

ぷよぷよの連鎖構築 AI に関する研究

柴澤弘樹

2017 年 2 月 3 日

第 I 部

序論

1 背景

1.1 ゲームを取り巻く状況

我々人類は、遊びと共に生きている。ホイジンガは「ホモ・ルーデンス」の中で、遊びが文化よりも古いものであると述べた [7]。現在遊ばれているコンピュータゲームもまた、遊びの一種といえる。藤田による調査 [26, 28, 27] の中で、コンピュータゲームの一般への普及は、1980 年代からであったことが述べられている。

現在では、スマートフォンの普及と共にゲームプレイヤーがさらに増えている。ファミ通ゲーム白書 2016[21] により、アプリゲームを含むゲーム市場の拡大が報告された。さらには、e スポーツとして、ゲームでプレイヤー同士が競う大会も多く開催されてきた。大会やプロプレイヤーにはスポンサーがつき、世界規模の市場となっている [12]。これらの市場は、今後もますますの成長が期待される。

このような背景の中で、遊びとしてのゲームはより身近なものになっている。様々なハードウェアでインターネットを介した協力、対戦プレイが可能となっており、遠く離れたプレイヤー同士で同じゲームをプレイすることも当然のこととなった。それによって、遊び相手として人間が一般的になり、AI を相手に長時間遊ぶゲームは限られたものとなってきている。ゲーム AI 技術は著しい発展を遂げてきたものの、遊び相手としては未だ向上の余地があるといえる。

インターネットを介した対人プレイにおいては、以下のような問題が挙げられる。

- 対応した環境の必要性
- 十分なプレイ人口の必要性
- プレイスキルを考慮したマッチング
- 気兼ねない相手とはならない
- 対人への抵抗感

以下、簡単に説明する。環境は、ゲームのハード、ソフト両面で整っている必要がある。インターネット環境の不足や、対応していないゲームではオンラインプレイが不可能となる。プレイ人口は、時間や発売からの経過日数などの影響を受ける。十分にプレイヤーが揃わなければ、マッチングができない。それに付随し、プレイスキルのバランスを考慮したマッチメイキングも必要となる。また相手が人間であるため、他プレイヤーを無視した勝手なふるまいが全くできない問題もある。ゲームを楽しんだり練習したりするため、あえて無駄な動きや非効率な行為をすることができないのである。心情的に、他人とゲームをプレイすることを忌避するプレイヤーも一定数いると考えられる。

以上の問題の通り、遊び相手としての人間が、必ずしも適しているとはいえない場面が存在する。このとき、ゲーム AI は一つの解決策となる。そのため、プレイヤーに楽しさを提供する手法を研究し、遊び相手としての AI を実現することに一定の意義があると考えられる。

1.2 ゲーム AI の現状

これまでのゲーム AI の研究は、強い AI を目指したものだった。IEEE CIG をはじめとする AI コンペティション [16] や、人間との戦いによって強さが競われてきた。特に盤上ゲームでは、1997 年の Deep Blue、2013 の ponanza、そして 2015 年の AlphaGo [15] が、それぞれプロ棋士に初勝利を収めた。いまや AI は、人間と同等以上の強さをもつようになったと言える。つまり、AI の強さにおける研究は、一定の成果が得られたと言われるようになってきている。

強い AI を求めるこれまでの研究は、探索アルゴリズムの研究であったという見方ができる。完全情報ゲームの着手に評価を与え、最適解を探索する手法が様々提案されてきた。着手に対しどのような評価を与え、計算資源の中でいかに効率よく探索を行うかが問題だったのである。AlphaGo では、ディープニューラルネットワークによる評価と、モンテカルロ木探索によってそれらの問題を解決した。これらの手法は、AI の挙動がブラックボックスとなり、全容の理解が困難となる問題がある。その一方で汎用性が高く、様々

に応用されている [13, 3]。完全情報ゲームに関しては十分な成果が得られたことから、今後の強さに関する研究は、不完全情報ゲームや General Game Playing(GGP) へ移行してゆくと思われる。

以上で述べた通り、これまでのゲーム AI は、強さや性能を評価する場として、ゲームを用いていたにすぎない。人間プレイヤーのゲーム体験を向上させる域には達していないのである。しかし、強さが十分な領域に達したことで、今後はユーザエクスペリエンスを考慮した AI の研究が活発化してゆくといわれている [22, 25]。Yannakakis はゲーム AI 研究の分野を以下の 10 個に分類した [19]。

1. Non-player character(NPC) behavior learning
2. Search and planning
3. Player modeling
4. Games as AI benchmarks
5. Procedural content generation
6. Computational narrative
7. Believable agents
8. AI-assisted game design
9. Gneral game AI
10. AI in commercial games

競争としてのみではなく、デザイナーやプレイヤーのことを考慮した AI 研究がなされていることが分かる。ゲームの遊びや楽しさという側面が、今後ますます注目されてゆくことが予想される。

2 目的

本研究では、人を楽しませる相手としての AI を構成する手法を提案することを、最終目的とする。特にパズルゲームの「ぷよぷよ」を対象とし、その対戦 AI を実装する。「ぷよぷよ」は、以下のような特徴をもつ。

- 長年にわたって親しまれているゲームである
- 大会や e-sports での対戦実績がある
- プレイヤースキルの幅が非常に広い
- 対戦用 AI の性能が不十分である

- 探索問題として難しい題材である

「ぷよぷよ」は長年にわたって、対戦ゲームとして親しまれてきた。それにも関わらず、対戦相手としての AI は未だ不十分である。また不完全情報ゲームであり、探索問題としても課題が残されている。以上の観点から、「ぷよぷよ」の AI が研究対象として意義をもつと考えられる。

対戦における「ぷよぷよ」のプレイでは、連鎖構築や戦術が重要となる。上級者のプレイでは、十分な威力の連鎖を保持しつつ、相手を妨害したり、相手の妨害を阻止したりする戦術が用いられている。現状の AI では、どちらも不十分である。この解決のためにはまず、より基本的な行為である、連鎖構築を満足に行えなければならない。対戦中での連鎖構築は、連鎖数、柔軟性、リアルタイム性の 3 点を満足することが大切であると考えられる。本研究では、この 3 点を満たす連鎖構築アルゴリズムを実現することを目的とする。

「ぷよぷよ」の連鎖構築 AI の実装にあたり、技術的課題は以下に挙げられる。

- 長期的視点に基づく探索
- 不完全情報と妨害による先読みの難しさ
- 完成形のパターン化の難しさ
- 計算量の膨大さとリアルタイムの制約

大きな連鎖数を確保するためには、長期的視点に基づいてぷよの配置を決定する必要がある。同じ一手であっても、すぐに連鎖の威力を補強できる手よりも、数手先で連鎖数を増やすための布石とする手の方が有効である。このために先読みが必要になるが、配される手がランダムであること、リアルタイムに相手からの妨害があることを考慮すると、その可能性が膨大になる。さらにはそのランダム性から、連鎖の完成形をあらかじめ想定しておくことが、大きな手間となる。配された手から完成形を組みかえるような、柔軟性ある連鎖構築が望ましいのはそのためである。最後に、ぷよは操作をしなくとも自由落下するため、配置の決定に時間的制約がある。以上のような課題を解決する手法の研究が、本稿の目的である。

3 本論文の構成

第Ⅱ部

ぷよぷよについて

1 発売タイトル

最初の「ぷよぷよ」は、コンパイルより 1991 年に発売された。当初の「ぷよぷよ」には相殺がなく、1994 年の「ぷよぷよ通」によってはじめて導入された。以降のシリーズの基本形はこの「ぷよぷよ通」とされ、AC（アーケード）版は現在でも大会で利用される。その後もシリーズは続いたものの売り上げは振るわず、コンパイルは経営破綻に陥った。

2003 年の「ぷよぷよフィーバー」以降はセガによってぷよぷよが発売されている。2004 年の Windows 版「ぷよぷよフィーバー」で、初の公式ネット対戦が可能となった。対戦ルールは「クラシックルール」（実質的な「通」ルール）と「フィーバールール」が搭載されていた。その後も家庭用ゲーム機や携帯ゲーム機に向けて複数のタイトルが発売されている。2014 年には「ぷよぷよ」と同じく落ちものパズルゲームの「テトリス」を組み合わせ、**「ぷよぷよテトリス」**が発売された。「テトリス」と「ぷよぷよ」の対戦が可能となるなど、異色の作品であった。「ぷよぷよ」のルールは、主に「ぷよぷよ通」に準拠したものとなっている。最新作は 2016 年発売の「ぷよぷよクロニクル」で、インターネット対戦では「スキルバトル」「ぷよぷよ通」「ぷよぷよフィーバー」の 3 種類のルールがプレイできる。

その他、「ぷよぷよ!!クエスト」や「なぞぷよ」などの派生作品が、スマートフォン向けアプリなどで発売されている。これらはグラフィックやキャラクター、4 つつながると消える点などは踏襲しているものの、対戦要素が無くシステムも異なるため、従来のぷよぷよシリーズとは別のゲームであるといえる。

シリーズ全体を通し、特に対戦において「ぷよぷよ通」が基準となっていることが読み取れる。これまでに様々な新ルール搭載のタイトルが発売されてきたものの、「通」のルールに置き換わるものは未だ現れていない。そこで本稿でも「ぷよぷよ通」のルールに準拠し、以下の「ぷよぷよ」に関する研究を進める。

2 ルール

本研究で扱う、「ぷよぷよ通」の基本的なルールと用語について説明する。

2.1 フィールドとぷよ

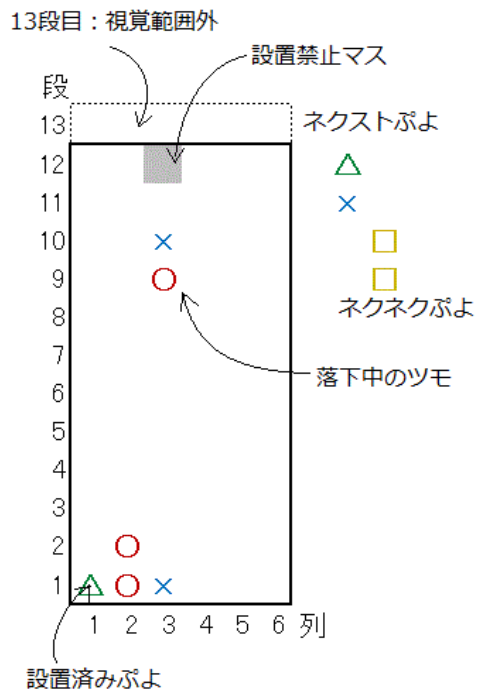


図1 ぷよぷよのゲーム画面の概略

ぷよぷよのゲーム画面の概略図を、図1に示す。ぷよぷよのフィールドは、横6列×縦12段が可視範囲である。以下、列数は左から数えたもの、段数は下から数えたものとする。配置するブロックは「ぷよ」と呼ばれる。3列目、12段目のマスにぷよを置くと窒息であり、負け（ばたんきゅ〜）となる。また、実際にはフィールドに13段目があり、12段目にぷよがある状態で「回し」という操作技術を用いて、視覚範囲外へぷよを設置できる。13段目のぷよは可視できない状態で繋がったり消えたりすることはない、12段目以下のぷよを消し、落下させることで連鎖に寄与する。

配石は2つのぷよがペアで現れ、これをツモという。ツモは落下中の物のほかに、次のツモ（ネクストぷよ）と次の次のツモ（ネクネクぷよ）が予め表示される。ぷよは複数の

色から成り、対戦では最大4色が標準である。すなわち、4色のぷよが2つずつ現れることから、ツモは全16通りが存在する。対戦におけるツモ順は各プレイヤーで同一である。ツモの配色はランダムであり、表示される手数に上限があることから、配石に関する情報の不完全性が認められる。

ツモは3列目の13段目と12段目に現れ、自然に落下してゆく。落下中のツモは横移動と4方向回転ができるため、すでに埋まっていない限りは任意の列、あるいはそれと隣り合った列に配置できる。ぷよの設置の際は、下段に空白を認めない。下に空洞がある場合、それより上のぷよは全て下に落下する。よって、段差がある隣り合った2列に対しツモを横向きに置いた場合、片方のぷよが落下する。これを「ちぎる」という（図2参照）。ツモの配置方法の選択は、異なる色の場合22通り、同色（ゾロ）の場合11通りとなる。

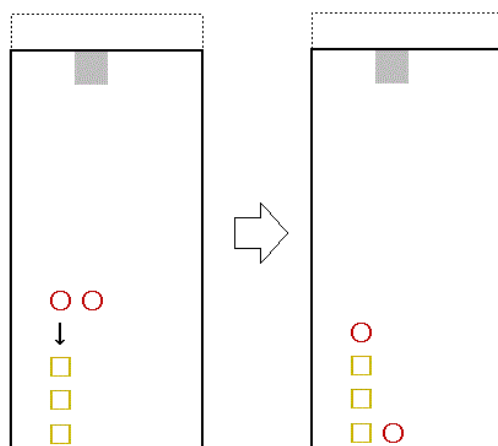


図2 「ちぎり」の発生の様子

2.2 ぷよの消去と連鎖

置かれた色ぷよは、上下左右に隣接したマスの同色ぷよと連結する。ここで色ぷよとは、後述の「おじゃまぷよ」ではない、ツモによって配されたぷよのことである。接続数が4以上となったぷよは、その時点ですべて消える。ぷよが消えたマスは空白となるが、その上部に別のぷよが存在する場合には、その落下により空白は埋められる。

ぷよが消えた後の落下により、さらに4つ以上接続されたぷよが現れた場合には、連続でぷよの消去が発生する（図3参照）。これを連鎖という。連続での消去が n 回発生した際には n 連鎖となる。ぷよが消えるために最低4つを必要とし、フィールドは全78マスであるから、最大連鎖数は19連鎖である。また、連鎖を発動させるためにぷよを消去す

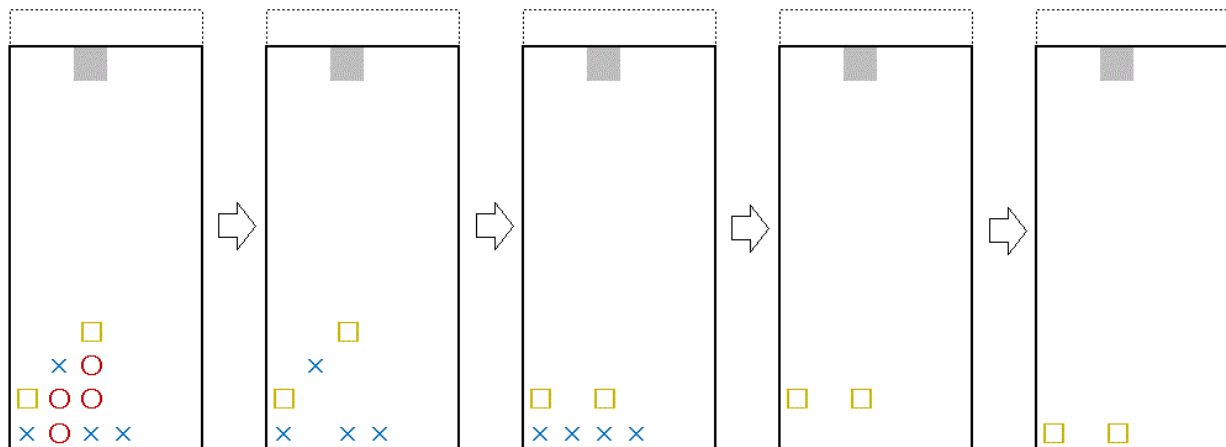


図3 「連鎖」の発生の様子

ることを、「発火」という。

発火時には3列目12段目にもぶよを設置することができ、即座にはゲームオーバーとされない。ぶよの消去および連鎖の発生中には、新たなツモは発生せず、プレイヤーは操作ができない。連鎖がすべて終了した後に、操作可能となる。この時、3列目12段目にぶよが残っていると、その時点でゲームオーバーとなる。

2.3 スコアとおじゃまぶよ

ゲーム中には各プレイヤーのスコアが計算される。スコアは下キーによるぶよの落下操作時と、ぶよの消去および連鎖時、全消し時に増加し、1ゲーム中に減少することはない。落下操作時には、1マスにつき1点が加算される。全消しの際には2100点が加算される。 n 連鎖におけるスコアは、式1で算出できる。

$$\begin{aligned}
 score &= \sum_{i=1}^n (score_i) \\
 &= \sum_{i=1}^n (delPuyo_i \times 10 \times bonus_i)
 \end{aligned} \tag{1}$$

ここで、 $score$ は発動した連鎖の合計スコア、 $score_i$ は i 連鎖目のみのスコア、 $delPuyo_i$ は i 連鎖目で消えたぶよの数、 $bonus_i$ は i 連鎖目でのボーナス係数を表す。

ボーナスは、以下の3つのボーナスの総和で表される。

- 多色ボーナス：同時に消した色の数によるボーナス
- 連結ボーナス：消したぷよの接続数によるボーナス
- 連鎖ボーナス：連鎖数によるボーナス

それぞれのボーナス値を、表 1、表 2、表 3 に示す。ただし、ボーナスの総和が 0 の時には、値を 1 とする。

表 1 多色ボーナス

色数	1	2	3	4
ボーナス	0	6	12	24

表 2 連結ボーナス

連結数	4	5	6	7	8	9	10	11 以上
ボーナス	0	2	3	4	5	6	7	10

表 3 連鎖ボーナス

連鎖数	1	2	3	4	5	6	7	8	9	10
ボーナス	0	8	16	32	64	96	128	160	192	224
連鎖数	11	12	13	14	15	16	17	18	19	-
ボーナス	256	288	320	352	384	416	448	480	512	-

対戦では、算出されたスコアに応じて、対戦相手におじゃまぷよが降る。スコア 70 点がおじゃまぷよ 1 つに相当する。70 点以下は切り捨てられ、次回に持ち越される。対象となるスコアは、これまでにおじゃまぷよに換算されていないすべてのスコアである。つまり、落下操作や全消しで加算されたスコアは、次のぷよの消去時におじゃまぷよの換算に使われる。

おじゃまぷよは連鎖終了後、相手が落下中のツモを設置した時点（相手が次のツモを引く前）で降る。おじゃまぷよの落下列は都度ランダムに選ばれる。ただし一度に複数のおじゃまぷよが落ちる際には、偏りが少なくなるように落下列が選ばれる。例えば 6 個以下のおじゃまぷよが 2 段以上で降ることはなく、おじゃまぷよ 6 個はすべての列に 1 つずつ降る。なお、一度に降るおじゃまぷよの数は 30 個（5 段）が上限であり、一度おじゃま

ぶよが降ると次の一手が配される。上限を超えておじゃまぶよが存在する場合、1 手のツモを置いたのちに再度降ってくる事となる。

自分のフィールドにおじゃまぶよが降るまでには、相手の連鎖発火から連鎖が終了し、さらに自分が 1 手を置くまでの猶予がある。相手からおじゃまぶよが送られている間に自分の連鎖を発火すると、おじゃまぶよの相殺ができる。相殺の後に得点の高かったプレイヤーが、相手にスコアの差分だけおじゃまぶよを降らすことができる。相殺の途中で相手の連鎖が終了しており、かつ相殺し切れなかった場合には、自分の連鎖終了後即座におじゃまぶよが降ってくる。

3 ぶよぶよの対戦

3.1 対戦文化と大会

ぶよぶよの対戦は、ゲームの発売以降長い間にわたって続いてきた。公式大会の歴史は 1995 年にまでさかのぼることができ、コンパイル主催で「全日本ぶよマスターズ大会」や「ばよえ〜んツアー」などが大々的に行われていた。

公式大会以外にも、有志によるゲームセンターやインターネット上での大会は数多く行われてきた。特に上位プレイヤーは、ゲームセンターの AC 版「ぶよぶよ通」で強さを競い、全日本一位の座を目指した。試合形式は 100 本先取戦とされ、時として 200 試合近くに及ぶ対戦によって勝敗がつけられた。試合は非公式であったものの、十分に試合回数を重ねることで、偶然性を排した強者のランキングがコミュニティの中で共有されていた。

近年では、e スポーツ人気の高まりに伴い、ぶよぶよの大会にスポンサーがついたり、上位プレイヤーがテレビ番組に出演する機会が増えている。2016 年の主な大会では、セガ公式「ぶよぶよ」最強プレイヤー決定戦 [14]、第 5 回アーケード版ぶよぶよ通 S 級リーグ [30]、A 級リーグ [29]、統一王座戦 [6]、Red Bull 5G 2016 [4] などが開催された。このように、ぶよぶよの発売以降現在でも、対戦ゲームとしての人気は高い。実用のプレイ AI を実装するにあたっては、対戦を念頭に置くことが重要である。

3.2 対戦の基本的戦術

ぶよぶよは相殺のシステムにより、基本的に高い威力の連鎖を保持するプレイヤーが有利となる (2.3 参照)。お互いの連鎖を同時に発火した場合、スコアの高い連鎖が勝ち、一方的におじゃまぶよを送れるためである。連鎖の威力を上げるためには、同時に消すぶよの数や色の数を増やす、連鎖数を増やすといった方法がある。そのように連鎖を組むため

には、ぷよを消さずにフィールドに積み上げる必要性がある。

フィールドの多くがぷよで埋まっている状態では、少ないおじゃまぷよで残りのフィールドが埋まるために負けやすくなる。つまり、大連鎖を保持するプレイヤーを、小連鎖で倒すことが可能である。倒すまでには至らない場合でも、相手が連鎖を発火できない程度におじゃまぷよを送り、埋めてしまう（つぶし）ことが可能である。このように、単純に大連鎖のみの勝負とはならない奥深さがある。

さらに、片方のプレイヤーのみが大連鎖を発火した場面を考える。発火したプレイヤーの連鎖が進んでいる間も、もう片方のプレイヤーは操作が可能である。連鎖の終了までおじゃまぷよは降らないので、複数のツモを使って連鎖をさらに増やすことができる（伸ばし）。相手の連鎖が終了する前に後から伸ばした連鎖を発火すると、相手の連鎖を相殺し、おじゃまぷよを送り返すことすら可能である。このように、最初の発火時点で両者の連鎖威力が同等あるいは不利だったとしても、そのあとで状況を崩して有利にすることが可能である（後打ち有利）。

大連鎖を先に発火させるべく、相手の発火を促す戦術が成立する（催促）。大連鎖とは別に小連鎖を用意し、そちらを使って相手の発火を催促するのである。相殺できなければ潰されてしまい、発火しても不利な状況に陥る。このため、相手の小連鎖に対応するための小連鎖を、自分もまた用意する必要がある。相手の状況を見て（凝視）、自分の連鎖を組む技術が重要になったのである。

以上で述べた通り、ぷよぷよの対戦にあたっては、連鎖数や威力だけでは語れない駆け引きが生じる。ここで述べた前提を基にして、様々な戦術や連鎖がこれまでに生まれてきた。これらを活用した強い AI を実装することもまた、ぷよぷよ AI の重要な課題である。しかし、本稿ではそれを今後の課題とし、まずは大連鎖の構築に着目する。

4 連鎖構築能力

対戦において連鎖は、相手への攻撃手段でもあり、相手からの攻撃に対する防御手段でもある。そのため、連鎖を構築する能力は基本的かつ重要なものとなる。実戦の上で必要となる連鎖構築能力には、連鎖数、柔軟性、リアルタイム性が求められると考える。その内容を以下で説明する。

4.1 連鎖の威力と連鎖数

連鎖を構築する目的の大部分が、大きなスコアを得るためである。なぜなら、スコアこそが相手に送れるおじゃまぶよの量を決定づけるからである。2.3 で述べた内容から、スコアを大きくするためには、ぶよの消える数や色を増やし、連鎖数を増やすことが有効であると分かる。しかし、フィールドに設置できるぶよの数には限りがあることに注意を要する。すなわち、フィールドを最大限効率よく使うために、同時消しの数をより増やすか、連鎖数をより増やすかを考えなければならない。

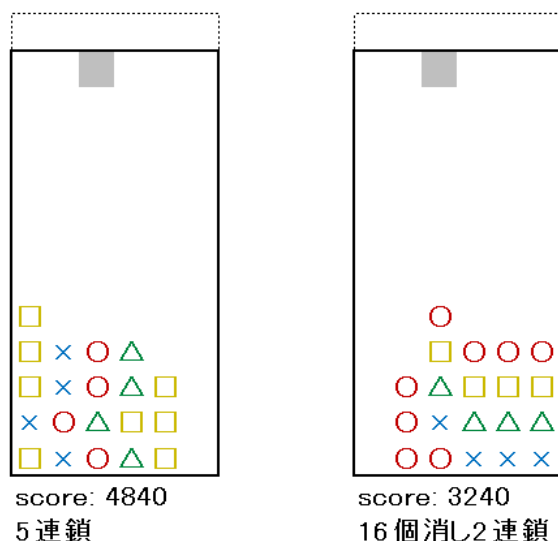


図4 連鎖数とスコア

例えば、20個のぶよを全て消す場合を考える。4個消し5連鎖が合計4840点になるのに対し、2連鎖同時消し（1連鎖目で4個、2連鎖目で4色16個同時消し）では3240点となる（図4参照）。つまり、同じぶよの数であれば、同時消しの数を増やすよりも、連鎖数を増やした方が、スコアが高くなるのである。これは連鎖ボーナス（表3）の上昇幅が大きく、さらに連鎖ごとに繰返し点数が加算されることに起因する。よって限られたぶよ数で威力の高い連鎖を構築するためには、同時消しの数を減らし、連鎖数の効率を上げることが重要である。

組んでいる連鎖に同時消しの数が多かったり、フィールドにおじゃまぶよが存在していたりすると、必然的に連鎖数の上限が減ってしまう。このような観点から連鎖の効率を考える時、「飽和連鎖量」という概念を用いる。そのまま最大まで連鎖を伸ばしたとき、最終

的な威力がどれほどとなるかを示す量である。現在保持している連鎖の量だけではなく、将来的にどれほど連鎖数を向上させることが可能かを考えることもまた、重要な能力である。そのような先読みによって、連鎖を完成させることよりもさらに伸ばすための布石を打つことを優先するような工夫ができ、飽和連鎖量を向上させることができるのである。

4.2 連鎖の柔軟性

連鎖は、大きく定型連鎖と不定形連鎖に分けることができる。定型は、決まった連鎖形と名称が知られたものを指し、不定形は定型で無い連鎖を指す。それぞれの連鎖の例を、図5と図6に示す。ぷよぷよでは配されるツモがランダムに決定されるため、それに合わせて連鎖を組む必要がある。定型連鎖では最終形に合わせてツモを配置してゆき、不定形連鎖ではツモに合わせて最終形を変えてゆく。どちらにしても、ツモに合わせて柔軟に配置を決定する必要がある。

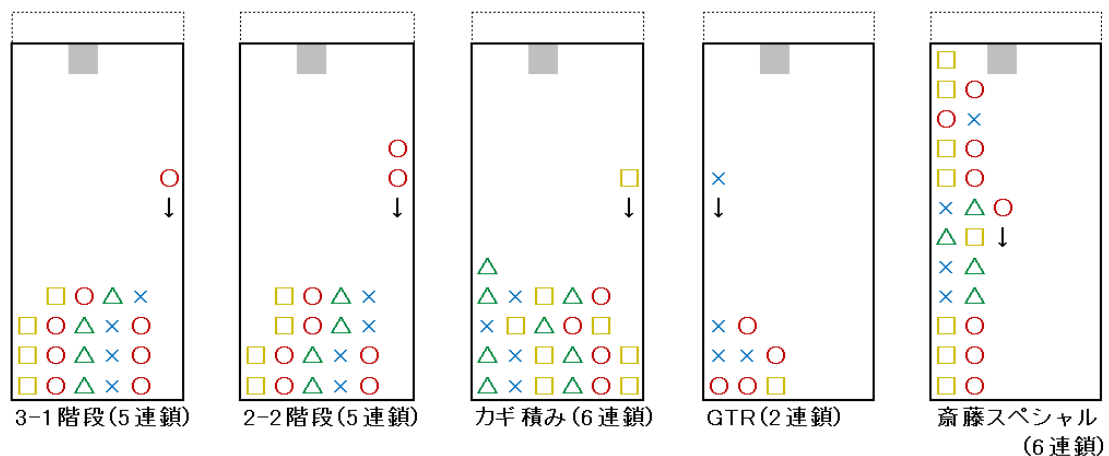


図5 定型連鎖の例

定型連鎖は完成形が定まってはいるものの、実際に組む時のぷよの配置が、固定化されたものであるとはいえない。まず、色の組み合わせに関して厳密に定めることが非現実である。例として 3-1 階段 5 連鎖を挙げると、図7に示したものは全て同じ形といえる。このように、同じ連鎖であってもその色の組み合わせは複数あり、完成形をただ一つに定めることができない。そして色の組み合わせは、ツモの落下順によって定まってゆくものである。一見して同じ形を組む場合であっても、ツモに応じた色の配置の変更が必要である。

その上、配置を固定化してしまうと、必ずしもツモを活かせないという欠点が出す

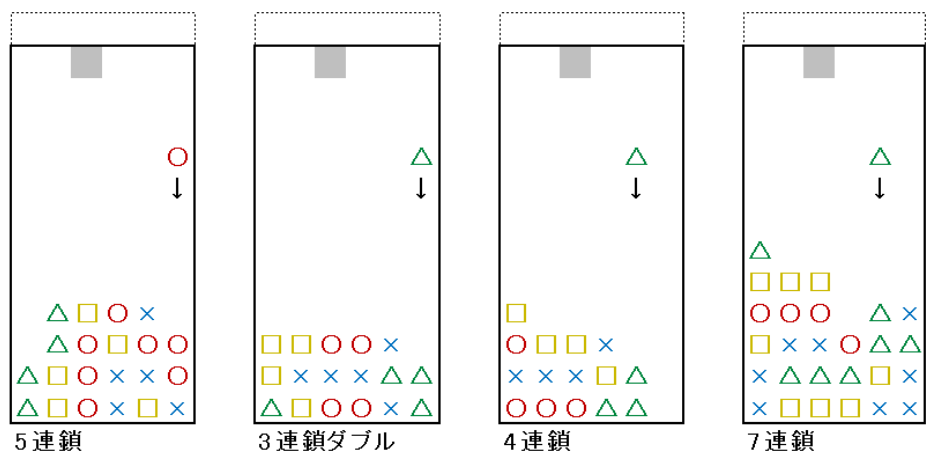


図6 不定形連鎖の例

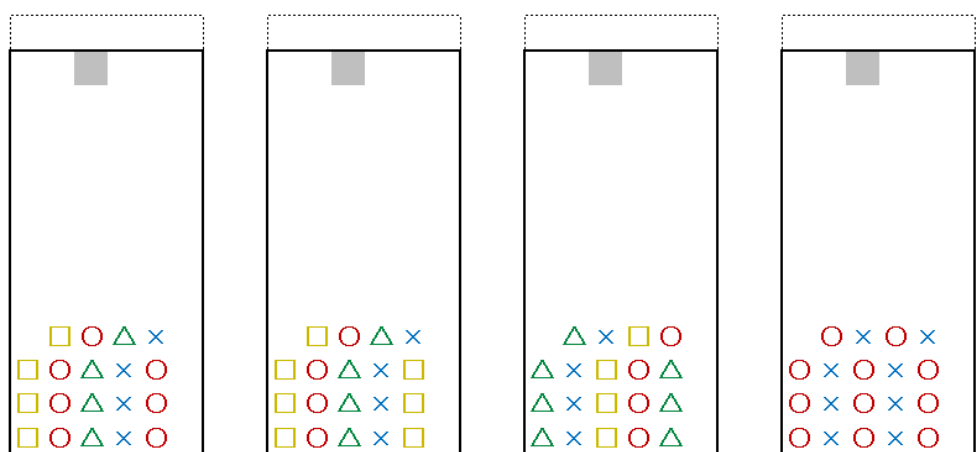


図7 色違いで同形の階段5連鎖

る。図8で示したような例では、決められた連鎖形にするためにはツモを消すか捨てるかしかできない。しかし同じような場面でも、わずかに形を変えることで図9のようにツモを有効活用できる。たとえ定型連鎖であっても、ある程度形を変えられるような柔軟性を持たすことが有効である。ツモに合わせてぶよを配置する柔軟性があることで初めて、余分なぶよを消して形を整えたり（図10）、定型連鎖をわずかに改変したり（図11）することが可能となる。このような配置の柔軟性をさらに顕著にしたものが、不定形連鎖である。

ツモを有効に活用するためには、配色に合わせて柔軟に形を選択し、ぶよを配置する能力が重要である。特に対戦では、相手からおじゃまぶよが送られてきた場合に、定型連鎖をそのまま維持することが困難となる。さらに相手に合わせた連鎖の用意などを見据えた

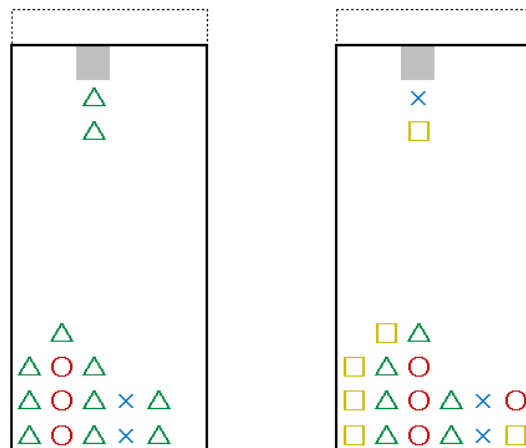


図 8 3-1 階段でツモの置き場が無い例

場合、完成形を定めておくことは困難である。対戦で不定形連鎖が好まれるのは、ツモにあわせて連鎖形を変化させる柔軟性が大きな利点となるためである。

4.3 リアルタイム性

これまでに、連鎖を組む能力としてぶよの置き方に着目してきた。もう一つの重要な能力が、早さである。この早さには 2 種類があり、連鎖を組む早さと、連鎖自体の早さが考えられる。

連鎖を組む早さが勝負に影響することは明らかである。同じ威力の連鎖を組めるが、その早さが異なる 2 プレイヤーの対戦を考える。この場合、片方のプレイヤーが最大連鎖に到達した時点で、もう片方のプレイヤーは相手に劣る威力の連鎖しか保有できていない。当然、連鎖の打ち合いでは速度の遅いプレイヤーが負ける可能性の方が高くなる。威力のポテンシャルが同じでも、リアルタイムに進むゲームであるから、ある時点での保有連鎖量が異なることで有利不利が生じるのである。

ただし、3.2 で述べた通りに、後から連鎖を打つプレイヤーはさらに連鎖を伸ばす余地がある。相手の連鎖が終了するまでの時間、十分な速度で連鎖を組むことができれば、逆に有利な状況に立てる可能性がある。このように、連鎖を組む速度に応じてリアルタイムに戦況が変化する。早くツモをさばけるプレイヤーの方が、連鎖を増やすことや相手を妨害することにより多くのリソースを費やすことができるため、有利となる。

また連鎖数が増えると、アニメーションの描画が繰り返されるために発火から連鎖終了までの時間が長くなる。相手に対し即座におじゃまぶよを送りたい場合などでは、連鎖数

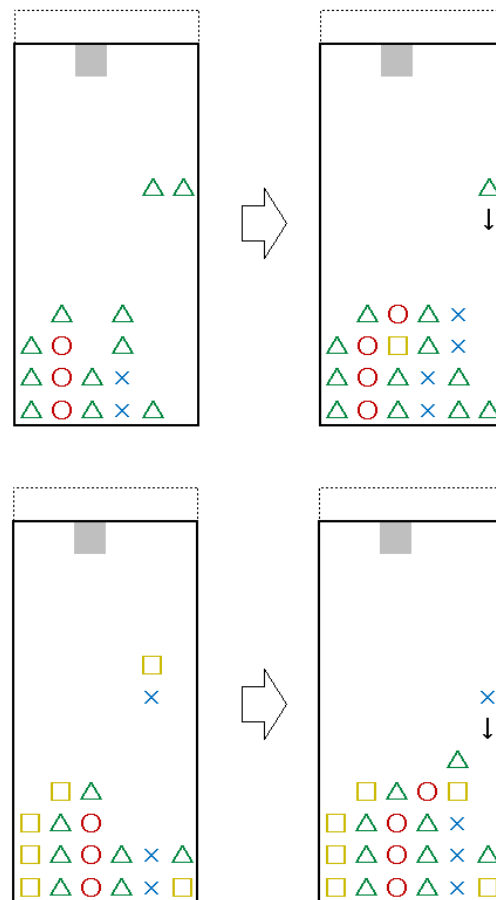


図9 工夫した階段連鎖と完成形

を少なくする方が有効である。威力を犠牲にしても、同時消しや連鎖数の削減により機を逃さずに攻撃すべき局面が存在する。逆に相手が短時間の連鎖を仕掛けてきている時にも、将来的な連鎖数より現時点での短期的な連鎖の完成を目指すべきである。

連鎖構築能力としてのリアルタイム性には、連鎖を構築する早さを高めること、現時点と将来の盤面状況を判断して連鎖を組みかえることが求められる。対戦ではリアルタイム性が高いことを念頭に置き、時間の制約を意識することが肝要である。

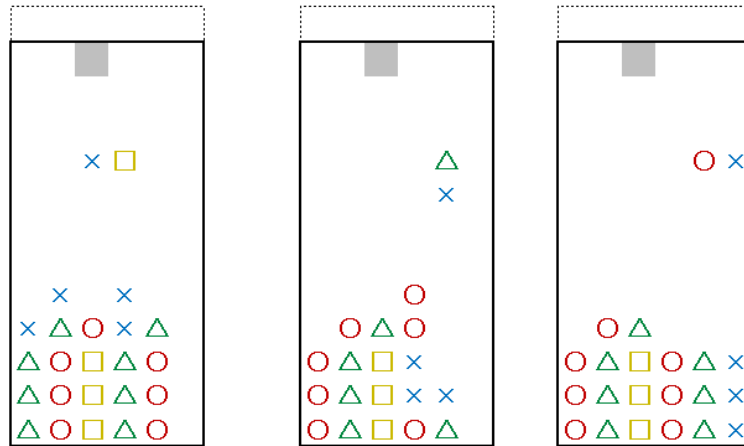


図 10 余分なぷよの消去による整地の例

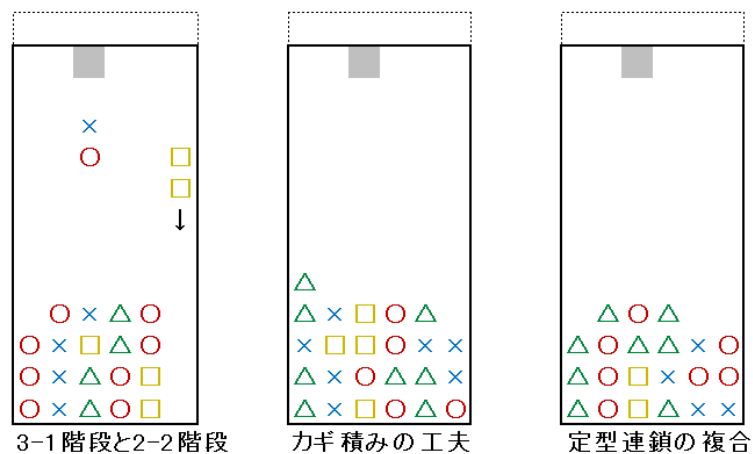


図 11 定型連鎖の改変と組み合わせの例

第 III 部

関連研究

1 DQN

これまでに多くのゲーム AI 研究がなされてきた。近年は、特定のゲームに関する強さを追求することに一旦の区切りをつけ、様々な応用へと挑戦が進んでいる [19, 8]。

2015 年には、DQN(Deep Q-Network) による AI が Atari 2600 のゲーム 49 本の学習

を行った [10]。その結果、43 のゲームで従来研究を上回り、29 のゲームでプロの人間プレイヤーと同等以上の性能を示した。この手法では、ゲームの事前知識なしに学習を行い始め、最終的にゲームのプレイ操作を学習するまでに至ったのである。

DQN は、Q 学習とディープラーニングを組み合わせた手法である。ある時刻 t におけるゲームの状態 s_t を入力とするディープニューラルネットワークを用いて、行動 a に対する評価値 (Q 値) を出力として得る。ニューラルネットワークのパラメータ θ は、以下の式で更新する。

$$\theta_{t+1}(s_t, a_t) = \theta_t + \alpha(R_{t+1} + \gamma \max_a Q_t(s_{t+1}, a; \theta_t) - Q_t(s_t, a_t; \theta_t)) \nabla_{\theta} Q_t(s_t, a_t; \theta_t) \quad (2)$$

ここで、 a_t は t でとった行動、 R_{t+1} は s_t および a_t によって得られた報酬、 α は学習係数、 γ は割引係数を表す。

DQN を使うための環境の実装には、Atari 2600 の環境である ALE(Arcade Learning Environment)[1] の他、スーパーファミコン用の環境である RLE(Retro Learning Environment)[2] が存在している。Bhonker らは 5 つのスーパーファミコンゲームにおける DQN の性能を報告している [2]。

DQN は事前知識を持たないため、様々なゲームに応用できる汎用性が示されている。ただし、学習の開始時点では操作がランダムとなるため、報酬が偶然的に得られないパックマンのようなゲームでは、十分な性能を示せないことが課題となっている。また、ディープニューラルネットワークによる学習のため、学習した知識はブラックボックスとなり、その解析のためにはさらなる技術が必要になる。それでも、繰り返しのゲームプレイからゲームに関する知識を得られるこの手法は、強力なものである。

現状、ぷよぷよのようなパズルゲームに対する DQN の研究は未だなされていない。ぷよぷよは 4 つ以上のぷよをつなげることで得点を得られるが、これはランダムなゲーム行動の中でも起き得る現象である。同様に、学習の中で連鎖が起きる可能性も十分にありうる。そのため、DQN による学習が実用的である可能性が示唆される。DQN によりぷよぷよの連鎖構築知識が得られるかを確かめることは、有意義といえるであろう。

2 ゲームと楽しさ

「楽しさ」とは何か、という問いは分野を問わず、難しい問題である。人は様々な場面で楽しさを感じるが、それが本質的にどういったものかを明確に示すことは難しい。

Csicszentmihalyi[5] は、楽しさを「フロー理論」によってモデル化した。フロー理論で

は楽しさを、物事に没入したフロー状態として捉える。フロー状態は、人のスキルと課題の挑戦度がちょうど良いバランスとなった場合にのみ起きると主張される（図 12 参照）。スキルに対して課題が簡単過ぎれば退屈であり、課題が難し過ぎれば不安となって没入を妨げる。フロー状態は、様々なゲームやスポーツ、さらには日常生活においても起こりうるものである。

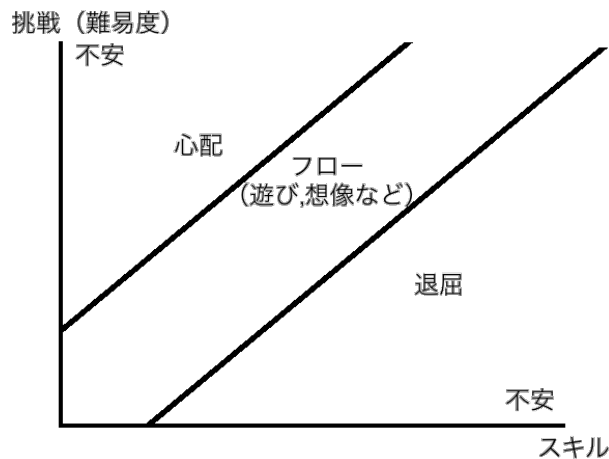


図 12 フロー理論の概略図

また Malone[9] は、コンピュータゲームの楽しさに着目し、「挑戦」「好奇心」「ファンタジー」が重要であると主張した。挑戦は Csicszentmihalyi と同様であり、加えて予測不可能性やランダム性としての好奇心と、表現やデザインとしてのファンタジーを要素に挙げた。Malone の観点からは、同じレベルの挑戦にも、プレイヤーの知識に応じた質的な違いが生じうることを考慮できる。

これらのモデルは様々な娯楽やゲームに適応できる汎用的なものである。そのため、実際にはどのようなゲーム要素が楽しさの因子となるかを、具体化することが求められる。山下ら [23] はアンケート調査によってゲームの楽しさの因子を分析した。「ゲーム本来の楽しさ」という因子には「挑戦的な」「好奇心を駆り立てる」などが含まれ、Malone の主張を裏付けた。さらにゲームの種類を分類し、それぞれの分類ごとに求められる楽しさの因子が異なることを示した。

以上の議論から、ゲームには全体的に挑戦や好奇心が求められ、それはプレイヤーに見合ったものでなければならないとわかる。プレイヤーのスキルや個々のゲームに応じて、挑戦や好奇心の表現方法が変わるのである。

例えばぶよぶよの連鎖構築では、「挑戦」に連鎖数や構築速度、「好奇心」に連鎖の柔軟

性を当てはめられるであろう。プレイヤースキルの上昇とともに連鎖数の上限や速度に対する挑戦のハードルも上昇する。同じ連鎖数でも様々な連鎖形に触れ合い、構築することは好奇心を刺激すると思われる。さらに対戦では、戦術の正確さや幅広さが楽しさの要因に関わるものと考えられる。

3 楽しさに関するゲーム AI

人を楽しませることを目的とする AI 研究が、徐々に行われつつある。フロー理論を踏襲し、プレイヤーのスキルと挑戦のバランスをとるために AI の強さを調整するアプローチが研究されている。このような研究は DDA(Dynamic Difficulty Adjustment) と呼ばれ、Tan ら [17] はレースゲームで、中川ら [11] は格闘ゲームで、それぞれリアルタイムに AI エージェントの強さを調整した。これらの手法はゲーム内スコアを参照し、拮抗した試合を実現する。

しかし、スコアのみを評価基準として難易度調整をすると、AI の強さが不自然に変化してしまうことが指摘されている。池田 [25] は囲碁、将棋における AI において、手加減の自然さが重要であることを主張した。自然な AI のために、人間を模倣する AI や、プレイヤーの特徴を考慮する Player Modeling の研究が複数行われている [19, 8]。

Yannakakis ら [18] は Player Modeling に、Malone の主張した挑戦と好奇心を取り入れた。Bug Smasher という光るタイルを踏むゲームで、点滅の早さを挑戦、光る位置のエントロピーを好奇心とした。これらのゲーム状態とプレイヤーの動きを入力とするニューラルネットにより学習を行い、プレイヤーの楽しさを推定することができた。

Csikszentmihalyi や Malone の提唱した概念は、実用的な応用に活かされている。対戦 AI の分野では未だ研究が不十分であるものの、挑戦や好奇心を考慮した AI により、人と同等以上の楽しさを得られる対戦相手を実現できる可能性がある。

4 ぷよぷよ AI

これまでにぷよぷよの連鎖構築 AI は複数研究されてきた。主なアルゴリズムに、ポテンシャル最大化法、モンテカルロ法、定石配置法がある。以下でそれぞれについて説明する。

4.1 ポテンシャル最大化法

ポテンシャル最大化法は、ぷよぷよとその変形である「崩珠」で利用され、モンテカルロ法や定石形配置法との比較に用いられた基礎的な手法である [24, 31]。その手続きは、以下の通りである。

1. n 手先までの配置可能な手および盤面を全て列挙する。
2. n 手目以外でぷよを消去する手を除外する。
3. n 手目で連鎖を発火したとき、スコアが最大となる手を選択する。ただし、候補手が複数ある場合には、その中からランダムに選択する。
4. 選択された手に至る一手目を、現在のツモの配置として決定する。

この手法では、予告されたツモを用いて総当たりで配置法をシミュレーションする。ぷよぷよでは 3 手分のツモが示されるため、 $n = 3$ として実装されてきた。一方で、その先の手を一切考慮しないために、長期的には効率の悪い連鎖となることが多い。富沢ら [31] の実験では、最大限まで連鎖を伸ばした時でも、平均 8.80 連鎖程度であると示された。よって、3 つの連鎖構築能力の内、連鎖数に問題を抱えた手法であるといえる。一方、ツモごとに連鎖形を変えたり、候補手をランダムに選択したりすることから、連鎖の柔軟性はとても高いと考えられる。

4.2 モンテカルロ法

ポテンシャル最大化法では、探索深さ（手数）が浅いために、十分な先読みが出来ないことが問題であった。そこで、幅有線探索ではなく深さ優先探索の手法として、UCT 探索を用いた研究がなされた [24]。

ここで、探索深さが 4 以上の場合には、未知のツモと配置法の双方についてシミュレーションが必要であることに注意を要する。大月らは、プレイヤーが任意に選択できる配置法と、そうでないツモの探索を階層化して区別した。具体的には、前者を操作対象ノード、後者を非操作対象ノードとし、ノードの選択と評価において区別を行っている。選択の際には、非操作対象ノードを UCB 値によってではなく、探索回数の低さによって選ぶ。また評価値の更新では、非操作対象ノード i の評価値 $Q_{i,unc}$ および操作対象ノード j の評価値 $Q_{j,c}$ を、式 3 および式 4 によって定義した。

$$Q_{i,unc} = \max_{j \in S_i} Q_{j,c} \quad (3)$$

$$Q_{j,c} = \mu \min_i Q_{i,unc} + (1 - \mu) \frac{1}{N} \sum_{i=1}^N Q_{i,unc} \quad (4)$$

なお、 S_i はノード i の子ノードの集合、 μ は戦略に関するパラメータ、 N は非操作対象ノードの数を表す。これらの式により、最も連鎖の威力を高めるツモの配置を、将来的なツモの違いによる影響を抑えつつ決定できる。

本手法を用いた実験により、従来の UCT 探索による手法やポテンシャル最大化法より、有意に連鎖数が向上することが示された。しかし、1 手あたりの計算時間を 60 秒に設定し、ポテンシャル最大化法より約 1 連鎖の向上にとどまったことは、この手法の問題点を示唆している。ポテンシャル最大化法はもとより、連鎖数に関して不十分なアルゴリズムであり、そこに 1 連鎖程度の向上があったとしても、未だ威力が不足していると考えられる。その上に探索に大きな計算資源が必要となり、リアルタイム性の確保が難しい点も解決の必要がある。

4.3 定石形配置法

ぶよぶよの連鎖構築において、これまで行われた探索アルゴリズムは非効率であった。そこで富沢ら [31] は知識として定石形、すなわち定型連鎖を用いた連鎖構築アルゴリズムを提案した。その手続きは、以下の通りである。

1. 定石テンプレートを関連性行列として複数用意しておく。
2. n 手先で可能なすべての盤面を関連性行列として列挙する。
3. 各盤面とテンプレート行列の合致度を算出する。
4. 合致度が最大となる盤面に至る 1 手目を決定する。

ここで、関連性行列とは各マスにおいて、色の関係性を表した行列である。テンプレート行列においては 2 つのマスの関係が (1) 同色である、(2) 異色である、(3) どちらでも良い、として表現し、盤面の状態行列においては (3) を片方もしくは両方のマスが空きである、として表現した。この工夫により色が抽象化され、用意するテンプレートの数を減らすことが可能となる。

実験においては階段 8 連鎖を 10 種、挟み込み 8 連鎖を 12 種用意した。それぞれのテンプレート群で、おおよそ 24 手で完成（合致度 0.95 以上）し、多くの場合で人間の熟達者と遜色無い結果となった。1 手あたりの計算時間も 0.47 秒から 0.71 秒程度であり、十分に短い。なお、10 % ほどの試行では大幅に手数を要したことから、柔軟性の不足が指摘

されている。この解決のためには、テンプレート群に複数の連鎖形を含めるなど、テンプレートの数を増やすことが有効であると示された。

さらに、定石形配置法とポテンシャル最大化法を組み合わせた最大連鎖について検討している。それによれば、平均では 41.9 手で 10.99 連鎖を構築することができている。ポテンシャル最大化法が 41.2 手で 8.80 連鎖であることと比較すると、+ 2 連鎖の向上が見られる。ただ、テンプレート 8 連鎖を 24 手で完成させたことを考慮すれば、ポテンシャル最大化法では 18 手を使って 3 連鎖程度を増加させたにすぎず、ポテンシャル最大化法の効率の悪さが際立つことを指摘できる。

定石形配置法によって、定型連鎖の構築には一定の成果が得られている。しかし、依然として不定形連鎖や、定型連鎖を崩した連鎖などの構築には、課題が残されている。連鎖形を増やすことは、テンプレートを増やすことで可能ではあるものの、連鎖形は膨大な数があるからテンプレート行列を増やすことに大きな手間がかかり、合致度の算出処理により多くの計算資源を必要とするなどの問題がある。定石配置法では連鎖数、構築速度の点で優れるが、柔軟性に関しては未だ不十分であるといえる。

5 その他のパズルゲーム AI

テトリス、パネルでポン：どのようにこの研究と関わるか？

第 IV 部

DQN による学習

DQN を始めとするゲーム AI 実行のプラットフォームとして、ALE[1] や RLE[2] が存在する。また、DQN も DQN-chainer[20] のように実装済みのものが公開されている。

ぷよぷよはスーパーファミコンでもいくつかのタイトルが発売されており、中でも「ぷよぷよ通」が存在している。RLE を用いて DQN の学習を行い、ぷよぷよのプレイ知識を獲得することが期待できる。そのため、今回は RLE と DQN-chainer を用いてスーパーファミコンのタイトル「す〜ぱ〜ぷよぷよ通 リミックス」の学習を行う。学習により、連鎖構築知識を獲得できるかを確かめ、考察することを目的とする。

1 とこぷよ

始めに、一人用モードの「とことんぷよぷよ」（とこぷよ）において学習を行った。とこぷよでは時間制限や敵プレイヤーの妨害は無く、ぷよを消し続ければ延々とプレイできる。また、プレイを続けることで落下速度が変化したり、レベルが上がってぷよを消去したときの獲得点数がレベル倍されたりする。時間さえかければ、淡々とぷよを消し続けて大きなスコアを獲得できるのである。

一方で、ぷよを積み上げて連鎖を構築することで、効率よく大きな点数を得ることが出来る。繰り返しの学習過程で、小連鎖であれば偶然起こり得ると考えられる。理想的には、その連鎖の知識を学習し、学習が進むにつれて連鎖数が増大してゆくことが期待された。

1.1 実験方法

表 4 DQN の学習設定

学習係数 α	0.3
割引き係数 γ	0.99
状態 s	4 フレーム分のゲーム画像
操作 a	左右下方向キー、右回転、左回転（5 種類）
報酬 R_{t+1}	時刻 t から $t+1$ の間で増加したスコア
記録状態数	100000
学習ステップ数	50000
繰り返し数	100

2 対戦

第 V 部

連鎖ポテンシャル法の改良

1 探索深さの変更

2 全探索

3 途中での消去

第 VI 部

人間の知識適用

1 連鎖構築知識の抽出と実装

2 結果と考察

第 VII 部

結論

1 研究成果

2 今後の課題

参考文献

- [1] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. *ArXiv e-prints*, July 2012.
- [2] N. Bhonker, S. Rozenberg, and I. Hubara. Playing SNES in the Retro Learning Environment. *ArXiv e-prints*, November 2016.
- [3] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, pp. 1–43, March 2012.
- [4] Red Bull. Red bull 5g 2016. <http://www.redbull.com/jp/ja/games/events/1331814625758/red-bull-5g-2016>. Last visited: 2017–2–3.
- [5] Mihaly Csikszentmihalyi. 楽しみの社会学. 新思索社, 2000. 今村浩明 訳.
- [6] e Sports Runner. 統一王座戦. <http://esports-runner.com/puyopuyo/undisputed-championship/>. Last Visited: 2017–2–3.
- [7] Johan Huizinga. ホモ・ルーデンス. 中央公論社, 1971. 高橋英夫 訳.
- [8] R. Lopes and R. Bidarra. Adaptivity challenges in games and simulations: A survey. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 3, No. 2, pp. 85–99, June 2011.
- [9] Thomas Malone. What makes computer games fun? (abstract only). *SIGSOC Bull.*, Vol. 13, No. 2-3, pp. 143–, May 1981.
- [10] 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, 02 2015.
- [11] Y. Nakagawa, K. Yamamoto, and R. Thawonmas. Online adjustment of the ai’s strength in a fighting game using the k-nearest neighbor algorithm and a game simulator. In *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, pp. 494–495, Oct 2014.

- [12] SuperData Research, カドカワ株式会社（編）. e スポーツグローバルマーケットレポート 2016. カドカワ株式会社, 2017.
- [13] S. Risi and J. Togelius. Neuroevolution in games: State of the art and open challenges. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. PP, No. 99, pp. 1–1, 2015.
- [14] SEGA. セガ公式「ぷよぷよ」最強プレイヤー決定戦. <https://segafes.com/puyo-saikyo/>. Last visited: 2017–2–3.
- [15] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, Vol. 529, pp. 484–503, 2016.
- [16] IEEE Computational Intelligence Society. Ieee conference on computational intelligence and games. <http://www.ieee-cig.org/>. Last visited: 2017–2–3.
- [17] C. H. Tan, K. C. Tan, and A. Tay. Dynamic game difficulty scaling using adaptive behavior-based ai. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 3, No. 4, pp. 289–301, Dec 2011.
- [18] G. N. Yannakakis and J. Hallam. Real-time game adaptation for optimizing player satisfaction. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 1, No. 2, pp. 121–133, June 2009.
- [19] G. N. Yannakakis and J. Togelius. A panorama of artificial and computational intelligence in games. *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 7, No. 4, pp. 317–335, Dec 2015.
- [20] Naoto Yoshida. Dqn-chainer. <https://github.com/ugo-nama-kun/DQN-chainer>. Last visited: 2017–2–3.
- [21] カドカワ株式会社（編）. ファミ通ゲーム白書 2016. カドカワ株式会社, 2016.
- [22] 陽一郎三宅. デジタルゲームにおける人工知能技術の応用の現在 (特集: エンターテインメントにおける ai). 人工知能: 人工知能学会誌, Vol. 30, No. 1, pp. 45–64, jan 2015.
- [23] 利之山下, 孝昭清水, 裕栗山, 友茂橋下. コンピュータゲームの特性と楽しさの分析. 日本教育工学会論文誌, Vol. 28, No. 4, pp. 349–355, apr 2005.
- [24] 龍大月, 新一前田, 信石井. 不完全情報ゲームに対する階層化したモンテカルロ探索

- とそのぷよぷよへの適用. 電子情報通信学会技術研究報告. NC, ニューロコンピューティング, Vol. 113, No. 500, pp. 275–280, mar 2014.
- [25] 心池田. 楽しませる囲碁・将棋プログラミング (特集: 娯楽の or). オペレーションズ・リサーチ: 経営の科学, Vol. 58, No. 3, pp. 167–173, feb 2013.
- [26] 直樹藤田. 米国におけるビデオ・ゲーム産業の形成と急激な崩壊—現代ビデオ・ゲーム産業の形成過程 (1). 経済論叢, Vol. 162, No. 5, pp. 440–457, nov 1998.
- [27] 直樹藤田. 「ファミコン」開発とビデオ・ゲーム産業形成過程の総合的考察—現代ビデオ・ゲーム産業の形成課程 (3). 経済論叢, Vol. 163, No. 5, pp. 511–528, may 1999.
- [28] 直樹藤田. 「ファミコン」登場前の日本ビデオ・ゲーム産業—現代ビデオ・ゲーム産業の形成過程 (2). 経済論叢, Vol. 163, No. 3, pp. 311–328, mar 1999.
- [29] 日本ぷよ連盟. 第 5 回アーケード版ぷよぷよ通 a 級リーグ. http://puyo.in/league_a2015/. Last visited: 2017-2-3.
- [30] 日本ぷよ連盟. 第 5 回アーケード版ぷよぷよ通 s 級リーグ. http://puyo.in/league_s2015/. Last visited: 2017-2-3.
- [31] 大介富沢, 心池田, シモンビエノ. 落下型パズルゲームの定石形配置法とぷよぷよへの適用. 情報処理学会論文誌, Vol. 53, No. 11, pp. 2560–2570, nov 2012.