

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[‡]
宇部工業高等専門学校

1. はじめに

テトリスは 1985 年に開発されたコンピュータゲームである。人工知能分野の研究対象ともなっており、テトリスを自動でプレイするプログラム(テトリスコントローラ)の開発が行われている。最適化問題として定式化することが可能であるため、これまでに様々な最適化手法が適用された。その中で最も良い結果を報告しているのが Thiery ら[1]および Boumaza[2]である。Thiery らは、表 1 に示す 8 個の特徴量を用いた線型の評価関数を cross-entropy method を用いて最適化することで、平均 3,500 万ラインの性能を示すテトリスコントローラの開発に成功した[1]。また Boumaza は、covariance matrix adaptation evolution strategy を適用することで、平均 3,630 万ラインを達成した[2]。

本研究の目的は、従来の記録を上回る性能を示すテトリスコントローラを開発することである。評価関数として非線型のニューラルネットワーク(NN)を利用し、遺伝的アルゴリズム(GA)を用いて最適化を行った。

表 1 テトリスコントローラで用いられる特徴量

名称	説明
LandingHeight	直前においたピースの高さ
ErodedPieceCells	消えたラインの数× ピースの中で消えたブロックの数
RowTransitions	横方向にスキャンしたとき、 セルの内容が変わる回数
ColTransitions	縦方向にスキャンしたとき、 セルの内容が変わる回数
NumHoles	穴の数
CumulativeWells	井戸の高さの階和の和
HoleDepth	穴の上のブロック数
RowsWithHoles	穴のある行

2. 手法

2.1. 評価関数の最適化

今回構築したテトリスコントローラの概要を図 1 に示す。現在のピースの回転と移動を考慮し、次の盤面として考えられる全ての状態を生成する。そして、各盤面について評価関数を用いて評価を行い、最も高い評価値を持つ盤面を選択する。

Designing an Tetris Controller Using a Neural Network and Genetic Algorithm

[†] Manami Miyazaki, Ube National College of Technology

[‡] Masamoto Arakawa, Ube National College of Technology

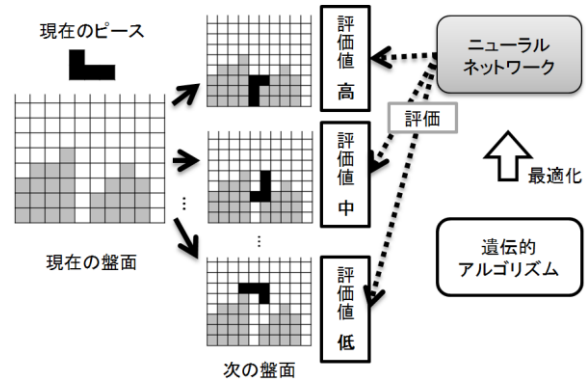


図 1 テトリスコントローラの概要

評価関数として、Thiery らの用いた 8 個の特徴量を入力とする NN を採用し、GA によって重みの最適化を行った。GA における適応度は、個体によって表現される評価関数を用いて 100 回のテトリスゲームを行ったときの平均スコアとした。テトリスのスコアの分布は幾何分布にほぼ従うため[1]、信頼性の高い値を得るためには多くの試行が必要である。したがって、現実的な時間で最適化を行うためには、計算時間の短縮が欠かせない。

本研究では、S ピースと Z ピースの出現頻度を高くすることでこれを実現した。テトリスにおいて、S、Z ピースは隙間なく積むことが難しく、これらの出現が多い場合にはスコアが低くなることが知られている。また予備実験により、S、Z ピースの出現頻度が高い場合のスコアと、通常ゲームのスコアとの間には、強い正の相関があることが確認されている。したがって、この方法を用いることにより、最適化の質をほぼ低下させずに計算時間を短縮することが可能である。

2.2. Pseudo-two-level search

通常のテトリスゲームにおいては、現在のピースに加え、次に出現するピースの情報が利用可能であるが、本研究および既往の研究[1, 2]においては、この情報を利用しないルールで性能の比較を行なっている。もし次のピースが既知であれば、図 1 に示される各候補盤面について、もう 1 手先の盤面候補を得ることが可能となるため、スコアの劇的な向上が期待される。

本研究では、次のピースの情報がない場合においても、擬似的に 2 段階の探索を行う手法として、

pseudo-two-level search を提案する。1 段階目の各盤面について、7 種のピースのそれぞれを擬似的に次のピースとして扱い、最適な置き方を決定する。そして、それらの盤面の評価値を平均することで、1 段階目の各盤面の評価値とする。

3. 結果

3.1. コントローラの最適化

GA を用いて評価関数の最適化を行った。GA のパラメータは、世代数 500、個体数 100、交叉数 5、複製数 5、突然変異率 1%とした。NN の中間層に含まれるニューロンの数は、予備実験の結果に基づき 5 個とした。また、適応度の計算においては、S ピースおよび Z ピースの出現頻度を他のピースの 3 倍に設定した。

GA による最適化を 15 回行い、各試行において最も高い適応度を示した評価関数を得た。そして、各評価関数を用いて通常のテトリスゲームを 100 回行ったときの平均スコアを指標とし、最適な評価関数を選択した。得られた評価関数の性能を既存のテトリスコントローラと比較するため、各 1,000 回のシミュレーションを行った結果を表 2 に示す。

既存のコントローラの性能が、約 3,200 万ラインおよび約 3,000 万ラインであったのに対し、GA によって最適化したコントローラは、約 6,000 万ラインの性能を示した。標準偏差および中央値についても同様に高い値を示している。試行回数が 1,000 回の場合、平均値の 95%信頼区間は約±6.3%である。したがって、既存のコントローラと比較し、有意に性能の高いコントローラを構築することに成功したと結論する。

表 2 テトリスコントローラの性能比較

	平均値	標準偏差	中央値
提案手法	60,228,109	56,841,860	39,091,684
Boumaza	32,115,826	34,240,371	20,934,030
Thiery	29,782,036	27,620,546	20,780,267

3.2. 人間のプレイヤーとの比較

今回得られたコントローラと、人間のプレイヤーとの比較を行った。手動でテトリスをプレイできるプログラムを作成し、3 人の被験者について 100 ゲームの平均スコアを算出した。ただし時間短縮のため、S ピースおよび Z ピースの出現頻度は 5 倍、盤面の高さは 10 とした。結果、人間のプレイヤーはそれぞれ 60 ライン、24 ライン、16 ラインであったのに対し、GA によって得られたコントローラは 80 ラインの性能を示した。このことから、今回得られたコントローラは、標準的な人間のプレイヤーと比較し、高い性能を持っていることが確認された。

図 2 に、コントローラが選択したピースの置き方の例を示す。灰色はすでに盤面に存在しているブロックであり、黒色はコントローラが選択したピースの置き方を示す。これらの置き方は被験者の意見と異なるものであったが、コントローラの方が平均スコアが高いことから、人間のプレイヤーがこれらの置き方を参考にすることでスコアが上昇すると考えられる。

一方で図 3 に示すように、コントローラの判断が適切でないと考えられる例も見られた。これらの盤面においても適切な判断が行えるようにコントローラを調整することで、さらに高いスコアが得られるものと期待される。

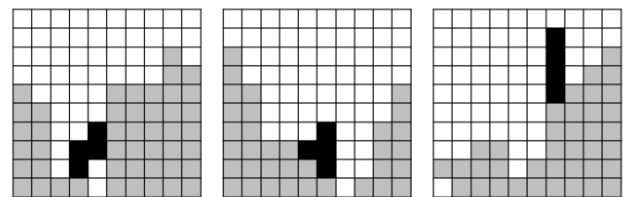


図 2 適切な判断を行った例

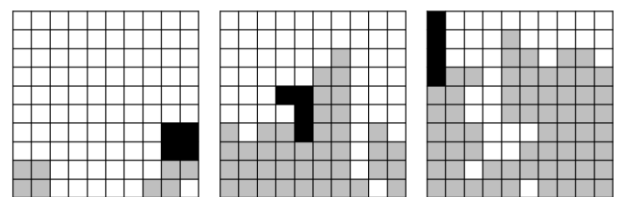


図 3 適切でない判断を行った例

4. おわりに

本研究では、GA を用いて NN の重みを最適化することで、高い性能を示す評価関数の開発を行った。そして pseudo-two-level search を用いることで、既存のコントローラよりも高いスコアを示すテトリスコントローラを構築することに成功した。主な要因は、NN を用いることで、従来の線形式では表すことが出来なかった複雑な関係を表現出来たこと、pseudo-two-level search によって、次のピースを考慮に入れた判断が可能になったこと、の二つである。

今後は、新たな特徴量の開発や、より効率的な最適化手法の適用によって、さらに高い性能を持つテトリスコントローラの開発を目指す予定である。

参考文献

- [1] C. Thiery, B. Scherrer, Building controllers for Tetris, ICGA J. **32** (2010) 3-11.
- [2] A. Boumaza, On the evolution of artificial Tetris players, in: Proceedings The IEEE Symposium on Computational Intelligence and Games, Milan, Italy, 2009, pp. 387-393.