Heuristic Analysis

Thilina Shihan Weerathunga

June 2017

1 Parameters used for heuristic definitions.

- relative_mobility:- Comparing the number of my moves to the opponents and normalizing it. Parameter value ranges from (-1, 1).
- opponent_block_ability :- Winning strategy for the game is to isolate and block the opponent. This parameter measures the ability to block the opponent on their next move. Here the intersection of my_moves and opp_move is measured. If it is not empty, then maximum more play has the ability to block some of the opp_moves.
- corner_domination:- Less number of corner moves relative to the opponent is better/penalizing the moves for the maximizing player which are in the corner. Being in a corner in late game (less than 25% of the board is empty) is bad.
- center_ability:- Ability to take over the center on next move.
- my_mobility:- length(my_moves), Number of legal moves maximizing player has.
- relative_center_domination:- Distance to the center relative to the opponent's The shorter the distance as compared to the opponents is better.

2 Strategies used for each custom_score functions.

2.1 custom_score

- relative_mobility (My mobility vs opponent's (normalized [-1.0 ... 1.0])).
- opponent_block_ability (Ability to block opponent on next move).
- corner_domination (Less number of corner moves relative to the opponent is better/penalizing the moves for the maximizing player which are in the corner).

- Score is calculated using a weighted method for each parameter.

return float(50 * relative_mobility + 40 * opponent_block_ability + 10*corner_domination)

2.2 custom_score_2

- relative_mobility (My mobility vs opponent's (normalized [-1.0 ... 1.0])).
- center_ability (Ability to take over the center on next move).
- my_mobility = length(my_moves),Number of legal moves maximizing player has.
- opponent_block_ability (Ability to block opponent on next move).
- Score is calculated using a weighted method for each parameter.

return float(60 * relative_mobility + 10 * my_mobility + 20 * center_ability + 10 * opponent_block_ability)

2.3 custom_score_3

- my_mobility = length(my_moves),Number of legal moves maximizing player has.
- opponent_block_ability (Ability to block opponent on next move)
- relative_center_domination (Distance to the center relative to the opponent's (The shorter the distance as compared to the opponent's is better)

returnfloat(40 * opponent_block_ability + 30 * my_mobility + 30 * relative_center_domination)

3 Results

Performances of above mentioned heuristics are evaluated by running tournament.py script. Final result is summarized below. Number of matches played in each case = 1000.

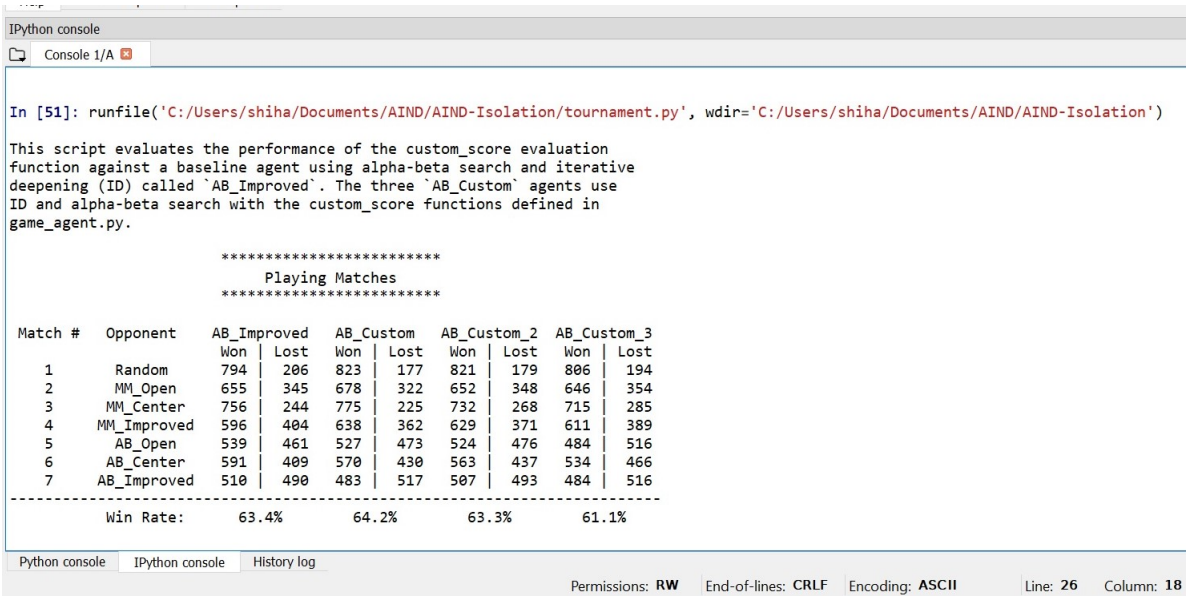

Figure 1: Performance of defined custom_score_methods