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

Match	Opponent	AB_Improved	AB_Custom	AB_Custom_2	AB_Custom_3
		Won | Lost	Won | Lost	Won | Lost	Won | Lost
1	Random	9 | 1	9 | 1	10 | 0	9 | 1
2	MM_Open	6 | 4	6 | 4	6 | 4	5 | 5
3	MM_Center	8 | 2	9 | 1	9 | 1	9 | 1
4	MM_Improved	5 | 5	6 | 4	5 | 5	5 | 5
5	AB_Open	5 | 5	4 | 6	3 | 7	5 | 5
6	AB_Center	3 | 7	5 | 5	5 | 5	7 | 3
7	AB_Improved	6 | 4	5 | 5	2 | 8	2 | 8
	Win Rate:	60.0%	62.9%	57.1%	60.0%

The heuristics used are as follows:

I chose this *AB_Custom* heuristics because it had consistent performance against AB_Improved. The same cannot be said for *AB_Custom_2* and *AB_Custom_3* because while they perform Random moves by opponent, against AB_Improved, they fail quite miserably.

By weighting opponent's move twice as much as it own and taking a difference, AB_Custom amplifies the separation between max and min values that much more. The only opponent that it lost out to is AB_Open.

One of the fascinating aspects for me was the impact of weighting factor used in AB_Custom and AB_Custome_3 on the outcome. Changing the opponent move weighting factor from 2 to 10 caused a noticeable degradation in performance.