

完全情報ゲームの探索を用いたガイスター AI の研究

川上 直人^{1,a)} 橋本 剛^{1,b)}

概要: 近年, 囲碁や将棋の AI が人間のトッププレイヤーに勝利するなど完全情報ゲームの研究は大きな成果を上げており, 次のターゲットとして不完全情報ゲームが注目されている. バックギャモン, 麻雀, ポーカーにおいてはトッププレイヤー相当の実力を持つ AI が研究されているが, ガイスター, 軍人将棋など, チェスや将棋とルールが似ている不完全情報ゲームでは強い AI の研究があまり行われていない. 本研究ではそのようなゲームのうち比較的ルールが単純なゲーム「ガイスター」を題材とする. 本研究では, 見えない相手駒の全組み合わせに対して完全情報ゲームの探索を用い, 集計する手法に焦点を当て, いくつかの手法を提案し, 対戦実験で性能を評価する. 実験結果から, 従来プログラムの性能を大きく上回る手法を確認できた. また, 実験の中で, 完全情報ガイスターとしてはとても弱い評価関数が, 通常のガイスターでは強い評価関数になることもあるという知見を得た.

Research of Geister AI using search for complete information games

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

Abstract: In recent years, research of complete information games has made great results as AI of Go and Shogi won the top player of human. Incomplete information games are drawing attention as the next target. In incomplete information games, AI of backgammon, mahjong and poker are studied and that are equivalent to top human players. However, there are few research aiming strong AI for incomplete information games which are similar to rules of chess, such as Geister or Military Shogi. Here, we focus on the game of "Geister" which rule is relatively simple out of such games. In this research, we use search method of complete information games, and think how to use the result of such search. We propose several methods and evaluate the performance by selfplay experiments. From the experimental results, we could confirm the method which greatly exceeds the performance of the conventional program. Also, in the experiment, we got the knowledge that the very weak evaluation function in the "Complete Information Geister" may become a strong evaluation function in ordinary Geister.

1. はじめに

コンピュータの性能, 学習アルゴリズムの発達によって, 完全情報ゲームの研究は大きな成果を上げている. 2013 年には将棋 AI 「Ponanza」が人間のプロ棋士に勝利し, 2015 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. 完全情報ゲームのいくつかはコンピュータプレイヤーが人間トッププレイヤーの実力を上回ったため, 次のターゲッ

トとして不完全情報ゲームが注目されている.

不完全情報ゲームでも, バックギャモン [3], 麻雀 [7], ポーカー [2] においてはトッププレイヤー相当の実力を持つ AI が研究されている. しかし, ガイスター, 軍人将棋など, チェスや将棋とルールが似ている不完全情報ゲームでは, 強い AI の研究があまり行われていない. そこで, チェスや将棋とルールが似ている不完全情報ゲームを対象とし, 高い実力を持つ AI の研究をすることにした. 本研究では, そのようなゲームの一つである「ガイスター」を対象とする. ガイスターを選んだ理由は, ルールが単純でプログラムの性能評価がしやすいと考えたためである. 本研究では, 完全情報ゲームの探索を用いる.

ガイスターを題材とした研究はいくつか報告されてい

¹ 松江工業高等専門学校
National Institute of Technology, Matsue College

^{a)} keimahatokusyu@gmail.com

^{b)} hashimoto@matsue-ct.jp

る [6][5][4]. [5] では、強化学習を用いている。また、証明数探索 (dfpn) による必勝判定をおこなっている。また、[4] では全状態について MinMax 探索をおこなっており、GPW2017 ガイスター AI 大会で優勝している。本研究では、[5] の必勝判定を応用することで、従来の [4] に圧勝したため、その手法を紹介する。また、[4] の MinMax 探索で用いられている評価関数を改善し、全状態を探索した結果の集計法のバリエーションを増やし、[5] の必勝判定を応用した AI と対戦実験をおこなったので、その結果を報告する。

2 章ではガイスターについて述べる。3 章では先行研究について述べる。4 章では方針を述べる。5 章では評価関数について述べる。6 章では完全情報ガイスターについて述べる。7 章では集計法の工夫について述べる。8 章では紫駒の応用について述べる。9 章では AI の実装について述べる。10 章では実験方法について述べる。11 章では実験結果について述べる。12 章では考察をおこなう。13 章ではまとめをおこなう。14 章では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) 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り、背中に印がついた駒を相手に印が見えないように配置し、駒を取り合う。ガイスターでは、背中に青い印のついた良いオバケ駒と、背中に赤い印のついた悪いオバケ駒の 2 種類を用いる。ガイスターの初期配置を図 1 に示す。各プレイヤーは最初にそれぞれの駒を 4 個ずつ自陣 8 マスに自由に配置し、交互に駒を動かしていく。駒を動かせる範囲は縦横に隣接する 4 マスであり、1 回に動かせる駒は 1 個である。また、相手の駒があるマスに自分の駒を移動させることで、相手の駒を取ることができ、取った駒の色を知ることができる。勝利条件は、相手の青い駒を全て取る・自分の赤い駒を全て取らせる・自分の青い駒を 1 個盤外へ脱出させる、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、ブラフやハッタリといった戦略を要求される。

3. ガイスターに関する先行研究

先行研究には、モンテカルロ法を用いた手法 [6]、強化学習と必勝手探索を用いた手法 [5]、MinMax 法を適用した手法 [4] がある。

[5] では、3 層ニューラルネットワークを用いた Q 学習の結果を利用して手を決定している。しかし、3 手で必ず勝てるような局面であっても必勝手を指すことができなかつたため、簡単な必勝局面を見落とさないために、必勝手探索をおこなっている。具体的には、相手駒を「取ったら赤に変化するが脱出時には青に変化する駒」として扱うことで証明数探索 (Dfnp) をおこない、必勝局面の判定をしている。[5] は、モンテカルロ法を用いたプレイヤーに勝ち越して

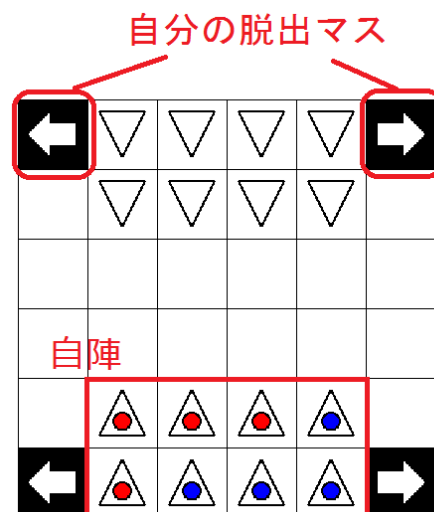


図 1 ガイスターの初期配置

いるものの、ランダム AI に対する勝率が 70%程度と弱い。

[6]では、局面推定によってプレイアウト回数の工夫をしたモンテカルロ法をすることで、単純なモンテカルロ法に対して勝率 55%、猪突猛進 AI(青駒をゴールへ近づける AI)に対して勝率 98%という結果を出している。一見優れた結果に見えるが、[5]が単純なモンテカルロ法に勝ち越していること、[6]が単純なモンテカルロ法との対戦で 45%負けていることから、そこまで強くないと考えられる。

[4]は全状態に対して「互いに駒色の見えるガイスター」を考え同じ深さの MinMax 探索をしている。全状態に対して同じ深さの探索をおこなう利点は、「どの状態の探索結果もそこそ信用できる点」「探索の深さが決まっているのでデバッグや対戦結果の考察が容易な点」にある。実際 [4]は、GPW2017 杯ガイスター AI 大会で 6 プログラム中 1 位を獲得している。しかし、MinMax 探索に用いている評価関数が弱く、同大会のエキシビジョンマッチで人間プレイヤーと 3 回対戦したところ全敗してしまった。このように、ガイスターの研究はいくつかあるものの、強いプログラムがまだ報告されていないのが現状である。

4. 方針

本研究では、GPW2017 ガイスター AI 大会優勝 AI「NaottiMinMax」[4]をベースに以下の 3 方法を提案、実装し評価を行う。

- 評価関数の強化
- [4]で使われる、新しい探索結果集計法の提案
- [5]の必勝局面探索で使われた「紫駒」の応用

5. 評価関数

[4]では、「青駒の個数の差」のみを評価している。しかし、これだけでは弱いのでメタヒューリスティックを用いた改善をおこなう。[4]の MinMax 探索内部では「互いに駒色が見えるガイスター（以降、完全情報ガイスターとよぶ）」を考えている。また、[5]の必勝局面探索でも（相手の駒を特殊な駒としているとはいえ）完全情報ガイスターを考えている。したがって、直感的には「完全情報ガイスター」が強くなるような評価関数を使うのが良いと考えられる。完全情報ガイスターでは、「青駒を多く保持している」「駒をより前進させている」プレイヤーが有利だという経験則がある。そこで、本研究では青駒の個数のほかに、「各駒のゴールまでのマンハッタン距離の和」を使用する。

具体的には、

- 青駒の個数による評価 + weight * 駒の位置による評価で計算する。weight は定数とし、実験によって調整する。ただし、勝ち状態、負け状態の評価値は ∞ , $-\infty$ とする。

だが、完全情報ガイスターが強くなるような評価関数を使うことで、本当にガイスターを強くすることができるかは自明でない。そこで本研究では「完全情報ガイスター」

「通常のガイスター」両方に対して、評価関数のパラメータ weight を調整する実験をおこない、勝率などの比較をおこなう。

5.1 青駒の個数による評価

自分の青駒の個数 - 敵の青駒の個数を評価値とする。

5.2 駒の位置による評価

各駒から相手陣の隅マスまでのマンハッタン距離を計算し、その和を評価値に用いる。ガイスターは経験上、駒を前進させることで有利になる傾向があるため、相手陣に近づくほど高い評価を得られるように評価値を計算する。具体的には以下のように計算する。

- 自分の各駒について敵陣の近いほうの隅マスまでのマンハッタン距離を計算し、その和を Dist1 とする。
- 敵の各駒について自陣の近いほうの隅マスまでのマンハッタン距離を計算し、その和を Dist2 とする。

このとき、「Dist2 - Dist1」を駒の位置による評価とする。

6. 完全情報ガイスター

完全情報ゲームの探索を用いる手法において、完全情報ゲームの探索を強化するとガイスターも強くなるかを検証するために、互いに駒の色が見えるガイスターを考える。以降、完全情報ガイスターと呼ぶ。完全情報ガイスターの例を図 2 に示す。完全情報ガイスターでは、相手駒の色を観測できるため、取られることで有利になりやすい「赤駒」を積極的に相手陣に近づけ、取られることで不利になりやすい「青駒」は相手駒から離れるように動くような戦い方になりやすいと考えられる。実際、深さ 6 程度の MinMax 探索（評価関数は 5 章のもの）を実装し自己対戦させると、そのような試合になりやすい。また、将棋などのチェスゲームに慣れている人間プレイヤーと深さ 6 の MinMax 探索をおこなう AI を対局させたところ AI 側が勝利したため、相手駒の色が観測できたとしても戦略は非自明であると考えられる。

7. 集計法の工夫

本研究では、以下のようなメタヒューリスティックな集計法を 3 つ考え、実験する。(1) は NaottiMinMax[4] で用いた集計法である。

(1) 最小最大化

各状態について 1 手目を手 a にするという条件の下で MinMax 探索をおこない評価値を計算し、その最小値を手 a の評価値にする。そして、評価値が最大の手を指す。例えば、図 3 のように探索がおこなわれた場合、手 1 の評価値は $\min(5, 3, 4) = 3$ 、手 2 の評価値は $\min(7, 8, 1) = 1$ となる。そのため、評価値が大きい手 1 が選択される。

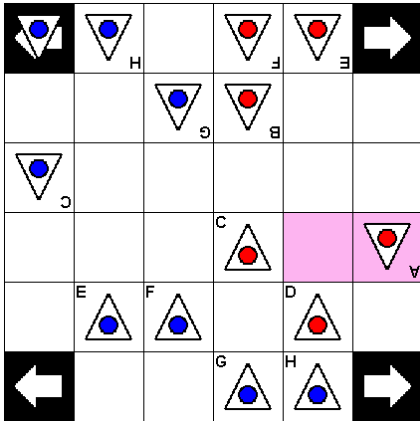


図 2 完全情報ガイスターの例

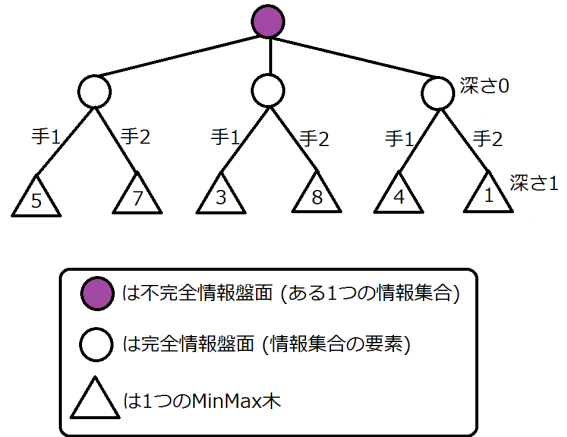


図 3 集計法のサンプルケース

(2) 平均最大化

各状態について1手目を手aにするという条件の下でMinMax探索をおこない評価値を計算し、その平均値を手aの評価値にする。そして、評価値が最大の手を指す。例えば、図3のように探索がおこなわれた場合、手1の評価値は $(5 + 3 + 4) / 3 = 4$ 、手2の評価値は $(7 + 8 + 1) / 3 = 5.333\dots$ となる。そのため、評価値が大きい手2が選択される。

(3) 最善手多数決

各状態についてMinMax探索をおこない最善手を求める。手aが最善手となる状態の個数を手aの評価値にする。そして、評価値が最大の手を指す。例えば、図3のように探索がおこなわれた場合、手1が最善手となる状態は1個、手2が最善手となる状態は2個となる。そのため、その個数が大きい手2が選択される。

8. 紫駒の応用

敵の盤上の駒を「取ったら赤に変化するが、脱出できる駒（以降、紫駒とよぶ）」としたあと通常のMinMax探索をおこない指し手を決定する。末端の評価関数は、5章で説明したものを用いる（ただし、評価関数の中では、相手の青駒の個数を「4 - 既に取った相手の青駒の個数」として考える）。探索において、既に取っている敵駒の色は分かっているものとして考える。すなわち、「取った紫駒、赤駒の個数が4個以上になるか、紫駒が脱出したら負け」として探索をおこなう。

この方針の特徴は、自分にとって最悪の状況を常に想定できることにある。例えば、深さkでMinMax探索をおこないもし必勝局面だと結論付けられれば、敵の状態によらずk手以内で必ず勝つことができる。また、ここで必勝局

面だと結論づけられなければ敵の状態によってはk手以内で勝つことができない。

なお、集計法の工夫(1)の最小最大化を用いて必勝判定をおこなうことは難しい。例えば、図4においては、どちらの状態を仮定しても手1が必勝手になるが、実際には状態が確定していないため必勝手を構成することができない。この方針の場合は、必勝局面ではないと結論付けられる。

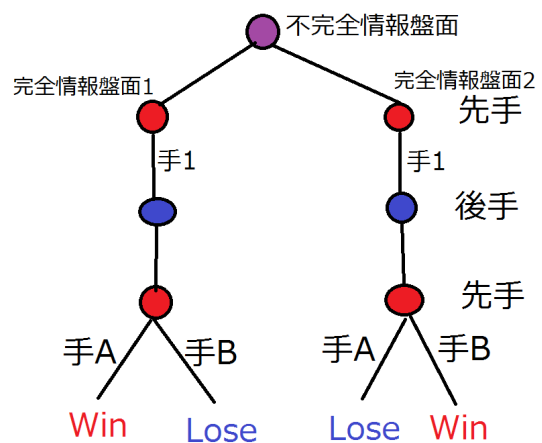


図 4 必勝局面ではない例

9. AIの実装

GAT杯に出場したNaottiMinMax[4]をベースに、5章で

提案した評価関数を加えた AI を実装し、性能評価に使う。7,8 章に基づく AI の性能評価のため、ランダム AI、猪突猛進 AI、赤猪突猛進 AI を作成する。ランダム AI は初期配置、駒の動き共にランダムなものである。ただし実装の簡単化のため、青駒が 1 手で脱出できる場合は脱出するようにしている。猪突猛進 AI は、最も敵陣の隅マスに近い青駒に近いほうの相手陣の隅マスへ近づけるものである。なお移動方向の候補が複数ある場合は、上右下左の順で優先する。また既に自分の青駒が敵陣の隅マスにいる場合は、脱出をおこなう。赤猪突猛進 AI は、最も敵陣の隅マスに近い赤駒に近いほうの敵陣の隅マスへ近づけるものである。なお移動方向の候補が複数ある場合は、上右下左の順で優先する。また既に自分の赤駒が敵陣の隅マスにいる場合は、その赤駒を右、下、左のいずれかの方向に移動させる。

10. 実験方法

5 種類の対戦実験をおこなう。実験 1,2,3,4,5 において、以下の 3 つの設定は共通とする。

- 初期局面は一様乱数により決定
- 300 手で決着がつかない場合は引き分け
- 初期配置は毎試合一様ランダムに決定

ただし、猪突猛進 AI、赤猪突猛進 AI の初期配置は図 5, 6 のようにする。猪突猛進 AI、赤猪突猛進 AI の初期配置を固定したのは、比較的良好な結果が出ている先行研究 [6] と実験条件を揃えるためである。

また、実験 1,2,3,4 においては以下の 1 つの設定も共通とする。

- 各組み合わせについて、先後入れ替えで 50 試合ずつおこなう。

10.1 実験 1

7 章に基づく AI (A, B, C)、8 章に基づく AI (D) を作成し、総当たりの対戦をおこなう。MinMax 探索の深さは A,B,C においては 5、D においては 5 または 6 とし、評価関数のパラメータは、 $weight = 0.001$ とする。

また、ここで一番強かった AI を X としたとき、X とランダム AI、X と猪突猛進 AI、X と赤猪突猛進 AI の対戦をおこない、X がこれら 3 つの単純戦略よりも強いかどうか検討する。

10.2 実験 2

完全情報ガイスターにおいて、評価関数パラメータ設定の実験をおこなう。具体的には、評価関数のパラメータ $weight = 0, 0.001, 100$ としたもの (E, F, G) を作成し、総当たりの対戦をおこなう。MinMax 探索の深さは 6 とする。

10.3 実験 3

実験 1 で一番強かった AI を X としたとき、X について

評価関数パラメータ設定の実験をおこなう。具体的には、評価関数のパラメータ $weight = 0, 0.001, 100$ としたもの (H, I, J) を作成し、総当たりの対戦をおこなう。MinMax 探索の深さは 6 とする。

10.4 実験 4

NaottiMinMax[4] は、評価関数のパラメータ $weight$ を 0 にした A と同じである。実験 3 で一番強かった AI と「NaottiMinMax」を対戦させ、どちらが強いか検討する。MinMax 探索の深さはどちらも 6 とする。

10.5 実験 5

実験 3 で一番強かった AI 同士を対戦させ、先手後手の勝率差、赤駒と青駒の動きの特徴について検討をおこなう。この実験では対戦回数を 100 とする。MinMax 探索の深さは 6 とする。

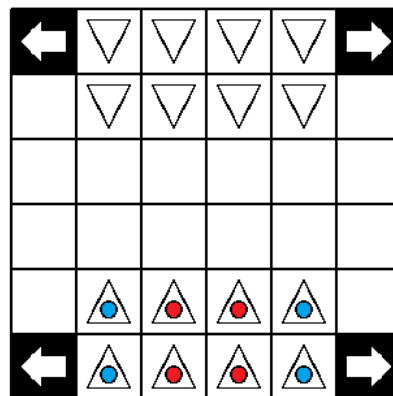


図 5 猪突猛進の初期配置

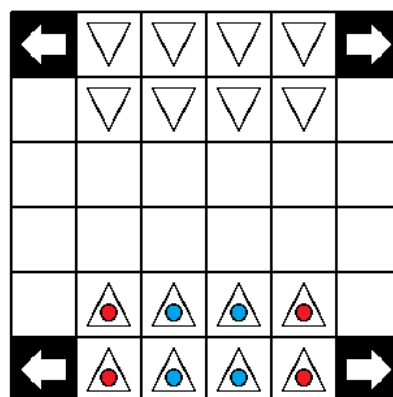


図 6 赤猪突猛進の初期配置

11. 実験結果

各実験の結果を示す。*1. 実験結果を見やすくするために 実験 1~5 で用いた AI の名称とその内容についての一覧表を表 1 に示す。表 1 の weight は、評価関数のパラメータである。

表 1 AI 名の対応

AI 名	戦略	weight
A	最小最大化	0.001
B	平均最大化	0.001
C	最善手多数決	0.001
D	紫駒	0.001
E	完全情報	0
F	完全情報	0.001
G	完全情報	100
H	紫駒	0
I	紫駒	0.001
J	紫駒	100

11.1 実験 1 の結果

実験 1 の結果を表 2,3,4,5 に示す。表 2 は探索の深さを 5 で統一したときの A,B,C,D の総あたりの結果, 表 3 は総あたりの統計情報, 表 4 は D の深さを 6 に増やした場合の対戦結果, 表 5 は D と単純戦略の対戦結果を表したものである。表 2 の各マスの 1 行目は行方向のプレイヤーを先手とした場合の勝敗を表し, 2 行目は列方向のプレイヤーを先手とした場合の勝敗を表す。

表 2 より, A と D の勝率が高いことが分かる。また, 表 3 より D(紫駒 AI) は思考時間や勝ち数などを見ると, 特に優秀であることが分かる。表 4 より D の探索の深さを 5 から 6 に増やすと, 少しではあるが勝率が上がることが分かる。表 5 より, D は猪突猛進などの単純戦略にも高い勝率を上げていることが分かる。特に, 猪突猛進と D の対戦では, 100 回中 97 回 D が勝っており, モンテカルロ法を用いた先行研究 [6] と同程度の勝率を上げていることが分かる。また, 表 2 より C の勝率がかなり低いことも分かる。

表 2 勝敗表 (行プレイヤー勝ち数-列プレイヤー勝ち数-引き分け数)

	A	B	C	D
A		31-7-12 31-11-8	46-4-0 45-5-0	11-26-13 13-29-8
B	11-31-8 7-31-12		33-17-0 34-16-0	13-25-12 17-26-7
C	5-45-0 4-46-0	16-34-0 17-33-0		10-39-1 10-39-1
D	29-13-8 26-11-13	26-17-7 25-13-12	39-10-1 39-10-1	

表 3 総あたりの統計情報

AI 名	思考時間 [sec]	勝ち数	勝ち越し回数
A	4363.07	177	2
B	1755.87	115	1
C	1590.73	62	0
D	76.59	184	3

表 4 AI-D との対戦結果

先手	後手	先手勝ち数	後手勝ち数	引き分け数
D	A	23	8	19
D	B	32	12	6
D	C	45	4	1
A	D	11	25	14
B	D	16	29	5
C	D	6	43	1

表 5 D と単純戦略の対戦結果

先手	後手	先手勝ち数	後手勝ち数	引き分け数
D	猪突猛進	47	3	0
D	赤猪突猛進	38	1	11
D	ランダム	49	1	0
猪突猛進	D	0	50	0
赤猪突猛進	D	0	31	19
ランダム	D	0	50	0

11.2 実験 2 の結果

実験 2 の結果を表 6 に示す。表 6 の各マスの 1 行目は行方向のプレイヤーを先手とした場合の勝敗を表し, 2 行目は列方向のプレイヤーを先手とした場合の勝敗を表す。

表 6 より, F の勝率は高く, G の勝率がとても低いことが分かる。よって, 完全情報ガイスターにおいては「青駒の個数」を重視すると強く, 「駒の位置の評価」を重視すると弱くなる事が分かる。

表 6 勝敗表 2 (行プレイヤー勝ち数-列プレイヤー勝ち数-引き分け数)

	E	F	G
E		3-15-32 7-23-20	26-0-24 44-0-6
F	23-7-20 15-3-32		43-0-7 45-1-4
G	0-44-6 0-26-24	1-45-4 0-43-7	

11.3 実験 3 の結果

実験 3 の結果を表 7 に示す。H,I,J はそれぞれ D(紫駒 AI) の評価関数のパラメータを weight=0, 0.001, 100 にしたものである。また, 表 7 の各マスの 1 行目は行方向のプレイヤーを先手とした場合の勝敗を表し, 2 行目は列方向のプレイヤーを先手とした場合の勝敗を表す。

表 7 より, I の勝率が高いことが分かる。よって, 不完全情報ガイスターにおいては「青駒の個数」を重視すると

*1 <https://github.com/j1211/gpw2018data>

強いことが分かる。しかし、Jも勝率が高いことが分かる。よって、不完全情報ガイスターにおいては完全情報ガイスターの場合と異なり、「駒の位置の評価」を重視しても強いことが分かる。

表 7 勝敗表 3 (行プレイヤー勝ち数-列プレイヤー勝ち数-引き分け数)

	H	I	J
H		3-27-20 0-28-22	16-17-17 22-17-11
I	28-0-22 27-3-20		19-17-14 28-15-7
J	17-22-11 17-16-17	15-28-7 17-19-14	

11.4 実験 4 の結果

実験 4 の結果を表 8 に示す。NaottiMinMax は A から駒の位置の評価を取り去った (weight=0 にした) ものであり、GPW2017 ガイスター AI 大会の優勝 AI である。

表 8 より、必勝判定を応用した AI(I) は NaottiMinMax に圧勝していることが分かる。

表 8 I と NaottiMinMax の対戦結果

先手	後手	先手勝ち	後手勝ち	引き分け
I	Naotti	13	0	37
Naotti	I	1	26	23

11.5 実験 5 の結果

実験 5 の結果を表 9, 10, 11 に示す。表 9 は I 同士を 100 回対戦させたときの勝敗、表 10 は 3 種類の勝利条件それぞれを何回満たしたかを表したものである。表 11 は特徴的な移動 (後述) をした駒が赤駒であった回数、青駒であった回数を記録したものである。ここで述べた「特徴的な移動」とは以下の 2 種類のこである。

- 「逃げ」 … 相手の駒と隣接する駒を、相手の駒と隣接しないマスに動かす移動。
- 「追いかけ」 … 相手の駒と隣接しない駒を、相手の駒と隣接するマスに動かす移動。

ただし、相手の駒を取るような移動、局面の外へ脱出するような移動については、「逃げ」「追いかけ」としてカウントしない。

表 9 より、少し先手が有利に見えるが、先手が有意に有利とは言えない。

また表 10 より、I (紫駒 AI, weight = 0.001) 同士の対戦では脱出手によって決着がつきやすいことが分かる。最悪の状況を想定するという性質上、赤駒を取って負ける試合はないようである。また、表 11 より青駒は敵駒から逃げるように動きやすく、赤駒は敵駒を追いかけるように動きやすいことが分かる。

表 9 I 同士の対戦結果

先手勝ち数	後手勝ち数	引き分け数
30	24	46

表 10 各勝ち方をした回数 (I 同士の対戦)

先後	青を取って勝利	赤を取られて勝利	青脱出で勝利
先手	7	0	23
後手	3	0	21

表 11 駒の動き方の特徴 (I 先手の場合)

特徴的な動き	赤だった回数	青だった回数
逃げ	174	439
追いかけ	749	153

12. 考察

実験 1 では、必勝判定を応用した AI(D) が強かった。D が強いのは、他の AI 比べて負ける原因が一つ少ないからだと考えられる。D は最悪の状況を想定するという性質上、赤駒を取って負けることがない。なお、C(最善手多数決) の勝率が低い理由は、相手の赤駒を取るほど相手の盤上にある青駒の比率が増え、積極的に相手の駒を取ってしまうからだと考えられる。実際、他 AI と C の対戦履歴を分析すると、赤駒を全て取られて負けているケースが目立っていた。

また実験 2, 3 より、完全情報ゲーム探索の強さがガイスターの強さと大きく異なる場合があることが分かる。特に、完全情報ガイスターで 200 戦中 1 勝しかできなかった評価関数は、通常のガイスターに適用すると 200 戦中 66 勝もしていた。その理由は、青駒を相手駒の前へ前進させたときの相手の対応を考えると分かりやすい。完全情報ガイスターの場合は、青駒だと分かっているので多くの場合自信を持って取ることができ、相手有利になりやすい。しかし通常のガイスターの場合は、相手の駒が分からない。実験 2, 3 は最悪の状況を考える AI(D) を用いておこなっているが、D は相手の駒を基本的に赤と扱うため、その駒を基本的に取らない。その結果、数で圧倒されて負けることがある。このように、情報が開示されているかどうかで相手の取る戦略が大きく変わることが、この結果の原因だと考えられる。

また実験 4 より、必勝判定を応用した AI(D) は、GPW2017 ガイスター AI 大会で優勝した AI 「NaottiMinMax[4]」よりもかなり強くなることが確認できた。よって、[4] の改良は成功したと考えられる。

また実験 5 より、先手後手の勝率の差はほとんどないと考えられる。また本研究の手法において、赤駒は相手の駒に近づきやすく、青駒は相手の駒から遠さがりやすいといった特徴が確認できた。

13. まとめ

本稿では、将棋やチェスとルールが似ている不完全情報

ゲームの一つである「ガイスター」を対象とし、完全情報ゲームの探索を用いる戦略を考えた。その結果、[5]の必勝判定手法を応用することで、GPW2017杯ガイスター大会優勝プログラム NaottiMinMax[4]に圧勝するなど、高い実力を持たせることができた。また、本研究の手法において、完全情報ゲーム探索の強さがガイスターの強さと大きく異なるケースが見られた。また本研究の手法において、赤駒は相手の駒に近づきやすく青駒は相手の駒から遠ざかりやすいこと、先手後手に有意な勝率の差が見られないことが分かった。

14. 今後の方針

先行研究や人間との対戦をすることで、本研究で作成したAIの性能を明らかにする。また、ポーカーAIなどで注目されている regret 解析などを用いた手法についても検討する。また、本研究の手法では赤駒と青駒に違いが見られたため、状態の推定をされやすいという問題点があるため、推定AIなどに強い戦略を考えていく予定である。

参考文献

- [1] David Silver, et al. Mastering the Game of Go without Human Knowledge. Nature 550, pp.354-359, 2017.
- [2] Michael Bowling, Neil Burch, Michael Johanson, Oskari Tammelin. Heads-up limit hold'em poker is solved, Science Vol. 347, Issue 6218, pp.145-149, 2015
- [3] TESAURO G. TD-Gammon, a self-teaching Backgammon program, achieves master-level play. Neural Computation 6, pp.215-219, 1994
- [4] 川上 直人, 橋本 剛: ガイスター AI の研究, 情報処理学会研究報告, Vol.2018-GI-39, No.6, pp.1-6, 2018
- [5] 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
- [6] 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
- [7] 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