

# Memetic Algorithmに基づくプレイヤーの習熟度を 考慮したゲームバランス調整機構の試作

鴛海太一†

† 静岡大学情報学部情報科学科

福田直樹‡

‡ 静岡大学 学術院情報学領域

## 1 はじめに

ゲームコンテンツの増大化に伴い、ゲーム開発の大規模化が進んでいる。ゲーム開発における作業を効率的にすることが求められているが、品質を維持しつつ効率的にすることは非常に困難であることが予想される。この問題を解決するために、Procedural Content Generation (PCG) が研究されている [1]。PCG とは、アルゴリズムを使ってゲームコンテンツを自動的に生成することである [2]。ゲームコンテンツとは、レベル、マップ、ルール、テクスチャー、アイテムなどのゲームに含まれるもの全てである。

本研究では、PCG の新たな問題としてゲームバランスの自動調整に着目する。本研究におけるゲームバランスは、ゲーム内のパラメータの偏りであると定義する。パラメータは、一部のゲームコンテンツに付随する数値として表現される。もっとも代表的なパラメータとしては、キャラクターの能力値としてのパラメータがある。パラメータをキャラクターごとに変えることによって、キャラクターに特徴を与えることができるが、異なる種類のキャラクターを扱えるようにした上でゲーム難易度を適切に調整することは容易ではない。

本研究では、その題材となるゲームとして個々のキャラクターが多くのパラメータを持つ格闘ゲームを使用し、習熟したプレイヤーがプレイした場合にもゲームバランスを調整可能なゲームバランス調整機構の試作について述べる。

## 2 ゲームバランス調整機構の構成

本研究で作成するゲームバランス調整機構は、ゲーム開発者がキャラクターの特徴とパラメータの種類を与えた際に、それに合わせたパラメータを調整を行えるようにすることを目的にしている。本研究では、習熟したプレイヤーがプレイした場合にもバランス調整を可能とするために、Memetic Algorithm によるバランス

調整を行えるようにし、プレイヤーの習熟を再現するために、強化学習を使用してキャラクター AI を逐次生成する機構を用意する。パラメータの種類や特徴等は、本研究にあたり新たに設計した記述言語 Fighting Game Description Language (FGDL) によって記述される。

## 3 ゲームバランス調整機構の実装

ゲームバランス調整機構には、パラメータ調整機構と強化学習器という2つのモジュールがある。この2つのモジュールの具体的な実装方法を、次に述べる。

本研究で試作するパラメータ調整機構は、Memetic Algorithm を用いてパラメータを最適化する。本研究では、格闘ゲームにおいて複数の特徴を持つキャラクターが存在する場合に、それらのキャラクターの技のパラメータを最適化することを目的とする。このパラメータ調整機構で用いる Memetic Algorithm での個体は、全てのキャラクターのパラメータをまとめたものとする。個体にするパラメータは整数であるため、それを2進数にエンコードし、遺伝的操作を行う。Memetic Algorithm は、Hybridise Evolutionary Algorithm の一種であり [3]、Genetic Algorithm との相違点は、交叉や突然変異などの遺伝的操作に、解く問題特有の知識に基づいた実装、または局所探索法を使用した実装を含むことである。本研究では、交叉と突然変異に変更を加える。

本研究で実装する Memetic Algorithm では、交叉と個体の改善に2つの評価値として Evaluation A と Evaluation B を使用する。式(1)は、この2つの評価値を求める際に使用する式である。式(1)中の  $w$  は、その個体内のキャラクターで総当たり戦をした時の各キャラクターの勝率とする。

$$Eval = \begin{cases} 2w & (0 \leq w \leq 0.5) \\ -2w + 2 & (0.5 < w \leq 1.0) \end{cases} \quad (1)$$

式(1)では、総当たり戦をした時の全ての対戦を評価する。Evaluation A では、これらの評価値を合計した値をその個体の評価値とする。Evaluation B では、個体ごとではなく、キャラクターごとに評価値を割り当てるため、個体内の各キャラクターについて、そのキャラ

A Preliminary Approach for Skill-aware Video Game Parameter Adjustment using Memetic Algorithm

†Taichi Oshiumi Department of Computer Science, Shizuoka University  
Email: cs13033@s.inf.shizuoka.ac.jp

‡Fukuta Naoki Department of Informatics, Shizuoka University  
Email: fukuta@cs.inf.shizuoka.ac.jp

クターが行った対戦の評価値で基準値を超えるものがいくつあるかを各キャラクターの評価値とする。

これらの評価値を使って交叉, 改善を行う。Evaluation A は, 交叉をする親を選ぶ際に使用され, Evaluation B は, 交叉をした後の子孫の改善に用いる。子孫の改善では, 各個体のもっとも Evaluation B による評価が低いキャラクターを, 局所探索法によって評価を高める。

習熟したプレイヤーに対してもゲームバランスが適切になるようにゲームバランスを調整するために, キャラクターの行動選択を決定する AI を強化学習によって個体ごとに作成する。これは, Memetic Algorithm の評価値として用いられる勝率を算出するための習熟したプレイヤー間での対戦をシミュレートする際に使用される。強化学習には Q-Learning[4] を使用する。また,  $\epsilon$ -greedy 法を使用し, 学習に多様性を持たせた。

#### 4 予備実験

本研究で試作するゲームバランス調整機構の特性を確かめるために, 独自に開発した格闘ゲームを使用し, 予備実験を行う。本予備実験で扱う対戦ゲームを, 以降では Simple Fighting Game (SFG) と呼ぶことにする。図 1 は SFG の様子である。

SFG では, 2 体のキャラクターを使用し, どちらかのキャラクターの体力が 0 以下になるか, あるいはゲーム開始から 60 秒経過することでこのゲームは終了し, 終了時のキャラクターの体力が多い方が勝者となる。本予備実験では, 1 秒間に 60 フレームのキャラクターの動作が可能のため, 1 回の対戦で最大 3600 フレームとなる。

キャラクターの持つパラメータには, 技の届く範囲, 技のヒット時に相手の体力を減らす量, 技を選択してから技が出るまでのフレーム数, 技が当たり判定を持つフレーム数, 技が終了してから次の行動を起こすことができるまでのフレーム数, 相手のガードを無視することができるかを表す数値の 6 つがある。キャラクターに特徴をつけるために, 「技を選択してから技が出るまでのフレーム数を 0 にする」「技が終了してから次の行動を起こすことができるまでのフレーム数を 0 にする」「相手のガードを無視することができるかを表す数値を 0 にする」という 3 種類の特徴のうち 1 つを各キャラクターに持たせられるようにした。したがって, 本予備実験では特徴の異なった 3 種類のキャラクターが用意される。

本研究では, Memetic Algorithm における集団の個体数を 10, 最大世代数を 1000, 交叉率を 0.6, 突然変異率を 0.05 とした。また, 強化学習の学習中に用いる  $\epsilon$ -

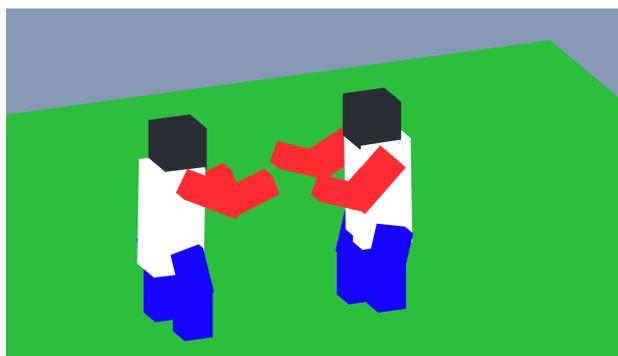


図 1: Simple Fighting Game の様子

greedy 法の  $\epsilon$  の値は 0.3 とした。本予備実験では, 1 世代あたり約 10 分かかり, 1 世代ごとに平均 115.96% 評価値が上昇することを観測した。

#### 5 おわりに

本研究では, 格闘ゲームにおける各操作キャラクターへの特徴付けを保ちながら, その操作に習熟したプレイヤーに対してもゲームバランスが適切になるように調整可能とする機構の試作について述べた。この機構では, パラメータ最適化中の各個体に習熟したプレイヤーの動作を強化学習によって学習させることによって, 習熟したプレイヤーがプレイした場合にもバランス調整が崩れないようにする機構の作成をした。

#### 参考文献

- [1] Khalifa, A., Perez-Liebana, D., Lucas, S. M. and Togelius, J.: General Video Game Level Generation, *Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO '16*, ACM, pp. 253–259 (2016).
- [2] Julian, T., Emil, K., David, S. and N., Y. G.: What is Procedural Content Generation?: Mario on the Borderline, *Proceedings of the 2Nd International Workshop on Procedural Content Generation in Games, PCGames '11*, ACM, pp. 3:1–3:6 (2011).
- [3] Eiben, A. and Smith, J.: Hybridisation with Other Techniques: Memetic Algorithms, *Introduction to Evolutionary Computation*, Springer, chapter 10, pp. 167–183 (2015).
- [4] Sutton, R. S. and Barto, A. G.: *Introduction to Reinforcement Learning*, MIT Press (1998).