# Analysis of the Heuristics

Following heuristics were chosen and implemented

* **H1** (AB\_Custom): *# of player’s moves – 2 x # of opponent’s moves*
* **H2** (AB\_Custom\_2): *The square of the distance between the player and the opponent, calculated as*

*y\_i, x\_i = game.get\_player\_location(i) , where i is either the player or the opponent*

*score = (y\_pl - y\_opp)2 + (x\_pl - x\_opp)2*

* **H3** (AB\_Custom\_3): *# of player’s moves / # of opponent’s moves*

If # of player’s moves = 0, player loses

If # of opponent’s moves = 0, opponent loses

The results of the tournament with these heuristics are as follows:

Table 1 Results of the Tournament

                        \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

                             Playing Matches

                        \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 Match #   Opponent    AB\_Improved   AB\_Custom   AB\_Custom\_2  AB\_Custom\_3

                        Won | Lost   Won | Lost   Won | Lost   Won | Lost

    1       Random      10  |   0     9  |   1     8  |   2    10  |   0

    2       MM\_Open      8  |   2     6  |   4     5  |   5     6  |   4

    3      MM\_Center     7  |   3     9  |   1     9  |   1    10  |   0

    4     MM\_Improved    6  |   4    10  |   0     7  |   3     8  |   2

    5       AB\_Open      5  |   5     7  |   3     3  |   7     4  |   6

    6      AB\_Center     2  |   8     5  |   5     4  |   6     8  |   2

    7     AB\_Improved    4  |   6     6  |   4     4  |   6     7  |   3

--------------------------------------------------------------------------

           Win Rate:      60.0%        74.3%        57.1%        75.7%

As it can be seen **H1** performed quite well, as it aggressively tries to minimize opponent’s available moves (hence the factor 2). In fact, **H1** is very similar to AB\_Improved, but is more aggressive because of the factor 2. Hence compared to AB\_Improved, **H1** has been more successful against the other opponents:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AB\_Improved | H1 | H2 | H3 |
| Random | 100.0% | 90.0% | 80.0% | 100.0% |
| MM\_Open | 80.0% | 90.0% | 50.0% | 60.0% |
| MM\_Center | 70.0% | 90.0% | 90.0% | 100.0% |
| MM\_Improved | 60.0% | 100.0% | 70.0% | 80.0% |
| AB\_Open | 50.0% | 70.0% | 30.0% | 40.0% |
| AB\_Center | 20.0% | 50.0% | 40.0% | 80.0% |
| AB\_Improved | 40.0% | 60.0% | 40.0% | 70.0% |
|  |  |  |  |  |
| Overall | 60.0% | 74.3% | 57.1% | 75.7% |

On the other hand, **H1** vs. AB\_Improved provided mixed results. In certain cases, both were equally successful (5-5), in certain cases **H1** had more wins, in certain cases, AB\_Improved.

One can also observe that **H3** performs quite well against AB\_Improved. This is probably due to the fact that it is actually a simpler heuristic than AB\_Improved (and **H1**), which allows the search to go deeper.

Overall, **H3** seems to be the best heuristic, as

1. it is very simple to calculate (low complexity) and hence it’s execution time is shorter
2. it performs better on average (overall higher win-rate of 75.7%) and
3. with alpha-beta pruning it can explore the three deeper
4. it takes advantage of the opponent’s moves
5. performs better against AB\_Improved