

1)

a.

“basicUtility” function is the lesser of the two utility functions I created and only uses win and loss conditions to calculate the utility by assigning 1 to a win and -1 to a loss and 0 otherwise. I chose to implement this function as it seemed to be the simplest utility function possible.

“advancedUtility” function is the better of the two utility functions I created. It adds 1000 to the utility for a win, subtracts 1000 from the utility for a loss, adds the number of potential win conditions for each square, subtracts the number of potential loss conditions for each square, and then multiplies the total by the number of remaining moves. I chose to implement this as it both accounts for victory/loss conditions as well as conditions in which no victory or loss has occurred

b.

3 x 3 Sample results:

3 x 3 mini max with a max ply of 9 and basic utility function

Turn 2: max depth = 8

Turn 6: max depth = 4

Turn 2: number of nodes explored = 27732

Turn 6: number of nodes explored = 21

Turn 2: wall time = 1.582 seconds

Turn 6: wall time = 0.002seconds

3 x 3 alpha beta with max ply of 9 and basic utility function

Turn 2: max depth = 4

Turn 6: max depth = 3

Turn 2: number of nodes explored = 292

Turn 6: number of nodes explored = 18

Turn 2: wall time = 0.061seconds

Turn 6: wall time = 0.001 seconds

3 x 3 mini max with a max ply of 9 and advanced utility function

Turn 2: max depth = 8

Turn 6: max depth = 4

Turn 2: number of nodes explored = 27732

Turn 6: number of nodes explored = 21

Turn 2: wall time = 2.467 seconds

Turn 6: wall time = 0.002 seconds

3 x 3 alpha beta with max ply of 9 and advanced utility function

Turn 2: max depth = 4

Turn 6: max depth = 3

Turn 2: number of nodes explored = 1787

Turn 6: number of nodes explored = 19
Turn 2: wall time = 0.214 seconds
Turn 6: wall time = 0.002seconds

3 x 3 Analysis:

Gameplay is identical across both searches and utility functions

Basic utility slightly outperformed advanced utility over all

With basic utility performing almost identically to advanced utility for mini-max search with the only difference being that basic utility had smaller wall times for both early and mid game, but scored identically in search depth and nodes explored for both early and mid game

For alpha beta search basic and advanced utility scored identically for search depth in both early and mid game, basic utility scored significantly better for nodes searched in early game and marginally better in late game, and basic utility scored better in wall time in early game and marginally better in late game

Alpha Beta search requires fewer ply than Mini-Max search over all

Alpha Beta search explores fewer nodes than Mini-Max search over all

Alpha Beta search is faster in the early game but is comparable in the mid game compared to Mini-Max search

4 x 4 Sample results:

4 x 4 mini max with a max ply of 5 and basic utility function

Turn 2: max depth = 5
Turn 8: max depth = 5
Turn 2: number of nodes explored = 360360
Turn 8: number of nodes explored = 9568
Turn 2: wall time = 14.965 seconds
Turn 8: wall time = 0.464 seconds

4 x 4 alpha beta with a max ply of 5 and basic utility function

Turn 2: max depth = 5
Turn 8: max depth = 4
Turn 2: number of nodes explored = 2312
Turn 8: number of nodes explored = 155
Turn 2: wall time = 0.12 seconds
Turn 8: wall time = seconds

4 x 4 alpha beta with a max ply of 7 and advanced utility function

Turn 2: max depth = 7
Turn 8: max depth = 4
Turn 2: number of nodes explored = 29896

Turn 8: number of nodes explored = 270
Turn 2: wall time = 1.503 seconds
Turn 8: wall time = 0.024 seconds

4 x 4 mini max with a max ply of 5 and advanced utility function

Turn 2: max depth = 5
Turn 8: max depth = 5
Turn 2: number of nodes explored = 360360
Turn 8: number of nodes explored = 12904
Turn 2: wall time = 48.666 seconds
Turn 8: wall time = 1.609 seconds

4 x 4 alpha beta with a max ply of 5 and advanced utility function

Turn 2: max depth = 5
Turn 8: max depth = 5
Turn 2: number of nodes explored = 8338
Turn 8: number of nodes explored = 1018
Turn 2: wall time = 1.132 seconds
Turn 8: wall time = 0.136 seconds

4 x 4 alpha beta with a max ply of 7 and advanced utility function

Turn 2: max depth = 7
Turn 8: max depth = 7
Turn 2: number of nodes explored = 188342
Turn 8: number of nodes explored = 6795
Turn 2: wall time = 26.855 seconds
Turn 8: wall time = 1.057 seconds

4 x 4 Analysis:

Overall, the basic utility function either preforms comparably to out preforms the advanced utility function in terms of depth searched, number of nodes searched and wall time, however the basic utility function vastly under preforms in terms of actual gameplay as it is very easy to beat and sometimes fails to even prevent the player from winning

Gameplay is identical across both searches

Alpha Beta search will use equal ply to Mini-Max search, how ever Alpha Beta can be run with a larger maximum ply

Alpha Beta search explores fewer nodes than Mini-Max search at both start and midgame

Alpha Beta search is faster than Mini-Max search at both start and midgame

5 x 5 Sample results:

5 x 5 mini max with a max ply of 3 and advanced basic function

Turn 2: max depth = 3

Turn 14: max depth = 3

Turn 2: number of nodes explored = 12144

Turn 14: number of nodes explored = 1320

Turn 2: wall time = 0.795 seconds

Turn 14: wall time = 0.09 seconds

5 x 5 alpha beta with a max ply of 3 and advanced basic function

Turn 2: max depth = 3

Turn 14: max depth = 3

Turn 2: number of nodes explored = 550

Turn 14: number of nodes explored = 130

Turn 2: wall time = 0.037 seconds

Turn 14: wall time = 0.011 seconds

5 x 5 alpha beta with a max ply of 4 and advanced basic function

Turn 2: max depth = 4

Turn 14: max depth = 4

Turn 2: number of nodes explored = 1010

Turn 14: number of nodes explored = 218

Turn 2: wall time = 0.096 seconds

Turn 14: wall time = 0.026 seconds

5 x 5 mini max with a max ply of 3 and advanced utility function

Turn 2: max depth = 3

Turn 14: max depth = 3

Turn 2: number of nodes explored = 12144

Turn 14: number of nodes explored = 1122

Turn 2: wall time = 4.854 seconds

Turn 14: wall time = 0.449 seconds

5 x 5 alpha beta with a max ply of 3 and advanced utility function

Turn 2: max depth = 3

Turn 14: max depth = 3

Turn 2: number of nodes explored = 4087

Turn 14: number of nodes explored = 412

Turn 2: wall time = 1.438 seconds

Turn 14: wall time = 0.121 seconds

5 x 5 alpha beta with a max ply of 4 and advanced utility function

Turn 2: max depth = 4

Turn 14: max depth = 4

Turn 2: number of nodes explored = 11095

Turn 14: number of nodes explored = 4092

Turn 2: wall time = 3.649 seconds

Turn 14: wall time = 1.284 seconds

5 x 5 Analysis:

Overall, the basic utility function either preforms comparably to out preforms the advanced utility function in terms of depth searched, number of nodes searched and wall time, however the basic utility function vastly under preforms in terms of actual gameplay as it is very easy to beat and sometimes fails to even prevent the player from winning

Gameplay is identical across both searches

Alpha Beta search will use equal ply to Mini-Max search, how ever Alpha Beta can be run with a larger maximum ply

Alpha Beta search explores fewer nodes than Mini-Max search at both start and midgame

Alpha Beta search is faster than Mini-Max search at both start and midgame

Overall Analysis:

While the basic utility function may return an action more quickly, search fewer nodes, and use fewer ply. The actions it makes are significantly worse than those of the advanced utility. This is due to it weighting every win it encounters equal and every loss it encounters equal and will simply act based on the first one it encounters regardless of how many turns away it is. This works for the 3x3 board as it is small which mitigates this issue, however on either of the larger board it is probably less challenging to play against the computer picking moves at random.

Therefore, I will be using the advanced utility function for part two of this assignment

2)

- a. My choice of move ordering for each of the three boards is based off the idea that Alpha Beta search will be more efficient when the computer must constantly be blocking the player from winning as this reduces the number of moves it can make. This causes the utility provided by the utility function to be more extreme which causes the alpha beta search to remove more branches and become more efficient.

For 3x3 X's moves:

(1,1), (0,1), (1,2)

For 4x4 X's moves:

(2,2), (1,2), (2,1), (1,0)

For 5x5 X's moves:

(2,2), (3,2), (2,3), (3,3), (3,1), (3,4), (4,4)

b.

3 x 3:

3 x 3 alpha beta with max ply of 9

Turn 2: max depth = 6

Turn 6: max depth = 2

Turn 2: number of nodes explored = 1787

Turn 6: number of nodes explored = 7

Turn 2: wall time = 0.096 seconds

Turn 6: wall time = 0.001seconds

Analysis:

The alpha beta search with the move ordering out preformed the alpha beta search without move ordering overall, with the alpha beta search with move ordering significantly out preforming the alpha beta search without move ordering at turn 6 in depth searched, number of nodes searched and in execution time.

4 x 4 alpha beta with a max ply of 7

Turn 2: max depth = 7

Turn 8: max depth = 4

Turn 2: number of nodes explored = 188342

Turn 8: number of nodes explored = 2182

Turn 2: wall time = 24.979 seconds

Turn 8: wall time = 0.406 seconds

Analysis:

The alpha beta search with the move ordering out preformed the alpha beta search without move ordering overall, with the alpha beta search with move ordering significantly out preforming the alpha beta search without move ordering at turn 8 in depth searched, number of nodes searched and in execution time.

5 x 5 alpha beta with a max ply of 4

Turn 2: max depth = 4

Turn 14: max depth = 4

Turn 2: number of nodes explored = 11095

Turn 14: number of nodes explored = 3144

Turn 2: wall time = 3.532 seconds

Turn 14: wall time = 0.932 seconds

Analysis:

The alpha beta search with the move ordering either out preformed or was equal to the alpha beta search without move ordering overall, with the alpha

beta search with move ordering out performing the alpha beta search without move ordering at turn 14 in number of nodes searched and in execution time.

Overall Analysis:

By forcing the computer to constantly be blocking the player from winning the number of moves it must consider is reduced which reduces the number of nodes it needs to search and thus also improves its wall time and potentially its search depth as well.