

Reversi(Alpha-Beta-Pruning)-Observations

2011cs1012

The experiment was carried out with the following:

BOT:<http://www.othelloonline.org/othello-version2.php>(BLACK)

CPU:The C++ program(WHITE)

Heuristics used:

1. Coin parity(disc_count)(w2)
2. Weighted Coin Parity(weighted_disc_count)(w1)

Version2-+8 corners,edges+6 and rest all +2.

Version1(implemented)-a more sophisticated grid of values which can be seen in the code on line 23.

```
int weight[8][8]={ {8,1,6,6,6,6,1,8},
                   {1,0,3,3,3,3,0,1},
                   {6,3,5,4,4,5,3,6},
                   {6,3,4,5,5,4,3,6},
                   {6,3,4,5,5,4,3,6},
                   {6,3,5,4,4,5,3,6},
                   {1,0,3,3,3,3,0,1},
                   {8,1,6,6,6,6,1,8}
                   };
```

3. Actual Mobility(mobility):Number of valid moves available currently(w3)

The return values of all these function were scaled for proper weighing.

The following were the observations:

Ply number(depth)	Weight(w1,w2,w3)	Final score(CPU, BOT) Version2	Final score (CP U,B OT) Version1	Number of nodes generated	Number of nodes explored	Number of nodes pruned
6	(0,0,1)	26,38	26,38	219091	43471	21022
	(0,1,0)	25,39	25,39	234074	39375	12037
	(0,1,1)	49,15	49,15	209393	54770	11032
	(1,0,0)	38,26	42,22	227654	34345	22133

	(1,0,1)	21,43	24,40	228897	59378	22432
	(1,1,0)	34,30	36,28	227991	43441	21171
	(1,1,1)	45,19	49,15	229094	44370	22032
9	(0,0,1)	17,47	17,47	1225131	233120	101372
	(0,1,0)	33,31	33,31	1424440	213278	111391
	(0,1,1)	25,39	25,39	1324640	218721	101211
	(1,0,0)	43,21	46,18	1319745	223273	100171
	(1,0,1)	43,21	47,17	1121120	223277	114573
	(1,1,0)	30,34	31,33	1211711	211179	103402
	(1,1,1)	37,27	42,22	1124740	223278	110672
11	(0,0,1)	25,39	25,39	31240011	5383947	2616715
	(0,1,0)	33,31	33,31	32153517	4099951	2041110
	(0,1,1)	47,17	47,17	30013670	5093441	2546220
	(1,0,0)	44,20	48,16	31214129	4893940	2424120
	(1,0,1)	42,22	48,16	32351337	5270343	2630171
	(1,1,0)	26,38	30,34	33113144	5497312	2745511
	(1,1,1)	15,49	20,44	31253050	5493951	2746170

Here we see that the if the heuristics are equal weighted we get a somewhat better performance as compared to using only one heuristic at a time. Thus, equally weighing the heuristics gives the player a better move in terms of mobility, more score, more number of pieces on the board and occupying the corners.

Bonus: We observe that the version1 of weighted heuristic performs much better than version2 because in version 2 squares other than the corners and the edges are treated as same which is a poor estimate of the advantage of that position. For instance, version 1 of the heuristic tells us that the squares adjacent to the corner are of least advantage which is true in actual gameplay.