

ソフトウェア基礎特論 期末レポート

4 年 15 組 28 番 高野昂平

1 ゲームとプレイヤー

ゲームとしては、**オセロ**を選んだ。先手のプレイヤーを黒とし、モンテカルロ木探索を用いてその強さを評価する。また、後手はランダムの場合と、MCTS を実装した場合の 2 パターンを試した。

モンテカルロ木探索の子ノード選択の基準としては UCB1 を選択した。以下の式の w はそのノードにおける評価値、 n はそのノードの訪問回数、 N はそれぞれの子ノードの訪問回数の合計値、 C は定数で探索がうまくいくようなパラメータとなっている。第 2 項が訪問回数が少ないほど大きくなる項であり、訪問回数が少ないノードに重きを置く場合、定数 C の値を大きくすべきである。ただし、大きくしすぎると、第 1 項の評価値が軽視されるため、調整が必要となっている。今回の実装では、 $C = \sqrt{2}$ とした。

$$\frac{w}{n} + C \sqrt{\frac{\ln N}{n}}$$

2 実装と実験環境

実装にあたっては、言語を Python とし、MacOS 上での実験を行った。

3 勝率について

各条件のもと、20 戦の対戦を行い、その勝率を評価する。本来は 100 戦ほど対戦を行わせるべきではあるが、実行時間との兼ね合いを見て、20 戦とした。

MCTS プレイヤーにおけるパラメータは、1 手を導くのに何回の評価を行うのか (evaluate number) と、ノードを展開する際に何回訪問したノードを展開するか (expand base number)、の 2 つである。以下の表において、evaluate number を eval_num、expand base number を eb_num としている。

3.1 対ランダムプレイヤー

表 1 にランダムプレイヤーとの対戦の結果を示す。

表 1 対ランダムプレイヤーでの勝率

パラメータ		勝率		
eval_num	eb_num	勝ち	負け	引き分け
1	0	12	8	0
5	2	12	7	1
10	3	15	5	0
20	5	14	4	2
30	10	19	0	1
50	10	20	0	0
100	30	20	0	0
150	50	19	1	0
200	70	20	0	0

結果を見てわかるように、MCTS プレイヤーとランダムプレイヤーの対戦においては MCTS プレイヤーが勝ち越していることがわかる。ただし、評価回数 (eval_num) が 1 回であったり、5 回であったりするものに関しては、ほとんど MCTS による評価が行われておらず、この勝率が正しいものであるかは不明である。

その他の MCTS プレイヤーについては、特に、評価回数が 30 回を超えるものに関しては表を見てわかるように勝率が 9 割を超えていることから、ランダムプレイヤーに対して圧倒的な強さを持っていることがわかる。

3.2 対 MCTS プレイヤー

以下の表 2 に MCTS プレイヤー同士の対戦結果を示す。

表 2 対 MCTS プレイヤーでの勝率

パラメータ				勝率		
先手 (黒)		後手 (白)		勝ち	負け	引き分け
eval_num	eb_num	eval_num	eb_num			
100	30	100	50	13	7	0
200	50	100	50	14	5	1
200	50	300	50	9	11	0
200	70	1000	200	6	14	0

パラメータの異なるプレイヤー同士で戦わせることでどのパラメータが MCTS プレイヤーの強さに影響を与えるか知ることができる。そして、この表を見てわかるのが、まず、表の 1 行目を見ると、評価回数が同じ場合、eb_num の値が少ない方が強いということである。ただ、対戦回数が少ないことから、たまたま、eb_num の値が少ない方が強いということである。これは、この eb_num の値が大きすぎる場合、有望な手を深く読むことができず、正しく評価できないのではないかと思う。

次に、評価回数が多い方が強いということが表の 2 行目、3 行目、4 行目からわかる。やはり、評価回数の多い方がより正確に次の強い手を導くことができるということだと思う。