

プレイの質を反映する動的難易度調整機能を備えた 対戦テトリス AI

Tetris AI with Dynamic Difficulty Adjustment Reflecting Play Quality

菅野 健太

Kenta Sugano

法政大学情報科学部デジタルメディア学科

E-mail: kenta.sugano.7y@stu.hosei.ac.jp

Abstract

In competitive games, dynamic difficulty adjustment—which adjusts the challenge level according to the player’s skill—is widely adopted as a key technique for keeping players immersed. Traditional research on dynamic difficulty adjustment primarily focused on keeping players’ flow state. It was achieved through approaches that decreased difficulty, such as reducing it after a failure. This extends the player’s survival time, but it also hides the player’s own mistakes and inefficient play, leading to lost opportunities for skill improvement. This paper proposes an AI that reverses the traditional concept of dynamic difficulty adjustment and provides instructional feedback to improve player skill. Specifically, using Tetris as the target, it measures in real time the gap between the “optimal move” calculated by a rule-based AI and the player’s actual moves. This paper also proposes a negative feedback mechanism that immediately increases difficulty when a low-quality move occurs. In addition, two types of visual feedback mechanisms are implemented to enhance player skill improvement more effectively. The first is real-time evaluation using pop-up displays during matches, and the second is cumulative evaluation that can be reviewed in graph format after matches. This paper investigates how the combination of these dynamic difficulty adjustments and visual feedback affects players’ learning efficiency.

1. はじめに

一般的な対戦ゲームでは、プレイヤーの没入感を維持するための重要な技術として、プレイヤーのスキルレベルに応じて難易度を調整する動的難易度調整が広く研究されている。従来の動的難易度調整に関する研究の多くは、主に失敗した際に難易度を下げるなどの手加減を行うアプローチを採用し、プレイヤーのフロー状態の維持を主目的としていた。これによってプレイヤーの生存時間は延長されるが、プレイヤー自身のミスや非効率なプレイがシステムによって隠蔽され、スキル向上の機会が失われるという課題がある。

本研究では、従来の動的難易度調整の概念を逆転させ、プレイヤーのスキル向上を目的として指導的な介入を行うシステムを提案する。具体的には、対戦テトリスを題材とし、ルールベース AI によって算出された最適手とプレイヤーの実プレイとの乖離をリアルタイムで計測し、その乖離が大きく質の低い手を打った場合に即座に難易度を上昇させる、罰則的なフィードバック機構を提案する。また、プレイヤーのスキル向上をより強力に促進するため、2 種類の視覚的フィードバック機構を実装する。1 つ目は対戦中に「Good」や「Bad」のようなポップアップを表示することによるリアルタイム評価であり、2 つ目は対戦後にリプレイを通じた各場面における最適手の提示とグラフ形式で確認できる累積評価である。本論文は、これらの動的難易度調整と視覚的フィードバックの組み合わせが、プレイヤーの学習効率にどのような影響を与えるかを調査する。

2. 関連研究

Fan らは AI が弱い対戦相手に合わせて手加減する際に、不自然で不可解な悪手を打つことを避け、あくまで「ありそうな手」の中から次善手を選択する多面的なフレームワークを提案した [1]。これにより、AlphaGo Zero 以降の現代的な囲碁 AI の Leela Zero や KataGo といったプログラムが価値ネットワークを用いて局面を直接評価していることにより AI の強さを調整したり、特定の戦略を誘導したりする従来手法が適用できなかった問題を解決し、プレイヤーの体験を損なうことなく接戦を演出することを可能にした。

Hsueh らは候補となる手を生成しその良し悪しを評価する超人的プログラムと、手の自然さを評価するプログラムを組み合わせることで、弱いプレイヤーに適したプログラムを作成することを提案した [2]。これにより、指し手の良さが有利になり、悪い手を指せば不利になるという分かりやすい因果関係がゲームの結果に反映されて学習効果が高まるとともに、ゲームの最後まで集中力とモチベーションを保ちやすくすることを可能にした。

3. 提案手法

3.1. 概要

本研究では、プレイヤーのスキル向上支援を目的とし、自身のプレイの質に応じて AI の目標とする形勢が動的に

変化するテトリス対戦システムを提案する。システムの基盤となるゲームエンジンおよびAIアルゴリズムには、高速かつ強力なテトリスAIとして知られるオープンソースの Cold Clear¹を採用した。本システムでは、この Cold Clear を提案手法の動的難易度調整に対応できるように再実装し、プレイヤーの操作を監視して最適手との乖離度を測定する機能と、プレイヤーの盤面状況と最適手との乖離度の累積値からAIの目標とする形勢を調整する機能を追加する。また、学習効果を高めるための視覚的フィードバックとして、対戦中および対戦後の二段階の可視化を行う。図1は本システムにおける対戦中の画面である。

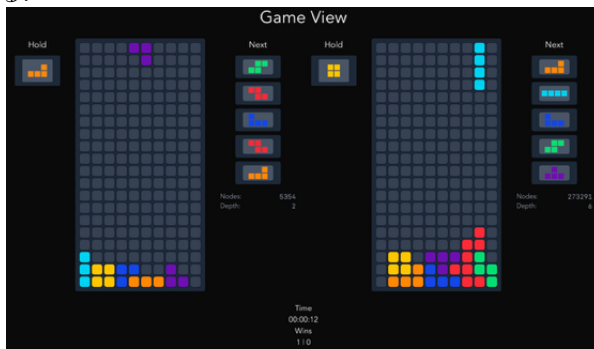


図1 ゲーム画面

3.2. プレイの質に基づく難易度調整

本システムの中核となるのは、最適手との乖離度を用いた動的難易度調整である。AIはプレイヤーの現在の盤面におけるアルゴリズム上の最適手を常に計算しており、プレイヤーが実際に行った手の場合の評価値と比較する。プレイヤーの手がAIの予測した最適手の評価値にやや劣る、あるいは同等以上であれば質の高いプレイ、下回る場合は質の低いプレイと判定する。その際、単発の評価値ではなく、減衰を伴う累積値を用いて目標形勢を決定する。その目的は、一度のミスが敗北へ直結することを防ぐ点にある。これにより、質の低いプレイが続けば累積ペナルティによって状況が加速的に悪化し、逆に質の高いプレイを継続すれば累積値が減衰して状況が好転するという、連続的なプレイの質が勝敗の決定要因となるゲームバランスを実現する。

具体的には、システムは形勢を動的に制御するために、以下の計算モデルに判定結果を適用し、AIが目標とする形勢値 $V_{\text{target}}(t)$ を決定する。まず、予測された最適手の評価値 $V_{\text{predicted}}(t)$ と実際に配置された手の評価値 $V_{\text{placed}}(t)$ の差に定数のオフセット値 c を加えて、時刻 t における最適手との乖離度 $d(t)$ を計算する。

$$d(t) = V_{\text{placed}}(t) - V_{\text{predicted}}(t) + c$$

そして、前回の累積値 $D_{\text{acc}}(t-1)$ に減衰率 γ を乗じ、今回の乖離度 $d(t)$ を加算することで、時刻 t における最適手との乖離度累積値 $D_{\text{acc}}(t)$ を算出する。

$$D_{\text{acc}}(t) = \gamma \times D_{\text{acc}}(t-1) + d(t)$$

最後に、AIが目標とする形勢値 $V_{\text{target}}(t)$ は、プレイヤーの現在の盤面評価値 $V_{\text{player}}(t)$ から、累積乖離度 $D_{\text{acc}}(t)$ に反映率 α を乗じた値を減算することで求められる。

$$V_{\text{target}}(t) = V_{\text{player}}(t) - \alpha \times D_{\text{acc}}(t)$$

3.3. 段階的に適応される速度調整

難易度調整で操作速度がプレイヤーと大きく乖離していると、プレイの質による動的難易度調整を無視して対戦の結果が決まってしまう可能性がある。そこで、プレイヤーの入力速度に動的に適応する速度調整機能を実装する。具体的には、プレイヤーの各操作(移動、回転、ハードドロップ等)に要したフレーム数を計測し、最新の20回分を保持して移動平均を算出する。この値をAIの各操作における遅延フレーム数としてリアルタイムに適用することで、常にプレイヤーの現在の操作速度と拮抗する、あるいはわずかに上回るようなテンポでの対戦を実現し、操作速度が対戦の結果に影響を与えすぎないように維持する。

3.4. 対戦中のプレイの質の可視化

プレイヤーの学習効果を高めるためには、自身の行動に対する即時的な評価が必要である。本システムでは対戦中におけるリアルタイムの視覚的フィードバックとして、図2のように、プレイヤーの一手ごとに行っている質の評価を可視化する機能を実装する。判定基準は3.2節で述べた評価手法に基づき、プレイヤーの実際の手とAIの予測した最適手の評価値の乖離度によって決定される。評価値が基準を満たした(最適手にやや劣る、あるいは同等以上の)場合は「Good」、下回った場合には「Bad」というポップアップを配置直後に画面の中央に短時間オーバーレイ表示する。これにより、プレイヤーは複雑な数値情報を読み解くことなく、自身の直近の判断が適切であったかどうかを直感的に認識し、プレイ中の行動修正に繋げることが可能になる。

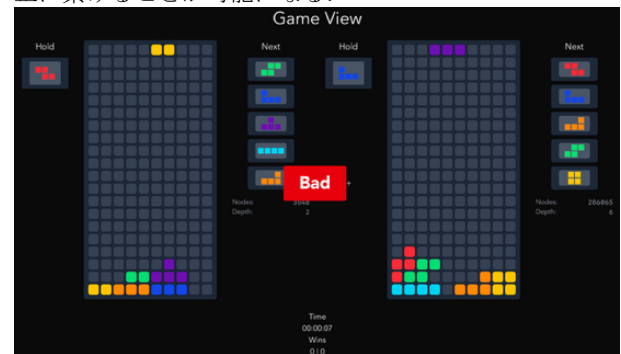


図2 対戦中のフィードバック

3.5. 対戦後のプレイの質のグラフ化

対戦後には、学習内容を定着させるための振り返り支援として、図3のようにリプレイと同時にプレイの質の推移をグラフ化して提示する。具体的には、対戦開始から終了までの全ての手について、プレイヤーの配置した手の評価値と、その場面におけるAIの最適手の評価値の

¹ <https://github.com/MinusKelvin/cold-clear>

推移, これらの差を累積した値の推移を時系列グラフとして描画する. このグラフにより, プレイヤーは単なる勝敗の結果だけでなく, ゲーム全体の流れの中で「どの場面で判断ミスが多発したか」や「安定して質の高い手を打っていた区間」を客観的に把握・分析することを可能にする. この事後的なフィードバックは, プレイヤー自身の癖や弱点の発見を促し, 次回のプレイに向けた具体的な改善点の発見に役立つと期待される.

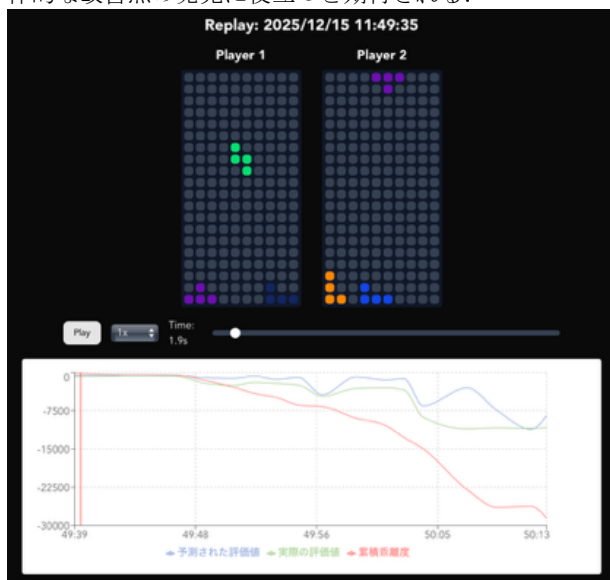


図3 対戦後のフィードバック

4. 実装

4.1. 対戦テトリス

本システムの構築にあたり, AI の計算性能とモダンなユーザーインターフェース(UI)の両立を図るため, バックエンドに Rust, フロントエンドに Web 技術を採用するアプリケーションフレームワーク Tauri を用いた. テトリスのゲームロジックおよび AI の探索アルゴリズムは, メモリ安全性と実行速度に優れた Rust 言語で実装した. 特に, 探索空間の広い AI の演算処理がゲームのフレームレートや UI の応答性に悪影響を及ぼさないよう, Rust の並列処理機能を活用したマルチスレッドアーキテクチャを採用している. これにより, ゲーム進行, AI の最適手計算, およびフロントエンドとの通信をそれぞれ独立したスレッドで非同期に実行し, 高負荷時でも遅延のないリアルタイム処理を実現した.

本実装には, 後述する検証実験を効率的かつ再現性をもって行うためのシミュレーション機能を搭載している. これは, 人間のプレイヤーの代わりに, 評価関数の重みを調整してパフォーマンスを低下させた「弱くした AI」を動作させる機能と, 探索木の深さを調整してより先読みを可能にした「強くした AI」である. この機能により, 被験者を集めることなく, 特定のスキルレベル(例: 初心者のように頻繁に悪手を打つ状態)を定常的に再現し, 提案手法の挙動検証を行うことを可能にしている.

これらの実験条件を柔軟に制御するため, GUI ベースの詳細な設定画面を実装した. 設定画面では, プレイヤーおよび対戦相手のエージェント種別(人間/通常 AI/弱体化 AI)の切り替えや, 動作速度調整・動的難易度調整機能の有効/無効を個別に設定できる. 加えて, 3 章で定義した提案手法のハイパーパラメータである減衰率や反映率についても, 数値を直接入力して即座にシステムへ反映させることが可能であり, パラメータ感度の調整や比較実験を容易に行える設計としている.

4.2. 評価関数

AI の盤面評価関数は, Cold Clear に実装されている評価器(Evaluator)を移植し, 再実装した. Cold Clear は現代のテトリス AI で最高峰の性能を持つが, 本研究の目的は一般的なテトリスの基礎スキル向上であるため, 超人的な特定の動きに特化した一部の機能は意図的に除外している. 具体的には, 特定の開幕定石を機械的に実行する Opening Book, およびパーフェクトクリアをループ的に狙い続けるための探索アルゴリズム PC-Loop は無効化した. 本実装で採用した評価関数の主な重みとその設定値を表 1 に示す. これらの重み付けにより, AI は効率的なライン消去と地形の平坦化を優先する標準的な強豪プレイヤーとして振る舞う.

表 1 評価関数の重みの一部抜粋

重み	評価値
back to back	52
top quarter	-511
cavity cells	-173
overhang cells	-34
tslot	8,148,192,407
well depth	57
well column	20,23,20,50,59,21,59,10,-10,24
wasted t	-152
b2b clear	104
clear1	-143
clear4	390
tspin3	602
mini tspin1	-158
perfect clear	999
combo garbage	150

4.3. 難易度調整

3 章で提案した難易度調整をシステム上で動作させるため, バックエンドの Rust プロセス内で, プレイヤーの現在の盤面評価値 V_{player} と累積乖離度 D_{acc} を永続的なステート(状態)として保持する設計とした. プレイヤーがミノを設置するたびに, AI スレッドは非同期で評価計算を行い, これらのステート変数を更新する. 算出された目標形勢値 V_{target} は即座に AI に渡され, この値を満たす手候補を選択するように探索木を枝狩り, あるいは評価バイアスを適用して挙動を決定する. これにより, プレイヤーの直近のパフォーマンスに応じたリアルタイムな挙動の変化を実現している.

4.4. 速度調整

プレイヤーの操作速度への段階的な適応については、操作イベントごとに現在蓄積されているフレームカウンタの値を Rust の FIFO に適した VecDeque という型の変数に後ろから追加していき、フレームカウンタを 0 にリセットする方法で実装した。変数の長さが 20 を超えた場合に先頭を取り除くことで、常に保持されている値が 20 個になるようにしている。

4.5. 対戦中のプレイの質の可視化

対戦中の視覚的フィードバックについては、Rust 側で判定された Good/Bad のシグナルをフロントエンドへ Tauri の機能を用いてイベント発火することで実装した。React 側はこのイベントを受け取り、盤面レイヤーの上にポップアップとして 0.5 秒間オーバーレイ表示を行っている。

4.6. 対戦後のプレイの質のグラフ化

対戦後の視覚的フィードバックについては、対戦中にミノが配置された時間のタイムスタンプ、その時点で予測されていた最適手の評価値、実際に配置された手の評価値、これらの差分(乖離度)の情報を CSV 形式で全て記録し、後からそのファイルを読み込んでグラフ化できるように実装した。また、リプレイ機能を実現するための盤面状態や落下中のミノの状態などを保存する際に、予測されていた最適手の配置を保存しておき、リプレイの盤面内にゴーストとして最適手を半透明に表示することで、直感的に最適手を把握できるようにした。

5. 実験

5.1. AI との対戦による動的難易度調整の評価

意図的に弱くした AI と強くした AI との対戦により、動的難易度調整が意図した通りに動作していることを確認する実験を実施した。本実験で使用した弱くした AI は、盤面の凹凸や高さを気にせず、相手に多くの火力を送ることができるテトリスや T-Spin のような強い攻撃をひたすら避けて殆ど火力を送れない 1, 2 列程度でこまめに消すように調整している。また、最適手の探索を深さ 1 で行っているのに対して、強くした AI は深さ 3 で探索を行うように調整している。

実験では、意図的に弱くした AI と強くした AI を、通常の AI と動的難易度調整を用いずに同じ速度で対戦させた。動的難易度調整の反映率と減衰率はどちらも 0.9 として、動的難易度調整を用いた場合と用いなかった場合でそれぞれ 100 回ずつ対戦させた結果とそれぞれの対戦時間を表 2、表 3 に示す。

表 2 AI との対戦結果

	動的難易度調整	強さを変更した AI (回)	本研究で構築した AI (回)
対戦相手を弱くした場合	あり	31	69
	なし	0	100
対戦相手を強くした場合	あり	100	0
	なし	45	55

表 3 AI との対戦時間

	動的難易度調整	平均時間(秒)	最短時間(秒)	最長時間(秒)
対戦相手を弱くした場合	あり	65.568	14.149	878.195
	なし	20.190	9.732	40.598
対戦相手を強くした場合	あり	41.407	9.099	114.956
	なし	63.418	14.346	286.623

5.2. 被験者との対戦による動的難易度調整と可視化手法の評価

14 人の被験者に実際に本システムを使ってもらい、アンケートに回答してもらった。被験者のテトリスのプレイ歴は、ほとんど/まったくやっただことがないが 5 人、多少遊んだことがあるが 6 人、ある程度遊んだことがある、そこそこ遊んだことがある、かなり遊んだことがあるがそれぞれ 1 人である。キーボードでのテトリスのプレイ方法がわからない被験者には、事前に多少操作に慣れてもらうため、動的難易度調整および動的難易度調整が無効な単純な AI と 1, 2 回対戦してもらった。その後、動的難易度調整が常に有効な状態で実験を行った。

5.3. 手順とアンケート結果

具体的な手順は以下の通りである。

- ① 動的難易度調整と可視化手法が無効な状態で AI と 3 回対戦する。
- ② 動的難易度調整と可視化手法が有効な状態で AI と対戦し、対戦後にリプレイとグラフを確認することを 3 回繰り返す。
- ③ アンケートに回答する。

ただし、対戦中の熟練度の変化による影響をなるべく打ち消すために、半分の被験者には手順①と②を逆にして行った。

アンケートの質問は、以下の 5 段階評価の 3 問と自由記述のコメントである。

- Q1. 動的調整(速度や AI の強さ)が自身のスキルに対して適切に負荷を感じたか
- Q2. Good/Bad 表示が自身のミスを意識するきっかけになったか
- Q3. リプレイでの最適手表示やグラフが具体的な改善策を理解するのに役立ったか

5.4. 結果

アンケートでは Q3 についてはほぼ全員から 4 の評価を得たのに対し、Q1, 2 については評価が分かれた。自由記述では、対戦中の可視化については「Good」「Bad」の表示を対戦中に見ている余裕がないか、見ることで集中が削がれるという意見や、今の手の評価をより具体化してほしいという意見があった。対戦後の可視化については、数手先まで見るとどのような手を目指しているのかがわかりやすいという意見や、対戦中にもこれを表示するモードがあればより学習しやすいという意見などがあった。また、各操作の頻度などをもとにプレイヤーの傾向を分析する機能があると良いといった意見もあった。

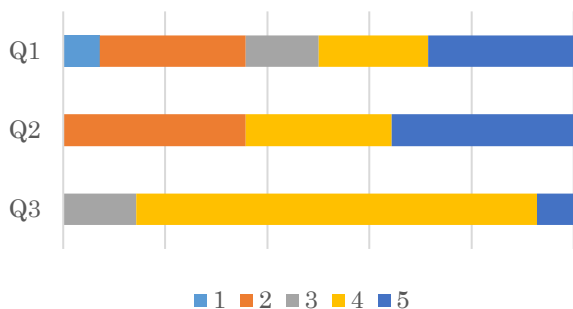


図4 アンケート結果

6. 議論

意図的に弱くした AI との対戦では、動的難易度調整が無効な状態では意図した通り通常の AI が圧勝する結果となっているのに対して、有効な状態では弱い AI 側が 31% の割合で勝利してしまっている。これは、動的難易度調整を行う際にプレイヤー側の盤面評価値に足し引きする形で目標となる形勢を算出しているため、プレイヤーに引き摺られる形で AI 側も盤面状況が悪くなった際に、小さい火力やミノの順番などで詰んでしまうのが原因であると考えられる。この傾向は被験者との実験でも確認された。また、操作速度が一定以上遅い被験者に対して、AI 側の操作にかかるフレーム数が、ミノが固定されるまでのフレーム数を上回ることによって、意図した通りに AI がミノを動かさず、ミスが多発してしまう現象が発生した。ただし、強くした AI との対戦では、動的難易度調整が有効な状態で意図した通り強い AI が圧勝しており、対戦時間も短くなる結果となった。これは、基準とする盤面評価値を AI 側の盤面評価値にするか、プレイヤーの盤面評価値と平均をとったものにするかで、AI 側の盤面が著しく悪化するのを防げば対策できると考えられる。また、AI 側の操作速度に下限値を設けるか操作種別ごとに動的な速度調整を変えることで AI 側が意図した通りに動けない問題も解消すると考えられる。

可視化手法については概ね高評価で、改善可能な点も多く見つかった。対戦中の可視化については、音楽ゲームのように実際に配置されたミノの近くにスコア形式で最適手との乖離度を表示することで、視点移動コストを最小化すると共にスコアの増減という形で手の向上を促せると考える。対戦後の可視化については、AI が予測した数手先の最適手の表示と、その結果発生する T-Spin やテトリスのような技を提示することで、AI の予測の意図をわかりやすくできると考える。また、プレイヤーの各操作や技の頻度を記録・分析したデータを表示し、プレイヤーがどのような手を取りやすい傾向にあるかを認識できるようにすることで、苦手とする技などの把握をしやすくできると考える。

7. おわりに

本論文では、プレイヤーのスキル向上を目的とし、プレイの質に応じて AI の目標とする形勢が動的に変化する

テトリス対戦システムを提案した。動的難易度調整では、改善すべき点が多く見つかったものの、プレイヤーの手の良し悪しに対して罰則的なフィードバックを与える手法の対戦テトリスの学習支援システムにおける有効性を検証できた。可視化手法では、対戦中の可視化手法は認知負荷で改善すべき点が見つかったものの、対戦後の可視化手法が特に学習支援効果が高いことを確認でき、さらなる改善案も得られた。

今後の課題は、評価関数のパラメータ自体を変更することによって、プレイヤーに難易度調整を悟られにくくする自然な動的難易度調整の実現である。これにより、プレイヤーに手加減されているという不快感を与えにくくなり、モチベーションの維持に役立つと考える。

文 献

- [1] T. Fan, Y. Shi, W. Li and K. Ikeda, "Position Control and Production of Various Strategies for Deep Learning Go Programs," *Proc. IEEE TAAI*, 2019.
- [2] C.-H. Hsueh and K. Ikeda, "Playing Good-Quality Games with Weak Players by Combining Programs with Different Roles," *Proc. IEEE CIG*, pp. 612-615, 2022.