Heuristic Analysis

AB_Custom – Moves forward difference

Outputs a score equal to the difference in number of moves available to the two players with decreasing weight multiplier, for each player summing the number of currently available squares n moves forward:

Function moves_forward_difference(game,player,n) returns score
for move = 1 to n do:
p = player free positions at move in game
o = opponent free positions at move in game
score = score + (1/move) x (p-o)

	a	b	c	d	e	f	g	h	
8	4	3	2	3	2	3	2	3	8
7	3	2	3	4	1	2	1	4	7
6	4	3	2	1	2	3	2	1	6
5	3	2	3	2	3	♞	3	2	5
4	4	3	2	1	2	3	2	1	4
3	3	2	3	4	1	2	1	4	3
2	4	3	2	3	2	3	2	3	2
1	3	4	3	2	3	2	3	2	1
	a	b	c	d	e	f	g	h	

Distance from the f5-square, counted in knight moves

This heuristic achieved an overall 75.1% win rate and 61% win rate against AB Improved.

AB_Custom_2 – Depth first search length score

Create a graph representation of the board where each square index is a vertex with edges to its adjacent knight moves squares.
Run a depth first search on that graph (up to a feasible limit such as 6 or 7)
And return the longest path or limit if reached as the score.

This heuristic achieved an overall 59.9% win rate but was not as good as the Open, Center or Improved Heuristics.

AB_Custom_3 – Breadth first search width score

Similar to the DFS heuristic but run a BFS up to a limit of a visited vertices and return the maximum number of visited vertices from that position.

This heuristic achieved an overall 58.7% win rate but was not as good as the Open, Center or Improved Heuristics.

Tournament Results

```
*************************
        Playing Matches
*************************

Match #    Opponent     AB_Improved     AB_Custom     AB_Custom_2     AB_Custom_3
                        Won | Lost     Won | Lost     Won | Lost     Won | Lost
   1        Random      94  |   6      94  |   6      88  |  12      94  |   6
   2       MM_Open      83  |  17      83  |  17      69  |  31      64  |  36
   3      MM_Center     88  |  12      91  |   9      88  |  12      87  |  13
   4     MM_Improved    76  |  24      81  |  19      70  |  30      57  |  43
   5       AB_Open      55  |  45      54  |  46      36  |  64      39  |  61
   6      AB_Center     60  |  40      62  |  38      41  |  59      32  |  68
   7     AB_Improved    54  |  46      61  |  39      27  |  73      38  |  62
------------------------------------------------------------------------------
        Win Rate:        72.9%          75.1%          59.9%          58.7%

Process finished with exit code 0
```

Win Rates

	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
Random	94%	94%	88%	94%
MM_Open	83%	83%	69%	64%
MM_Center	88%	91%	88%	87%
MM_Improved	76%	81%	70%	57%
AB_Open	55%	54%	64%	39%
AB_Center	60%	62%	41%	32%
AB_Improved	54%	61%	27%	38%
Overall	**72.9%**	**75.1%**	**59.9%**	**58.7%**

Conclusion:

Moves forward heuristic (AB_Custom) performed better than the DFS and BFS heuristics versus each single other agent and heuristic.

AB_Custom is similar to AB_Improved in its evaluation criterion, the moves difference but it predicts its score predicts outcome deeper into the game state tree, inspecting available moves 3 moves forward.

It performed similarly to AB_Improved vs. AB_Open and I speculate that this is either because the moves difference advantage over the open moves heuristic is limited to a subset of possible game states or that again the additional complexity of the former hurts their overall performance vs. the simpler heuristic.

As for AB_Custom_2 and AB_Custom_3, I suspect their low win rate vs. Open, Center and the Improved heuristics is due to the former heuristics complexity and not to their lesser predictive capabilities.

For the DFS and BFS each position evaluation includes graph construction and search and so it consumes too much of the overall move search time.

I believe that using a heuristic which applies DFS in some cases, say when there are only one or two available moves from a position and game move count is over some threshold and using a quicker heuristic otherwise would perform better.

Recommending using *AB_Custom* heuristic based on:

1. *Win Rate:* Beating AB_Improved at 61% of games, AB_Center at 62% of games and AB_Open at 54% of games and being significantly better than the other two tried heuristics.
2. *Complexity:* Being less complex than AB_Custom_2 and AB_Custom_3, with complexity $O(b*d)$ where max depth is 3 and max breadth is 8.
3. *Predictive Capability:* giving an evaluation of the board an additional 3 moves forward from max search depth for both players predicts game outcome well.