

Evaluation Of The Performance Of Each Evaluation function.

All of the heuristic performed better than the AB_Improved agent, over four test, the first evaluation function had an average score of 55.725%, the second evaluation function had an average score of 53.6%, while the third had an average score of 51.425% against the AB_improved agents average of 43.2%. First evaluation function performs the best, next to the second evaluation function and finally the third evaluation function.

Below are the tables of each of the test showing the results for the AB_improved agent and each of the 3 evaluation functions .

Match #	Opponent	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
		W L	W L	W L	W L
1	Random	7 3	9 1	8 2	9 1
2	MM_Open	4 6	5 5	4 6	4 6
3	MM_Center	3 7	7 3	5 5	6 4
4	MM_Improved	2 8	2 8	3 7	4 6
5	AB_Open	6 4	5 5	5 5	7 3
6	AB_Center	6 4	8 2	7 3	4 6
7	AB_Improved	4 6	5 5	6 4	4 6

	Win Rate:	45.7%	58.6%	54.3%	54.3%

Match #	Opponent	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
		W L	W L	W L	W L
1	Random	8 2	9 1	8 2	7 3
2	MM_Open	3 7	6 4	2 8	3 7
3	MM_Center	6 4	7 3	7 3	8 2
4	MM_Improved	1 9	4 6	2 8	4 6
5	AB_Open	4 6	6 4	5 5	8 2
6	AB_Center	5 5	5 5	8 2	6 4

7 AB_Improved 2 | 8 2 | 8 5 | 5 4 | 6

Win Rate: 41.4% 55.7% 52.9% 57.1%

Match # Opponent AB_Improved AB_Custom AB_Custom_2 AB_Custom_3

W | L W | L W | L W | L

1 Random 8 | 2 7 | 3 8 | 2 9 | 1

2 MM_Open 1 | 9 3 | 7 3 | 7 3 | 7

3 MM_Center 3 | 7 6 | 4 7 | 3 6 | 4

4 MM_Improved 2 | 8 4 | 6 4 | 6 1 | 9

5 AB_Open 4 | 6 5 | 5 8 | 2 4 | 6

6 AB_Center 7 | 3 5 | 5 3 | 7 4 | 6

7 AB_Improved 4 | 6 5 | 5 4 | 6 5 | 5

Win Rate: 41.4% 50.0% 52.9% 45.7%

Match # Opponent AB_Improved AB_Custom AB_Custom_2 AB_Custom_3

W | L W | L W | L W | L

1 Random 7 | 3 9 | 1 8 | 2 6 | 4

2 MM_Open 4 | 6 5 | 5 3 | 7 4 | 6

3 MM_Center 5 | 5 5 | 5 8 | 2 4 | 6

4 MM_Improved 3 | 7 6 | 4 5 | 5 3 | 7

5 AB_Open 4 | 6 5 | 5 3 | 7 6 | 4

6 AB_Center 3 | 7 5 | 5 4 | 6 6 | 4

7 AB_Improved 5 | 5 6 | 4 7 | 3 5 | 5

Win Rate: 44.3% 58.6% 54.3% 48.6%

Each of the evaluation functions are essentially the same in principle. Below is a illustration describing the evaluation functions:

Function(gamestate, player)

Return (board's center-player's position) + (number of daughter moves available from move being considered – number of moves available to the opponent)

I factored in two considerations while determining the kind of evaluation function I want to make use of:

1. Centrality of move: Since the isolation piece moved like chess knights, the closer the next move brings the piece to the center of the board, the higher the value returned. This is because a knight at the center of the board has more moves available to it than a knight that is anywhere else.
2. In addition to giving higher scores to moves that have more daughter moves, I also gave more score to how much they minimized the opponent's chance by penalizing the move for moves available to the opponent if that move is made.

The major difference in each of the function is that the first function penalized the number of moves available to the opponent 3 times compared to 2 for the second function and 1 for the third function. This helps to "keep the enemy closer".

I believe that the constants with which the number of enemy moves were multiplied created the difference in performance.