

動的な難易度調整により対戦して楽しい格闘ゲーム AI

邓士达¹ 伊藤毅志¹

概要: 対戦ゲームにおいて、対戦相手の強さが適度であることは、楽しさを維持する上で重要であることは知られている。本研究では、動的に難易度を調整して適度な難易度を実現し楽しさを維持する格闘ゲーム AI の実現を目指す。従来の研究から、モンテカルロ木探索(MCTS)を用いて強さを調整する手法は提案されているが、十分な強さを実現するものではなかった。一方で、MCTS の候補手の生成において、遺伝的アルゴリズムを用いることで、有用な候補手だけに絞ることで探索を効率化し、強さを実現する研究もある。ここでは、この2つの研究を結びつけたプロトタイプシステム (TestAI) を構築する。評価実験として、MCTS だけを利用する動的難易度調整システム(MCTS_DDA)とこの TestAI の性能の比較を行った。その結果、TestAI は人間の上級者レベルのプレイヤーにも十分なレベルの強さを実現することができた。一方で、勝率を調べると、中級レベル以下の対戦相手には勝ちすぎてしまう可能性も示唆された。そこで、勝率を50%に近づけるために、攻撃の命中率も考慮した新たな難易度調整手法を考案した。

キーワード: モンテカルロ木探索、遺伝的アルゴリズム、動的難易度調整、格闘ゲーム

Fighting game AI with dynamic difficulty adjustment to make it fun to play against

DENG SHIDA^{†1} TAKESHI ITO^{†1}

Abstract: It is well known that the opponent's strength is important to maintain the fun in a fighting game. In this study, we try to develop a fighting game AI that dynamically adjusts the difficulty level to maintain a moderate level of difficulty and fun. Previous studies have proposed the use of Monte Carlo Tree Search (MCTS) to adjust the strength, but they have not been able to achieve sufficient strength. On the other hand, another study have used genetic algorithms in the generation of candidate moves in MCTS to streamline the search and achieve strength by focusing only on useful candidates. Here, we construct a prototype system (TestAI) that connects these two studies. As an evaluation experiment, we compare the performance of this TestAI with a dynamic difficulty adjustment system (MCTS_DDA) that uses only MCTS. The results show that the TestAI is able to achieve a sufficient level of strength for human advanced level players. On the other hand, examination of the win rate suggested that it might win too much against opponents below the intermediate level. Therefore, we devised a new difficulty adjustment method that also took into account the hit rate of attacks to bring the win rate closer to 50%.

Keywords: Monte Carlo tree search, genetic algorithms, dynamic difficulty adjustment, fighting game

1. はじめに

プレイヤーの没入感を維持することは、楽しいゲームを作るための重要な要素である。Chen は、Csikszentmihalyi が提唱するフローの概念を用いて[1]、図1のようなゲームにおけるフロー状態（没入感）を支える一つの要素として、適度なレベルのチャレンジングな状況を提示することが必要であることを説明している[2]。図1において x 軸はプレイヤーのゲームに対する能力を表し、y 軸はゲームの困難さを表している。この図において、ゲームの困難さがプレイヤーの能力を大きく上回っていると不安を感じ、逆に困難さに比べてプレイヤーの能力のほうが上回ってれば、退屈を感じてしまう。その丁度よい「FLOW ZONE」にスキルと難易度が入っていれば、プレイヤーがゲームを楽しむことができることを表している。このように、ゲームにおいて適度に難易度を調整することは、ひとつの大きな課題である。しかし、手動で難易度を選択するような旧来の手法（例えば

「初級」、「中級」、「上級」を三段階で切り替えるようなもの）では、レベルの中間の強さのプレイヤーにとっては、それぞれの難易度は簡単すぎたり難しすぎたりしてしまい、適度な難易度調整手法であるとは言えない。

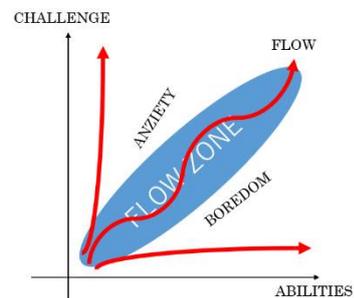


図1 ゲームにおけるフロー

Figure 1 Flow in games.

本研究では、格闘ゲームを題材にして、どのようなレベルのプレイヤーでもゲームを楽しくプレイできるような動的

¹ 電気通信大学
The University of Electro-Communications

難易度調整を実現するシステムの構築を目指す。

2. 関連研究

2.1 動的難易度調整

Ishihara らは、格闘ゲーム FightingICE を題材にして、モンテカルロ木探索 (MCTS) において動的難易度を調整する手法を提案した[3]。FightingICE は、オープンソースのリアルタイムの格闘型対戦ビデオゲームである。リアルタイム設定では、各プレイヤーには 16.67 ms ごとに更新された現在のゲーム状態が与えられる。Ishihara らの手法では、ヒットポイント (HP) に着目して、AI の HP と対戦相手の HP を比較して、AI の方が優勢の時には適度に難易度を調整したアクション (合法手) が選択され、AI の方が劣勢の時には相手に大きなダメージを与える攻撃性を高めるアクションを選択する。

この AI は、中級者レベル以下が相手のときにはある程度難易度を調整したプレイが実現されたが、上級者が相手のときには殆ど勝てなかった。プレイヤーが取れる行動は非常にたくさんあり、AI が次の行動を決めるためには、アクションの組み合わせを大量に調べる必要がある。しかし、処理速度の問題上、すべてのアクションの組み合わせに対して MCTS を用いてリアルタイムで (約 16ms 以内に) 探索を行うこと難しい。これが十分な強さの AI を実現できない理由であると考えられる。

2.2 遺伝的アルゴリズムを用いた計算効率の向上

Kim らは、遺伝的アルゴリズムを用いることで、MCTS の計算効率を向上する手法を提案している[4]。

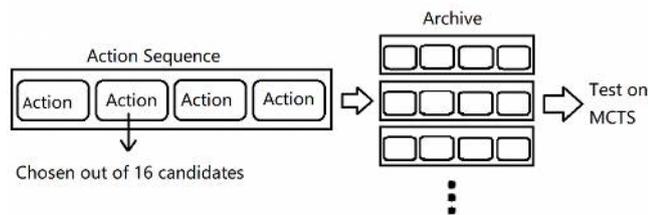


図 2 アクションシーケンスの構成プロセス

Figure 2 Construction Processes of Action Sequence.

図 2 は行動の組み合わせであるアクションシーケンス (AS) を構成するプロセスである。FightingICE では、合計 56 個の可能なアクションがある。そのうち、16 個の地上アクションが有望な候補アクションであることが知られている。遺伝的アルゴリズムでは、構造内の各遺伝子は、この 16 個の候補遺伝子からランダムに選択されたアクションであり、その遺伝子が 4 つ連なって、アクションシーケンス染色体 (AS 染色体) を構成する。AS 染色体は、選択、交叉、突然変異などの遺伝子オペレーターを反復で処理し、遺伝子アーカイブに保存される。その中で、正の報酬を伴う AS 染色体が選択されると、良い AS であるアクション

コンボを得ることとなる。これを繰り返して、有望な AS を蓄えたアーカイブを用いることで、MCTS において有望な AS を優先的に探索することができるようになることで、効率の良い探索を行うことが可能となる。この手法を利用する AI と利用しない MCTS の AI を対戦させたところ、利用した AI の方の勝率が 99% となり、この手法の有効性が示されている。

3. プロトタイプシステム

本研究では、Ishihara らの手法では十分な強さが得られなかった動的難易度調整の手法に、Kim らの手法を組み合わせることで、十分な強さを有し、しかも動的な難易度調整を行うシステムの作成を試みる。

AS 染色体の良さを示す適応関数は、HP の差で定義される。先行研究に倣って、以下のようなパラメータ設定 (集団サイズ: 48, 遺伝子の長さ: 4, 適応関数: 対戦相手との HP 差, 交叉率: 三点交叉, 突然変異率: 0.01) の下で、遺伝的操作を行う。そのような操作を 50 世代行い、その結果得られた正の報酬のある 4 つの特定の染色体を選択する。そして、この結果を MCTS に適用する。

MCTS において UCT で用いるノードの評価値 $eval_j$ 値は式(1)で与えられる。

$$eval_j = (1 - \alpha) B_j + \alpha E_j \quad (1)$$

ここで、 B_j は AI の攻撃性を表しており、式(2)で示される。

E_j は式(3)を使用して定義された難易度調整のための評価値である。係数 α はゲーム状況に基づいて式(4)によって動的に決定される。

$$B_j = \tanh \frac{beforeHP_j^{opp} - afterHP_j^{opp}}{Scale} \quad (2)$$

$$E_j = 1 - \tanh \frac{|afterHP_j^{my} - afterHP_j^{opp}|}{Scale} \quad (3)$$

$$\alpha = \frac{\tanh \left(\frac{beforeHP_j^{my} - beforeHP_j^{opp}}{Scale} \right) + 1}{2} \quad (4)$$

$beforeHP_j^{my}$ と $beforeHP_j^{opp}$ は j 番目のシミュレーション前の AI と対戦相手の HP を表す。また、 $afterHP_j^{my}$ と $afterHP_j^{opp}$ は j 番目のシミュレーション後の AI と対戦相手の HP を表し、 $Scale$ は定数である。

4. 評価実験

4.1 目的

3 章の手法で実現したプロトタイプシステム (TestAI) を構築し、従来の Ishihara らの AI よりも十分に強くなっているのかを確かめる。また、実際に動的な難易度の調整が出来ているのかについても調べる。

4.2 方法

比較対象として、Ishihara らの提案手法を実装した AI システム MCTS_DDA を構築した。

TestAI と MCTS_DDA に対して、それぞれ 2019 年に IEEE-COG で開催された AI の強さを競う大会に参加した 9 つの AI と 100 試合ずつ対戦させて、その強さを比較した。実験に用いた対戦条件は、IEEE-COG で用いられたルールに則って、双方の初期 HP を 400 とし、先に HP が 0 になった方を負けとした。また、制限時間を 60s とし、それが切れた時点で HP が少ない方も負けとした。

それぞれの対戦の勝率と終了時の HP の差の平均を計算し、どの程度、動的に調整が出来ているのかを比較した。

4.3 結果

MCTS_DDA と TestAI に対する 2019 年の IEEE-COG における大会の各 AI との対戦結果を、それぞれ表 1 と表 2 に示す。表の左の欄は、大会における成績順であり [5]、上から順に強いプログラムを表している。

表 1 MCTS_DDA と 2019 年大会出場 AI との対戦結果

Table 1 Game results for MCTS_DDA against the 2019 competition AIs.

	勝率	平均HP差
ReiwaThunder	0.00%	320
RHEA_PI	0.00%	
LGIST_Bot	0.00%	
FalzAI	0.00%	
Toothless	0.00%	199
HaibuAI	0.00%	
DiceAI	20.00%	97
MuryFajarAI	0.00%	
TOVOR	47.00%	

表 2 TestAI と 2019 年大会出場 AI との対戦結果

Table 2 Game results for TestAI against the 2019 competition AIs.

	勝率	平均HP差
ReiwaThunder	0.00%	178
RHEA_PI	0.00%	
LGIST_Bot	2.00%	
FalzAI	6.06%	91
Toothless	10.10%	
HaibuAI	19.00%	78
DiceAI	60.20%	
MuryFajarAI	96.96%	
TOVOR	74.00%	

表 1 を見ると、TOVOR に対しては 50%弱程度の強さに調整されているものの、他の AI に対しては殆どが 0% の勝率となっているのに対して、表 2 を見ると、DiceAI に対してまで勝ち越している。このことから、遺伝的アルゴリ

ズムを組み合わせる手法で、明らかに強くなっていることが示された。表 2 の中で TestAI と勝率 60%程度であった DiceAI は人間の上級者と対戦させたところ、上級者レベル以上の強さの AI であることがわかった。このことから、TestAI は、上級者に対しても十分に強さの調整が可能なレベルに強くなっていることが確認された。

また、終了時の HP の差を見ると、MCTS_DDA よりも TestAI の方が HP の差が小さくなっており、最後まで良い勝負を演じている可能性が示唆された。しかし、勝率を見ると、例えば MuryFajarAI に対して、96%以上の勝率を挙げるなど、勝ちすぎる傾向が見られた。

5. 改良 AI の提案

5.1 改良の方針

適度な強さの AI を実現するというためには、対戦の勝率が 50%に近いことが期待される。表 2 の結果のように、TestAI は、それより弱い AI に対して勝率が高すぎる傾向がある。これでは、良い対戦相手とは言えないと考える。TestAI において、強さの調整で用いていた指標としては、HP の差に着目していた。勝率を調整するためには、勝敗に直結する値に着目する必要がある。

人間と AI のプレイを見ると、攻撃が外れたときに、相手の攻撃が当たることで大きなダメージを受ける傾向が見られた。そこで、新たな指標として、攻撃の命中率に着目して、自分の攻撃の命中率と相手の攻撃の命中率の差を用いることにする。

5.2 α の改良

上述の方針を実現するために、3 章における式(4)を改良し、命中率の差も考慮した新しい式(5)を提案する。

$$\alpha = \frac{\tanh\left(\frac{before_{hi}^{my} - before_{hi}^{opp}}{Scale1}\right) + \tanh\left(\frac{hi_{opp} - hi_{my}}{Scale2}\right) + 1}{3} \quad (5)$$

式(5)の $Scale1$ と $Scale2$ は、定数を表しており、調整することで、HP の差を少なくし、勝率を 5 割に近づけていく。

6. 今後の方針

今後は、5 章の改良方針に沿って改良 AI を作成する。改良 AI により、どの程度勝率が 5 割に近づいたかを表 2 の結果と比較する。また、この改良によって、対戦する人間が対戦相手として「好敵手と感じる AI になっているのか」「対戦して楽しく感じるのか」について、主観的評価を行い、この改良の有効性を示していきたい。

謝辞 本研究は、JSPS 科研費 18H03347 の助成を受けたものです。

参考文献

- [1] Csikszentmihalyi, M.: "Flow : the Psychology of Optimal Experience" Harper Perennial, (1990).
- [2] Chen, J.: "Flow in games" (and everything else), Comm. ACM, Vol.50, No.4, pp.31–34 (2007).
- [3] M. Ishihara, S. Ito, R. Ishii, T. Harada and R. Thawonmas, "Monte-Carlo Tree Search for Implementation of Dynamic Difficulty Adjustment Fighting Game AIs Having Believable Behaviors," 2018 IEEE Conference on Computational Intelligence and Games (CIG), pp. 1-8 (2018),
- [4] Man-Je Kim, Jun Suk Kim, Donghyeon Lee, Sungjin James Kim, Min-Jung Kim, Chang Wook Ahn, "Integrating Agent Actions with Genetic Action Sequence Method", Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp.59-60 July 13–17, (2019).
- [5] 2019 Fighting Game AI Competition in IEEE-COG: <http://www.ice.ci.ritsumei.ac.jp/~ftgaic/index-R19.html> (参照 2020 -10-12)