Building an Adversarial Game Playing Agent

Games experiment – 100 rounds with fair matches

Results :
depth of 3

Opponent	Matches	Time Limit (ms)	Win Percentage	Execution time minutes(m). Seconds (s)
GREEDY	200	150	83.0%	2m 9.399s
GREEDY	200	150	82.8%	2m 8.292s
GREEDY	200	150	81.2%	2m 8.082s
MINIMAX	200	150	64.5%	2m 40.773s
MINIMAX	200	150	60.2%	2m 42.839s
MINIMAX	200	150	55.2%	2m 40.355s
SELF	200	150	50.0%	3m 2.505s
SELF	200	150	50.0%	3m 5.587s
SELF	200	150	50.0%	3m 3.645s

Example :
Running 200 games:
++++-+-+--++++++++++-++--+-+-++++++--+-++++++++++++++++++--++++-+++-+-+-
+++-+++++++++++++++-+++++-+++-++++++++++++++++++++++++-++--++-+-
++++++++++++++-+-+++-+++++++++++++++-+-++-++++++++-++++++++-+--++++
Your agent won 83.0% of matches against Greedy Agent

Questions :

What features of the game does your heuristic incorporate, and why do you think those features matter in evaluating states during the search?

#(own_liberties) - 3 * #(opp_liberties) + #remaining_own_liberties - 3 * #remaining_opp_liberties

- Providing more open liberties correspond to better probability of winning the game hence the modified heuristic.
  This adds a bias that triples weight of opposition liberties : #(own_liberties) - 3 * #(opp_liberties)

- Adding the above with #remaining_own_liberties - 3 * #remaining_opp_liberties (same bias for the opponent liberty weight)

Analyze the search depth your agent achieves using your custom heuristic. Does search speed matter more or less than accuracy to the performance of your heuristic?

Opponent	**Player Search Depth**	**Time Limit ms**	**Win pct**	**Execution time minutes(m). Seconds (s)**
MINMAX	2	150	42.2%	1m 41.961s
MINMAX	3	150	61.0%	2m 43.188s
MINMAX	4	150	60.5%	5m 8.071s
Greedy	2	150	64.0%	1m1.020s
Greedy	3	150	79.5%	2m 8.672s
Greedy	4	1500	0.0 %	Timeout

From the above table, the heuristic works good with search depth level of 3.However, the agent does not gain much by going deeper in the search levels. Search time increases significantly with increased depth