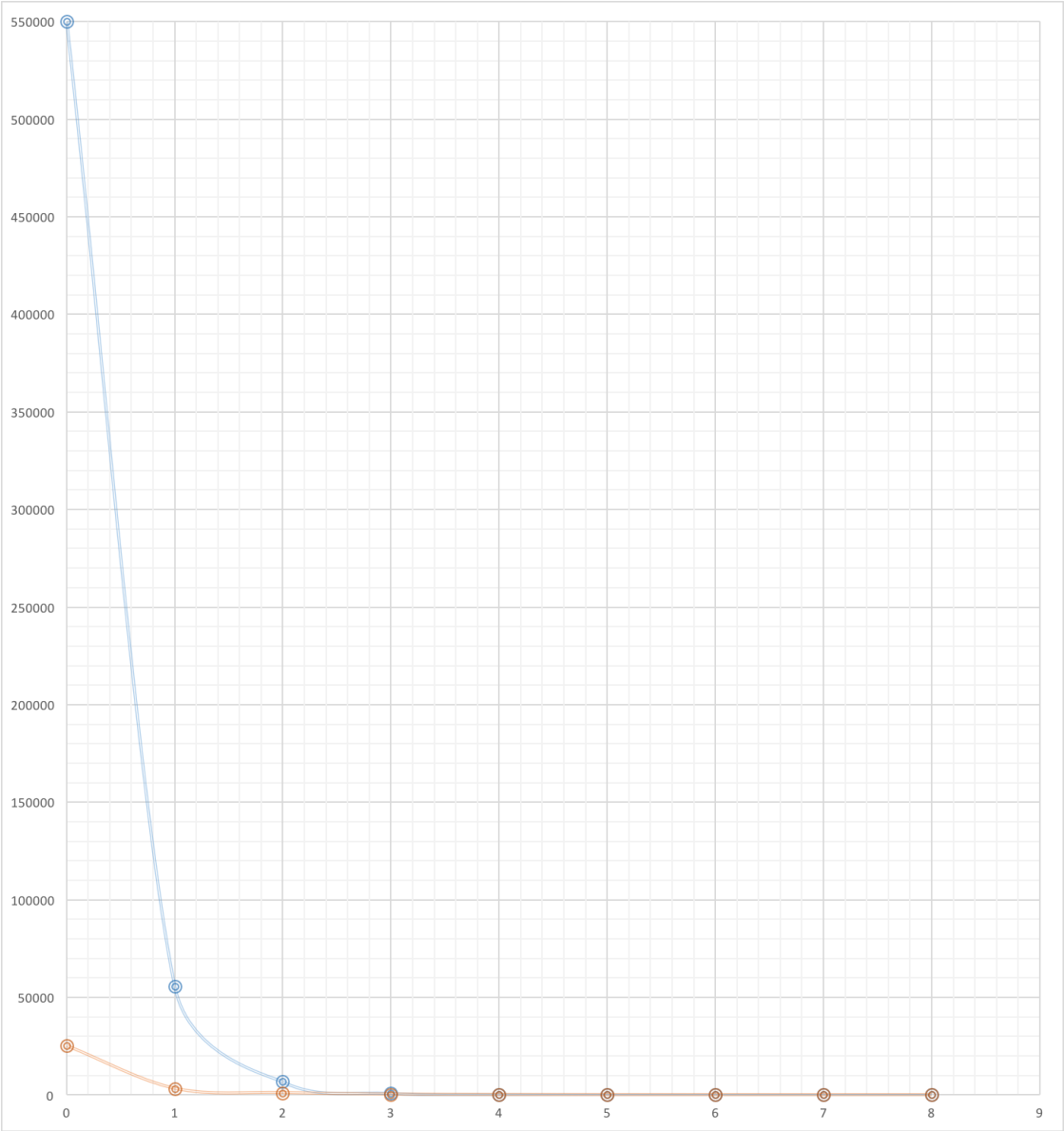


Figura 2: Min-Max (Azul), Alfa-Beta (Vermelho)

6.3 Min-Max VS Alfa-Beta - Optimizado (Y=Segundos, X=Nível da Árvore)

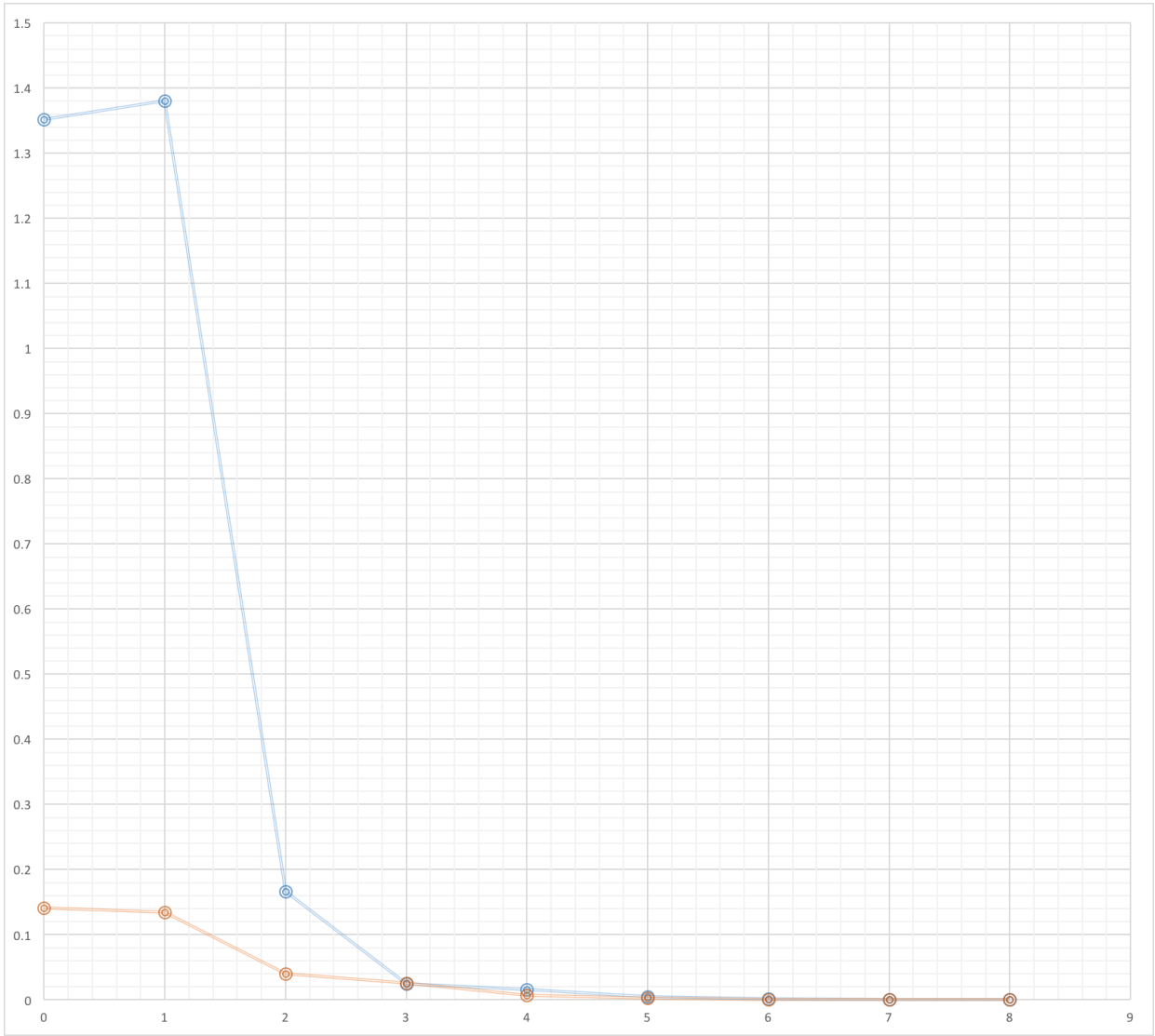


Figura 3: Min-Max (Azul), Alfa-Beta (Vermelho)

6.4 Min-Max VS Alfa-Beta - Optimizado (Y=Nós, X=Nível da Árvore)

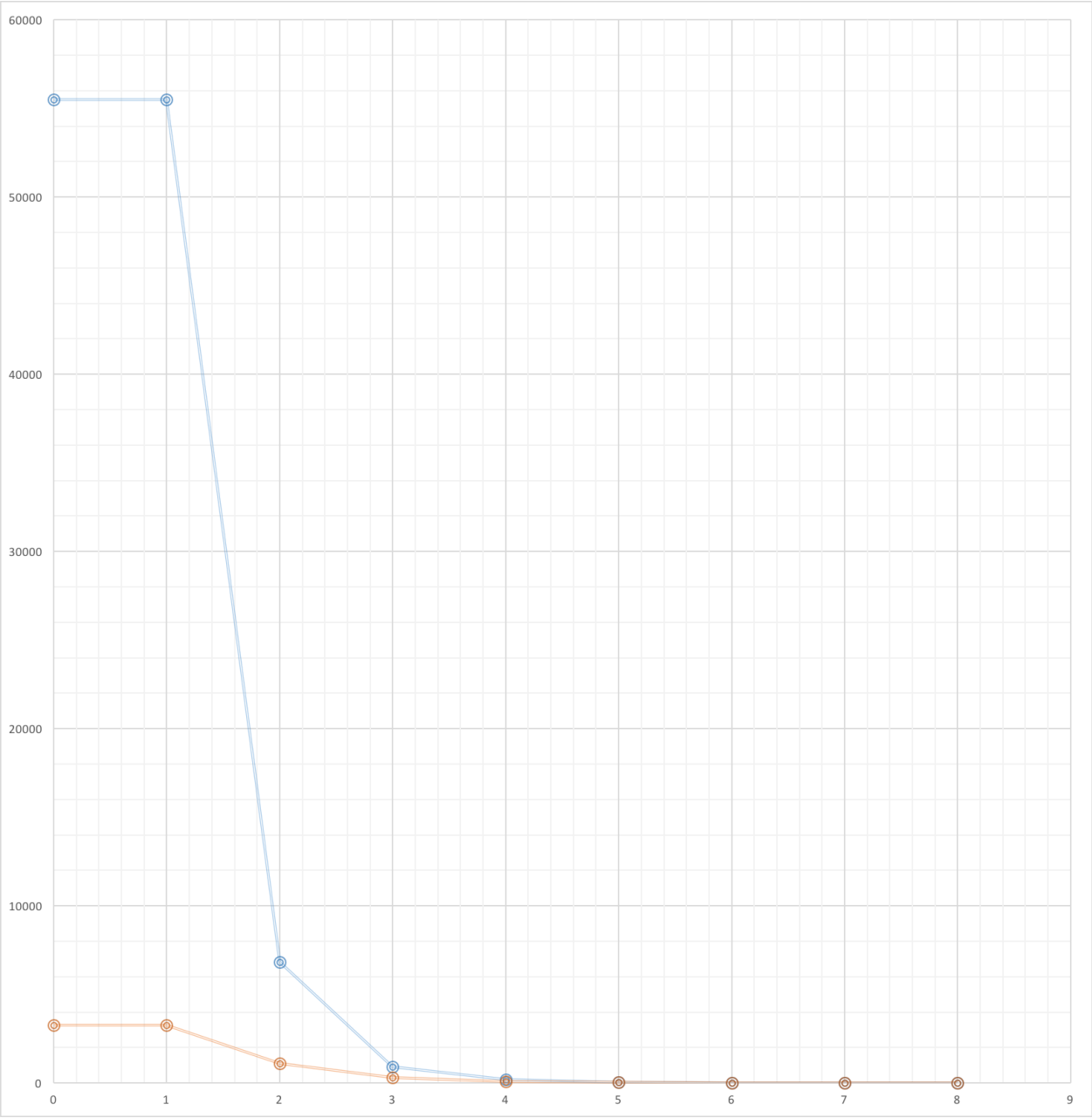


Figura 4: Min-Max (Azul), Alfa-Beta (Vermelho)