

『紫駒』を用いたMinMax探索によるガイスターAIの研究

川上 直人^{1,a)} 橋本 剛^{2,b)}

受付日 2020年12月30日, 採録日 2021年7月7日

概要: 近年, 囲碁や将棋などの完全情報ゲームだけでなく, 不完全情報ゲームの研究がさかんに行われている。麻雀, ポーカーといったカード型ゲームでは人間を超える AI が実現している。一方, 駒移動型不完全情報ゲームではまだ十分強い AI が報告されていない。本研究では, 駒移動型不完全情報ゲームのテストベッドとして注目されている「ガイスター」を題材とし, 終盤が強い AI に焦点を当て, まずは MinMax 探索を用いた強い AI プレイヤ開発を目指す。ベースライン手法とした「色配置全列挙」には必勝手順を正しく選択できない場合がある。そこで, AND/OR 木探索に用いられた『紫駒』のアイデアを導入する手法「紫駒 AI」を提案し, MinMax 探索に実装する。紫駒 AI を実装したプログラムは, 対戦実験の全プログラム (ランダム, ルールベース, 色配置全列挙) に勝ち越した。紫駒 AI は, 負けにつながる危険な手を避け, 確実な勝ちを狙う。また, 紫駒 AI は GAT2020 ガイスター AI 大会で 8 チーム中 1 位を獲得した。

キーワード: 不完全情報ゲーム, ゲーム木探索

Research of Geister AI by MinMax Method Using the Idea of “Purple Piece”

NAOTO KAWAKAMI^{1,a)} TSUYOSHI HASHIMOTO^{2,b)}

Received: December 30, 2020, Accepted: July 7, 2021

Abstract: In recent years, there has been a lot of research on imperfect information games as well as perfect information games such as Go and Shogi. In card-based games such as mahjong and poker, AIs that can outperform humans have been achieved. On the other hand, strong AI has not yet been reported for piece-moving incomplete information games. In this study, we focus on AI that is strong in endgames, using “Geister”, which has attracted attention as a testbed for piece-moving imperfect information games, and first aim to develop a strong AI player using MinMax search. We employ an “Enumerating All Color Placements” method as a baseline method, but it may not be able to correctly select the winning move. Therefore, we propose “Purple Piece AI”, a method to introduce the idea of “Purple Piece” used in AND/OR tree search, and implement it in MinMax search. The program that implements the Purple Piece AI outperforms all the programs in the competitive experiments (random, rule-based, and EACP). The Purple Piece AI avoids risky moves that lead to losses, and aims for a sure win. In addition, the Purple Piece AI won the first place out of eight teams in the GAT2020 Geister AI competition.

Keywords: incomplete information game, game tree search

1. はじめに

近年, ハードウェア・アルゴリズムの発達により, オセ

ロ, チェス, 将棋, 囲碁など完全情報ゲームでは, トッププレイヤーに勝る AI が実現し大きな成果をあげた。2013 年には将棋 AI 「Ponanza」が人間のプロ棋士に勝利し, 2016 年には囲碁 AI 「AlphaGo [1]」が囲碁トッププレイヤーのイ・セドル氏に 4 勝 1 敗で勝利し, 世間からの注目を集めた。

不完全情報ゲームの研究もさかんに行われており, ポーカー [2], 麻雀 [3], [4] といったカード型ゲームでは, トッププレイヤー相当の AI が実現している。

¹ 株式会社 Lightblue Technology
Lightblue Technology, Chiyoda, Tokyo 101-0052, Japan

² 松江工業高等専門学校
National Institute of Technology, Matsue College, Matsue,
Shimane 690-8518, Japan

a) keimahatokusyu@gmail.com

b) hashimoto@matsue-ct.jp

一方、チェスとルールが似た駒移動型の不完全情報ゲームでは、Kriegspiel [5], 衝立詰将棋 [6], Chinese Dark Chess [7], ガイスター [8], [9], [10], [11], [12], [13], [14] *1の研究などが行われている。文献 [5], [6] は、いずれも不完全情報の詰将棋に似た問題を題材として、AND-OR search solver を提案し、問題の解答に成功している。強い AI プレイヤに関する研究としてはモンテカルロ木探索を使った報告が多い。文献 [7] は、モンテカルロ木探索にさまざまなヒューリスティックスを加えた手法を提案し、軍人将棋に似た Chinese Dark Chess の AI プレイヤを開発している。比較的ルールが簡単なガイスターは、近年 AI 大会が定期的実施され、駒移動型不完全情報ゲームのテストベッドとして注目されている。しかし、このジャンルのゲームでトッププレイヤを超える強さを持つ AI はまだ報告されていない。

本稿では、ガイスターを題材とし、強い AI プレイヤ作成を目的とする。ガイスターは青赤 2 種の駒を用いた 2 人対戦の不完全情報ゲームで、盤上の駒色は非公開だが、取ると色を観測できる。ガイスター AI の先行研究として、モンテカルロ法 [8], Q 学習と必勝手探索 [9], ルールベースによる駒推定 [10], 駒推定と深層学習 [11], ナッシュ均衡を求める CFR [12], Deep CFR [13] を用いるもの、限定的なルールベースのキーパー戦略 [14] が報告されている。しかし、人間より強い AI は報告されておらず、AI 大会でもさまざまな手法が試されている。AlphaZero [15] を使ってガイスター AI を作ることも考えられるが、不完全情報ゲームで AlphaZero の強化学習がうまく働くかは分かっておらず、過去に AlphaZero を用いたガイスター AI が大会に参加した際も結果は芳しくなかった。ガイスターでは、囲碁における UCT [16] のように多くのプログラムが採用する共通手法はまだ確立されていない。

ガイスターでも ISMCTS [17] のようにモンテカルロ木探索を用いる手法が有効な可能性はあるが、比較的シンプルなルールで分岐の少ないガイスターの終盤ではチェスや将棋の探索手法が有効ではないかと予想し、本稿では MinMax 探索をガイスターに適用するアプローチを考え、特に相手の状況が分からないゲームでは難しい探索内での必勝判定に焦点を当てる。探索内での必勝判定は終盤の強さに大きく影響すると考えられる。

不完全情報の扱い方が問題となるが、ベースライン手法として、まず色配置を決め、完全情報探索を行い、最後に集計する「色配置全列挙」を紹介する。次に、色配置全列挙では必勝判定を行えないケースを説明し、その解決策として「紫駒 AI」を提案する。

紫駒 AI の有効性を評価するため、紫駒 AI を実装し、色配置全列挙、ランダム、ルールベースとの対戦実験によっ

て性能を比較する。また、参考として、紫駒 AI を用いたプログラムの大会成績を紹介する。2 章ではガイスターについて述べる。3 章では関連研究を紹介する。4 章ではベースライン手法と提案手法について述べる。5 章では AI の実装について述べる。6 章では対戦実験について述べる。7 章では紫駒 AI の懸念点を論じる。8 章では考察を行う。9 章ではガイスター AI 大会の戦績について述べる。10 章ではこれまでの議論をまとめる。11 章では今後の方針を述べる。

2. ガイスター

2.1 ガイスターのルール

ガイスター (Geister) は、オバケの形をした駒を使って戦うドイツ発祥の 2 人用チェスゲームである。ガイスターでは、背中に青または赤の印がついた駒を用い、互いに印を隠して戦う。先手側から見たガイスター初期配置例を図 1 に示す。各プレイヤは赤駒、青駒を 4 個ずつ自陣 8 マスに自由な組合せで初期配置する。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。交互着手でゲームは進行し、手番プレイヤは駒を 1 つ動かす。駒は上下左右 1 マス移動する。自駒があるマスには移動できないが、相手駒があるマスに移動させると相手駒を取り、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、盤外へ脱出させることができ、勝利となる。勝利条件は、相手の青い駒をすべて取る、自分の赤い駒をすべて取らせる、自分の青駒を 1 つ盤外へ脱出させる、のいずれかである。ガイスターでは、お互いに盤上の相手の駒の情報が分からない状態で駒を動かしていくため、ブラフなどの戦略も要求される。

2.2 移動表現、駒数表記

本稿では、駒の移動を移動元マス (a1-f6)、移動方向

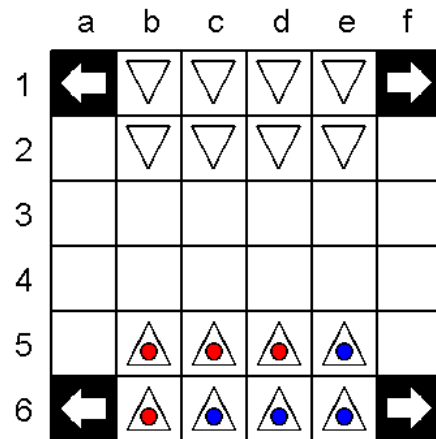


図 1 ガイスター初期配置例

Fig. 1 Example of Geister's initial placement.

*1 メビウスゲームズ：ガイスター、ボードゲーム・カードゲームの店メビウス ゲームズ、入手先 <<http://www.mobius-games.co.jp/Gester.htm>> (参照 2020-10-14)

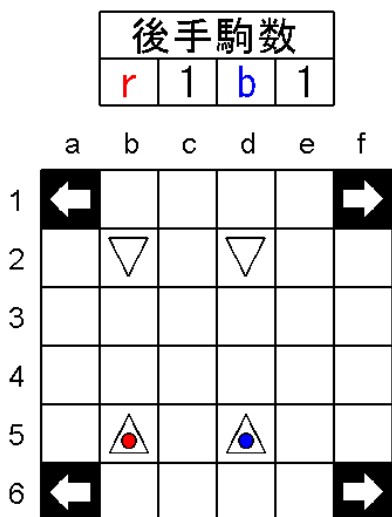


図 2 局面表現
Fig. 2 Expression of position.

(↑→↓←), 駒色を表す文字によって表記する. 駒色は先手赤, 先手青, 後手赤, 後手青をそれぞれ ‘R’, ‘B’, ‘r’, ‘b’ と表記し, 後手の色不明駒を ‘u’ と表記する. たとえば, 図 2 において, b5 から b4 へ先手赤駒を移動する動きは b5 ↑ (R), d2 から d3 へ後手色不明駒を移動する動きは d2 ↓ (u) と表記する. また, ガイスターでは今までに取った駒の色から盤上に残っている相手赤駒, 相手青駒の個数が分かるため, 図 2 のように表現する. 図 2 では, 後手の色不明駒のうち赤 (r) が 1 つ, 青 (b) が 1 つである.

3. 関連研究

チェスとルールが似た駒移動型不完全情報ゲームの研究では, まず Kriegspiel [5], 衝立詰将棋 [6] をそれぞれ題材とした研究で AND-OR search solver が提案され, 詰将棋に似た不完全情報ゲームにおける問題の解答に成功しており, ガイスターの詰み探索で同様の手法が使える可能性はあるが, ガイスタープレイヤー AI としては使えない. Chinese Dark Chess [7] では, モンテカルロ木探索による AI プレイヤーが開発されているが, ゲーム固有のヒューリスティックを多く加えているため, 汎用性が低くガイスター AI への応用は難しい. また, 情報集合をノードとしモンテカルロ木探索を行う手法 ISMCTS [17] が提案され, いくつかの不完全情報ゲームで MCTS の性能を上回っている. この手法がガイスターで有効な可能性はあるが, 本稿ではまず MinMax 探索をガイスターに適用するアプローチを考える.

ガイスター AI の先行研究として, モンテカルロ法 [8], Q 学習と必勝手探索 [9], ルールベースによる駒推定 [10], 駒推定と深層学習 [11] などが報告されている. いずれも, 人間と試合を繰り返し勝ち越すなどの成果は報告されていないが, 文献 [9] では本稿の根幹となる重要なアイデアを提案しているため, 次段落で説明する.

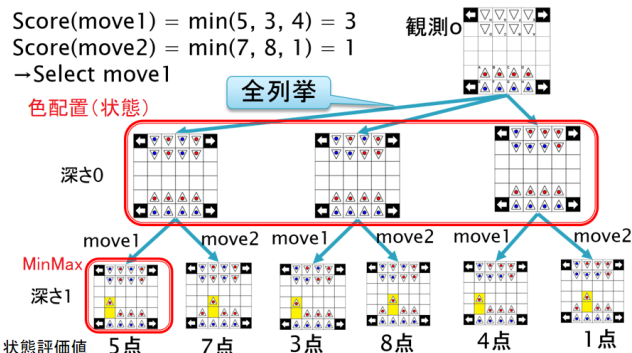


図 3 色配置全列举の例
Fig. 3 Example of the EACP.

文献 [9] では, 3 層ニューラルネットワークを用いた Q 学習を行い, 観測局面を入力とし, 最も高い Q 値を出力した手を選択している. 学習を進めやすくするため, 学習時には手前の 4 列にいる駒を後退不可にするなどの着手制限を加えている. これにより, 人間プレイヤーが使う序盤定跡などを学習できている. しかし, 3 手の必勝局面であっても必勝手を指すことができていないなど強さの面で課題があった. そこで, 簡単な必勝局面を検出するために, 必勝手探索を行うための手法を提案している. 具体的には, 相手の駒を「取ったら赤, 脱出するときは青に変化する駒 (紫駒)」と見なす方法である. これは最悪の状況を想定する方法であり, 完全情報ゲームの探索によって必勝手を検出できる. 先行研究では必勝手探索に特化したアルゴリズムである dfpn [18] を用いて必勝手を探索し, 必勝でない場合に Q 学習の結果を用いるという改善を行っている. これによりランダム AI に対する勝率が約 60% から約 70% へ改善されている. 紫駒のアイデアは, ガイスター終盤の探索に有効な手法として, 次章で取り上げる.

4. 提案手法

本章ではベースラインとなる手法と提案手法について述べる. 4.1 節ではベースライン手法として「色配置全列举」について述べ, 4.2 節ではその問題点を説明し, 4.3 節では問題点を解決した提案手法「紫駒 AI」について述べる.

4.1 色配置全列举

色配置全列举 (Enumerating All Color Placements, EACP) による手法を説明する. 例を図 3 に示す. 色配置全列举では, 各手の評価値を計算し評価値最大の手を選ぶ. 各手の評価値を計算する際, 「相手駒の色配置」を観測できないことが問題となる. しかし, 現局面と対応する色配置はただか 70 通りしかないため, すべての色配置を試すことができる. そこで手の評価値を計算する際に, 現局面と対応する各色配置について, その手を指した直後の局面から MinMax 探索を行い, 「色配置評価値」を集計する. 色配置全列举では, 「色配置評価値の最小」を

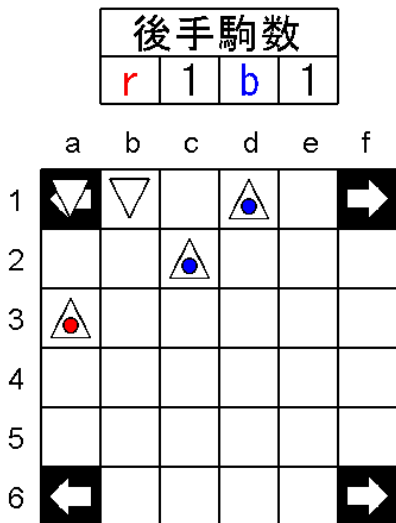


図 4 色配置全列挙がうまくいかない局面
Fig. 4 Position where the EACP does not work.

手の評価値としている。最小を取るによりすぐ負ける手を避けると考えられる。色配置全列挙を実装した AI は、GPW2017 ガイスター AI 大会で優勝するなどの成果をあげている。しかし、色配置全列挙では、原理的に必勝手を正しく求められないことがある。

4.2 色配置全列挙の問題点

色配置全列挙では、各色配置について独立にゲーム木を探索する。決定論的に決められるので一見最善を尽くせるように思えるが、必勝手を正しく選択できない場合がある。

具体例を図 4 に示す。先手の手を考える。後手の色配置は 2 通りある。ここで先手が 1 手目に赤を a3 ↑ (R) と a2 に移動する場合を考える。まず a1 の駒が青の場合、後手は a1 ↓ (b) と動かすと a2 の赤を取り負け、動かさないと a2 ↑ (R) で a1 の青を取られて負けとなるため、先手勝ちである。a1 が赤の場合、後手は a1 ↓ (r) と動かすと a2 の赤を取り負け、b1 ↓ (b) または b1 → (b) と動かすとその駒を取られ負けとなるため、先手勝ちである。よって図 4 では、後手の色配置どちらを仮定しても a3 ↑ (R) が最短の 3 手で勝てる手となる。そのため、4.1 節の色配置全列挙では a3 ↑ (R) を選択する。

しかし、実際には相手駒の色配置を特定できないため、a3 ↑ (R) を選択し 3 手で確実に勝つことはできない。図 4 の局面では、d1 → (B) と青駒を動かし f1 からの脱出を狙うのが正解手となる。このように、すべての色配置を考え探索しても、簡単な必勝手さえ原理的に求まらない場合がある。また、色配置全列挙では色配置の数だけ木探索を行うため計算量が大きくなる。

そこで、探索内での必勝手判定を可能にするため、先行研究で AND/OR 木探索に用いられた「紫駒」のアイデア導入を考える。

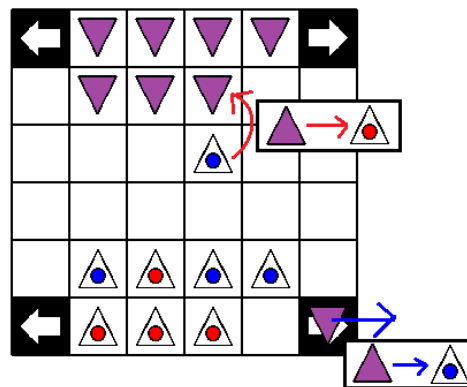


図 5 紫駒概念
Fig. 5 Concept of the purple piece.

4.3 紫駒 AI

先行研究 [9] では、紫駒を「取ったら赤、脱出時は青に変化する駒」と定義し (図 5)、相手駒を紫駒に変換することで、必勝手探索を行っていた。たとえば図 4 の局面で、先手が a1, b1 にある駒を紫駒と見なし探索を行ったとする。このとき探索では、後手駒を 1 つでも取ったら先手負けと考えるため、a3 ↑ (R) は選択されず、正しい必勝手である d1 → (B) が選ばれる。

先行研究では紫駒を必勝手を見つけるのに特化した探索法 dfpn [18] と組み合わせ、必勝手の検出を行っていた。しかし必勝手が見つからない場合は別手法を用いており、ランダム AI に勝率 70% 程度の勝率であった。そこで本稿では、紫駒のアイデアを MinMax 探索など 2 人完全情報ゲームの探索に導入する手法「紫駒 AI」を提案し、強さの向上を目指す。

紫駒 AI では、盤上の相手駒を紫駒と見なし、MinMax 探索を適用する。紫駒 AI はつねに最悪のケースを想定して探索するため、必勝手順が見つければ必勝であることが保証される。また、相手駒を取った場合の振舞いなどが確定するため、完全情報ゲームの探索手法を用いることができ、必勝手以外の良手を探索できる。本稿では、紫駒 AI の探索手法として MinMax 探索を用いる。紫駒 AI では、ゲームツリーの探索を 1 回のみ行うため、色配置全列挙よりも速く計算できる。

5. AI の実装

紫駒 AI 「PurpleMax」の有効性を評価するため、本稿では、色配置全列挙 AI、紫駒 AI、ベンチマーク AI を使用する (表 1)。

Naotti1.0 は GPW2017 杯ガイスター AI 大会優勝 AI であり、評価関数は「青駒の個数差」である。Naotti1.1 は Naotti1.0 の評価関数を強化した AI で、評価関数は、青駒を残したい、ゴールに攻めたいという経験則に基づく。具体的には、青駒の個数差 (自分 - 相手) を b 、各駒の近い脱出口までのマンハッタン距離を合計したものの差 (自分

表 1 AI 説明

Table 1 Description of AIs.

| AI 名 | 概要 |
|-----------|-----------------------|
| PurpleMax | 紫駒 AI |
| ランダム | ランダム行動 |
| 青猪突 | ルールベース 1 |
| 赤猪突 | ルールベース 2 |
| Naott1.0 | GPW2017 優勝 AI, 色配置全列挙 |
| Naott1.1 | Naott1.0 の評価関数を強化 |

駒 – 相手駒) を D , 定数 $C = 0.001$ とし, $b - CD$ とする. Naott1.0, Naott1.1 では MinMax 探索 ($\alpha\beta$ 探索) の深さを 5 とする.

次に, 紫駒 AI 「PurpleMax」を実装する. 探索には MinMax 探索を用いる. 探索深さは 5 とする. ルートプレイヤーを自分, 非ルートプレイヤーを相手と呼ぶ. 局面評価関数は Naott1.1 と同様だが, 探索中は取った駒を赤と見なすため, 相手の青個数は変化しない.

また, ベンチマーク AI として, ランダム, 青猪突, 赤猪突を実装する. ランダムはランダムに駒を選択しランダムな方向に動かす AI である. ただし勝敗条件などの実装を簡単にするため, 1 手で脱出できる場合は脱出を行う. 青猪突は相手陣に最も近い青駒を相手陣に近づけるルールベース, 赤猪突は赤駒で同様のことをしたルールベースである.

6. 対戦実験

6.1 実験方法

本実験では, PurpleMax とベンチマーク AI, Naott1.0, Naott1.1 の対戦を行う. どのプログラムも初期配置をランダムに選択する. 各組合せで, 先後を入れ替え 100 試合ずつ, 合計 200 試合行う.

6.2 実験結果

表 2 に Naott1.1 と Naott1.0 の比較, 表 3 に PurpleMax の勝敗, 表 4 に PurpleMax, Naott1.1 の対戦合計思考時間 (200 試合) を示す.

表 2 より, Naott1.1 が Naott1.0 に対し高い勝率をあげており, 評価関数の改良には成功していることが分かる. 表 3 より, PurpleMax は Naott1.1 以外には大きく勝ち越し, Naott1.1 に対しては正規分布により片側検定したところ有意水準 2.5% で勝ち越している. 表 4 より PurpleMax の合計思考時間 (200 試合) は Naott1.1 の約 1/49 である. よって, 思考時間が短いにもかかわらず, Naott1.1 よりも強くなっていると考えられる.

思考時間には大きく差がついているが, これは深さを 5 と固定しており, PurpleMax では 1 回だけ深さ 5 の MinMax 探索を行うが, Naott1.1 では色配置数だけ深さ 5 の MinMax 探索を行っていることによる. そのため, 評価関数の計算時間が同じであれば, Naott1.1 よりも PurpleMax

表 2 Naott1.1 の勝敗

Table 2 Naott1.1 wins and loses.

| 対戦相手 | 勝 | 敗 | 引き分け |
|----------|----|----|------|
| Naott1.0 | 75 | 13 | 112 |

表 3 PurpleMax の勝敗まとめ

Table 3 Summary of PurpleMax wins and losses.

| 対戦相手 | 勝 | 敗 | 引き分け |
|----------|-----|----|------|
| ランダム | 200 | 0 | 0 |
| 青猪突 | 200 | 0 | 0 |
| 赤猪突 | 116 | 0 | 84 |
| Naott1.0 | 90 | 2 | 108 |
| Naott1.1 | 87 | 50 | 63 |

表 4 合計思考時間 (200 試合)

Table 4 Total thinking time (200 games).

| AI 名 | 思考時間 [sec] |
|-----------|------------|
| Naott1.1 | 439.715 |
| PurpleMax | 8.972 |

表 5 行動回数と色の関係

Table 5 Relationship between number of actions and color.

| | 合計回数 | 赤 | 青 | 赤の割合 | 青の割合 |
|----|------|-----|-----|-------|-------|
| 接近 | 628 | 580 | 48 | 0.924 | 0.076 |
| 逃げ | 533 | 90 | 443 | 0.169 | 0.831 |

の方が短い思考時間で同じ深さを読むことができる.

7. 紫駒 AI の懸念点

紫駒 AI は初期配置を除いて純粋戦略である. すなわち, 同じ局面に対して必ず同じ行動をとる. そのため, 癖を見抜かれると弱い可能性がある. 本章では, 駒色と動きの関係を調べ, 紫駒 AI の課題点を分析する.

7.1 実験方法

PurpleMax どうしを 100 回対戦させ, 先手側について次の 2 種類の行動が起きた回数を駒色ごとにカウントする.

- 「接近」…相手駒と上下左右に隣接していない駒を, 相手駒と上下左右に隣接するマスに動かす行動
- 「逃げ」…相手駒と上下左右に隣接する駒を, 相手駒と上下左右に隣接しないマスに動かす行動

ただし, 駒を取る手, 脱出手, 負けと見なす局面の手はカウントしない.

7.2 実験結果

実験結果を表 5 に示す. 表 5 より, 「接近」「逃げ」ともに赤駒, 青駒の行動回数に偏りがあり, 特に「接近」を行った駒は赤が 92.4% を占める. 紫駒 AI は行動から状態を推定されやすい可能性があり, 対策を行った相手には弱いと考えられる.

8. 考察

対戦実験より、本稿で実装した紫駒 AI「PurpleMax」は、GPW2017 優勝 AI の評価関数改良版「Naotti1.1」よりも短い思考時間にもかかわらず、Naotti1.1 を含むベンチマーク AI いずれに対しても強いと考えられる。PurpleMax が強い理由は、必勝手を発見できること、負ける可能性を回避できることだと考えられる。

まず、必勝手を発見できる例を図 6 に示す。図 6 は先手の 7 手必勝局面であり、必勝手は e2 ↑ (R) で後手の d1 → を防ぎ、f3 (B) の f1 からの脱出を図る手である。しかし EACP では、f3 ↑ (B)、d1 → の後、c1、e1 いずれか青の方を取れるので f3 ↑ (B) を必勝手と見なしてしまう。紫駒 AI では、f3 ↑ (B)、d1 → としたとき c1、e1 をいずれも紫駒と考えるため f3 ↑ (B) を必勝手と見なすことはなく、探索深さが 5 手あれば、e2 ↑ (R) を選択する。ゴールへ移動させた青駒は相手駒が隣接しない時点で勝ちと判定できるため、この例では探索深さ 5 で十分となる。

次に、負ける可能性を回避できる例を図 7 に示す。図 7 は先手の 1 手目が b6 ← (R) 以外のとき、負ける可能性のある局面である。後手が a5 ↓ と駒を動かすと、先手は a6 の駒を取っても取らなくても負けになる可能性がある。しかし EACP では、f3 ↑ (B) を選択した場合、後手 a5 ↓ ならば a6、b5 いずれか青の方を取ることができ、それ以外ならば f1 からの脱出ができると考え、f3 ↑ (B) を必勝手と見なしてしまう。紫駒 AI では、b6 ← (R) 以外のとき、後手が a5 ↓ と動かした場合に負けと考えるため、それを阻止するために b6 ← (R) と相手ゴールを防ぐ手を選択する。その後、f3 (B) を f1 から脱出させることができる。

7 章の実験結果より PurpleMax は駒を推定されやすい

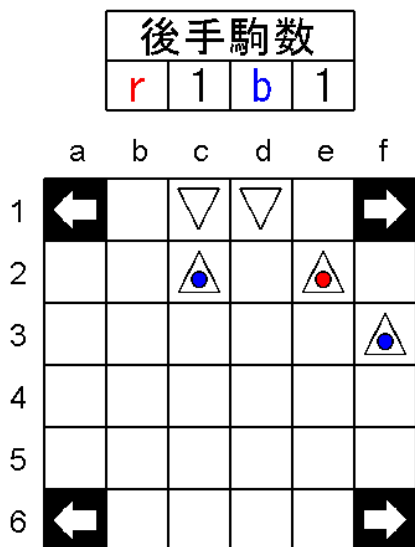


図 6 紫駒 AI が必勝手を発見できる局面例

Fig. 6 Example of a position in which the Purple Piece AI can find a winning move.

可能性がある。たとえば PurpleMax が「接近」を行った場合、接近した駒を赤と決め打っても、およそ 9 割当たることになる。原因として、PurpleMax の評価関数に青駒数をを用い明示的に青駒温存を促していること、初期配置以外に確率的な行動をとらないことが考えられる。また、青駒を突進し相手駒に隣接させるような手を選ぶことがほとんどなく、消極的と感じられる手を選ぶことが多い。理由として、相手駒を紫駒と見なすことで悲観的になりやすいだけでなく、自駒の色を完全情報として探索していることがあげられる。本来は相手から自駒の色は分からないはずだが、自駒の色を確定した状態で探索するので、突進した青駒は相手に取られ不利になると評価しやすい。

紫駒 AI をルールの違う他の不完全情報ゲームにそのまま応用することはできないが、紫駒のように探索中の分岐でルートプレイヤーから見て都合の悪そうな状態を仮定することで、強いプレイヤーができる可能性がある。軍人将棋、軍旗（中国の有名な軍人将棋系ゲーム）は特定のマスに到達すると勝ちとなるルールがあるため、衝突でほとんど負けを仮定しても、勝ちまで読み切ることで結果として強い終盤プレイヤーができるかもしれない。衝立将棋系への応用は盤面が大きいと難しそうだが、たとえばどうぶつしょうぎのようなサイズなら強いプレイヤーができるかもしれない。

9. ガイスター AI 大会の戦績

参考として、本稿の AI「PurpleMax」をベースとしたプログラムのガイスター大会での成績を紹介する。PurpleMax ベースプログラムは、ガイスター AI 大会*2に 4 回参加し、

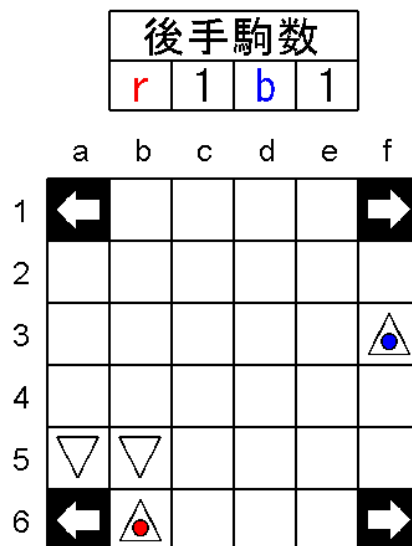


図 7 紫駒 AI が相手駒の脱出を阻止して勝つ局面例

Fig. 7 Example of a position in which the Purple Piece AI wins by preventing a opponent's piece from escaping.

*2 ガイスター AI 大会

<http://www2.matsue-ct.ac.jp/home/hashimoto/geister/GAT/>

表 6 ガイスター AI 大会の戦績 (PurpleMax)

Table 6 Results of the Geister AI Tournaments (PurpleMax).

| 大会名 | 参加数 | 予選勝敗 | 予選順位 | 決勝順位 |
|---------|-----|--------------|------|------|
| GPW2018 | 5 | 5 勝 1 敗 2 分 | 2 | - |
| GAT2019 | 7 | 11 勝 1 敗 0 分 | 1 | 2 |
| GPW2019 | 4 | 4 勝 2 敗 0 分 | 1 | 2 |
| GAT2020 | 8 | 10 勝 2 敗 2 分 | 1 | 1 |

表 7 GAT2019 多対戦大会 対戦結果 (勝-敗-分)

Table 7 Tournament table with multiple matches in GAT2019 (win-lose-draw).

| | Naotti | 立木 | SSM | 計 | 順位 |
|---------------------|--------|-------|-------|--------|----|
| NaottiMinMax | | 6-3-1 | 5-2-3 | 11-5-4 | 1 |
| 立木システムプレイヤー | 3-6-1 | | 2-6-2 | 5-12-3 | 3 |
| StrongStrongMasashi | 2-5-3 | 6-2-2 | | 8-7-5 | 2 |

1 回優勝, 3 回準優勝を果たした。大会は毎年 2 回ペースで開催されており, PurpleMax ベースプログラムが参加したのは, GPW2018, GAT2019, GPW2019, GAT2020 大会である。各大会それぞれ NaottiMinMax, NaottiMinMax, Murasaking, Naotti2020-3 という名前で参加した。GPW2018 に参加した NaottiMinMax は PurpleMax と同一で, それ以降に参加した AI はいくつか変更点はあるが, いずれも PurpleMax と大きな差はない。大会では各組合せ先後 1 試合ずつの総当たり戦を行った。GPW2018 では勝ち数, 総当たり戦のみの大会だったが, GAT2019 以降の 3 大会ではポイント制 (勝ち 3, 負け 0, 引分 1) を導入し, 予選リーグ Top2 が決勝でプレーオフする形式であった。また, GAT2019 大会では試験的に多対戦部門も行われ, 各組合せで先後を入れ替え 5 試合行われた。

ガイスター大会の戦績を表 6 に示す。大会では 200 手で未決着なら引き分けとしている。GPW2018 大会では, 引き分けを除く勝率は最も高かったが, 引き分けにより勝ち数を稼げず, 2 位となった。GAT2019 大会では予選 1 位となり, 決勝では敗れたが, 実力がより反映されやすい多対戦大会では, 表 7 の結果となり, 優勝した。さらに GAT2020 では, GPW2019 大会で優勝した相手「立木システムプレイヤー」に勝利し優勝した。

GAT2020 決勝戦 2 試合目では, 後手 Naotti2020-3 (PurpleMax) が 51 手ですべての赤を取らせ勝利した。まず, 37 手で図 8 の局面となり, b2 ← (r), a3 ↑ (B) と Naotti 側が a2 の赤を取らせ, 赤をあと 1 つ取らせれば勝利する展開となった。その後, Naotti2020-3 は赤駒を前進させ, 図 9 の局面で a5 ↓ (r) と赤駒をゴールへ前進, その後 b6 ← (R) と立木側に赤を取らせ勝利した。対戦相手の「立木システムプレイヤー」は駒の色をルールベースによって推定する AI である。PurpleMax は, 青駒より赤駒を前進させやすい傾向にあるが, 2 試合目は 4 段目に初めて侵入したのが 46 手目と終盤であり, 赤駒も含め慎重な

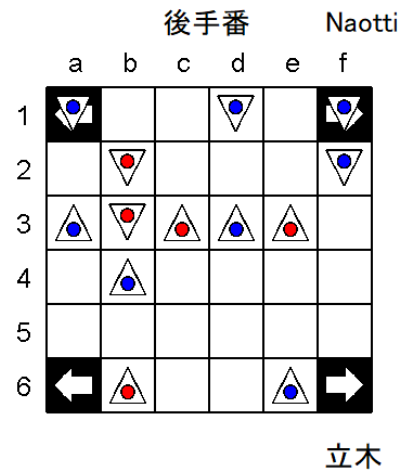


図 8 GAT2020 決勝 2 試合目 (38 手目)

Fig. 8 GAT2020 Final, 2nd game (38th move).

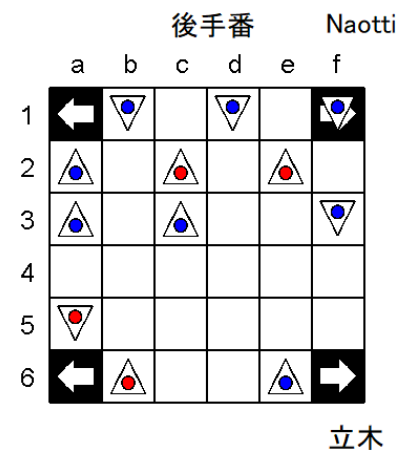


図 9 GAT2020 決勝 2 試合目 (50 手目)

Fig. 9 GAT2020 Final, 2nd game (50th move).

攻めであった。その結果として, 相手に赤を青と誤認させ取らせることに成功したと考えている。

10. まとめ

不完全情報ゲーム「ガイスター」を対象とし, 相手の状況が分からないゲームでは難しい探索内での必勝判定に焦点を当てた。ベースライン手法とした「色配置全列挙」には必勝判定を原理的に正しく行えない場合があるが, 紫駒 [9] を用いることで必勝判定を行うことができる。本稿では, AND/OR 木探索のみに用いられていた紫駒のアイデアを MinMax 探索に導入し, 探索内で必勝判定を行いながら評価値の高い手を探す「紫駒 AI」を提案した。結果, 対戦実験に用いたすべてのプログラム (ランダム, ルールベース, ベースライン手法) に勝ち越した。紫駒 AI は, 負けにつながる危険な手を避け, 相手の隙を待ち, 確実な勝ちを狙う。紫駒 AI を実装したプログラム「PurpleMax」は, 対戦実験に用いたすべてのプログラムに勝ち越し, GAT2020 ガイスター AI 大会でも 8 チーム中 1 位を獲得した。

11. 今後の方針

今後の課題として、駒推定のされやすさを考慮した評価関数の設計、確率的な戦略の導入があげられる。本稿では MinMax 探索を用いているが、評価関数は作り込まれていない。そのため、評価関数を強化学習などで作り込むことで、より高い性能を示す可能性がある。確率的な戦略としては、まず ISMCTS [17] と同様の手法を MinMax 探索に導入し、紫駒 AI と性能を比較しながら新たな手法を模索していきたい。また、モンテカルロ法などの探索法を用いてみるのも興味深い。さらに、ポーカーの手法などをベースに混合戦略（確率的な戦略）を取る AI も考えたい。

謝辞 本研究は JSPS 科研費 17K00514 の助成を受けたものです。

参考文献

- [1] Silver, D. et al.: Mastering the Game of Go without Human Knowledge, *Nature*, Vol.550, pp.354–359 (2017).
- [2] Bowling, M. et al.: Heads-up limit hold'em poker is solved, *Science*, Vol.347, No.6218, pp.145–149 (2015).
- [3] 水上直紀, 鶴岡慶雅: 期待最終順位に基づくコンピュータ麻雀プレイヤーの構築, 第 20 回ゲームプログラミングワークショップ, pp.25–30 (2015).
- [4] Li, J. et al.: Suphx: Mastering Mahjong with Deep Reinforcement Learning, *Artificial Intelligence*, arXiv:2003.13590v2, pp.1–28 (2020).
- [5] Russell, S. et al.: Efficient belief-state AND-OR search, with application to Kriegspiel, *Proc. 19th International Joint Conference on Artificial Intelligence (IJCAI '05)* (2005).
- [6] Sakuta, M. et al.: Solving problems with uncertainty: A case study using Tsuitate-Tsume-Shogi, *Proc. Game Programming Workshop in Japan*, Vol.99, pp.145–152 (1999).
- [7] Yen, S. et al.: Design and Implementation of Chinese Dark Chess Programs, *IEEE Trans. Computational Intelligence and AI in Games*, Vol.7, No.1, pp.66–74 (2015).
- [8] 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).
- [9] 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).
- [10] 末續鴻輝, 織田祐輔: 機械学習を用いないガイスターの行動アルゴリズム開発, *GAT2018*, pp.13–16 (2018).
- [11] 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).
- [12] Chen, C. and Kaneko, T.: 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).
- [13] Chen, C. and Kaneko, T.: 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).
- [14] 伊藤雅士, 大久保壮浩, 木谷裕紀, 小野廣隆: ガイスター AI のキーパー戦略の有効性, 情報処理学会研究報告, Vol.2019-GI-42, No.3, pp.1–7 (2019).
- [15] Silver, D. et al.: A general reinforcement learning algo-

- rithm that masters chess, shogi, and Go through self-play, *Science*, Vol.362, No.6419, pp.1140–1144 (2018).
- [16] Kocsis, L. and Szepesvári, C.: Bandit Based Monte-Carlo Planning, *ECML 2006*, Lecture Notes in Computer Science, Vol.4212, pp.282–293 (2006).
- [17] Cowling, P.I., Powley, E.J. and Whitehouse, D.: Information Set Monte Carlo Tree Search, *IEEE Trans. Computational Intelligence and AI in Games*, Vol.4, No.2, pp.120–143 (2012).
- [18] Nagai, A.: Df-pn Algorithm for Searching AND/OR Trees and Its Applications, Ph.D. thesis, Department of Information Science, University of Tokyo (2002).



川上 直人 (学生会員)

1997 年生。2019 年松江工業高等専門学校電子情報システム工学専攻卒業。2021 年北陸先端科学技術大学院大学修士課程修了。第 43 回ゲーム情報学研究会若手奨励賞受賞。



橋本 剛 (正会員)

1970 年生。1994 年京都大学農学部卒業。1996 年東京大学大学院理学系研究科修士課程在学中に中国雲南民族学院へ留学。1997 年同大学院中途退学。2002 年静岡大学大学院博士後期課程修了。博士 (工学)。同年学術振興会特別研究員 PD。2003 年カナダアルバータ大学客員研究員。2005 年北陸先端科学技術大学院大学講師。2010 年松江工業高等専門学校准教授。2020 年同教授。ゲーム情報学研究に従事。コンピュータ将棋協会、ロボカップ日本委員会各会員。本会シニア会員。