

# Heuristic Analysis: Isolation

Esteban Rodríguez Betancourt

April 28, 2018

## Strategy 1

This strategy tries to maximize both the number of moves available and the difference between moves available to the player and its opponent. The formula is simple to calculate:

$$p^2 - o^3$$

where  $p$  is the number of available moves to the player, and  $o$  is the number of moves available to the opponent.

This was implemented as `custom_score`.

The performance of this strategy is shown in table 1.

Opponent	AB_Improved		AB_Custom	
	Won	Lost	Won	Lost
Random	9	1	10	0
AB_Improved	6	4	6	4
AB_Custom	4	6	8	2
AB_Custom_2	3	7	7	3
AB_Custom_3	5	5	7	3
AB_Open	5	5	4	6
AB_Center	4	6	9	1
MM_Improved	7	3	9	1
MM_Custom	8	2	6	4
MM_Custom_2	6	4	9	1
MM_Custom_3	8	2	6	4
MM_Open	10	0	7	3
MM_Center	9	1	8	2
Win Rate:	64.6%		73.8%	
No self vs self	65.0%		73.33%	

Table 1: *AB\_Custom* Tournament Results

## Strategy 2

This strategy just tries to maximize the number of available moves to the player, using the formula:

$$p^2$$

To me it makes sense to test it, as more moves makes less probable to loss.

This was implemented as `custom_score_2`.

The performance of this strategy is shown in table 2.

Opponent	AB_Improved		AB_Custom_2	
	Won	Lost	Won	Lost
Random	8	2	10	0
AB_Improved	4	6	5	5
AB_Custom	4	6	7	3
AB_Custom_2	6	4	5	5
AB_Custom_3	6	4	6	4
AB_Open	5	5	5	5
AB_Center	6	4	5	5
MM_Improved	8	2	6	4
MM_Custom	8	2	8	2
MM_Custom_2	7	3	7	3
MM_Custom_3	8	2	10	0
MM_Open	6	4	4	6
MM_Center	8	2	9	1
Win Rate:	64.6%		66.9%	
No self vs self	66.67%		68.33%	

Table 2: *AB\_Custom\_2* Tournament Results

## Strategy 3

This strategy just tries to minimize the number of available moves to the opponent, using the formula:

$$-o^2$$

Like in strategy 2, the idea was to make more probable to the opponent to lose, by forcing less and less available moves.

This was implemented as `custom_score_3`.

The performance of this strategy is shown in table 3.

Opponent	AB_Improved		AB_Custom_3	
	Won	Lost	Won	Lost
Random	10	0	10	0
AB_Improved	4	6	1	9
AB_Custom	4	6	4	6
AB_Custom_2	3	7	6	4
AB_Custom_3	6	4	6	4
AB_Open	5	5	5	5
AB_Center	6	4	6	4
MM_Improved	8	2	7	3
MM_Custom	9	1	7	3
MM_Custom_2	8	2	6	4
MM_Custom_3	6	4	10	0
MM_Open	7	3	5	5
MM_Center	10	0	8	2
Win Rate:	66.2%		62.3%	
No self vs self	68.33%		62.50%	

Table 3: AB\_Custom\_3 Tournament Results

## Results Analysis

After testing *Improved*, *Custom*, *Custom\_2* and *Custom\_3* (the four in AB pruning version) against everyone (*Improved*, *Custom*, *Custom\_2* and *Custom\_3*. both alpha-beta pruning and minimax) the first obvious result is that alpha-beta pruning performs better than minimax.

The winning rate of AB pruning algorithms vs minimax algorithms was 79.29% which makes sense, as AB pruning is capable of exploring wider than minimax.

Scoring Function	Win Rate
<i>AB_Improved</i>	66.67%
<i>AB_Custom</i>	73.33%
<i>AB_Custom_2</i>	68.33%
<i>AB_Custom_3</i>	62.50%

Table 4: Winning Rate per scoring function

As shown in *table 4* *AB\_Custom* and *AB\_Custom\_2* have higher winning rate than *AB\_Improved*.

As a curious fact, *AB\_Custom* and *AB\_Custom\_2* played 20 against each other, and each won 10 games each. This could be caused as *Custom* is a more complex score function, in fact it do twice the work done by *Custom\_2*. In that case, *Custom* may be a better score function, but *Custom\_2* is capable to analyse a bit more branches.

After the tournament measurements, we can claim that *Custom* scoring function works better than *Custom\_2* and *Custom\_3* because:

	AB Imp	ABC	ABC 2	ABC 3
AB Improved	100	60	57.5	55
AB Custom	40	100	50	35
AB Custom 2	42.5	50	100	50
AB Custom 3	45	65	50	100
AB Open	50	40	50	50
AB Center	53.3	90	50	60
Random	90	100	100	100

Table 5: Winning Rates Percent

- As shown in *table 4*, its win rate was higher: 73.33%. The next one was *Custom\_2* with 68.33%.
- As shown in *table 5*, it had a win rate higher against *Improved*, *Custom\_3*, *Center* and *Random*.
  - It ties against *Custom\_2* with *Custom\_3*, with 50% win rate each.
  - It performs worse against *Open*, where *Improved*, *Custom\_2* and *Custom\_3* had a higher winning rate: 50% versus just 40%.
- Custom* tries both to penalize opponent available moves, and reward having own available moves. *Custom\_2* and *Custom\_3* just tries to optimize one of those metrics.
- Custom* penalization is higher when the opponent has more available moves.