Vol.2023-GI-49 No.10
2023/3/17

深層強化学習を用いた多様な戦略を持つガイスタープレイヤーの作成への試み

新堀 和紀[1,a)]　横山 大作[2,b)]

概要：本研究では, 強さを保ったまま多様な戦略をとることができるガイスタープレイヤーを作成することを目指し, 学習結果に多様性をもたらすような深層強化学習の実現を試みる. 深層強化学習では各コンピュータプレイヤーを自己対局により学習させるが, 初期配置に制約を与えて自己対局を行うことで, 多様性のある戦略を学習させられるか, を検証する. 戦略の多様性の指標としては, 指し手の一致率, 勝敗が決した時に満たされた勝利条件の割合を用いることとする. 提案手法と, 多用な戦略を学習する目的を持つ既存手法の Map-Elites によるガイスターのコンピュータプレイヤーを構築し, ランダムプレイヤーとの対戦を行い, 勝率と戦略の多様性の両面から評価を行ったところ, 提案手法はランダムな合法手を指すプレイヤーに対して最低でも 0.81 の勝率が得られ, 作成したプレイヤーの 78%が勝率 0.9 を越えたが,MAP-Elite では最大でも 0.66 となった. また,4 つのプレイヤーが同一局面で同じ手を指す確率は, 提案手法は 0.23 であり,MAP-Elite では Map-Elites では 0.017 であった. このことから提案手法では,Map-Elites より強く,Map-Elites には及ばない程度ではあるが, 多様な戦略を持つプレイヤーを作成できることがわかった.

1. はじめに

近年, ゲームをコンピュータプレイヤーにプレイさせる研究は盛んである. これは, 結果が明確であるため, 手法や技術の評価が行いやすい為である. 将棋や囲碁のような完全情報ゲームでは, 既に人間の実力を上回る性能をもつプレイヤーが多く作成されている. 本研究で題材とするガイスターは不完全情報ゲームであり, 相手には見えない駒を使用して行うゲームである. 自分の駒の正体を隠すことや, 相手の駒の正体を予測することが重要である. しかしながら, 人間は相手プレイヤーの戦略を見抜くことが出来てしまう可能性がある. このような事は, 戦略の欠点を突いたハメ手が可能になる可能性があり, さらに, 人間がコンピュータプレイヤーと対戦するときの楽しさを減少させてしまう. また, 偏った戦略を相手に人間が練習対局を行うと, 人間プレイヤーの上達をうまくサポートできない恐れもある. このため, ガイスターのコンピュータプレイヤーには, 多様な戦略を持つことが求められる.

ガイスターは有利になる可能性が低い合法手が存在するゲームであり, そのような手を選ばずに, 多様な戦略を持つコンピュータプレイヤーを作成する必要がある.Map-Elites など, 多様性を得るために提案されてきた従来手法では, 近年急速に発達してきた強化学習を利用したプレイヤーほどの強さが得られないことが予想される. そこで, 本研究では, 一定の強さを持ち, 多用な戦略を持つコンピュータプレイヤーを実現することを目標とし, 強化学習を利用できるプレイヤー構築手法を提案する. 提案手法は, ある程度強化学習を行った後に, ガイスターの駒の初期配置に制約を与えて追加学習を行うことで, 多用な戦略を学習させることを目指すものである. ランダムプレイヤーとの対戦による Map-Elites との比較実験を行ったところ, 提案手法は勝率 0.9 を越えるようなプレイヤーをおおむね学習できており, これは Map-Elites の勝率 0.65 を上回った. 一方, 多様性については, 提案手法はある程度の能力にとどまった.

本論文の構成は以下の通りである. 第 2 章では研究対象ゲームであるガイスターの紹介と関連研究について述べる. 第 3 章では提案手法のアルゴリズムを, 第 4 章では比較対象とする既存手法のアルゴリズムを述べる. 第 5 章で評価実験と結果を述べ, 第 6 章で考察を行い, 第 7 章でまとめを行う.

[1] 明治大学大学院
[2] 明治大学
[a)] ce225010@meiji.ac.jp
[b)] dyokoyama@meiji.ac.jp

2. 関連研究

2.1 ガイスター

ガイスターは,2 人のプレイヤーが相手のプレイヤーからどちらの種類か分からない 2 種類の駒を 6 × 6 の盤面の上で交互に動かしながら対戦するボードゲームである.2 種類の駒は赤駒 (悪いお化け) と青駒 (良いお化け) の 2 種類であり, どちらの駒も縦か横に 1 マス進む事が出来る. 移動先が盤外の場合, 脱出口以外には進めず, 敵の駒がいる場合は取ることが出来る. このとき取った駒の正体は見ることが出来る. 勝利条件は 3 種類あり, 自分の青駒を敵陣置くにある脱出口まで進める, 相手の青駒をすべて全てとる, 自分の赤駒を全て取らせるのいずれかを満たせば勝利である. 駒の初期配置は, 盤面の手前側中央 2 × 4 のマスに自由に 8 つの駒を配置する. また, 自分の青駒を進めると勝利になる脱出口は, 相手側, つまり自分から見て奥側の両端のマスに隣接した位置にある. プレイヤーは相手の駒の正体を推察し, 自分の駒の正体を隠しながら, 勝利条件を満たすことを目標としてプレーを行う.

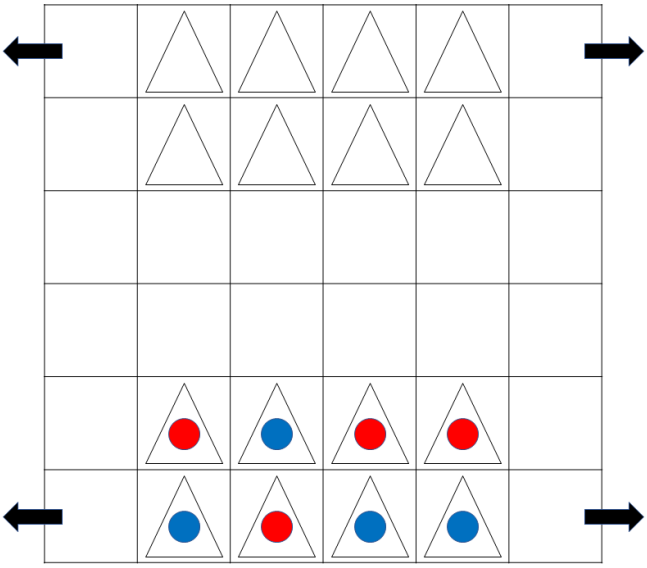

図 1 初期配置例

ガイスターはただ正体を隠して騙しあいを行うゲームではなく, 有利になる指し手が存在することが分かっている. ガイスター AI のキーパー戦略の有効性 [1] では, 自陣の脱出口を守るキーパーと呼ばれる駒を配置する戦略の有効性が述べられている. このように, 局面によっては有利になる手や不利になる手が存在することが分かっている.

2.2 ガイスターのプレイヤー

ガイスターは不完全情報ゲームであるため, 完全情報ゲームとまったく同じような手法でプレイヤーを作成することは難しい. このため, これを解決するための研究が行われてきた. 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 [2] では, 探索を行うことでコンピュータプレイヤーを作成している. この研究は, 本研究と違い強いプレイヤーを作成することを趣旨としているが, 本研究で作成した方策をもとに, 探索を行うことが可能であるため, 関連した研究である. 強化学習を用いたガイスター AI の作成に関する関連研究として, 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 [3] や, 深層強化学習を用いたガイスター AI の構築 [4] がある. これらの研究では, ランダムな合法手を指すプレイヤーよりは強いプレイヤーが出来たが, 依然としてより強いプレイヤーを作成する余地が残っている. また, 学習に際して戦略の多様性を考慮していない点でも, 本研究とは趣旨が異なる.

2.3 多様な戦略の実現

多様な戦略を実現する既存手法として,Map-Elites がある. これは, 遺伝的アルゴリズムや遺伝的プログラミングにおいて, 多様な戦略を保持しながら解空間を探索する手法である. このため, 大局的な解の探索を行うことができるため, 効率的に解を探索できる手法である. 近年では,Map-Elites における被覆率を指標とした局所的探索手法 [6] のように局所的な探索効率の上昇を目指す研究も行われている. 今回の研究では, 多様性のある集団を作るだけでなく, 一定の強さを担保することも目的であり,Map-Elites ではある程度強いプレイヤーを作成するまでに時間がかかってしまう事が予想されるため, 深層強化学習を用いる事にした. そのため,Map-Elites を対照手法として比較することにした.

また, ガイスターでは進化的計算を行っている研究に, モンテカルロ木探索ガイスターにおける遺伝的プログラミングを用いたプレイアウト方策の作成 [5] がある. この研究では, 様々な関数を作成し, その関数を用いて作成されたプログラムをもとに着手を行うガイスタープレイヤーを作成している. 対照実験では, この研究を参考に評価関数の作成を行った.

3. 提案手法

3.1 多様性の実現方針

本研究では, 一定以上の強さを保ったまま複数のコンピュータプレイヤーを作成することを目標としている. このため, 提案手法では事前に, 通常の自己対局によって深層強化学習を行い, このプレイヤーを初期配置の制約を掛けた自己対局によって学習を行うことにより, 学習に用いる棋譜の性質を変えることにより, 多様な戦略プレイヤーを作成する.

3.2 事前学習の学習サイクル

本研究ではまず, 畳込みニューラルネットワークを用い, 初期配置の決定とその後の着手に関して, 局面を入力とし, 合法手のうちどの手をどのくらいの確率で指すかを出力す

るネットワークを作成する. このネットワークの出力に従ってガイスターをプレイヤーするエージェントを作成する.

その後, ランダムに合法手を指すプレイヤーに 10000 局自己対局をさせ, その結果できた棋譜を用いてエージェントを学習させる. 学習は方策勾配法を用いて学習し, エージェントがある局面での合法手の中で, どの手をどのくらいの確率で指すかを更新する. この後学習を行った後のエージェントで 10000 局自己対局を行う. この間, 今までに生成した棋譜を使い再び, 最低 1 回ネットワークを更新し続ける. この最初の一回目の更新を 1epoch 目とする. その後 10000 局終了時に学習に用いるエージェントのネットワークを, 最新の重みに更新する. 以後, 対局, 学習, 更新を繰り返す. また, この対局, 学習, 重みの更新の 1 回を 1epoch とする.

これを事前学習とし 250epoch 行う. この後, 事前学習のニューラルネットワークの重みを初期値とし, 駒の初期配置に制約を付けたエージェントを 3 種類と通常のエージェントの計 4 種類を作成する. ここから, 各エージェントの学習に用いる棋譜の対戦相手の比率を固定し, 独立して同様の手順で学習を進めていく.

3.3 提案手法の学習サイクル

提案手法では事前学習と同様に 10000 局毎の学習サイクルで学習を進めていく. また, 提案手法と同じように,. 学習サイクルは以下の通りである.

step1: 棋譜の作成

初期配置の制約があるエージェントは, 学習用の対局時には, 一部の初期配置を行えないように制限されるが, 棋譜を用いた学習において, 初期配置の学習も行うようにする. このとき, 初期配置に制約を付けたエージェントの学習を自己対局を 50%, 初期配置の制約のないエージェントとの対局を 50%の割合で行い学習を進める. このとき, 初期配置の制約のないエージェントは事前学習終了時点のニューラルネットワークの重みを用いたエージェントである.

初期配置の制約のないエージェントは事前学習を継続する形で, 自己対局を 100%で実験を進める.

step2: 学習

以上の手順で作成された棋譜を用いてニューラルネットワークの重みの更新を行う. ニューラルネットワークの重みが更新されて以降は, 新たなニューラルネットワークの重みを用いて自己対局を行う.

また, 使用するネットワークは畳込みニューラルネットワークと LSTM を用いた構成となっており, 各手の着手確率と現局面の価値を出力する. このネットワークからの出力である, 方策と局面の価値が学習の対象である.

これらの学習サイクルを 1epoch とし, 事前学習が完了したプレイヤーを 0epoch としてカウントする.

3.4 初期配置の制約

本実験では, 初期配置の制約を行うエージェントを 3 つ作成する. 各エージェントは, 初期配置の内相手陣の側に配置された青駒の個数がそれぞれ 0 個か 1 個,2 個,3 個か 4 個に制限される. 各エージェントはそれぞれ, 制限を満たすような初期配置以外は行えないような状態で対局を行う.

4. 対照手法

本研究では一定の強さを持ち, 多様な戦略をもつコンピュータプレイヤーの作成を目的としていたが, 多様な戦略をもつ集団を作成する手法として,Map-Elites が挙げられる. この手法は遺伝的アルゴリズムをベースに, 解の空間を分割し, 分割された空間内の最善の個体を保持することで, 集団の多様性を保つものである. このことで, 解の探索効率が上がるというアルゴリズムであるが, 同時に多様なエージェントを作成できるアルゴリズムでもある. 今回は勝利条件の多様さを目指した条件設定とした. 具体的には, 個体数 300 とし, 勝利条件割合に基づいて, 各 5%区切りにした空間を用い, 実験を行った.

この手法で作られたエージェントと, 提案手法で作成されたエージェントを比較することで, 提案手法の性能を評価する.

4.1 Map-Elites での学習

この実験では,Map-Elites を用いて局面の評価値を学習することで, 複数のコンピュータプレイヤーの集団を作成する. 各コンピュータプレイヤーは, 遺伝子と呼ばれるパラメーターを持ち, そのパラメーターに従いがイスターの局面の評価値を決定する. 最初に, ランダムな遺伝子を持つプレイヤーを 300 個体生成する. 各プレイヤーは, 各着手を行うとき, すべての合法手に対し, その手を指した次の局面の局面評価値を求め, それが最大の手を着手する. ただし, 初期局面ではランダムに駒を置くこととする. これを第 1 世代とする. このようにできた個体の集団を総当たりで, 同一対戦カードにつき先後一局ずつ対戦させ, 各試合に勝てば勝ち点 2, 引き分けたら勝ち点 1 として評価する. このときに, 勝ったプレイヤーには三つある勝利条件のうち, どの条件を満たして勝利したかを記録しておく. 総当たりによる評価が終了したのち, 各プレイヤーの勝利条件の比率をもとに, プレイヤーを分類する. 具体的には各プレイヤーの勝利ゲームに占める, 勝利条件の割合を,3 つの勝利条件それぞれに関して,5%ごとにグリッドを区切り, 分類する. この後, 同じ分類に入ったプレイヤーのうち最も勝ち点の高かったプレイヤーを残す. これをグリッドの占有と呼ぶ. 最後に, グリッドを占有したプレイヤーで交叉を行う. 交叉では, ランダムな 2 個体を選択し, 選択した個体の遺伝子をも

とに新たな個体の遺伝子を作成する. このとき,1%の遺伝子に関しては, ランダムなデータに差し替えることとする. ここで新たに作成された個体の集団を第 2 世代とする. この後, 第二世代も同様の評価, 交叉, のサイクルを重ね, 学習を進める. このようにしてできたプレイヤー集団と, 提案手法を比較する.

4.2 遺伝子の設定

実験概要で述べた通り, 今回は Map-Elites でガイスターの局面評価値を学習する. 局面評価値は, 以下の点数の和である.

- 自分の赤駒の数に基づく点数
- 自分の青駒の数に基づく点数
- 敵の赤駒の数に基づく点数
- 敵の青駒の数に基づく点数
- ゴールに最も近い自分の赤駒のゴールまでの距離に基づく点数
- ゴールに最も近い自分の青駒のゴールまでの距離に基づく点数
- ゴールに関して, 自分の駒が 1 つ以上いるか, 相手の駒が 1 つ以上いるかに基づく点数
- 敵が隣にいる自分の赤駒があるかどうかに基づく点数
- 敵が隣にいる自分の青駒があるかどうかに基づく点数

これらの評価項目は, モンテカルロ木探索ガイスターにおける遺伝的アルゴリズムを用いたプレイアウト方策の作成 [5] を参考に作成した. この研究では, 遺伝的プログラミングでガイスターのプレイヤーを作成しているが, そこで局面の情報を利用するときに, 局面がある条件を満たしているか判定する関数を用いて情報を取り出していた. そこで, 効率よく評価値を作成するためにこの研究を参考に評価関数と遺伝子の設定を行った.

5. 評価

5.1 多様性と強さの評価基準

多様性の基準として, 以下の基準で評価を行う.

- 事前学習プレイヤーとランダムな合法手を指すプレイヤーの対局中の局面セットを用意し, 手の一致率を調べる
- 決着時に満たされた勝利条件の実現確率を調べる

また, 勝利条件は以下の通りである.

- 赤駒を全て取らせる
- 青駒を全て取る
- 脱出口から青駒を 1 つ脱出させる

を採用する. ただし, 両プレイヤーの指し手の合計数が 200 を越えた場合引き分けとする. また, 強さの評価基準としてはランダムな合法手を指すプレイヤーへの勝率を採用する.

5.2 実験上の特殊ルール

ガイスター本来のルールでは引き分けはないが, 永久に試合が終わらないことを防ぐため, 学習, 評価ともに両プレイヤーの指し手の合計数が 200 を越えた場合引き分けとする.

5.3 実験環境

エージェントの作成には以下の環境を用いた.

- OS:Ubuntu18.04.5LST
- CPU: AMD EPYC 7302 16-core (物理コア 16 個, 論理コア 32 個)
- GPU: Geforce RTX 3090 8 個
- Docker: version20.10.5
- Docker イメージ: nvidia/cuda:11.2.1-base-ubuntu20.04

5.4 評価 1 勝率の測定

今回作成したエージェトと比較手法のエージェントを, ランダムな合法手を指すエージェントと対戦を先手後手を入れ替え各 50 局, 計 100 局行い勝率の測定を行う. 以後, 実験では, 学習済みエージェントは, 着手確率の最も高い手を着手する. 提案手法に関しては 10epoch 毎のエージェントの評価を行い.Map-Elites は同時に作成されるエージェントが多いので, グリッドを占有したプレイヤーのみ評価を行い, 学習に使用した試合数が, 提案手法の事前学習を超える 28 世代目と提案手法の 50epoch 目を超える 51 世代目,100epoch 目を超える 74 世代目,150epoch 目を超える 96 世代目に関して評価を行う.

5.4.1 提案手法の勝率の測定結果

作成した各エージェントの勝率の推移は図 2 のようになった.

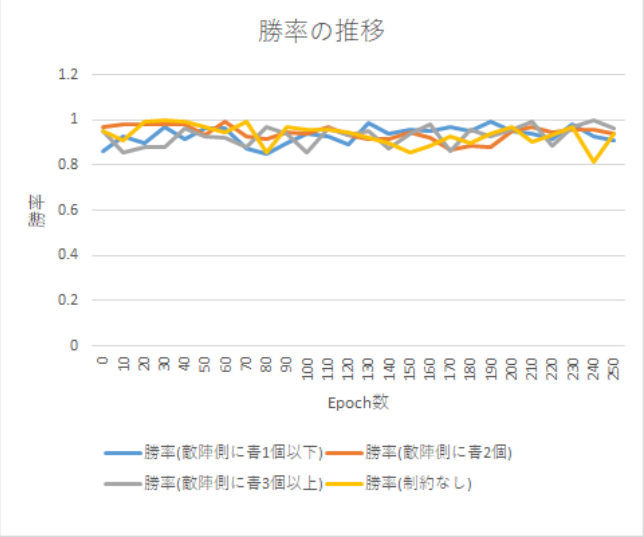

図 2 勝率の推移（提案手法）

最も勝率の低いプレイヤーは制約なしの 240epoch で 79

勝 18 敗 3 分け, 勝率は 0.81 であり, 最も勝率の高いプレイヤーは 30epoch で 100 勝し, 勝率は 1 であった. また, 作成した 104 エージェントの内,81 エージェントが勝率 0.9 を越え,37 エージェントが勝率 0.95 を越えた. このことから, 当初の目的の内, 最低限の強さを持つコンピュータプレイヤーの作成には成功したと考えられる.

5.4.2 Map-Elites の勝率の評価

Map-Elites の勝率は図 3 のようになった. グラフの縦軸が勝率, 横軸が学習時のリーグ戦内の成績が, 占有した個体の中で何位だったかを示している. また, 表 1 に各世代の勝率のうち, 最小と最大のものを抜粋した.

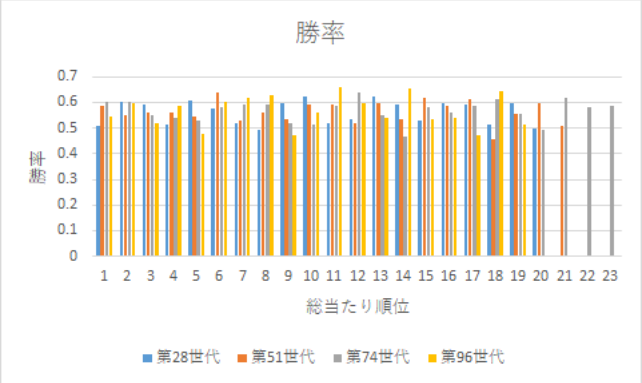

図 3 勝率

表 1 Map-Elites の勝率

世代数	最小値	最大値
28	0.49	0.63
51	0.46	0.64
74	0.47	0.64
96	0.47	0.66

このことから, 勝率に関しては提案手法の方が良い結果を残している. また, リーグ内の順位とランダムな合法手を指すプレイヤーとの対戦の勝率に, 関連性は見られなかった.

5.5 評価 2 指し手一致率の測定

ランダムな合法手を指すと事前学習時のエージェントの対局の棋譜から局面を抜粋し, 同一の局面で同一の手を指すエージェントの割合を調べる. 棋譜は事前学習時のエージェントが先手の試合と後手の試合を 50 局ずつ行った試合の棋譜を用いる. この棋譜の初期配置を行う局面を含む全ての局面から, 事前学習のエージェントの手番の局面を用いた場合と, ランダムな合法手を指すエージェントの手番の局面を用いた場合, そして両方のプレイヤーの手番の局面を用いた場合で実験を行った. エージェントの数が多いため, 各世代終了時の占有をしている個体のうち, 提案手法と同数の 4 個体を総当たりの成績が良い順に抜き出して評価を行う.

5.5.1 提案手法の指し手一致率の測定

作成した各エージェントのある局面での, 指し手の一致率は以下のようになった. 初めに, 事前学習のプレイヤーとランダムな合法手を指すプレイヤーの双方の手番での指し手の一致率を測定した結果は図 4 のようになった. このうち, 事前学習のプレイヤーの手番での指し手の一致率を測定した結果は図 5 にランダムな合法手を指すプレイヤーの手番での指し手の一致率を測定した結果は図 6 のようになった. また,50epoch 毎の指し手の一致率を表 2 にまとめた.

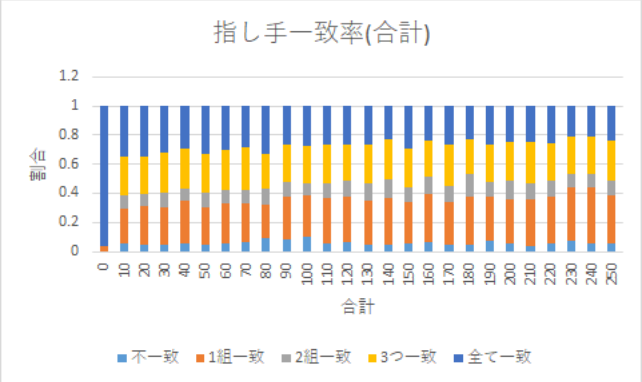

図 4 指し手の一致率（合計）

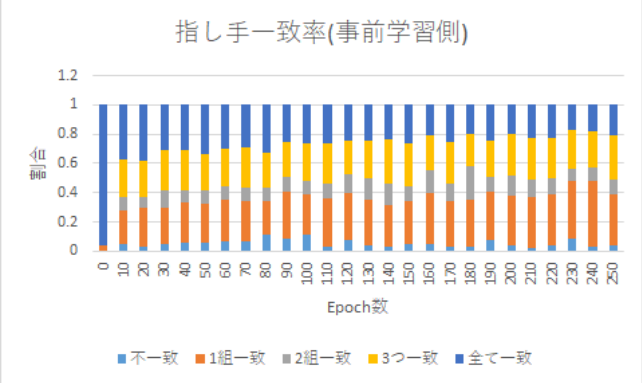

図 5 指し手の一致率（事前学習プレイヤー手番時）

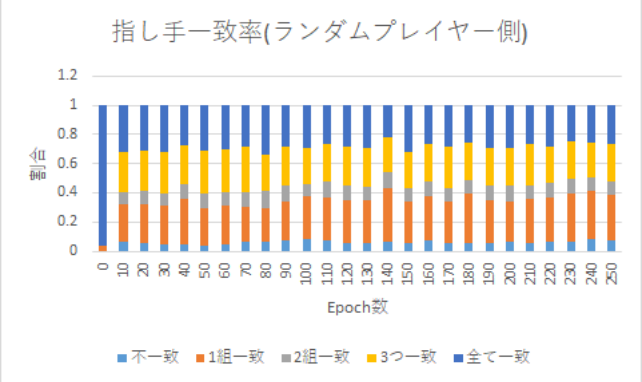

図 6 指し手の一致率（ランダムな合法手プレイヤー手番時）

表 2 指し手一致率（合計）

Epoch 数	不一致	1 組一致	2 組一致	3 つ同じ	全て同じ
0	0.000	0.037	0.000	0.000	0.963
50	0.047	0.260	0.094	0.273	0.326
100	0.098	0.286	0.086	0.251	0.279
150	0.053	0.284	0.102	0.270	0.290
200	0.051	0.311	0.123	0.270	0.245
250	0.058	0.328	0.096	0.284	0.234

提案手法の指し手の一致率の推移から, 学習が進むにつれ, 指し手の多様性を獲得した事が分かった. 特に, 全てのエージェントが同じ手を指す確率は epoch が進むにつれて下がっている. また, 指し手が一致している組が 1 組できる割合は上がっている. つまり,4 つのエージェントが 3 つの指し手を指す局面が増えている. また, 特定の手が有利になることが多い局面もあるため, 指し手が分散しない局面もあることから, 不一致の割合が上がらなかった可能性がある. このことから, エージェント間での指し手の多様性を獲得できたと考えられる.

5.5.2 Map-Elites の指し手一致率の測定

指し手の一致率の実験結果を表 3, 表 4, 表 5 にまとめた.

表 3 Map-Elites の指し手一致率（合計）

世代数	不一致	1 組一致	2 組一致	3 つ同じ	全て同じ
28	0.576	0.368	0.015	0.023	0.018
51	0.572	0.364	0.021	0.026	0.017
74	0.570	0.363	0.021	0.029	0.017
96	0.570	0.362	0.024	0.028	0.017

表 4 Map-Elites の指し手一致率（事前学習側）

世代数	不一致	1 組一致	2 組一致	3 つ同じ	全て同じ
28	0.527	0.390	0.020	0.028	0.035
51	0.528	0.388	0.024	0.026	0.033
74	0.519	0.391	0.025	0.033	0.033
96	0.528	0.380	0.029	0.030	0.033

表 5 Map-Elites の指し手一致率（ランダムプレイヤー側）

世代数	不一致	1 組一致	2 組一致	3 つ同じ	全て同じ
28	0.626	0.345	0.010	0.018	0.000
51	0.616	0.340	0.018	0.026	0.001
74	0.623	0.334	0.018	0.025	0.000
96	0.612	0.343	0.018	0.026	0.001

指し手一致率に関しては, 提案手法より優れた結果となっている.Map-Elites では, 割合としては 3 つ同じ場合と, 全て同じ場合が提案手法に比べ少なく, 代わりに全て違う指し手を指す場合が多い. この点は,Map-Elites のエージェントは勝率が高くないことから, 指したほうがよい手を指さない事がある可能性が考えられる.

5.6 評価 3 勝利条件の割合の測定

ランダムな合法手を指すと各学習済みエージェントの間で先手後手を入れ替え各 50 局, 計 100 局の対局を行い, ガイスターにある 3 つの勝利条件の比率ががどのように推移するかを調べる.

5.6.1 勝利条件の割合の測定

作成した各エージェントの満たした勝利条件の割合は図 7, 図 8, 図 9, 図 10 のようになった. どのエージェントも勝利条件の比率が学習途中で変動している. このため, それぞれの性質がわかりにくいため,0epoch から 250epoch までのエージェントの勝利条件の比率の平均を, 各エージェントに関して求め結果は表 6, 表 7 のようになった.

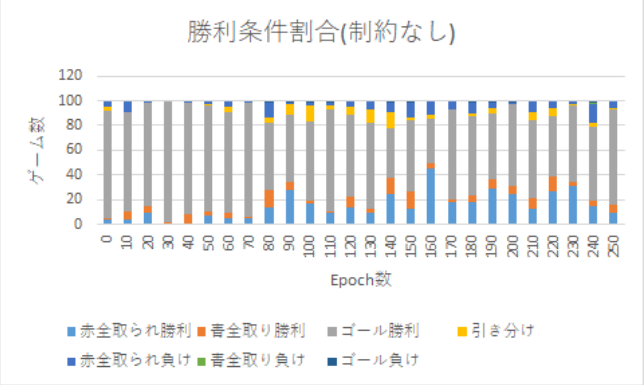

図 7 勝利条件割合 (制約なし)

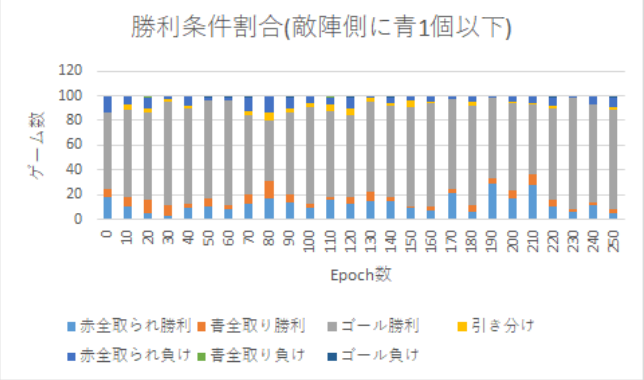

図 8 勝利条件割合 (青駒の個数 1 個以下)

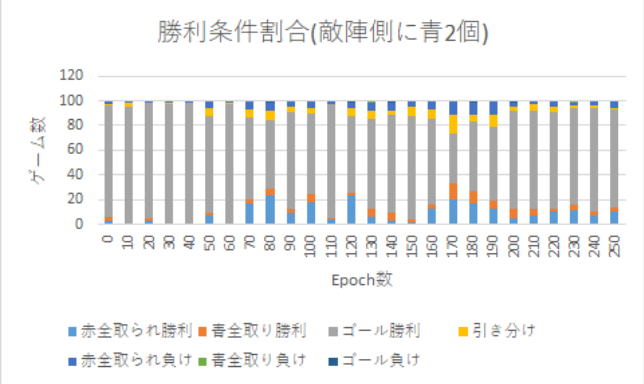

図 9 勝利条件割合 (青駒の個数 2 個)

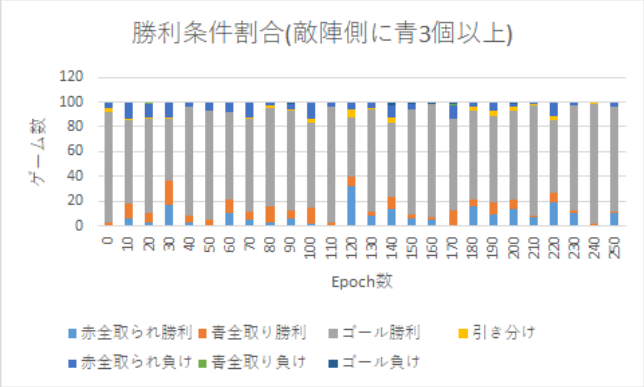

図 10 勝利条件割合 (青駒の個数 3 個以上)

表 6 勝利条件割合 (勝ち)

敵陣側青駒数	赤勝利	青勝利	ゴール勝利
0,1	12.38	5.46	73.27
2	8.85	4.15	77.19
3,4	8.04	6.65	76.69
制限なし	15.00	5.92	69.04

表 7 勝利条件割合 (負け・引き分け)

敵陣側青駒数	引き分け	赤負け	青負け	ゴール負け
0,1	2.31	6.12	0.08	0.38
2	4.38	5.12	0.08	0.23
3,4	1.69	6.46	0.08	0.38
制限なし	3.73	5.73	0.12	0.46

この実験結果の内, 赤駒を取られて勝利する場合の勝利条件を満たした割合は, 初期配置の制約がないエージェントである点が目立つ. 赤駒を前に置くエージェントの方が, この勝利条件を満たしやすいが, 自己対局時にそのような戦術の棋譜が増えた可能性がある. それにより, 赤を取らせる戦術が通用しなくなり, 赤を取らせる勝利条件の比率が下がったと考えられる.

5.6.2 Map-Elites の勝利条件の割合の測定

Map-Elites のプレイヤーの満たした勝利条件は図 11, 図 12, 図 13, 図 14 のようになった.

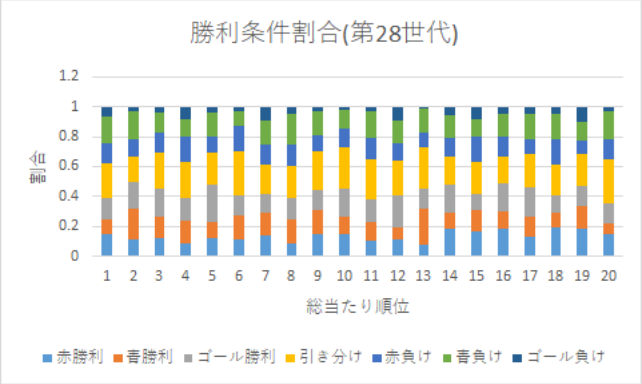

図 11 MAP-Eliets (28 世代目)

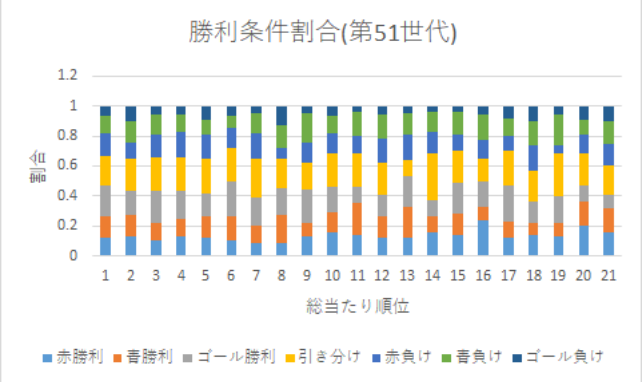

図 12 MAP-Eliets (51 世代目)

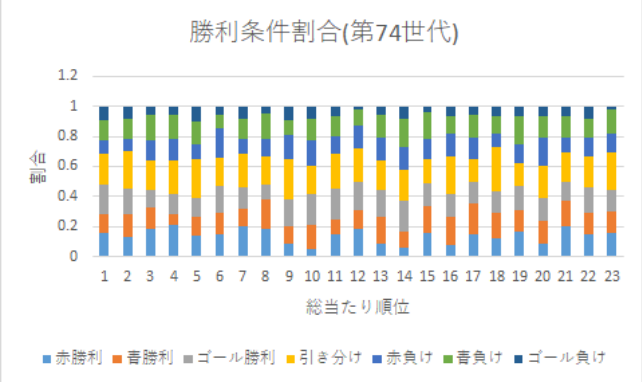

図 13 MAP-Eliets (74 世代目)

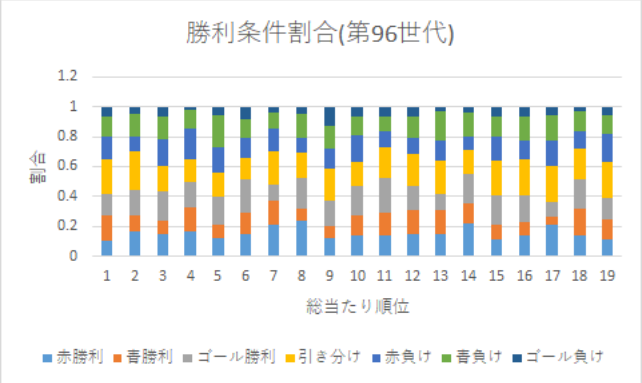

図 14 MAP-Eliets (96 世代目)

提案手法と勝率が違うため比較しにくいが, 勝利条件の比率も提案手法よりも各プレイヤーの差異が大きい. また, ランダムな合法手を打つプレイヤーに勝ちやすい, ゴールに到達する勝利条件の比率が低い. このことが, 勝率の低さにつながっており, 間接的に勝利条件の比率が異なる個体が生成されやすいと考えられる.

6. 考察

提案手法では, 一定の強さを持つ多様性な指し手を指すプレイヤーを作成することに成功した. 一方で, 勝利条件割合は epoch ごとの差異が大きかったが平均すると, 赤を取らせる勝利条件やゴールへ向かう勝利条件の比率は差異が大きかった. これは, ゴールへ向かう勝利条件は, 一回でも満たせば勝利であるため, これを狙う戦略をとれば比率が上がり, それを防ぐために駒を取ってくるプレイヤーに対して, 赤駒を取らせる戦術を狙えば, 赤駒を取らせる戦術が有効となるためであり, プレイヤーが意図すればそれらの確率が上がるためだと考えられる. また, 赤駒が前に多くなる制約を掛けたエージェントが, 制約のないエージェントより赤駒を取らせる勝利が少ないことから, 赤駒に制約を掛けたエージェントでは, 学習途中でこの戦略を多用し, その対策を学習したことにより, 赤駒を取らせる戦術の有効性が下がった可能性がある. このように, 制約を与えることで多様性な戦略を持つプレイヤーを作成する事が可能であると考えられる.

7. まとめ

本研究では, 深層強化学習を用い, 一定の強さを持つ, 多様な戦略を持つガイスターのコンピュータプレイヤーを, 事前学習を行ったエージェントを, 異なる初期配置の制限を与えた状態で自己対局させることによって作成した. 実験の結果, ランダムな合法手を指すプレイヤーに対して, 最低でも 0.81 以上の勝率を残し, 勝率は作成したエージェントのうち 78%で 0.9 を越えた. 指し手の一致率では,4 個のエージェント間全てで指し手が一致する確率は, 提案手法は 0.23 程度,Map-Elites では 0.017 程度であり, 異なる戦略を持つコンピュータプレイヤーを作成できた. また, 勝利条件の比率では, 提案手法はゴール勝利が 75%を占めてしまう結果となったが, 赤勝利の比率は最大で 7%程度違うエージェントの作成には成功した. しかしながら, 指し手が一致する局面もあるため, さらに多様な戦略を持つプレイヤーを作成することは, 今後の課題である.

一方で, 実際のガイスターでは, 勝率を犠牲にしても裏をかく作戦も存在するが, そのような手法は提案手法では実現できない. そのような戦術は, 教師有り学習によるファインチューニングを行う方法やそのような戦術に報酬を与える方法,Map-Elites のような別の手法を採用する方法やヒューリスティックなプレイヤーを作成する方法が考えられる. このような戦術を持つ多様なプレイヤーを作成する事も, 今後の課題である.

参考文献

[1] 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, pp1-7, 2019
[2] 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, pp1-6, 2014
[3] 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, 平成 27 年度電気通信大学 情報・通信工学専攻修士論文
[4] 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, The 24th Game Programming Workshop 2019,pp130-135, 2019
[5] 栃川　順平, 竹内　聖悟. モンテカルロ木探索ガイスターにおける遺伝的アルゴリズムを用いたプレイアウト方策の作成. The 26th Game Programming Workshop 2021,pp124-129, 2021
[6] 田村謙次, Map-Elites における被覆率を指標とした局所的探索手法, 情報処理学会第 84 回全国大会, 5B-02, 2022