JAIST Repository
https://dspace.jaist.ac.jp/

Title	テトリスにおけるT-spin構成力向上のための問題生成
Author(s)	及川，大志
Citation	
Issue Date	2020-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/16383
Rights	
Description	Supervisor: 池田　心，先端科学技術研究科，修士（情報科学）

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修士論文

テトリスにおける T-spin 構成力向上のための問題生成

1810027　及川 大志

主指導教員　池田 心
審査委員主査　池田 心
審査委員　飯田 弘之
上原 隆平
長谷川 忍

北陸先端科学技術大学院大学
先端科学技術研究科
（情報科学）

令和 2 年 2 月

概要

ゲームは人類の大事な文化であり，また情報技術の良いテストベッドでもある．深層学習などの発達により多くのゲームで人間よりも強いコンピュータプレイヤが作成できるようになってきている一方で，最近ではゲームそのものを面白くするために，人間プレイヤの満足度向上を目的とした人工知能の研究も注目されてきている．そのためにマップやストーリーの自動生成，チームメイト，あるいはプレイヤの上達を支援する教育者としての役割など，対戦相手としてだけでなく様々な場面で人工知能が活用されつつある．

「テトリス」は落下型パズルゲームの始祖ともいえる人気ゲームである．テトリスは 1 人用ゲームとしてのルールが有名だが，近年では 2 人以上で遊ぶ「対戦テトリス」にも注目が集まってきている．対戦テトリスでは同時に複数行を削除することによって対戦相手に攻撃を送ることができるが，通常の削除よりも大きな攻撃力を得られるいくつかのテクニックが存在する．特に「T-spin」と呼ばれる特殊な削除方法による攻撃は強力で，これを上手く構築できるか否かがプレイヤの強さを左右する．その一方で，T-spin が発生する条件は初心者にとって難解であることから “最初の壁” とされており，多くのプレイヤは上達を目指す過程でこの技術の習得に難儀する．

T-spin を構築するためには，ランダムな順序で与えられる着手ブロックを巧みに組み合わせて特定の形を構築する必要があるが，これは容易ではない．さらに，ブロックを配置するまでの時間制限や相手プレイヤからの攻撃によるプレッシャーのために，実戦で T-spin 構築の練習をすることは難しい．そのため,「実戦的な」「ただし実戦のようなプレッシャーのない」環境での練習が望ましいが，現時点では効率の良い練習環境が整っているとは言えない．このように「磨きたい技術が実戦では練習しにくい，練習環境も乏しい」というのは多くのゲームで言われていることであり，解決が望まれている．

本研究の目的は,「ゲームにおける “難解かつ上達していく上で重要なテクニック” を人間プレイヤに効率的に習得させるにはどうしたら良いか」を解明することである．そして具体的に，テトリスにおいて T-spin 技術を習得させるための教育システムを構築すること・およびその有効性を検証することを目標とした．

本研究ではテトリスにおける T-spin を効率的に教育するための補助問題として「詰め T-spin」を提案した．詰め T-spin は囲碁における詰碁や将棋における詰将棋と同様に「二手で T-spin 可能な地形とせよ」といった形式の部分問題であり，上級者が実戦の中で行っている T-spin 構成に必要な能力を訓練できるものになっている．

一般にこういった詰め問題は人間の職人によって作成されるが，人力では少なからず時間がかかるため問題数には限りがある．そこで本研究では T-spin 可能な地形からブロックを抜き取る「逆向き生成法」によってコストをかけずに多様な詰め T-spin 問題の自動生成を実現した．

自動生成された問題にはつまらないものなども含まれるため，本研究では難易

度・面白さ・有用性などの観点からプレイヤにとって良い問題のみを自動で選別するための推定モデルの生成に取り組んだ．実験は初心者プレイヤ 10 人を対象に一手の詰め T-spin 問題を 42 問ずつ解いてもらい，各問題に対して感じた面白さと難しさをそれぞれ 5 段階評価してもらった．結果として，面白さと難しさには相関係数 0.95 を超える強い相関があり,「難しいほど面白い」と感じる傾向が見られた．次に，これらの評価値を目的変数とし，盤面から計算可能な値や着手から得られる特徴量を説明変数として，教師あり学習よって推定モデルを生成した．生成した面白さと難しさの推定モデルはテストデータでの平均絶対誤差がそれぞれ，0.401，0.400 と高い精度となった．

より実戦的な二手問題についても同様の実験を行った．初心者プレイヤ 15 人を対象に面白さと難しさの相関を求め，推定モデルの性能を評価した．結果として，二手問題における面白さと難しさの相関係数は 0.85 を超え，依然として高い相関が見られた．その一方で，二手問題では「難しすぎて面白くない」と感じるような，一手問題にはなかったような傾向も現れた．二手問題の面白さ推定モデルの性能は平均絶対誤差が 0.27 と非常に良い精度を記録した．

一手問題においても二手問題においても，推定モデルによって大量の問題の中から選別された面白さ・難しさ推定値の高い問題を上級者が確認したところ，初心者が反復練習する上で有用な問題が選別できていることが確認できた．

問題の生成と選別が実現したため，これらが実際にプレイヤの T-spin 構成力をどれだけ向上させられるかを検証した．初心者 15 人を対象に，(1) 対戦のみで訓練するグループと (2) 詰め T-spin 問題を組み込んだ訓練をするグループに分けて 2 時間の訓練を行い，どの程度プレイヤが成長したか比較した．結果として (1) の T-spin 利用頻度が全く伸びなかったのに対し，後者 (2) では平均して 3.63 倍という大きな伸びを見せた．さらに，(3) 面白い問題のみを組み込んだ訓練をした場合には 4.34 倍となり，本研究のアプローチの有効性が示された．

Abstract

Games are not only an important culture of human beings but also a good testbed for research in information science. With the development of techniques such as deep learning, computer players have already achieved superhuman levels in many games. Other than this, more and more attention in recent years has been given to research in artificial intelligence (AI) for making games more interesting and improving players' satisfaction. For example, AIs are applied to generate game content such as maps and stories automatically, create teammate computer players, and assist human players in improving playing skills by computer coaches.

Tetris is a popular falling block puzzle game, which was originally designed as a single-player game. In recent years, competition variants for two or more players also gain increasing popularity. In competition variants, when players clear multiple rows at a time, they can "attack" the opponents. Moreover, clearing rows by some special techniques can cause greater damage to opponents than usual. T-spin is such a technique and has a significant influence on players' strength. However, making patterns for clearing rows by T-spin is one of the biggest obstacles for beginners, which makes players struggle to improve their playing skills.

To make T-spin patterns, players need to arrange cleverly the blocks, or so-called tetrominoes, given in random order. The task itself is not easy, not to say the stress caused by time limits on placing tetrominoes or opponents' attacks. Thus, it is difficult to practice the T-spin technique through real plays. Environments close to real plays, but not stressful, for players to practice efficiently are desirable but insufficient. Not limited to Tetris, this is also an issue for many games.

This research aimed to clarify how to assist human players in quickly mastering important but hard-to-learn techniques. More specifically, we focused on a training system for learning the T-spin technique in Tetris. We also verified the effectiveness of the system.

In this research, to train players efficiently for learning T-spin, we proposed "T-spin puzzles." The concept was similar to tsume-go (life and death problems for the game of Go) and tsume-shogi (mating problems for shogi). Players would be asked to accomplish tasks such as "making a T-spin pattern with the two given tetrominoes." Although the puzzles are only parts of the gameplay, players could become familiar with T-spin patterns so as to improve their skills.

Usually, puzzles were designed by human experts. However, a massive generation was impractical due to time-consuming and the limited number of experts. To overcome this problem, we applied the "reverse method," which removed tetrominoes one at a time from T-spin patterns, to generate various T-spin puzzles automatically.

Among generated puzzles, some might be boring and thus unsuitable for training

players. To enable training players by proper puzzles only, with considering factors such as difficulty and interestingness, we worked on models that could predict those factors for puzzles. In more detail, we conducted a subject experiment that asked ten beginners to solve 42 "one-step T-spin puzzles" and rate the interestingness and the difficulty in five-grade evaluation (1, 2, 3, 4, and 5) for each puzzle. In an n-step T-spin puzzle, players can make T-spin patterns with n given tetrominoes in order. From the analyses, the correlation coefficient between interestingness and difficulty was higher than 0.95, which showed a very high correlation. We concluded that the more difficult the puzzles, the more the players tended to feel interesting. Furthermore, we applied supervised learning to predict the beginners' ratings from some manually designed features of the puzzles. The models were highly accurate, where the mean absolute errors (MAEs) of interestingness and difficulty were 0.401 and 0.400, respectively.

A similar experiment was then conducted on two-step T-spin puzzles, which we considered closer to real plays. Fifteen beginners solved and rated the interestingness and the difficulty of 50 two-step T-spin puzzles. From the results, the correlation coefficient between interestingness and difficulty was higher than 0.85, which still showed a high correlation. Different from one-step puzzles, players felt uninteresting about some too-difficult two-step puzzles. The model for predicting the interestingness had a very high accuracy with an MAE of 0.27.

We also asked a Tetris expert to check one-step and two-step T-spin puzzles that were predicted to be interesting or difficult by the models. The expert approved that the selected puzzles should be useful for beginners to practice.

Based on the automatic generation and selection of T-spin puzzles, we verified how much players' T-spin skills could be improved. Fifteen beginners were divided into two kinds of groups for comparison and trained for two hours. The first kind was trained by playing a competition variant of Tetris only, while the second kind by also combining T-spin puzzles. For players in the first kind of groups, the frequency of clearing rows by T-spin did not increase at all. In contrast, the frequency was significantly increased by 3.63 times on average for players in the second kind of groups. Further, among these players, for those provided with only interesting puzzles, the frequency was increased by 4.34 times. The experiments demonstrated the effectiveness of the proposed approach in this research.

目 次

第 1 章　はじめに **1**

第 2 章　対象ゲーム **4**
2.1 テトリス 4
2.1.1 対戦テトリス 5
2.1.2 テトリスの基本戦略 6
2.2 T-spin と「詰め T-spin」 7

第 3 章　関連研究 **9**

第 4 章　詰め T-spin 問題の自動生成 **11**
4.1 T-spin Double 可能な形の自動生成 11
4.2 逆向き生成法による詰め問題の生成 13

第 5 章　面白さ/難しさ推定モデル **14**
5.1 面白さ/難しさに関するアンケート 14
5.1.1 一手問題の結果 14
5.1.2 二手問題の結果 16
5.1.3 個人差の分析 18
5.2 教師あり学習 19
5.2.1 LightGBM 19
5.2.2 学習に用いた特徴量 19
5.3 推定モデルの生成 [初心者] 22
5.3.1 一手問題の推定モデルと選別された問題 22
5.3.2 二手問題の推定モデルと選別された問題 26
5.4 推定モデルの生成 [上級者] 29

第 6 章　教育効果 **31**
6.1 実験設定とトレーニング前の成績 31
6.1.1 トレーニング前の実力測定（CPU と対戦） 31
6.1.2 トレーニング前の実力測定（詰め T-spin 問題） 33
6.1.3 トレーニングメニューの考案とグループ分け 34
6.1.4 トレーニングの実施 35

6.2 トレーニング後の成績とその考察 36
6.2.1 トレーニング後の被験者プレイヤ全体の成績と考察 36
6.2.2 トレーニングメニューによる成長の違い 39

第 7 章 スマートフォンアプリ 42
7.1 より難しい問題の生成 42
7.1.1 ドネイトとは 42
7.1.2 ドネイト問題の生成法 44
7.2 提供形式 46
7.2.1 厳選問題集 46
7.2.2 フラッシュモード 49
7.2.3 練習モード 51

第 8 章 おわりに 52

Appendix 57
A T-spin Triple 57
B 図 5.13，5.14 の二手問題の解答 58
C 図 5.16 の二手問題の解答 58
D 厳選問題集の問題例 59

図 目 次

2.1 テトリスの基本ルール: (a) ブロックの種類，(b) 現在の盤面, (c) L テトロミノの操作, (d) 行の削除, (e) 削除後の盤面 4
2.2 お邪魔ブロック発生の例 5
2.3 一人プレイでよく現れる地形 6
2.4 T-spin の例: (a) 現在の盤面, (b) 落下と接地, (c) 回転と T-spin Double 7
2.5 一手詰め T-spin 問題の例と解答 8
2.6 三手詰め T-spin 問題の例と解答 8

4.1 ベースの地形の生成手順 11
4.2 T-spin を実戦らしくするためのノイズ付与 12
4.3 逆向き生成法による問題生成手順 13

5.1 実験に使用したツール 15
5.2 問題に対する難しさと面白さの付け方 16
5.3 一手問題と二手問題の度数分布の比較 17
5.4 プレイヤ A とプレイヤ B の評価の付け方の違い 18
5.5 一手問題の特徴量の例 20
5.6 二手問題で追加した特徴量の例 21
5.7 一手問題の推定モデルの性能 22
5.8 一手問題の面白さ推定モデルの寄与度 23
5.9 着手テトロミノがベースの地形の空白部分に接する辺数の面白さへの影響 24
5.10 面白さの推定値が高い一手問題の例 24
5.11 面白さの推定値が低い一手問題の例 25
5.12 二手問題の推定モデルの性能 (a) 面白さ (b) 難しさ 27
5.13 面白さの推定値が高い二手問題の例 28
5.14 面白さの推定値が低い二手問題の例 28
5.15 上級者の推定モデル (点線は $x = 4.7$ を表す) 29
5.16 面白さの推定値が 4.7 を超える二手問題の例 30

6.1 人間プレイヤをトレーニングする実験の流れ 31
6.2 訓練前の 3 敵合計の勝数分布 32
6.3 訓練前の実力測定用詰め T-spin 問題の正解数分布 33

6.4 訓練後の 3 敵合計の勝数分布 36
6.5 訓練後の実力測定用詰め T-spin 問題の正解数分布 38
6.6 各グループの勝率の成長率の比較 39
6.7 1 分当たりの T-spin 頻度の向上（個別） 40
6.8 1 分当たりの T-spin 頻度の向上（個別） 41

7.1 ドネイトの手順 43
7.2 ドネイトの例 43
7.3 ドネイト問題の完成形の生成手順例 44
7.4 ドネイト問題を生成する逆向き生成法の手順例 45
7.5 厳選問題集モードを遊ぶ流れ 46
7.6 問題の選択画面 47
7.7 詰め T-spin 問題を解く画面 47
7.8 「よかった，次へ」ボタンと「ふーん，次へ」ボタン 48
7.9 フラッシュモードを遊ぶ流れ 49
7.10 フラッシュモードの表示形式 49
7.11 問題カテゴリの選択画面 50
7.12 練習モードを遊ぶ流れ 51

A.1 T-spin Triple の手順 57
B.2 初心者モデルの面白さの推定値が高い二手問題の解答 58
B.3 初心者モデルの面白さの推定値が低い二手問題の解答 58
C.4 上級者モデルの面白さの推定値が高い二手問題の解答 58
D.5 厳選された三手問題 59
D.6 厳選された三手問題の解答 60

表 目 次

2.1 消去方法によるお邪魔ブロック生成行数の一覧 7

5.1 平均評価値の統計量の一手問題と二手問題の比較 17
5.2 1 人の評価値と 14 人の平均評価値の MAE 26

6.1 訓練前の各 CPU プレイヤに対する勝率と試合時間（秒）の統計量 . 32
6.2 3 つのグループとトレーニングを行う前のスコア 34
6.3 訓練後の各 CPU プレイヤに対する勝率と試合時間（秒）の統計量 . 37
6.4 各グループのトレーニングによる成績の変化 39

第1章　はじめに

近年，ボードゲームやビデオゲームにおけるコンピュータプレイヤ（以下，AI プレイヤ）はハードウェアやアルゴリズムの発展に伴い目覚ましい進歩を遂げている．ボードゲームにおける AI プレイヤはそのゲームの深い理解に基づく入力特徴量や，定石や棋譜といった人間によって蓄積されてきたそのゲームに関する知識を必要とせずとも，人間を遥かに上回る強さを得ることが可能となった．例えば AlphaZero アルゴリズムに基づくプログラム [1] は，チェス・将棋・囲碁という三種類の古典的なボードゲームの中で人智を超えたプレイを実現した．ビデオゲームの例としては，Atari2600 のゲームの多くで深層強化学習 [2, 3] に基づくプログラムが人間よりも高い得点を得ている．これら以外の多くのゲームにおいても，AI プレイヤは人間と同等もしくはそれ以上の実力に到達している．

このように人間よりも強い AI プレイヤの研究が盛んに行われている一方で，最近ではゲームそのものを面白くするために，人間プレイヤの満足度向上を目的とした人工知能の研究も注目されてきている．そのためにマップやストーリーの自動生成，チームメイト，あるいはプレイヤの上達を支援する教育者としての役割など，対戦相手としてだけでなく様々な場面で人工知能が活用されつつある．人間プレイヤを楽しませるための研究として，Hunicke と Chapman[4] は一人用シューティングゲームの難易度を動的に調整する確率的手法を提案した．池田と Viennot[5] はモンテカルロ木探索（MCTS）に基づいた手法により，囲碁で対局相手の人間プレイヤが楽しいと感じるように “手加減” を行う AI プレイヤの作成を試みた．

またプレイヤを教育する目的の研究として，高橋ら [6] は “なぞぷよ問題” の自動生成に取り組んだ．この研究は国内でも高い人気を誇る落下型パズルゲームであるぷよぷよを対象に，同ゲーム内で重要な技術的要素である「連鎖」を学ぶ上で補助となる問題を自動生成する手法を提案している．これにより，プレイヤがより簡単に技術向上を目指せる環境の提供を目指した．単に問題を作成するだけでなく，問題の難易度・面白さ・有用性などを教師あり学習で推定し，それらの水準が高いものを選別する点が重要である．この問題意識やアプローチは我々の研究ともよく似ている．

ぷよぷよと並んで人気の落下型パズルゲームに「テトリス」がある．テトリスは 1 人用ゲームとしてのルールが有名だが，近年では 2 人以上で遊ぶ「対戦テトリス」にも注目が集まってきており，中でも 2019 年 2 月に公開された Nintendo Switch の「Tetris 99」は販売開始から 2ヶ月でユーザ登録数が約 280 万人を超える [7] な

ど人気が高まってきている．テトリスは盤面の横 1 行[1]全てをブロックで埋めることによってその行の全てのブロックを削除できる．対戦テトリスでは同時に複数行を削除することによって対戦相手に攻撃を送ることができるが，通常の削除よりも大きな攻撃力を得られるいくつかのテクニックが存在する．特に「T-spin」と呼ばれる特殊な削除方法による攻撃は強力で，これを上手く構築できるか否かがプレイヤの強さを左右する．その一方で，T-spin が発生する条件は初心者にとって難解であることから “最初の壁” とされており，多くのプレイヤは上達を目指す過程でこの技術の習得に難儀する．

T-spin を構築するためには，ランダムに与えられる着手ブロックをプレイヤが巧みに組み合わせる必要がある．しかしながらランダムな順序で与えられるブロックで狙った形を構築することは容易ではない上，ブロックを配置するまでの時間制限や相手プレイヤからの攻撃によるプレッシャーのために，実戦で T-spin 構築の練習をすることは難しい．そのため，ぷよぷよの連鎖と同様に「実戦的な」「ただし実戦ではない」環境での練習が不可欠である．にも関わらず，現時点では効率の良い練習環境が整っているとは言えない．このように「磨きたい技術が実戦では練習しにくい，練習環境も乏しい」というのは多くのゲームで言われていることであり [8]，解決が望まれている．

本研究の目的は,「ゲームにおける “難解かつ上達していく上で重要なテクニック” を人間プレイヤに効率的に習得させるにはどうしたら良いか」を解明することである．そして具体的に，テトリスにおいて T-spin 技術を習得させるための教育システムを構築すること・およびその有効性を検証することを目標とする．

ゲームにおけるプレイヤの練習・教育を促す方法の一つに「囲碁における詰碁」「将棋における詰将棋」など，単純化した部分問題を用いる方法がある．こういった詰め問題は “娯楽” と “実際の対局における技術力向上の手段” という 2 つの側面を持ち合わせており，初心者だけでなく中級者・上級者がさらなる技術向上を目指すための題材として，一人で練習できることや不得意な技術を選んで重点的に練習できることなどの利点から人気が高い．そこで本研究ではテトリスにおける T-spin を効率的に教育するための補助問題として「詰め T-spin」を提案する．詰め T-spin は詰碁や詰将棋と同様に「二手で T-spin Double 可能な地形とせよ」といった形式の部分問題である．テトリスの上級者は各局面において自盤面と予告されている着手ブロックの状況から T-spin を構築可能かを判断している．詰め T-spin 問題も問題盤面と与えられる n 手のブロックから T-spin の構築を考えるため，実戦に必要な思考プロセスを学ぶ上で有用であると考える．

一般にこういった詰め問題は人間の職人によって作成される．しかしながら，問題作成には少なからず時間がかかるため問題数には限りがある．そこで本研究では詰め T-spin 問題の自動生成法を提案し，コストをかけずに大量の問題を生成することに取り組む．プレイヤの意欲を持続させるためにはプレイヤにとって面白

[1]テトリス上級者のコミュニティでは「n 列消し」または「n ライン消し」という言い方が普通だが，本論文では「n 行消し」で統一する．

い問題である必要がある．しかし，自動生成された問題がプレイヤの得手不得手や嗜好にあったものであるとは限らない．もしプレイヤが自身の実力に適さない難易度の問題を提示されてつまらなさを感じた場合，意欲低下の原因になる．そこで，本研究では自動生成された問題から，難易度・面白さ・有用性などの観点に基づきプレイヤの習熟度にあった良い問題を自動的に選別するための推定モデルの生成を目指す．

また詰め T-spin 問題を使うことでプレイヤをどの程度効率的に成長させることができるかを調べるために，通常の練習方法でトレーニングするグループと，詰め T-spin 問題を活用した練習方法でトレーニングするグループの成長度合いを比較検証することで，その有効性を示す．

プレイヤにとって良い問題を選別するためには推定モデルの精度を高める必要がある．一般に教師あり学習によって生成するモデルの精度を上げるためには，学習に用いるデータが大量であるほど好ましい．また，我々は本研究の成果を効果的に社会へ還元したいと考えている．そこで，これら 2 つの目的を達成するために，詰め T-spin 問題を大量かつ気軽に遊べるシステムをスマートフォンアプリ化し，世に提供する．

なお．本研究の内容はゲームプロブラミングワークショップ 2018 で発表した “テトリスにおける T-spin 構成力向上のための問題作成”[9] および，The 16th International Conference on Advances in Computer Games にて発表した “Improving Human Players’ T-Spin Skills in Tetris with Procedural Problem Generation”[10] らを基に整理・加筆したものである．

本論文の構成は次の通りである．第 2 章では T-spin のテクニックを含むテトリスのゲーム，および詰め T-spin について紹介し，第 3 章ではゲームにおけるコンテンツを自動生成する手法として注目されている PCG（Procedural Content Generation) についてのいくつかの研究を紹介する．第 4 章では逆向き生成法を用いた詰め T-spin 問題の自動生成法について説明し，第 5 章で自動生成された問題の面白さと難しさを推定するモデルの構築とその性能について述べる．第 6 章では詰め T-spin 問題がどの程度プレイヤの技術向上を支援できているかを検証するために行なった実験の詳細について述べ．第 7 章では詰め T-spin 問題をより多くのユーザが触れられるように開発したスマートフォンアプリについて述べる．そして最後に，第 8 章で結論を述べ，今後の研究の方向性について議論する．

第2章　対象ゲーム

2.1　テトリス

テトリス [11] は 1984 年に Pajitonov によって公開された，落下型パズルゲームの始祖である．日本では 1989 年に任天堂から発売されたゲームボーイ版の「テトリス」が 430 万本を超える売上を記録した．テトリスにはシリーズによって細かいルールの違いが存在するが，概ね共通する基本ルールを図 2.1 を用いて以下で述べる．

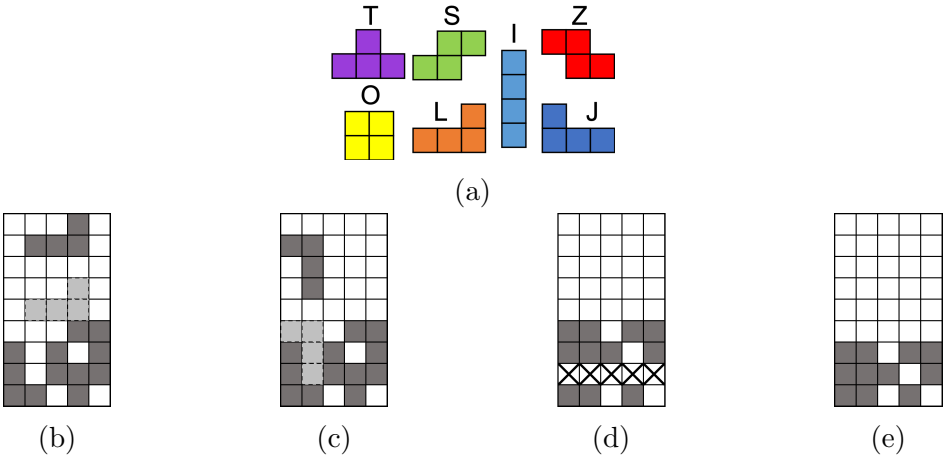


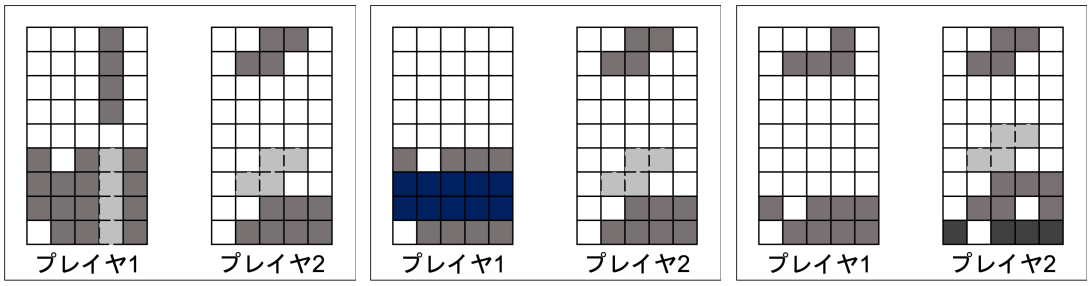
図 2.1: テトリスの基本ルール: (a) ブロックの種類，(b) 現在の盤面, (c) L テトロミノの操作, (d) 行の削除, (e) 削除後の盤面

- テトロミノ：4 つの正方形によって構成されたブロックをテトロミノという．図 2.1(a) に示すように，それぞれをアルファベットに擬えて I テトロミノ，J テトロミノなどと呼ばれ，計 7 種類存在する．
- 盤面とマス：盤面は縦横 20×10 の二次元格子からなり，下方向に重力を持つ．左右と上下の両端に位相的つながりはない．一つのマスは空きであるかブロックの一部によって敷き詰められているものとする．図 2.1(b) に簡略化した図を示す．
- 配テトロミノ：プレイヤにはいずれかのテトロミノが概ねランダムに与えられ，これは数手先まで予告されている．この予告をネクストとも呼ぶ．

- 着手：プレイヤはテトロミノを左右に移動・回転・落下させることができる．図 2.1(c) に L テトロミノの着手例を示す．
- 削除：横 1 行全てのマスがブロックによって埋められると，その行のブロックは全て削除される．削除された行よりも上部にあるブロックは，削除した行数分そのままの形で落下する．削除は一度に 1 行から最大 4 行まで起こり得る．2 行以上の削除は同時消しと呼ぶ．図 2.1(d) に削除が発生する盤面の例，2.1(e) に削除が発生した後の地形を示す．
- ボーナス：一度に削除させる行が多いほど高得点が得られる．また，対戦テトリスにおいては削除させる行が多いほど大きな攻撃を送れる．

2.1.1 対戦テトリス

対戦テトリスは 2 人以上のプレイヤがお互いに攻撃を送り合いながら勝利を目指すモードである．攻撃は相手盤面にお邪魔ブロックを送ることで行う．お邪魔ブロックの生成例を図 2.2 に示す．図 2.2(a) はプレイヤ 1 が I テトロミノを落下させようとしている状態である．I テトロミノをそのまま落下させると，図 2.2(b) に示すように 2 行が埋まる．プレイヤ 1 が 2 行同時消しを行なった場合，図 2.2(c) に示すように，プレイヤ 2 の盤面最下部に迫り上がる形でお邪魔ブロックが 1 行生成される．このとき，お邪魔ブロックは 10 マス全てが埋まっているわけではなく，一つだけランダムに空きマスを持つ．



(a) 初期盤面　(b) 2 行の削除　(c) お邪魔ブロックが発生

図 2.2: お邪魔ブロック発生の例

2.1.2　テトリスの基本戦略

テトリスの基本戦略は一人テトリスの場合と対戦テトリスの場合で異なる．

一人テトリスの場合は,「高い得点を取りたい」場合と「あるステージをクリアしたい」場合がある．後者の場合，ゲームスタート時から図 2.3(a) に示すようなバラバラな地形が与えられるモードもあり，この場合とにかく地形を綺麗に積み上げていくことで，どんな配テトロミノがきても対応可能な地形を保つことが大切になる．

(a) 25 行消去のルールの初期盤面例　　(b) 平積み（右端開け）

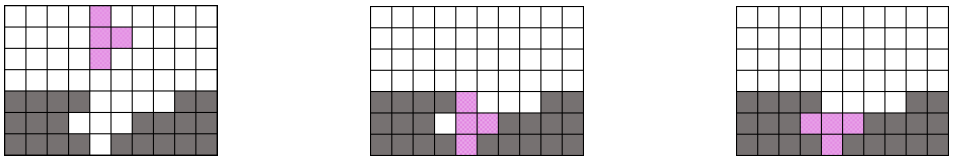
図 2.3: 一人プレイでよく現れる地形

対戦テトリスの場合，一人テトリスの要素に加えて相手への効率的な攻撃を考えなければならない．そのためには攻撃効率というものを考える必要がある．例えば 1 行ずつ消しても攻撃にはならず，I テトロミノを用いた 4 行消しにはボーナスが追加されるなど，消した行数あたりの攻撃行数は比例しない．したがって，対戦テトリスにおいて初心者プレイヤはまず 2.3(b) のような “1 列だけ空白としたまま積み上げ，I テトロミノで 4 行消しを狙う” プレイヤスタイルを取る者が多い．これは対戦テトリスにおける最も基本的な戦略であり,「平積み」とも呼ばれている．

もしプレイヤ全員が平積みで対戦テトリスを行なっているとしたら，プレイヤの強さは「積みの正確性」と「積み上げる速度」といった単純な要素によって決まる．しかしながら，対戦テトリスには対戦のスタイルをより複雑かつ多様化させているテクニックがいくつか存在する．それらの中でも特に重要視されている「T-spin」について次項で詳細を説明する．

2.2 T-spin と「詰め T-spin」

T-spin とは，通常の落下では入らないような隙間に，T テトロミノを一度接地させてから回転させ，うまくねじ込む技術のことを指す．T-spin の例を図 2.4 に示す．図 2.4(a) の状態から T テトロミノを落下させ接地させると図 2.4(b) のような盤面になる．もしこのまま削除が発生すると下から 2 行目に一つ空白が残ってしまう．テトリスはルール上，接地してからわずかではあるが操作可能な時間が存在する．この時間を利用して，図 2.4(b) の状態で T テトロミノを時計回りに 90 度回転させると図 2.4(c) のような地形となる．この例のように，T-spin の後に 2 行の削除が発生する場合 T-spin Double と呼ばれる．また T-spin による 1 行消しを T-spin Single，3 行消しを T-spin Triple と呼ぶ（特殊な消し方であり，本研究では扱わない．Appendix に例を示す.）．



(a) (b) (c)

図 2.4: T-spin の例: (a) 現在の盤面, (b) 落下と接地, (c) 回転と T-spin Double

テトリスにおいて，T-spin を用いて削除すると大きなボーナスが得られる．こちら側が T-spin によって n 行を消した場合，相手にどれだけのお邪魔ブロックを送れるかを表 2.1 にまとめる．通常の削除方法の場合 4 行の同時消しを除いて $n-1$ 行しか送れないのに対して，T-spin による削除は $2n$ 行と効率よく大量の攻撃が送れる．またこれらの中でも T-spin Double は上級者同士の対戦で多用され，実施のための典型的なパターン（いわゆる定石形）も多く生み出されていることから，対戦テトリスにおける最も重要な技術の一つである．

表 2.1: 消去方法によるお邪魔ブロック生成行数の一覧

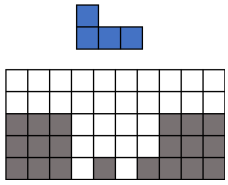
削除行数	通常の消し方	T-spin
1	0	2
2	1	4
3	2	6
4	4	-

多くのテトリスプレイヤは前述したような定石形を記憶し，反射的に構築可能か否かを判断できるようになるまで実戦の中で反復練習している．しかしながら定石形の数は膨大で，全てを暗記することは非常に労力のかかることである．もしパターンを暗記できたとしても，実戦のランダムな配テトロミノを用いて狙った形を構築することは難しい．それゆえ T-spin は多くの初心者の上達に立ちはだかる “最初の壁” となっている．

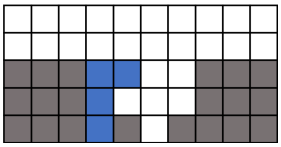
中・上級者らは，実際のプレイの中で配テトロミノを積み上げながら，(1) 盤面の状況およびネクストの状況を正しく認識し，(2) T-spin を構築可能か否か判断し，(3) それを構築するか否かを瞬時に決めている．本研究ではこのプロセスに焦点をあて，“実際のプレイに現れるような T-spin の構築問題” を切り取り，「詰め T-spin」として問題を定義する．詰め T-spin 問題では対戦相手を倒すという目的がない代わりに，「一手で T-spin Double 可能な形とせよ（図 2.5(a)）」「三手で T-spin Double 可能な形とせよ（図 2.6(a)）」「二手で T-spin Triple 可能な形とせよ」といった「手数」と「目標」のセットが与えられる．これを解くことにより (1) から (2) までの思考プロセスを反復練習することが可能である．

この形式は囲碁における詰碁や将棋における詰将棋と似ており，“実際の対局における技術力向上の手段” と “娯楽” という 2 つの側面を持ち合わせている．したがって，一人で練習できることや不得意な技術を選んで重点的に練習できること，さらに楽しみながら練習できることなどの理由から，初心者だけでなく中級者・上級者がさらなる技術向上を目指すための題材としても適切であると考える．

詰め T-spin は，与えられる手数が増えるほど想定し得る構築の組み合わせパターンが増加するため，難しくなることが予想される．本研究においては，実際のプレイで予告されているテトロミノの数と同様に短手数の問題を対象とする．

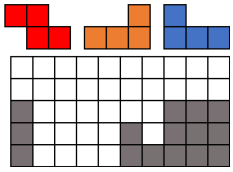


(a) 一手 (J) T-spin Double

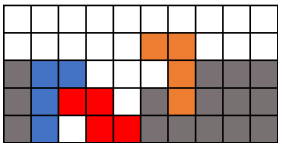


(b) 解答

図 2.5: 一手詰め T-spin 問題の例と解答



(a) 三手 (Z，L，Z) T-spin Double



(b) 解答

図 2.6: 三手詰め T-spin 問題の例と解答

第3章　関連研究

近年，コンピュータプログラムによってゲームコンテンツを自動生成することを目的とした Procedural Content Generation[12] という研究分野が盛んになってきている．しかしながら詰碁や詰将棋など，パズル問題を自動作成するという試みは古くから行われており，その目的・手法は多岐にわたる．通常，プロ棋士やパズル作成家による人手の作成は，“作品” と呼べるような素晴らしい問題作成には適しているが，コストがかかってしまうために大量生産には向かない．多くのプレイヤに楽しんでもらえるレベル・プレイヤの多種多様な嗜好に対応するためには自動生成が必要な場合も多く，その上で可能な限り質も落とさないための試みがいくつも行われてきた．

例えばナンプレ（数独）において，藤原ら [13][14] は問題そのものを自動作成するプログラムを公開・販売している．この研究は人工知能と知識工学を活用し，パズル職人が問題を手作りで問題を手作りする際の思考を問題作成に取り入れることで，“良質な” 問題の自動生成を実現した．藤原は “良質” の定義を「対称性がある美しいデザインで，25 個以下の少量の数字，解き味がよく，レベル分けと技術が身につく問題が，本当の意味での良質．この傾向はむしろ初心者向けの問題に現れる」[15] と述べている．この研究は良質な問題の普及によって「パズルの楽しさを知ってもらうこと」を目標としている．また Mantere ら [16] はそれまで解を求めるために使用されることが一般的であった遺伝的アルゴリズムを数独の問題生成と評価に用いた．

面白い問題を作ることを目的とした研究の例として，広瀬ら [17] は将棋における詰将棋を対象とし，逆算法を用いて問題生成する方法を提案した．逆算法とは，ある詰将棋の詰め手順の最後の方を基本として，詰み手順の最初の方に手を肉付けしていくものである．詰将棋の一つに，曲詰めと呼ばれる詰めあがりに趣向を凝らしたもので詰め上がりが決まっているものがあり，逆算法を使うことで曲詰めの創作が可能になる．広瀬らは「内容の良さ」,「完成度の高さ」,「解き難さ」を評価値の要素とし，曲詰めの問題の作成を行った．実際に生成された問題は，専門誌において好評だった．

山崎ら [18] はパズルゲーム「パネルでポン」を対象とし，面白い問題の生成法を提案した．このとき面白さに関わる要素として連鎖回数のみに着目しており，長い連鎖が行えるほど面白い問題としている．問題創作手順は，まず同じ種類のパネルが 3 枚以上並んでおり，パネルが消える条件を満たしている 6 種類のパターンのパネルを用意する．パターンのどれかを盤面の最下段に配置し，直前に配置し

たパターンに食い込ませるように新たなパターンを配置していき，これらを設定した連鎖回数，手数を満たすまで繰り返すことで問題を作っていく．次に解の個数を調べて問題として成立しているのかを判断している．実験では 100 問中 43 問がパズル問題として成立していた．この手法も逆算法の一種である．

大町ら [19] は，上海ゲームのやりがいのある問題の生成を試みた．上海ゲームは不完全情報性があるため，“ 本来クリア確率が最も高くなる手 ” を選択したとしても，それがゆえにクリア不可能（裏目に出る）ことが生じ得る．このようなことが頻繁に生じては面白くない．大町らは，問題をランダムに生成したあと，上級者エージェントと初級者エージェントにプレイさせ,「初心者のほうが上級者よりも高い確率でクリアできる問題」を，好ましい行動が裏目に出る問題と解釈，棄却するアプローチをとった．

石飛ら [20] は，詰将棋の面白さを推測するために，各問題を証明数探索で解いた場合のノード数や証明数・反証数を用いることを提案している．そしてそれらが面白い条件を満たすように駒を追加・削除することでより面白い問題を生成するという試みを提案している．

そして高橋ら [6] は，なぞぷよ問題の作成にランダム生成法と逆向き生成法を使用し，大量の問題を自動生成し，後者の方が優れていることを示した．さらに盤面から得られる特徴量と被験者実験の感性評価から回帰分析を行い，盤面の特徴が問題の面白さにどのような影響を与えるのかを分析し，面白いと思われる問題を選択出題するという試みを行った．この問題意識やアプローチは我々の研究ともよく似ている．

第4章　詰めT-spin問題の自動生成

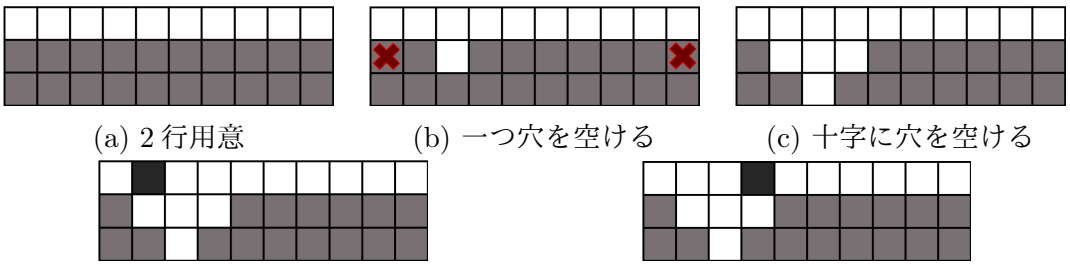
本章では，2.2節で提唱した「詰めT-spin問題」をコンピュータに自動生成させる方法を説明する．面白い問題を作ったり難易度を調整する作業は5章に譲り，ここではともかく詰めT-spin問題になっている盤面＋配テトロミノの組み合わせを発見する方法を述べる．

我々のアプローチは基本的には高橋ら[6]の手法とよく似ている．すなわち，まず「これでT-spin Doubleが可能である」ような完成形，いわば詰み状態を生成することにする．そのうえで，そこから一手分のテトロミノを取り除くことで「一手の詰め問題」，続いてさらに一手分取り除くことで「二手の詰め問題」を作っていく．詰み状態の生成を4.1節で，取り除くプロセスを4.2節で詳述する．

4.1　T-spin Double可能な形の自動生成

我々はT-spinの完成形を図4.1に示すような手順で自動生成した．

1. 全てのマスが埋まっている2行を用意する．図4.1(a)
2. 2行目のうち，端2つを除いた8マスのいずれかを“中心”とする．図4.1(b)
3. 中心，中心の左右および中心の下を空きマスとする．図4.1(c)
4. 中心の左上か右上のどちらかのマスを埋め，完成．図4.1(d), 図4.1(e)



(a) 2行用意　(b) 一つ穴を空ける　(c) 十字に穴を空ける

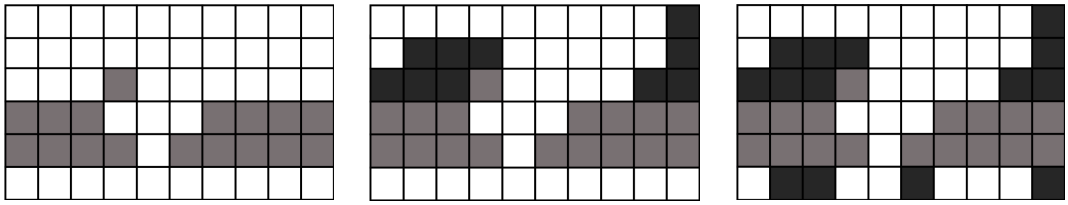
(d) 左上を埋める　(e) 右上を埋める

図 4.1: ベースの地形の生成手順

図4.1の手順によって生成されたT-spin Double可能な地形は，T-spin Doubleを行う上で最小の地形である．本研究ではこれを“ベースの地形”と呼ぶ．

実際の対戦で現れる T-spin Double 可能な地形は，ベースの地形のような最小限のものであることはごく稀であり，大抵はいくつかの凹凸を含んでいる．これを考慮し，我々はベースの地形に対して図 4.2 に示すような手順でノイズ地形を加えることにより，より実戦に現れるような地形に近づけた．

まず，図 4.2(b) に示すようにベースの地形の 1 行上と 2 行上に「真下のマスが埋められていた場合，10〜20%程度の確率で埋める」という条件でノイズ地形を追加した．さらに，図 4.2(c) のようにベースの地形の 1 行下にもランダムに虫食いが発生するような地形を追加した．



(a) ベースの地形を用意　(b) 上部にノイズ地形を追加　(c) 下部にノイズを追加

図 4.2: T-spin を実戦らしくするためのノイズ付与

4.2 逆向き生成法による詰め問題の生成

我々は，T-spin 可能な地形から任意のテトロミノ一つ抜くことで詰め T-spin 問題を生成する「逆向き生成法」を考案した．以下で逆向き生成法による詰め T-spin 問題の生成手順を図 4.3 を用いて説明する．ここで，説明を簡単にするために，本論文において指定されていない場合，T-spin は T-spin Double のことを指すこととする．

図 4.3(a) に示すように，まず前節の方法で T-spin 可能な地形を生成する．この地形から任意のテトロミノを一つ抜き取ることにより，詰め T-spin の一手問題を作成することができる．例えば図 4.3(a) の地形から図 4.3(b) のように S テトロミノを取り除くことで，図 4.3(c) に示す一手詰めの問題を生成できる．同様に，図 4.3(d) のように J テトロミノを抜き取ることで図 4.3(e) に示すような一手問題が生成可能である．さらに，4.3(f) のようにこれらのテトロミノを両方とも抜き取ることで 4.3(g) に示すような二手問題を生成することもできる．同様の手法を繰り返すことで，n 手問題を作成することができる[2]．

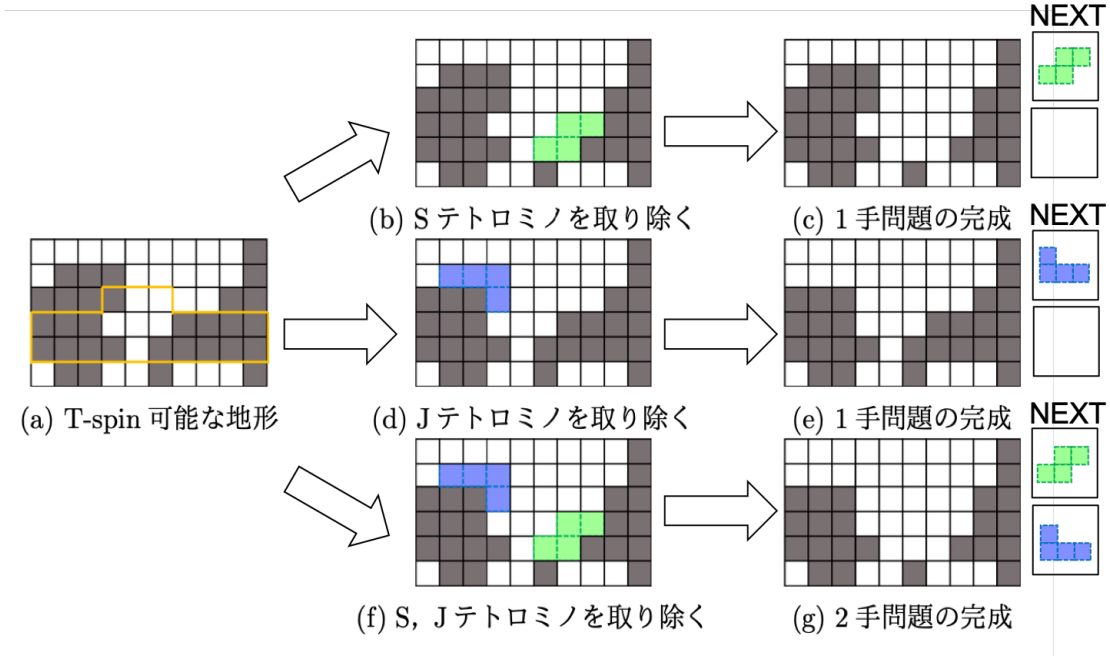


図 4.3: 逆向き生成法による問題生成手順

以上より，我々は n 手の詰め T-spin 問題の自動生成を実現した．なお，本研究ではまず初心者を対象とした問題生成を行なっていくため，“横滑りやラインの削除，回転入れなどの高度な技術を要するような問題” は対象外とし，単純な落下だけで構築可能な問題のみを対象とした．

[2] この方法で生成された問題には 2 つ以上の解が存在する可能性があるが，現時点では調べていない．

第5章　面白さ/難しさ推定モデル

逆向き生成法によって大量に生成した問題の中には，T-spin を学ぶ上で「適切な難易度で面白いもの」や「簡単すぎて（あるいは難しすぎて）つまらないもの」など様々な質のものが存在する．初心者の技術力向上を支援するコンテンツとしてプレイヤのモチベーションを維持するためには，可能な限り面白く，かつ適切な難易度の問題を出題する必要がある．本研究ではプレイヤにとって良い問題のみを提供するために，プレイヤが問題から感じる面白さと難しさについてのアンケート調査を行い，教師あり学習によって推定モデルを生成することで，面白い問題のみを選別することを提案した．次節から詳細について述べる．

5.1　面白さ/難しさに関するアンケート

我々は，詰め問題から人間が感じる難しさと面白さには相関があると仮説を立てた．なぜなら，極端な例を言えば小学生に微分積分をやらせても大抵は難しすぎて面白くなく感じ，逆に高校生に四則演算を解かせても簡単すぎてつまらなく感じるからだ．このことから，我々はプレイヤにとって “良い問題” となるためには，少なくとも実力にあっていることが必要だろうと考えた．

そこで我々は，まずはテトリスにほとんど触れたことがない初心者を対象に，詰め T-spin 問題から感じる面白さ，難しさを調べるためのアンケート形式の被験者実験を行った．

5.1.1　一手問題の結果

「ゲームはほぼ毎日やる」「テトリスには熱中したことがない」「22〜26 歳の」「男性」10 人を対象に，以下の手順で一手詰め T-spin 問題を 42 問出題するアンケート実験を行った．

- 手順 1：T-spin の概要を説明
- 手順 2：出題される一手問題を解く
- 手順 3：各問題に対して「面白さ」「難しさ」の評価をつける
- 手順 4：手順 2，3 を 42 問分行う．

このとき手順 3 は 5 段階評価で行い,「難しさ」は答えを導き出すためにどれだけ悩んだか,「面白さ」は答えに意外性を感じたり爽快感があったかなどの度合いを評価してもらった．このときどの被験者にも同様に，4 章の方法で作られた問題を特別な選別をすることなく用意し，問題盤面の虫食いが多くなっていくような順序で出題した．実験のツールは図 5.1 を用いて行った．

ツールを用いた実験において，被験者は問題を 1 問解いたら元の問題に戻ることはできず，どの問題も 1 分以内という制限時間を設け，時間内に解けなかった場合は不正解とした．T-spin の完成形を作るために問題から与えられる “着手テトロミノ” の操作は，ドラッグ・ドロップによる移動と右クリックによる回転で行い，正解の配置を見つけたら「解けた」ボタンを，1 分経っても解けなかった場合「解けなかった」ボタンをそれぞれ押してもらった．その後，面白さと難しさについて 5 段階評価を付けてもらい，その問題の解答を確認してから次の問題へ進んでもらった．

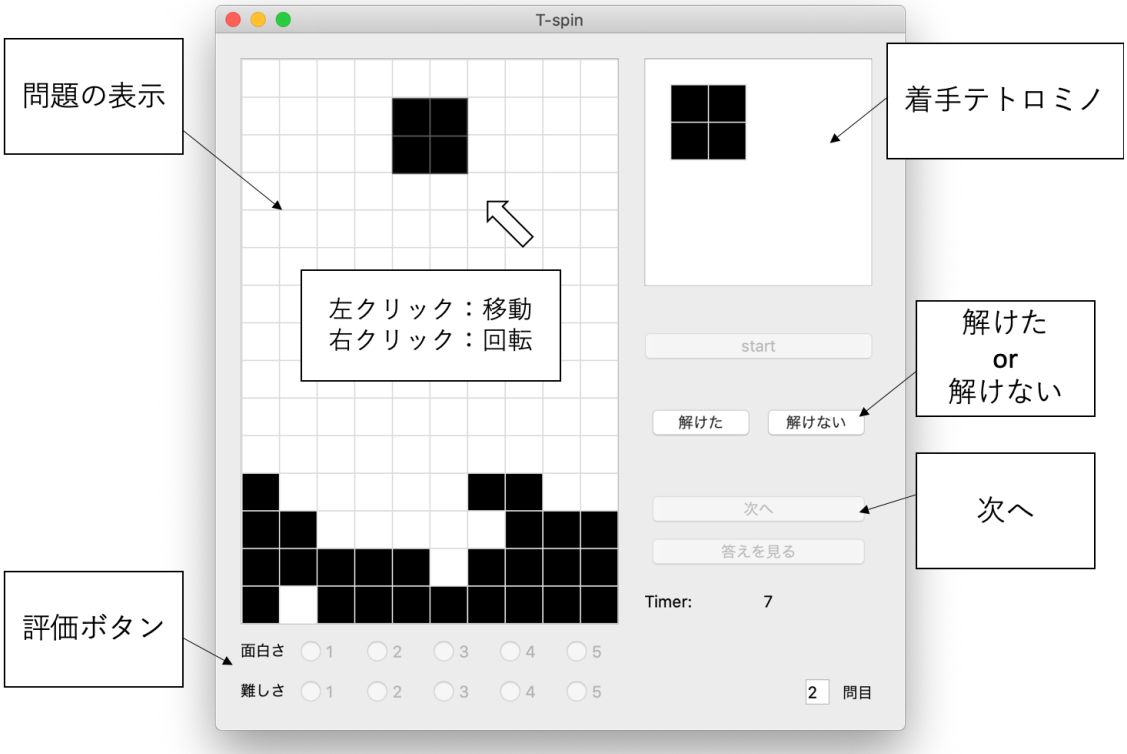
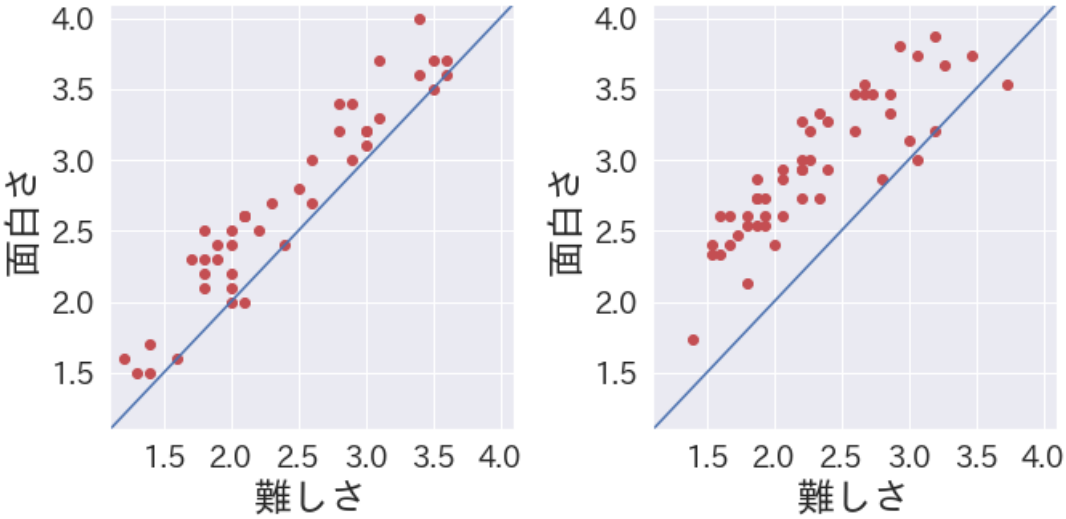


図 5.1: 実験に使用したツール

実験で出題した 42 問の一手詰め T-spin 問題それぞれに対する 10 人からの面白さ，難しさの平均評価値を算出し，横軸：難しさ，縦軸：面白さでプロットしたグラフを図 5.2(a) に示す．グラフより，難しさと面白さには非常に強い正の相関

（相関係数：0.952）があることがわかり，初心者プレイヤは一手問題に対して “難しく感じる問題ほど面白い” という評価をつけることがわかった．



(a) 一手問題 (青線は $y = x$)　　(b) 二手問題 (青線は $y = x$)

図 5.2: 問題に対する難しさと面白さの付け方

5.1.2　二手問題の結果

続いて，二手問題についても同様に調査を行った．15 人のテトリス初心者（うち 8 人は一手問題の実験と同一人物）を対象に，50 問の二手詰め T-spin 問題を出題した．このとき，問題は 4 章の方法でランダムに生成した二手問題で，それぞれのプレイヤが解く出題順序もランダムに設定した．またツールは図 5.1 と同じものを使用し，一手目を置いた後にその手を元に戻すことはできない．

各問題に対する難しさと面白さの平均値をプロットしたグラフを図 5.2(b) に示す．二手問題においても一手問題と同様に，難しさと面白さの間に強い正の相関（相関係数 0.859）があることが確認できた．その一方で，最も面白さが高いのは難しさが 3.2 付近であり，それより難しいと感じる問題は面白さが下がっている，いわゆるマウンテンカーブの傾向が見られた．よって初心者プレイヤは二手問題に対して “難しいと感じるほど面白いが，中には難しすぎて面白くないとものもある” と感じる傾向があることがわかった．

一手問題 42 問と二手問題 50 問それぞれの平均評価値の統計量を表 5.2 に示す．また一手問題と二手問題それぞれの面白さと難しさの分布ヒストグラムを図 5.3 に示す．表 5.2 で一手問題と二手問題の比較を行うと，被験者が追加されていることを踏まえても以下のことが見て取れる．

- 傾向 1：一手問題よりも二手問題の方が面白さの平均値が高い．

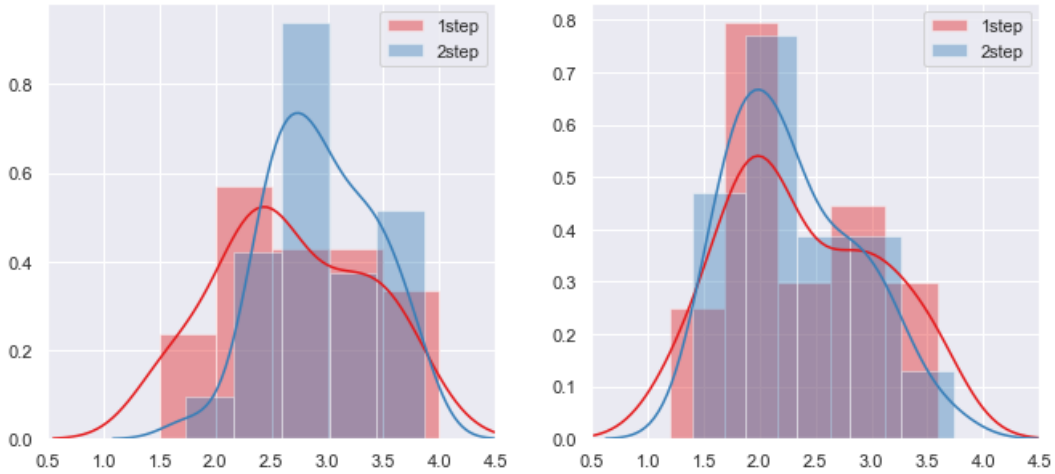
- 傾向 2：一手問題と二手問題を比べたとき，標準偏差は二手問題の方が小さい．
- 傾向 3：一手問題と二手問題の難しさの平均値にはそれほど差がない．

傾向 1 および図 5.3(a) から，初心者にとって二手問題は一手問題よりも面白いと感じる問題が多いことがわかる．その理由として，二手問題は一手問題と比べて「着手テトロミノが T-spin の構成に全く関与しない “本当に面白くない問題”」が少なくなるために面白さを感じやすくなることが言える．また着手テトロミノが増えることによって，実際のテトリスで遊んでいる感覚に近づいたことも要因の一つであると考えた．これらは傾向 2 の結果にも影響を与えていると考えた．

傾向 3 について，我々は一手問題よりも二手問題の方が難しさ平均値が高くなることを予想していた．なぜなら二手問題は着手テトロミノの組み合わせにより複雑な問題が増える可能性が高くなると考えたからだ．にも関わらず傾向 3 や 5.3(b) のように両者にそれほど差が確認できなかった理由は，プレイヤは難しさの絶対的な評価軸を持っているわけではなく，そのとき解いたいくつかの問題から相対的に評価を定めているからであると考えた．そのため，前述した予想のような結果は一手問題と二手問題を混ぜた問題を出題することで明らかとできるだろう．

表 5.1: 平均評価値の統計量の一手問題と二手問題の比較

	平均値		標準偏差	
	面白さ	難しさ	面白さ	難しさ
一手問題	2.683	2.385	0.676	0.684
二手問題	2.950	2.306	0.478	0.569



(a) 面白さ　　(b) 難しさ

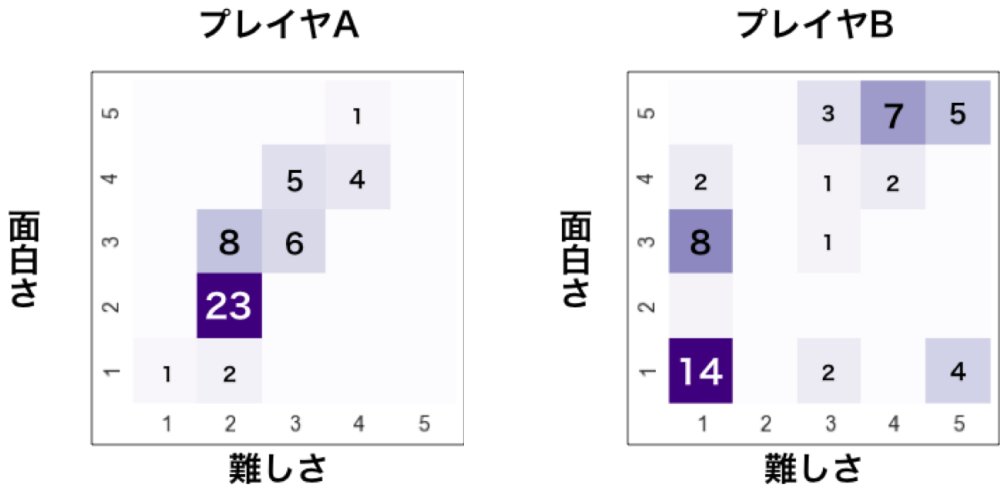
図 5.3: 一手問題と二手問題の度数分布の比較

5.1.3　個人差の分析

上記 2 つの分析は初心者プレイヤである被験者全員の平均評価値，すなわち “初心者プレイヤに共通する傾向” について行った．つまり，個人のデータではなく 10 人ないし 15 人の平均値だけを見てきた．次に，我々は “初心者プレイヤの評価の個人差” について注目した．

二手問題のアンケート結果について，15 人の被験者の中からプレイヤ A とプレイヤ B の難しさと面白さ評価の違いを図 5.4 に示す．プレイヤ A は難しさと面白さの付け方に相関が確認できるのに対して，プレイヤ B の付け方はバラバラであることがわかる．また，プレイヤ A は評価値が 2，3，4 と中立的な評価が多いのに対して，プレイヤ B は 1 や 5 の極端な評価も多く付けていることが見て取れる．つまり，初心者プレイヤが二手問題から感じる面白さ・難しさには似通った傾向が見られる一方で，評価の付け方には個人差があることがわかった．

このような結果を踏まえると，本来であれば個人の嗜好に合わせた問題提供こそが理想的と言えるが，面白さや難しさといったデータを 1 人分だけ扱うとなると，データの少なさからノイズの影響が大きくなることが懸念される．そのため，まずは複数人のデータから平均値を算出し，これを用いて教師あり学習することによって “実力層に合わせた推定モデル” を作成していく．次節からはこの手法について説明する．



(a) プレイヤ A　　(b) プレイヤ B

図 5.4: プレイヤ A とプレイヤ B の評価の付け方の違い

5.2 教師あり学習

初心者プレイヤが問題から感じる面白さや難しさには似た傾向があることがわかったため，本節では初心者にとって良い問題を選別することに取り組む．しかしながら，上級者が「面白い」「有用だ」と思う問題をルールベースで選別したとして，初心者にとってそれが面白い問題であるとは限らない．そこで我々は教師あり学習を用いることで，初心者と似た感性で問題を評価できる推定モデルを生成することに取り組んだ．

教師あり学習とは機械学習の手法の一つであり，主にラベル付けされた訓練データからモデルを学習し，未知のデータや将来のデータを予測できるようにするための手法である．本研究では前節の被験者実験によって集めた面白さ・難しさの評価値集合と盤面から計算可能な値や着手の種類・手順から得られる特徴量を訓練データセットとして利用し，教師あり学習を行うことで面白さ・難しさそれぞれの推定モデルを生成した．

本節では使用した教師あり学習フレームワークと，学習に用いた詰め T-spin 問題の特徴量について簡単に説明する．

5.2.1 LightGBM

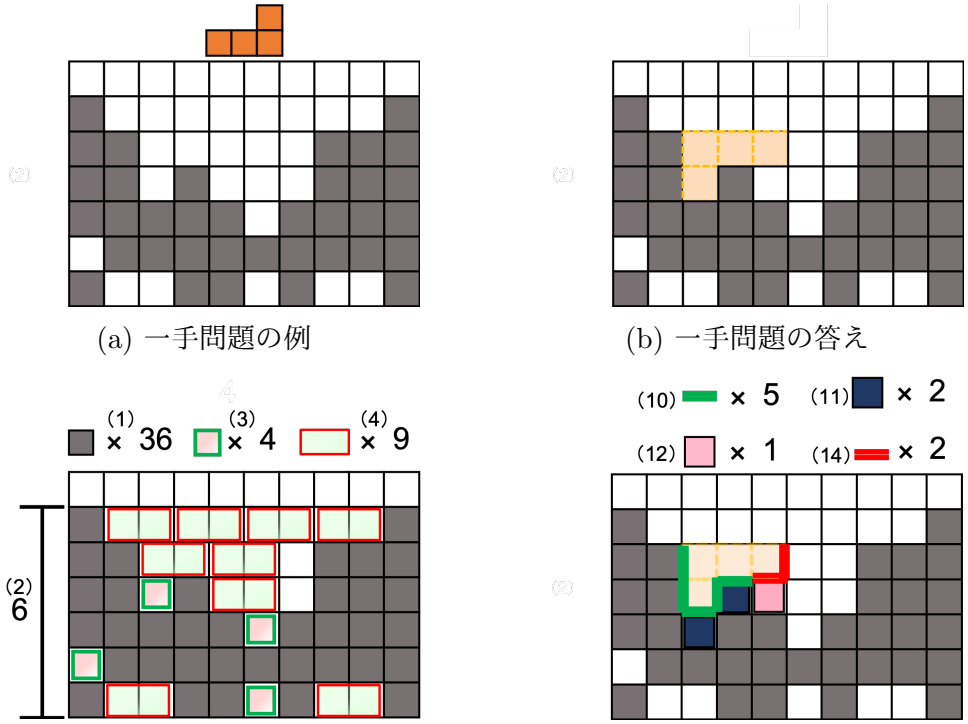
LightGBM[21] とは，Microsoft によって開発された決定木ベースのアルゴリズムを使用する勾配ブースティングフレームワークである．詳細については割愛するが，大まかに以下のような特徴を持つ．

1. モデル訓練にかかる時間が短い
2. メモリ効率が良い
3. 推測精度が高い
4. 大規模データセットも学習可能
5. モデルに対する特徴量の寄与度の可視化が容易

プレイヤにとって良い詰め T-spin 問題を作っていく上で，どの特徴量が推定モデルに影響を与えているかを知ることは重要である．よって得られた結果の意味を解釈しやすい決定木ベースであり，寄与度の可視化が容易である本フレームワークを使用した．

5.2.2 学習に用いた特徴量

教師あり学習によって面白さ推定モデルを生成するために，一手問題では 14 種類，二手問題では 22 種類の地形や着手から得られる特徴量を使用した．以下に一手問題で扱った特徴量を示す．また，わかりにくい特徴量について問題例と共に図 5.5 に例示する．



(a) 一手問題の例　(b) 一手問題の答え

(c) 地形から得られる特徴量の例　(d) 着手から得られる特徴量の例

図 5.5: 一手問題の特徴量の例

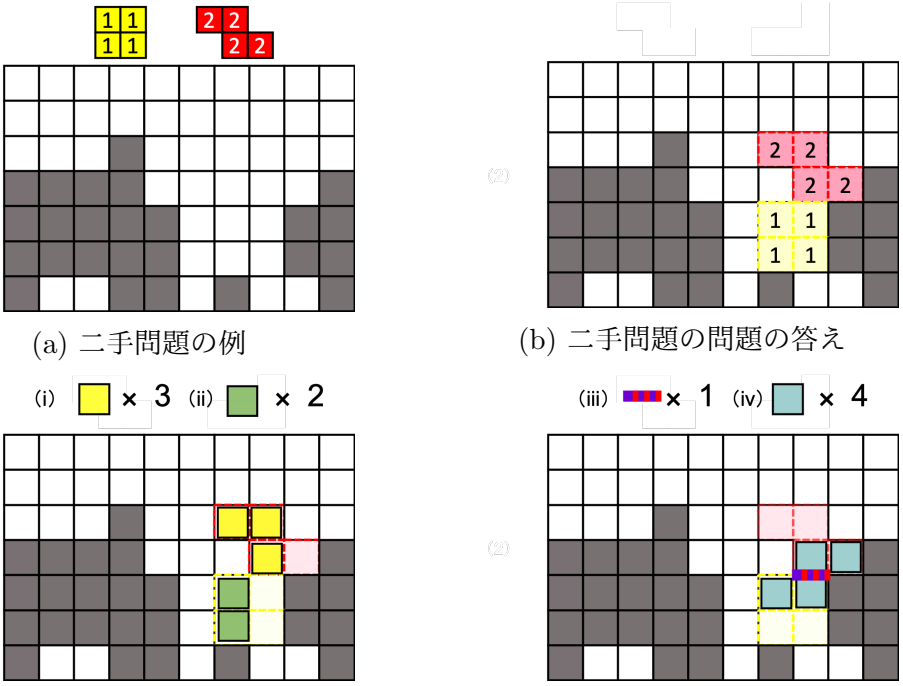
(1) 問題に既に埋められているマスの数 （図 5.5(c) では 36）
(2) 地形の最大高さと最小の高さの差 （図 5.5(c) では 6）
(3) 地形に存在する一つの空白の数 （図 5.5(c) では 4）
(4) 地形に存在する 2 つ続きの空白の数 （図 5.5(c) では 9）
(5) 地形に存在する 3 つ続きの空白の数
(6) 置くテトロミノの種類
(7) 置くテトロミノが T,J,L であるかどうか
(8) 置くテトロミノが S,Z,I であるかどうか
(9) 置くテトロミノが O であるかどか
(10) 正解位置が周囲の埋まったマスと接している辺の数 （図 5.5(d) では 5）
(11) 正解位置の一つ下にある埋まったマスの数 （図 5.5(d) では 2）
(12) 正解位置の一つ下にある空白の数 （図 5.5(d) では 1）
(13) 正解位置の下にある全ての空白の数
(14) 正解位置がベースの地形の空白部分に接している辺の数（図 5.5(d) では 2）

それぞれの特徴量には面白さ・難しさに寄与すると考えられる理由があって採用している．例えば (3) は値が大きければ大きいほど虫食いの数が増えるような特徴量である．通常のプレイでは虫食いは少ないほど好ましい．したがって，もし

この値が大きければ初心者にとって見慣れない，あるいは悪い印象を受けるような問題地形になることが予想され，これが面白さや難しさに寄与するのではないかと考えた．

続いて二手問題で追加した特徴量を示す．こちらも問題と共に図 5.6 に例示する．二手問題は一手問題よりも難しい問題となることが想定される．なぜなら一手目と二手目の組み合わせによって複雑な形を構築可能になるからだ．そこで我々は (iii) のような一手目と二手目の着手テトロミノが接し合う辺の数が面白さ・難しさに寄与するのではないと考えて採用した．より難しい問題になった場合やより高い精度を得る必要がある場合には特徴量をリッチにする，あるいは盤面を Convolutional Neural Network で処理するなどの必要があるかもしれない．

(i) 一手目の上にある二手目のブロックの数 （図 5.6(c) では 3）
(ii) ベースの地形の十字の下にある着手ブロック数 （図 5.6(c) では 2）
(iii) 一手目と二手目のブロックが接している辺の数 （図 5.6(c) では 1）
(iv) ベースの地形の下二行に配置される着手ブロック数 （図 5.6(c) では 4）
(v) 一手目のテトロミノの種類



(a) 二手問題の例　(b) 二手問題の問題の答え

(c) 特徴量の例 (i)，(ii)　(d) 特徴量の例 (iii)，(iv)

図 5.6: 二手問題で追加した特徴量の例

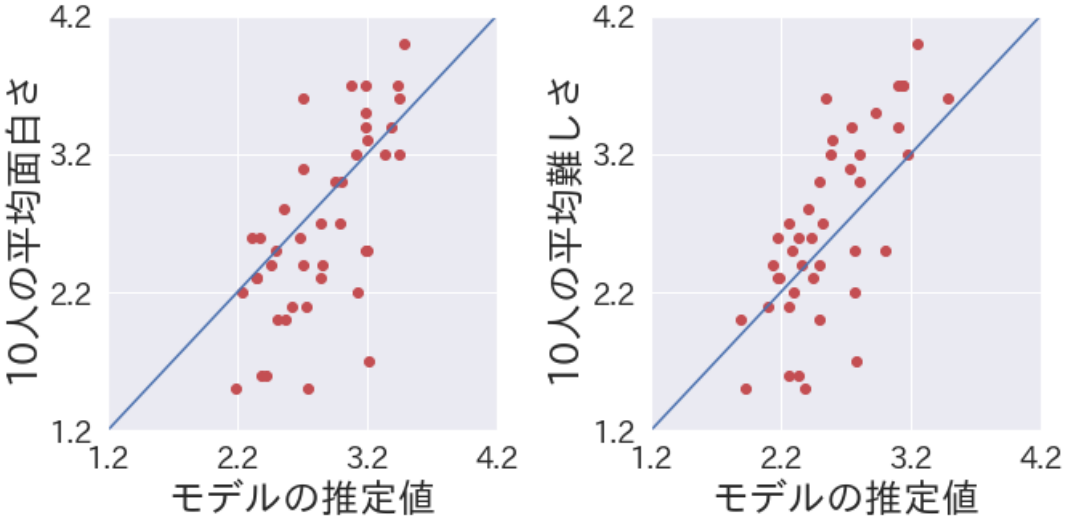
5.3 推定モデルの生成 [初心者]

人間プレイヤを訓練するための面白い問題を自動生成するために，LightGBM フレームワークを用いた教師あり学習によって，一手問題と二手問題それぞれの面白さ推定モデルを生成した．

5.3.1 一手問題の推定モデルと選別された問題

5.1.1 節で集めた 42 問の一手詰め問題に対する初心者 10 人の平均評価値を元に，面白さ，難しさ推定モデル[34]を生成し，3-Folds 交差検証で性能を検証した．面白さ推定モデルの精度は，対称平均絶対誤差率（SMAPE）が 15.65%，平均絶対誤差 (MAE) が 5 段階評価で 0.401 であった．このとき，MAE はおおよそ ±0.7 の範囲内だった．また，難しさ推定モデルについては SMAPE が 15.28%，MAE が 0.400 とほぼ同じであった．

横軸：テストデータの推定値，縦軸：10 人の平均値でプロットした結果を図 5.7 に示す．このグラフに示している青線はプロットの回帰直線ではなく，相関を見やすくするための $y = x$ 直線である．多少誤差はあるものの，推定値が高いものは実測値も高く，低いものは低く推定できている傾向が見られる．



(a) 面白さ推定モデル (青線は $y = x$)　　(b) 難しさ推定モデル (青線は $y = x$)

図 5.7: 一手問題の推定モデルの性能

[3]本研究の面白さモデルと難しさモデルはどちらも推定精度を高めるとためにハイパーパラメータをチューニングしている．本来であればチューニングに用いたデータではないテストデータでの精度を示すべきであるが，データ数の関係上今回はそれを行えていない．

[4]ハイパーパラメータ;learning rate:0.001，boosting type:gbdt，objective:regression，metric:rmse，sub feature:0.8，num leaves:2，min data:1，min hessian:1，verbose:-1

生成した面白さ推定モデルに対して，どの特徴量が大きく寄与しているかを表す「寄与度」を図 5.8 に示す．最も面白さに寄与しているのは「ベースの地形の空白部分に接する辺数」で，次いで「2 連続の穴の数」,「正解位置の真下の空きマスの数」，そして「L,J,T テトロミノが使われているかどうか」などといった特徴量が寄与していることがわかった．それぞれが推定モデルの結果に大きく寄与した理由について，我々は以下のように考察した．

- 「ベースの地形の空白部分に接する辺数」が多い問題は，言い換えれば “自分自身の着手によって T-spin 可能な地形を構築する問題かどうか” を表している．例えば図 5.9(a) に示すような問題はこの値が少なく，はじめからほとんどベースの地形が構築されている上，着手は全く関係ない場所である．にも関わらずその着手が正解であり，人間プレイヤは解いている感覚が少なくそれに伴い低い評価を付けることが予想され，実際にそうであった．逆に図 5.9(b) のようにこの値が高い問題は，はじめの盤面で T-spin 可能となりそうな地形は初心者にとって見つけづらく，パズルを解いている感覚を味わえるため，面白い問題であると評価することが予想され，こちらも実際にそうであった.（要確認）これらより,「ベースの地形に接している辺の数」は値が大きいほど「面白い」方向に寄与しているようである．
- 「正解位置の真下の空きマスの数」は値が 1 でもあれば，正解の着手テトロミノの下に虫食いを作ることになる．T-spin に慣れていない初心者にとって虫食い地形を作ることは可能な限り避けたいはずであり，それをある意味裏切る形の正解は面白さに影響を与えているようである．図 5.9(b) の問題はこれの典型例でもあり，もし最下段右から 4 個目のマスが埋まっていた場合，難易度と共に面白さが低下することが想定できる．
- 「配ミノが L，J，T かどうか」は，これら 3 つのテトロミノは回転によって 4 種類の形状を取り，これは他の 4 つのテトロミノよりもバリエーション数が多い．回転後のミノの形状を想像することは，まだテトリスをはじめたばかりのプレイヤにとって簡単なことではない．それゆえ問題で考える要素が増え，面白さに影響を与えているようである．

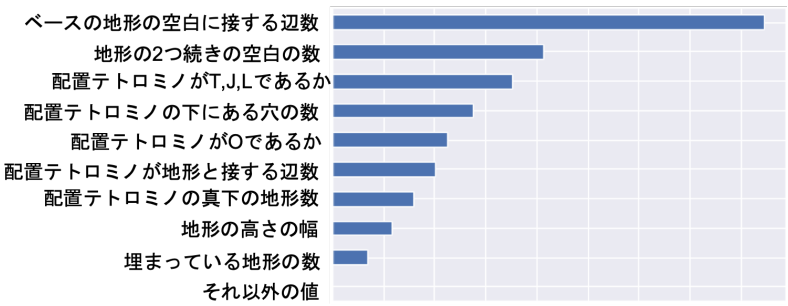
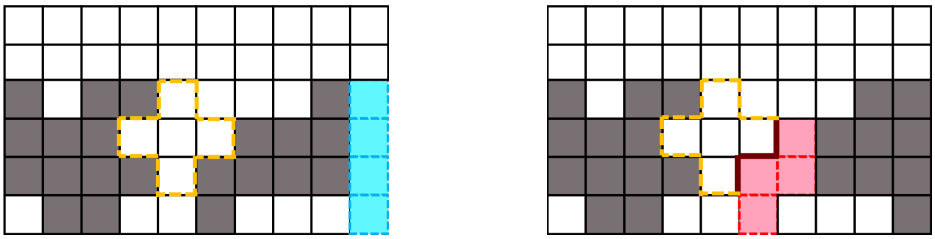


図 5.8: 一手問題の面白さ推定モデルの寄与度

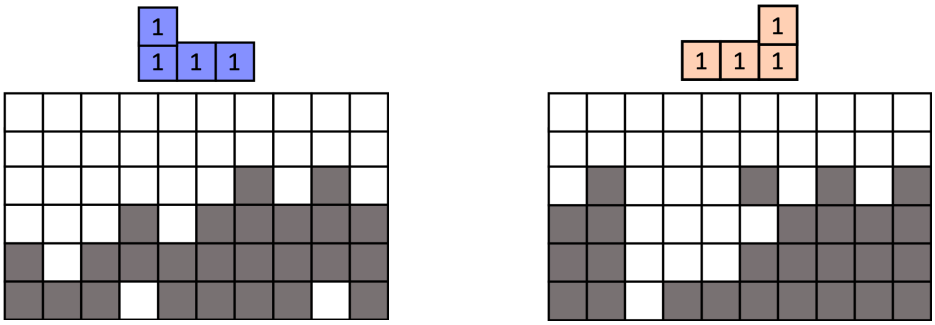


(a) 接する辺数:0　　(b) 接する辺数:3

図 5.9: 着手テトロミノがベースの地形の空白部分に接する辺数の面白さへの影響

面白さモデルの推定値が高かった問題を図 5.10 に示す．図 5.10(a) の問題は，盤面地形をみると凹凸がたくさんあるが故に，もし初心者プレイヤが実戦で直面した場合，どのように捌いたら良いか悩む難しい盤面であるように見える．しかしながら問題から与えられた J テトロミノを正解位置に配置できると，これらの凹凸を少し解消できるだけでなく T-spin が可能な地形にもなる面白い問題である．図 5.10(b) の問題は初心者プレイヤであれば与えられた L テトロミノを用いて 2 行消しをしたくなりそうな盤面であるが，あえて虫食いを作る正解配置を行うことで T-spin 可能な地形が作成可能な面白い問題である．

他の面白さモデルの推定値が高い問題も上級者がチェックしたところ，確かに初心者プレイヤの実力向上のための反復練習に適した，実戦的で多様な問題群が生成できていることが確認できた．



(a)　　(b)

図 5.10: 面白さの推定値が高い一手問題の例

面白さモデルの推定値が低かった問題を図 5.11 に示す．図 5.11(a) の問題は，正解の着手を行ったとしても，高低差の激しい乱れた地形となるため「解けた」という感覚があまり味わえない問題である．図 5.11(b) の問題は，T-spin を狙う狙わないに関わらず「普通はそこに置くだろう」という配置の仕方であり,「解いた」という感覚があまり味わえない面白くない問題であると言える．他の面白さモデルの推定値が低い問題も同様に，実戦ではあまり使いたくないような問題やあからさまに正解がわかる問題などが多く現れていた．

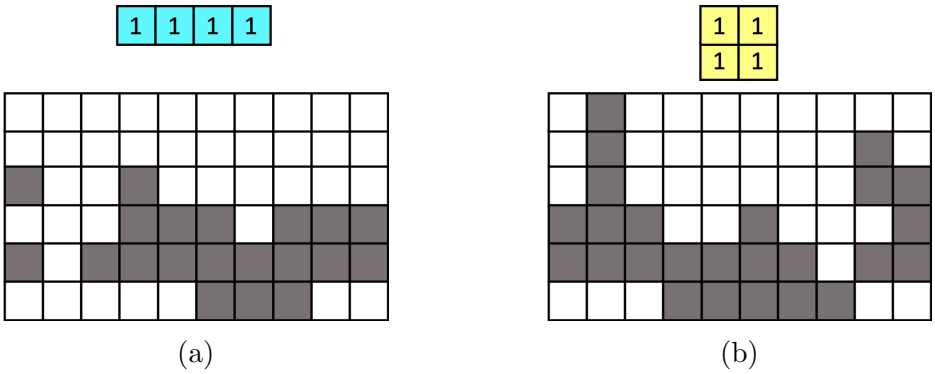


図 5.11: 面白さの推定値が低い一手問題の例

以上より，面白さ推定モデルを活用することによって面白い一手問題の選別をある程度実現することができた．

5.3.2 二手問題の推定モデルと選別された問題

5.1.2 節で集めた 50 問の二手詰めの問題に対する初心者 15 人の平均評価値を基に，面白さ[5]，難しさ推定モデル[6]を生成し,10-Folds 交差検証で性能を検証した．面白さ推定モデルの精度は，SMAPE が 8.741%，MAE が 5 段階評価で 0.261 であった．またこのとき，MAE はおおよそ ±0.4 の範囲内で，一手問題の面白さ推定モデル（MAE=0.401）よりも精度が高くなった．また，難しさ推定モデルについてはSMAPE が 14.77%，MAE が 0.345 とこちらも一手問題の面白さ推定モデル (MAE=0.400) よりも高い精度となった．

面白さと難しさについて，1 人の評価値とそれ以外の 14 人の平均評価値から MAE を求めることを 15 人分行い，それらの平均・最良・最悪を調べた結果を表 5.2 に示す．面白さについて，平均 MAE は 0.898 と推定モデルの MAE よりも 3 倍以上大きくなり，最良 MAE の場合でも 0.529 であることから推定モデルの方が優れていることが言える．難しさについて，最良 MAE が 0.482 と推定モデルの精度と近い評価をつけるプレイヤの存在が確認できたが，精度は推定モデルの方が優れている上，このプレイヤから評価値を集めることはやはりコストがかかるため，本推定モデルは有用であると言える．

表 5.2: 1 人の評価値と 14 人の平均評価値の MAE

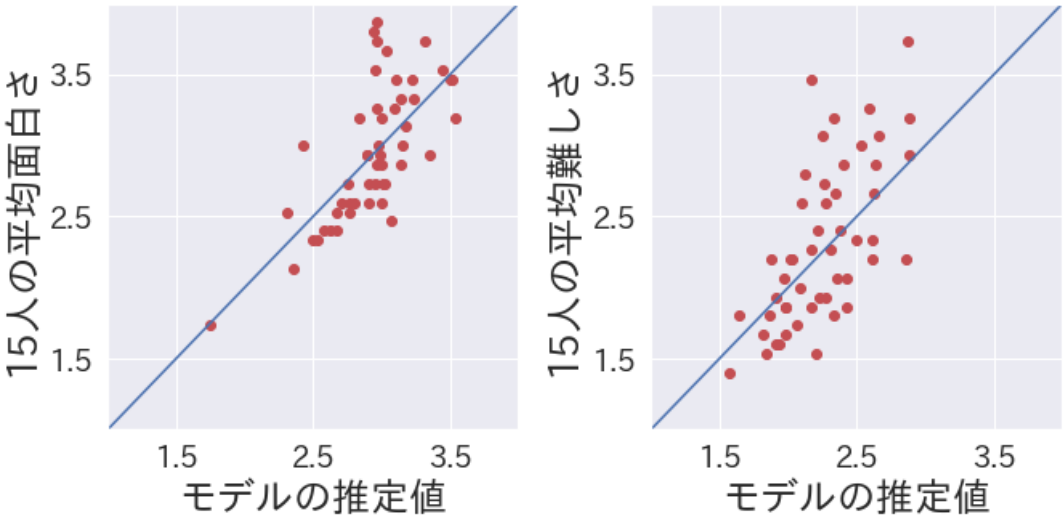
	面白さ	難しさ
推定モデル	0.261	0.345
平均	0.898	0.807
最良	0.529	0.482
最悪	1.368	1.175

[5]ハイパーパラメータ;learning rate:0.125，n estimeters: 1000，boosting type:gbdt，objective:regression，metric:rmse，sub feature:0.8，num leaves:3，min data:3，min hessian:1，verbose:-1

[6]ハイパーパラメータ;learning rate:0.15，n estimeters: 1000，boosting type:gbdt，objective:regression，metric:rmse，sub feature:0.68，num leaves:12，min data:3，min hessian:1，verbose:-1

横軸：テストデータ推定値，縦軸：15 人の平均値でプロットした結果を図 5.12 に示す．このグラフに示している青線はプロットの回帰直線ではなく，相関を見やすくするための $y = x$ 直線である．二手問題でも一手問題と同様に，推定値が高いものは実測値も高く低いものは低く推定できている傾向が見られる．

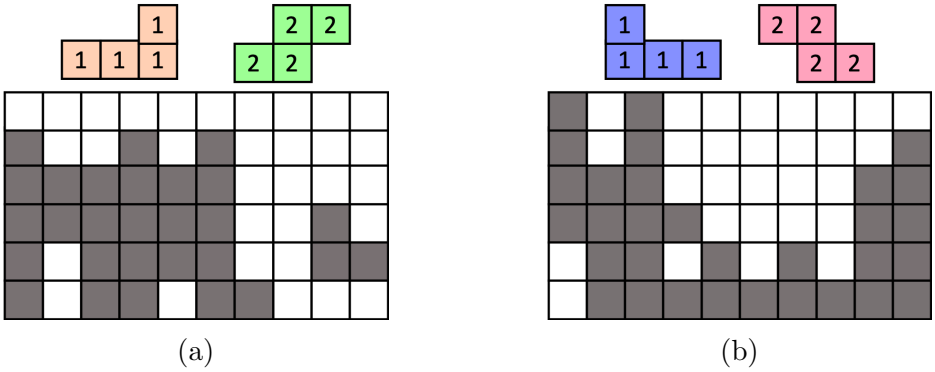
生成した面白さ推定モデルに大きく寄与している特徴量は図 5.8 とほとんど同様の傾向であった．しかし中には「ベースの地形の下にある着手ブロック数」であったり,「一手目と二手目のブロックが接している辺数」といった二手問題ならではの特徴量も寄与していることがわかった．



(a) 面白さ推定モデル (青線は $y = x$)　　(b) 難しさ推定モデル (青線は $y = x$)

図 5.12: 二手問題の推定モデルの性能 (a) 面白さ (b) 難しさ

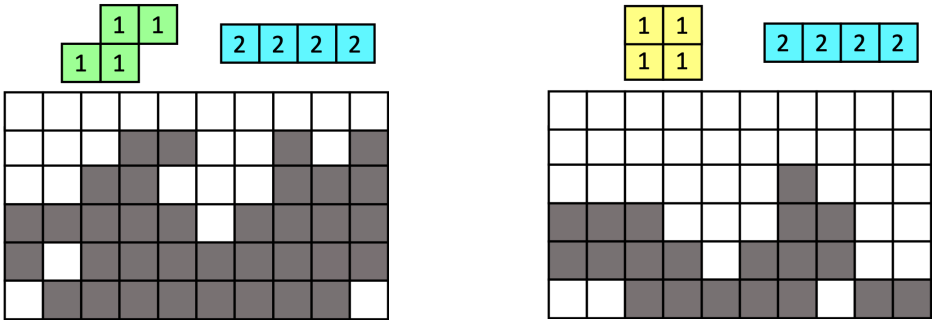
面白さモデルの推定値が高かった問題を図 5.13[7]に示す．図 5.13(a) の問題は，一手目の L テトロミノを置く場所が非常に見つけづらい問題である．しかしながら解答を見つけられたとき，その配置の意外性から面白さを感じられる問題である．図 5.13(b) の問題は，少なくとも平積みを行っていく上では絶対に行わない置き方をするため，初心者プレイヤは一筋縄では解答を見つけられない．したがって，この問題が解けた初心者プレイヤは難しいパズルが解けた時のような面白みを感じられるだろう．



(a) (b)

図 5.13: 面白さの推定値が高い二手問題の例

面白さモデルの推定値が低かった問題を図 5.14 に示す．二手問題の面白さ推定値が低い問題に関しても，図 5.11 で示した一手問題で面白さ推定値が低い問題と似た傾向を持っている．図 5.14(a) の問題は既に T-spin の形がほとんど完成していて面白みがなく，図 5.14(b) の問題の一手目は「普通はそこに置くだろう」という配置が正解になっていて面白くない．



(a) (b)

図 5.14: 面白さの推定値が低い二手問題の例

以上より，一手問題に続いて，二手問題でも面白い問題の選別をある程度実現することができた．

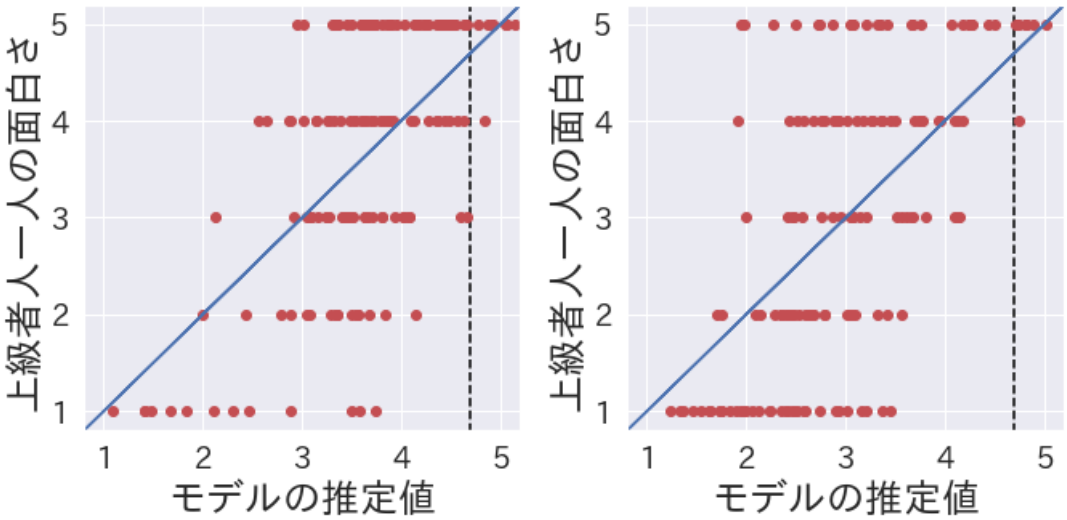
[7]二手問題の解答例は Appendix に示す

5.4 推定モデルの生成 [上級者]

前節で初心者とって面白く，あるいは難しく感じるような一手，二手詰め T-spin 問題をある程度良い精度で選別する方法を提案した．本節では詰め T-spin 問題をより幅広い実力層に楽しんでもらうために，上級者一人を対象としたアンケート調査を行い，推定モデルを生成することで問題の選別を試みる．

147 問の二手詰めの問題に対する上級者一人[8]の評価値から，面白さ・難しさ推定モデル[9]を生成し，10-Folds 交差検証で性能を検証した．面白さ推定モデルの精度は，SMAPE が 24.423%，MAE が 5 段階評価で 0.783 であった．また，難しさ推定モデルについては SMAPE が 36.77%，MAE が 0.940 であった．

横軸：テストデータ推定値，縦軸：上級者 1 人の評価値プロットした結果を図 5.15 に示す．このグラフに示している青線はプロットの回帰直線ではなく，相関を見やすくするための $y = x$ 直線である．また，どちらのグラフにおいても，推定値=4.7 に点線を示している．



(a) 面白さ推定モデル (青線は $y = x$)　　(b) 難しさ推定モデル (青線は $y = x$)

図 5.15: 上級者の推定モデル (点線は $x = 4.7$ を表す)

上級者モデルの SMAPE や MAE といった精度は初心者モデルと比較して大きく劣っているように見受けられる．これは人間がつける評価は様々な要因によって変化するため，それらのばらつきがモデルの精度に強く影響を与えた結果であると考えた．初心者モデルで精度が高まった理由は，複数人の平均値を取ること

[8] この上級者は筆者の及川である．及川は 2019 年 8 月に開催された “第 10 回 日本 e スポーツ学生選手権ぷよぷよテトリス部門” [22] のテトリス競技部門でベスト 16 に入るなどの成績を残している．

[9] ハイパーパラメータ;learning rate:0.125，n estimeters: 100，boosting type:gbdt，objective:regression，metric:rmse，sub feature:0.3，num leaves:8，min data:3，min hessian:1，verbose:-1

によって評価のばらつきを抑える効果があり，モデルの精度の安定につながったのだと考えた．その一方で，生成した上級者モデルで問題の選別が不可能かというと，そうとも言い切れない．図 5.15(a) と図 5.15(b) を見ると，モデルによる推定評価値が 4.7 を超えている問題であれば実際の評価値が 4 を超える問題のみ，すなわち面白い・難しい問題のみが選別できることがわかる．難しさについては難易度の高い問題が必ずしも良い問題とは限らないが，面白さについては面白い問題の方が好ましいため，面白くない問題を正しく推定できることの重要性はそこまで高くない．よって図 5.15(a) のモデルはある程度有用性があると言える．

面白さ推定モデルの推定値が 4.7 を超えている問題から 2 問を図 5.16 に示す．図 5.16(a) の問題は，虫食いをあえて作らなければならない着手があることに加え，一見 T-spin を作りそうなところとは別の位置で T-spin を作るため，解答がわかった時に爽快感がある面白い問題だと言える．図 5.16(b) の問題は，上級者プレイヤにとっては必須テクニックとされる有名な定石形に似た地形でありながら，それとは異なった置き方が正解となるため，上級者が面白いと感じやすい問題と言える．推定値が低い問題に関しては，図 5.14 に示したものと類似していたため割愛する．これらの傾向から，上級者プレイヤの嗜好をある程度模倣したモデルの生成を実現できたと言える．

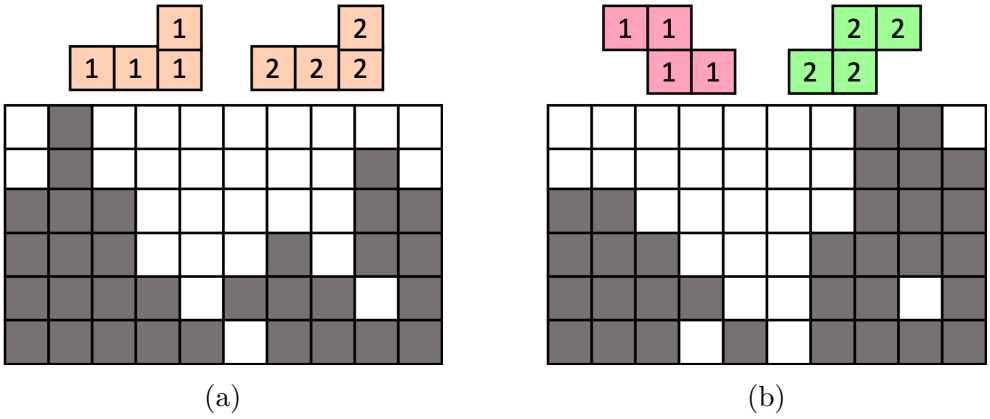


図 5.16: 面白さの推定値が 4.7 を超える二手問題の例

第6章　教育効果

4 章で詰め T-spin 問題の自動生成を実現し，5 章で自動生成した問題からプレイヤにとって面白い問題のみを選別する方法を提案した．本章では，面白い詰め T-spin 問題を使うことでプレイヤをどの程度効率的に成長させることができるかを調べるために，通常の練習方法でトレーニングするグループと，詰め T-spin 問題を活用した練習方法でトレーニングするグループの成長度合いを比較検証することで，その有効性を示す．実験の設定と各測定の結果を 6.1 節で，トレーニング前後の成績の変化を 6.2 節で詳述する．

6.1　実験設定とトレーニング前の成績

5.1.2 節の被験者と同一人物の 15 人の初心者を対象に，図 6.1 に示すような手順で実験を行った．各手順の詳細について次節から説明していく．

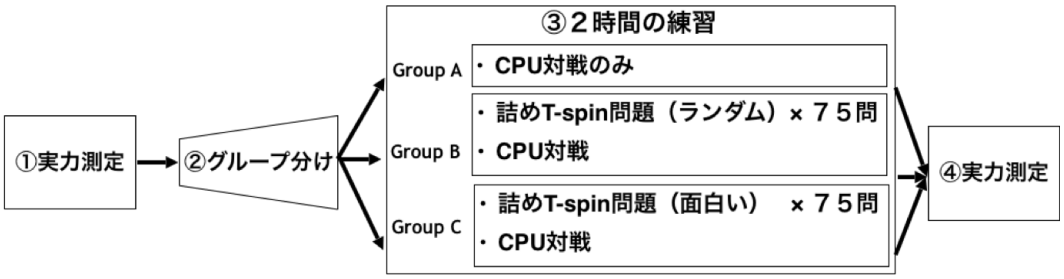


図 6.1: 人間プレイヤをトレーニングする実験の流れ

6.1.1　トレーニング前の実力測定（CPU と対戦）

トレーニングを行う前の被験者プレイヤの対戦テトリスにおける強さを調べるために，3 人の CPU プレイヤとそれぞれ 2 本先取を 1 セットとした 3 セット先取で対戦を行ってもらい，その戦績を記録した．対戦環境には Steam 版ぷよぷよテトリス [23] を採用し，同ゲーム内に内蔵されている CPU プレイヤの中から，強さに違いがある 3 人の CPU プレイヤであるアルル (弱)，エス（中），ゼッド（強）を選び，この順で対戦してもらった．被験者プレイヤは 3 セット先取を通して少なくとも 6 試合，最大で 9 試合の対戦を行った．

本測定では各被験者プレイヤの勝ち数と負け数，各試合にかかった時間，各試合で行った T-spin による攻撃回数，および試合のプレイ動画を記録した．被験者プレイヤが 3 人の CPU プレイヤと対戦した時の勝ち数の分布を図 6.2 に示す．一人のプレイヤを除いて，多くのプレイヤは 3 勝以下とほとんど勝てていないことがわかる．

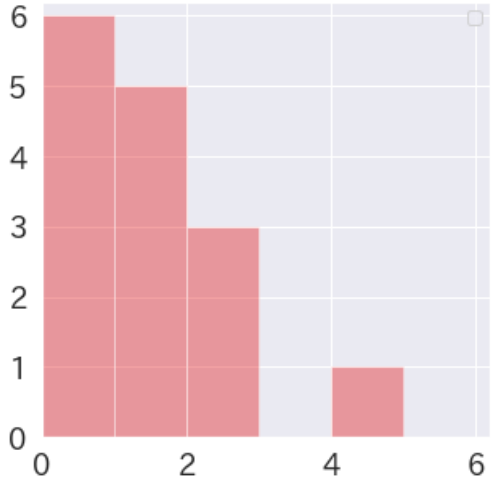


図 6.2: 訓練前の 3 敵合計の勝数分布

被験者プレイヤ全体の各 CPU プレイヤに対する勝率，およびそれぞれの CPU プレイヤとの対戦にかかった時間の統計量を表 6.1 に示す．勝率に関して，CPU プレイヤが強くなるにつれて勝率も高くなっているが，これは被験者は初心者プレイヤであることから発揮できる実力にムラがあることに加え，対戦順の関係で徐々にプレイに慣れて勝利できたプレイヤがいたことが原因であると考えた．時間に関して，各 CPU プレイヤとの平均および最短対戦時間が徐々に短くなっていることから，その他の多くのプレイヤは CPU プレイヤの強さが上がるにつれて早めに倒されたことが想定でき，各試合の録画からもそれが確認できた．1 回目の実力測定での 3 本先取が終わるまでの平均時間は 431.4 秒だった．

表 6.1: 訓練前の各 CPU プレイヤに対する勝率と試合時間（秒）の統計量

	アルル（弱）	エス（中）	ゼッド（強）	全試合
勝率	0.13	0.16	0.23	0.17
平均時間	503.0	442.1	349.2	431.4
最短時間	262	201	161	161

6.1.2　トレーニング前の実力測定（詰め T-spin 問題）

我々は詰め T-spin 問題をどれだけ解けるかという能力もプレイヤの実力を表す指標として活用できるのではないかと考え，これ調べるために 15 問の実力測定用の二手詰め T-spin 問題を各プレイヤに解いてもらった．このとき問題は一人の上級者によって選ばれた易しい，中程度，難しい問題を 5 問ずつ，徐々に難易度が上がるように出題した．実験に用いたツールは図 5.1 のものをベースに，各問題に 30 秒の時間制限を設け，問題の解答の開示はしないように設定したものを用いた．

トレーニングを行う前の実力測定用詰め T-spin 問題の正解数の分布を表すヒストグラムを図 6.3 に示す．問題の正解数の平均値は 12.1 問で最頻値は 4 人いた 13 問だった．全ての問題を正解したプレイヤは 1 人だけで，最も少ない正解数は 9 問で 1 人だった．

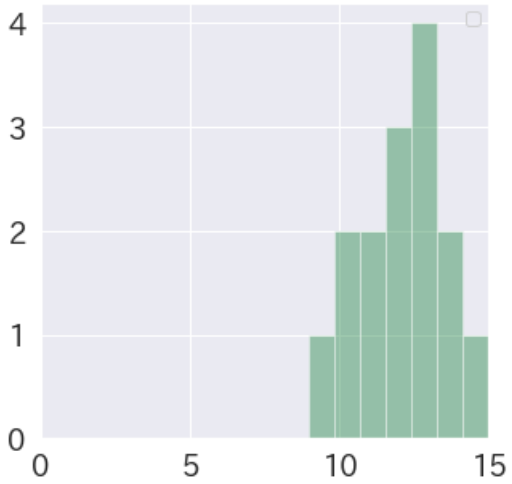


図 6.3: 訓練前の実力測定用詰め T-spin 問題の正解数分布

6.1.3 トレーニングメニューの考案とグループ分け

詰め T-spin 問題を活用した場合とそうでない場合のトレーニング効果の違いを比較するために，我々は以下 3 つのトレーニングメニューを提案した．

- メニュー A：40 分間のトレーニングで，全ての時間を CPU との対戦に費やす方法．この 40 分を 3 回繰り返す．
- メニュー B：40 分間のトレーニングで，訓練用の二手詰め T-spin 問題を 75 問解き，残り時間で CPU と対戦する方法．このとき，問題は 4 章の方法でランダムに生成された問題を提供した．この 40 分を 3 回繰り返す．
- メニュー C：40 分間のトレーニングで，訓練用の二手詰め T-spin 問題を 75 問解き，残り時間で CPU と対戦する方法．このとき，問題は 5 章の面白さモデルで選別された面白い問題（推定値が 3.4 より大きいもの）のみを提供した．この 40 分を 3 回繰り返す．

メニュー A とメニュー B，C の結果を比較することによって,「対戦のみでトレーニングしたグループの結果」と「詰め T-spin 問題を活用してトレーニングしたグループの結果」を比較できる．また，メニュー B とメニュー C の結果を比較することによって,「ランダムに生成された詰め T-spin 問題を活用してトレーニングしたグループの結果」と「面白い詰め T-spin 問題のみを活用してトレーニングしたグループの結果」を比較することができる．

上記のメニューを割り当てるために，我々は各グループの実力が均等となるように被験者プレイヤ 15 人を表 6.2 に示すような 3 つのグループに分割した．このとき，各プレイヤについて 6.1.1 節および 6.1.2 節で測定した「3 セット先取での勝数」＋「実力測定用詰め T-spin 問題の正解数」のスコアを実力の指標とし，この値がどのグループも近い値となるように分けた．各グループの勝ち数および詰め T-spin 問題の正解数も併せて表 6.2 に示す．

表 6.2: 3 つのグループとトレーニングを行う前のスコア

A (対戦のみ)		B (ランダムな二手問題)		C (面白い二手問題)	
勝数	正解数	勝数	正解数	勝数	正解数
7	59	4	62	5	61

6.1.4 トレーニングの実施

トレーニングは 1 セット 40 分で行い，各セット間に 10 分程度の休憩を挟みながら合計 3 セットの 120 分間行った．メニュー A を行うグループのプレイヤは CPU プレイヤとの対戦のみでトレーニングを行った．メニュー B と C を行うグループのプレイヤは，トレーニング用の二手詰め T-spin 問題を 75 問解いた後，CPU 対戦を行うことでトレーニングした．トレーニング用の詰め T-spin 問題を解く際，図 5.1 と同様のツールを用いたが，各問題には 20 秒の制限時間を設け，必ず解答を開示するように設定したものを用いた．75 問の T-spin 問題によるトレーニングは約 20 分かかり，CPU プレイヤとの対戦は 40 分間の残り時間で行った．

次節では被験者全体のトレーニング前後の成績比較と練習メニュー別に見た成長度合いの比較を行う．

6.2　トレーニング後の成績とその考察

本節では被験者プレイヤの実力が 120 分間のトレーニングによってどれだけ変化したかを比較し，その成長について考察する．主に被験者プレイヤ全体についての結果および考察を 6.2.1 節に，練習メニュー別の成長度合いの比較を 6.2.2 節に記す．

6.2.1　トレーニング後の被験者プレイヤ全体の成績と考察

3 セットのトレーニングが終了した後，6.1.1 節と 6.1.2 節と同様の方法で被験者プレイヤの実力を測定した．被験者プレイヤが 3 人の CPU プレイヤとトレーニング後に対戦した時の勝ち数の分布を図 6.4 に示す．図 6.2 に示したトレーニング前の分布と比較して，依然として勝ち数が 0 のプレイヤは多いが，勝ち数 3 以上を獲得できたプレイヤが増加したことも確認できた．このことから，何人かの被験者プレイヤはいずれかの CPU プレイヤと互角かそれ以上の実力に成長したことがわかる．

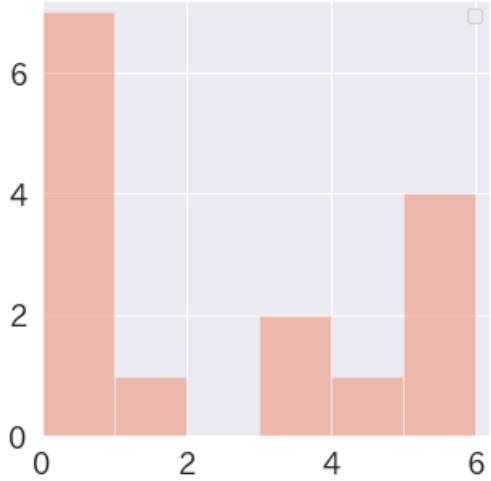


図 6.4: 訓練後の 3 敵合計の勝数分布

被験者プレイヤが各 CPU プレイヤと戦ったときの勝率と，それぞれの CPU プレイヤとの対戦にかかった時間の統計量を表 6.3 に示す．

表 6.1 に示したトレーニング前の結果と比較して，被験者プレイヤの CPU プレイヤに対する平均勝率を比較すると，トレーニング前の 0.17 から 0.36 と倍以上の値となっている．また試合時間に着目しても，トレーニング前の平均および最短時間が長くなっていることから，勝てなかったとしてもすぐに倒されてしまうプレイヤが減っており，120 分のトレーニングを通して多くの被験者プレイヤの実力が向上したことがわかる．

表 6.3: 訓練後の各 CPU プレイヤに対する勝率と試合時間（秒）の統計量

	アルル（弱）	エス（中）	ゼッド（強）	全試合（訓練後）	全試合（訓練前）
勝率	0.5	0.2	0.37	0.36	0.17
平均時間	573.6	470.46	361.06	468.4	431.4
最短時間	260	247	180	180	161

各 CPU プレイヤとの勝率を比較すると，最も勝てなかった相手はエスであることがわかる．この結果は，上級者が被験者プレイヤの対戦の録画をみて確認したところ，各 CPU プレイヤが用いていると思われる戦略が原因であると考えた．アルルとゼッドは図 2.3(b) に示したような平積みを行う傾向にあるため，CPU プレイヤが積み上げているタイミングで止め刺すための隙ができることや，まとまった攻撃を送ってくるためその攻撃を利用しやすいことから，初心者にとって戦いやすい CPU プレイヤであった．一方でエスはそこまで地形を高く積み上げることはせず，ひたすらミスなく 2 行，3 行消しを行ってくるプレイヤであるため隙が少なく，ミスプレイを起こしやすい初心者プレイヤにとって戦いにくい相手であった故に勝数が伸び辛かったのだと考察した．

トレーニング後の実力測定用詰め T-spin 問題の正解数の分布を表すヒストグラムを図 6.5 に示す．このとき，測定用の問題は 6.1.2 節の一回目の測定と同じ問題を同じ順で出題した．

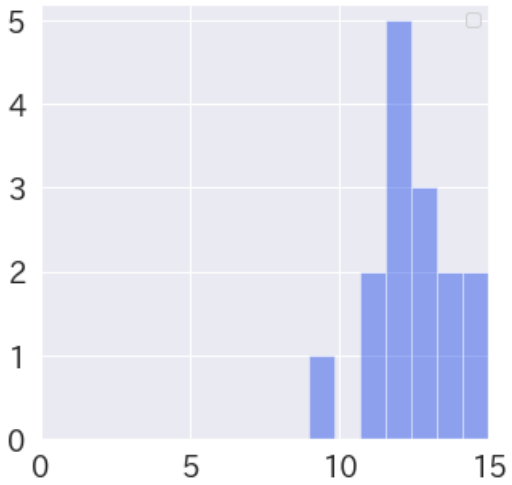


図 6.5: 訓練後の実力測定用詰め T-spin 問題の正解数分布

ヒストグラムをトレーニング前の図 6.3 とトレーニング後の図 6.5 で比較すると，正解数の分布に大きな違いは見られなかった．実力測定問題の正解数の平均値は 12.5 問で，最頻値は 5 人いた 12 問だった．全ての問題を正解したプレイヤは 2 人に増えたがトレーニング前に全問正解したプレイヤとは別人物であった．最も少ない正解数は 9 問で 1 人だったがトレーニング前のプレイヤとは別人物であった．実力測定用詰め T-spin 問題の結果について，次節で詳述するようなグループ別の比較を行ったとしても有意的な変化は見られなかったため，詰め T-spin 問題をプレイヤの実力測定に使うためには，工夫が足りなかったのだと考えた．改善のためには，例えば問題にかけられる時間的制約をより厳しくしたり，三手以上のより難しい問題を用意したり，あるいはある程度多くのプレイヤに決まった問題セットを解いてもらって，そのときの正解率を参考に測定用の問題セットを作ることなどが必要だと考える．

本節では被験者全体が 120 分のトレーニングでどれだけ成長したかを示した．次節ではトレーニング結果をグループごとに比較・考察することで，詰め T-spin 問題の有用性を示す．

6.2.2 トレーニングメニューによる成長の違い

本実験では被験者を 6.1.3 節の方法で練習メニュー別に 3 つのグループに分割した．それぞれのグループについて，トレーニング前後の注目すべき成績の変化項目をまとめたものを表 6.4 に示す．ここで#W は勝率[10]，F_{TS} は 1 分あたりの T-spin 回数，#Sol は実力測定用の詰め T-spin 問題の正解率を表す．

表 6.4: 各グループのトレーニングによる成績の変化

	A (対戦のみ)			B (ランダムな二手問題)			C (面白い二手問題)		
	#W	F_{TS}	#Sol	#W	F_{TS}	#Sol	#W	F_{TS}	#Sol
訓練前	0.200	0.21	0.787	0.121	0.24	0.827	0.156	0.18	0.813
訓練後	0.394	0.13	0.800	0.306	0.75	0.893	0.242	0.79	0.813

勝率について，全てのグループの勝率は向上した．各グループの勝率の成長率をまとめたものを図 6.6 に示す．各グループの勝率は，グループ A は 20.0±13.3 %から 39.4±16.7 %に，グループ B は 12.1±11.1 %から 30.6±15.0 %に，そしてグループ C は 15.6±12.6 %から 24.2±14.6 %に増加した．グループ C の結果は統計的に有意ではなかったが，どのグループでも 120 分のトレーニングで対戦テトリスにおけるプレイヤの強さを向上できることが示せた．

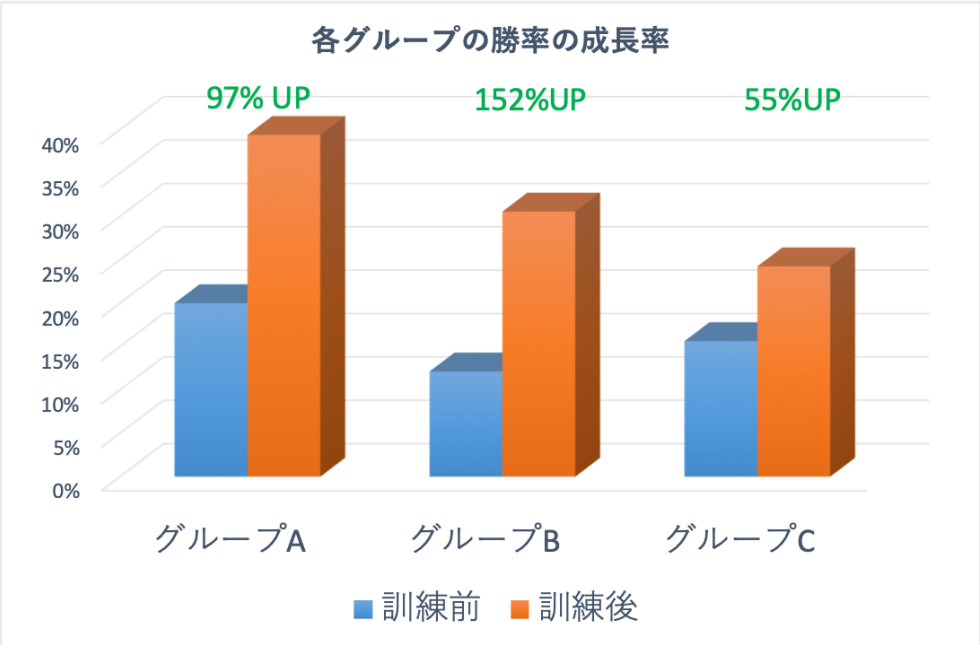


図 6.6: 各グループの勝率の成長率の比較

[10] 勝率 $= \frac{\text{勝ち数のグループ合計}}{\text{セット数のグループ合計}}$

詰め T-spin 問題の正解率については，グループ A が 78.7±9.3 %から 80.0±9.1 %に，グループ B が 82.7±8.6 %から 89.3±7.0 %に，そしてグループ C が 81.3±8.8 %から 81.3±8.8 %に変化した．トレーニングメニューに詰め T-spin 問題を活用したグループ B，C でも有意に向上されなかった．この結果はは 6.2.1 節の考察の通り，より工夫を凝らすことで改善されると考える．

一方で，プレイヤが対戦の中で T-spin による攻撃を行った頻度に注目すると，各グループの成長に明確な差を確認できた．被験者プレイヤの 1 分あたりの T-spin の頻度の成長率をグループ別にまとめたものを図 6.7 に示す．

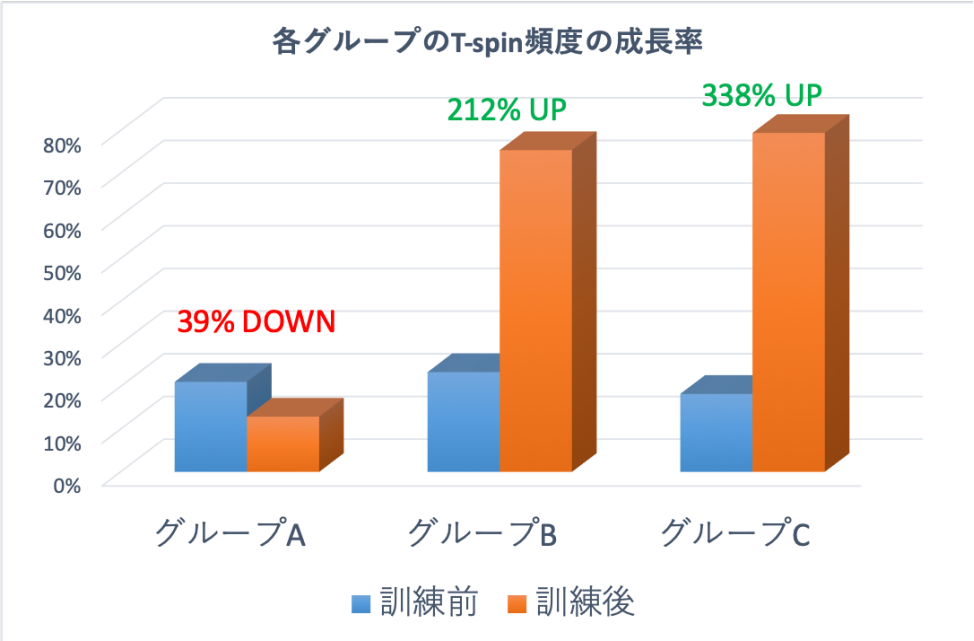


図 6.7: 1 分当たりの T-spin 頻度の向上（個別）

グループ A は訓練前と比較して T-spin の頻度が 39 %減少したのに対してグループ B，C はそれぞれ 212 %，334 %増加する大きな成長が見られた．このことから，詰め T-spin 問題を活用した訓練を行うことで，初心者プレイヤの実戦における T-spin 構成力を通常の練習方法よりも効率的に向上させられることが示せた．また，ランダムに生成した問題よりも面白い問題のみを解く練習メニューの方が T-spin を意識して戦うようになる可能性も示唆できた．

被験者プレイヤ 15 人の個別の T-spin 頻度の変化について，横軸：訓練前の 1 分当たりの T-spin 頻度，縦軸：訓練後の 1 分当たりの T-spin 頻度としてプロットしたものを図 6.8 に示す．一般に，グループ B と C のプレイヤは T-spin 構成力が向上しており，特にトレーニング前に T-spin をあまり使用しなかったプレイヤの成長は顕著に現れた．グループ A のプレイヤからはトレーニング後に T-spin 頻度が減る傾向がみられた．これについてグループ A の被験者に理由を尋ねたところ，120 分間の練習の中で，T-spin を意識するとミスが出て負けてしまったため平積み戦略を選択したとのことだった．

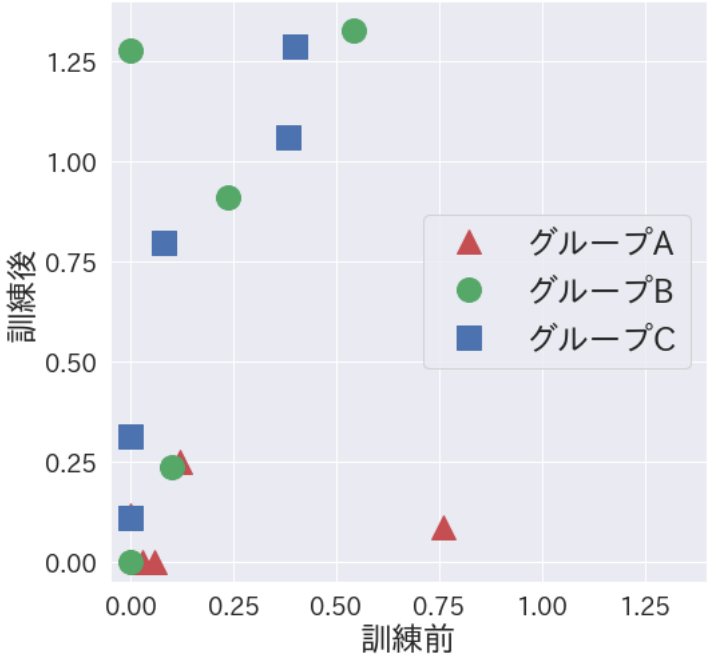


図 6.8: 1 分当たりの T-spin 頻度の向上（個別）

T-spin 頻度が高くなったプレイヤがそうでないプレイヤよりも勝率の向上しなかった理由を調べるために，グループ B，C のプレイヤと CPU の対戦動画を上級者がレビューしたところ，グループ B，C のプレイヤは T-spin のパターンを作ろうとしすぎており，テトロミノを配置する速度が低下していたことが分かった．また，T-spin を構築するための慣れない置き方に配置場所を間違えるミスプレイも少なくなかった．すなわち，T-spin から得られたメリットはプレイ速度の低下とミスによって相殺されてしまっていた．しかしながらこの結果は悲観的なものではなく，選手たちが T-spin の構成パターンに慣れるためにより長い時間トレーニングを受けることで，勝率の向上も十分に見込める．

以上の結果より，詰め T-spin 問題を活用したトレーニングメニューは通常の練習方法と比較して，テトリスにおける初心者プレイヤの上達を支援する上で有用であることが示せた．

第7章　スマートフォンアプリ

前節までの結果から，詰め T-spin 問題が初心者プレイヤの T-spin 構成力向上を効率的に支援できることが示せた．我々はこの結果を社会に還元するために，またより多くのデータを集めて個人の趣味・嗜好に合わせた面白さ・難しさ推定モデルの生成を実現するために，手軽に大量の詰め T-spin 問題を解けるスマートフォンアプリを作成した．それに伴い 4 章の方法を応用することで三手問題と四手問題を生成した他，上級者の間で多用されている定石形の一つを学ぶための問題生成にも取り組んだ．7.1 節で定石形問題の生成手順について説明し，7.2 節で問題の提供形式と収集するデータについて詳述する．

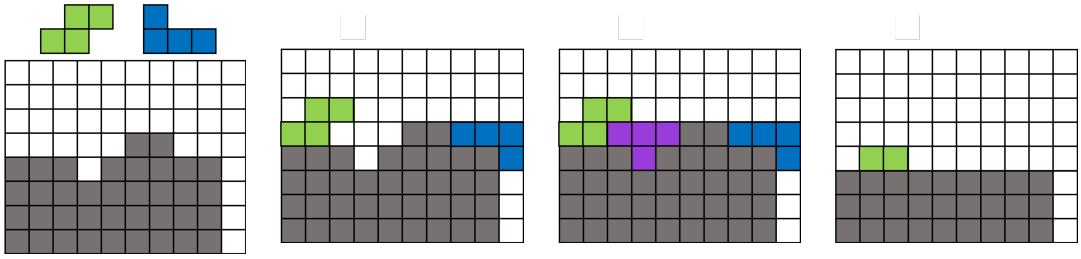
7.1　より難しい問題の生成

対戦テトリスの試合において，上級者プレイヤは絶え間なく T-spin による攻撃の応酬を繰り返している．これは上級者プレイヤが T-spin 実施のための典型的なパターンを定石形として暗記し，反射的に使用できるようにまで練習しているからである．このような定石形は数十種類存在するが，それらの中でも試合の中で頻繁に見られるテクニックである「ドネイト」に注目し，これを重点的に訓練できる詰め問題の自動生成を実現した．7.1.1 節でドネイトの定義について述べ，7.1.2 節でドネイト問題の自動生成法について述べる．

7.1.1　ドネイトとは

今回扱うような特定の構築パターンについて厳格な定義がされることは少ない．そこで本研究では 4000 人を越える関係者を擁するテトリスコミュニティの HARD DROP の解説ページ [24] を参考に，ドネイトを次項のように定義し，その概略を図 7.1 に示す．

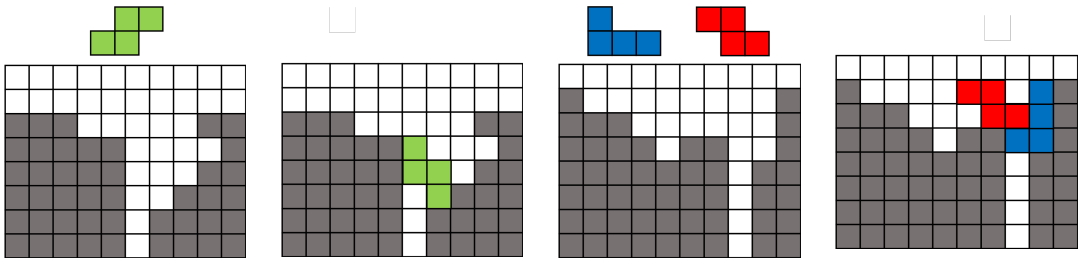
- ドネイトとは高さ 2 以上の縦穴を一度埋めてから T-spin へと繋げるテクニックのことであり，埋めた縦穴が T-spin による削除後に再び露出するようなものを指す．



(a) 右端に縦穴のある地形　(b) 穴を塞いで T-spin を可能な地形を構築　(c) T-spin によって 2 行削除　(d) (b) で塞いだ穴が再び露出する

図 7.1: ドネイトの手順

ドネイトとなり得る構成パターンの中から 2 つを図 7.2 に例として示す．図 7.2(a) は縦穴を含んだ階段のような地形であり，図 7.2(b) に示すように縦穴を塞ぐように S テトロミノを配置すると T-spin 可能な地形となる．このとき，完成した地形で T-spin を行うと塞いだ縦穴が露出する．図 7.2(c) の地形においても，図 7.2(d) に示すように J テトロミノと Z テトロミノを配置することで穴を塞ぎながら T-spin 可能な地形を構築することができる．図 7.2(d) の地形で T-spin を行うと，すぐに I テトロミノで 4 行消しが狙える地形が残る．このようにドネイトはその性質から平積みとの相性がよく，T-spin を覚え始めた初心者が実戦で勝ちを目指すためのステップアップのためにも有用であると考えた．次節ではドネイトを練習するための詰め問題を生成する手順について説明する．



(a)　(b)　(c)　(d)

図 7.2: ドネイトの例

7.1.2　ドネイト問題の生成法

プレイヤにドネイトを練習させるための詰め問題の生成は，図 7.3 に示すように通常の詰め T-spin 問題の生成法を応用することで簡単に実現できる．まずはじめに，図 7.3(a) に示すような「2〜4 の高さを持つ縦穴」がただ一つだけ存在するような地形を生成する．このとき縦穴の高さに関わらず，縦穴の一番上のマスから 3 つ上のマスに印を付けておく．続いて図 7.3(b), 図 7.3(b), 図 7.3(d) のように，生成した地形の上 3 行に対して 4.1 節の図 4.1 の方法で縦穴の位置に関係なくベースの地形を生成する．最後に図 7.3(c), 図 7.3(e), 図 7.3(g) に示すようにベースの地形の一番上の行に数個のノイズをランダムに付与するが，図 7.3(a) で印が付けられたマスは埋めないようにする．すなわち図 7.3(d) や図 7.3(e) といった地形は認めず，図 7.3(c)，図 7.3(g) のような地形のみを扱う．

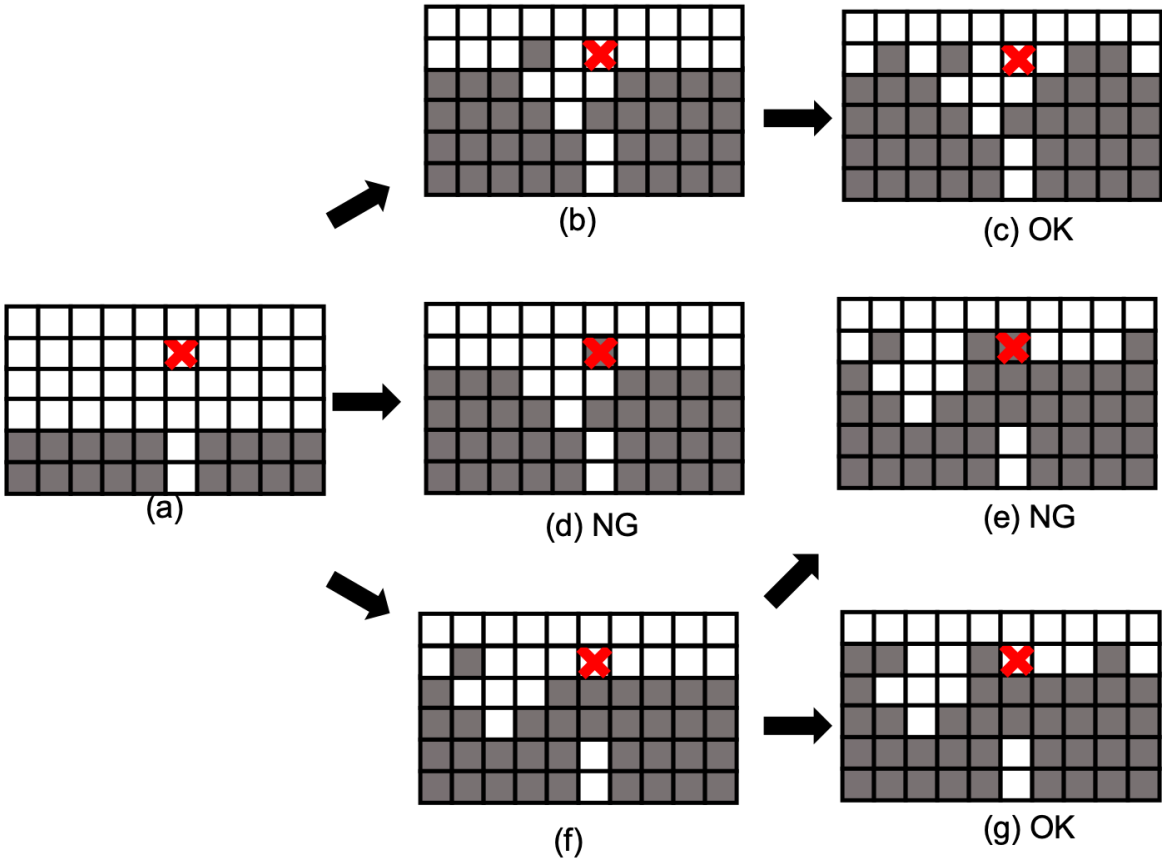


図 7.3: ドネイト問題の完成形の生成手順例

続いて，生成した図 7.3(g) の完成形からテトロミノを抜き取ることで二手詰めのドネイト問題を生成する手順の例を図 7.4 に示す．基本的な流れは 4.2 節の図 4.3 の方法と変わらないが，ドネイト問題の場合は図 7.4(b)，図 7.4(c)，図 7.4(d)，図 7.4(e) に示すようにテトロミノを抜き取ったマスに印を付けておき，n 手分 抜き取った後に縦穴の一つ上のマスに抜き取られた印がついているもののみを選ぶ点が異なっている．例えば図 7.4(e) に示すような，二手分のテトロミノを抜き取ったあとの縦穴の真上に印がない問題は，二手詰めの T-spin 問題にはなっていたとしても着手で縦穴を塞ぐことをしないためドネイト問題としては扱えない．

以上より，ドネイトを重点的に練習するための詰め T-spin 問題の生成を実現した．次節ではアプリケーション上で自動生成した問題をどのような形式でプレイヤに提供するかについて述べる．

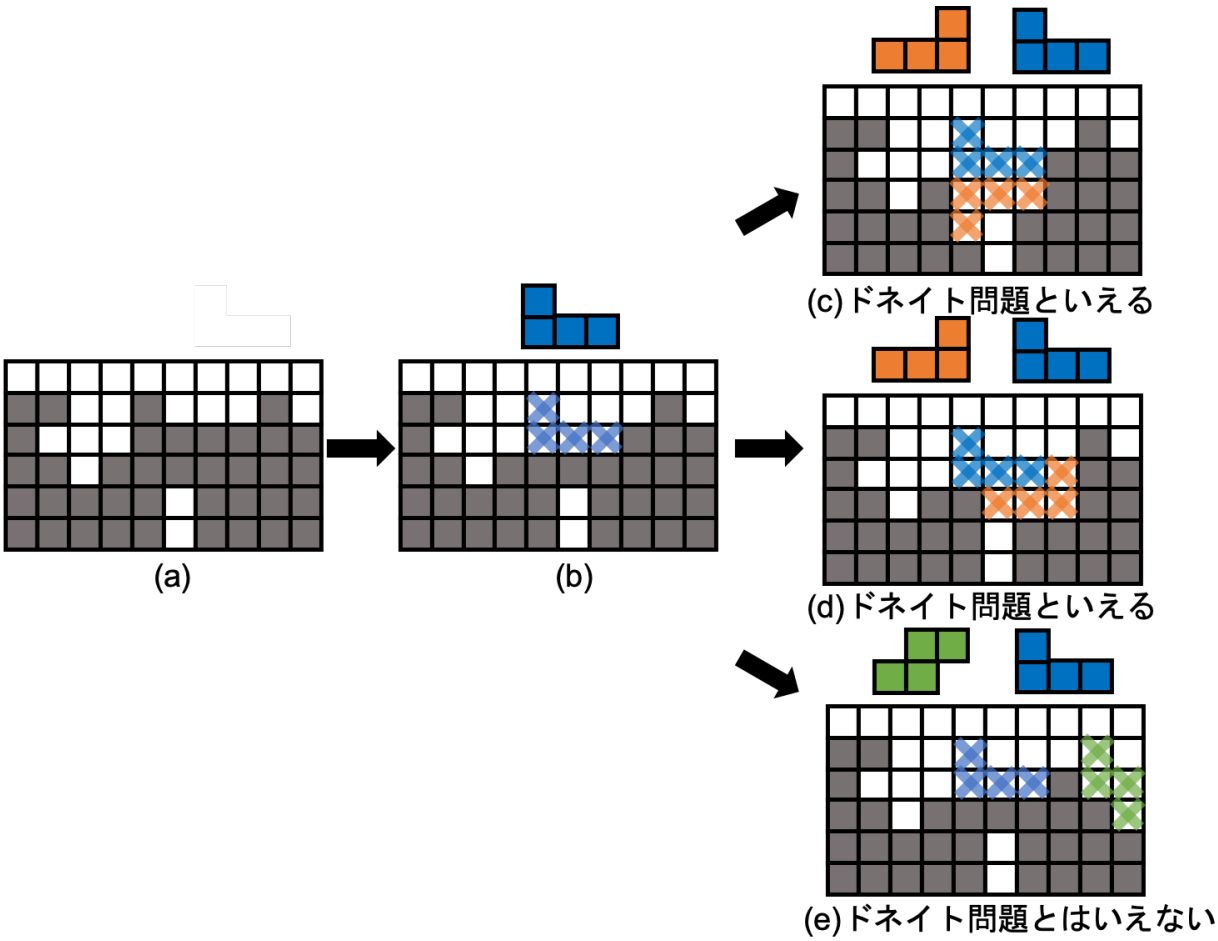


図 7.4: ドネイト問題を生成する逆向き生成法の手順例

7.2 提供形式

本アプリはプレイヤの T-spin 構成力向上を支援するために，大きく分けて 3 つのモードを搭載した．7.2.1 節から 7.2.3 節にかけて各モードの概略と収集するデータについて説明する．

7.2.1 厳選問題集

6.2.2 節の結果より，面白い詰め T-spin 問題を活用したトレーニングは初心者プレイヤの T-spin の構成力を効率的に向上させられることがわかった．そこで我々は厳選された面白い問題を一定数遊べる “厳選問題集” モードを考案した．本モード全体の流れを図 7.5 に示し，それぞれについて実際のスクリーンショットを用いながら説明する．

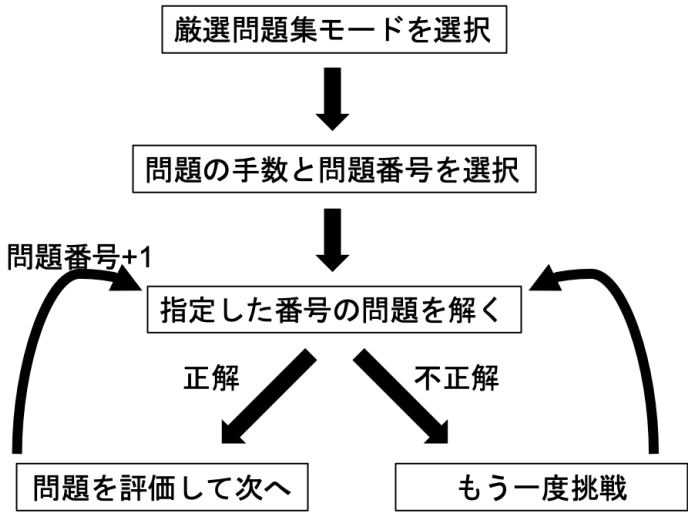


図 7.5: 厳選問題集モードを遊ぶ流れ

本モードではプレイヤが「解きたい問題の手数」と「問題番号」を図 7.6 の画面で選択することによって，指定した問題を解くことができる．提供する問題は一手から四手詰めまでの 4 種類で，それぞれ 50 問[11]ずつ用意した．このときプレイヤの意欲を高められるような良い問題のみ提供するために，一手・二手問題については 5 章の面白さ推定モデルによる推定値が 3.4 よりも大きい問題を選別した．三手・四手問題についてはどのような特徴量を持つ問題が難しいか不明であったため，二手問題の面白さ推定モデルで寄与度が高かった特徴量から面白くなると想定されるようにフィルターを設定し，選別を行った．

[11]厳選された三手問題の例を Appendix に示す．

図 7.6: 問題の選択画面

挑戦する問題を選択すると，次項の図 7.7 の画面で詰め T-spin 問題を実際に解ける．基本的な操作方法と合わせて次項に示す．なお，各操作ボタンは少しでも実戦の感覚と近づけるために一般的なゲームコントローラを模した配置としている．

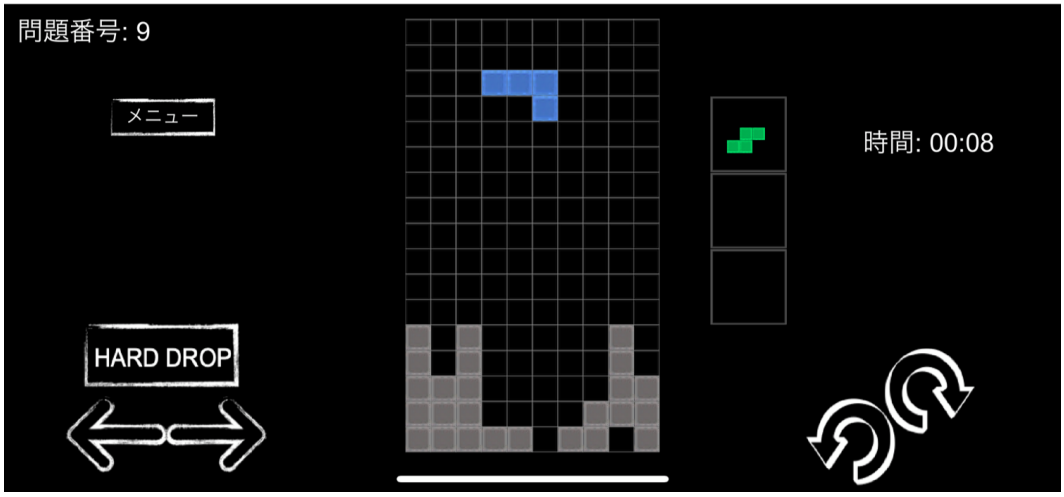


図 7.7: 詰め T-spin 問題を解く画面

- 左右移動：画面左下に配置されている左・右矢印をタップするか，画面を左右にスワイプすることで，着手テトロミノを移動させることができる．スワイプの場合は位置の変化量に応じたマス数分だけ移動する．
- 左右回転：画面右下に配置されている回転ボタンをタップすることによって，ボタンが表す方向に着手テトロミノを回転させることができる．
- 配置：HARD DROP ボタンをタップすることによって着手テトロミノを配置することができる．

厳選問題集モードではプレイヤに問題の答えを開示することはしない．プレイヤが n 手の配置を完了すると，プログラムが正解かどうかを判定し結果を表示する．もし不正解であれば,「もう一度」ボタンが表示され，何度でも同じ問題に挑戦することができる．もし正解であれば，図 7.8 の画面に遷移し，次の問題へ進むか他の問題やモード選択画面へ移動するかを選べる．このとき，次の問題へ進めるボタンは「よかった，次へ」と「ふーん，次へ」の 2 種類を設けており，可能な限りプレイヤのモチベーションを阻害しない形で，プレイヤが問題を面白いと感じるかどうかを 2 値でデータとして収集する．また，問題の一般的な難しさを推定するための指標として「問題を正解できたか」「その問題を初めて正解するまでにかかった時間」などのプレイヤ情報も記録しており，これらは次の問題へ進まなかった場合にも記録する．将来的にはこれらを処理して各問題に対する全ユーザの平均解答時間や正解率を表示することで，プレイヤのモチベーションを向上に活用できると考えている．

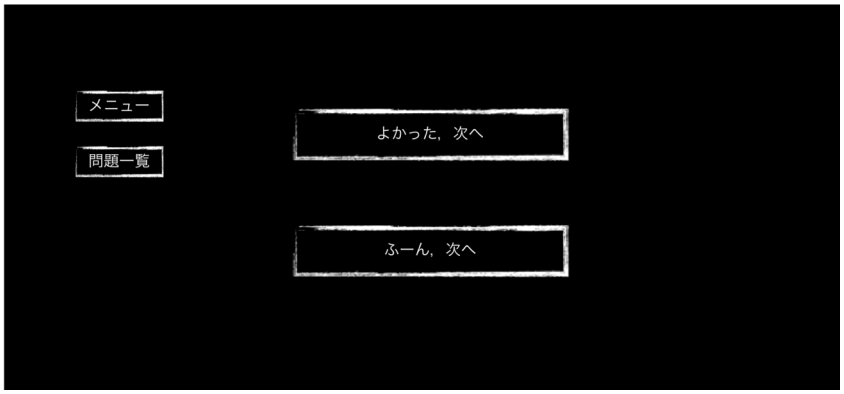


図 7.8: 「よかった，次へ」ボタンと「ふーん，次へ」ボタン

7.2.2　フラッシュモード

実際の対戦では T-spin を構築できるか否かを瞬時に判断できる能力が求められる．その力を詰め T-spin 問題によって養うために，我々は “フラッシュモード” を考案した．本モードの簡単な遊び方を図 7.9 に示す．

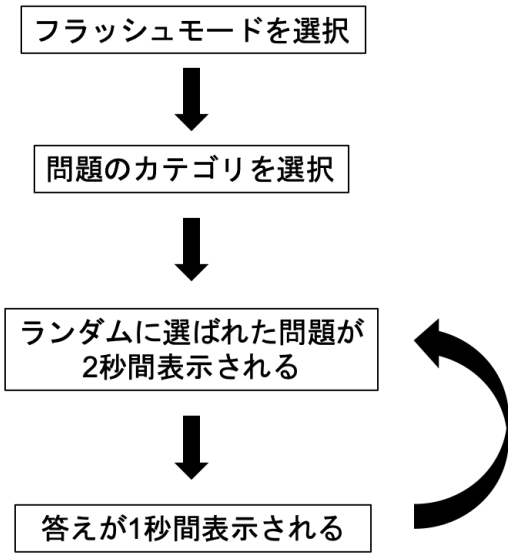
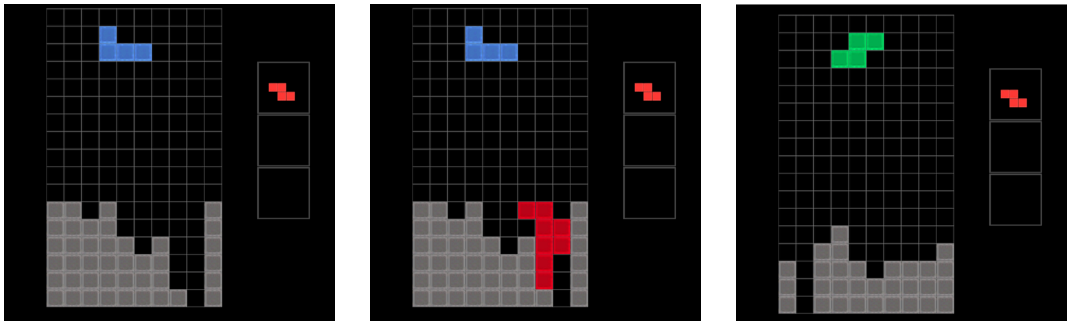


図 7.9: フラッシュモードを遊ぶ流れ

本モードでは他のモードのように実際に操作することはなく，図 7.10(a) のように 2 秒間問題を表示した後に図 7.10(b) のように 1 秒間解答例を示す．また，その後は同じカテゴリ内からランダムに選ばれた問題が図 7.10(c) のように表示され，以後これが繰り返される．



(a) 問題を 2 秒間表示　(b) 解答を 1 秒間表示　(c) 次の問題を 2 秒間表示

図 7.10: フラッシュモードの表示形式

問題のカテゴリは図 7.11 に示すように，一手問題から四手問題までの 4 種類に加え，二手から四手までのドネイト問題も選択可能とした．それぞれの問題はランダムに生成された物を 500 問ずつ用意した．

図 7.11: 問題カテゴリの選択画面

本モードではテンポよく思考トレーニングが行えることに加え，プレイヤに「どの程度の手数の問題ならば反射的に解けるのか」を自覚させることができ，プレイヤが自分自身の実力に適した難易度の問題を探す上で役立てられると考えた．ここで,「プレイヤが何問のフラッシュ問題を見たか」といったデータは，将来的に練習量と成長度を比較する際に活用可能であると考え，記録した．

7.2.3　練習モード

6.2.2 節の結果では面白い問題を解いたグループだけでなく，ランダムに生成された問題を解いたグループにも成長が見られた．そこで我々はランダムに生成された問題を大量に解ける “練習モード” を実装した．このモードの概略を図 7.12 に示す．

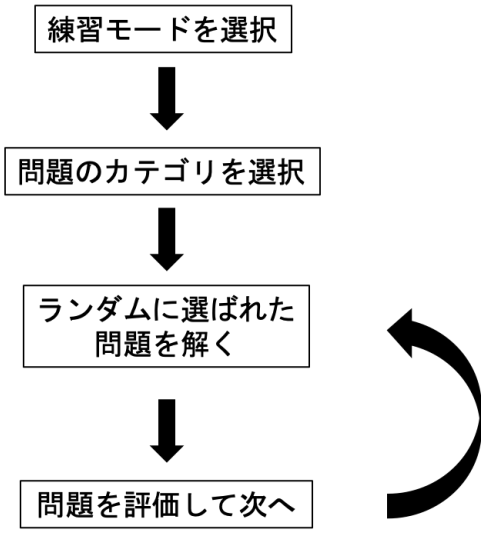


図 7.12: 練習モードを遊ぶ流れ

本モードは図 7.11 と同様の問題カテゴリを選択し，ランダムに選ばれた問題を 7.2.1 節の厳選問題集と同じ操作方法で解くモードである．ただし，練習モードではプレイ後に解答例を表示できる点が異なる．本モードでも n 手の配置が終了した時点で問題に対する 7.2.1 節と同じプレイヤデータを収集している．また,「プレイヤが練習問題を何問解いたか」というデータは,「フラッシュ問題を見た数」と同様にプレイヤの成長度を比較する際に活用できる可能性があると考えたため記録した．

これらの機能を機能を搭載したアプリは開発が既に完了しており，近日中に公開予定である．

第8章　おわりに

本研究では「ゲームにおける “難解かつ上達していく上で重要なテクニック” を人間プレイヤに効率的に習得させるにはどうしたら良いか」を解明するべく，まずはテトリスにおいて T-spin 技術を習得させるための教育システムを構築するために，T-spin の完成図を作成しテトロミノを抜き取ることで詰め問題を作成する手法を提案した．さらに，問題地形と正解着手から得られる特徴量から初心者プレイヤが感じる面白さや難しさを推定する手法を提案し，単純な問題においてはある程度正確に面白い問題のみ選別できることを示した．また上級者一人を対象とした推定モデルの生成も実現したことから，個人に合わせたトレーニングシステムを構築できる可能性が示せた．

さらに，3 つの異なるメニューで訓練したプレイヤを比較する実験も行った．この実験により，詰め T-spin 問題を活用した訓練がプレイヤの T-spin 頻度を効率的に向上させられることを示した一方で，勝率については通常の練習を行ったグループよりも成長が小さかった．通常の練習方法で訓練を行ったグループは，T-spin の重要性を理解しているのにも関わらず練習後の T-spin 頻度が伸び悩んだ．通常の練習方法で訓練したグループのプレイヤに「なぜ T-spin をあまりしなかったのか」尋ねたところ,「自分なりに T-spin の練習には挑戦してみたものの，ミスがでやすくなったため平積みの方が楽に勝てると考えたから」という回答を得た．これらから，多くの初心者プレイヤが T-spin に挫折する理由は T-spin による即時的な恩恵が得られなかったからであると考えた．このような「すぐに結果につながらない訓練」は，スポーツにおける地味な基礎トレーニングのように，その重要性や効果が高いことがわかっていても，自主的かつ継続的に取り組むことは難しい．本研究はテトリスにおける T-spin という苦労を要するが上達のために大切な第一歩を，詰め T-spin というパズル形式の問題として人間プレイヤに提供することで補助できることを示した．

今後の展望として，一手，二手の詰め T-spin 問題に関して面白さの推定が実現したため，三手問題や四手問題といったより複雑で実戦に近い問題を対象とした推定モデルの構築を目指す．初心者プレイヤが「難しすぎて面白くない」と感じ始める手数や問題の特徴を明らかとすることも興味深い．さらに，4 行消しや T-spin Triple といった技術も実戦では重要であるため，これらの技術に特化したトレーニングシステムも今回の方法と同様に構築することができる．あるいは，異なる技術を組み合わせて実戦力を総合的に向上させられるシステムの構築にも取り組みたい．

謝辞

本研究を進めるにあたり，未熟な学生だった私を懇切丁寧で多大なる御指導，御助言を通して自分の研究に誇りが持てる研究者へと成長させてくださった主指導教員の池田心准教授に深謝いたします．また副指導教員である飯田弘之教授，国際会議の論文執筆に多大なる協力をしてくださったHsueh Chu-Hsuan助教，本学に入学するきっかけを与えてくださった上原隆平教授，そして共にゲームの面白さを探究してきた池田研究室のメンバーに心より感謝いたします．最後に，これまで温かく見守ってくれた家族に感謝します．

参考文献

[1] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy P. Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, Vol. 352, No. 6419, pp. 1140–1144, 2018.

[2] Steven Kapturowski, Georg Ostrovski, John Quan, Rèmi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *The Seventh International Conference on Learning Representations (ICLR 2019)*, 2019.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, No. 7540, pp. 529–533, 2015.

[4] Robin Hunicke and Vernell Chapman. AI for dynamic difficulty adjustment in games. In *AAAI-04 workshop on Challenges in Game Artificial Intelligence*, pp. 91–96. AAAI Press, 2004.

[5] Kokolo Ikeda and Simon Viennot. Production of various strategies and position control for Monte-Carlo Go - entertaining human players. In *2013 IEEE Conference on Computational Inteligence in Games (CIG 2013)*, pp. 145–152. IEEE, 2013.

[6] 高橋竜太郎, 池田心. 連鎖力向上のためのぷよぷよの問題作成. 研究報告ゲーム情報学（GI）, Vol. 39, No. 10, pp. 1–7, 2018.

[7] 2019 年 3 月期決算説明会. [`https://www.nintendo.co.jp/ir/pdf/2019/190426_3.pdf`]. pp. 12, 2019,（アクセス：2020/02/05）.

[8] Yubin Liang, 池田心. リズムゲームの上達を支援するコンテンツ自動生成法. 研究報告ゲーム情報学（GI）, Vol. 39, No. 11, pp. 1–7, 2018.

[9] 及川大志, 池田心. テトリスにおける t-spin 構成力向上のための問題作成. ゲームプログラミングワークショップ 2018 論文集, pp. 175–182, 2018.

[10] Oikawa Taishi, Ikeda Kokolo, and Chu-Hsuan Hsueh. Improving human players' t-spin skills in tetris with procedural problem generation. In *The 16th International Conference on Advances in Computer Games*, 2019.

[11] Tetris. [`https://tetris.com`]. （アクセス：2020/02/05）.

[12] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, 2016.

[13] TIMEINTERMEDIA - パズル自動生成エンジン. [`https://www.timedia.co.jp/service/ai/puzzle-engine/`]. アクセス：2020/02/05）.

[14] パソコン初心者の館. [`http://www.pro.or.jp/~fuji/`]. アクセス：2020/02/05）.

[15] やじうま PCwatch - 高品質なナンプレ問題を自動生成する人工知能システム. [`https://pc.watch.impress.co.jp/docs/2006/0906/yajiuma.htm`]. アクセス：2020/02/05）.

[16] Timo Mantere and Janne Koljonen. Solving, rating and generating Sudoku puzzles with GA. In *2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, pp. 1382–1389. IEEE, 2007.

[17] Masayuki Hirose, Takumi Ito, and Hitoshi Matsubara. 逆算法による詰め将棋の自動創作. 人工知能学会誌, Vol. 13, No. 3, pp. 452–460, 1998.

[18] 山崎隆介, Grimbergen Reijer. 連鎖型パズルゲームにおけるパズル問題の自動創作. ゲームプログラミングワークショップ 2013 論文集, pp. 118–121, nov 2013.

[19] 大町洋, 佐藤直之, 池田心. 複数ソルバを用いた上海ゲームのインスタンス生成. ゲームプログラミングワークショップ 2013 論文集, pp. 126–129, nov 2013.

[20] 石飛太一. Deep　証明数探索と詰将棋の美観評価. PhD thesis, 北陸先端科学技術大学院大学, 2016.

[21] LightGBM. [`https://www.microsoft.com/en-us/research/project/lightgbm/`]. バージョン: 2.3.1,（アクセス：2020/02/05）.

[22] 第 10 回日本 e スポーツ学生選手権　ぷよぷよテトリス部門. [`https://jeslpuyoteto.wixsite.com/jesl-puyoteto`]. （アクセス：2020/02/05）.

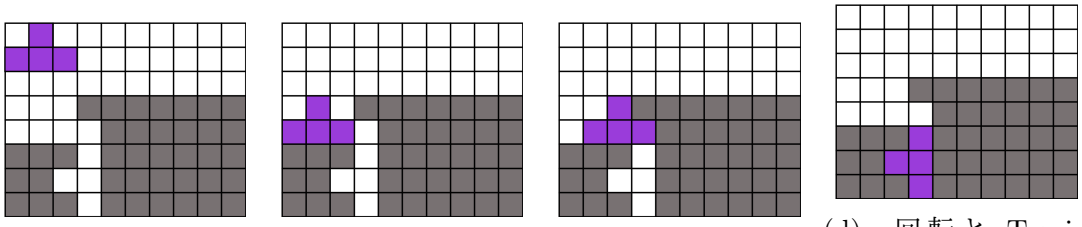
[23] Puyo Puyo™Tetris®. [https://store.steampowered.com/app/546050/Puyo_PuyoTetris/]. （アクセス：2020/02/05）.

[24] HARD DROP - Donating. [https://harddrop.com/wiki/Donating]. （アクセス：2020/02/05）.

Appendix

A　T-spin Triple

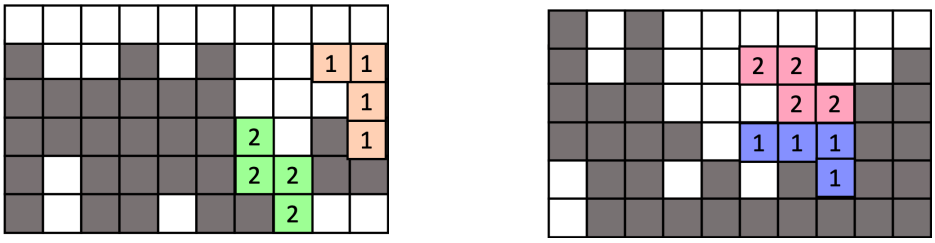
T-spin Triple の手順を図 A.1 に示す．図 A.1(a) の状態から T テトロミノを落下させ接地させると図 A.1(b) のような盤面になる．接地してすぐに，右に 1 マス移動させて図 A.1(c) の盤面とし，そこから反時計回りに 90 度回転させることで図 A.1(d) のように T テトロミノを無理やりねじ込むことができる．特に図 A.1(c) から図 A.1(d) の遷移は特殊であり，テトロミノの回転法則を理解する必要があるため初心者にとって難しすぎると判断し，本研究では扱わなかった．



(a) 初期状態　(b) 接地　(c) 右に 1 マス移動　(d) 回転と T-spin Triple

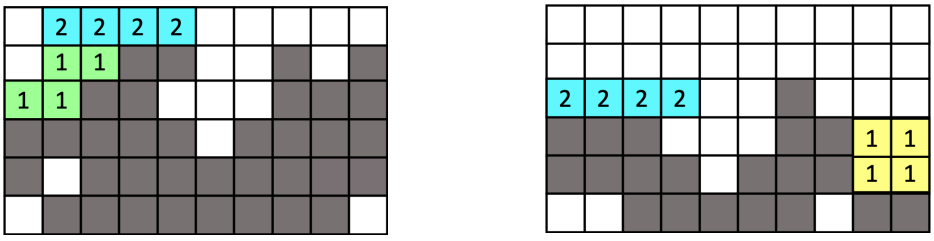
図 A.1: T-spin Triple の手順

B　図 5.13，5.14 の二手問題の解答



(a)　(b)

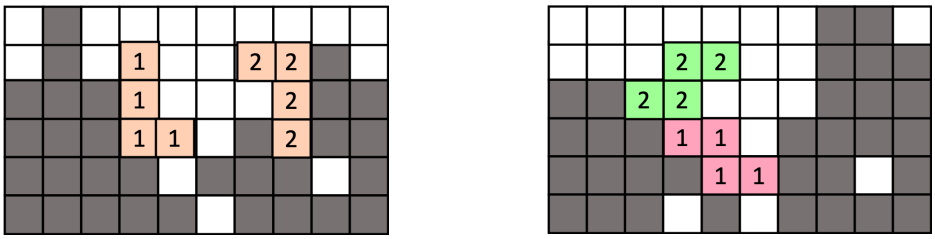
図 B.2: 初心者モデルの面白さの推定値が高い二手問題の解答



(a)　(b)

図 B.3: 初心者モデルの面白さの推定値が低い二手問題の解答

C　図 5.16 の二手問題の解答

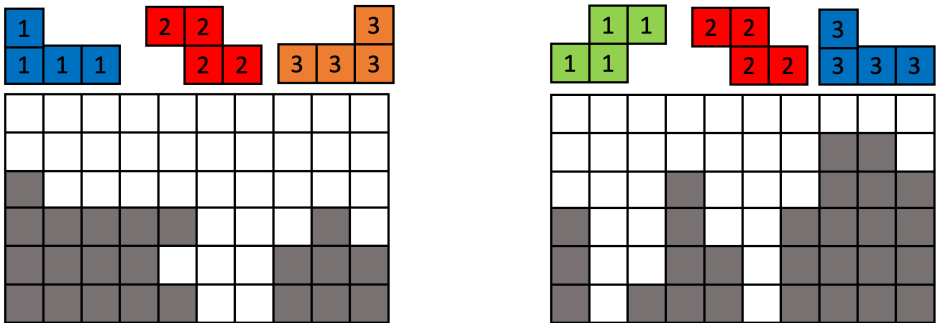


(a)　(b)

図 C.4: 上級者モデルの面白さの推定値が高い二手問題の解答

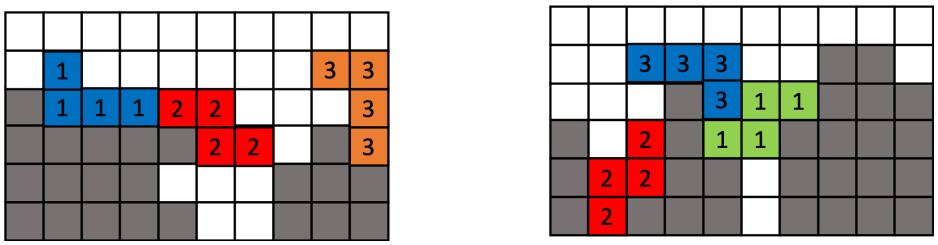
D　厳選問題集の問題例

厳選問題集の三手問題の中でも筆者が特に難しいと感じた 2 問を以下に示し，解答は次項に掲載する．



(a)　　(b)

図 D.5: 厳選された三手問題



(a) (b)

図 D.6: 厳選された三手問題の解答