

二人不完全情報ゲーム『ゲイスター』におけるCFRおよび 利得評価関数を用いた混合戦略AIの研究

川上 直人^{1,a)} 池田 心^{1,b)}

概要: 近年、囲碁や将棋などの完全情報ゲームだけでなく、麻雀、ポーカーといった不完全情報ゲームにおいて、トッププレイヤーの実力を上回るAIが報告されている。Head's up limit pokerでは2015年にナッシュ均衡に近い解が得られた。一方、駒移動型の二人零和不完全情報ゲーム「ゲイスター」では、3×2盤・短手数の場合を除き、プログラムが人間の實力を上回るなどの成果が報告されていない。特に、GAT2020杯ゲイスターAI大会で優勝した「紫駒AI」は、初期配置以外に確率的な行動が無く、戦略を知っている相手には弱い。例えば、紫駒AIは青駒をめったに相手駒に隣接させず、赤駒をよく隣接させる特徴がある。そのため、隣接させてきた場合には赤駒だろうという推測が容易にできてしまう。そこで本稿では4×4盤ゲイスターにおいて、確率的な行動を取らせるためにCFRを用いる。ただし、CFRの計算量を抑えるために数手で木の展開をやめ、未決着なリーフノードにおいて利得評価をおこなう関数を設計する。本稿では、紫駒AIとそれに付け込むAIを含む4種類のベンチマークプレイヤーとの対戦実験により、提案手法によるAI2種類の性能評価をおこなった。結果、ベンチマーク同士の対戦では、どのAIについても勝率15%を下回る苦手な相手が存在した一方で、提案手法では最も苦手な対戦相手に対して勝率24%, 19%を挙げた。

A study on mixed strategy AI using CFR and gain evaluation functions in the two-player imperfect information game "Geister"

NAOTO KAWAKAMI^{1,a)} KOKOLO IKEDA^{1,b)}

1. はじめに

近年、囲碁や将棋など完全情報ゲームの人工知能(AI)は、トッププレイヤーの実力を上回るなど大きな成果を挙げている。2015年には、モンテカルロ木探索と深層学習を組み合わせた囲碁AI『AlphaGo[1]』が、プロ囲碁棋士にハンディキャップなしで初めて勝利し、世間から注目を集めた。

完全情報ゲームが大きな成果を挙げている一方、ポーカーや麻雀といった不完全情報ゲームのAIも注目され、大きな進歩を遂げている。不完全情報ゲームは、ゲームの状態を完全には観測できないため、ゲーム木探索をおこなうのが困難である。アプローチの1つは、2008年に提案

され、一部のゲームで大きな成果を挙げた Counterfactual Regret minimization (CFR)[2]である。有名な例として、2015年には、CFRの性能をさらに高めた CFR+[3]により、ヘッズアップリミットホールデムポーカーにおいてナッシュ均衡に近い戦略が計算された[4]。もう1つのアプローチとして、深層強化学習が挙げられる。2019年には深層学習を用いた麻雀AI『Suphx[5]』が天鳳十段を達成した。一方、『ゲイスター』などまだ十分強いコンピュータプレイヤーが報告されていないゲームもある。

ゲイスター[6]は青駒、赤駒をチェスのように取り合う、二人零和確定不完全情報ゲームである。このゲームの面白い点は、まだ取っていない相手駒の色を観測できない点であり、相手駒の色をそれまでの動きから推測したり、逆に推測されにくい動きをすることが重要である。ゲイスターAIの研究は発展途上である。例えば紫駒AI[9]と呼ばれる

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

a) keimahatokusyu@gmail.com

b) kokolo@jaist.ac.jp

手法は、2020年のAI大会で優勝しているが、駒の動きに法則性があり、行動から状態を推定されやすい。具体的には、相手駒に隣接させる駒はほとんど赤駒であり、これを知られてしまうと簡単に負けてしまう。また、短い手数制限を加えた3×2盤のガイスターでは、確率的な行動戦略が分析されているが[13]、終局まで展開したゲーム木を考えているため、計算量が大きい。

本稿では、4×4盤のガイスターにおいて、ガイスターを表現するゲーム木を数手先までしか展開しないことで、現実的な時間で確率的行動を取らせる方法を提案する。本稿では、ゲーム木のリーフノードに対して期待利得を推測するヒューリスティックな評価値を与える。評価値の与え方には様々なバリエーションが考えられるが、本稿では2つの方法を試みる。まず、未決着なリーフノードを引き分けと評価する方法を試す。次に、完全情報なガイスターについて真の勝敗を計算し、リーフノードが持つ完全情報盤面を評価する方法を試みる。なお、本内容は第一著者の修士論文研究[15]の一部である。

2章ではガイスターのルールについて述べる。3章ではガイスターの先行研究を紹介する。4章では先行研究の課題について述べ、5章では提案手法を紹介する。6章では提案手法の事前準備として完全情報局面の勝敗計算をおこなう。7章では対戦実験をおこない性能などを評価する。8章では本稿の総括をおこない今後の方針について述べる。

2. ガイスターについて

ガイスター (Geister) [6] は、2種の駒を用いて遊ぶ二人零和確定不完全情報ゲームである。ドイツのゲームデザイナーであるアレックス・ランドルフ氏によって考案され、1982年、ドイツ年間ゲーム大賞にノミネートされた。ガイスターの大きな特徴として、まだ取っていない相手駒の色を観測できないことが挙げられる。相手駒の色は、それまでの動きなどから推測する必要がある。

ガイスターでは、6×6のマスからなる36マスの盤面を用いる。最初、各プレイヤーは青駒、赤駒を4個ずつ、自陣8マスに自由な組み合わせで初期配置する。初期配置をおこなった盤面例を図1に示す。本稿では、マスの座標をa-f, 1-6の組み合わせで表現する。先手から見て、左上をa1, 右上をf1, 左下をa6, 右下をf6とする。先手自陣はb5, e6を対角線とする長方形領域、後手自陣はb1, e2を対角線とする長方形領域である。

ゲームは交互着手により進行する。手番プレイヤーは自分の駒を1つ選び、上下左右4マスのいずれかに動かす。自駒があるマスには移動できないが、移動先に相手駒があればそれを取り、色を観測できる。図1の矢印が書かれたマスはゴールを表しており、先手はa1, f1, 後手はa6, f6をゴールとする。ゴールに自分の青駒が置かれている場合、次のターンにその駒を盤外へ脱出させることができ、勝利

先手脱出口

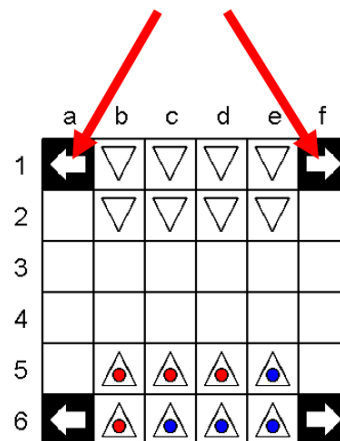


図1 ガイスター初期配置例

となる。なお赤駒は盤外へ脱出できない。

ガイスターでは次の3条件のいずれかを満たしたプレイヤーが勝利となる。

- 相手の青駒を全て取る
- 自分の赤駒を全て取らせる
- 自分の青駒を1つでも脱出させる（ゴールに移動してから、もう1手番必要）

3. ガイスターの先行研究

ガイスターのAIプレイヤーについて、完全情報ゲームの手法を応用した研究が報告されている。三塩・小谷は、モンテカルロ法を不完全情報ゲームに応用し、さらにプレイアウト回数の改善をおこなう手法として Using Past Playout (UPP) を提案した [7]。結果、改善前のモンテカルロ法に対し、勝率55%を達成した。佐藤は、強化学習を用い、さらに短い必勝手を見落さないよう必勝手の検出をおこなった [8]。具体的には、相手駒を「取ったら赤、脱出時は青に変化する駒（紫駒）」とみなすことで、最悪の状況を想定できるようにした。結果、モンテカルロプレイヤーに勝ち越し、ランダムAIには勝率約70%を達成した。川上・橋本は、佐藤の研究 [8] で用いられた「紫駒」をゲーム木探索に応用し、「紫駒AI」を提案した [9]。紫駒AIは、GPW2017杯ガイスター大会優勝AIに圧勝し、GPW2018杯ガイスターAI大会では準優勝、GAT2020杯ガイスターAI大会では優勝した。

確率的な行動戦略を立てるアプローチとしては、CFR[2]をベースとした研究が報告されている。Chen Chen, Tomoyuki Kaneko は、Chance-sampled CFR[11] および Pure CFR[12] を3×2盤のガイスターに適用し、戦略の分析、計算時間の比較をおこなった [13]。文献 [13] では、2~20手の手数制限を加えた3×2盤のガイスターについて、ナッ

シュ均衡を近似的に求めた。しかし、ゲーム木を全て展開しているため計算量が大きく、より大きなガイスターを扱うことは難しい。そこで同著者は、深層学習の技術をCFRに取り入れた手法 Deep CFR により、6×6 盤ガイスターの AI を開発した [14]。結果、ランダムプレイヤに勝率 70%~85% と学習には成功したものの、実験時間内では十分に学習が進まなかった。具体的には川上・橋本の研究で開発された深さ 1 の PurpleAI に対して、履歴を考慮しない AI プレイヤでは勝率 38%、履歴を考慮する AI プレイヤでは勝率 34% となった。

4. 先行研究の課題

2020 年のガイスター AI 大会で優勝した紫駒 AI [9] は、駒の移動に法則性があり、それを知った相手には簡単に負けてしまう課題がある。紫駒 AI に限らず、確定的な行動をおこなうプレイヤは、どのような局面でどのような行動を取るかを対戦相手に完全に熟知された場合、行動から状態を絞り込まれるなど、付け込まれる余地が生まれやすい。

本研究では、確率的行動を取ることで、付け入る余地を減らすことを試みる。ガイスターは二人零和不完全情報ゲームであるため、最も苦手な相手に対する勝率を最大化する戦略は、ナッシュ均衡に従う戦略となる。そのため、究極的にはナッシュ均衡を求めることがゴールとなる。

ガイスターはゲーム木によって表現することができるため、ゲーム木を生成し、CFR [2] などを適用することでナッシュ均衡を求めることができる。ガイスターを表現するゲーム木を概念を図 2 に示す。図 2 は、先手初期配置、後手初期配置、先手駒移動、後手駒移動…の方法を木構造で表したものであり、各ノードには状態が書かれている。特に、深さ 2 以降のノードは完全情報な局面を持つ。完全情報ゲームと大きく異なる点は、情報集合と呼ばれる、手番プレイヤーにとって区別できないノードのグループが存在する点である。ガイスターにおける情報集合は、まだ取っていない相手駒の色組み合わせのみが異なるノード同士であり、同じ情報集合に属するノードでは同じ行動戦略を立てる必要がある。例えば、図 2 の緑破線は情報集合の一例である。結ばれた 2 局面は先手番にとって互いに区別できないため、異なる行動戦略を立てることができない。

先行研究では、短い手数制限を加えた 3×2 盤のガイスターについてナッシュ均衡を求めた [13]。しかし、終局まで展開したゲーム木を扱うため計算量が大きく、それ以上大きなサイズのガイスターについては現実的な計算量で行動を求めることができない。

5. 提案手法

5.1 アルゴリズムの概要

本研究では、ゲーム木展開における計算量の問題を解決するため、ゲーム木を数手先までしか展開せず、リーフ

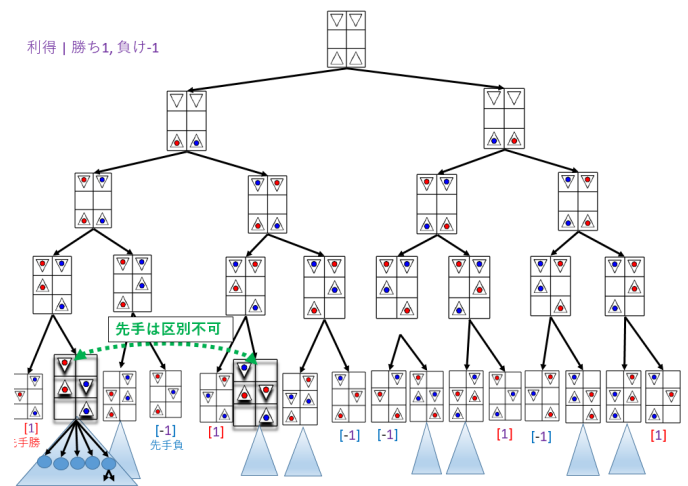


図 2 素朴なゲーム木 (終局まで展開)

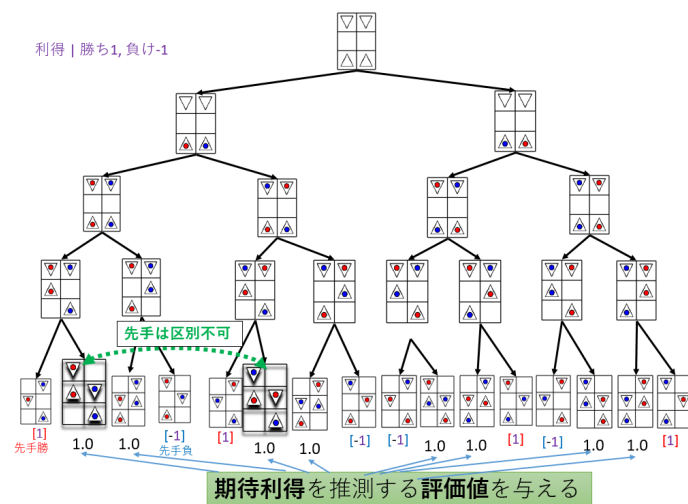


図 3 評価値を与えたゲーム木 (深さ 4 まで展開する例)

ノードの良し悪し (期待利得) を推測する評価値を与える方法を提案する。このような限定的なゲーム木を生成することで、現実的な時間で、現局面に対する確率的な行動を得ることが期待できる。図 3 に、提案手法の概念を示す。図 2 と比べると、3 手先以降 (深さ 5 以上) のノード展開をおこなわず、深さ 4 のノードに評価値を与えているところが異なる。図 3 は、完全情報なガイスターの勝敗を評価値とした例であり、どのリーフノードが持つ盤面も先手が後手青駒を取り勝利することから勝率 1.0 と推測している。

具体的な手順を以下に示す。ただし、駒の初期配置につ

いては単純なランダムでおこなう。リーフノードへの具体的な評価値の与え方は次節で述べる。

- (1) 現局面（過去の着手を含まない）について、先手の駒色、後手の駒色をそれぞれ隠した局面 S を生成する。
- (2) S をルートノードとし、ある深さまで展開したゲーム木を生成する。自分は深さ偶数手番とする。
- (3) リーフノードについて、評価値を与え、これを利得の代わりとする。
- (4) 生成したゲーム木において、CFR をおこない、混合戦略（各ノードに対する確率的行動）を計算する。
- (5) 現局面と対応するノードを参照し、確率的なアクションを取る。現局面に対応するノードとしては、深さ2のノード（初期配置が終わった直後の局面を持つ）であって、現局面と自分の色配置が一致するものを任意に取る。

5.2 リーフノードの利得評価関数

未決着なリーフノードについて、どのくらいの利得を期待できるかを評価するため、利得評価関数を設計する。最も素朴な評価は、未終局なら引き分け扱いとする方法だが、ゲーム序盤ではリーフノードの多くが未決着となるため、行動の良し悪しを評価することができない。実際には、リーフノードは完全情報な局面を持つので、その局面を評価することで、単に引き分けを返すよりも高い精度でリーフノードの良し悪しを評価できるのではないかと考えられる。そこで本稿では、各リーフノードについて、そのリーフノードが持つ完全情報局面の評価をおこなう。評価には様々な方法が考えられるが、本研究では、リーフノードが持つ完全情報局面から完全情報なガイスターをプレイした際の真の勝敗を求め、その勝敗を評価値として用いる。

本稿では、後退解析を用いて、互いに勝ちを目指して最善を尽くした場合の勝敗分を記録したデータベースを生成する。後退解析は、勝敗が決定した局面から手を戻していき、戻した局面の勝敗を決定していく手法である。相手に負け局面を渡すことができる局面は勝ち、どのように手を進めても相手に勝ち局面を渡してしまう局面は負けとなり、最後まで勝敗が確定しなかった局面は引き分けとなる。これは、相手の勝ちを阻止できる一方で相手に負け局面を渡すこともできないためである。

6. 完全情報局面の勝敗計算

本稿の利得評価関数を用いるための事前準備として、完全情報なガイスターにおける真の勝敗を計算した。具体的には、 4×4 盤、各プレイヤーの青赤が2個以下の完全情報局面を対象とし、後退解析をおこなった。実行環境として、Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz のシングルスレッドを用いたところ、実行時間は311.9秒、メモリ使用量は8124.4[MB]となった。ただし、左右反転で一致す

表 1 勝敗と局面数（先手目線）

全局面数	71,001,840
勝ち局面数	49,821,729
負け局面数	17,051,259
引き分け局面数	4,128,852

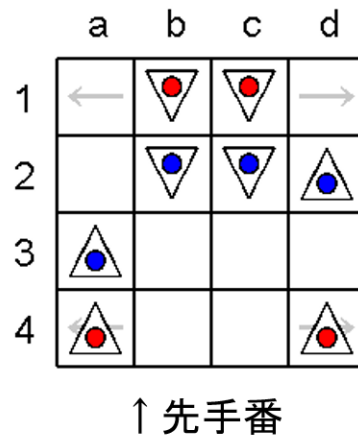


図 4 最長勝ち局面 (47 手)

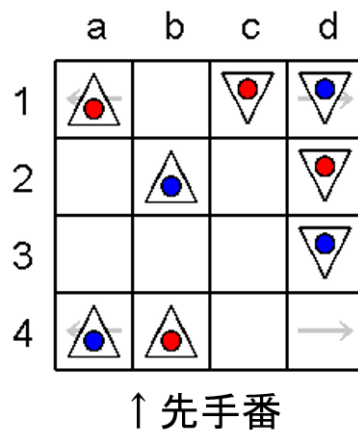


図 5 最長負け局面 (48 手)

る局面のうち片方のみを保存するなど、空間計算量について基本的な改善をおこなっている。

勝敗数の結果を表 1 に示す。表 1 に示すように、71,001,840 局面のうち、先手勝ちには 49,821,729 局面、先手負けは 17,051,259 局面、引き分けは 4,128,852 局面となった。また、最長勝ち局面は図 4 の局面となり 47 手勝ち、最長負け局面は図 5 の局面となり 48 手負けとなった。さらに、勝ち局面数と手数との関係は表 2、負け局面数と手数との関係は表 3 のようになり、手数が大きくなるごとに局面数が少なくなることを確認した。

7. 対戦実験

提案手法の性能を評価するため、ベンチマークプレイヤー

表 2 手数と勝ち局面数

1	21,405,186
3	13,300,150
5	4,188,579
7	2,643,147
9	1,993,674
11	1,661,412
13	1,312,694
15	1,020,795
17	762,849
19	548,515
21	386,897
23	258,268
25	153,386
27	88,879
29	50,575
31	26,115
33	12,319
35	5,096
37	2,085
39	736
41	264
43	80
45	24
47	4

表 3 手数と負け局面数

2	7,514,602
4	3,239,989
6	1,690,200
8	957,902
10	846,053
12	691,214
14	581,617
16	465,892
18	351,644
20	255,988
22	177,502
24	114,142
26	71,675
28	44,969
30	25,196
32	12,968
34	5,893
36	2,458
38	920
40	286
42	108
44	28
46	11
48	2

を用意し対戦実験をおこなう。対戦実験では、対戦相手が敵対的に戦略を選ぶ状況における付け込まれにくさの指標として、最も苦手な対戦相手に対する勝率（最小勝率）に焦点を当てる。また、推定のされやすさを分析するため、駒の移動に関する統計量を取る。さらに、勝敗の内訳などを調べ、どのような方法で勝利・敗北しているかを考察する。

7.1 実験方法

4×4 盤のガイスターについて対戦実験をおこない、勝敗などを調査する。まずベンチマークプレイヤー同士の総当たり戦をおこない、勝敗数などを分析する。その後、提案手法とベンチマークプレイヤーを戦わせ、最も苦手な対戦相手に対する勝率（最小勝率）の分析をおこなう。各組み合わせでは、先後手がそれぞれ 50 回ずつ現れるようにし、100 試合おこなう。初期配置は手前の一段（先手は a4-d4、後手は a1-d1）におこない、どのプログラムもランダムに選択する。なお、無限にゲームが終わらないのを防ぐため、200 手で決着がつかない場合は引き分けとする。ベンチマークプレイヤーとしては『ランダム』『猪突猛進』『紫駒 AI』『アンチ紫』を実装する。それぞれ以下の内容である。

- ランダム
 - 合法手のなかからランダムに選択し、行動する
 - ただし、1 手で脱出可能なときは脱出をおこなう
 - 攻めも守りも貧弱なため、弱い

● 猪突猛進

- 1 つの青駒をひたすらゴールへ近づける。取られたら別の青駒で同じことをする。
- 具体的には近い方のゴールまでのマンハッタン距離が最小の青駒を選ぶ。複数あれば、最も上段、それでも複数あれば、最も左列にある駒を選ぶ。
- ゴールへなるべく近づけるように移動する。移動方向が複数考えられるときは、上方向にゴールがあるとして、左、右、上、下の順で優先し、移動方向を選ぶ。ただし本実験の設定では、下が選ばれる状況は発生しない。
- もし、選んだ青駒がゴールにある場合は、その駒を脱出させる。
- 基本的には、近づけた駒を取らない相手に強く、近づけた駒を積極的に取る相手に弱い。

● 紫駒 AI

- 相手駒を「取ったら赤、脱出時は青に変化する駒（紫駒）」とみなし、深さ 5 の MinMax 法をおこなう。
- 評価関数は文献 [9] と同様、「青の個数」「駒がゴールからどれだけ離れているか」の 2 つから作る。
- 詰みを検出できるなどの強みがあり、GPW2018 杯準優勝 AI、GAT2020 杯優勝 AI にもこの手法が用いられた。

- 一方、赤ばかり相手駒に隣接させるなど、色を簡単に推定できるような動き方をする。

● アンチ紫

- 紫駒 AI の弱点を突くプレイヤー。
- 紫駒 AI が最初に突っ込ませる駒はほとんど赤だろうという仮説を基に設計する。
- 自駒に初めて隣接した相手駒を赤とみなし、ゲーム終了まで絶対取らないようにする。それ以外の駒は、取っても良い駒（青）とみなす。
- 自駒に隣接してくるまでは、紫駒 AI と同様の意思決定をおこなう。
- 深さ 5 の MinMax 法により完全情報なガイスターを探索し、手を決定する。
- 評価関数には、「青駒の個数差」「駒がゴールからどれだけ離れているか」の 2 つを用いる。相手の青駒数を考慮すること以外は紫駒 AI と同様である。自分の青駒を保持したい、相手の青駒（とみなした駒）を取りたい、ゴールへ駒を近づけたいという観点から設計する。
- 基本的には、赤ばかり近づける相手に強く、青ばかり近づける相手に弱い。

また、提案手法におけるプログラム名と利得評価関数の対応を次のようにする。

- CFR-A、未終局のリーフノードを引き分けと評価する。
- CFR-B、リーフノードが持つ完全情報局面について真

の勝敗を求め、評価値とする。

提案手法において、ゲーム木の展開深さは5とする。深さ0, 1では色配置の列挙をおこなっているため、駒移動に関する木の展開は3手までおこなうことになる。CFRのイテレーション数は1000とする。

7.2 実験結果

ベンチマークプレイヤーとの対戦結果を表4に示す。表4の最小勝率は、最も苦手な対戦相手に対する勝率を表している。

まず、4つのベンチマークプレイヤーの最小勝率に着目すると、いずれについても最小勝率15%を下回る苦手な相手が存在することが分かる。特に表4の紫駒AIは、ランダム相手に勝率90%、猪突猛進には勝率100%と大きく勝ち越した一方で、アンチ紫には勝率13.5%と大きく負け越した。紫駒AIの勝敗内訳は表5のようになり、アンチ紫に対しては6割の試合で青を全て取られ負けている。紫駒AIは相手駒に隣接させる駒がほとんど赤という特徴があるため、アンチ紫には多くの試合で赤を見破られ、見破られた駒以外を積極的に取られてしまったのではないかと考える。また、アンチ紫は猪突猛進に大敗し、猪突猛進は紫駒AIに大敗している。

一方、表4において、CFR-A, CFR-Bのベンチマークプレイヤーに対する最小勝率に着目すると、CFR-Aでは最小勝率が24%、CFR-Bでは最小勝率が19%となり、どのベンチマークプレイヤーよりも最小勝率が大きいことが確認できた。

また、駒移動に関する統計量を紫駒AI, CFR-A, CFR-Bについて計測した。まず、相手駒と隣接させる移動をおこなった回数を青、赤ごとに計測した。ただし、相手駒を取る場合は計測の対象外とした。結果、1試合あたりの隣接回数は表6のようになった。表6より、紫駒AIは赤ばかり隣接させている一方で、提案手法のCFR-A, CFR-Bは紫駒AIと比べて青もよく隣接させていることを確認できた。次に、相手駒が自駒に隣接した直後、その駒を取った割合を計測した。結果、表7のようになった。相手赤1個のとき、紫駒AIは隣接してきた駒を絶対に取らないのに対し、CFR-A, CFR-Bではたまたま駒を取っていることを確認できた。よって、相手の赤が残り1個の場合において、簡単に脱出されにくくなったと考える。

最小勝率、駒移動に関する統計量の結果から、提案手法のCFR-A, CFR-Bは紫駒AIと比べると対戦相手に付け込まれにくくなったと考えられる。しかし残念ながら、提案手法は全てのベンチマークプレイヤーに勝ち越すことはできなかった。その敗因を探るため、勝敗条件の内訳を取り、敗北条件を分析した。勝敗条件の内訳を表8, 9に示す。まず、CFR-A, CFR-Bは紫駒AIに負け越しているが、いずれも赤を全て取ってしまい敗北した試合が半分以上を占め

表4 ベンチマークプレイヤーとの対戦結果 (勝-敗-分)

横軸が相手	ランダム	猪突猛進	紫駒 AI	アンチ紫	最小勝率 [%]
ランダム		9-91-0	10-90-0	30-70-0	9
猪突猛進	91-9-0		0-100-0	92-8-0	0
紫駒 AI	90-10-0	100-0-0		12-85-3	13.5
アンチ紫	70-30-0	8-92-0	85-12-3		8
CFR-A	78-22-0	36-64-0	24-76-0	43-57-0	24
CFR-B	78-22-0	71-29-0	20-80-0	19-81-0	19

表5 紫駒AIの勝敗内訳

	ランダム	猪突猛進	アンチ紫
青取り勝ち	11	67	0
赤取り負け	0	0	0
赤取らせ勝ち	58	33	2
青取られ負け	0	0	60
脱出勝ち	21	0	10
脱出負け	10	0	25

表6 1試合あたりの隣接回数

	赤	青
紫駒 AI	3.61	0.23
CFR-A	2.06	1.37
CFR-B	2.35	0.69

ていることが分かった。これは、相手の赤が2駒から1駒に減るときのリスクを評価関数内で十分評価できていないことが理由ではないかと考える。また、CFR-Bはアンチ紫にも弱い。表9について、アンチ紫に対する「赤取られ勝ち」「青取られ負け」の数を比較すると、CFR-Bは赤取られ勝ちが8%、青取られ負けの方が32%と、青取られ負けも目立っていた。よって、青駒を取られるリスクについても過小評価していると考えられる。

提案手法のCFR-Bは、完全情報局面の評価をおこなったにもかかわらず、CFR-Aと比べ、最小勝率が低く、紫駒AI, アンチ紫どちらにも付け込まれていることが分かる。これは、リーフノードが持つ局面を別々に評価することで、本来区別できないはずの局面で異なる手順を指せると仮定してしまい、適切な評価値をつけられないことが原因と考えられる。例えば、図6において、2つの末端局面は先手番にとって互いに区別できないため、本来であれば異なる手順を指すことができず、(即座に)勝ちを確信できないはずの局面である。しかし、それぞれの局面を独立に考えると、それぞれ後手の青を取ることで先手必勝となる。よって、CFR-Bの評価関数ではどちらも先手必勝と判断してしまうことになる。これは過大評価となっており、このような適切でない評価が頻発することで、CFR-Bは上手くいかなかったと考えている。

8. おわりに

本稿では、二人零和不完全情報ゲーム『ガイスター』において、対戦相手に付け込まれる余地を減らすため、確率的

表 7 隣接してきた駒を取った割合

	全体	相手赤 1 個
紫駒 AI	0.20	0
CFR-A	0.21	0.11
CFR-B	0.33	0.20

表 8 CFR-A の勝敗内訳

	紫駒 AI	アンチ紫
青取り勝ち	4	0
赤取り負け	56	13
赤取らせ勝ち	0	36
青取られ負け	3	31
脱出勝ち	20	7
脱出負け	17	13

表 9 CFR-B の勝敗内訳

	紫駒 AI	アンチ紫
青取り勝ち	2	0
赤取り負け	64	41
赤取らせ勝ち	0	8
青取られ負け	3	32
脱出勝ち	18	11
脱出負け	13	8

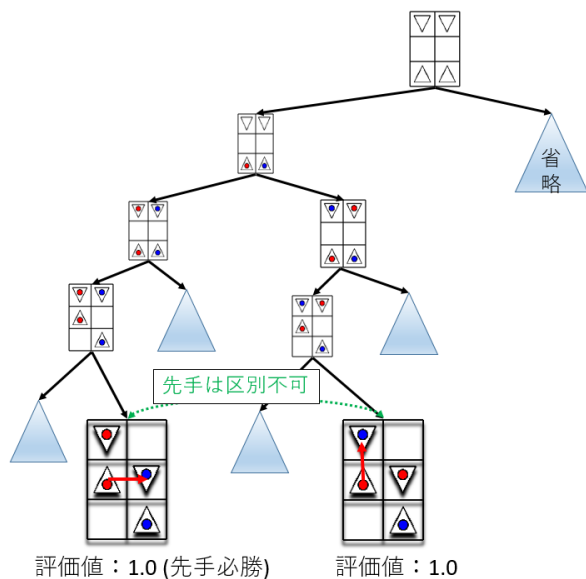


図 6 CFR-B が上手いかわかないケース

行動を取る AI を開発した。現実的な時間で確率的行動を計算するため、数手先までしかゲーム木を展開せず、リーフノードに評価値を与える手法を提案した。4 × 4 盤のガイスターについて、4 種類のベンチマーク AI との対戦実験をおこなった結果、ベンチマーク AI 同士の対戦ではどの AI についても勝率 15% を下回る苦手な相手が存在した一方で、提案手法 2 種とベンチマーク AI の対戦では、最も苦手な相手に対しても 19%, 24% の勝率となった。残念な

がら全てのベンチマーク AI に勝ち越すことはできなかったが、確率的な着手によって付け込まれる余地を減らすことには成功したと考えられる。

今後の方針として、過去の着手を考慮することが挙げられる。本研究の手法においては、過去の着手を用いず、現局面のみを入力としていたが、過去の着手を用いることでさらなる性能の向上が期待できる。例えば、脱出しなかった駒をそれ以降のターン赤とみなすことができれば、試合を有利に運ぶことができる。今後の方針として、過去の着手を用いた手法を考えたい。

参考文献

- [1] Silver, D. et al.: *Mastering the Game of Go without Human Knowledge*, Nature 550, pp.354-359 (2017).
- [2] Zinkevich, M., Johanson, M., Bowling, M. and Piccione, C. *Regret Minimization in Games with Incomplete Information*, Advances in Neural Information Processing Systems 20, pp.1729-1736 (2008).
- [3] Tammelin, O.: *Solving Large Imperfect Information Games Using CFR+*, arXiv:1407.5042 (2014).
- [4] Bowling, M. et al.: *Heads-up limit hold'em poker is solved*, Science Vol. 347, Issue 6218, pp.145-149 (2015).
- [5] Li, J. et al.: *Suphx: Mastering Mahjong with Deep Reinforcement Learning*, Artificial Intelligence, arXiv:2003.13590v2, pp.1-28 (2020).
- [6] ガイスター . [<http://www.mobius-games.co.jp/Gester.htm>]. (アクセス：2021/02/09).
- [7] 三塩 武徳, 小谷 善行: ゲームの不完全情報推定アルゴリズム UPP とそのガイスターへの応用, 情報処理学会研究報告, Vol.2014-GI-31, No.4, pp.1-6 (2014).
- [8] 佐藤 佑史: ガイスターにおける自己対戦による行動価値関数の学習, 電気通信大学学術機関リポジトリ (2015).
- [9] 川上 直人, 橋本 剛: 完全情報ゲームの探索を用いたガイスター AI の研究, 第 23 回ゲームプログラミングワークショップ, pp.35-42 (2018).
- [10] 木村 勇太, 伊藤 毅志: 深層強化学習を用いたガイスター AI の構築, 第 24 回ゲームプログラミングワークショップ, pp.130-135 (2019).
- [11] Neller, T. W. and Lanctot, M. *An Introduction to Counterfactual Regret Minimization*, Proceedings of Model AI Assignments, The Fourth Symposium on Educational Advances in Artificial Intelligence (EAAI 2013) (2013).
- [12] Gibson, R.: *Regret Minimization in Games and the Development of Champion Multiplayer Computer Poker-Playing Agents*, PhD Thesis, University of Alberta (2014).
- [13] Chen Chen and Tomoyuki Kaneko. *Counterfactual Regret Minimization for the Board Game Geister*, 第 23 回ゲームプログラミングワークショップ, pp.137-144 (2018).
- [14] Chen Chen and Tomoyuki Kaneko. *Utilizing History Information in Acquiring Strategies for Board Game Geister by Deep Counterfactual Regret Minimization*, 第 24 回ゲームプログラミングワークショップ, pp.20-27 (2019).
- [15] 川上 直人: 完全情報ゲームの評価値を用いた二人零和不完全情報ゲーム『ガイスター』における混合戦略 AI の研究, JAIST 学術研究成果リポジトリに公開予定