

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

末續鴻輝^{1,a)} 織田祐輔^{2,b)}

概要: ガイスターは、二人で行う不完全情報ゲームである。相手の駒の種類がわからない中で、その正体を推測し、かつ自分の駒の正体を推測されないようにプレイを行う必要があり、単純なゲームでありながら複雑な心理戦の様相を呈する。そのため人工知能研究の対象としても注目され、本年より GAT の種目の一つに、ガイスターの人工知能対戦が採用された。そこで本研究では、ガイスター AI 大会に出場させる人工知能の開発を行った。近年の人工知能に関する話題は、機械学習に関するものが注目されており、ルールベースに開発を行ったものはあまり話題に上らないが、本研究においては、機械学習を用いずに強くできるかを検証するため、機械学習は用いずに開発を行ったので、その行動アルゴリズムをここに紹介する。

Developping geister algorithms without machine learning

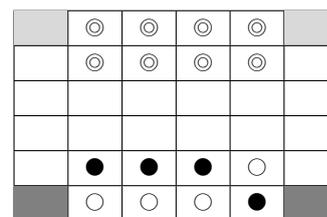
KOKI SUETSUGU^{1,a)} YUSUKE ORITA^{2,b)}

Abstract: Geister is a two player imperfect information game. Players need to guess the color of each piece of the opponent, so they also need to bluff. Since geister AI tournament will start as an event of Game AI Tournament, we developed a geister AI. Although machine learning is paid attention to in these days, this AI is developed without machine learning in order to verify whether it can be made strong without using it. We show the algorithms here.

1. はじめに

ガイスター (geister) は、2 種類の駒を自他ともに 4 個ずつ用いて行う 2 人不完全情報ゲームである。2 種類の駒は、正面からは見分けがつかないが、後ろから見ると赤いマーカーのついている駒 (以下、赤駒と呼ぶ)、青いマーカーのついている駒 (以下、青駒と呼ぶ) がある。縦横 6 マス、計 36 マスの盤面にプレイヤーは向い合い、手前中心の縦 2 マス横 4 マスの 8 マス領域に、赤駒青駒 4 個ずつを相手に色の見えないように配置する。初期配置の例は、図 1 を参照のこと。

各手番において、プレイヤーは自分の駒の一つを選び、縦横 4 方向のいずれかに一マスだけ動かす。その際、動かし



◎: 相手プレイヤー駒
●: 自分プレイヤー赤駒
○: 自分プレイヤー青駒
薄い灰色: 自分プレイヤー出口
濃い灰色: 相手プレイヤー出口

図 1 ガイスターの初期配置の例

た先に相手の駒がいた場合は、その駒を取る。また、自分の青駒が相手から見て最下段の両端 (このマスを“出口”と呼ぶ) いずれかにいる場合は、その駒を盤の外に出すこともできる。これは、本ゲームの勝利条件の一つである。ほかに勝利条件は二つ存在し、一つは相手の青駒を全て取ってしまうこと、もう一つは自分の赤駒を全て相手に取られることである。従って、プレイの最中においては、いかに

¹ 京都大学大学院人間・環境学研究科
Graduate School of Human and Environmental Studies, Kyoto University

² 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University

a) suetsugu.koki@gmail.com

b) orita.yusuke@lab.ime.cmc.osaka-u.ac.jp

相手の駒の色を読み、また自らの駒の色を誤認させるかが、勝利に向けて大きなカギとなる。

本ゲームは単純なルールでありながら、一方では戦術的な多様性をも含んでおり、ゲーム人工知能の研究対象として興味深い。これまでも、[1], [2] などの先行研究が知られており、また、2017年には情報処理学会ゲーム情報学研究集会のゲームプログラミングワークショップでガイスター AI 大会が行われた。こうした動きを受け、本研究ではガイスターの行動アルゴリズムを実装し、GAT2018に参加した。以下では、その行動アルゴリズムについて紹介する。

2. アルゴリズム

本研究で実装するガイスターの行動アルゴリズムは以下の通りである。

- (1) 初期配置+
ゲームの開始時に初期配置を行う。以下、各着手において下の 2-7 を行う。
- (2) 敵の駒を評価する。
- (3) 必勝手の検索を行い、発見した場合は採用する。
- (4) 必勝手を発見せず、かつ自分の青駒が残り一つしかない場合、その駒が取られないようにする。
- (5) そのような状況でもなく、敵の駒と隣接している場合、青らしいと思える駒があれば取る。
- (6) それもできない場合、キーパーと呼ばれる駒の位置調整を行う。
- (7) それもできない場合、戦型の前進を行う。
以下では、それぞれの行動について詳細を述べる。

2.1 初期配置

初期配置は、いくつかの配置のみを確率的に選択する。図 2-6 に配置と確率を示した。いずれも○を青駒、●を赤駒とする。合計が 50% になっているのは、左右対称の配置も同じ確率で選択するためである。主に前方に赤駒をまとめているのは、序盤の方が駒がぶつかって取られやすいと考えられるためである。

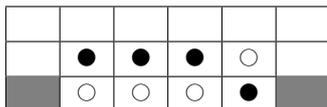


図 2 採用率：25%

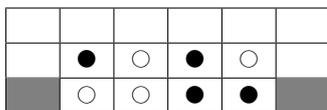


図 3 採用率：6.25%

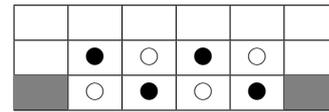


図 4 採用率：6.25%

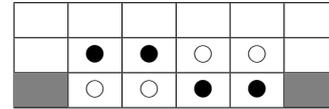


図 5 採用率：6.25%

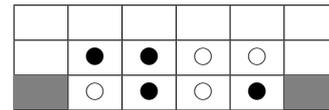


図 6 採用率：6.25%

2.2 敵の駒の評価

敵の駒それぞれについて、“青らしさ”を示す変数を用意し、手番ごとに更新する。変数の初期値は 0 であり、以下に敵の振る舞いとそれに対する変動の内容を表に示す。変数の値が正であるほど青らしく、負であるほど赤らしい。なお、ここで言う接敵数とは、上下左右のうち相手の駒と接している方角の数のことである。

敵の振る舞い	もとの値に対する変数の更新
前進し、接敵数が増えたが、自分から見て端の 1 段目または 2 段目に来た	+2.5
前進し、接敵数が増えた	-1.5
横に動き、接敵数が増えた	-1.0
前に移動して、接敵数が 0 になった	+4.0
それ以外の移動で、接敵数が 0 になった	+1.5
接敵数が 0 のまま変わらず、自分から見て 1 段目または 2 段目に来た	+10.0
接敵数が 1 以上であるのに、動かなかった	-1.2

表 1 敵の駒の色の評価

0.4	0.1	0.0	0.0	0.1	0.4
0.4	0.1	0.0	0.0	0.1	0.4
0.4	0.1	0.0	0.0	0.1	0.4
0.4	0.1	0.0	0.0	0.1	0.4
1.4	1.1	1.0	1.0	1.1	1.4
5.4	5.1	5.0	5.0	5.1	5.4

図 7 位置によって追加される青らしさ

また、この他に、もし青であったのならば以下でいう必勝手順 1. を実行できたにも関わらず、それを行わなかった場合は変数の値を -1000 とし、赤であったとみなすこととする。

2.3 必勝手の検索

ガイスターは不完全情報ゲームであるが、状況によっては、相手がどのように行動したとしても、また、相手の駒の内訳がどのようであっても、適切な着手を行うことによって必ず勝てる局面というものが存在する。以下に、そのような局面の例をいくつか紹介する。

1. 自分の青駒が出口からの距離が十分に短い場合、具体的には自らの駒と出口が図8のような位置関係である場合は、さらに3手かけることで必ず出口から青駒を出すことができる。このように、自分の青駒が出口から距離 k の位置にあり、その出口から距離 k の範囲に敵駒がない場合は、 $k+1$ 手かけることで必ず出口から青駒を出すことができる。ただし、相手にも出口から出る手が存在するかもしれないので、そのことについては別途処理を行う(後述)。

—	—	—	*	*	*
—	—	*	*	*	*
○	*	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*

図 8 必勝局面の例。—は駒なし、*は任意の状態、○は自分の青駒

2. 図9のような局面についても、左端の駒を前進させることで勝つことが可能である。相手には少なくとも二つ以上の赤駒が残っているため、右側にある自分の駒が相手の駒を取ってから、自分の青駒が出口に行くことができる。かと言って、相手の駒が隣接している自分の駒を取ってきても、守りがなくなるため自分の青駒を出口に向かわせることができる。

—	◎	☆	—	*	*
—	—	—	*	*	*
○	—	*	*	*	*
—	*	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*

図 9 必勝局面の例。—は駒なし、*は任意の状態、○は自分の青駒、☆は自分の駒で色は問わない、◎は敵の駒、ただし敵は残り赤二つ以上

本研究では、ほかにどのような必勝形があるか考察し、やはり同様に勝つことができるパターンである以下の図10, 11のパターンも合わせて必勝形テーブルにまとめ、着手選択アルゴリズムにおいては、一致している形があるかを調べた。

また、必勝形から勝利までの必要手数を数えた。これは、実際に必勝手順を採用しようとしても、相手が先に自陣の出口から出てしまう可能性があるためである。よって、相手のすべての駒について、青だった場合の出口までの手数

—	◎	●	*	*	*
—	—	—	*	*	*
○	—	*	*	*	*
—	*	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*

図 10 必勝局面の例。—は駒なし、*は任意の状態、○は自分の青駒、●は自分の赤駒、◎は敵の駒、ただし敵は残り赤二つ以上で、自分の赤駒は残り一つとする

—	●	*	*	*	*
—	—	*	*	*	*
○	*	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*
*	*	*	*	*	*

図 11 必勝局面の例。—は駒なし、*は任意の状態、○は自分の青駒、●は自分の赤駒、ただし敵は残り赤二つ以上で、自分の赤駒は残り一つとする

を検索し、その結果、自分の必勝手数の方が短かった場合は確実に勝つことができるので、その手を採用することとした。

2.4 自駒の保護

自分の青駒が残り一つしかなく、かつその駒が敵と隣接しているならば、安全なところへ動かせるならばその駒を動かす。

2.5 相手駒を取る

自分の駒が相手の駒と隣接している場合、その駒が青駒である可能性が高いならばその駒を取る。具体的に、残りの駒数と、その駒の中で何番目以内に青らしければ取るかを、以下のように表で示しておく。青らしさの順位は、先に述べた評価を元に順位づけるものとする。

青\赤	1	2	3	4
1	1	1	2	3
2	1	2	3	3
3	2	3	4	5
4	3	4	6	7

表 2 残りの赤、青駒の数に対して、全体の何番目に青らしい駒まで取るか

条件に合致する敵の駒が複数ある場合は最も青らしさの高いものを取ることにする。

2.6 キーパーの位置調整

初期配置において、相手プレイヤー出口と隣接している二つの駒をキーパーと呼ぶこととする。初期配置のアルゴリズムにより、必ず一つは赤駒、もう一つは青駒となる。キーパーは相手の青駒が攻め込んできた場合に守る働き

をするため、初期配置の位置から動いていた場合はそれを元に戻す。ただし、残りの自駒が三つ以下になった時点でキーパーはその役割を放棄し、他の駒と同様に動く。

2.7 戦型の前進

キーパー以外の駒において、最も後ろにある駒を一步前進させる。ただし、六分の一の確率で、その駒を前進ではなく横に動かす。

3. まとめと今後の課題

本研究ではガイスターのルールベース人工知能を作成し、ガイスター大会に出場させた。現在の段階では、教師あり学習や、ニューラルネットワークを用いた学習などの既存手法は利用していない。今後の展開としては、複雑な必勝形を見つけるアルゴリズムを開発し、必勝形テーブルを充実させる、現在の着手アルゴリズムについて、敵の駒の評価に関するパラメータを機械学習を用いて改善することができるか考える、あるいは、完全に別の、機械学習ベースの人工知能を構築するなど、いくつかの開発方法が考えられる。GAT2018の結果も鑑みながら、さらなるガイスター AI の発展を目指したい。

参考文献

- [1] 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
- [2] 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