Heuristic Analysis

by Oleg Polosin

Method

I wrote down the score formula in a generalized form as:
$S_{final} = compare(S_{my}, S_{opponent})$,
where S_{final} - it's a final score for particular game state, S_{my} and $S_{opponent}$ - my and opponent's scores correspondingly, $compare$ - function to compare players' scores.

Let's consider "AB_Improved" score function:
$S_{final} = m_{my} - m_{opponent}$.
Here $S_{my} = m_{my}$ is a number of my legal moves, $S_{opponent} = m_{opponent}$ is a number of opponent legal moves, $compare(S_{my}, S_{opponent}) = S_{my} - S_{opponent}$ - comparing function.

I explored different $compare$ and $S_{<player>}$ functions independently. After that I chose the best ones and combined them to get even better results.

Comparing Functions

To evaluate comparing functions, I fixed $S_{<player>}$ function as a number of legal moves for each player and explored the following functions:

1. Scores Difference

It's a difference between players' scores, where opponent's score multiplied by factor:

$S_{final} = S_{my} - aS_{opponent}$, where $a \in (0, \infty)$.

2. Scores Ratio

It's a ratio of my score and opponent score multiplied by factor:

$S_{final} = \frac{S_{my}}{aS_{opponent}}$, where $a \in (0, \infty)$.

3. Improved Scores Difference

It's a first score divided by a sum of my score and opponent's score multiplied by factor:

$S_{final} = \frac{S_{my} - aS_{opponent}}{S_{my} + bS_{opponent}}$, where $a, b \in (0, \infty)$.

Player Score Functions

To evaluate player score functions, I fixed $compare$ function as difference between scores and considered different score functions for different players:

1. Legal Moves

It's a number of player's legal moves multiplied by factor:
$S = am$, where $a \in (0, \infty)$, m - number of player's legal moves.

2. Deep Legal Moves

For each player's legal move I'm looking for the next legal moves. It can be done for any deep. For deep=1 score will be computed as number of moves of first level multiplied by 65 plus number of next moves for each first move. First moves should be more valuable than deeper ones and never overscored by next level of moves. For example, 2 legal moves with next 3 legal moves each better than 1 legal move with next 7 legal moves. In theory we can have maximum 8 legal moves for first turn and 8 legal moves for each of those moves for next turn (7 actually, but I generalized to 8 to simplify) and so on. So a branching factor of score varies from 0 to 64, we need to use base of 65 to prevent overscoring previous levels by next levels. I used the following formula for this score:

$$S(l_c, l_p, d) = \begin{cases} |M_{l_c}|, & \text{if } d = 0 \\ 65^d |M_{l_c}| + \sum_{m, m \neq l_p}^{M_{l_c}} S(m, l_c, d-1), & \text{if } d > 0 \end{cases},$$

where l_c - current player's location, l_p - previous player's location, M_{l_c} - set of legal moves for particular location, $|M_{l_c}|$ - number of moves, d - depth for which we want to compute a score.

3. Weighted Legal Moves

For this heuristic I looked at not just the number of legal moves, but how they are close to the center. I used the following formula for this score:

$$S = \sum_{m}^{M} \left(a - \frac{(c_x - m_x)^2 + (c_y - m_y)^2}{c_x^2 + c_y^2} \right),$$

where $a \in (1, \infty)$, M - set of player's legal moves m_x, m_y - coordinates of particular move, c_x, c_y - coordinates of the board center. The larger a, the more score depends on number of legal moves; the smaller a, the more it depends on closeness of moves to the center.

4. Weighted Player Position

Previous heuristic is difficult to calculate, because we need to go through each legal move. It's better to spend that time to look up one level more in the search tree. This heuristic has similar idea, but easier for computations. I take the number of player's legal moves and multiply it on score for player's distance to center:

$$S = m \left(a - \frac{(c_x - p_x)^2 + (c_y - p_y)^2}{c_x^2 + c_y^2} \right),$$

where $a \in (1, \infty)$, m - number of player's legal moves, p_x, p_y - coordinates of the player's position, c_x, c_y - coordinates of the board center. The larger a, the more score depends on number of legal moves; the smaller a, the more it depends on closeness of player to the center.

Results

Comparing Function

"Improved Scores Difference" function with parameters $a = 1$ and $b = 1$ gave me the best results for comparing score functions. I suppose this score works better because it takes into account that the difference between players' scores is more valuable when fewer legal moves are left.

For example, we have 2 boards to evaluate. On the first board we have 7 legal moves and our opponent has 5 legal moves, on the second board we have 4 legal moves and our opponent has 2. The second board is better for us because opponent has less chances to decrease the gap between moves before game ends. If we're using "Scores Difference" function, scores for these two boards will be the same: 2. If we're using "Improved Scores Difference", the second board score will be higher: ⅓ against ⅙.

See "AB_Custom_3" column in the evaluation results below.

Players' Scores

For the players' scores I chose the "Weighted Player Position" function with parameter $a = 1.7$ for "my" score and the "Legal Moves" function with $a = 1$ for opponent's score. They gave me the best results. This function can be written as:

$$S = m_{my}\left(1.7 - \frac{(c_x - p_x)^2 + (c_y - p_y)^2}{c_x^2 + c_y^2}\right) - m_{opponent},$$

where m_{my} - number of "my" legal moves, $m_{opponent}$ - number of opponent's legal moves, p_x, p_y - coordinates of "my" position, c_x, c_y - coordinates of the board center.

See "AB_Custom_2" column in the evaluation results below.

Final Score

For the final score function I've combined 2 previous custom scores. Final score function can be written as:

$$S_{final} = \frac{P_{my} - m_{opponent}}{P_{my} + m_{opponent}},$$

where P_{my} - my "Weighted Player Position" score with $a = 1.7$, $m_{opponent}$ - number of opponent's legal moves.

Evaluation

To evaluate the custom score functions I made 4 runs by 100 games each. In the results tables below you can see that in average "AB_Custom" player outperforms "AB_Improved" player on about 2.5%.

Conclusion

Simpler score functions work better than complex ones: we're spending a time on traversing more levels in the search tree rather than on long computations for each move. Efficiency of simple score functions strongly depend on chosen weights, and it's a very difficult process to find them manually. For this reason we need to apply machine learning approach running on millions of games to find optimal weights for our score functions.

Evaluation Results

Run 1:

Opponent	**AB_Improved**		**AB_Custom**		**AB_Custom_2**		**AB_Custom_3**	
	Won	Lost	Won	Lost	Won	Lost	Won	Lost
Random	93	7	94	6	98	2	98	2
MM_Open	77	23	75	25	70	30	77	23
MM_Center	92	8	88	12	90	10	89	11
MM_Improved	77	23	77	23	75	25	77	23
AB_Open	52	48	54	46	50	50	48	52
AB_Center	57	43	65	35	56	44	56	44
AB_Improved	53	47	58	42	50	50	52	48
Win Rate	**71.60%**		**73.00%**		**69.90%**		**71.00%**	

Run 2:

Opponent	**AB_Improved**		**AB_Custom**		**AB_Custom_2**		**AB_Custom_3**	
	Won	Lost	Won	Lost	Won	Lost	Won	Lost
Random	90	10	93	7	93	7	89	11
MM_Open	77	23	79	21	66	34	74	26
MM_Center	86	14	92	8	84	16	85	15
MM_Improved	75	25	61	39	77	23	65	35
AB_Open	50	50	57	43	60	40	43	57
AB_Center	56	44	61	39	62	38	68	32
AB_Improved	53	47	49	51	46	54	52	48
Win Rate	**69.60%**		**70.30%**		**69.70%**		**68.00%**	

Run 3:

Opponent	**AB_Improved**		**AB_Custom**		**AB_Custom_2**		**AB_Custom_3**	
	Won	Lost	Won	Lost	Won	Lost	Won	Lost
Random	89	11	91	9	95	5	92	8
MM_Open	78	22	79	21	77	23	79	21
MM_Center	89	11	86	14	93	7	86	14
MM_Improved	69	31	73	27	82	18	66	34
AB_Open	53	47	60	40	57	43	50	50
AB_Center	61	39	68	32	64	36	60	40
AB_Improved	46	54	53	47	54	46	60	40
Win Rate	**69.30%**		**72.90%**		**74.60%**		**70.40%**	

Run 4:

Opponent	**AB_Improved**		**AB_Custom**		**AB_Custom_2**		**AB_Custom_3**	
	Won	Lost	Won	Lost	Won	Lost	Won	Lost
Random	92	8	94	6	96	4	97	3
MM_Open	74	26	71	29	83	17	79	21
MM_Center	88	12	89	11	88	12	87	13
MM_Improved	72	28	75	25	73	27	74	26
AB_Open	45	55	59	41	53	47	42	58
AB_Center	52	48	59	41	61	39	68	32
AB_Improved	48	52	58	42	51	49	52	48
Win Rate	**67.30%**		**72.10%**		**72.10%**		**71.30%**	

Average win rate:

	AB_Improved	**AB_Custom**	**AB_Custom_2**	**AB_Custom_3**
Win Rate 1, %	71.60	73.00	69.90	71.00
Win Rate 2, %	69.60	70.30	69.70	68.00
Win Rate 3, %	69.30	72.90	74.60	70.40
Win Rate 4, %	67.30	72.10	72.10	71.30
Average Win Rate, %	**69.45**	**72.08**	**71.58**	**70.18**