Vol.2020-GI-43 No.9
2020/3/13

アクションゲームにおける特定のプレイヤの特徴を模倣するAIプレイヤの作成

池田 裕太郎[†1,a)]　池田 心[†1,b)]

概要：特定プレイヤの特徴を模倣する AI プレイヤは，特定プレイヤと遊んでいるような楽しさや，そのプレイヤとチームを組んだり対戦したりすることを想定した練習を提供できる点で有用である．プレイヤの特徴を模倣する方法としては，教師あり学習や履歴から模倣対象の価値観を推定する逆強化学習などのような方法が考えられるが，いずれの場合も大量の教師データが必要なため，学習が困難である．そこで本研究では，2D 横スクロール型アクションゲームである「スーパーマリオブラザーズ」シリーズを用い，特定プレイヤの特徴がどのようなところに表れるかを調査した上で，ペナルティ付き遺伝的アルゴリズムを用いてプレイヤの統計量を模倣することによって，比較的少ない教師データで特定プレイヤの特徴を模倣することを目指した．被験者実験や教師あり学習の結果から，AI はいくつかの統計量の違いからプレイヤどうしを 8 割程度の精度で区別できるが，人間は 6 割程度の精度でしか区別できないことがわかった．そこで模倣対象の特徴をそのまま模倣するのではなく，より強調した形で模倣した．最適化には NeuroEvolution と事例ベース政策最適化（EBP-GA）を用いた．EBP-GA を用いた最適化では 7 個の統計量に関して模倣対象の値に近づけることができた．

キーワード：模倣，統計量，遺伝的アルゴリズム，ペナルティ，ボーナス

Creation of AI Players Imitating Characteristics of Specific Players for an Action Game

YUTARO IKEDA[†1,a)]　KOKOLO IKEDA[†1,b)]

1. はじめに

近年，コンピューターゲームプレイヤ（以下ゲーム AI）は目覚ましい発展を遂げており，多くのゲームで人間のトッププレイヤに勝つなど強さの面では十分なものになりつつある．一方で，ゲーム AI には，強さだけではなく，味方や対戦相手となる人間プレイヤを楽しませることも求められている．そのためには人間プレイヤに違和感を与えてしまうような機械的なプレイではなく，自然な振る舞いをする必要がある．そこでゲーム AI 研究の次のステップとして，「人間らしい」ゲーム AI の研究が活発に行われている．例えば FPS ゲームにおける Turing Test の試み，2K BotPrize において，人間よりも人間らしいと評価されるゲーム AI が達成された [1]．

また，「人間らしい」振る舞いをするゲーム AI の延長として，「特定プレイヤらしい」ゲーム AI の研究も行われている [2]．特定プレイヤらしいゲーム AI には，特定プレイヤと遊んでいるような楽しさを提供できたり，特定プレイヤが対戦相手や味方にいることを想定した練習に利用できたりするなどさまざまな利点がある．しかしその一方で特定プレイヤから大量の教師データを集めるのは難しく，単純な教師あり学習や，履歴から模倣対象の価値観を推定する逆強化学習のような手法では学習が困難であるという問題がある．

[†1] 現在，北陸先端科学技術大学院大学
Presently with Japan Advanced Institute of Science and Technology
[a)] yutaro.ikeda@jaist.ac.jp
[b)] kokolo@jaist.ac.jp

そこで本研究では，「特定プレイヤの特徴がどのような部分に表れるか」，「それを少ない履歴から抽出，再現するにはどうしたらよいか」を解明することを目的とする．対象としては世界的に有名な 2D 横スクロール型アクションゲームである「スーパーマリオブラザーズ」シリーズを用いる．研究対象としてこれを選んだ理由としては，1) アクションゲームの中でも比較的行動の自由度が高く，特定プレイヤらしさが表れやすい，2) 2D 横スクロール型アクションゲームには複数人での協力プレイや対戦プレイが可能なタイトルも多数あり，特定プレイヤの特徴を模倣するゲーム AI の開発は有用である，の 2 点があげられる．

少ない教師データでプレイヤの特徴を模倣するには，「1 ゲーム中のジャンプの回数」のような統計的なデータを用いることが有効だと考える．Phuc らの手法 [3] を参考にし，模倣対象の統計量との差分に応じたボーナスやペナルティを与える遺伝的アルゴリズム（GA）を用いて模倣対象に近い統計量を持つエージェントを作成する．

2. Mario AI Benchmark

図 1　ゲーム画面

Mario AI Benchmark は，Togelius らによって開発された，世界的に有名な 2D 横スクロール型アクションゲームである「スーパーマリオブラザーズ」シリーズを模した研究開発用プラットフォームである．2009 年から行われている Mario AI Championship というマリオ AI の性能を競うコンテストのプラットフォームとして使われている．コンテストの種類としては，マリオを操作する AI の「上手さ」や「人間らしさ」を競うものや，ステージを生成する AI が人間にとってどれだけ魅力的なステージを生成できるかを競うものなどがある．Mario AI Benchmark は Web ページから無償でダウンロードできる [4]．

3. 関連研究

本研究の目的は，特定プレイヤらしい MarioAI プレイヤの作成である．本章では，これに関連して既存の「MarioAI プレイヤに関する研究」，および「特定プレイヤらしい AI プレイヤに関する研究」を紹介する．

3.1 MarioAI プレイヤに関する研究

高速なクリアを目指すものとしては A*アルゴリズムを用いたエージェントが有名である [5]．S 字の動きなど迂回を要しないステージでは極めて高速なクリアが可能である．

人間らしい MarioAI プレイヤを実現する研究も行われている．藤井らは，人間プレイヤに共通してみられる生物学的制約を表現し，それを A*探索や Q 学習に導入することで人間よりも人間らしくみえるようなエージェントの獲得に成功している [7]．

また Phuc らは「倒した敵の数」のような統計量に着目し，人間プレイヤの各統計量との差分に応じたペナルティを与える Neuroevolution を用いて少ない教師データで人間らしさを向上させることに成功している [3]．

3.2 特定プレイヤらしい AI プレイヤに関する研究

特定プレイヤを模倣する方法としては，特定プレイヤの状態行動対のデータを大量に取得し学習させる方法や探索アルゴリズムを特定プレイヤがよく用いる手や行動が出やすいように偏らせる方法などがある．田中らは，対戦型格闘ゲームにおいて，キャラクタ間の間合いなどのような状態とその時にとった行動の組を教師データとすることで特定のプレイヤの行動パターンを再現した [2]．

また，隅山らは落下型パズルゲームである「ぷよぷよ」を対象に，模倣対象のプレイデータから，模倣対象がよく使う定石形が表れやすいように探索を行うことで特徴を模倣した [8]．

4. 被験者実験

本研究の目標は，人間の目からみて特定プレイヤらしいと感じるような MarioAI プレイヤを作ることである．そのため，模倣 AI プレイヤを作成する前に特定プレイヤの特徴を人間がどの程度正確に認識できるのかを調べなければならない．そこで，人間の認識の精度を調べるための被験者実験を行った．

11 人の被験者に 2 つのプレイ動画を見比べてもらい，同一人物のプレイかどうかを判定してもらう被験者実験を行った．2 つの動画のプレイヤをそれぞれ P1，P2 とすると，まず P1 のプレイ動画を見てもらい，その後 P2 のプレイ動画を見てもらった．P1 と P2 のプレイ動画はそれぞれ別のステージのものとし，全部で 6 ペア見てもらった．

また，特定プレイヤの特徴が表れる統計量の候補を発見するために，被験者にそれぞれの問題に対する回答の根拠も記述してもらった．回答の根拠や統計量に関しては 5 章で記述する．

表 1 被験者実験の正答率

	問題数	正解	不正解	正答率
P1 = P2	31	15	16	0.48
P1 ≠ P2	29	21	8	0.72
全体	60	36	24	0.6

結果は表 1 のようになった．P1 ≠ P2 の問題の正答率は比較的高い値であり，少なくとも人間の目にも分かる形でプレイに違いが表れているといえる．しかし，正答率が 100 %に近いわけではない．P1 と P2 の両者が特徴の少ない平均的なプレイヤだった場合，両者のプレイに違いがあまり表れず，判断が難しくなるためだと考えられる．P1 = P2 の問題の正答率が低かった理由としては，同じプレイヤのプレイでも片方の動画にしか特定の振る舞いが表れない場合があり，その振る舞いに被験者が注目して別人のプレイだと判断してしまったことが考えられる．2 つのプレイヤが同じであることを正しく認識するためには，短い時間の 2 つの動画だけでは判断材料として不十分だった可能性がある．全体的にみると，被験者実験の設定下では，被験者は 2 つのプレイをうまく区別できない可能性があることが分かった．人間に特定プレイヤらしいと感じさせるには何らかの特別な工夫が必要そうだと分かった．

5. 特定プレイヤの特徴が表れる統計量の特定

5.1 統計量の候補の決定

4 章で記述した被験者実験を行った際，回答の根拠を記述してもらった．P1 = P2 の問題に正解した人の根拠には「どちらもファイアで敵を倒してから進んでいた」，「どちらもコインブロックをできるだけ叩こうとしていた」，「どちらも慎重さがなく，なるべく早くクリアしようとしていた」，「甲羅の使い方が似ていた」などがあった．また，P1 ≠ P2 の問題に正解した人の根拠には「片方は敵をほとんど倒さなかった」，「敵や穴と間合いを取る際の Left 入力の多さが違った」，「片方はダッシュの持続時間が長かった」，「片方はしゃがみを使っていた」などがあった．
これらの根拠に出てくる計算可能な数値は，特定プレイヤの特徴が表れる統計量として利用できる可能性が高いと考えた．そこで，正解した際の被験者の回答の根拠から，プレイヤの特徴が表れる統計量の候補を 26 種類決定した．以下に統計量の候補の例を示す．

表 2 統計量の候補の例

とったコインの数	倒した敵の数
甲羅を拾った回数	とったアイテム（きのこ+フラワー）の数
左キー入力回数	左キー入力時間
右キー入力回数	右キー入力時間
下キー入力回数	下キー入力時間
ジャンプキー入力回数	ジャンプキー入力時間
ダッシュキー入力回数	ダッシュキー入力時間
無操作時間	プレイ時間

5.2 統計量の絞り込み

26 種類の統計量をすべて模倣対象に近づけるには大きな計算コストを要することが予想される．そこで本節では，26 種類の統計量の候補から特定プレイヤらしさに強い関係のあるもののみを抽出して数を絞り込む．絞り込むために，各統計量について異なるステージ間の相関を調べる．図 2 にステージ間の相関の強い統計量，図 3 に相関の弱い統計量のグラフの例を示した．これは 12 人の各プレイヤのステージ 1 とステージ 2 の値である．
ある統計量が本当に特定プレイヤらしさを表すものであれば，その統計量についてはステージが違っても同じプレイヤで共通した傾向が表れるはずである．例えば，図 2 をみると，ステージ 1 でプレイ時間が長いプレイヤはステージ 2 でもプレイ時間が長い傾向がある．一方，図 3 をみると，そのような傾向はないため，必ずしも模倣しなければいけない統計量とは言えない．

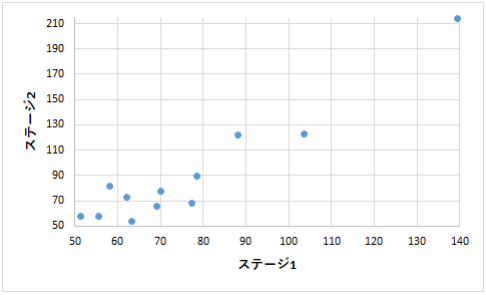

図 2 相関の強い統計量（プレイ時間）の各プレイヤの値

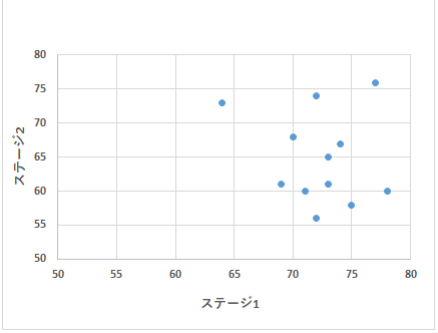

図 3 相関の弱い統計量（とったコインの数）の各プレイヤの値

図 2 のようになる統計量をプレイヤの特徴が表れる統計量として決定することにより，統計量を 6 個まで絞り込んだ．以下にその統計量を示す．

表 3　統計量の候補の例

左キー入力時間	右キー入力時間
ダッシュキー入力時間	下キー入力回数
無操作時間	プレイ時間 (s)

5.3　AI による区別は可能か?

人間の目から見て特定プレイヤらしい AI プレイヤを実現するためには，模倣に用いる統計量が 1) プレイヤの特徴を良く表していること，2) かつそれが人間の目からも分かること，の 2 つを満たしていなければならない．まず，1) を調べるために，各統計量を入力とした教師あり学習によって被験者実験における「正解」を予測できるかを検証する（実験 1）．次に 2) を調べるために，同様の方法で被験者実験における「被験者の回答」を予測できるかを検証する（実験 2）．（実験 1）は AI からみた「特定プレイヤらしさ」，（実験 2）は人間からみた「特定プレイヤらしさ」を扱っている点で 2 つの実験には明確に違いがある．

【実験設定】

- 入力は 6 次元で，「左キー入力時間」，「右キー入力時間」，「下キー入力回数」，「ダッシュキー入力時間」，「無操作時間」，「プレイ動画」についての 2 つの動画間の差の絶対値とした．
- 実験 1 では，出力は「同じプレイヤか，違うプレイヤか」の 2 値とした．
- 実験 2 では，出力は「被験者が同じプレイヤと答えたか，違うプレイヤと答えたか」の 2 値とした．
- データ数は 60 で内訳は P1 = P2 の問題が 31，P1 ≠ P2 の問題が 29 である．

【実験結果】

表 4　実験 1 の結果：同じ人・違う人の区別ができたか

教師データに対する正答率	0.85
テストデータに対する正答率	0.82

表 5　実験 2 の結果：被験者の回答を予測できたか

教師データに対する正答率	0.62
テストデータに対する正答率	0.62

表 4 と 5 より，正解はある程度うまく予測できているが，被験者の回答はうまく予測できていないことが分かる．このことから，絞り込んだ統計量にプレイヤの特徴が表れており，AI はそれを認識してプレイヤを区別できるが，人間にはそれができない可能性があることが分かった．仮にそうだとするならば，統計量を模倣対象の値に近づけるだけでは，人間の目からみて特定プレイヤらしい AI プレイヤを実現できない．我々の目的は AI プレイヤに「AI にとって特定プレイヤらしい」振る舞いをさせることではなく，「人間の観察者にとって特定プレイヤらしい」振る舞いをさせることであるため，統計量に表れる特定プレイヤの特徴を人間にも認識できるようにするための何らかの工夫をする必要があると考えた．

5.4　特徴の強調

4 章の被験者実験や 5.3 の教師あり学習の結果から，各統計量の値を模倣対象に近づけただけでは人間は気付かない可能性がある．そこで，人間の平均値から大きく値が離れた統計量を模倣対象の特徴が表れる統計量とみなし，それに関しては，模倣対象の値よりもさらに平均値から離れた値に近づけることによって模倣対象の特徴を強調する．

	人間の平均値	模倣対象の値		模倣AIにとってほしい値（基準値）
左キー入力時間(s)	6	4		4
右キー入力時間(s)	32	35		35
ダッシュキー入力時間(s)	40	**20**	→	**10**
下キー入力回数	6	5	強調	5
無操作時間(s)	20	17		17
プレイ時間(s)	80	**120**	→	**140**

図 4　強調の例

6.　模倣 AI プレイヤ作成

各統計量について，模倣対象との差分に応じたペナルティを与える GA を用いて模倣 AI プレイヤを作成する．模倣対象らしくない行動を，if-then ルールなどでハードに禁止することもできるだろうが，それではステージをクリアするための性能が著しく落ちてしまう可能性がある．そのようなときにペナルティという形でソフトに抑制できるというのがこの手法の利点である．

ベースとなるモデルとして，まずは Neuroevolution[10] を用いて実験を行った．次に EBP-GA[9] を用いて同様の実験を行った．

6.1　評価値について

個体は Mario AI Benchmark 内で評価される．評価値はゲーム内スコア ＋ボーナス − ペナルティで計算される．

ゲーム内スコアはどれだけ上手にゲームをプレイしたかを表しており，ステージを先に進むほどこの値は高くなっていく．この値を高めてクリアするように学習することが主な目的である．

我々の目的はステージをクリアして尚且つ各統計量について，模倣対象との差分ができるだけ小さいエージェント

を作成することである．そこで，ゲーム内スコアに加えて各統計量における模倣対象との差分に応じたボーナスやペナルティを与えることにより，それを実現する．以下に，ボーナスとペナルティの計算方法について述べる．

6.1.1 ボーナス

ステージを 8 分割して，各区画を通過するのにかかった時間の模倣対象との差分が少ないほど大きいボーナスを与える．ステージを分割してボーナスを与えることにより，ある一か所に長時間とどまるように学習することを防ぐ．i 区画におけるボーナスは i 区画から i + 1 区画に入る際に与えられる．また，ペナルティではなくボーナスにした理由としては，ペナルティにした場合，i 区画におけるペナルティが与えられることによって評価値が下がり，i 区画から i + 1 区画に進むのを避けるように学習してしまう可能性があるためである．各区画では 0〜100 点のボーナスがもらえる．各区画のボーナスは図 5 に示す関数を用いる．6.2.1 と 6.3.1 の実験では（設定 1），6.4 の実験では（設定 2）で行った．

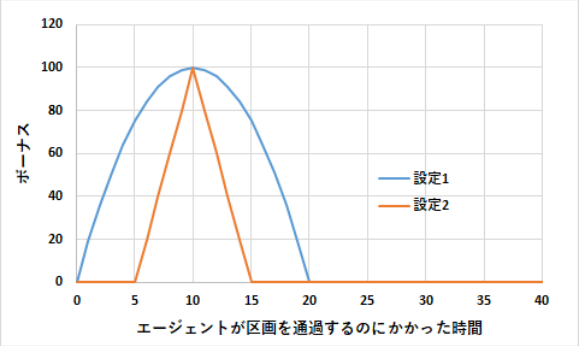

図 5 目標通過時間 10 秒の区間の通過時間に対するボーナスの値

6.1.2 ペナルティ

プレイ時間以外の統計量における模倣対象との差分に応じたペナルティを与える．エージェントのとった値に対するペナルティは図 6 に示す関数を用いる．

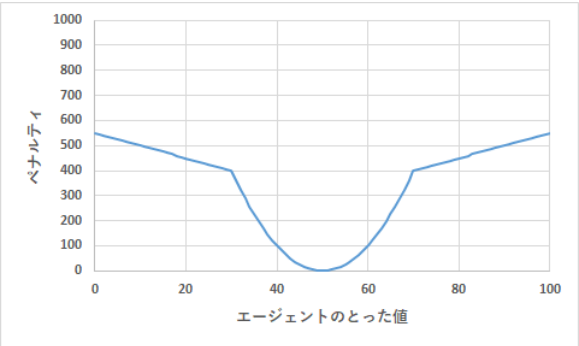

図 6 目標値 50 の場合のエージェントのとった値に対するペナルティ

6.2 Neuroevolution

Neuroevolution は GA により最適なニューラルネットワークを探索する手法である．通常ニューラルネットワークの学習では，構造が固定され，重みが back propagation で調節されることが多いが，Neuroevolution では重みを勾配法ではなく直接 GA で最適化すること，構造も最適化の対象になりうることが特徴的である．

Neuroevolution のマリオエージェントとしては Togelius らによるものがある．Togelius らは，環境情報を取得するマリオの周囲のマス目の数や，ニューラルネットワークの種類を変えて，それぞれのエージェントの性能を比較している [6]．ニューラルネットワークの種類としては，結合重みを変化させる多層パーセプトロン (MLP)，再帰型ニューラルネットワーク（SRN）や，結合重みに加えて構造そのものを変化させる HyperGP(hybrid neuroevolution/genetic programming algorithm) などがある．結果，学習したステージをクリアする性能に関しては，マリオの周囲 3 × 3 マスの環境情報を入力とした MLP が最も高かった（レベル 3 程度までクリア）．幅広いレベルのステージをクリアできることは，模倣 AI プレイヤの汎用性を上げるという点で重要である．そのため，今回は，Togelius らのマリオの周囲 3 × 3 マスの環境情報を入力とした MLP をニューラルネットワークとして利用することにした．

6.2.1 実験の説明

Neuroevolution のエージェントに

・プレイ時間
・ジャンプキー入力回数
・接地時間

に関して模倣対象との差分に応じたボーナスもしくはペナルティを与えて学習させ，ボーナスやペナルティを与えなかった場合と比べてどれだけ模倣対象の値に近づけることができるかを検証した．特定プレイヤらしさに関する統計量ではない「ジャンプキー入力回数」，「接地時間」をペナルティの項目として追加した理由としては，ボーナスやペナルティを与えなかった場合の Neuroevolution エージェントは常にジャンプしながら前に進んでいて不自然にみえたため，「ジャンプキー入力回数」と「接地時間」にもペナルティを与えることにより，そのような振る舞いを抑制しようとしたためである．学習は 1 試行 50000 世代で 10 試行行った．

【実験設定】

- 入力は 21 次元で，マリオが「接地してるかどうか」や「ジャンプ可能かどうか」，マリオの周囲 3 × 3 マスの「オブジェクトの有無」と「敵の有無」，「最後に着地し

てからの接地しているフレーム数」である．
- 出力は 5 次元で，「キー入力」である．

6.2.2 実験結果

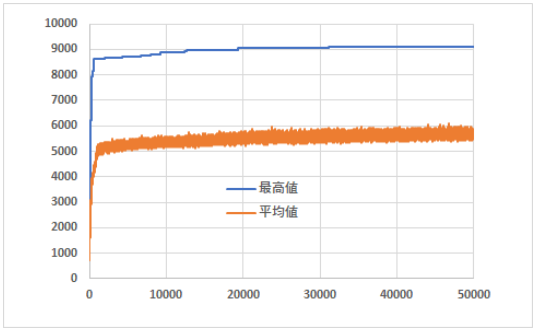

図 7 評価値

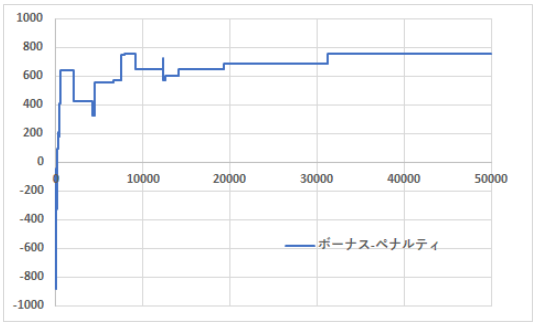

図 8 ボーナス − ペナルティ

10 試行中 10 試行クリアできていた．全ての試行において 200-500 世代（時間にすると標準的な PC で 1,2 分程度）でステージをクリアするエージェントを得ることに成功した．

図 7 に，1 試行 50000 世代についての集団内の評価値の最高値と平均値を示す．8600 点ほどに到達し，その後も評価値を上げていく様子が分かる．

図 8 には，ゲーム内スコア以外の，加算したボーナスと減算したペナルティの推移（集団内のベスト解の）を示す．ごく序盤はボーナスよりもペナルティが大きいが，50000 世代までの最高値は 755 であった．ボーナス − ペナルティの上限は 800 であるから，十分にボーナス − ペナルティを大きくできていた．

表 6 ボーナス・ペナルティなしとありの Neuroevolution エージェントの各統計量値の比較

	基準値	ボーナス・ペナルティなし	ボーナス・ペナルティあり
ジャンプキー入力回数	66.00	62.50	**67.00**
接地時間	23.00	3.52	**16.04**
プレイ時間（区画 1）	6.67	3.27	4.25
プレイ時間（区画 2）	7.88	2.50	3.56
プレイ時間（区画 3）	7.13	3.13	2.90
プレイ時間（区画 4）	7.71	3.56	5.44
プレイ時間（区画 5）	6.50	3.35	6.71
プレイ時間（区画 6）	6.33	3.81	6.48
プレイ時間（区画 7）	5.38	2.31	3.17
プレイ時間（区画 8）	3.96	2.83	3.98
プレイ時間（合計）	51.54	25.00	**39.00**

各統計量をある程度模倣対象の値に近づけることができていた．しかし，「接地時間」に関しては 7 秒，「プレイ時間」に関しては 13 秒程度の差があり，十分に模倣対象の値に近づけることができていなかった．各区画のプレイ時間の差分はそれほど大きくはないが，合計すると大きな差になってしまっていることがわかる．ボーナスの（設定 1）では，各分割地点における模倣対象との差分の変化によるボーナスの変化が小さい（2 秒離れていてもボーナスは 4 点しか減点されない）ため，模倣対象の値に近づけるための学習が十分に行われなかったことが原因だと考えられる．挙動に関しては，「ジャンプキー入力回数」と「接地時間」をペナルティとして与えたことにより，ジャンプせずに前に進んでいる場面も見られたが，まだジャンプしながら前に進んでいる印象が残っていた．

6.3 EBP-GA

EBP（事例ベース探索）は，何らかの示唆的な情報，例えばある状態の好ましさなどを表す事例の集合と，その形式に応じた推論の手法を用いて行動を選択するアルゴリズムを持つような政策のことである [9]．本研究では EBP の中でも，事例を状態と行動の対で直接表現する状態-行動型 EBP を採用し，個体のもつ事例集合を GA を用いて最適化する（EBP-GA）．個体の意思決定は現在の状態に最も近い事例を探し，その行動を返す 1 - nearest neighbor を用いた．

一般的に，Neuroevolution に比べて，EBP-GA には 4 つの利点が期待できる．1 つ目は 1 つのパラメータの値を変化させたときに全体に及ぼす影響が限定的であるという点である．つまり，パラメータの値を変化させることにより行動が大きく変化して評価値が悪化することが少ないため，評価値の高い親からは評価値の高い子供が生まれやすい．2 つ目は交叉の設計が自然であり，事例を混ぜれば，取る行動が親のどちらかと同じになるという点である．ニューラルネットワークの場合，重みを交叉させても取る行動が親と同じ行動になるとは限らない．3 つ目は各事例にはある状態に対して取るべき行動が明確に示されているため，説明可能性が高く，結果に対する原因の追究が容易であるという点である．4 つ目は人間のデータを初期値として用いるのが容易な点である．ニューラルネットワークにおいても教師あり学習することは可能だが EBP は人間の事例のデータをそのまま初期値として利用可能なのでニューラルネットワークよりも簡単である．その一方で，遺伝子が持つパラメータが多いため，進化が遅い場合がある，1 - nearest neighbor での意思決定に時間がかかるなどの欠点もある．

6.3.1 実験の説明

EBP-GA のエージェントに

・プレイ時間
・ジャンプキー入力回数
・接地時間

に関して模倣対象との差分に応じたボーナスもしくはペナルティを与えて学習させ，ボーナスやペナルティを与えなかった場合と比べてどれだけ模倣対象の値に近づけることができるかを検証した．学習は 1 試行 50000 世代で 10 試行行った．

【実験設定】

- 入力は 100 次元で，マリオの上下と前方 7 × 7 マスにおける「オブジェクトとの距離」,「敵との距離」,「オブジェクトどうしの位置関係」,「敵どうしの位置関係」に加えて,「接地してるかどうか」や「最後に着地してからの接地しているフレーム数」,「マリオの x 座標」,「マリオの x 座標が変化しなくなってからのフレーム数」である．
- 出力は 5 次元で,「キー入力」である．

6.3.2 実験結果

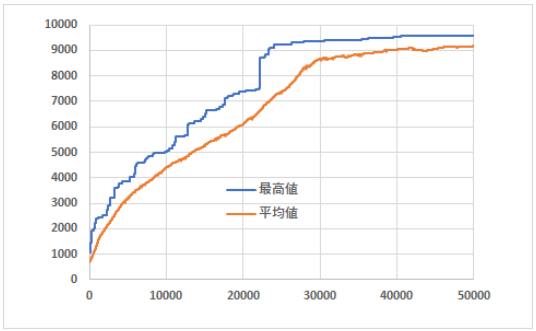

図 9 評価値

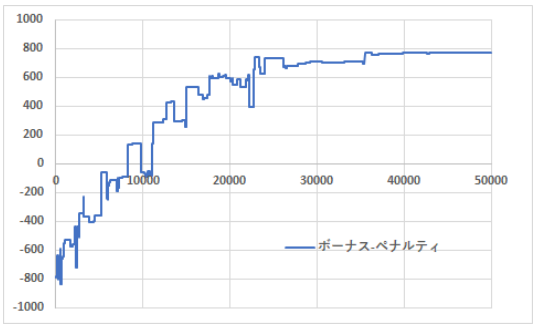

図 10 ボーナス − ペナルティ

全ての試行において 20000-30000 世代（時間にすると標準的な PC で 8 時間程度）でステージをクリアするエージェントを得ることに成功した．

図 9 に，集団内の評価値の最高値と平均値を示す．Neuroevolution の結果よりも 400 点ほど高い 9000 点ほどに到達し，その後も評価値を上げていく様子が分かる．

図 10 には，ゲーム内スコア以外の，加算したボーナスと減算したペナルティの推移（集団内のベスト解の）を示す．10000 世代あたりまではボーナスよりもペナルティが大きいが，50000 世代までの最高値は 771 であった．ボーナス − ペナルティの上限は 800 であるから，十分にボーナス − ペナルティを大きくできていた．

Neuroevolution のエージェントと比較すると，進化速度は遅いが，50000 世代目到達時のボーナス-ペナルティの 10 試行分の中央値は EBP-GA のほうが大きかった（Neuroevolution692, EBP-GA746）．よって EBP-GA は学習に時間はかかるが，最終的な性能は Neuroevolution のエージェントよりも高いといえる．

表 7 ボーナス・ペナルティなしとありの EBP-GA エージェントの各統計量の比較

	基準値	ボーナス・ペナルティなし	ボーナス・ペナルティあり
ジャンプキー入力回数	66.00	50.00	**66.00**
接地時間	23.00	8.63	**19.23**
プレイ時間（区画 1）	6.67	4.19	4.79
プレイ時間（区画 2）	7.88	3.40	7.85
プレイ時間（区画 3）	7.13	3.25	3.29
プレイ時間（区画 4）	7.71	3.40	5.23
プレイ時間（区画 5）	6.50	2.90	5.54
プレイ時間（区画 6）	6.33	3.71	5.69
プレイ時間（区画 7）	5.38	2.21	2.96
プレイ時間（区画 8）	3.96	2.75	2.90
プレイ時間（合計）	51.54	27.08	**39.73**

「プレイ時間」以外は十分に模倣対象に近づけることができていた．挙動に関しては，敵を狙ってファイアを撃ったり甲羅を投げる，必要なときにのみジャンプする，ダッシュの慣性を消すために進行方向と逆方向に入力するなどのプレイヤの意図が感じられる動きがみられ，Neuroevolution のエージェントより人間らしく感じられた．EBP-GA のほうが人間らしい理由としては EBP-GA は初期値として人間プレイヤのデータを与えていること，マリオの周囲の環境情報の取得範囲が Neuroevolution は 3 × 3 マスなのに対して，EBP-GA は 7 × 7 マスだということが考えられる．

6.4 ペナルティを増やした場合の実験の説明

ここまでの実験で，2 種類 10 個程度の統計量については模倣対象に近づけることができること，ただしプレイ時間については改良の余地があることが分かった．そこで次の実験では，統計量の種類を増やしても学習できるか，プレイ時間の改善は可能かを確かめた．

EBP-GA のエージェントに

・プレイ時間
・ジャンプキー入力回数
・接地時間

・左キー入力時間
・右キー入力時間
・ダッシュキー入力時間
・下キー入力回数
・無操作時間

に関して模倣対象との差分に応じたボーナスもしくはペナルティを与えて学習させ，ボーナスやペナルティを与えなかった場合と比べてどれだけ模倣対象の値に近づけることができるかを検証した．また，これまでは EBP-GA エージェントのプレイデータはすべて筆者のものを用いており，尚且つ統計量の基準値も筆者の値を用いていたが，今回は別プレイヤの基準値を用いた．さらに，各分割地点における模倣対象との差分の変化によるボーナスの変化を大きくして，よりプレイ時間を基準値に近づけやすくするために，ボーナスの設定を（設定 1）から（設定 2）に変更した．その他の実験設定は 6.3.1 と同じである．

6.4.1 実験結果

全ての試行において 20000-30000 世代（時間にすると標準的な PC で 8 時間程度）でステージをクリアするエージェントを得ることに成功した．

評価値 8200 点ほどに到達し，その後も評価値を上げていく様子がみられた．

50000 世代までのボーナス − ペナルティの最高値は 752 であった．ボーナス − ペナルティの上限は 800 であるから，十分にボーナス − ペナルティを大きくできていた．

ペナルティやボーナスを与える項目を増やしてもクリアするまでの世代数や評価値が収束するまでの世代数はあまり変わらなかった．ペナルティやボーナスを与える項目を増やしても適切に重みを設定してやれば，学習に与える影響は少ないと言える．

表 8 ボーナス・ペナルティなしとありの EBP-GA エージェントの各統計量の比較

	基準値	ボーナス・ペナルティなし	ボーナス・ペナルティあり
ジャンプキー入力回数	50.00	50.00	57.50
接地時間	24.96	8.63	**29.02**
左キー入力時間	3.63	0.46	**1.40**
右キー入力時間	21.96	18.88	**24.40**
ダッシュキー入力時間	52.67	22.85	**47.63**
下キー入力回数	0.00	0.00	0.00
無操作時間	1.79	1.75	1.77
プレイ時間（区画 1）	6.71	4.19	6.31
プレイ時間（区画 2）	9.75	3.40	9.58
プレイ時間（区画 3）	5.83	3.25	5.56
プレイ時間（区画 4）	8.75	3.40	8.40
プレイ時間（区画 5）	6.21	2.90	5.88
プレイ時間（区画 6）	6.29	3.71	6.04
プレイ時間（区画 7）	5.13	2.21	3.60
プレイ時間（区画 8）	6.13	2.75	5.35
プレイ時間（合計）	54.79	27.08	**51.35**

「ジャンプキー入力回数」以外は全体的に基準値に近づいていた．ボーナスの設定を変更したことにより，プレイ時間を改善することができた．また EBP に用いたプレイデータと基準値に用いた統計量が別のプレイヤのものであっても，統計量を基準値に近づけることができた．

「ジャンプキー入力回数」に関しては，ペナルティの重みが小さすぎたことが原因だと考えられる．

挙動に関しては，「ダッシュキー入力時間」や「プレイ時間」を模倣対象に近づけるために，障害物にぶつかりながらもダッシュキーを押していたり，障害物がない平たんな場所で止まっていたりするような不自然な挙動も少し見られた．これを改善するためには，不自然な動きを抑制するようなボーナスやペナルティを加える方法が考えられる．

7. おわりに

本研究では，「スーパーマリオブラザーズ」シリーズを対象に，「特定プレイヤらしさ」に関係する統計量を特定した．また，各統計量に関して，模倣対象との差分に応じてボーナスやペナルティを与える GA を用いて，各統計量の値を模倣対象に近づけることに成功した．今後は不自然な挙動を無くすようなペナルティを追加し，尚且つ模倣対象の特徴を強調した値に近づけるように最適化を行った上で，模倣 AI プレイヤの評価を行っていきたい．

参考文献

[1] Mihai Polceanu. Mirrorbot: Using human-inspired mirroring behavior to pass a turing test, Computational Intelligence in Games (CIG) 2013 IEEE, pp.1–8 (2007)
[2] 服部 裕介，田中 彰人，星野 准一，対戦型アクションゲームにおけるプレイヤの模倣行動の生成，第 17 回ゲーム情報学研究会，情報処理学会 (2007)
[3] Luong Huu Phuc, Kanazawa Naoto, Ikeda Kokolo, Learning Human-like Behaviors using NeuroEvolution with Statistical Penalties, IEEE Conference on Computational Intelligence and Games 2017 pp. 207-214 (2017)
[4] 「Mario AI Framework」`http://marioai.org/` (2020 年 1 月 20 日アクセス)
[5] Julian Togelius, Sergey Karakovskiy and Robin Baumgarten, The 2009 Mario AI Competition, Evolutionary Computation, pp.1-8 (2010)
[6] Julian Togelius, Sergey Karakovskiy, Jan Koutnik, Jurgen Schmidhuber, Super Mario Evolution, 2009 IEEE Symposiumon Computational Intelligence and Games, pp.156-161, (2009)
[7] 藤井叙人，人間らしい振る舞いを自動獲得するゲーム AI に関する研究，2016 年関西学院大学大学院博士論文 (2016)
[8] 隅山淳一朗，橋山智訓，田野俊一，ぷよぷよにおける人間のプレイデータの特徴量抽出，31st Fuzzy System Symposium (2015)
[9] 池田心，小林重信，喜多一，多様な戦略選択を可能にする事例ベースの政策表現とその GA による最適化，人工知能学会論文誌，25(2)，2010，351-362
[10] Yao,X., "A Review of evolutionary artificial neural networks," International Journal of Intelligent Systems, vol.8, pp.539-567, 1993.