

minimax法によるリバーシの最善手の探索

茨城県立竹園高等学校 パソコン部 亀崎 裕登 安藤 悠太 比山 雅貴

1.リバーシとは

<概要>

ゲームのルールは「相手の石を自分の石で挟む」という基本原理に基づく単純明快なものだが、コンピュータが発達した2020年現在でも必勝法などの完全解析がなされていないほどの奥深さを持ち、人間が最善手や絶対に負けない方法を把握するのは、まず不可能だと考えられている。この事を端的に表した「覚えるのに一分、極めるのに一生」という言葉がキャッチフレーズになっている。リバーシは、相手の石を自分の石で挟めばそれも自分の石になるという点に最大の特徴がある。

<ルール>

黒番、白番の順で交互に盤面の空いているマスに自分の色の石を打っていく。この際、今打った石と他の自分の色の石とで縦・横・斜めのいずれかの方向で挟んだ相手の色の石は、裏返して自分の色に変える。また、挟めるマスが1つもない場合はパスとなり、相手の手番となる。終局時点で黒石・白石の数を数え、多いほうが勝ちとなる。同数の場合は、引き分けとなる。

参考文献 フリー百科事典『ウィキペディア (Wikipedia) 』

2.盤面評価について

良い手を見つけるアルゴリズムを開発するためには盤面を正確に評価するプログラムを作成する必要がある。

リバーシは10の58乗通りあり、完全読みができないため、盤面を本当の意味で正確に評価することは不可能

「角をとれば有利」というような一般的に言われている理論を用いて盤面をできる限り正確に(優勢であるか劣勢であるか)評価することを目指す。

- 1.評価関数
- 2.着手可能場所
- 3.確定石

1.評価関数による評価

石がある場所によって評価を決める手法

・角をとるとその石は返されることがないため、そこから自分の石を増やすことができ、有利

・中央をとると着手可能数が増えるため有利

上記のような理論をもとに右のような評価関数を作成する。それぞれの石の評価値の総和を求める。

例

<黒> 0+0+0+0+(-3) = -3

<白> 0+0+0+0+2+(-3) = -1

30	-10	2	1	1	2	-10	30
-10	-20	-3	-3	-3	-3	-20	-10
2	-3	2	0	0	2	-3	2
1	-3	0	0	0	0	-3	1
1	-3	0	0	0	0	-3	1
2	-3	2	0	0	2	-3	2
-10	-20	-3	-3	-3	-3	-20	-10
30	-10	2	1	1	2	-10	30

2.着手可能箇所による評価

自分(コンピューター)が次の番どこに置くことができるかで評価する。これについても評価関数を使用する。

リバーシでは一般的に打てる箇所が多い(選択肢が多い)と有利であると考えられている。

例<次が黒の番であった場合の黒の評価>

1マスにつき基礎点を20点(評価関数の最低点の絶対値)として計算する。
(20+0)+(20+0)+(20-3)+(20-3)+(20+1) = 95

30	-10	2	1	1	2	-10	30
-10	-20	-3	-3	-3	-3	-20	-10
2	-3	2	0	0	2	-3	2
1	-3	0	0	0	0	-3	1
1	-3	0	0	0	0	-3	1
2	-3	2	0	0	2	-3	2
-10	-20	-3	-3	-3	-3	-20	-10
30	-10	2	1	1	2	-10	30

3.確定石の数による評価

確定石(隅など相手に返されない石)の数で評価する。

確定石を増やすことは自分の石の数を確実に増やすことができ、勝利に近づけることができると言える。

ここでは確定石の数をそのまま評価値として扱う

例<黒>

赤く囲まれた部分が確定石(4つ)

30	-10	2	1	1	2	-10	30
-10	-20	-3	-3	-3	-3	-20	-10
2	-3	2	0	0	2	-3	2
1	-3	0	0	0	0	-3	1
1	-3	0	0	0	0	-3	1
2	-3	2	0	0	2	-3	2
-10	-20	-3	-3	-3	-3	-20	-10
30	-10	2	1	1	2	-10	30

評価の重みの設定

最後に1~3までの評価を考慮して最終的に盤面の評価を数値で表す。それぞれの重要性は異なると考えられるので「それぞれの値と重みの積」の和を求める。

例

自分の石の評価関数による評価	$-3 \times 10 = -30$	1025 (盤面の評価値)
相手の石の評価関数による評価	$-1 \times 15 = -15$	
着手可能箇所による評価	$95 \times 10 = 950$	
確定石の数による評価	$4 \times 30 = 120$	

最適な重みを見つけるために異なる重みを設定したアルゴリズム同士で対戦させ、勝率を計算した。基準の重みを決め、「基準の重みで計算をするアルゴリズム」と「基準とは1つ重みを変えてたアルゴリズム」を対戦させた。ただし、複数の局面に対応できる重みを見つけるため20回に1回の確率でランダムな手を打つようにした。

始めに自分の石の重みを10と定めて(追加する重みを変更するためこの値はどのように定めても問題がない)以下のように重みを追加していく。

相手の石の重み

自分の石	相手の石	基準との勝率	基準 自分の石:10,相手の石:0
10	5	66.66666667	
10	10	66.66666667	
10	15	70	
10	20	63.33333333	相手の石:15のときに勝率が最大になる。

着手可能箇所の重み

自分の石	相手の石	着手可能	基準との勝率	基準 自分の石:10,相手の石:15,着手可能箇所:0
10	15	5	50	
10	15	10	56.66666667	
10	15	15	56.66666667	着手可能箇所:10,15のときに勝率が最大になる。
10	15	20	40	今回は、ほかの評価方法への影響を抑えるため小さい10を採用した。

確定石の数の重み

自分の石	相手の石	着手可能	確定石	基準との勝率	基準 自分の石:10,相手の石:15,着手可能箇所:15,確定石:0
10	15	10	10	50	
10	15	10	20	46.66666667	
10	15	10	30	63.33333333	
10	15	10	40	60	
10	15	10	50	33.33333333	確定石:30のときに勝率が最大になる。

自分の石:10,相手の石:15,着手可能箇所:10,確定石:30のときに勝率が最大になる。

3.MiniMax法による探索

盤面の評価を可能にしたのち、場合分けを行い数手先の内容を樹形図で表していき、その中から最もよい手を探索する。MiniMax法はその1つの手法である。

MiniMax法では人間は最善手(コンピューターにとって不利な手)を打って来ると仮定して探索を行う。

COMが打つことができる場所をすべてリストアップする

それぞれに対して...

その場所にうち、次の番に人間が打てる場所をすべてリストアップする

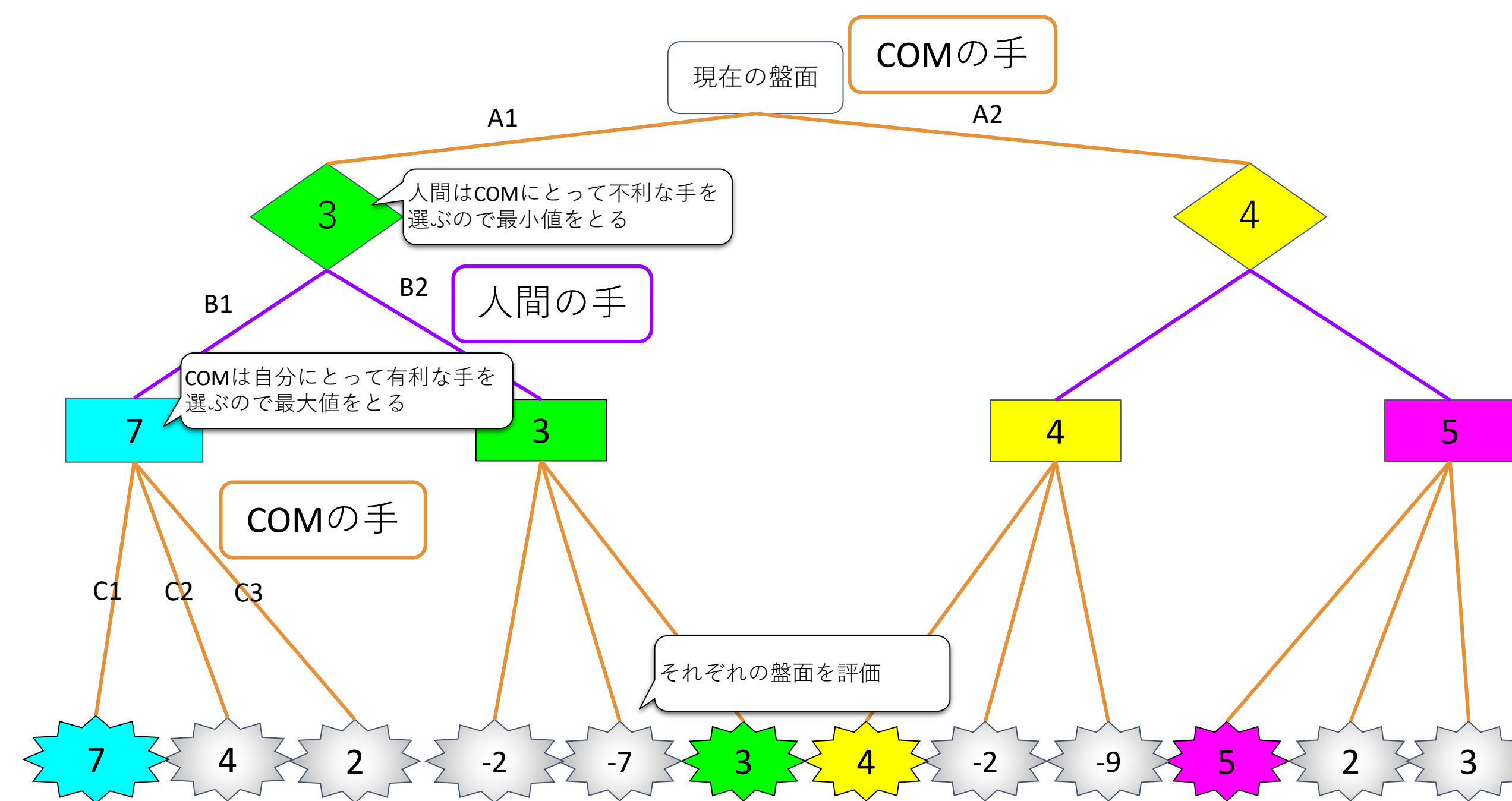
それぞれに対して...

その場所にうち、次の番にCOMが打てる場所をすべてリストアップする

上記の処理をデバイスの処理能力に合わせて複数回行う(先読みをする)。今回はスマートフォンでスムーズな動作をさせるため4回とした。

最終的に算出されたすべての盤面を評価する。

<MiniMax法の計算の例>



現在の盤面ではCOMは「A1」と「A2」という手を打つことができる。

「A1」打つと、次の番で人間は「B1」と「B2」という手を打つことができる。(A2の場合は略)

「B1」と打つと次の番でCOMは「C1」と「C2」と「C3」という手を打つことができる。(B2,B3の場合は略)

「C1」、「C2」、「C3」と打つとそれぞれ7,4,2という評価になる。COMは自分に有利な手を打つからCOMはC1を選ぶと考えられる。よって「B1」と人間が打った場合の評価は7ということになる。同様にして「B2」と人間が打った場合の評価は3ということになる。

人間はCOMに不利な手を打つから「A1」と打った場合の評価は3となる。同様にして「A2」と打った場合の評価は4となる。

よってCOMは「A2」の手を打つ。

4.結果・考察

制作したアルゴリズムと対戦し、人間には不可能な速度で数手先まで局面を予想できているということが実感できた。しかし、リバーシの対戦経験が多い人に勝つまでには至らなかった。これは角を取らせないことよりも着手可能箇所を増やすことを優先してしまうなど重みづけに問題があることがわかった。しかし、重みはコンピューター同士で対戦させ最適化するため、理論上は最善だと考えていた。その後、重みを手動で調整することで、このような問題が発生したのはコンピューターと人間の打つ手の傾向が異なるためだったということがわかった。今後はこの問題を解決するため、強化学習などを用いて人間が打った手を学習させて幅広い傾向の手に対応できるアルゴリズムを作成していきたい。