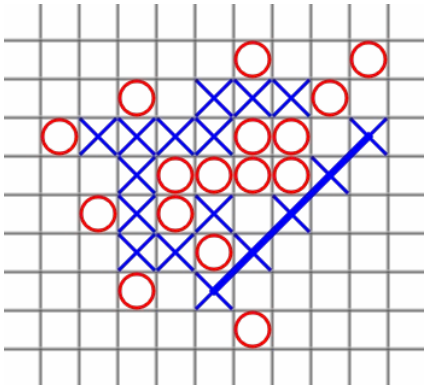
MNK game

Wadood Alam, Matthew Pacey, Joe Nguyen

Introduction the game

Generalization of tic tac toe game

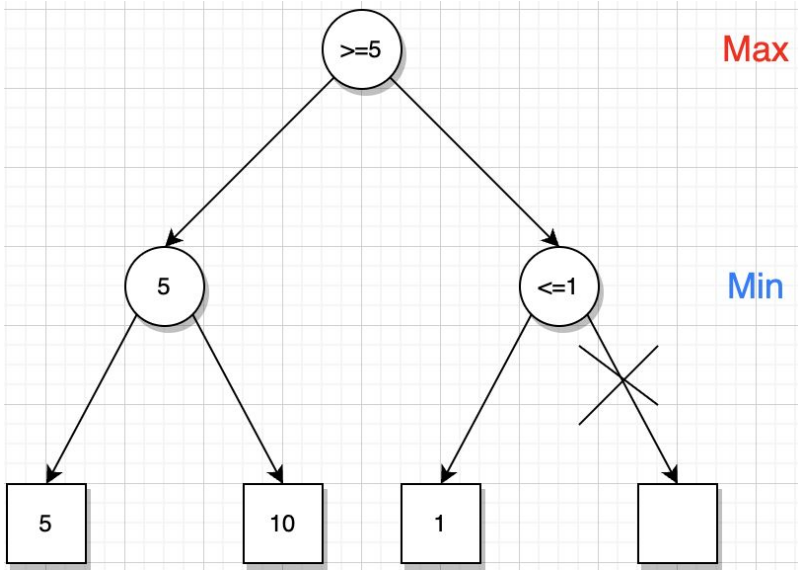
- **M x N** board
- the winner is the player who first gets **K** stones of their own color in a row, horizontally, vertically, or diagonally

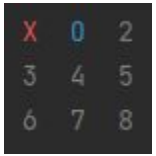


Example of a completed 11,10,5-game

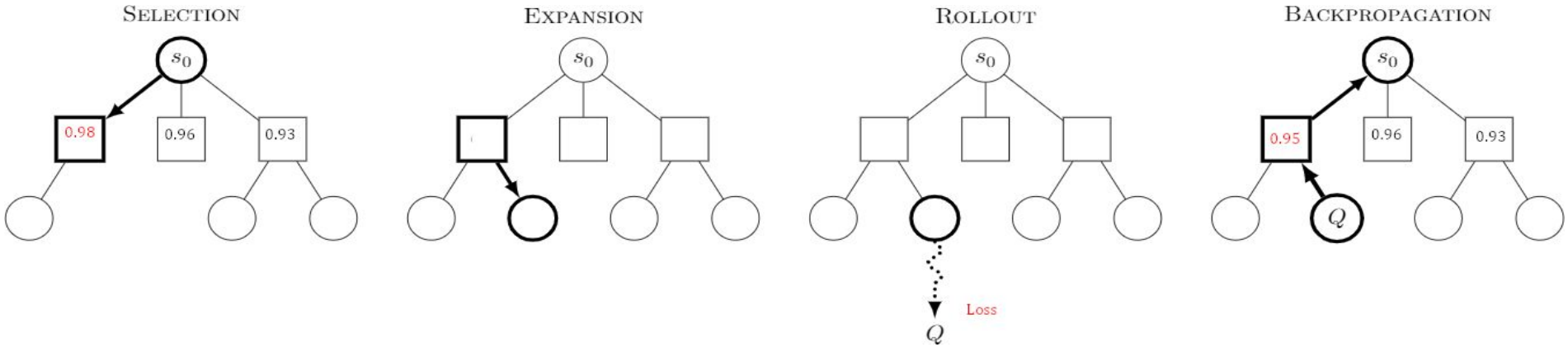
Search Algorithm: Alpha Beta Pruning

- Minimax: Min player tries to minimize the utility of max player i.e player 2 prevents player 1 from winning
- Alpha-Beta: Optimized version of minimax
  - Prune when current value <= alpha if MIN
  - Prune when current value >= beta if MAX
- Heuristic Ordering: Select the empty cell with the most occupied neighbors





Search algorithm: Monte Carlo Tree Search



Node Selection: value = $X_i + C * \sqrt{(p_i / n_i)}$ where:

Exploitation: X_i = Node wins / games

Exploration: C = constant (1.1); p_i = log(parent node's games); n_i = (node games);

Node Expansion Heuristic: Select the empty cell with the most occupied neighbors

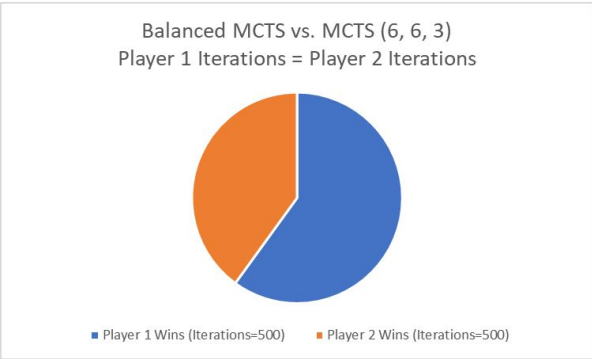
Experiments

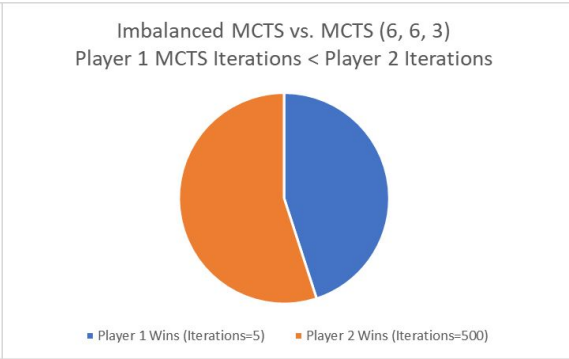
- 6 different boards: 3x3, 3x4, 4x4, 4x5, 5x5, 6x5
  - 3 square; 3 rectangular
- Minimax + Alpha-beta with heuristic ordering
  - Depth Limit = 5
- Varying *K* from 3 to *M* where *M* is the num of rows
- Number of iterations in MCTS = 500
- ***α***-β vs MCTS
- MCTS vs ***α***-β
- MCTS vs MCTS with varying MCTS Iterations per player

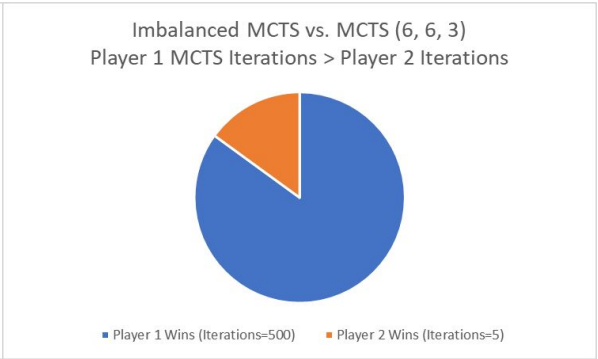
Results: MCTS Iteration Variance

- Player 1(first to move) has a slight advantage
- Increasing MCTS Iterations allows Player 2 to overcome advantage

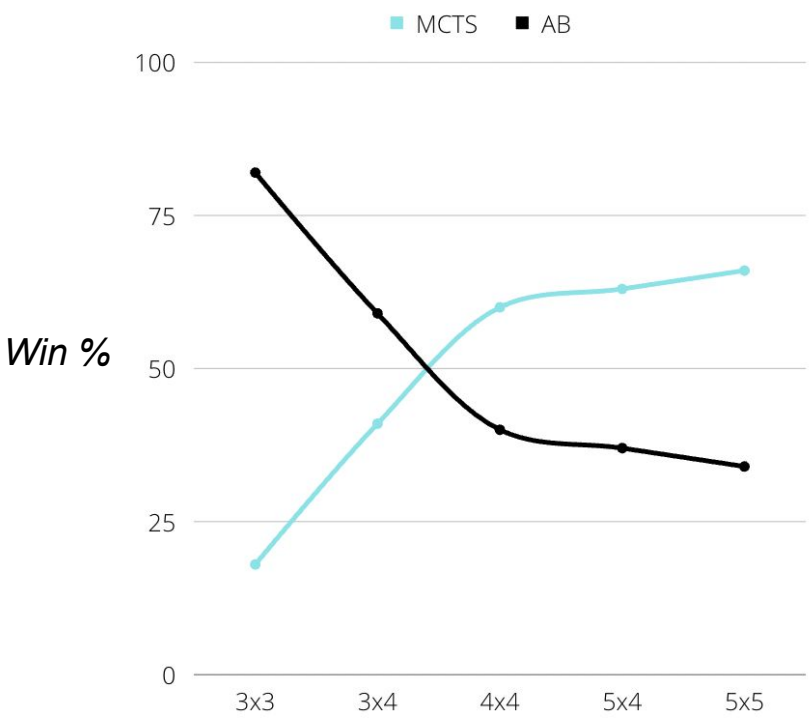
P1 Wins	P2 Wins	P1 Iterations	P2 Iterations
60	40	500	500
45	55	5	500
85	15	500	5



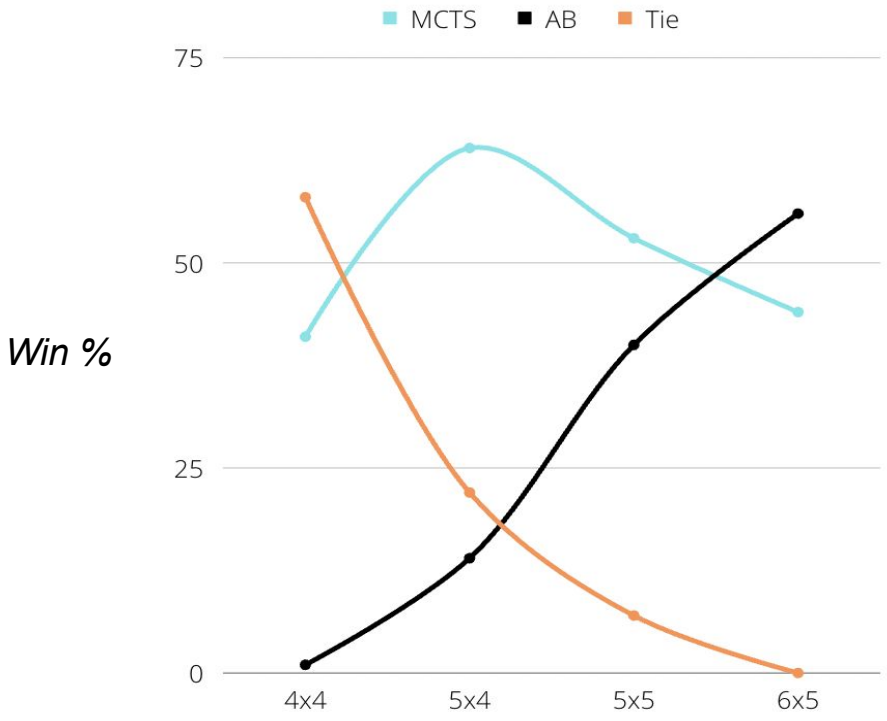




MCTS(1) VS AB(2)

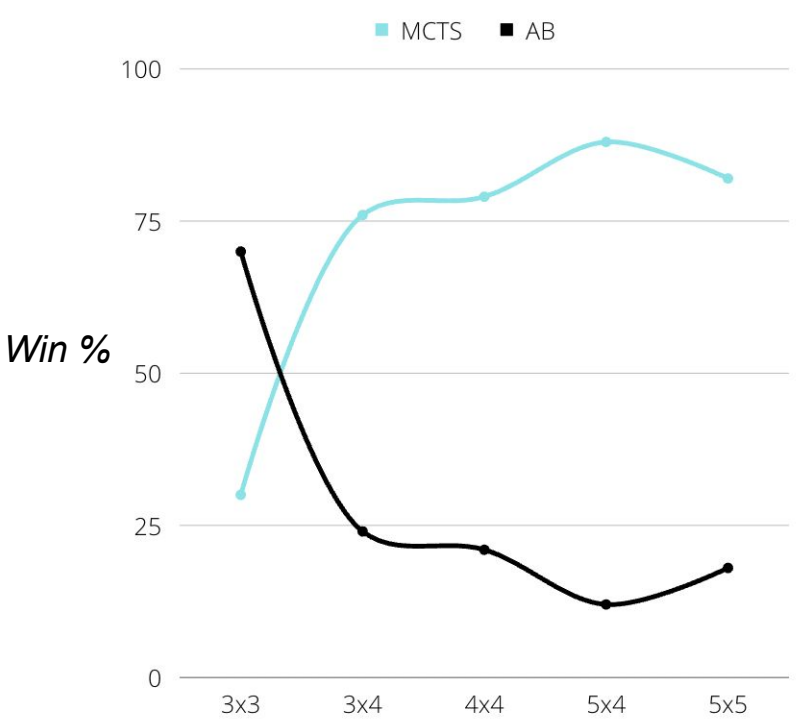


K = 3; MCTS VS AB

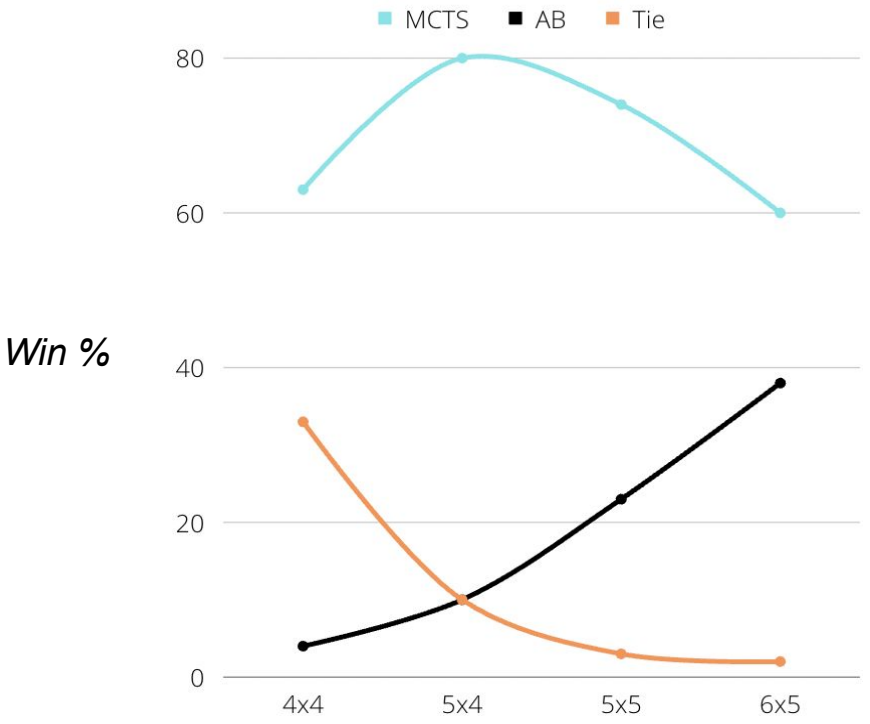


K = 4; MCTS VS AB

AB(1) VS MCTS(2)



K = 3; AB VS MCTS



K = 4; AB VS MCTS

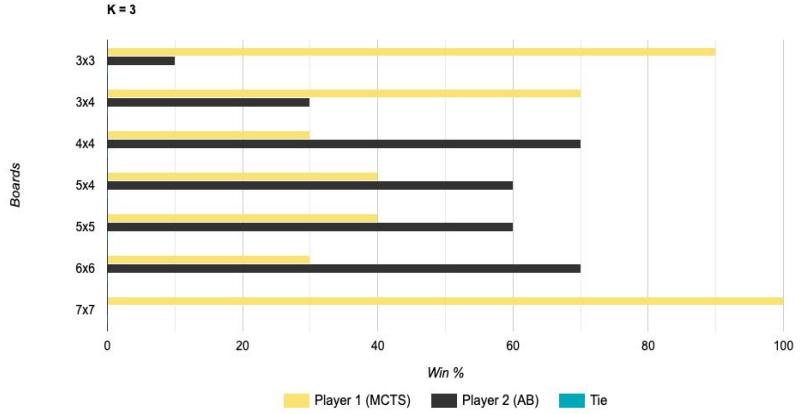
Discussion

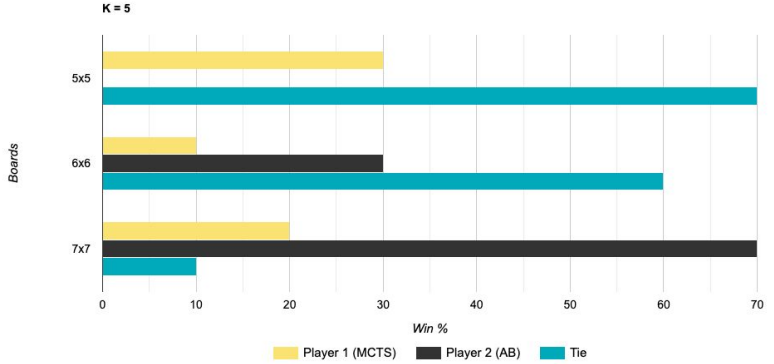
- Player 1 has an advantage
- High K values led to more ties
- Keeping a consistent depth limit of 5, our data makes sense because as the board size increases more nodes need to be expanded. By a lower depth limit it is expected to have more ties. The tradeoff is that it takes less compute time.
- Increasing the number of MCTS Iterations increases score. The optimal value seemed to be 500 iterations (number of playouts to simulate for each move decision), past that not many wins were gained at the expense of longer compute time.
- MCTS is better than Alpha-Beta when the problem is simple unless K = 4, where Alpha-Beta is unable to model a large tree search
- With heuristic and without heuristic there is a clear difference between the number of nodes expanded (hence more time taken). Thus we decided to keep a consistent, lower depth limit of 5

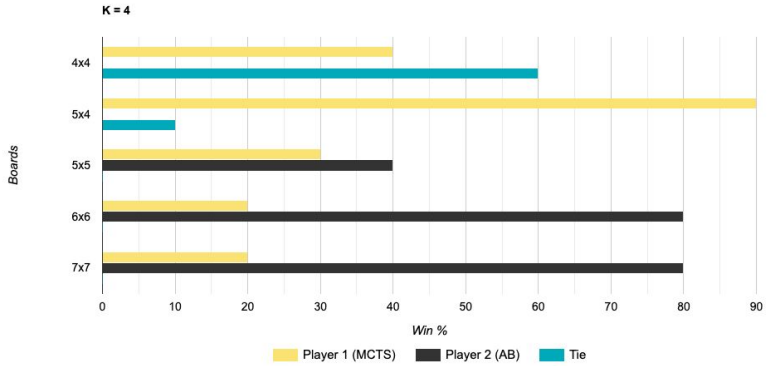
Thank you!

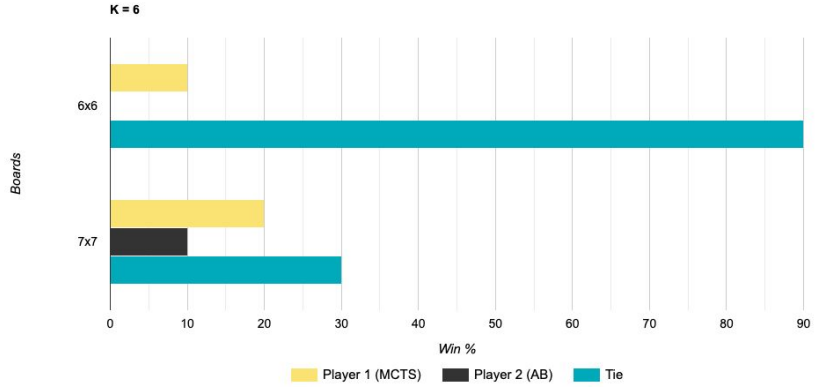
Questions & Concerns

MCTS VS AB

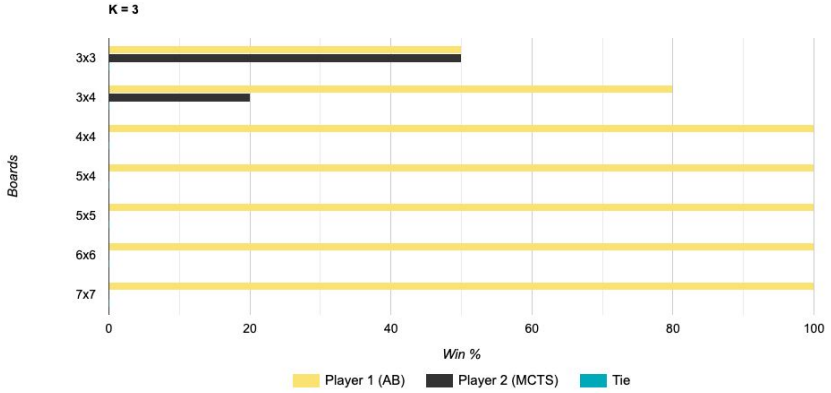

K = 3
Boards
3x3
3x4
4x4
5x4
5x5
6x6
7x7
0
20
40
60
80
100
Win %
Player 1 (MCTS)
Player 2 (AB)
Tie

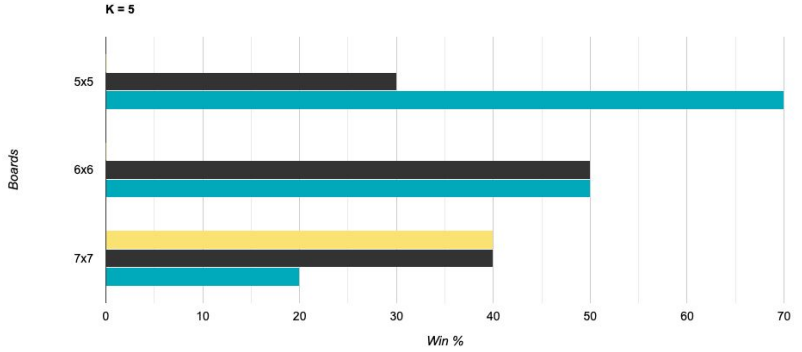

K = 5
Boards
5x5
6x6
7x7
0
10
20
30
40
50
60
70
Win %
Player 1 (MCTS)
Player 2 (AB)
Tie

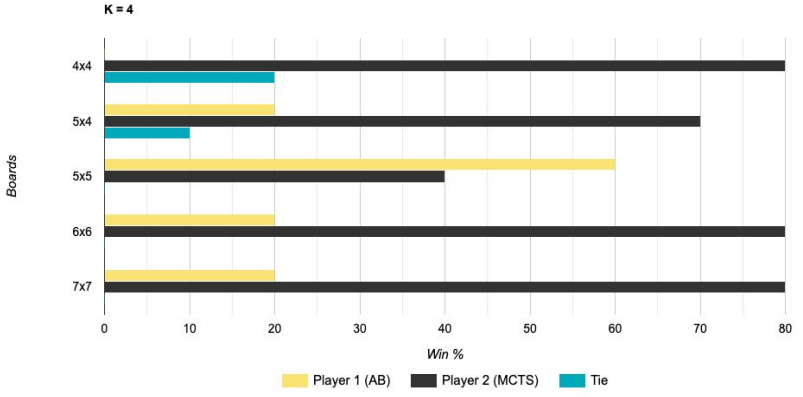

K = 4
Boards
4x4
5x4
5x5
6x6
7x7
0
10
20
30
40
50
60
70
80
90
Win %
Player 1 (MCTS)
Player 2 (AB)
Tie

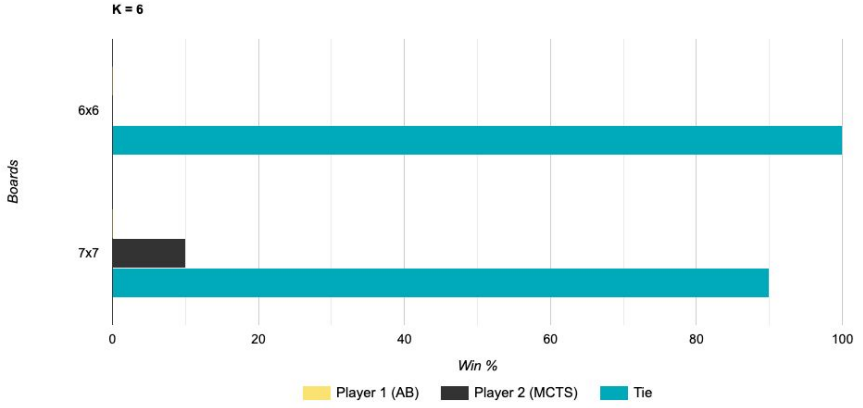

K = 6
Boards
6x6
7x7
0
10
20
30
40
50
60
70
80
90
Win %
Player 1 (MCTS)
Player 2 (AB)
Tie

AB VS MCTS


K = 3
Boards
3x3
3x4
4x4
5x4
5x5
6x6
7x7
0
20
40
60
80
100
Win %
Player 1 (AB)
Player 2 (MCTS)
Tie


K = 5
Boards
5x5
6x6
7x7
0
10
20
30
40
50
60
70
Win %


K = 4
Boards
4x4
5x4
5x5
6x6
7x7
0
10
20
30
40
50
60
70
80
Win %
Player 1 (AB)
Player 2 (MCTS)
Tie


K = 6
Boards
6x6
7x7
0
20
40
60
80
100
Win %
Player 1 (AB)
Player 2 (MCTS)
Tie

Minimax + Alpha-beta with / without heuristic ordering

Max_depth = inf

Size of the game	Number of nodes expanded		Percentages compared to without
	Without heuristic ordering	With heuristic ordering	
3 x 3 k = 2	165	26	15.75%
3 x 3, k = 3	18297	536	2.93%
4 x 4, k = 2	813	47	5.78%
4 x 4, k = 3	1024394	4242	0.41%!!
4 x 4, k = 4	Too long to run	Too long to run	