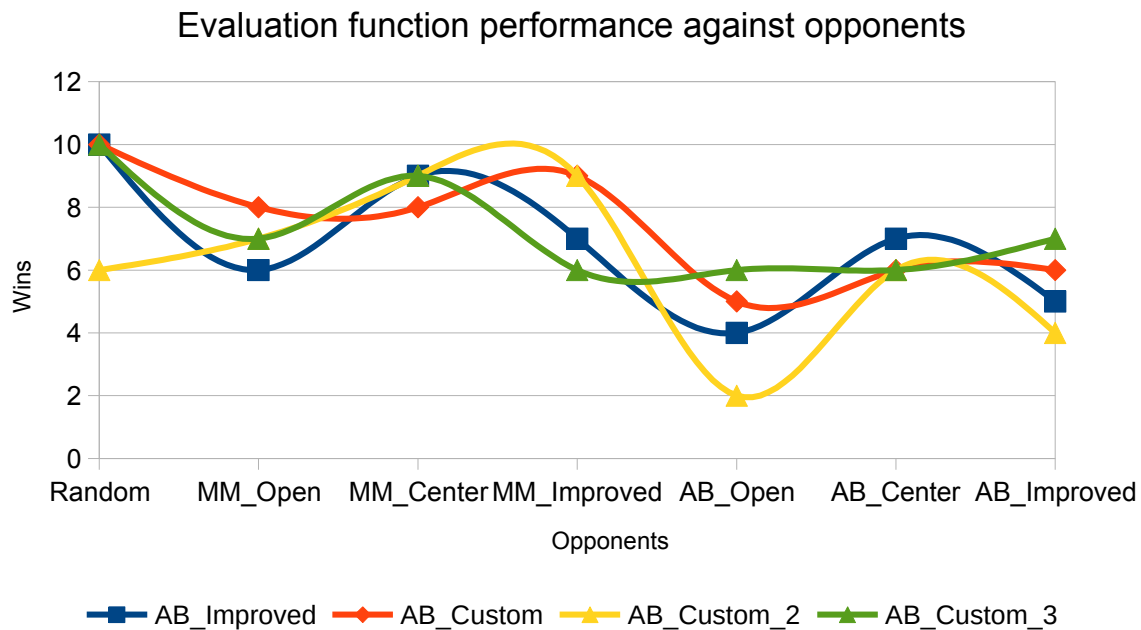
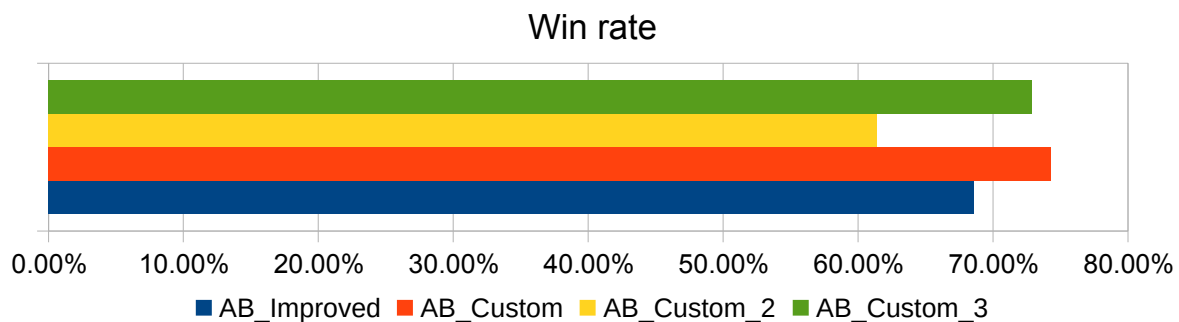


Heuristic analysis

Overview of heuristic analysis implemented as part of Game-Playing Agent project where agent is playing a game against another one on a chess like board where pieces (2 in total) move like a knight in chess.



There were 3 evaluation functions that were implemented, each of those was tested in a tournament and their winning rate was measured when playing against agents with predefined basic evaluation functions.



Strongest **AB_Custom** heuristic (win rate 74.30%) was implemented by counting difference between own and opponent moves and difference of center position score (in first part of the game), and added attack score which was more of experiment and needs tuning to achieve any desirable effect. Move count was used not only from existing level of the game but from one ply deep as well, this seemed to increase winning rate, mostly because choice was backed up with higher trust that move is going to succeed. Center position score was necessary to make sure agent tries to stay there to increase its move choices later in the game, this was measured and its impact was reduced as game moved forward. Last attack move was added to award agent when it captures positions

where opponent can move, but real value of its implementation should be measured separately and currently its influence is small.

AB_Custom heuristic function was storing position scores in a basic hash table although its impact on alpha beta pruning was not measured.

Second to strongest was quite simple **AB_Custom_3** heuristic, its win rate was close to the first one (72.9%) although evaluation function measured only distance between 2 pieces on the board! Its success is most likely caused by the fact that distance between players gives them more space to move.

Weakest evaluation function was **AB_Custom_2** (61.4%) which measured difference between players distance to center, but its weakest part was the fact that in the end of the game center position does not give much value as most of the board is used already.

I would recommend choosing **AB_Custom** heuristic as it performed better with win rate 74.30% (marginally better than **AB_Custom_3**), its computational intensity was reduced as hashtable was used to store computed values and most of the calculations were retrieved from it when using iterative deepening. In addition to that **AB_Custom** is weighing scores and gives more weight for active player versus opponent, for example when available moves are counted, their value is reduced to 50% for active player and to 40% for inactive, this indicates to agent that it is more important to look into active player position rather opponent.