**Heuristic Analysis**

Three Heuristic functions which are used in the analysis are :

1. **Custom\_score** *(First Heuristic)* : In this heuristic during the opening game the player gives more value to the positions where more options are available to play in future and at the same time tries to obstruct the opponent player's move. But, more priority is given to the number of moves that player can play in future as opposed to obstruction of moves of opponent player.  
    But, during the end game more focus is given to obstructing the opponent player's move as compared to number of moves available for the player.

During opening game :

heuristic value = 2 \* num\_of\_available\_moves - num\_of\_opponent\_moves

During end game :

heuristic value = num\_of\_available\_moves - 2\*num\_of\_opponent\_moves

1. **Custom\_score\_2** *(Second Heuristic)* : According to this heuristic , first it is checked if number of moves available to player is same as number of moves available to opponent. If it is different, then heuristic value is set as number\_of\_player\_moves - number\_of\_opponent\_moves.

But, if the above numbers are same, then instead of returning zero, additional calculations are done which checks the centrality of the players location using the Manhattan distance as distance metric. The more in center the player's position is, the better is its heuristic value.

1. **Custom\_score\_3** *(Third Heuristic)* : In this heuristic, the opening game is kept simple by using just the open move heuristic. As the number of played moves increases after a certain threshold, the player starts chasing after opponent.

During the opening game :

heuristic value = number of moves available to play

During the end game :

heuristic value = num\_of available\_moves - 2\*num\_of\_opponent\_moves

Following are the observations which are found after running the tournament.py **three times** at different times in a day :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Opponent | AB\_Improved | | Custom\_score | | Custom\_score\_2 | | Custom\_score\_3 | |
|  | Won | Lost | Won | Lost | Won | Lost | Won | Lost |
| Random | 27 | 3 | 28 | 2 | 24 | 6 | 25 | 5 |
| MM\_Open | 17 | 13 | 22 | 8 | 17 | 13 | 19 | 11 |
| MM\_Center | 26 | 4 | 26 | 4 | 23 | 7 | 21 | 9 |
| MM\_Improved | 18 | 12 | 17 | 13 | 18 | 12 | 21 | 9 |
| AB\_Open | 16 | 14 | 18 | 12 | 18 | 12 | 16 | 14 |
| AB\_Center | 16 | 14 | 16 | 14 | 17 | 13 | 17 | 13 |
| AB\_Improved | 15 | 15 | 17 | 13 | 18 | 12 | 15 | 15 |
| Win Rate | 64.3% | | 68.5% | | 64.2% | | 63.8% | |

The tournament.py is run different times of the day because it may be possible that at certain times the computer may be running some other intensive process in background, thus affecting the result somehow.

**It can be observed from the above table that Custom\_score which was following the *First Heuristic* performed the best on average.** But, it is easily seen that beating the AB\_improved Heuristic which is simply the heuristic "own\_move - opponent moves" is very difficult indeed.

The *First Heuristic* was better than the *Second Heuristic* because in the second heuristic, the calculation of centre distance is more costly as compared to simple heuristic calculation in first heuristic. Thus, during the opening game when there are many nodes to be evaluated the performance is reduced(lesser tree depth is analyzed) due relatively complex heuristic evaluation of second heuristic.

The *First Heuristic* performed fairly as compared to *Third Heuristic* because empirically it seems that in the opening game considering an extra factor of opponents move is better than just considering number of open moves heuristic which is done in the third heuristic.