Heuristic Analysis

Following 3 heuristic evaluation functions are implemented and analyzed by using them in *CustomPlayer* agent in comparison to the ID_Improved agent.

1. `their_moves` :
   This evaluation function returns the negative of the number of legal moves available for the opponent agent for non-terminal states. For terminal states it returns +Inf if agent won else returns -Inf. So the function penalizes states with high number of opponent moves compared to states with lower number of opponent moves.

2. `blank_diff_theirs` :
   This function returns the number of blank squares that cannot be occupied by the opponent in its next move for non-terminal states. For terminal states it returns +Inf if agent won else returns -Inf.

3. `moves_diff`:
   This function takes a parameter called gamma, and returns a score calculated as `num_my_moves – ( num_opponent_moves * gamma )` for non-terminal states. For terminal states it returns +Inf if agent won else returns -Inf.

Performance of evaluation functions when used with agent with iterative deepening.

1) ID_Improved (reference eval function) vs other agents in the tournament

ID_Improved VS	Won	Lost	win_pct
Random	19	1	95
MM_Null	18	2	90
MM_Open	16	4	80
MM_Improved	10	10	50
AB_Null	16	4	80
AB_Open	13	7	65
AB_Improved	13	7	65
Sum	105	35	
Win pct	75		

2) their_moves vs other agents in the tournament.

their_moves VS	Won	Lost	win_pct
Random	18	2	90
MM_Null	18	2	90
MM_Open	14	6	70
MM_Improved	15	5	75
AB_Null	15	5	75
AB_Open	13	7	65
AB_Improved	9	11	45
Sum	102	38	
Win pct	72.86		

Note that the win percentages of their_moves are noticeably lesser than the reference function.

3) blanks_diff_theirs

blanks_diff_theirs VS	Won	Lost	win_pct
Random	18	2	90
MM_Null	19	1	95
MM_Open	13	7	65
MM_Improved	11	9	55
AB_Null	15	5	75
AB_Open	15	5	75
AB_Improved	15	5	75
Sum	106	34	
Win pct	75.71		

The performance of blank_diff_theirs seem more or less closer to that of reference when compared with other standard agents.

4a) moves_diff eval function with parameter 1.5 vs standard agents.

moves_diff_1.5 VS	Won	Lost	win_pct
Random	19	1	95
MM_Null	18	2	90
MM_Open	11	9	55
MM_Improved	13	7	65
AB_Null	16	4	80
AB_Open	13	7	65
AB_Improved	11	9	55
Sum	101	39	
Win pct	72.14		

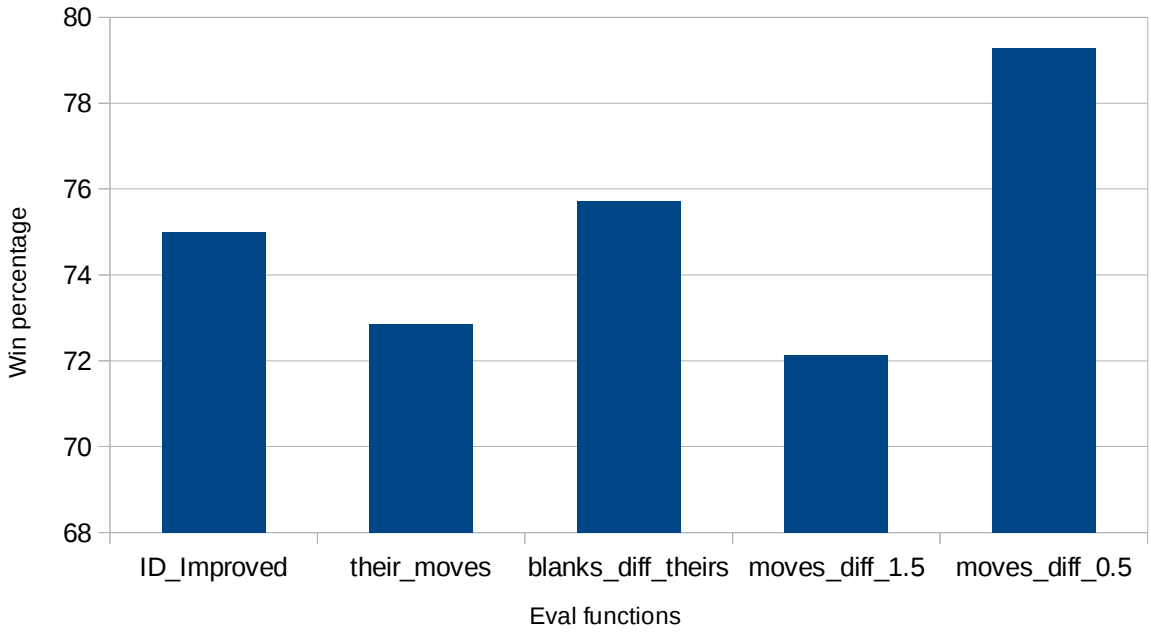
This eval function seems to fare less successful against standard agents when compared to ID_improved reference.

4b) moves_diff eval function with parameter 0.5 vs standard agents.

moves_diff_0.5 VS	Won	Lost	win_pct
Random	19	1	95
MM_Null	19	1	95
MM_Open	14	6	70
MM_Improved	14	6	70
AB_Null	17	3	85
AB_Open	14	6	70
AB_Improved	14	6	70
Sum	111	29	
Win pct	79.29		

There is a significant improvement in win percentage (overall and individual) when compared with the reference.

Following bar chart shows the overall win percentage of agents using the different evaluation functions with iterative deepening and alpha-beta pruning.



From the above statistics, moves_diff (with param gamma = 0.5) can be preferred over the other functions due to following reasons :

1. The agent that uses it has the best win percentage when compared to the rest of the evaluation functions considered (79% vs 75% of reference).

2. move_diff_0.5 function is easy to compute. It involves only an extra floating point multiplication when compared to 'improved' reference function.

3. Even though the agent that used move_diff_0.5 is only comparable in performance to the reference against weak agents like Random and MM_Null, it shows significantly higher performance against *tougher and comparable opponents* (agents that use sophisticated techniques like Alpha-Beta pruning and those using non-trivial evaluation functions) like *AB_Improved* and *MM_Improved*. For example the reference agent won only 50% times against MM_Improved while move_diff_0.5 agent won 70% of the times against the same opponent.