Part I

Alpha-Beta pruning improved the time complexity

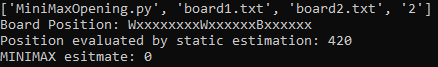
Test1: WxxWBxxxxxWBxxxBxBWBxxW

Board1:xxxxxxxxxWxxxxxxBxxxxxx

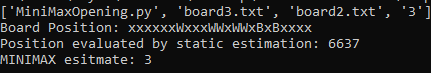
Board3:xxxxxxxxxxWWxWWxBBBxxxx

**Opening vs ABOpening**

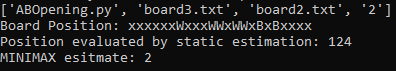
MiniMaxOpening.py case1



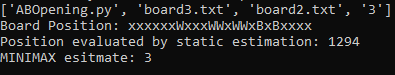
MiniMaxOpening.py case2



ABOpening.py case1

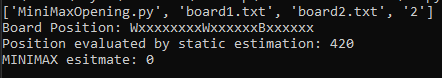


ABOpening.py case2

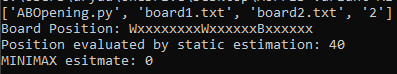


After applying alpha-beta cut, the executed time reduce from 420 to 40 in first case and 6637 to 1294 in second case

MiniMaxOpening.py case1 case 3



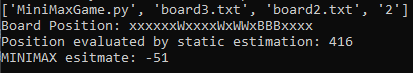
ABOpening.py case3



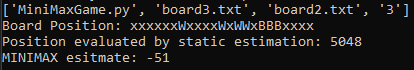
420->40

**MiniMaxGame vs ABGame**

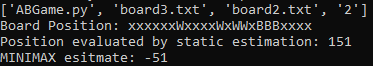
MiniMaxGame.py case1



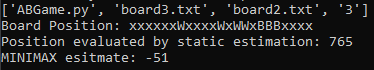
MiniMaxGame.py case2



ABGame.py case1

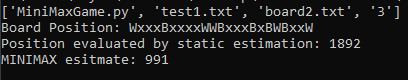


ABGame.py case2

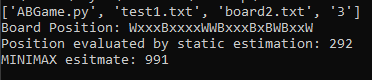


After applying alpha-beta cut, the executed time reduce from 416 to 151 in first case and 5048 to 765 in second case

MiniMaxGame.py case3



ABGame.py case3

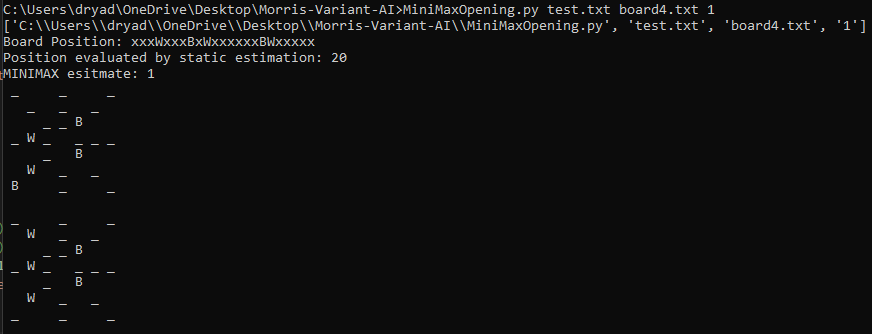


1892->292

Part II

New static evaluation improved the AI

MiniMaxOpening.py case 1



The Input

The result

In this program, the white takes the 0 location of black which is not the best choice due to the 1 depth parameter. The white can only predict 1 step further.

Here I introduce the new evaluation function will check the potential mill like ‘W X W’, ‘W W x’, ‘x B B’.

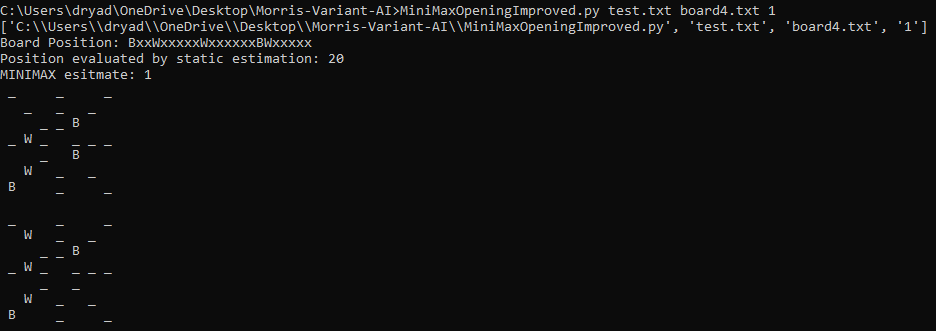
In those situations the function will increase or decrease the score.

‘W X W’ will add 0.4 score.

‘x B B’ wll minus 0.4 score.

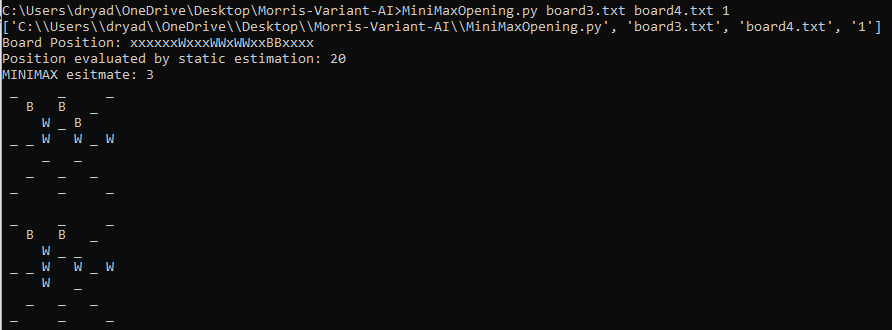
With this new function the program prone to choose the location which will build the mill or destroy the opponent mill.

MiniMaxOpeningImproved.py case 1

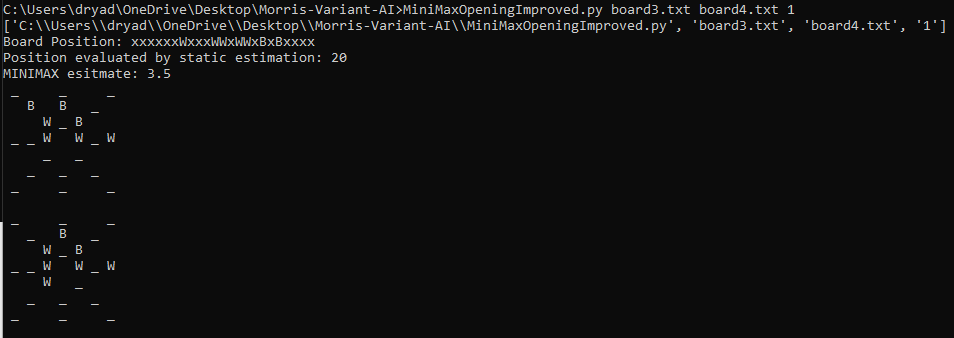


Instead of choosing the 0 location, the new program chooses to destroy the black potential mill.

MiniMaxOpening.py case 2



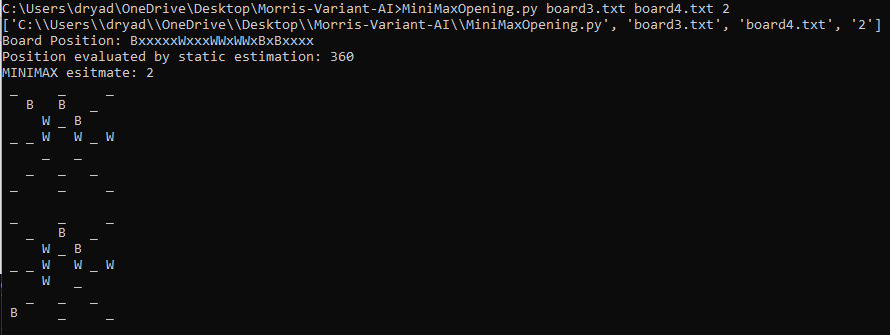
MiniMaxOpeningImproved.py case 2



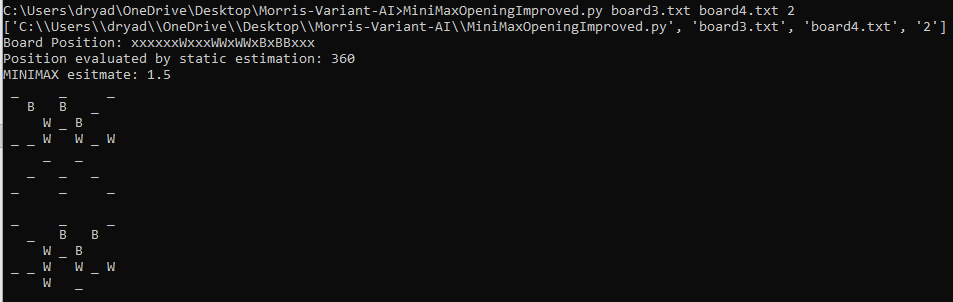
Again, old program only took the first black in the board.

New program took the one in potential mill.

MiniMaxOpening.py case 3



MiniMaxOpeningImproved.py case 3

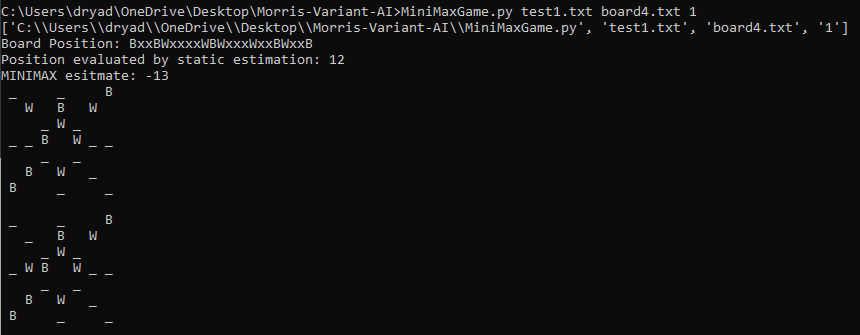


This time white destroy the black potential mill due to the 2 depth.

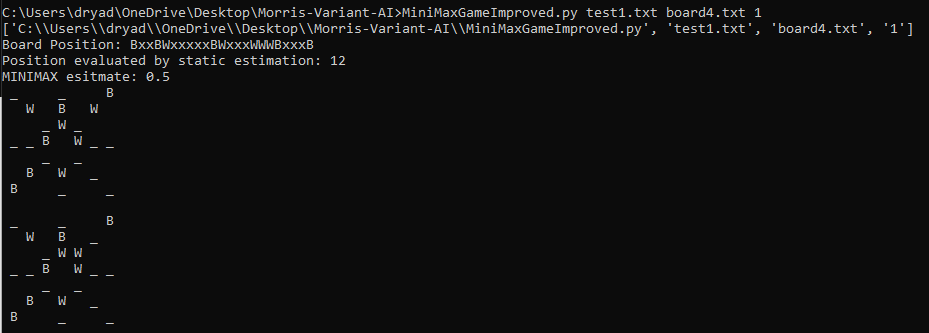
But black choose the 0 location.

In the new program, black creates two potential mills with one move.

MiniMaxGame.py case1

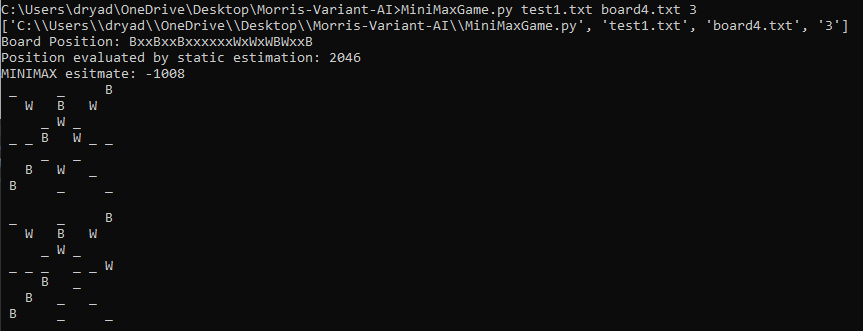


MiniMaxGameImproved.py case1

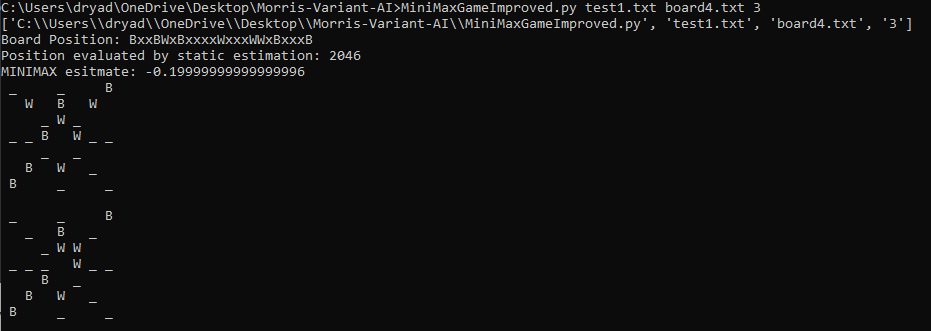


The new program can reach better position.

MiniMaxGame.py case2



MiniMaxGameImproved.py case2



The new program can reach better position again.