

Heuristic Functions:

For this analysis, the three heuristic functions tested are as follows:

1. Custom_score:

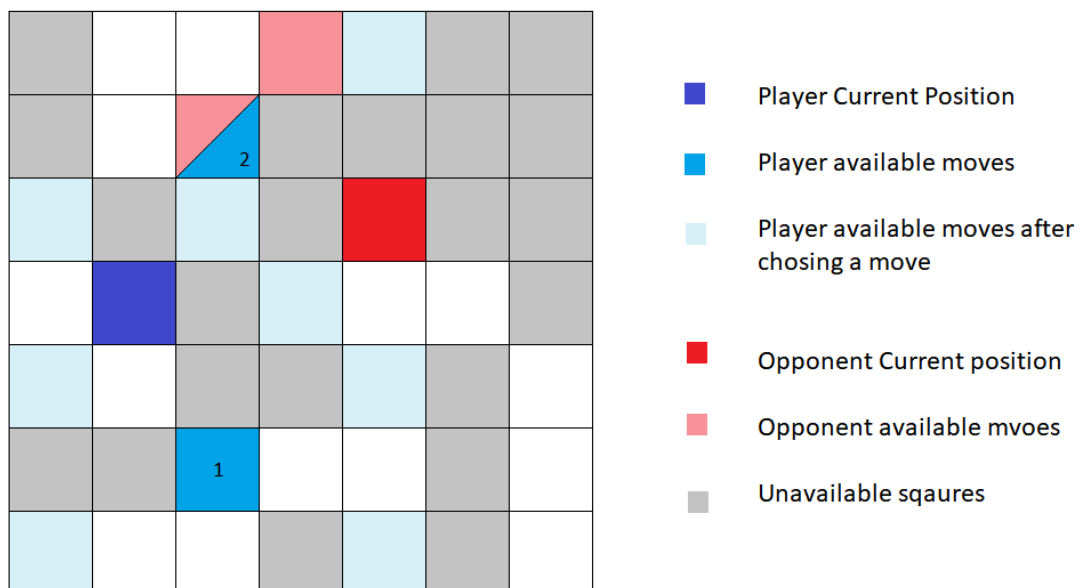
$$3 \left(\frac{own_{moves}}{opponent_{moves}} \right) + \frac{1}{(h-y)^2 + (w-x)^2}$$

- Here (h,w) are board center's coordinates, and (x,y) is the location of our CPU player
- This Heuristic mainly relies on maximizing the ratio between the CPU player's legal moves and opponent's legal moves.
- The secondary component tries to maximize the inverse of the distance from the center.
- The secondary component will be the deciding factor when there is a tie between multiple moves with the same player moves/opponent moves ratio.
- This function combines the custom_score_2 and custom_score_3 heuristics.

2. Custom_score_2:

$$\frac{own_{moves}}{opponent_{moves}}$$

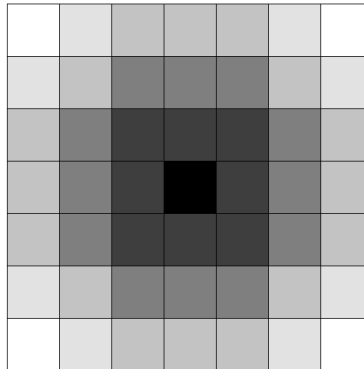
- This heuristic tries to maximize the ration between the CPU player's available legal moves and the opponent's legal moves.
- The heuristic is different from Improved_score heuristic as it rewards minimizing the opponent moves more.
- Consider the scenario below (only for visualization, may not be accurate) where node_1 has player_moves = 5 and opponent_moves = 2; and node_2 has player_moves = 3 and opponent_moves = 1. The Improved_score heuristic will chose node_1 since the difference between the player and opponent moves is greater. However, Custom_score_2 will choose node_2 since the ratio 3:1 is greater than 5:2.



3. Custom_score_3:

$$\frac{1}{(h - y)^2 + (w - x)^2}$$

- Again (h,w) is the center of the board and (x,y) is the location of our player
- In this heuristic we try to maximize the inverse of the distance from the center
- This will guide our player to give priority to the central squares which have the max available moves and then move its way outwards.



Results:

To improve our analysis the number of matches was increased from 5 to 10. The results of tournament.py are shown below.

Match #	Opponent	AB_Improved		AB_Custom		AB_Custom_2		AB_Custom_3	
		Won	Lost	Won	Lost	Won	Lost	Won	Lost
1	Random	20	0	19	1	19	1	19	1
2	MM_Open	14	6	17	3	13	7	12	8
3	MM_Center	19	1	17	3	19	1	19	1
4	MM_Improved	14	6	15	5	17	3	15	5
5	AB_Open	13	7	12	8	12	8	10	10
6	AB_Center	13	7	12	8	12	8	11	9
7	AB_Improved	8	12	10	10	9	11	9	11

Win Rate:		72.1%		72.9%		72.1%		67.9%	

In the results, we see that our custom_score heuristic could outperform the Improved_score heuristic but only with a very small margin. It would not be advisable to suggest that either of the heuristic was significantly better. Custom_score_2 was equivalent to Improved_score in win rate. Custom_score_3 was worse off compared to the other 2 custom heuristics and Improved_score. All heuristics could provide only a slight advantage when competing against other functions tested. The Alpha-Beta pruning however seems to provide a huge advantage when designing an isolation player.