

対戦型ゲームにおける戦略多様性についての StGA法を用いた自動分析手法の提案とその評価

山本 界人^{1,†1} 水野 竣介^{1,†2} ターウォンマツト ラック^{1,a)}

受付日 2014年2月21日, 採録日 2014年9月12日

概要: 本稿では, 対戦型ゲームにおける戦略多様性の観点からゲームバランスの分析を自動的に行う手法を提案し, その有用性を評価した. ゲーム開発におけるゲームバランスの分析, 調整は面白いゲームを作成するためには不可欠な要素である. 一方で, その分析, 調整のプロセスには時間的なコストがかかる. 開発期間に限りがあるゲーム開発の現場では, 十分にゲームバランスの分析, 調整を行うことができない場合も少なくない. このため, ゲームバランスを自動的に分析する手法が必要とされている. 既存のゲームバランスの自動分析手法は, 1つの状況を分析するために大きな計算時間を必要とする, もしくは事前に人間がゲームタイトルに依存する専門的な知識を必要とするものだった. このため, 人間が知識を獲得していない, かつ多くの状況が存在するゲームへと適用する場合には大きな計算時間を必要とした. そこで, 本稿では, Stochastic Genetic Algorithm (StGA) を用いて, 専門的な知識なしに多くの状況を持つゲームを分析する手法を提案する. 国際 AI 大会のプラットフォームとして利用されている FightingICE を対象にした実験から, 本手法の有用性を確かめた.

キーワード: ゲームバランス分析, ゲームバランス可視化, 遺伝的アルゴリズム, 強化学習, ゲーム設計/デザイン支援

Proposal and Evaluation of an Automatic Analysis Method Using StGA for Strategic Diversity in Action Games

KAITO YAMAMOTO^{1,†1} SHUNSUKE MIZUNO^{1,†2} RUCK THAWONMAS^{1,a)}

Received: February 21, 2014, Accepted: September 12, 2014

Abstract: In this paper, we propose and evaluate an automatic analysis method for game balance from the perspective of strategic diversity in action games. Analysis and adjustment of game balance in game development are essential elements in order to create an interesting game. However, time cost is required for such analysis and adjustment. In typical game development, often with limited development time, it is not possible to adjust and analyze game balance completely. Therefore, an automatic method for analyzing the game balance is required. So far, existing automatic analysis methods need significant computational time to analyze a situation, or require specialized knowledge obtained by humans based on the game title in advance. Therefore, a large amount of computational time is required when applying those systems to games with many situations, of which knowledge is not yet acquired by humans. In this paper, we propose a method that uses Stochastic Genetic Algorithm (StGA) for analyzing that sort of games. From experiments using FightingICE, which is used as the platform for an international AI tournament, the effectiveness of our method is confirmed.

Keywords: game balance analysis, game balance visualization, genetic algorithms, reinforcement learning, game design support

¹ 立命館大学情報理工学部
College of Information Science & Engineering, Ritsumeikan University, Kusatsu, Shiga 525–8577, Japan

^{†1} 現在, 立命館大学大学院情報理工学研究科
Presently with Graduate School of Information Science & Engineering, Ritsumeikan University

^{†2} 現在, 株式会社ディンプス
Presently with Dimps Corporation

^{a)} ruck@ci.ritsumeik.ac.jp

1. はじめに

本稿では、対戦型ゲームの開発における重要な要素であるゲームバランスの分析に着目する。ゲームバランスを「戦略の優位性の公平さ」として定義する。ここで、戦略とは利得獲得を目的としたプレイヤーの行動の時系列と定義する。

ゲームおよびキャラクターなどの状態から構成される、あるゲーム状況で、他プレイヤーの行動にかかわらずプレイヤーにとって他のすべての戦略よりも大きい利得が得られるような戦略を支配的戦略と呼ぶ。対象ゲームの展開を通じて、ほとんどの状況において支配的戦略となるような戦略を強い戦略と定義し、強い戦略が存在するような場合にはそのゲームのゲームバランスが悪いと評価する。逆に強い戦略がない場合にはゲームバランスが良いと評価する。

ゲームバランスが悪い場合には、プレイヤーは強い戦略をつねに実行することで容易に勝利することができる。つまり、ゲームバランスが悪い場合、ゲームに勝つ方法が単純なものとなり面白くないゲームとなる。逆に、良いゲームバランスを持つゲームはプレイヤーにとって勝つための戦略を状況に応じて獲得する必要があるため、複雑で面白いゲームとなる。また、AI（ルールベースならびに木探索といった人工知能の手法などによって操作される、人間の操作しないキャラクターを指す）の研究を行うためのプラットフォームを開発する際は、良いゲームバランスを保証することで、AIの研究者に対してAIにバランスの穴を突かせず状況に応じて戦略を獲得させる、より普遍性の高い研究を要求することができる。

現在、ゲーム開発におけるゲームバランスの分析、調整は主に人の手によって行われているが、それには時間的コストがかかる。このため、開発期間に限りがあるゲーム開発の現場では、十分にゲームバランスの調整を行うことができない場合も少なくない。ゲームを開発する際に、エンターテインメント性を機械的な手法のみで事前評価することは難しい。しかし、競技性を持つゲームにおいては、デザイナーが戦略の強弱の情報を事前に得ることで、調整項目を想定した上で調整を行えるため、コストの削減を図ることができ、よりゲームバランスの良い、面白いゲームを開発しやすくなると考えられる。

しかし、ゲームのすべての状況における支配的戦略を機械的に抽出することは一般的には難しいとされている。単純なゲームであれば戦略空間は小さなものであるが、格闘ゲームのように複雑なゲームになるとその戦略空間は大きくなり、空間全体を探索するには膨大な計算時間を要する。このため、強い戦略を効率良く探索し、ゲームバランスの分析、可視化を自動的に行う手法が必要である [1]。

2. 関連研究

文献 [1] においては、分析対象のゲーム上で共進化アルゴリズムの2つの母集団に基づいて動作する、ゲーム環境観察用 AI（以下エージェント AI）を互いに対戦させながら、最適な戦略を選択する遺伝子を持つ個体を探索させる。その後、世代ごとに得られた最大適応度の個体の持つ遺伝子情報を Heat Map という、座標と遺伝子情報の種類を、適応度と色の濃淡を対応付ける手法によって図示することで戦略の強弱関係の可視化を行っている。

文献 [2] においては、共進化アルゴリズムと遺伝的プログラミング、ロジスティック回帰分析を利用して、自動的にゲームバランスの良いゲームを生成するシステムを提案している。ゲームバランスの良いゲームの自動生成という研究目的は本研究とは研究対象は異なるが、ゲームのパラメータを設計する過程においてゲームバランスの自動分析を行っている。

近年のゲームバランスについての研究 [3] においては、複数の戦略を設定されているキャラクターの組からなるチームの強弱のバランスを自動分析するシステムを提案している。Artificial Neural Network (ANN) をキャラクターごとに用意し、教師なしで学習させる。それらの ANN に基づき操作されるキャラクターらをゲーム上で動作させた場合の勝率の観点からゲームバランスを分析している。しかし、本研究が対象とする戦略間の強弱関係のバランスに対しての自動分析手法とは分析対象が異なる。

2.1 既存手法の課題点

文献 [1] の手法は、共進化アルゴリズム [4] に基づいて、着目母集団における個体の適応度評価を行う際に対戦相手の母集団からサンプリングされた代表個体と呼ばれる個体との個体の関係性から適応度評価を行う。母集団中のすべての個体について、ゲームの処理に基づいたシミュレータ上で、評価対象の個体に示された戦略と対戦相手の個体に示された戦略をそれぞれの AI に実行させ、その結果から適応度の評価を定めている。このため、1世代あたりの適応度評価には母集団の大きさとゲームの処理時間に比例する計算時間を必要とする。1世代あたりの適応度評価ではゲームの状況を変更せずに各個体の評価を行うため、多くのゲーム状況を分析するためには大きな世代数を必要とする。

文献 [2] では、個体の評価関数を設定するためにロジスティック回帰分析を用いている。これによって個体の評価に必要な時間が削減されている。しかし、ロジスティック回帰分析を行うためには、状況に応じて適切な動作を行う人間や AI などから回帰分析に必要な教師データを取得する必要がある [5]。ゲームの専門的な知識をすでに獲得していることが前提となるため、事前にゲームのついでに分

析を人間が行う必要がある。このため、ゲームバランスの自動的な分析という目的には適さない。

2.2 課題点への対策

本稿では、既存研究の課題点を StGA [6] に基づき操作されるエージェント AI の利用による解決を試みた。StGA は、1 つの個体が戦略を実行して得られた結果を元に母集団のすべての個体について評価を更新する学習というプロセスと個体に遺伝的操作を適応するプロセスを繰り返すことで最適な遺伝子を持つ個体を探索する、強化学習と遺伝的アルゴリズムの双方に着想を得たアルゴリズムである。StGA は環境の利得行列が不可視、動的に変化する場合にも、利得を得られる確率が最も大きくなるような遺伝子を持つ個体に個体分布が収束する [7]。共進化アルゴリズムと同様に強い戦略が存在する場合、その戦略を選択するような遺伝子を持つ個体に個体分布が収束するため、ゲームバランスの分析に用いることができると考えられる。

また、StGA では 1 回の学習ごとにゲームの状況を更新する。1 つの個体の戦略実行結果に着目し、母集団のすべての個体に対し適応度の更新を行うため、学習においてゲームの処理は 1 回のみ行われる。このため、StGA をエージェント AI に適用することで、既存手法と比較して、同じゲームの処理の実行回数でもより多くのゲーム状況を分析することができる。また、ゲームの処理に基づく結果から学習を行うため個体の適応度の評価に専門的な知識を必要としないという利点がある。

3. 提案手法

本章では、StGA を利用したゲームバランスの分析手法について述べる。本手法の概念図を図 1 に示す。提案手法は、文献 [1] において提案されたゲームバランス分析手法に基づく。

エージェント AI は母集団から適応度に応じて確率的に個体を選択する。選択した個体の遺伝子に則って戦略を選択し実行する。その結果をゲーム環境から取得し、個体について適応度更新（以下、選択から適応度更新までの工程

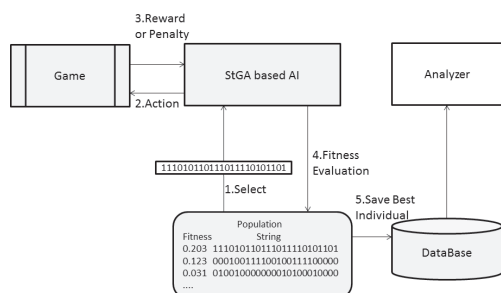


図 1 ゲームバランス分析手法の概念図

Fig. 1 The conceptual diagram of the game balance analysis method.

を学習とする)を行う。一定回数(以降 n)の学習が終わるたびに、母集団の中で最も高い適応度を持つ個体をデータベースに登録した後、遺伝的操作を行う。一定の世代数(以降 m)と同じ数の個体がデータベースに保存されるとゲームプロセスを終了する。その後、図 1 における Analyzer にてデータベースに保存された個体の持つ遺伝子の情報から戦略の強弱の可視化を行う。

3.1 エージェント AI

本節では StGA を利用したエージェント AI の遺伝子構造、適応度の初期化とその更新方法、遺伝的操作についての詳細を述べる。

3.1.1 遺伝子構造

ゲームバランスの自動分析に StGA を適用するため、遺伝子を戦略の選択頻度として符号化する。ゲーム中にプレイヤーが選択可能な全戦略の集合を S 、母集団を POP 、その i 番目の要素を POP_i とする。個体は $8 \cdot |S|$ のビットからなる数字列で表現され、8 ビットごとに対応する行動の選択度合を符号なし 2 進数として表現する。 i 番目の戦略 S_i に対応するビット列を BS_{S_i} とすると、その個体における S_i の選択確率 P_{S_i} は

$$P_{S_i} = \text{BIN}(BS_{S_i}) / \sum_{k=0}^{|S|-1} \text{BIN}(BS_{S_k}) \quad (1)$$

として計算される。式中の $\text{BIN}()$ は引数であるビット列を 2 進数として解釈した正値を返す関数である。

図 2 に遺伝子構造と戦略選択の関係を表す。この構造によって、多くのビットが 1 となっているビット列に対応する戦略の選択確率は高くなり、多くのビットが 0 となっているビット列に対応する戦略の選択確率は低くなる。

3.1.2 適応度の初期化と更新手法

母集団の各個体はランダムに 0 か 1 で初期化され、初期適応度は $f_i(t=0) = 1/r$ ($i = 1, \dots, r; t$ は更新回数; r は母集団中の個体数 = $|POP|$) として与えられる。

StGA では、Stochastic Learning Automata の Linear

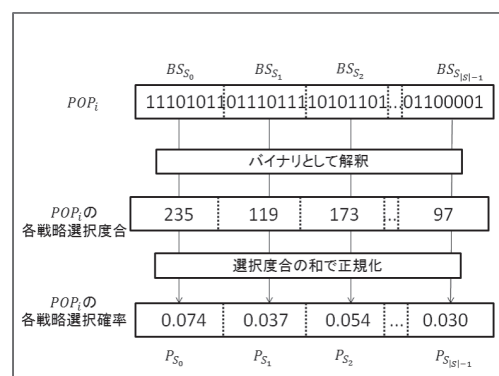


図 2 遺伝子構造と戦略選択の関係

Fig. 2 Relationship of strategy selection and the gene structure.

Reward-Penalty-Scheme (L_{R-P}) を用いて母集団内におけるすべての個体の適応度の再評価を行う。本来の L_{R-P} では得られる結果を2値として仮定しているが、ゲームによっては得られる結果が2値ではなく実数値や整数値として得られ、その大きさが戦略の強弱に関係する場合がある。このため、以下のように利得の大きさを考慮した適応度更新を行えるように L_{R-P} に変更を加える。

i 番目の個体が戦略を行った場合に、得られた利得が正の値である場合には式 (2) を、得られた利得が負の値である場合には式 (3) を用いて個体の評価を行う。得られた利得が0である場合は学習を行わない。

$$f_i(t+1) = f_i(t) + \left(\frac{R}{R_{max}}\right) \sum_{i \neq j}^r \alpha f_j(t)$$

$$f_j(t+1) = f_j(t) - \left(\frac{R}{R_{max}}\right) \alpha f_j(t) \quad (\forall j \neq i) \quad (2)$$

$$f_i(t+1) = f_i(t) - \left(\frac{R}{R_{min}}\right) \sum_{i \neq j}^r \left\{ \frac{\beta}{(r-1)} - \beta f_j(t) \right\}$$

$$f_j(t+1) = f_j(t) + \left(\frac{R}{R_{min}}\right) \left\{ \frac{\beta}{(r-1)} - \beta f_j(t) \right\} \quad (\forall j \neq i) \quad (3)$$

$f_k(t+1)$ は k ($k = 1, \dots, r$) 番目の個体における更新後の適応度を指し、 $f_k(t)$ は k 番目の個体における更新前の適応度を指す。また α , β はそれぞれ利得が正の場合、負の場合の学習率を指す。ここで、 R は得られた利得の大きさである。 R_{max} は得られた利得の中で最も大きかったものを指し、 R_{min} は得られた利得の中で最も小さかったものを指す。

R が正の値である場合には、戦略を実行した k 番目の個体の適応度を上昇させ、他の個体の適応度を減少させる。逆に、 R が負の値である場合には、 k 番目の適応度を減少させ、他の個体の適応度を上昇させた上で、母集団内のすべての個体に対し適応度の平均化を行う。いずれの場合にも適応度の総和は一定に保たれる。

α および β が0の場合には、それぞれ R が正の値の場合、負の値の場合に適応度を更新しないことを意味する。 $\frac{R}{R_{max}}$, $\frac{R}{R_{min}}$ の項によって、 R の大きさを過去に得られた R の最大値および最小値によって正規化した値が適応度の更新量に対して乗算される。この構造によって得られた利得の絶対値が大きいほど、適応度をより大きく修正する機能を持たせることができる。

3.1.3 遺伝的操作

遺伝的操作は、交叉、突然変異、選択の3つの操作からなる。

交叉では母集団の中からランダムに2つの個体を選択し、その片方の個体のビット列から $[0, 8 \cdot |S| - 1]$ の範囲でランダムに選ばれた番号までのビットをコピーし、残りの番号のビットを他方の個体のビットからコピーした新しい

1つの個体を作成する。この個体の適応度は交叉操作で選択された個体の適応度の平均値とする。

突然変異では、母集団の中からランダムに選択した1つの個体を対象にそれぞれのビットを P_m の確率で反転させた新しい1つの個体を作成する。この個体の適応度は突然変異操作で選択された個体の適応度をそのまま継承する。

選択では、交叉、突然変異によって生成された個体を母集団の中で最も適応度の低い2個体と入れ替え、母集団内の各個体の適応度の総和から各個体の適応度の正規化を行う。

3.2 Analyzer

本研究で用いた分析器について述べる。まず、要素数が $|S|$ の *Score* という配列の全要素の値を0とする。データベースに保存されたすべての個体について、最も高い選択度合を持つ戦略と同じ番号の *Score* の要素値を1上昇させる処理を行う。その後、*Score* の各要素値を m で割ることでその範囲を $[0, 1]$ とする正規化処理を行う。得られた *Score* の各要素値 (以下獲得 *Score*) についてグラフとして出力することで可視化を行う。

StGA を用いたゲームバランスの分析手法のアルゴリズムを Algorithm 1 に示す。Algorithm 中の *DataBase* は各世代で保存された最大適応度個体の集合であり、 $DataBase_i$ は *DataBase* に i 番目に保存された個体を指す。

4. FightingICE

本章では、実験に利用する“FightingICE”に関する説明を行う。FightingICEは対戦型格闘ゲームに属するゲームであり、対戦型格闘ゲームAIのコンテスト*1 (図3) に利用されている [8]。対戦型格闘ゲームは、複数のプレイヤーが操作するキャラクター達が攻防を行い、キャラクターごとに設定された耐久値 (以下 *HP*) を減らしあい、規定時間後にキャラクターの残り *HP* を比較して勝敗を決定するデ

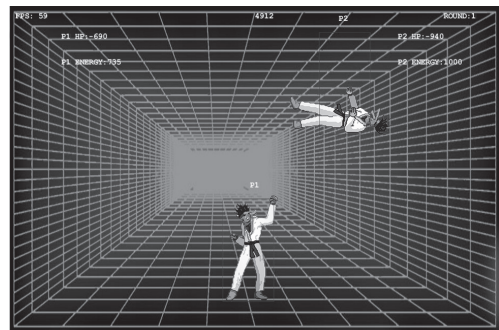


図3 2013年大会の1位AI対2位AIの対戦様子
 Fig. 3 Screenshot of a fight between the 1st-rank AI and the 2nd-rank AI at the 2013 competition.

*1 FTGAI Competition, <http://www.ice.ci.ritsumeai.ac.jp/~ftgaic>

Algorithm 1 Algorithm of the game balance analysis method.

```

//ゲーム上での学習と遺伝的操作
t ← 0
Rmin ← 0
Rmax ← 0
DataBase ← ∅
for i ← 0 to r - 1 do
    Initialize POPi
    fi(t) ← 1/r
end for
for i ← 0 to m - 1 do
    for j ← 0 to n - 1 do
        POPAct ← Roulette wheel selection(POP)
        R ← Action(POPAct)
        if R < Rmin then
            Rmin ← R
        end if
        if R > Rmax then
            Rmax ← R
        end if
        if R ≠ 0 then
            //式 (2) または式 (3) を用いた適応度評価
            UpdateFitness(POP, t, R, α, β)
        end if
    end for
    Add Best(POP) to DataBase
    Crossover
    Mutation
    Selection
end for
//Analyzer の処理
for i ← 0 to |S| - 1 do
    Score[i] ← 0
end for
for i ← 0 to m - 1 do
    s ← argmaxx(BIN(DataBasei.BSSx))
    Score[s] ← Score[s] + 1
end for
for i ← 0 to |S| - 1 do
    Score[i] ← Score[i]/m
end for
Output(Score)
    
```

デジタルゲームの一種である [9]。模倣学習による人間らしい AI [10] やゲームの難易度調整を自動的に行う AI [11], [12] など、研究対象として広く利用されている。さらに、状況の数が大きく、ゲームバランス分析の検証実験に適すと考えられる。

格闘ゲームではキャラクターがプレイヤーの操作に応じて、移動、攻撃、防御などの行動をとる。キャラクターには座標と、それに対応する hit box が定められており、移動や攻撃などの行動によってその値が変化する。キャラクターが攻撃を行うと、発生、持続、硬直の3つのプロセスが行われる。持続の段階において attack hit box が発生する。この間、キャラクターはプレイヤーからの新たな入力を受け付けない。キャラクターの hit box と相手のキャラ

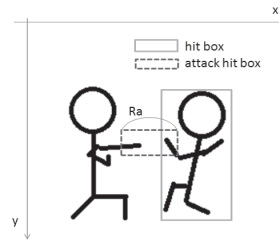


図 4 攻撃に成功

Fig. 4 Successful attack.

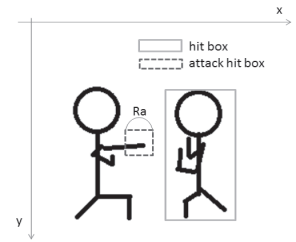


図 5 攻撃を失敗

Fig. 5 Failed attack.

クターの attack hit box が重なると、自身のダメージの計算が行われる。これらの処理を繰り返し行い、相手に定められた耐久値（最大 HP）以上のダメージを与えるか、もしくは規定時間後に残り HP が相手キャラクターのそれより大きいと勝利となる。FightingICE は、例外的に規定時間後に残り HP が相手キャラクターよりも大きいのみを勝利条件としている。

FightingICE では新しいキャラクターを作成できる。後述するそのキャラクターの行動のパラメータも自由に設定できるが、そのパラメータの種類は既存の格闘ゲームの中でも典型的なものを採用している。本研究では、提案手法の検証のために行動の強弱に寄与するパラメータを再設定した。

4.1 パラメータ

キャラクターの行動には、attack hit box が発生するものと、そうでないものがある。行動には空間などに関する複数のパラメータが存在する。さらに、attack hit box が発生する場合には、それに加えて威力などに関するパラメータが設定される。ここでは、実験に関係のあるパラメータについて説明を行う。

●威力に関するパラメータ

威力に関するパラメータは attack hit box が相手の hit box に重なった場合に与えるダメージの値 Da である。 Da が大きいほど大きな利得を得ることができるため、強い戦略となる。

●空間に関するパラメータ

空間に関するパラメータは attack hit box の x 方向の大きさを表す値 Ra である。図 4 および図 5 に hit box, attack hit box の攻撃の成否の関係を示す。図中にて hit box および attack hit box はそれぞれ実線と点線で示されている。図 4 では attack hit box と hit box の範囲が重なっているため攻撃に成功している。しかし、図 5 では attack hit box と hit box の範囲が重なっていないため攻撃に失敗している。 Ra が大きいほど攻撃を与える条件を満たしやすいため、より強い戦術となる。

表 1 設定した定数

Table 1 Constant set.

	$ S $	r	α	β	n	m	P_m
定数名	戦略数	個体数	正の学習率	負の学習率	学習数	世代数	突然変異確率
設定値	25	30	0.01	0.01	100	2,000	0.1

5. 実験

戦略と行動を 1 対 1 に対応させ、それらの戦略の強弱を提案手法が分析できるかを 4 つの実験により検証する。分析手法に用いる定数は表 1 に示したものをを用いる。戦略数以外のパラメータは予備実験を行い、進化過程において最適なものを選んだ。戦略集合は、FightingICE の代表的な行動からなる。これらの選出は移動、防御にあたる行動すべてと、いくつかの攻撃にあたる行動を対象とした。攻撃にあたる行動の選出は、地上攻撃と空中攻撃、しゃがみ攻撃の数の関係を地上攻撃 > 空中攻撃 > しゃがみ攻撃となるように設定した。これは一般的な格闘ゲームの行動の数の関係に基づく。この結果、戦略数 $|S|$ は 25 となった。

実験では、各戦略の獲得 $Score$ の 100 試行における平均を調べる。このとき、エージェント AI との対戦相手の AI に異なる振舞いを持つ 2 種類の AI (StGA に基づく操作とランダムな戦略選択に基づく操作) を用いた実験から本手法の有用性を検証する。

ゲーム環境から得られる R は、式 (4) で表されるように、戦略実行前と戦略実行後の自分の HP (以下 HP_{My}) と相手の HP (以下 HP_{Opp}) の差の加減度合である。

$$R = (HP_{My}(t+1) - HP_{Opp}(t+1)) - (HP_{My}(t) - HP_{Opp}(t)) \quad (4)$$

ここで、 t および $t+1$ とは戦略実行前、戦略実行後のゲーム内時間を指す。 HP は初期値を 0 とし、ダメージを受けると値が減少する。

実験 1 では、1 つの戦略を除いて、すべての戦略に正の値の R (以下報酬) を獲得する手段がないように設定する。唯一の戦略が報酬を得る手段を持つため、この戦略は強い戦略となる。この実験によって、強い戦略の獲得 $Score$ が高くなるかを確かめる。残る 3 つの実験では、2 つ、または 4 つの戦略に報酬を得る手段を設定するが、これらの戦略のパラメータに強弱関係が発生するように設定したゲームを対象に実験を行う。この実験によって、性質の異なるパラメータでの強弱関係によって獲得 $Score$ に変化があるかも検証する。

各実験での attack hit box に関するパラメータの Da と Ra は表 2 にまとめる。attack hit box が設定される行動のみ相手にダメージを与えることができる。基準となる $STAND_A$ の Da は 100 とし、 Ra はキャラクターの画像の大きさである 256 pixel の半分の 128 とした。以下に、各

表 2 実験環境のパラメータ

Table 2 Parameters of the experimental environment.

実験	attack hit box 設定行動	Da	Ra
実験 1	$STAND_A$	100	128
実験 2	$STAND_A$	200	128
	$STAND_B$	100	128
実験 3	$STAND_A$	100	256
	$STAND_B$	100	128
実験 4	$STAND_A$	200	256
	$STAND_B$	200	128
	$STAND_FA$	100	256
	$STAND_FB$	100	128

実験の詳細を示す。

● 実験 1

$STAND_A$ という行動に attack hit box が発生するようにパラメータを設定し、残りのすべての行動には attack hit box が発生しないようにパラメータを設定する。 $STAND_A$ にのみ、報酬を得る手段が設定されるため、 $STAND_A$ に獲得 $Score$ が集中すると予想される。

● 実験 2

$STAND_A$ に加えて $STAND_B$ という行動にも、attack hit box が発生するようにパラメータを設定し、残りのすべての行動には attack hit box が発生しないようにパラメータを設定する。 $STAND_B$ の Da は $STAND_A$ の半分であるため、攻撃が成功した際得られる報酬が $STAND_A$ よりも小さくなる。このため、 $STAND_A$ の方が $STAND_B$ よりも強い戦略となる。 $STAND_B$ の獲得 $Score$ が、 $STAND_A$ の獲得 $Score$ よりも低ければ正しい分析が行えているといえる。

● 実験 3

実験 2 と同じように $STAND_A$ と $STAND_B$ という行動にしか、attack hit box が発生しないようにパラメータを設定するが、 Da ではなく Ra に差をつける。 $STAND_B$ の Ra は $STAND_A$ の半分であるから、 $STAND_A$ と比較して、報酬を得ることのできる状況が少なくなる。このため、 $STAND_A$ の方が $STAND_B$ よりも強い戦略となる。 $STAND_B$ の獲得 $Score$ が、 $STAND_A$ の獲得 $Score$ よりも低ければ正しい分析が行えているといえる。

● 実験 4

実験 4 では $STAND_A$, $STAND_B$, $STAND_FA$

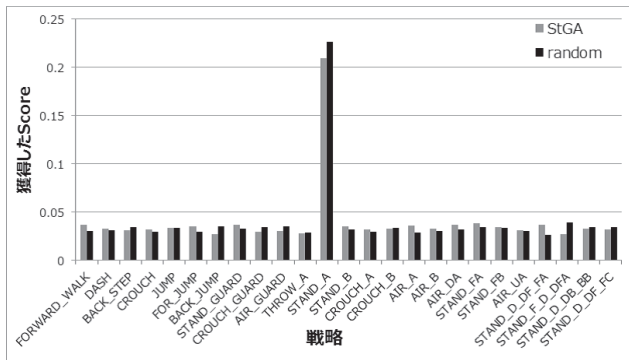


図 6 実験 1 での各戦略の獲得 Score の平均

Fig. 6 The average of Score earned for each strategy in Experiment 1.

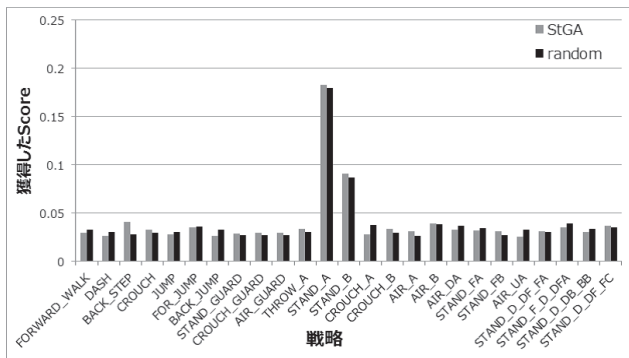


図 7 実験 2 での各戦略の獲得 Score の平均

Fig. 7 The average of Score earned for each strategy in Experiment 2.

と *STAND_FB* という行動でのみ attack hit box が発生するように設定する。その上で D_a と R_a の値について、それぞれ 2 種類の値を用意し、その組合せを各行動に設定する。*STAND_A* のパラメータにはともに高い値の組を、*STAND_FB* には低い値の組を設定する。それぞれの獲得 Score が最も高い値、attack hit box を設定した戦略の中で最も低い値になれば正しく分析を行えているといえる。

5.1 結果

● 実験 1

図 6 に各戦略の獲得 Score のグラフを示す。StGA が対戦相手の場合の *STAND_A* の獲得 Score の平均は 0.210 である。*STAND_A* を除く戦略の中で最も値が高い戦略であっても、その値の平均は 0.038 である。ランダム操作が対戦相手の場合の *STAND_A* の獲得 Score の平均は 0.226 であり、その他の戦略の獲得 Score にも StGA が対戦相手の場合と同様の傾向が見られた。単一の強い戦略が存在するケースにおいて、相手の AI にかかわらず獲得 Score が強い戦略に集中することが分かる。

● 実験 2

実験 2 の結果を図 7 に示す。StGA が対戦相手の場合

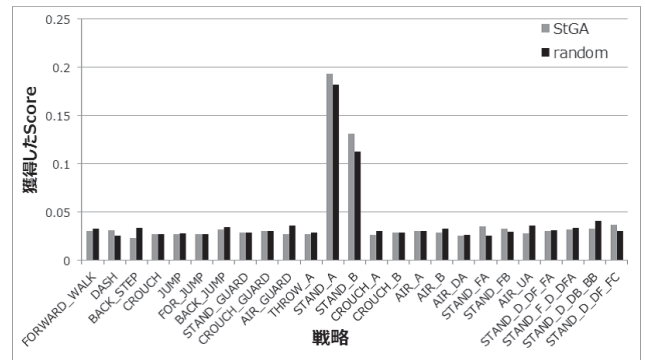


図 8 実験 3 での各戦略の獲得 Score の平均

Fig. 8 The average of Score earned for each strategy in Experiment 3.

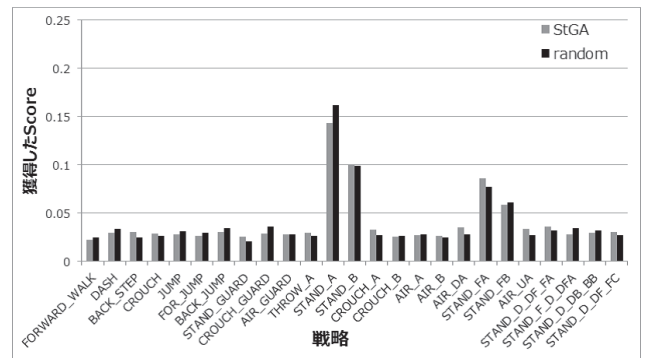


図 9 実験 4 での各戦略の獲得 Score の平均

Fig. 9 The average of Score earned for each strategy in Experiment 4.

の *STAND_A* および *STAND_B* の獲得 Score の平均はそれぞれ 0.183, 0.091 である。ランダム操作が対戦相手の場合の *STAND_A* および *STAND_B* の獲得 Score の平均はそれぞれ 0.180, 0.087 である。

● 実験 3

図 8 に各戦略の獲得 Score のグラフを示す。StGA が対戦相手の場合の *STAND_A* および *STAND_B* の獲得 Score の平均はそれぞれ 0.193, 0.131 である。ランダム操作が対戦相手の場合の *STAND_A* および *STAND_B* の獲得 Score の平均はそれぞれ 0.182, 0.113 である。

● 実験 4

実験 4 の結果を図 9 に示す。StGA が対戦相手の場合の *STAND_A*, *STAND_B*, *STAND_FA* および *STAND_FB* の獲得 Score の平均はそれぞれ 0.143, 0.101, 0.086, 0.059 である。ランダム操作が対戦相手の場合の *STAND_A*, *STAND_B*, *STAND_FA* および *STAND_FB* の獲得 Score の平均はそれぞれ 0.162, 0.099, 0.077, 0.061 である。

5.2 考察

実験 1 の結果より、提案手法は単一の強い戦略が存在する場合に、強い戦略と弱い戦略の分析および可視化を行う

ことができているといえる。また、実験2, 3の結果より、提案手法は報酬を得ることのできる複数の戦略が存在する場合に、パラメータの種類にかかわらず、その強弱に応じて正しく獲得 Score の値を振り分けることができているといえる。さらに、実験4の結果より、複数種類のパラメータの組合せによって強弱関係が生じる場合にも、正しく獲得 Score を振り分けることができているといえる。

これらの結果より、StGA を適応した AI を利用すれば、AI にゲームタイトル依存の専門的な知識を埋め込まなくとも強い戦略と弱い戦略の分析を行うことができるといえる。

6. まとめ

本稿では、対戦型ゲームにおける戦略多様性の観点からゲームバランスを自動的に分析する手法について述べた。既存手法では、探索すべき状況の数が大きいゲームを分析する場合には、大きな計算時間がゲーム内容に関する事前知識が必要であった。このため、著者らは強化学習アルゴリズムと遺伝的アルゴリズムに基づく StGA をエージェント AI に利用することでこれらの課題点の解決にあたった。FightingICE を用いた実験から、提案手法によるゲームバランスの分析が可能であることを示した。

今後は、Ms.Pac-Man [13] やリアルタイム戦略ゲーム [14] などの、格闘ゲーム以外へのゲームジャンルへの適用した場合の有用性を検証する。さらに、これらのゲームを対象としたゲームバランスの自動調整システムをゲームメーカーと共同で開発する予定である。

参考文献

- [1] Leigh, R., Schonfeld, J. and Louis, S.J.: Using co-evolution to understand and validate game balance in continuous games, *Proc. 10th Annual Conference on Genetic and Evolutionary Computation*, pp.1563–1570, ACM (2008).
- [2] Chen, H., Mori, Y. and Matsuba, I.: Evolutionary Approach to Balance Problem of On-Line Action Role-Playing Game, *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pp.1–4, IEEE (2012).
- [3] Fang, S.-W. and Wong, S.-K.: Game team balancing by using particle swarm optimization, *Knowledge-Based Systems*, Vol.34, pp.91–96 (2012).
- [4] Hillis, W.D.: Co-evolving parasites improve simulated evolution as an optimization procedure, *Physica D: Non-linear Phenomena*, Vol.42, No.1, pp.228–234 (1990).
- [5] Alpaydin, E.: *Introduction to Machine Learning* (2010).
- [6] Munetomi, M., Takai, Y. and Sato, Y.: *StGA: An application of a genetic algorithm to stochastic learning automata*, Vol.27, No.10, Wiley Online Library (1996).
- [7] 富川裕樹, 棟朝雅晴, 高井昌彰: 利得行列が不可視である行列ゲームへの StGA の応用, *電子情報通信学会論文誌 D-II, 情報・システム, II-情報処理*, Vol.80, No.2, pp.700–702 (1997).
- [8] Lu, F., Yamamoto, K., Nomura, L.H., Mizuno, S., Lee,

- Y. and Thawonmas, R.: Fighting game artificial intelligence competition platform, *2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, pp.320–323, IEEE (2013).
- [9] 梶並知記: 対戦型格闘ゲームプレイヤーの戦略的思考の分析に関する一方法論, *日本デジタルゲーム学会 2011 年次大会*, pp.124–132 (2012).
- [10] 星野准一, 田中彰人, 濱名克季ほか: 模倣学習により成長する格闘ゲームキャラクタ, *情報処理学会論文誌*, Vol.49, No.7, pp.2539–2548 (2008).
- [11] 中川明紀, 柴崎智哉, 逢坂翔太, Thawonmas, R.: ニューラルネットワークによる格闘ゲーム AI の難易度調整及び行動多様性向上手法, *ゲーム学会和文論文誌*, Vol.3, No.1, pp.35–40 (2009).
- [12] Cho, B.H., Jung, S.H., Seong, Y.R. and Oh, H.R.: Exploiting Intelligence in Fighting Action Games Using Neural Networks, *IEICE Trans. Inf. and Syst.*, Vol.E89-D, No.3, pp.1249–1256 (2006).
- [13] Nguyen, K.Q. and Thawonmas, R.: Monte-Carlo Tree Search for Collaboration Control of Ghosts in Ms. Pac-Man, *IEEE Trans. Computational Intelligence and AI in Games*, Vol.5, No.1, pp.57–68 (2013).
- [14] Nguyen, K., Wang, Z. and Thawonmas, R.: Potential Flows for Controlling Scout Units in StarCraft, *Proc. 2013 IEEE Conference on Computational Intelligence and Games (CIG2013)*, pp.344–350 (2013).



山本 界人

2014 年立命館大学情報理工学部知能情報学科卒業。現在、同大学大学院情報理工学研究科情報理工学専攻博士課程前期課程に在籍。ゲームバランスの自動分析手法の研究に従事。



水野 竣介

2014 年立命館大学情報理工学部知能情報学科卒業。同年 4 月より株式会社ディンプスに勤務。在学中、ゲーム内キャラクターの自動制御および対戦相手の行動予測の研究に従事。



ターウォンマット ラック
(正会員)

1994 年東北大学大学院工学研究科情報理工学専攻博士課程修了。博士(工学)。2004 年 4 月より立命館大学情報理工学部知能情報学科教授。計算知能およびゲーム AI の研究に従事。