

# 効果的に T-spin を行えるテトリスエージェントの検討

植村 侑生<sup>1,a)</sup> 横山 大作<sup>1,b)</sup>

**概要:** 近年のテトリスにおいて必須テクニックとされている T-spin は、そのセットアップの複雑さから従来手法の評価関数では不利な行動と判断され構築が困難であった。そこで本研究では、穴が形成される列や地形のパターンマッチ等 31 種の特徴量を導入した評価関数を、遺伝的アルゴリズムを土台とする手法を用いて最適化し、複数の探索アルゴリズムを用いた盤面探索を行うことによって行動選択を行うコンピュータプレイヤーの構築を試みる。探索に使用するアルゴリズムとして、ビームサーチと確率的最良選択を用いた探索のそれぞれについてアルゴリズム間の性能を比較検討した。構築されたプレイヤーは比較対象の上級者プレイヤーを攻撃量と T-spin 作成効率の両面で上回り、提案手法の有効性が示された。

## Building Controllers for Efficient T-spin Execution in Tetris

YUKI UEMURA<sup>1,a)</sup> DAISAKU YOKOYAMA<sup>1,b)</sup>

**Abstract:** In modern competitive Tetris, a technique known as "T-spin" is considered to be essential. However, its complex setups are often judged to be disadvantageous by conventional evaluation functions, making its construction difficult for computer controllers. In this research, we propose an evaluation function that incorporates 31 categories of features, including terrain pattern matching and hole positioning, specifically designed to favor T-spin setups. The weights for these features are optimized using a method based off of Genetic Algorithm (GA). We then constructed a controller using this function and compared the performance of two search algorithms: Beam Search and a Stochastic best-first search algorithm. The results demonstrate that the constructed player surpassed advanced human players in both attack power and T-spin efficiency, proving the effectiveness of our proposed method.

### 1. はじめに

『テトリス』は、1984 年にソビエト連邦の科学者アレクセイ・パジトノフによって開発された、世界で最も著名なパズルゲームの一つである。その分かりやすいゲーム性に加え、運の要素が少なくプレイヤーの実力が勝敗に大きく影響することから、近年では e スポーツの競技種目としても注目を集めている。

近年リリースされているテトリス作品の多くでは、ルールセットにいくつかの共通仕様が採用される傾向にある。その中でも特に重要なのが、T 字型のテトリミノ（以下、T ミノ）を特定の地形で回転させてラインを消すことで通常

よりも高い攻撃力を得られる「T-spin」と呼ばれるテクニックの導入である。T-spin は主に対戦型のテトリスにおいて活用されることが多く、2 ライン消去（T-spin Double、以下 TSD）で 4 ライン消去と、3 ライン消去（T-spin Triple、以下 TST）で 4 ライン消去の 1.5 倍と同等の攻撃力を発生させるなど、少ない手数で高い攻撃力を発生させることが可能である。そのため、現代の対戦テトリス環境において T-spin は勝敗を分ける必須戦術として位置づけられている。

しかしながら、T-spin を戦術として組み込むことは従来手法を用いたエージェントでは困難であった。T-spin を実行するためには、図 1 に示すように意図的にブロックの下に空洞を作ったり（「屋根」と呼ばれる地形）、盤面を凹ませたりする必要がある。これは Thierry ら [1] や宮崎ら [2] によって提案されている「盤面上の穴の数」や「地形の凹

<sup>1</sup> 明治大学大学院 理工学研究科  
Graduate School of Science and Technology, Meiji University  
<sup>a)</sup> ce235009@meiji.ac.jp  
<sup>b)</sup> dyokoyama@meiji.ac.jp

凸」をペナルティとして盤面を評価する従来のエージェント研究では「悪い状態」と判断される状態である。そのため T-spin の準備段階として要求される盤面は「悪い状態」と評価されてしまい、エージェントが T-spin の構築手順を発見することは困難であった。

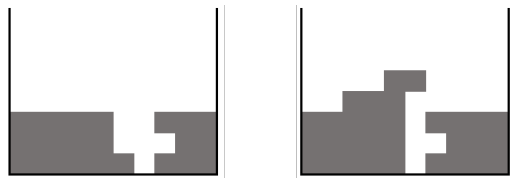


図 1: T-spin に要求される地形の例

加えて、近年のテトリス作品の多くにおいてプレイヤーは次に操作可能となるブロック (NEXT) を 5 手先まで見ることが可能である。トッププレイヤーに要求される一手約 0.5 秒という判断速度に対し、愚直に全探索を行うと 5 手先までの探索空間が約  $1.18 \times 10^{11}$  通りに達するという計算量的困難性も存在する。本研究ではこれらの課題を解決し、T-spin を効率的に行うことのできるエージェントの構築を目指す。そのために本論文では T-spin を構築する際に頻出する地形を積極的に評価する特徴量を導入した評価関数を設計し、それを用いて効率的に最善手を探す探索部を実装したエージェントを提案する。

本稿の構成は以下の通りである。まず 2 章にて近年のテトリス作品における詳細なルールと従来手法の問題点を述べる。次に 3 章にて提案手法を述べ、4 章にて提案手法に関する実験と結果の考察を述べる。最後に 5 章にてまとめと今後の展望について述べる。

## 2. テトリスにおける基本戦術と関連研究

この章では近年のテトリス作品におけるルールを述べた後、従来研究によって提案されている手法について説明し、T-spin を効率的に構築するうえで従来手法にどのような問題点があるのかについて議論する。

### 2.1 テトリスのルール

プレイヤーはテトリミノと呼ばれるブロックを操作し、フィールドの横一列をブロックで埋めることでラインを消去する。対戦型のテトリスではこのライン消去が相手プレイヤーへの「攻撃」となり、攻撃を受けたプレイヤーはフィールドの下から「おじゃまブロック」と呼ばれるブロックがせり上がるペナルティを受ける。おじゃまブロックはプレイヤー自身が設置したブロック同様にライン消去を行うことでしか消すことができず、これによってフィールドが圧迫され新たなテトリミノを置くスペースがなくなると敗北となる。したがって、対戦では相手よりも効率的に多くの攻撃を送り続けることが勝利の鍵となる。

この攻撃効率を劇的に高める手段が「T-spin」である。T-spin とは、T ミノを図 2 に示すような特定の地形で回転させながら設置することでラインを消去する特殊なテクニックを指す。T-spin によるライン消去は、同じライン数を通常の方法で消去した場合と比較して多くの攻撃量を発生させるという特徴を持つ (表 1)。例として 2 ラインを消去する TSD は通常の 4 ライン消去とほぼ同等の攻撃力を持ち、攻撃手段として極めて効率的である。

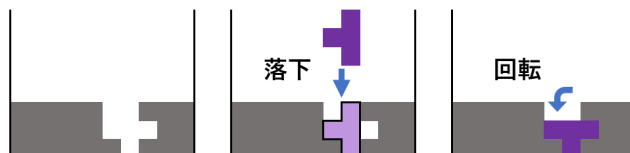


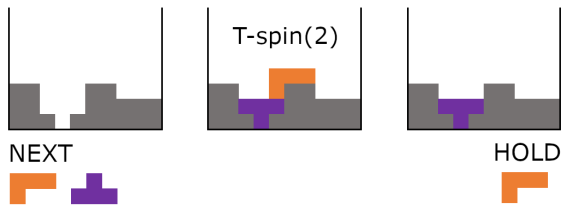
図 2: 典型的な T-spin の例

表 1: ライン消去の際発生する攻撃量

消去ライン数	T-spin	攻撃量
1	✓	2
2	✓	4
3	✓	6
1		0
2		1
3		2
4		4

こうした戦術的な判断を支えるシステムとして、「NEXT」と「HOLD」が存在する。プレイヤーは現在操作中のテトリミノに加え、次に出現するテトリミノの情報が 5 手先まで「NEXT」として与えられる。これにより、プレイヤーは数手先の状況を予測した計画的な地形構築が可能となる。これに加え、プレイヤーは操作中のテトリミノを 1 つだけ保留し、後から好きなタイミングで NEXT の先頭と入れ替えて使用する「HOLD」を行うことができる。これにより、特定の地形で必要となるテトリミノを温存するなど、戦術の柔軟性が大幅に向上する。

一方で、これらの機能はエージェントが考慮すべき探索空間を大きく増加させる。一つのテトリミノにおける設置可能な合法手は平均的におよそ 30~40 通りほど存在し、HOLD 機能によって実質的に 2 種類のテトリミノから選択できることから 1 手あたり約 70 通りの選択肢が生まれると仮定できる。現在操作中のテトリミノに加え NEXT で見える合計 6 手先までを単純に全探索しようとする、その探索空間は約  $70^6 \approx 1.18 \times 10^{11}$  通りにも達し、現実的な時間内での完全な探索は不可能となる。



(a) 初期状態 (b) 想定解 (c) 従来手法の回答

図 3: 従来手法では発見できなかった手

## 2.2 従来研究

テトリスエージェントに関する従来研究の多くは、通常の 4 ライン消去を主な目的として設計されてきた。Thiery ら [1] や宮崎ら [2] によって提案された手法はその代表例であり、彼らの用いた評価関数は「盤面の高さ」や「穴の数」、「表面の凹凸」といった特徴量をペナルティとして計算するものであった。これによりエージェントは可能な限り盤面を平坦に保ち、盤面上に空洞を作らないような手を選択するようになった。

しかし、この評価方針は T-spin を戦術として組み込む上で問題点を抱えている。前述の通り T-spin を実行するためには事前に「屋根」となる地形を意図的に構築する必要がある、これは従来手法においてはペナルティとして避けられてきた状態にあたる。そのためビームサーチのような探索アルゴリズムを用いる場合、これらの手は探索の早い段階で評価の低い候補として枝刈りされてしまい、有望な手が探索から除外されてしまうといった問題がある。

例として図 3 に示すような盤面に対して Thiery ら [1] によって提案された手法を適用した結果を示す。この盤面では初期状態から L 字型のテトリミノを図のように設置することで TSD の地形を構築し、2 手以内に TSD を行うことが可能である。しかしこの手を選択することで一時的に盤面に大きな凹凸と屋根が生じてしまうことから、実際のエージェントは目先のペナルティが少ない T ミノを直接地形にはめ込み通常の 2 ライン消去を行う手を選択した。このように従来手法では TSD の前段階となる地形が十分に評価されていないことから、長期的にリターンのある手に対しても悪手と判断してしまっているといえる。

## 2.3 関連研究

本研究における課題は、リアルタイム性や非有限性、長期的な目標達成のため目先の報酬を犠牲にする非貪欲な戦略が求められる点において、他のゲームを対象とした研究と共通点を持つ。

例えば吉田ら [3] は、格闘ゲームプラットフォーム「FightingICE」においてモンテカルロ木探索（以降、MCTS）を適用したエージェントが、従来のルールベースエージェントを上回る性能を示したことを報告している。この研究に

おける課題としては 1 フレーム（約 16.67ms）以内に行動を決定しなければならないリアルタイム制約下において、単純なルールベースエージェントでは対応が難しい大局的な戦略判断が求められる点が挙げられ、本研究の課題と多くの共通点を持つ。

また Samothrakis ら [4] は、アーケードゲーム「Ms.Pac-Man」において MCTS を適用したエージェントが高い成果を上げたことを報告している。このゲームでは一手あたりに許容される時間幅が限定されるリアルタイム性、明確な終わりがなくスコアを最大化し続ける非有限性や、目先の報酬だけでなく長期的な生存やスコアの最大化を優先する非貪欲な戦略が求められる点において、本研究における課題と多くの共通点を持つ。

## 3. 提案手法

本研究で提案するエージェントは盤面の状態を評価する「評価関数」と、それを用いて最善手を探す「探索部」の要素から構成される。評価関数には T-spin を構成する際に頻出する地形を評価する特徴量を導入し、その構築を促す。探索部ではこの評価関数を用いて、計算コストを抑えつつ複数手先までを効率的に探索する方法を検討する。

### 3.1 評価関数の全体設計

本エージェントは複数手先までの手の組み合わせを木として展開し、最終的に最も評価値の高くなる一手を選択することで行動を決定する。選択する手の評価は、その過程となる「行動評価」の和と最終的に到達した盤面の「地形評価」の 2 つから算出される。

ここで各行動評価  $r$  はライン消去や T-spin の実行など、各手数でプレイヤーが行った行動そのものに対して与えられる報酬を表す。例として、TSD を実行した手には高い正の報酬が、1 列の通常ライン消去を実行した手には負の報酬が与えられる。

各地形評価  $e$  は盤面そのものの状態に対して与えられる評価値である。これは盤面の高さや穴の数といった、その盤面が持つ静的な特徴から産出されるものである。

最終的なノードの評価値  $E$  は探索開始ノードから最深部のノードに至るまでの一連の行動評価の総和  $R$  と、最深部ノードにおける地形評価  $e$  の和、すなわち  $E = R + e$  によって算出される。プレイヤーは  $E$  が最大となるような一連の手順を発見し、その手順の最初の一手を選択することで行動を決定する。

本エージェントによって行われる行動評価の簡略化された一例を図 4 に示す。なお、この図では各層において評価値  $E$  が高いノードほど左に配置されているものとする。与えられた状態において、エージェントは赤枠で示されたノード（以下、目的ノード）に到達することを目標とする。したがって、エージェントは目的ノードに至る手筋（図中

の太線)の初手にあたる行動評価  $r_{0,2}$  で表される手を選択する。このとき目的ノードの評価値  $E_{3,0}$  は以下の式によって求められる。

$$E_{3,0} = r_{0,2} + r_{1,1} + r_{2,0} + e_{3,0}$$

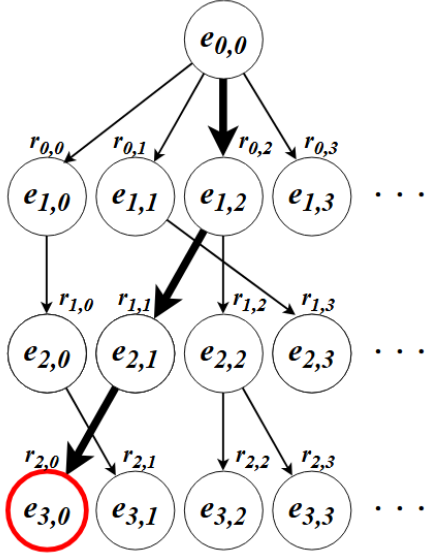


図 4: 行動評価ロジックの概略図

本研究において行動評価および地形評価に用いる評価関数は表 2 および表 3、表 4 に示す計 31 種の特徴量と、それぞれに設定された重みの総和によって定義される。ここで、本研究で採用した特徴量は盤面の安定性を維持するための基本的な特徴量と、T-spin 構築を積極的に誘導するため独自で選出した特徴量に大別される。以降の節では、これらの特徴量について詳細を述べる。

### 3.2 基本的な盤面評価

T-spin の構築のみを評価の対象とした場合、エージェントは T-spin の形を作ることのみに固執しそれ以外の状況で盤面を著しく悪化させてしまう可能性がある。これを防ぎエージェントの基本的な性能を担保するため、Thiery ら [1] および宮崎ら [2] の用いた特徴量を参考に「盤面の高さ」や「表面の凹凸」などの表 2 に示す計 9 種の特徴量を採用した。これらは主に盤面を不必要に高くしたり、解消が困難な屋根を作るなどといった無意味な悪形の選択を防ぐためのペナルティとして機能し、エージェントの基本的な動作を安定させる目的を持つ。

### 3.3 井戸の位置

本研究では T-spin のセットアップを積極的に誘導するため、盤面における最も低い列（以下、井戸）の位置を評価する特徴量 *wellDistance* を独自に導入した。

T-spin の中でも特に使用頻度が高い TSD はその地形的

表 2: 基本的な地形評価に用いる特徴量

名称	特徴量の説明	種類
height	盤における最高地点の高さ	1
heightLevel	height が段階的に用意された閾値を超えているか	3
ceiling	各列の最上段より下に存在する空マス (穴) の合計数	1
ceilDepth	各列における穴の深さの合計値	1
steepness	隣列との高さの差をとった合計	1
sSteepness	隣列と高さの差をとり 2 乗した合計	1
flatness	隣列と高さが変わらなかった列数	1

表 3: T-spin 誘導のため用いられる地形評価の特徴量

名称	特徴量の説明	種類
wellDistance	盤の両端から井戸までの最小列数を表す One-hot ベクトル	5
pattern	事前に定義されたそれぞれのパターンの発見数	9

表 4: 行動評価に用いる特徴量

名称	特徴量の説明	種類
lineClear	通常ライン消去の列数を表す One-hot ベクトル	4
tSpin	T-spin によるライン消去の列数を表す One-hot ベクトル	3
tWasted	T-spin が発生しない T ミノの設置を行ったかを表す boolean	1

な特徴から、井戸が盤面の左右両端の列ではなく端から 3~4 列ほど中央に寄った位置にある場合に構築しやすいことが一般的に知られている。これは TSD を構成する地形が盤面の表面に現れている際、その盤面で最も低い列は必然的に両端の列とはなり得ないためである (図 2)。そのため井戸が盤面中央寄りの特定の列に存在する場合に正の評価を与えることを期待し、井戸の位置を示す特徴量を One-hot ベクトルの形式で定義した。これにより、エージェントは自然と T-spin を狙いやすい盤面構造を維持することが期待される。

### 3.4 パターンマッチング

T-spin の構築をより直接的に誘導するため、盤面に特定の地形パターンが現れた際に報酬を与える特徴量 *pattern* を導入した。これはエージェントが T-spin に至るまでの一連の手順を発見する上でのヒントとして機能するものであり、その完成度に応じて「完成パターン」と「直前パターン」の 2 種類に分類される。使用した完成パターンを図 5 に、直前パターンを図 6 に示す。ここで 1 はマスが埋まっていること、0 はマスが空であること、- は don't care を意味し、これらのパターンを左右反転した形についても検知を行った。



1	0	0
0	0	0
1	0	1

(a) TSD パターン

1	0	0
0	0	0
0	1	-
0	0	-
0	-	-

(b) TST パターン

図 5: 使用した完成パターン

0	0	0
1	0	1

(a) TSD パターン

0	0	0
0	0	0
0	1	-
0	0	-
0	-	-

(b) TST パターン 1

1	0	0
0	0	0
0	0	-
0	0	-
0	-	-

(c) TST パターン 2

図 6: 使用した直前パターン

完成パターンは、T ミノを設置すれば即座に T-spin が実行可能である地形パターンを指す。図 5 に示すような地形パターンが盤面に発見された場合、盤面評価に報酬が与えられる。なお形が完成していてもラインが埋まっておらず、T ミノを設置してもライン消去が発生しない、あるいは規定ライン数よりも少ないライン消去が発生する場合も考慮し、本パターンによって与えられる報酬は「想定される T-spin を実行した際、実際に消去されるライン数」を重みに掛けることで算出するものとした。

一方直前パターンは、完成パターンが構築される 1~2 手前の段階で頻出する地形を指す。TST をはじめとする一部の T-spin のセットアップは複数手に渡る手順を必要とするため、完成形のみを評価対象とした場合その途中の手順が悪手と判断され枝刈りされてしまう可能性がある。そのため図 6 に示すようなセットアップ途中の地形にも報酬を与えることで、エージェントを T-spin の完成へと段階的に誘導するものとした。

### 3.5 評価関数の最適化

本研究で設計した評価関数は、それぞれの特徴量に対する重みの組み合わせによってその性能が大きく左右される。これらの重みを人手でゼロから調整することは非常に困難であるため、本研究ではその最適化を遺伝的アルゴリズム（以下、GA）と手動での調整の 2 段階で行った。

まず、宮崎らの研究 [2] においてテトリスエージェントの評価関数を最適化する上で有効性が示されている GA を用いることでベースラインとなる重みを決定した。GA における各個体は 31 種の特徴量に対する重みにそれぞれ対応し、その適応度は実際にエージェントを動作させることで評価される。具体的には一つの個体に対して 5000 入力ステップのゲームを 100 回試行し、その間の平均攻撃量を

適応度とした。

次に GA によって得られた重みを基準値として、各特徴量の重みを個別に変動させながら性能を計測することで手動での調整を行った。例として、TSD の直前パターンに対する重みを調整した際のエージェントの平均攻撃量、および設置された T ミノのうち T-spin に活用された割合（以下、T-spin 効率）の変化を図 7 に示す。ここで青で示される攻撃量に着目すると、重みが 0 から 150 付近の範囲では性能がほぼ最高値で安定していることが分かる。一方赤で示される T-spin 効率の推移に注目すると、重みが 100 から 250 程度の範囲で T-spin 効率が最大化されていることが分かる。これらの結果から判断し、本特徴量の重みとしては 120 付近が最適であると決定した。他の特徴量についても同様の手法で調整を重ね、最終的な重みセットを完成させた。

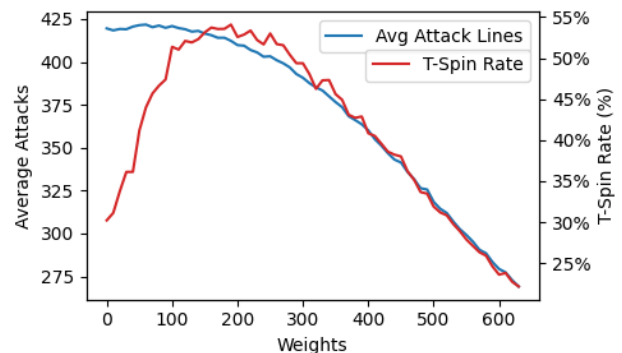


図 7: TSD 直前パターンの重みとプレイヤ評価の変化

### 3.6 探索方法

エージェントは前節までで定義した評価関数および与えられる NEXT 情報を用いて複数手先の盤面までを探索し、最終的に最も評価の高くなる一手を探索する。本研究では探索アルゴリズムとして、テトリスのゲーム性質との相性が良く高速であるビームサーチを採用した。加えて、ビームサーチが持つ局所最適解の問題を解決する可能性を持つ確率的最良選択アルゴリズムを比較対象として実装し、両者の性能について比較検討を行った。<sup>\*1</sup>

ビームサーチは各探索深度で有望な候補を一定数に絞り込むことにより高速な探索を実現する。少ない計算量で有意な結果を得ることのできる点に加え、探索深度を問わず常に最適な手を選択し続けることで一定の成果をあげられる [1] というゲームの性質上、ビームサーチはテトリスと相性の良い探索法であると考えられる。一方で短期的に評価が下がる有効な手筋については探索の早い段階で枝刈り

<sup>\*1</sup> 予稿では探索アルゴリズムの比較対象として MCTS を実装するとしていたが、実装に誤りがあったため、本稿ではここで紹介する確率的最良選択アルゴリズムを比較対象として採用した。

してしまい、局所最適解に陥るという課題が挙げられる。

本研究で採用する確率的最良選択アルゴリズムは、探索に確率的な選択を取り入れることで、より広域の探索を試みる手法である。このアルゴリズムでは、1手の決定にあたり末端ノードの発見と展開を規定回数だけ実行する。各末端ノードの発見・展開では、ルートノードから評価値に基づく重み付き乱択を用いて子ノードを選択していく。未展開のノードに到達した時点でそのノードを展開し、評価を上位ノードに反映させる。評価値が高いノードほど深く探索が行われる確率が高くなるものの、確率を用いることで評価値が低い手筋についても探索される可能性がある。そのため、ビームサーチでは枝刈りされてしまうような、長期的なリターンを持つ手筋を発見できることが期待される。しかし、その探索的な性質から計算コストが高くなるという明確な課題が挙げられる。

## 4. 評価実験

### 4.1 評価方法

性能評価の対象として3.3節の評価関数とビームサーチ(幅12, 深さ5)を組み合わせたエージェント(以下, ビームサーチプレイヤー)および同評価関数と確率的最良選択アルゴリズムを組み合わせたエージェント(以下, 確率的最良選択プレイヤー)を作成し測定を行った。なお, 確率的最良選択プレイヤーは1手の決定にあたり1000回の探索を実行し, 各ノードにおける選択を行う際には評価値の5乗を重みとした確率的選択を用いるものとした。比較対象としては日常的にテトリスをプレイする上級者1人, 月に数回程度プレイしている中級者2人を被験者としたリプレイデータを用いた。

またエージェントの性能を客観的に評価するため, 本研究ではテトリミノに対する左右移動や回転, HOLDといった操作を1入力ステップとし, 一定入力ステップのゲームを複数回行った際の平均攻撃量および, 設置した全TミノのうちT-spinに使用された割合を示す「T-spin 効率」の2つを評価指標として採用した。

実験は, 近年のテトリス作品で採用されているルールに準拠して独自に開発したクライアント上で行った。なおプレイヤー間の性能比較において, 本研究で作成したエージェントに対しては5000入力ステップのゲームを100回試行した際の平均値を採用したが, 人間プレイヤーは長時間のプレイに伴う集中力の低下を考慮し, 5000入力ステップのゲーム(約10~15分のプレイに相当)を5回試行した際の平均値をそれぞれ評価値として採用した。

### 4.2 提案手法の有効性の検証

ビームサーチプレイヤー, 確率的最良選択プレイヤーおよび人間プレイヤーに対し実験を行い得られた評価値を表5に示す。これにより, ビームサーチプレイヤーは攻撃量および

T-spin 効率の両面において人間プレイヤーを大きく上回ったことが確認できる。特に T-spin 効率に注目すると, 上級者1が28.8%であったのに対しビームサーチプレイヤーは57.1%と約2倍の数値を記録した。この結果は本研究において提案する評価関数が T-spin の構築に対して有効に機能していることを示しており, また数手先に得られる報酬を探索の浅い段階の評価値に反映させることで, 有望な手順を枝刈りから保護できていることを示している。

表 5: 構築したプレイヤーと人間プレイヤーの性能比較

プレイヤー名	平均攻撃量	T-spin 効率 (%)
ビームサーチプレイヤー	<b>426.4</b>	<b>57.1</b>
確率的最良選択プレイヤー	284.0	18.4
上級者 1	330.4	28.8
中級者 1	251.2	23.8
中級者 2	175.0	29.8

一方確率的最良選択プレイヤーについて着目すると, 攻撃量については上級者プレイヤーと中級者プレイヤーの中間程度, T-spin 効率については人間の中級者プレイヤーを下回る結果となった。この原因は確率的最良選択がその特性である大局的な探索を優先するあまり, 有望な手筋を深く掘り下げる前に展開の少ない手の検証に多くの計算リソースを費やしてしまい, 結果として探索効率が大幅に低下してしまったことによるものと考えられる。

この問題はシミュレーション回数を大幅に増やすことによって解決することが期待されるが, 4.4章でも論じるように確率的最良選択は既に多くの計算コストを要しており, 1手あたりの思考時間に制約がある中でシミュレーション回数を単純に増やすことは困難である。

### 4.3 各特徴量の有効性の検証

本エージェントにおいて提案した特徴量が T-spin 効率の高さに寄与していたことを証明するため, 特定の特徴量を無効化した際の性能変化を比較する Ablation test を行った。本実験では4.2章の性能比較実験において最も高い性能を示したビームサーチプレイヤーを基準となる Base プレイヤーとし, 図5および図6に示した各種 T-spin の完成パターンと直前パターンの両方を無効化したプレイヤー (NoTSD, NoTST, NoPatterns), 直前パターンのみを無効化したプレイヤー (NoPreTSD, NoPreTST), および3.3章にて言及した井戸の位置に関する特徴量 *wellDistance* を無効化したプレイヤー (NoWellDistance) からなる計6種のプレイヤーを作成し, 性能を比較した。(表6)

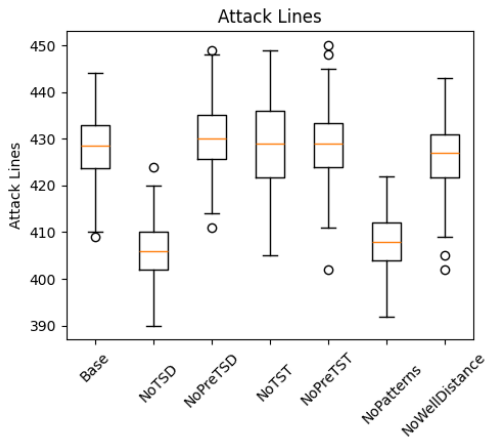
実験結果を表7および図8に示す。結果より, 本研究において提案した特徴量の中でも特に TSD のパターン, および井戸の位置に関する特徴量が大きく T-spin の実行に寄与していることが明らかになった。

表 6: Ablation test を行ったプレイヤーと有効化された特徴量

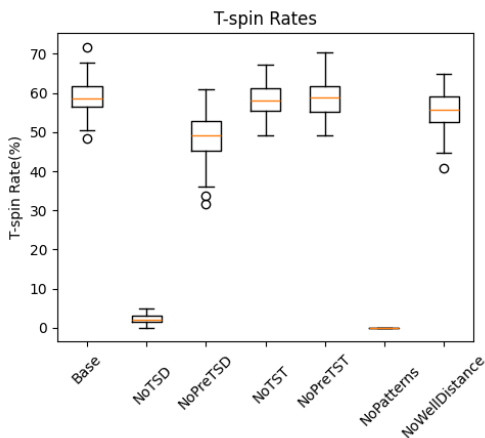
プレイヤー名	TSD 完成 (図 5a)	TST 完成 (図 5b)	TSD 直前 (図 6a)	TST 直前 (図 6b, 6c)	wellDistance
Base	✓	✓	✓	✓	✓
NoTSD		✓		✓	✓
NoPreTSD	✓	✓		✓	✓
NoTST	✓		✓		✓
NoPreTST	✓	✓	✓		✓
NoPatterns					✓
NoWellDistance	✓	✓	✓	✓	

表 7: Ablation test の結果

プレイヤー名	平均攻撃量	変化率 (%)	T-spin 効率 (%)	変化率 (%)
Base	426.40	-	57.10	-
NoTSD	406.06	-5.06	2.30	-96.10
NoPreTSD	429.81	0.49	48.70	-17.49
NoTST	428.65	0.22	58.27	-1.27
NoPreTST	428.16	0.10	58.60	-0.71
NoPatterns	407.34	-4.76	0.00	-100.00
NoWellDistance	426.30	-0.33	55.57	-5.85



(a) 攻撃量



(b) T-spin 効率

図 8: Ablation test の結果

TSD の完成パターンおよび直前パターンを無効化した NoTSD プレイヤは T-spin 効率が-96.10%と著しく低下しており、Base プレイヤにおいて実行されている T-spin のほとんどがこの TSD パターン検知によるものであることが分かる。また、直前パターンのみを無効化した NoPreTSD プレイヤにおいても T-spin 効率が-17.49%と顕著な低下が確認され、セットアップを段階的に誘導するヒントとして直前パターンが有効に機能していたことが示された。

一方、TST に関連するパターンを無効化した NoTST および NoPreTST プレイヤにおいては顕著な性能変化が見られなかった。TST は TSD に比べ地形構築に要する手順が多く、他の特徴量によって悪手と判断される点も TSD と比較すると多い。そのため探索深度内に構築が完了しないと判断される場面が多く、結果的に TST パターンの評価が最終的な行動選択に影響を与える場面が限定的であったことによるものと考えられる。

また *wellDistance* を無効化した NoWellDistance プレイヤでは累計攻撃量に大きな変化は確認されなかったものの、T-spin 効率に関して-5.85%と一定の低下が確認された。これは 3.3 章における仮説の通り、井戸の位置を中央に誘導することが T-spin の発見に対する間接的なヒントとして有効に作用していたことを示している。

#### 4.4 探索手法の比較と考察

ビームサーチプレイヤーおよび確率的最良選択プレイヤーについて、計算コストの観点から 1 手あたりの平均探索時間および平均探索ノード数を比較した結果を表 8 に示す。

表 5 および表 8 の結果より、平均攻撃量や T-spin 効率

表 8: ビームサーチプレイヤーと確率的最良選択プレイヤーの比較

プレイヤー名	平均探索時間 (ms)	平均ノード数
ビームサーチプレイヤー	41	2721
確率的最良選択プレイヤー	977	44167

といった主要な性能指標においてビームサーチは大きな優位性を示し、計算コストについても大幅に抑えられていることから、すべての指標においてビームサーチプレイヤーが確率的最良選択プレイヤーを上回ったことが確認できる。特に計算コストの観点において、確率的最良選択プレイヤーは1手あたり約977msの時間を要しており、これはビームサーチプレイヤーの約20倍に相当する。探索ノード数についても確率的最良選択の方が多いにも関わらず性能が及ばなかったことから、本研究の課題設定においては確率的最良選択が効率的に有望な手を探索できていないものと考えられる。

一方ビームサーチに認められる課題点として、図9aおよび図10aに示すような一部の局面においては最適解を選択できていないという点が挙げられる。この局面において想定される最適解は図9bおよび図10bに示す手筋であり、これらの手筋では提示盤面からTSDを一回行うことが可能である。しかし実際にビームサーチプレイヤーが選択したのは、T-spinを行わずにTミノを設置し攻撃を発生させない図9cおよび図10cに示す手筋であった。

この選択は提案手法の評価関数により該当局面においてTSDの構築に必要な手筋が悪手と判断されてしまい、ビームサーチによる枝刈りが早期に行われてしまったことによるものと考えられる。今後の課題としては既存の良い選択を行う能力を維持しつつ、こうした特定の局面においても最適解を発見できるよう評価関数の重み調整および新たな地形パターンの模索を行うことが挙げられる。

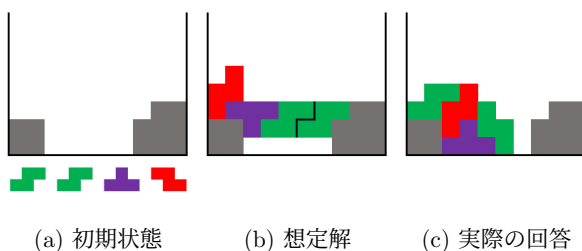


図 9: T-spin の発見に失敗した例 (1)

## 5. おわりに

本研究では、テトリスにおいてT-spinを効率的に行うことのできるエージェントの構築に取り組んだ。T-spinのセットアップで頻出する地形パターンなどを特徴量として組み込んだ評価関数を設計し、ビームサーチと組み合わせ

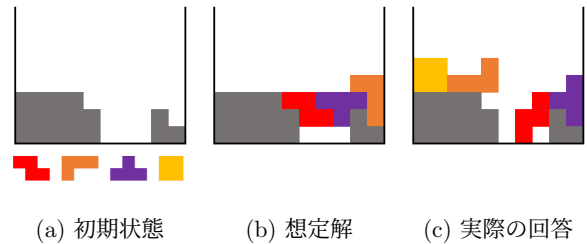


図 10: T-spin の発見に失敗した例 (2)

ることで、標準的な人間プレイヤーと比較して高い攻撃量とT-spin効率を達成するエージェントを構築できることを確認した。また確率的最良選択との比較実験を通して、本研究の課題設定における探索アルゴリズムとしてビームサーチの優位性を確認した。

本研究における今後の課題点の一つとしては、ビームサーチの性質上解決が困難な局面が複数存在するという点が挙げられる。複数手先にまたがる複雑なセットアップについても正確な評価を行い最適解へと誘導できるよう、新たな地形パターンの考案や特徴量の改良を行う等、さらなる評価関数の改善が求められる。

また本研究にて提案したエージェントはT-spinを効率的に行うことにのみ着目しており、近代的な対戦型テトリスにおいて同じく重要な戦術とされる「REN」や「パーフェクトクリア」といった戦術には対応していない。今後はこれらを組み込むことで、より総合的な判断能力を持つエージェントへの拡張を行いたい。

最後に、構築したエージェントが提示する最適手をプレイヤーの効果的な学習につなげる具体的な手法を確立することも今後の重要な研究課題である。今後はこれらの課題を解決することで、テトリスにおけるプレイヤーの技能向上に貢献できる学習ツールを実現できると期待される。

## 参考文献

- [1] Building Controllers for Tetris, Christophe Thiery and Bruno Scherrer, ICGA Journal, vol. 32, pp.3-11, 2009
- [2] ニューラルネットワークと遺伝的アルゴリズムを用いたテトリスコントローラの開発, 宮崎 真奈実, 荒川 正幹, 情報処理学会全国大会 2012 (1), pp.539-540, 2012
- [3] Application of Monte-Carlo tree search in a fighting game AI, S. Yoshida, M. Ishihara, T. Miyazaki, Y. Nakagawa, T. Harada and R. Thawonmas, 2016 IEEE 5th Global Conference on Consumer Electronics, pp.1-2, 2016
- [4] Fast Approximate Max-n Monte Carlo Tree Search for Ms Pac-Man, S. Samothrakis, D. Robles and S. Lucas, IEEE Transactions on Computational Intelligence and AI in Games, vol.3, no.2, pp.142-154, 2011