

Heuristic description:

- Custom score

First custom heuristic explores the ideal weight assignment to player's moves attribute as well as opponent moves attribute to increase the average win rate using this simple yet effective criteria. Various aggressive, conservative and optimized weight combinations were assigned to maximize win rate.

In addition, this criterion was enhanced by multiplying with number of filled spaces throughout the length of the game given that its more important to trust available moves later in the game to win.

- Custom score 2

The first heuristic has a "cap" to its performance, especially during various stages of the game. In this heuristic, various criterion is used to provide score based on how many turns have been played. During the first quarter, we player to stay as close to the center, hence weighted "own_dist – opponent distance" is used. During next two potential quarters of the game first custom heuristic is used. During last quarter of the game, score is purely based on weighted "available moves" for the player.

Performance of this heuristic was significantly better than first heuristic which confirmed that having different criteria during opening, middle and closing is important.

- Custom score 3

Since the isolation variant uses knight moves, the traditional concepts such as distance to center, don't apply as well. Whereas with knight isolation, the distinctive pattern of moves lets you steal moves from other players.

In this heuristic, if the player has access to the same move as the opponent a higher score is returned. If the player has access to same move as opponent and the amount of moves available is less, then even higher score is returned.

Average win ratio custom heuristics :

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	10	0	10	0	8	2	9	1
2	MM_Open	8	2	7	3	8	2	9	1
3	MM_Center	8	2	9	1	10	0	9	1
4	MM_Improved	8	2	5	5	5	5	8	2
5	AB_Open	6	4	6	4	8	2	6	4
6	AB_Center	6	4	6	4	6	4	8	2
7	AB_Improved	5	5	4	6	3	7	3	7
Win Rate:		72.9%		67.1%		68.6%		74.3%	

Heuristic performance analysis:

Heuristic	Custom Score 1	Custom Score 2	Custom Score 3
Random	<ul style="list-style-type: none"> - Average wins 95% - superior evaluation function leads to victory 	<ul style="list-style-type: none"> - Average wins 100% - superior evaluation function leads to victory 	<ul style="list-style-type: none"> - Average wins 100% - superior evaluation function and depth leads to victory
Minimax	<ul style="list-style-type: none"> - Average wins 80% - Though relies on same basic metrics (available moves) key differentiator is weight multiplier used (filled spaces) 	<ul style="list-style-type: none"> - Average wins 80% - higher depth reaches compared to minimax, regardless of heuristic used by cpu agent - perform better than cpu agents during mid and end game 	<ul style="list-style-type: none"> - Average wins 90% - significantly higher depth reaches compared to minimax, regardless of heuristic used by cpu agent - performs extremely well in long running end games
AlphaBeta	<ul style="list-style-type: none"> - Average wins 64% - Performs well when games are short running 	<ul style="list-style-type: none"> - Average wins 68% - performs well when game under 40 occupied places - performs well when players are on farther away from each other 	<ul style="list-style-type: none"> - Average wins 72% - performs well when game over 35 occupied places - performs well when both players are closer to each other

Heuristic recommendation:

Out of the heuristics I have built, I would recommend **custom_score_3** due to the following:

1. Win rate is consistently higher than heuristics that purely take the depth or available moves into account
2. Due to different function used during different stages of the game, optimal scores are returned when playing against variety of opponents
3. Due to simplicity of the heuristic, the evaluation runs quicker hence allowing to reach depth larger than other two functions. On average function reached depth of 19,100 when playing against AB_Improved