Vol.2019-GI-42 No.3
2019/7/19

ガイスター AI のキーパー戦略の有効性

伊藤 雅士[1]　大久保 壮浩[2]　木谷 裕紀[2]　小野 廣隆[2]

概要：強い AI 設計はゲーム情報学における重要なテーマになりつつある．毎年開催されている GAT(Game AI Tournaments) においても, 多くのゲームプレイング AI 競技が行われ, 年々その競技レベルは上がってきている．競技種目の一つの「ガイスター」は, 全世界で親しまれている不完全情報テーブルゲームであり, その 2019 年の大会 GAT2019 では, 複数のヒューリスティクスからなるルールベース AI である StrongStrongMasashi が優勝した．本論文では, StrongStrongMasashi に実装されたいくつかのヒューリスティクスのうち, 特にキーパー戦略の有効性について考察を行う．ある条件を満たす状況においては, 適切なキーパー戦略をとることにより, 敵の必勝を防ぐことができることを示す．またその条件を満たさない場合も, 提案するキーパー戦略が勝率を上げるのに有効であることを公開されている他 AI などとの対戦シミュレーションにより示す．

The effect of keepers in Geister game playing AI

Abstract: Designing strong AI's for imperfect information games is now a central topic in the field of Game Informatics. Geister, which is popularly played by human players, is also a target imperfect information game for game AI designs. In fact, it is selected as a target game in an anual event about game AI competitions in Japan, named GAT (Game AI Tournaments). In the Geister section of GAT held in March 2019, StrongStrongMasashi won the first place. StrongStrongMasashi is a rule-based AI program, which adopts several heuristics. A distinctive strategy of StrongStrongMasashi is that it sets a diffensive position called *keeper*, which blocks goals. In this paper, we discuss the effect of the existence of keepers in Geister from two viewpoints. One is a theoretical result: we show that it is possible to lead the opponent to have no pure winning strategy by using keepers under certain situations. The other is an empirical one: a computational experiment through simulation plays such as StrongStrongMasashi versus other AI's shows that the existence of keepers is effective in terms of winning rate.

1. はじめに

近年, ゲームプレイング AI に関する技術がめざましく成長しており, 将棋や囲碁などの完全情報ゲームにおいては, トッププロ棋士の実力を超える AI が作成されている [4]. 完全情報ゲームのゲームプレイング AI の研究が進むにつれて, 不完全情報ゲームにおいてもゲームプレイング AI に関する研究が進み, ポーカーゲームにおいてはゲーム理論を用いた解析により, どのプレイヤにも長期的には負けない AI が作成されている [1]. ガイスターは 1982 年にドイツ年間ボードゲーム大賞を受賞するなど世界中で親しまれている二人不完全情報ゲームである．ガイスターは 6×6 に区切られた盤面上で 2 種類 4 個ずつの合計 8 個の駒を移動させていき, 相手の一方の種類の駒をすべて獲得するといった複数ある勝利条件のうち, 一つを実現したプレイヤが勝利するゲームである．このゲームは将棋や囲碁などと異なり, それぞれの盤上の駒は所有者が誰であるかは分かるものの, その種類が非公開の形で行われる不完全情報ゲームである．ガイスターは GAT(Game AI Tournament) の公式種目になるなど, 近年ゲーム AI に関しても注目されているもののガイスターや軍人将棋, Stratego のような駒移動型の不完全情報ゲームの解析は難しく [3], まだまだ研究の余地がある [2], [5], [6], [7], [8], [9]. 本稿では 2019 年の GAT を優勝したルールベース AI「Strong Strong Masashi」に対して対戦シミュレーションを行い, 有効なヒューリスティクスの発見を目的とする．その中でキーパー戦略が有効である局面を示し, その戦略の有効性を議論する．

[1] 名古屋大学情報文化学部
School of Informatics and Science, Nagoya University
[2] 名古屋大学大学院情報学研究科
Graduate School of Informatics, Nagoya University

2. 準備

2.1 ガイスターのルール

2.1.1 ガイスターの準備

ガイスターは, 6×6 の盤上 (図 1) で自分の駒を動かし, お互いの駒を取り合うゲームである. 各プレイヤはそれぞれ 8 個の駒をゲーム開始前に保有しており, そのうち 4 個は赤マーカーが, 残りの 4 個は青マーカーが相手プレイヤに見えないようにマーキングされている. この 8 個の駒を初期配置可能な 2×4 の領域内 (図 2) にそれぞれの色のマーカー駒を配置する. このとき, 各駒の配置は相手プレイヤに分からないように行われる.

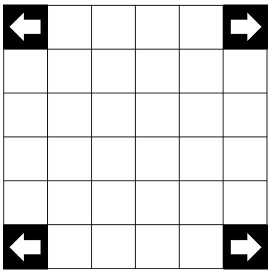
図 1 ガイスターの盤面

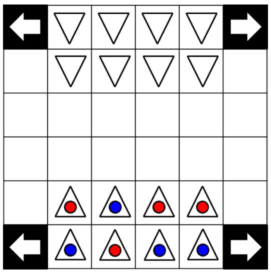
図 2 駒の初期配置

適当な方法で先手プレイヤを決め, 先手プレイヤから一手ずつ交互にゲームが行われる.

2.1.2 ゲーム中に可能な着手

各手番において, 手番プレイヤは自分の駒を一つ選び, 縦横 4 方向のいずれかの隣接領域に駒を動かす. このとき, 駒を動かした先に非手番プレイヤの駒が存在する場合は, その駒をゲームから除外する. また, 駒を動かした先に手番プレイヤの駒がある場合, その移動は行うことができず, 不正手として扱われ, 手番プレイヤの反則負けとなる. また, 盤面外への駒の移動や既にゲーム外へ除外された駒の移動も同様に不正手として扱われ, 反則負けとなる. また, 6×6 の盤面の隅にある脱出口 (図 2 に示すように, それぞれの対戦相手の初期配置に近い隅にプレイヤの脱出口が存在する.) に青マーカーがマーキングされている駒がある場合, **脱出**を宣言することができる.

駒を動かすと手番プレイヤの手番は終了し, 非手番プレイヤが手番プレイヤに変わる.

2.1.3 勝利条件

ガイスターにおいて, 以下の三つが勝利条件であり, このいずれかを先に満たしたプレイヤーが勝者となる.

1 相手の青マーカーがマーキングされている駒 (良いオバケなどとも呼ばれる. 本稿では以下**青駒**とする) をすべて取ること.
2 自分の赤マーカーがマーキングされている駒 (悪いオバケなどとも呼ばれる. 本稿では以下**赤駒**とする) がすべて相手に取られること.
3 脱出を宣言すること (自プレイヤの脱出口に自プレイヤの青駒がいる場合にのみ脱出は宣言可能).

以下では, ガイスターの盤面に対し, 図 3 のように番地を定義する. また, 駒の初期配置の番地に対し, 本論文では図 4 のように定める.

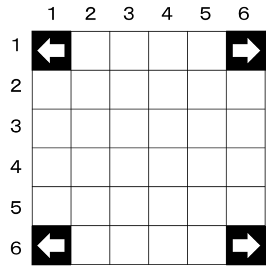

図 3 盤面上の番地

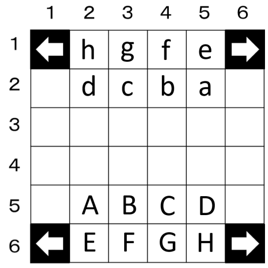

図 4 初期配置の番地

例えば, 図 2 における赤駒の位置は $(2,5),(3,6),(4,5),(5,5)$ であるが, 初期配置に対しては簡単のため, AFCD のように呼ぶ.

また, 以下ではゲーム外のあらゆる情報からゲーム内の情報を手に入れることができないとする.

3. キーパーの有効性

本研究では**キーパー**と呼ばれる相手脱出口の付近に駒を配置し, 守らせる戦略が有効であることを示す. まず, 証明に使用するキーパーアルゴリズム (Algorithm 1) を記述する.

このキーパーアルゴリズムを取っているプレイヤに関して以下の定理が成立する.

定理 1. *ゲーム開始からキーパーアルゴリズムを取っているプレイヤがキーパー駒が取られていない且つ, そのプレイヤの赤駒が 3 つゲームから除外されているとき, その対戦相手プレイヤは必勝戦略を保持しない.*

ここで必勝戦略を保持しないとは, 相手のいかなる手 (戦略) に対して 確実に勝利する方法が存在しないことを意味する. また, キーパーアルゴリズム内ではキーパー駒の色と他の駒の行動は無関係であり, 過去の行動からキーパー色の推定ができないことに留意されたい.

以下では上記のキーパーアルゴリズムを用いることによって, 条件を満たすとき必勝戦略が存在しないことを示す.

証明. 赤駒が 3 つ取られているプレイヤ (以下ではキーパープレイヤと呼ぶ) が保持する残り駒数に対して場合分け (残り駒数がかそれ以外か) を行い, そのいずれにおいてもキーパーアルゴリズムを取る限り, 相手プレイヤが必勝戦略を保持しないことを示す.

Case1:キーパープレイヤの残り駒が 2 個のとき,

Algorithm 1 キーパーアルゴリズム

1. キーパーの初期配置
初期配置として左右の脱出口に最も近い番地に青駒, 赤駒を一つずつ配置する．このとき, 左右どちらにどちらの駒を置くかは無作為且つ, 他の駒の配置とは独立に決定する．

2. 初期値の設定
相手のそれぞれの駒に初期値 0 を与える．また, 確率 $p(0 < p < 1)$ を与える (後述の通り, 赤駒とみなした駒に対して 1 を与える確率を表している.).

3. 開始後のキーパーの移動
ゲーム開始直後の手番とその次の手番において, それぞれの脱出口に最も近い駒を脱出口に移動させる．

4. 行動分岐
以降, 残り駒の駒数に従った動作を行う．残り駒が 3 つ以上であれば 5 へ, 残り駒が 2 つであれば 6 の操作を行う．

5. 残り駒 3 個以上のときの行動
脱出口にある駒以外すべてが取られるまで脱出口にある駒以外を移動させる．このとき移動先のうち, 相手駒が存在しない番地があればその移動先に移動する．

6. 残り駒 2 個のときの行動分岐
自陣のキーパーの位置に従った動作を行う．全ての自陣の脱出口にある場合は 7 へ, 片方が自陣の脱出口にある場合は 8 へ移動する．いずれでもない場合は 9 へ移動する

7. 残り 2 個, 両キーパーが脱出口のときの行動
相手駒がおかれていない番地に移動可能な駒があれば, 移動可能な駒の中から無作為に一つ選び相手駒がおかれていない番地へ移動する．選んだ駒の全ての可能な移動先に相手駒が置かれている場合, 置かれている相手駒の値が 0 である番地が存在すれば, その点を選択する．置かれている相手駒の値が 0 である番地が存在しなければ, 無作為に一つ自駒を選び, 無作為な方向へ移動を行う．移動後 6 へ戻る．

8. 残り 2 個, 片方のキーパーが脱出口のときの行動
(ア) 脱出口に相手の駒がない, または相手の赤駒が残り 2 個以上であれば脱出口にない駒を脱出口に移動させる．
(イ) 相手の赤駒残数が 1 つのとき, かつ, 脱出口に相手駒があるとき値が 1 である駒が他になければ, 確率 p で脱出口にある駒の値を 1 に更新する (このとき確率 $1 - p$ で値の更新を行わない．また, 値が 1 である駒が他に存在する場合も更新を行わない.)．その後, 脱出口の相手駒の値が 0 であればその駒を取り, 脱出口に移動する．脱出口の相手の駒の値が 1 であるとき, 脱出口に存在するキーパー駒が相手駒がおかれている番地に移動可能であれば, 移動可能な駒の中から無作為に一つ選び相手駒がおかれていない番地へ移動する．選んだ駒の全ての可能な移動先に相手駒が置かれている場合, 置かれている相手駒の値が 0 である番地が存在すれば, その点を選択する．
移動後 6 へ戻る．

9. 残り 2 個, 両キーパーが脱出口外のときの行動
(ア) いずれかの駒が脱出口に隣接していない場合, 無作為に駒を選択し, 合法手のいずれかを行う．
(イ) 両方の駒が脱出口に隣接している場合
脱出口に相手の駒がなければ, その駒を脱出口に移動する．脱出口相手駒の値が 0 であればその駒を取り, 脱出口に移動する (このとき, 値が 1 である駒は最大でも一つであるため, どちらかの駒を取ることになることに留意したい.).

このとき対戦相手 (以下では非キーパープレイヤと呼ぶ) が必勝戦略を保持しないことを背理法で示す．
非キーパープレイヤが必勝戦略を持つと仮定すると, 非キーパープレイヤは以下のうち何れかを必ず実現可能である．

勝利条件 1　キーパープレイヤの赤駒をとる (=赤駒を 4 つ取ることによってキーパープレイヤの勝利条件を満たす) ことなく, キーパープレイヤの青駒をすべて取ることができる．

勝利条件 2　非キーパープレイヤの青駒が全て取られる (=青駒が 4 つ取られることによってキーパープレイヤの勝利条件を満たす) ことなく, 非キーパープレイヤの赤駒がすべて相手に取られることができる．

勝利条件 3　必ず非キーパープレイヤの青駒が脱出口に位置し, 脱出を宣言すること．

このとき, 勝利条件 1 を満たすためには, キーパープレイヤの駒のうち, 移動した駒か脱出口にある駒を取る必要があるが, 過去の行動からキーパー色の推定ができないため, 確実に青駒をとることができない．従って矛盾が生じる．
勝利条件 2 を満たすためには, 非キーパープレイヤの赤駒が 3 つ取られた状態でキーパープレイヤが赤駒を取る必要があるが, その状態でキーパープレイヤ側が駒を取るタイミングは, 脱出口に存在する駒に対して与えた値が 0 であるときだけである．しかしながら, この事象を確実に起こすことは不可能であるため, 矛盾が生じる．
勝利条件 3 を満たすためには, 脱出口に移動する駒が青駒であり, その後その駒が取られることなく, 次のターンを迎える必要があるが, この事象を確実に起こすことは不可能であるため, 矛盾が生じる．

Case2:キーパープレイヤの残り駒が 3 個以上のとき,
以下ではこのときにキーパープレイヤの保持する駒が赤駒が残り 1 個, 青駒が残り 2 個以上であることに留意されたい．キーパープレイヤは自身の手番においてキーパー以外の駒を動かす．このとき対戦相手 (以下では非キーパープレイヤと呼ぶ) が必勝戦略を保持しないことを背理法で示す．
非キーパープレイヤが必勝戦略を持つと仮定すると, 非キーパープレイヤは以下のうち何れかを必ず実現可能である．

勝利条件 1　キーパープレイヤの赤駒をとる (=赤駒を 4 つ取ることによってキーパープレイヤの勝利条件を満たす) ことなく, キーパープレイヤの青駒をすべて取ることが必ずできる．

勝利条件 2　非キーパープレイヤの青駒が全て取られる (=青駒が 4 つ取られることによってキーパープレイヤの勝利条件を満たす) ことなく, 非キーパープレイヤの赤駒がすべて相手に取られることが必ずできる．

勝利条件 3　必ず非キーパープレイヤの青駒が脱出口を配置し, 脱出を宣言すること．

このとき, 青駒を 2 つ以下しか取っていないため, 勝利

条件 1 を 1 手で満たすためことはできない.

また, 勝利条件 2 を満たすためには赤駒が 3 つ取られた状態で赤駒が取られる必要があるが, この状況はキーパー以外のキーパープレイヤの駒が赤駒のみに囲まれているとき以外起きない. しかし, 赤駒のみで囲うためには赤駒は最低でも 2 つ必要であるが, 赤駒は 4 つしかないため, 3 つとられた状態で赤駒のみで囲むことはできない,

また, 勝利条件 3 を満たすためには, 脱出口にあるいずれかの駒を取らなければいけないが過去の行動からキーパー色の推定ができないため, 確実に青駒をとることができない. 従って矛盾が生じる.

したがって題意は示された. □

4. シミュレーションによる検証

3 節で示した定理は, キーパーを配置することにより敵の確実な勝ちを防ぐことができる場合があることを示している. しかし, このことはキーパーを準備することが勝率の向上に有効であることを, 直ちに示すものではない. なぜならばある駒にキーパーとしての役割を課すことは, 駒の動きを制限することでもあり, それ自体が勝敗に影響を及ぼしうるためである. すなわち, ガイスターのようなゲームにおいて, ある「戦略」「戦術」「戦法」の良し悪しを単体で評価することは極めて困難である. このため, 本研究ではキーパー配置の有効性の評価をシミュレーションにより行うことを考える.

この目的のため, 本研究では 2 つの実験を実施する. 実験 1 はキーパー駒の積極的な配置戦略が勝ち負けに与える影響を調べることを目的とするものである. 比較のため, キーパー駒を積極的に配置する通常の StrongStrongMasashi(以下,SSM とする）と, キーパー駒を積極的には配置しない（つまり, 脱出口に必ずしも駒を向かわせない）No Keeper SSM(以下では NK とする) を用意する. この NK は, キーパー駒に関する扱いを除けば通常の SSM と同一のルール（アルゴリズム）で行動を決定する. これら二つに対し, 2 つの既存プログラムとの対戦成績を比較することにより, キーパー配置の有効性を検証するのが実験 1 である. 既存 AI プログラムとしては多種多様なものを準備することが望ましいが, 今回は入手・準備のしやすさから代表的な AI として, ランダムプレイヤとモンテカルロ木探索ベースの西尾 AI（後述）を対戦相手として取り上げる.

実験 2 は, キーパー駒の存在とそれに適した初期配置があるかどうかを調べるためのものである. GAT2019 に出場した標準の SSM は駒の初期配置を固定していたが, SSM 自体は任意の初期配置を選ぶことができる. このため, どの初期配置が SSM のキーパー戦略に適しているかを調査することが本実験の目的である. 配置自体の影響に注目するため, 本実験では各初期配置の SSM に各初期配置の NK との対戦成績を先手後手の組合せまで考慮した上で集計し, その勝率を観察する.

4.1 対戦プログラム

以下の 4 種のプログラムを対戦に使用する.

- SSM
  GAT2019 で用いたプログラムである. 脱出口に最も近い駒を脱出口に移動させ, その後その駒はほとんど脱出口から動かない設定 (キーパー戦略) を組み込んでいる. 乱数を全く使用しない. プログラムは JAVA 8 で実装されている.
- NK
  SSM からキーパー戦略を除いたプログラムである. このため, 初期配置を自由に選ぶことができる. SSM と同様に乱数を全く使用しないため, SSM との対戦では, 同じ初期配置であれば必ず同じ結果が出力される. 同様に, プログラムは JAVA 8 で実装されている.
- ランダムプレイヤ
  連続対戦を行う場合に i 戦目のとき Seed の値を $i-1$ に設定する. 初期配置および自分の手はその Seed の値による乱数で選択する. `https://github.com/miyo/geister_server.java/` にて公開されている（2019 年 6 月 27 日閲覧）.
- 西尾 AI
  2017 年 11 月の GPW でのガイスター AI 大会に出場した西尾泰和氏によるプログラムである. GPW 及び GAT での一手の制限時間である 10 秒間モンテカルロ木探索を行う. ただし連続対戦 i 戦目の時 Seed 値を $i-1$ に設定し, 初期配置はその Seed の値による乱数で選択する. 実験において時間制約のため完全な再現性がないので注意が必要である. Python 2 系で実装されている. `https://github.com/nishio/gpcc_geister` にて公開されている（2019 年 6 月 27 日閲覧）.

4.2 実験環境

実験に用いる集計用プログラムは, JAVA 8 によって実装し, 計算機は Intel Core i7-7700HQ CPU 2.80GHz, 8GB RAM を搭載したものを使用した. また, 実験で行われる対戦は GAT2019 で使用された, サーバと 2 つのクライアント間の TCP/IP 通信を用いた環境で行った.

4.3 実験結果

本節では, 行った実験の結果を示す.

4.3.1 実験 1 の結果

表 1 に, SSM とランダムプレイヤの 100 戦の対戦結果を記す. ランダムプレイヤでは第 i 戦目の Seed 値を選ぶことができるため, これを $i-1$ とした上で先手・後手の組合

せを考慮した実験となっている．SSM は決定性プログラムであるため，この実験では再現性が保証されている．同様に表 2 に，NK とランダムプレイヤの 100 戦の対戦結果を記す．SSM, NK 共，ランダムプレイヤに対してはほとんど同程度の勝率となっている．

表 1　SSM 対ランダムプレイヤー

SSM	勝ち	負け	分け
先手	48	2	0
後手	44	6	0
計	92	8	0
引き分けを除いた勝率			92.0%

表 2　NK 対ランダムプレイヤー

NK	勝ち	負け	分け
先手	47	3	0
後手	46	4	0
計	93	7	0
引き分けを除いた勝率			93.0%

そもそも SSM が相手駒の奪取と脱出を積極的に目指すヒューリスティクスを採用しているため，ランダムプレイヤを相手にすると SSM も NK も攻め込まれる前に決着が着いてしまう傾向にある．つまり，ランダムプレイヤとの対戦では SSM, NK ともに相手駒が自陣出口に近づくことが稀であるため，キーパー戦略の有無は勝率に影響を及ぼさない結果となったと考えられる．

次に西尾 AI と SSM, NK の対戦結果をそれぞれ表 3, 4 に記す．ランダムプレイヤと同様，西尾 AI では第 i 戦目の Seed 値を選ぶことができるため，これを $i-1$ とした上で先手・後手の組合せを考慮した実験となっている．ただし，ランダムプレイヤとは異なり，西尾 AI は CPU 時間による探索打ち切りなどの仕組みのため，必ずしも実験の再現性が保証されない．　ランダムプレイヤとの対戦とは異なり，SSM, NK 共に引き分けの割合が増えている．ただし，NK に関しては後手番のときにその傾向が顕著である．

表 3　SSM 対西尾 AI

SSM	勝ち	負け	分け
先手	31	7	12
後手	28	8	14
計	59	15	26
引き分けを除いた勝率			79.7%

表 4　NK 対西尾 AI

NK	勝ち	負け	分け
先手	30	13	7
後手	33	6	11
計	63	19	18
引き分けを除いた勝率			76.8%

NK と比べると SSM は引き分けの数が多くなっている．これはキーパーによって相手の脱出を妨害し，負けにくくなっているためと考えられる．

いずれにしても，勝ち数自体は SSM, NK ともに同程度（総計するとあるいは NK がやや多い）が，引き分けを除いた勝率では SSM が 79.7%, NK が 76.8%と，SSM が少し上回っている．上述の引き分けの多寡と併せて考えると，キーパー配置の効果が表れていると言える．

4.3.2　実験 2 の結果

本節の冒頭で述べたように，本実験では各初期配置の SSM に各初期配置の NK との対戦成績を先手後手の組合せまで考慮した上で集計し，その勝率を観察する．ありうる初期配置の数は，8 つの番地のうち，どの番地を赤（あるいは青）のための 4 つとして選ぶかの場合の数となるので，

$$\binom{8}{4} = \frac{8!}{4!4!} = 70$$

ある．実験では 70 通りの SSM と 70 通りの NK を先手後手を入れ替える形で，$70 \times 70 \times 2 = 9800$ 対戦を行い，集計した．その結果を表したのが表 5 である．SSM の勝率は 47.9%と半分を切っているようにも思われるが，引き分けが 10.7%あるため，引き分けを除いた上での勝率を考えると 53.7%となり勝ち越す形となっている．この結果から，キーパー戦略は劇的ではないものの平均的にはその効果を表していると言える．

表 5　SSM(70 通り) 対 NK(70 通り)

SSM	勝ち	負け	分け
先手	2440	2008	452
後手	2256	2043	601
計	4696	4051	1053
率 [%]	47.9	41.3	10.7

次に，どの初期配置の成績が良かったかを観察する．表 6 は実験 2 で勝率の高い初期配置と低い初期配置の勝率 (引き分けあり) を表したものである．例えば，第 1 列目は SSM が CEFG の位置に赤駒を配置した初期配置となっている．表の第 1 列が初期配置における赤駒の位置を表しており，上から下に最も高いものから順に並べてある．ただし，紙面の都合上全配置の成績ではなく，最上位 4 配置，最下位 4 配置の計 8 配置の成績のみが記してある．つまり，最も勝率の高い配置が CEFG でその際の勝率が 81.4%と良い成績を達成しているのに対し，最も勝率の低い配置は，BCDG でありその勝率は 22.9%となる．この結果は，初期配置とキーパー戦略には強い関連があることを示唆するものと言える．

さて，少なくとも本シミュレーションからキーパー戦略

表 6 初期配置ごとの勝率 (引き分けあり)

初期配置	勝率 [%]
CEFG	81.4
BFGH	77.1
BCFG	77.1
BCFH	77.1
⋮	⋮
ABDH	24.3
ABDF	22.9
ACEH	22.9
BCDG	22.9

に適した初期配置が存在すると考えられるものの初期配置が 1 か所違うだけで大きく勝率が変化するとは考え難い. 実際, 表 6 を見ると, 上位 2 位, 3 位, 4 位の配置は 2 位 3 位が BFG と 3 か所が共通で, 3 位 4 位は BCF と 3 か所が共通している. このような観点から, 赤駒の大まかな配置を元に初期配置の良し悪しを分類することを考える. まず初期配置における番地を外側 4 箇所 (ADEH) と内側 4 箇所 (BCFG) に分け, そのいずれに赤駒が配置されているかの個数で分類したときの勝率を表したのが表 7 である. この結果から, 初期配置において赤駒が内側にあるほどキーパー戦略に適すると考えられる.

表 7 内側 4 箇所の赤駒の数

内側 4 箇所の赤駒の数	勝率 [%]	敗率 [%]	分率 [%]
0	31.4	55.7	12.9
1	37.5	51.7	10.7
2	47.9	40.4	11.7
3	57.5	33.7	8.8
4	77.1	17.1	5.7

表 8 前方 4 箇所の赤駒の数

前方 4 箇所の赤駒の数	勝率 [%]	敗率 [%]	分率 [%]
0	68.6	25.7	5.7
1	54.7	33.3	12.1
2	51.4	37.0	11.6
3	32.9	59.0	8.2
4	32.9	60.7	6.4

さて, 今度は初期位置における番地を前方 4 箇所 (ABCD) と後方 4 箇所 (EFGH) に分けた場合の赤駒の個数による分類と勝率の関係を表したのが表 8 である. この結果から, 初期配置において赤駒が後方にあるほどキーパー戦略に適すると考えられる. 最後に出口に置くキーパーの赤駒の数ごとの勝敗が表 9 である. この結果から, キーパーには赤 1, 青 1 を配置するキーパー戦略が最も勝率が高く, キーパー戦略に適すると考えられる.

表 9 赤のキーパー数

赤キーパー	勝率 [%]	敗率 [%]	分率 [%]
0	41.7	49.4	8.9
1	51.2	37.5	11.3
2	45.4	43.5	11.1

5. まとめ, 今後の展望

本論文では, 限られた条件において対戦相手が必勝戦略を持たないようなキーパー戦略があることを証明し, またキーパー戦略の有効性について 2 つの実験を行った. 実験の結果において, キーパー戦略は一定の有効性があることが分かったが, いかなる対戦相手に対しても有効な戦略であるというには不十分である. これを解決するために対戦相手が必勝戦略を持たない状況を拡張することが考えられる. 例えば, 自分の赤駒を 3 つ取っていない相手に対し, 必勝戦略を持たないような戦略が存在するかを考える. 現状, 各出口にキーパーが 1 つでは, 自分の赤駒を 3 つ取っていない相手に 2 つ以上の駒で攻め込まれた場合, 脱出されてしまう可能性がある. これを防ぐためにはキーパー数を拡張する (各出口に近い位置にサブキーパーを配置する.) などのさらなる工夫が必要であると考える. つまり, 相手の脱出の妨害を行うことができる最適なキーパー数とその配置を考慮することでキーパー戦略を拡張することが出来る.

また, 青駒による脱出の必勝局面探索を効率的に行うことで, 逆に相手が必勝局面を持たないように自身の戦略を考えることが出来るため, 必勝局面探索の効率化も今後の課題の一つである.

参考文献

[1] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold’em poker is solved. *Science*, 347(6218):145–149, 2015.
[2] Chen Chen and Kaneko Tomoyuki. Counterfactual regret minimization for the board game geister. In ゲームプログラミングワークショップ *2018* 論文集, 2018.
[3] Sergiu Redeca and Adrian Groza. Designing agents for the stratego game. 2018.
[4] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.
[5] Mishio Takenori and Kotani Yoshiyuki. Estimation algorithm upp for imperfect information in games and application for geister. 研究報告ゲーム情報学（*GI*）, 2014.
[6] 三塩武徳, 藤田桂英, et al. Upp による駒価値評価関数に基づいた negogeister ai. 研究報告ゲーム情報学 *(GI)*, 2015(7):1–6, 2015.
[7] 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. In ゲームプログラミングワークショップ *2018* 論文集, 2018.
[8] 石井 岳史，川上 直人，橋本 剛，池田 心. 不完全情報ゲーム『ガイスター』における 2 種の詰め問題の提案と考察.

Technical Report 1, 2019.

[9] 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. In *GAT2018* 論文集, volume 2018, pages 13–16, 2018.