Heuristic Analysis: Isolation

Esteban Rodríguez Betancourt

April 23, 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 the table below:

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%	

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 the table below:

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%	

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 the table below:

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%	

Results Analysis

After testing Improved, Custom, Custom_2 and Custom_3 (the four in AB pruning version) against everyone (Improved, Custom, Custom2 and Custom3. 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.

Also, the win rate of the scoring functions is:

Scoring Function	Win Rate
AB_Improved	66.67%
AB_Custom	73.33%
AB_Custom_2	68.33%
AB_Custom_3	62.50%

So AB_Custom and AB_Custom_2 have higher winning rate than AB_Improved.

As a curious fact, AB_Custom and AB_Custom 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.