

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

富沢 大介^{1,a)} 池田 心^{1,b)} シモン ビエノ¹

受付日 2012年2月20日, 採録日 2012年9月10日

概要: 囲碁における定石や将棋における定跡は、主にゲーム序盤で用いられる決まった手順の互角の応酬であり、長い時間をかけて研究・洗練された人智の結晶である。テトリスやぷよぷよなどの落下型パズルゲームの多くでも、展開を有利にするための定石形が存在し利用されている。これらのゲームを囲碁や将棋と比較すると、自分と相手双方に盤が存在し邪魔は間接的にしか行われぬ一方、操作対象の与えられ方（俗にツモと呼ばれる）にランダム性があり、状況に応じて用いる定石や配置順を変えていかなければならない難しさがある。本論文では、関連性行列という形で状態と定石を表現する定石形配置法を提案し、これをぷよぷよにおける連鎖の構成に適用することでその有効性を示す。

キーワード: 落下型パズルゲーム, ゲーム木探索, 定石形表現, ぷよぷよ

Stylized Assignment in Tile-matching Puzzle Game and its Application to Puyo-Puyo

DAISUKE TOMIZAWA^{1,a)} KOKOLO IKEDA^{1,b)} SIMON VIENNOT¹

Received: February 20, 2012, Accepted: September 10, 2012

Abstract: In Go and Shogi, standard patterns and sequence of moves have been developed over the years by human players, mainly in the opening of the game. Such standard patterns, used by expert players to reach a winning position, also exist in tile-matching video games like Tetris and Puyo-Puyo. In tile-matching games, the interaction between the players is only indirect through separate boards, but the randomness of the tiles appearing in the game is a major difficulty, not found in Go or Shogi. In this paper, we propose a tile arrangement method for finding good moves in tile-matching games, through the use of a relevant matrix that represents the current situation and the knowledge of standard patterns. We show the effectiveness of the proposed method by applying it to the construction of chains in Puyo-Puyo. The resulting Puyo-Puyo AI player is significantly stronger.

Keywords: tile matching puzzle game, game tree search, representation of standard patterns, Puyo-Puyo

1. はじめに

ほとんどのボードゲームやパズルゲームでは、与えられた盤面の状態に対して行動を選択し、ゲームの目的に沿って望ましい状態に導いていくことが求められる。ゲームによって盤面にはさまざまな大きさや形状があり、行動にも追加・削除・移動・交換などさまざまなものがあるが、どのような場合でもゲームごとに特定の“好ましい盤面のパ

ターンや手順”が存在し、活用されている場合が多い。囲碁における定石 [9]、将棋における定跡などはその典型であり、最善かつ互角と思われる手順の応酬がアマチュアからプロ棋士に至るまで当然のように利用されている。ゲーム中の短い時間の思考や探索では最善手の特定は困難であり、長い時間をかけて研究・洗練された人智の結晶である定石・定跡を活用することは多くの場合有効である。

本論文では、落下型パズルゲーム（俗に落ちもの、落ちゲーとも呼ぶ）における定石形に着目し、これをゲーム AI で構成することを目的とする。「テトリス」や「ぷよぷよ」を代表とする落下型パズルゲームはおおむね、1 人が 1 つの盤（フィールド）を持ち、与えられた操作対象（ピース、

¹ 北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology, Nomi,
Ishikawa 923-1292, Japan

^{a)} s1010043@jaist.ac.jp

^{b)} kokolo@jaist.ac.jp

ぶよ、石など呼び方はさまざまである)を位置と向きを定めて落下させ、複数回繰り返して特定の条件を満たすことでまとめて消去するという枠組みを持つ。各要素にゲームごとの違いはあるものの、好ましい盤の状態(定石形)があり、それを戦略的に構成することが強いAIを作成するために重要であろう点は共通していると考ええる。

囲碁や将棋といった二人確定完全情報ゲームと比較すると、落下型パズルゲームでは敵プレイヤーに自分の盤を直接的に邪魔されることが少ない点、一方、与えられる操作対象(俗にツモと呼ばれる)にほとんどの場合ランダム性がある点が大きく異なる。このランダム性のために、プレイヤーは自分の目標とする定石形をつねに同じ手順では達成できず、状況に応じて多様な手順を用いたり、別の定石形を選択したりする必要に迫られる。

本論文では、落下型パズルゲームの多くに共通する特性を考慮し、関連性行列という形で盤の状態や定石形を表現して着手を選択する方法を提案する。そのうえで、多くの達人プレイヤーが高度な対戦を行うなど落下型パズルゲームの代表格である「ぷよぷよ」に適用し、定型連鎖と呼ばれる定石形を効率的に構成できること、またAIがゲーム中に定石形を効率的に構成できることを複数の実験により示す。

2. 対象クラスと目的の設定

2.1 対象クラス

現在、落下型ゲームに分類されるゲームは市販のものだけでも数十種類にのぼり[10]、盤の大きさ・形状、操作の種類、消去の条件、各要素の呼称、着手のタイミング、プレイヤー人数、勝利条件などは非常に多岐にわたる。本論文で提案する手法は多くのゲームにそのままあるいはわずかの変更で適用可能であると考え、議論の具体化のために対象クラスをある程度限定し、同一の呼称を用いることにする。本論文で想定するゲームの各要素は以下のとおりである。

- [プレイヤー数] 1人のプレイヤーが1つの盤を持つ。複数のプレイヤーがゲームをする場合は、それらは間接的にしか影響を与えないものとする。
- [盤とマス] 盤は n 個のマスからなり、有限かつ離散的であるとする。盤の形状・次元・位相・重力の有無は問わないが、ゲーム中は不変であるとする。多くの場合、重力のある長方形格子状の盤が用いられる。
- [石] 1つのマスは、空であるか、1つの石が占めるものとする。石には複数で有限の種類または状態が存在することを想定する。プレイヤーは盤のすべてのマスの情報を得られるものとする。
- [配石] プレイヤーには、1つまたは複数の石からなる配石(ツモ)が与えられる。石の種類は通常、ランダムに決められる。ただし多くの場合、1手先、2手先の配石の種類は予告されており木探索は可能である。

- [着手] 盤の状態と配石から、有限個の合法手集合が定まる。これは多くの場合、配石の回転と移動を意味する。プレイヤーは合法手を1つ選択して着手する。
- [状態遷移] 盤の状態と着手から、(落下、消去などのルールに基づき)次状態が決定される。状態遷移は配石を除いて確定的であるとする。

「テトリス」、「ぷよぷよ」、「コラムス」、「Dr. マリオ」、「パズルボブル」、「ヨッシーのクッキー」、「パネルでポン」など著名な落下型パズルの多くはこのクラスに属する。ただし最後の2つは、配石を操作して着手するわけではない(着手は盤上の石の移動によって行う)点や、状態遷移と配石の区別が明確でない点などに注意が必要であり、他にもアクション性・非同期性が強いゲームなどでは本論文で提案する手法の適用が現実的ではないものも多いと考える。なお、他プレイヤーの行動が無視できるとすれば、このクラスは配石のみに不確定性を持つ有限状態離散行動の完全観測マルコフ決定過程[7]の1つであるという言い方もできる。

2.2 直接的な状態表現の限界と本論文の目標

定石の最も単純な表現は、「あるパターンを持つ状態の場合、ある行動をとるのが良い」というものであり、状態やパターンから行動、ないし行動評価値への写像は囲碁などでも直接的[4]間接的[5]にしばしば用いられる。しかし配石がランダムに与えられるようなゲームではこの手法はとりにくく、状態を評価して良い状態に到達することを目指すような方法をとらざるをえない。

現在の盤の状態を $s = [s_1, s_2, \dots, s_n]$ と各マスの状態の配列で直接的に表すとする。各マスの状態の好ましさが他のマスと独立であるならば、それを関数の配列 $t = [t_1, t_2, \dots, t_n]$ で表し、たとえば加算 $t(s) = \sum_{i=1}^n t_i(s_i)$ や乗算を用いて状態を評価することができる。任意の状態のスカラ評価値が計算できるなら、探索などを用いて評価値が高くなるような着手を選択することで目的の状態に近づけることが可能であろう。たとえば、図1(a)はテトリスにおける定石形の1つであり、左端1列には石が存在せず、それ以外には石が存在する状態が好ましいということである。テトリスでは、1度落下した石には機能の違いは存在せず、状態は2通りしかない。「上部に石がある空マス」を好ましくない別状態として扱うことはありうるが、それでもせいぜ

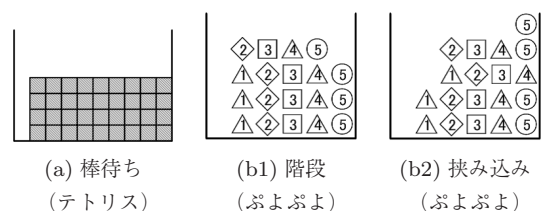


図1 定石形の例

Fig. 1 Example of standard patterns.

い3つの状態表現で十分であり、これ以上複雑な定石表現は不要である。

一方で、図1(b)は4章で詳述するぷよぷよにおける定石形の1つである。ぷよぷよには通常4種類の石があり、◇□△○の記号はその種類を表している。この種類の組合せだけを正解とするなら、テトリスと同様の状態表現を用いることは可能である。しかし実際には石の種類に絶対的な意味は存在せず、これ以外の組合せ、たとえば1と数字の振ってある部分が○や□だったり、5と数字の振ってある部分が□や◇であっても問題ない。これらの組合せはこの場合4×34通りあり、それぞれを別の定石形として直接的な状態表現で用意することは効率的ではない。

そこで本論文では、2つのマスの状態の関係、特に石の種類が同じか違うかに着目することにする。そのうえで、定石形を、「ある2つのマスの石が同じ種類であるべきなのか、違う種類であるべきなのか、どちらでもよいのか」を表す重み付き隣接行列（関連性行列と呼ぶ）の形式で表現することで効率化を果たすことを目標とする。たとえば図1(b)であれば、これは「1のマスどうしは同じ種類でなければならない」「1のマスと2のマスは違う種類でなければならない」「1のマスと3のマスはどちらでもよい」などのように1通りに表現できる。

2つ（やそれ以上）のマスの状態の関係を評価して合算、探索に用いる手法は将棋では頻繁に用いられる[6]。本論文で提案する手法は、2マス関係のうち、落下型ゲームに頻繁に登場する概念だけを効率的に表現することを目指したものと見える。

3. 関連性行列を用いた定石形配置法

3.1 全体の枠組み

盤上の石どうしの関連性を表した行列を関連性行列と呼び、これを状態表現と定石形表現の両方に用いる。提案手法の概要を図2に示す。実際の着手の決定手順は以下のとおりである。

- (1) 定石形を関連性行列で表現し、「定石テンプレート」として m_T 個用意する。
- (2) ゲーム時は、配石の予告されている数手先までのすべての局面を探索し、その葉ノード（状態）を関連性行列の形で「状態行列」として表現する。
- (3) 各葉ノードと各テンプレートを比較して合致度を計算する。
- (4) 合致度に応じて最も有望な葉ノードを選択する。葉ノードに至る1手目の行動を着手する。

3.2 状態行列

状態行列 S は盤のある注目するマス i の石の種類に対して他のマス j の石の種類が「同じ」か「違う」か「どちらかのマスが空き」かを表す3値 $n \times n$ の行列とする（式

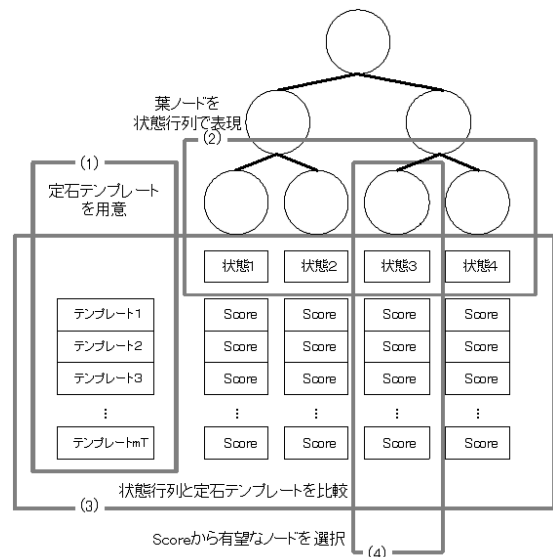


図2 関連性行列と探索による着手選択
Fig. 2 Framework of template matching search.

(1)). n は盤面のマスの個数である。同じ場合 +1, 違う場合 -1, どちらかが空きの場合は 0 で表す。状態行列 S は対称行列で、実際に必要なのは右上三角成分のみとなる。

「マス i に特定の種類の石がある」という情報はこの表現では扱えず、また「どちらか一方のマスのみが空き」「両方のマスが空き」の2つを区別することもできない。多くのゲームでそのような情報は不要と考えているが、必要な場合には何らかの拡張が必要である。

$$S = \begin{pmatrix} s_{1,1} & \cdots & s_{n,1} \\ \vdots & \ddots & \vdots \\ s_{1,n} & \cdots & s_{n,n} \end{pmatrix} \quad (1)$$

$$S_{i,j} = \begin{cases} +1 & \text{両方に石があり、同種類の場合} \\ -1 & \text{両方に石があり、種類が異なる場合} \\ 0 & \text{どちらかまたは両方のマスが空きの場合} \end{cases} \quad (2)$$

3.3 テンプレート行列

テンプレート行列 T は定石形を関連性行列の形で表現したものである。盤のある注目するマス i の石の種類に対して他のマス j の石の種類が「同じであるべき」か「異なるべき」か「どちらでもよい」かを表す実数値 $n \times n$ の行列とする（式(3)）。同じであるべき場合は正值を、異なるべき場合は負値を与える。さらに、その絶対値は優先的に着手すべき箇所が大きくなるように設定する。テンプレート行列 T は対称行列で、実際に必要なのは右上三角成分のみとなる。

$$T = \begin{pmatrix} t_{1,1} & \cdots & t_{n,1} \\ \vdots & \ddots & \vdots \\ t_{1,n} & \cdots & t_{n,n} \end{pmatrix} \quad (3)$$

$$\begin{cases} t_{i,j} < 0 & \text{異なるべき場合} \\ t_{i,j} > 0 & \text{同じであるべき場合} \\ t_{i,j} = 0 & \text{どちらでもよい場合} \end{cases} \quad (4)$$

3.4 合致度の算出方法

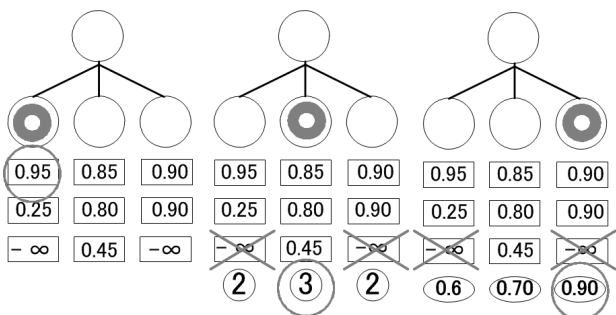
式 (5) は、本論文で用いる合致度の定義である。状態行列 S とテンプレート行列 T の比較は、基本的に要素ごとの積を加算することで行う。つまり、同じ種類であるべき場合に同じ・違う種類であるべき場合に違うなら正値を与え、加算する。しかし、多くのゲームの多くの定石形では、同じ種類であるべき場合に違う種類、違う種類であるべき場合に同じ種類の石のペアがあることは致命的であり、これは本論文ではハード制約として扱うことにする。すなわち、1つでもこのような食い違いがある場合には合致度は $-\infty$ とすることにする。もちろんこれをソフト制約として扱う拡張もありうる。

分母は定石形が完成した場合に合致度 1 とするための正規化項である。用いるテンプレートが似通ったものの場合、多少重みに大小があったとしてもこの正規化によりお互いを適正に比較することができる。大きく異なる多様なテンプレートを混合する場合には、適正な比較ができるように何らかの重みづけの工夫が必要である。

$$score_t(S) = \begin{cases} -\infty & \text{if } \exists i, j \quad s_{i,j} t_{i,j} < 0 \\ \frac{\sum_{i=1}^n \sum_{j=1}^n s_{i,j} t_{i,j}}{\sum_{i=1}^n \sum_{j=1}^n |t_{i,j}|} & \text{otherwise} \end{cases} \quad (5)$$

3.5 最善手の選択

1つの状態行列に対して各テンプレートに対する合致度が得られるので、最善手の選択手法にはさまざまな方法を考えることができる。図 3 (a) 単純に最大の合致度を持つものを最善手とする手法、同 (b) 食い違いのないテンプレート数が最多となるものを最善手とする手法、同 (c) 食



(a) 最大選択 (b) 最少違反選択 (c) 最大平均選択

図 3 複数の合致度から実際の着手を決定する手法

Fig. 3 Techniques to decide the next move from matching scores.

い違いのないテンプレートの中で合致度の平均をとり、その最大のものを選択する手法などである。本論文の 5 章の実験では、予備実験から (a) を用いることにした。

3.6 計算量

合法手数を k 個、探索深さ (先読み数) を d 、盤のマス数を n 、用いるテンプレート数を m_T とすると、提案手法の計算量は $O(k^d n^2 m_T)$ である。ただし、多くの場合盤上のすべての箇所を見る必要はないこと、あるノードで合致度が $-\infty$ のテンプレートはその子ノードでも合致度が $-\infty$ である場合が多く計算が不要であること、並列化が可能なこと、などの理由で実際の計算量はこれよりもはるかに小さくなる。

4. ぷよぷよとは

1991 年にコンパイル社から発売されたぷよぷよシリーズ (現在はセガ社が権利を保有) は、対戦の要素を加えたことでテトリスを凌ぐ落下型パズルゲームの代表として認知されており、現在でも多くの上級者たちが高レベルな対戦を繰り返している。ぷよぷよでは、適切な石の配置によって“連鎖”と呼ばれる状況を起こして相手を攻撃することが中心的な課題となる。双方が同時に攻撃した場合には、攻撃力の弱い側のみが被害を受けるため、ランダムな配石に合わせて「できるだけ効率良く長い連鎖を構成する能力」が「相手との駆け引き」と同様、重要となる。上級者同士の戦いでは 12~14 連鎖程度が組めることが必須とされている。ぷよぷよの既存研究としては、一般化ぷよぷよの NP 完全性 [3] など、数学的見地からの研究はいくつかなされているが、強くすることを目的とした研究はまだ不十分であり、AI の強さも上級者に及ばないことがアルゴリズム大会の結果より分かっている [2]。

囲碁や将棋と同じく、ぷよぷよにも“定型連鎖”と呼ばれるいくつかの定石形が存在する (図 1 (b))。そこで本論文では、3 章で提案した定石形配置法をぷよぷよの定型連鎖構成に適用し、これを従来型の木探索に基づく手法と比較することで有効性を示す。本章ではぷよぷよのルールを述べた後、従来の AI で用いられているポテンシャル最大化法について説明し、特に長い連鎖を目指すうえで従来法にどのような問題があるかを議論する。

4.1 ぷよぷよのルール

図 4 を用い、2.1 節で述べた問題クラスの各要素ごとにぷよぷよのルールを説明する。

- [プレイ人数] 1人のプレイヤーが1つの盤を持ち、通常2人でプレイする。連鎖 (後述) によって相手を攻撃する以外は相手の盤に邪魔をすることはできない。
- [盤とマス] 盤は通常横6縦13の2次元格子からなり (図 4 (a))、下方向に重力を持つ。左右端、上下端に位相的つながりはない。

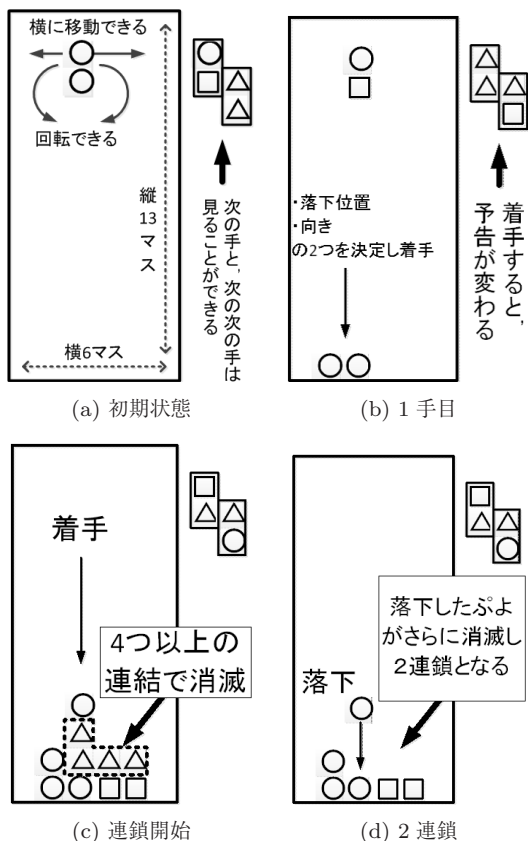


図 4 基本的なぷよぷよのルール
Fig. 4 The rules of Puyo-Puyo.

- [石] 4つのマスは、空であるか、1つの石（ぷよと呼ばれる）が占めるものとする。石には4つの種類（色）が存在する。他に、後述するおじゃまぷよと呼ばれる石も存在する。
- [配石] プレイヤーには、2つの石からなる配石（配ぷよ）が与えられる。石の種類はランダムに決められるが、2つ分先まで予告されており（図 4(a)）、現在操作するものを含め深さ3の探索が可能である。
- [着手] プレイヤーは、2つの石を回転および左右に移動し、落下させる（図 4(b)）。盤横幅が6の場合、最大で22通りの合法手がある。
- [状態遷移] 同じ種類の石が4つ以上上下左右に連結すると、その集団は消滅する（図 4(c)）。消滅した集団の上部にある石は重力に従い落下するが、それにより新たに4つ以上の連結が生じ消滅が起きた場合、これを連鎖と呼ぶ（図 4(d)）。

N 段階の落下と消滅が繰り返されたとき、それを N 連鎖と呼ぶ。原理的には最大で $[6 \times 13/4]$ で19連鎖が発生しうる。 N 連鎖を達成すると、およそ N の2乗に比例した攻撃力で相手を攻撃することができ、攻撃力に応じた数の「おじゃまぷよ」と呼ばれる特殊な石が相手の盤上に落下する（他に、5個以上の石を連結した場合にも若干攻撃力は上昇する）。おじゃまぷよは4つ連結しても消えず、上下左右に隣接するいずれかの（おじゃまぷよ以外の）石が

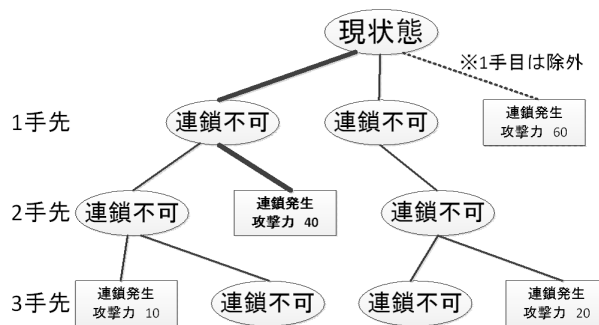


図 5 ポテンシャル最大化アルゴリズム
Fig. 5 Search tree of potential maximization search.

消えた場合にのみ消えるため、これが落下することは大きな妨害となる。双方が攻撃を行った場合には攻撃力の大きいほうがその差分だけ一方的に相手におじゃまぷよを落下させることができるため、大きい連鎖を構成する能力が本ゲームにおいて重要となる。

4.2 ポテンシャル最大化アルゴリズム

ぷよぷよのAIは長らく偶然に頼った連鎖構成法が主流であり、せいぜい4~5連鎖程度の攻撃を仕掛けてくるものがほとんどであった。アクションゲームということもあり、また上級者も少ない段階では市販ゲームとしてはこれで十分であったということだろう。アクション性を排除したぷよぷよの変形である「崩珠」では、1999年にポテンシャル最大化アルゴリズムを用いたAIが公開されており[1]、一部ルールに違いはあるものの10連鎖程度の攻撃を行うことで注目された。このアルゴリズムは以下の手順で、配石の予告されている3手先までを木探索し、葉ノードでの攻撃力が最も大きくなる手を選ぶ。

- (1) 着手可能な手を3手先まですべて列挙する。
- (2) 特別な場合を除き、1手目に石が消滅する手（連鎖を開始する手）を除外する。
- (3) 2, 3手先に連鎖を開始できる変化のうち、攻撃力が最大であるものを選択しそこに至る経路の1手目を着手する。

図5に木探索の例を示す。通常のぷよぷよでは現在操作するものを含めて3手先までの配石が分かっているため、自分の盤についてだけなら3手先の局面をすべて列挙することが可能である。基本的にはより長い連鎖を構成していくことを目指すため、1手目で連鎖を開始する手（つまり溜めてきた石を減らしてしまう手）は通常除外する。ただし、(a) 盤の残り空点が少なくなってこれ以上の連鎖数向上が見込めない場合、(b) 相手を倒すのに十分な連鎖数がすでにある場合、(c) 相手に攻撃されており反撃が必要な場合、(d) 連鎖数を向上させるために邪魔な石を部分的に消す場合など、特別な場合には選ぶこともある。一方で、3手先まで読んでも連鎖が生じない場合は4手目以降は読

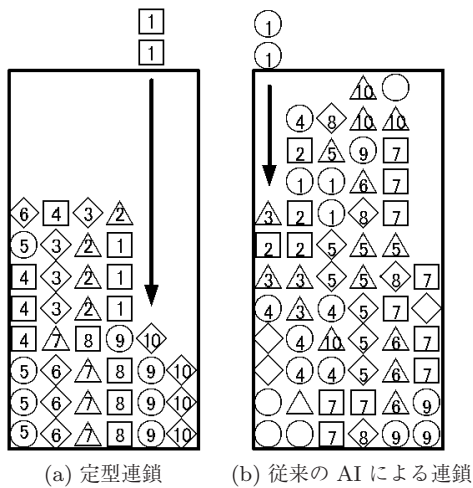


図 6 10連鎖の例。数字は連鎖で消えていく順序を表す。左の例では、□が消えることで△が落下し、2連鎖目を生じ、続いて◇、□、○が次々に消えることで最終的に10連鎖になる。右の例では、たとえば5連鎖目に9個の消しが生じ非効率である

Fig. 6 Examples of 10-chains. Smart one created by hand (left) and dirty one created by the program (right).

まず、攻撃力を0とする。これは、その先の配石が予告されていないために、10通りすべての可能性を考慮する必要があり計算量が 10×22 倍になること、仮にこれを短時間で読めたとしてもそこから、(1) 楽観的に最良の場合を選ぶのか、(2) リスク回避的に平均や最悪の場合を選ぶのかなど難しい判断を必要とするからである。結局、2手先・3手先に連鎖を開始できる中で最もその攻撃力が大きいものを選び、その1手目だけを選択することになる。状態遷移して次の配石が予告されたら、1回前の探索結果は破棄して再度探索する。

4.3 ポテンシャル最大化アルゴリズムの問題点

前節のアルゴリズムには2つの問題がある。

- (1) 盤面を効率良く使えず、連鎖数が頭打ちになる。
- (2) 大局的な目的に基づいて戦略的に連鎖を構成できない。

(1) について、ポテンシャル最大化では攻撃力が評価基準になるため、連鎖数を増やせない場合には5個以上の石を消すことで攻撃力を向上させようとする。これは短期的には攻撃力向上に役立つが、空間的な無駄を生みやすく、長い連鎖を作るのに不利な局面へと向かう傾向がある。図6(b)は従来型のAIによる10連鎖の例であるが、整然としてコンパクトな図6(a)に比べ、複雑・乱雑で大きなスペースを要し、これ以上の連鎖数の上昇が見込めないことが分かる。AIの組む連鎖が整然としていないことはある種の多様性・意外性による楽しさを生じさせうるものの、対人戦の練習・上級者の相手をするという意味では欠点である。

(2) の大局的な目的に基づいて連鎖が組めないことについて、人間とAIの対戦を例にあげて説明する。図7に、JAISTCUP [2]における、人間とAIの対戦における序盤

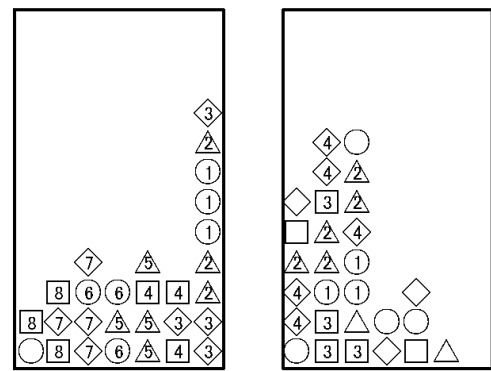


図 7 人間(左)対AI(右)序盤
Fig. 7 Human vs. AI (early stages).

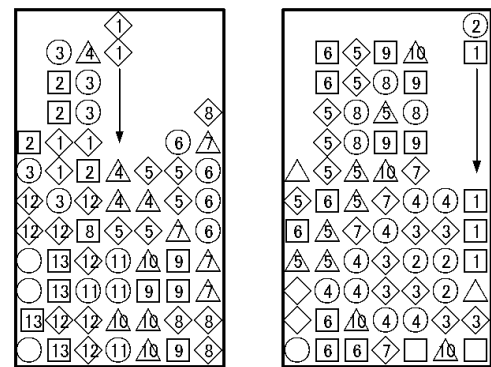


図 8 人間(左)対AI(右)終盤
Fig. 8 Human vs. AI (end game).

14手目までの一例を示す(盤面のサイズはぶよぶよとは異なり 7×11 である)。AI側は○を消すことで現在4連鎖がいつでも開始できる形である。一方人間側は8連鎖を行う目処がついているが実際の連鎖のためには4連鎖目と8連鎖目の□2つ、6連鎖目の○1つ、2連鎖目の△1つが足りない状況となっている。人間はこのように、配石に合わせて「こういう形の連鎖を組む」という想定図を思い浮かべて、それにあてはめていくことをしばしば行う。図5で示したポテンシャル最大化法によってつねに連鎖を開始できるように徐々に構築を行うAIではこれではできない。図8は図7に続き、連鎖を実際に開始する局面である。人間の連鎖は初級者でも紙上で目で追うことができるほど簡明であり、ほぼすべてのぶよを使って13連鎖となっている。一方でAIの連鎖は5連鎖目以降から非常に分かりにくく、最終的に10連鎖となっているのが不思議なほどである。これらの状況から、3章で提案した手法をぶよぶよにおける定型連鎖の構成に用いることは、空間の効率的利用および配石の戦略的活用につながり、結果として連鎖数の向上およびAIの強さの向上を導くと考える。

5. ぶよぶよにおける定石テンプレートの作成

着手決定の枠組みは3章で一般的に定義したとおりである。ぶよぶよという特定のゲームについて行わなければな

らないことは、目標とする定石形を定め、3.3節で説明した関連性行列の形で定義することである。78×78の行列を複数手作業で準備することは現実的ではないため、目標とする定石形を効率的にテンプレート行列化するために次の4つの段階を踏むことにする。

- (1) ラベリング表の作成
- (2) 真理値表の作成
- (3) 重み付け表の作成
- (4) テンプレート行列への変換

このうち手順(1)、(2)、(3)の一部と(4)は自動化できている。本論文の実験で用いたテンプレート(計23種類)を作るのに必要な時間はたかだか2時間程度であった。将来的にはより多くのテンプレートを作成あるいは棋譜から抽出するための自動化手順が必要かもしれないが[8]、78×78行列という響きほどには大きな人的時間コストを要していないことは強調しておきたい。

(1) ラベリング表の作成

まず、作成したい定型連鎖を、1回目に消える4つ(またはそれ以上)の石の集合、2連鎖目に消える石の集合のように区別して考える(図9(a))。これらそれぞれにABCなど大文字のラベルを割り振る。さらに、定型連鎖本体に隣接する境界部分には自動でabcなどすべて異なる小文字を割り振る。それ以外の部分は便宜的に0を割り振る(図9(b))。

(2) 真理値表の作成

次に、2つのラベルのついたマスが「同じ種類の石であるべき」「違う種類の石であるべき」「どちらでもよい」を表す真理値表(図10左)を作成する。この真理値表の多くの部分は、ラベリング表から自動生成される。たとえば、ラベルAのマスとラベルAのマスは明らかに同じ種類の石でなければならないし、ラベルAのマスとラベルBのマスは隣接しており、明らかに違う種類の石でなければならない。自明なルールにより定まる関係のほかに、手動で

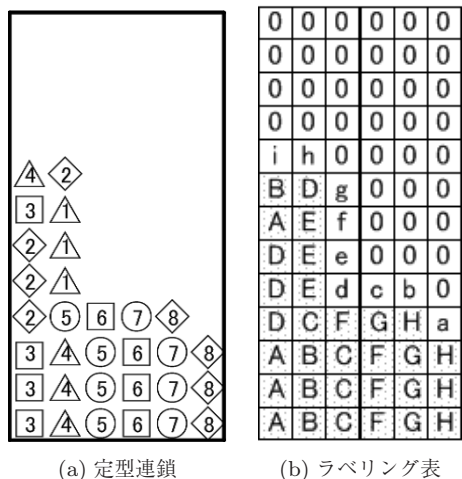


図9 ラベリング表の作成
Fig. 9 Labeling the cells.

関係を定めなければならない場合もある。たとえば、図9のAとCが同じ種類である場合、2連鎖目のD(◇2)が消えたあとに左端下から4段目にA(□3)が落下し、下側のAと右側のCを巻き込んで4連鎖で終了してしまうという、意図しない挙動を起こしてしまう。そこで、AとCは「違う種類の石であるべき」という手動の調整が必要になる(図11左)。

またBとiなどといったように端の小文字の部分は連鎖という観点でみればここは同色でもかまわない箇所だが、長い連鎖を構成するためには最小限の消滅数の4個消しで連鎖を組むことが好ましいため、ここでは異色という制約を課している。これは利用者の任意である。ほかにもたとえばdは一見Fと同色でもかまわないが、dとDとFが同色の場合には連鎖が失敗するためdとFなども異色とする制約を課した。

(3) 重み付け表の作成

定型連鎖を構成するために、要となる部分は優先的に配置していくことが好ましいことが経験的に分かっており、重み付け表はそれを指定するための表である(図10右)。まずはデフォルト値として、大文字ラベル(連鎖の本体)



図10 生成された真理値表と重み付け表
Fig. 10 The relationship matrix.

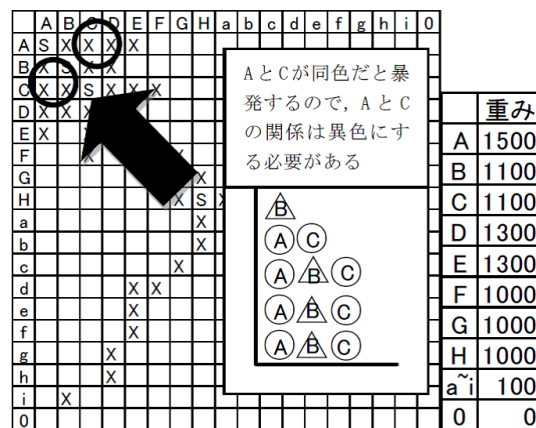


図11 修正された真理値表と重み付け表の例
Fig. 11 Required matrix modifications.

には 1,000, 小文字ラベル (境界) には 100, 0 ラベルには 0 が割り振られる. そのうえで, 手動の調整が行われる (図 11 右). 今回の例でいえば折り返しと呼ばれる色制約がきつくなる部分やそれとともない置きにくくなるであろう部分の重み付けを大きくしている.

(4) テンプレート行列への変換

最後に, ラベリング表・真理値表・重み付け表をもとに, 任意の 2 つのマスの関係性を計算し, 3.3 節で述べたようなテンプレート行列を自動生成する. すなわち, マス i, j に対するラベル l_i, l_j を求め, 重み w_{ij} を l_1, l_2 の重みの平均としたうえで, 真理値表が S なら $t_{ij} = w_{ij}$, X なら $t_{ij} = -w_{ij}$ とする. 平均ではなく最小値を使うことも有望である.

6. 実験

本章では, 提案手法の有効性を確認するために複数の実験を行う. 実装には, ポテンシャル最大化アルゴリズムや状態遷移則などのソースが公開されており AI 部のみの追加実装が可能な崩珠プラットフォーム [2] を用いた.

まず 6.1 節では, 3 つの異なるテンプレート群を対象にそれが完成するまでの手数を計測し, 人間の熟練者と比較する. 続いて 6.2 節では提案手法と従来手法を組み合わせることによって AI の連鎖構成能力やその結果としての攻撃力が向上したかどうかを単体および対戦によって検討する.

6.1 定石形完成手数の比較

提案手法の有効性を示すために, 提案手法を実装した AI が定型連鎖を完成させるまでの手数をぶよぶよ熟練者と比較した. ここでの定型連鎖の完成とは, 人間の場合は連鎖部分がすべて配置されること, AI の場合は合致度が 0.95 以上となることと定義した. 合致度 1 は境界部分も含めすべて配置された場合にしか達成されないため, 実際上連鎖部分がすべて配置される条件として 0.95 以上というものを予備実験により定めている.

テンプレート群としては, 「階段 8 連鎖 (10 種)」, 「挟み込み 8 連鎖 (12 種)」, 「ドミノ型 6 段 (1 種)」の 3 つを用いることとした. 前者 2 つは, ぶよぶよの典型的な定石形で, 折り返しと呼ばれる色制約の強い箇所の部分が微妙に異なるものを複数用意しランダムなツモでもロバストに対応できるようにしたものである. 最後の 1 つは, ぶよぶよには登場しないような配置パターンであり, 汎用性テストのために用いる. ドミノ型では左から 1・2 番目, 3・4 番目, 5・6 番目の石が同色になるように下から 6 行分が構成されたときを完成とする (離れているペア間に色制約はなく, 構成は困難ではない). それぞれのテンプレートの完成例を図 9 (a), 図 12 (a)~(f) に示す.

試行は 50 回, 毎回異なる乱数の種に基づく配石を与え

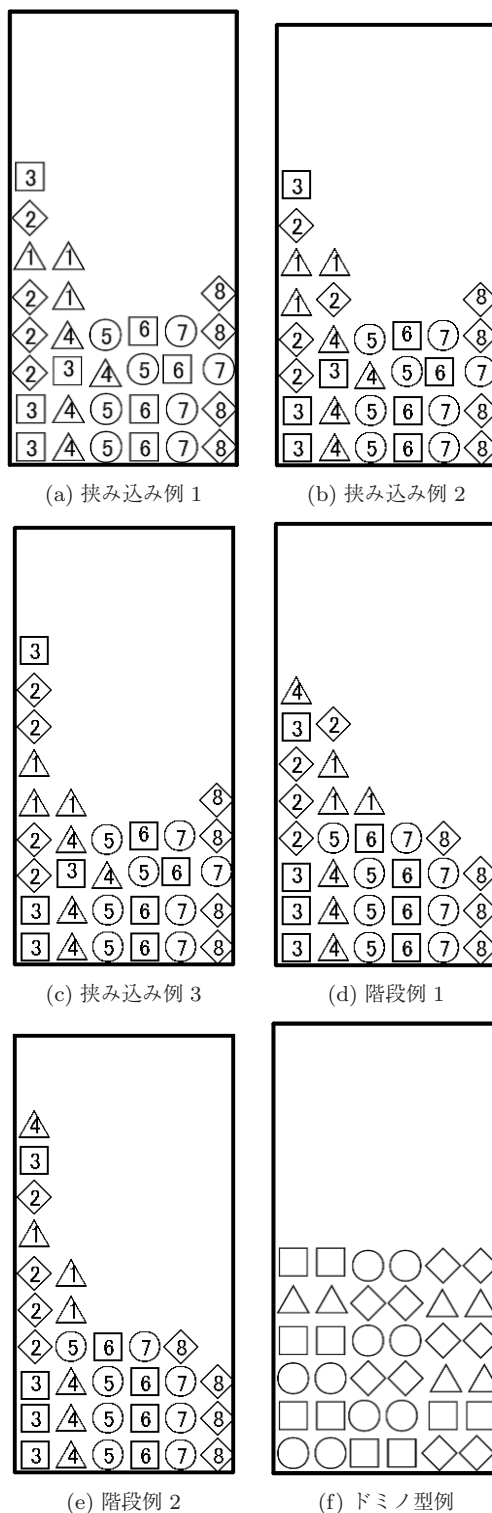


図 12 テンプレートの完成例

Fig. 12 Example of complete template.

るが, ぶよぶよ熟練者と AI にはそれぞれ同じ配石を与えて不公平さを抑制した. 8 連鎖には 32 個の石が必要であり, 最短でも 16 手を要する. ドミノ 6 段には 36 個の石が必要であり, 最短でも 18 手を要する. しかし配石の偏り具合によっては, 必要な種類の石がこないために, 途中で不要な石の消去を行わざるをえない状況など, 完成が大幅に遅れざるをえないこともあることに注意されたい. ある

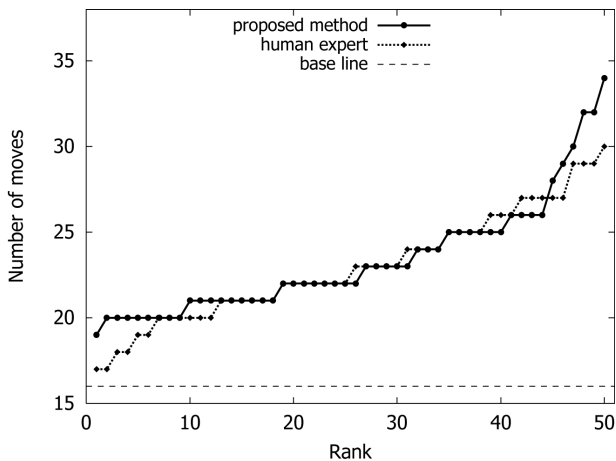


図 13 階段 8 連鎖の完成手順比較

Fig. 13 Number of actions required for executing an 8-chain (stair).

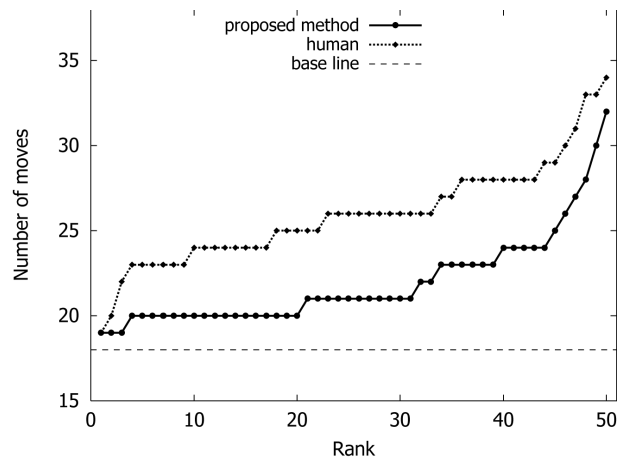


図 15 ドミノ型 6 段の完成手順比較

Fig. 15 Number of actions required to complete a 6-step (domino).

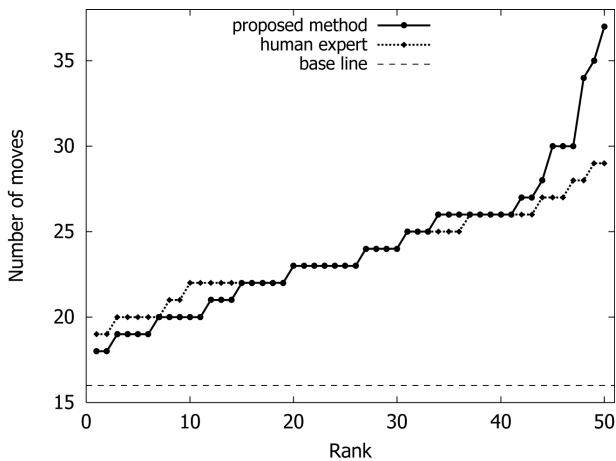


図 14 挟み込み 8 連鎖の完成手順比較

Fig. 14 Number of actions required for executing an 8-chain (key).

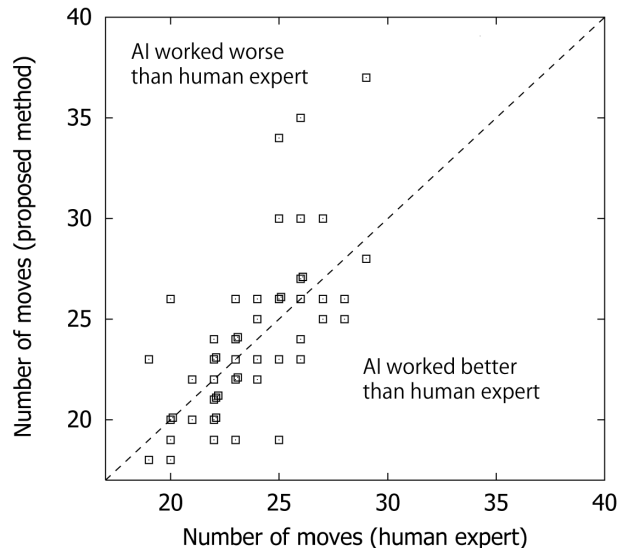


図 16 挟み込み 8 連鎖の場合の、同じ配石での完成手順比較

Fig. 16 Completion procedure comparison with the same seed for executing an 8-chain (key).

順番の配石について、最善の戦略を用いた場合の完成手数
の見積りは非常に困難であるため、人間の熟練者との比較
を行っている。

結果 1-1：各テンプレート群での完成手数比較

図 13, 図 14 はそれぞれ階段 8 連鎖, 挟み込み 8 連鎖
のテンプレートを用いた場合の, 完成までに要した手数を
少ない順に並べたものである。階段 8 連鎖の平均手数は
AI が 23.40 手, 人間が 22.98 手, 挟み込み 8 連鎖の平均手
数は AI が 24.06 手, 人間が 23.68 手と, 0.4 手ほど劣るも
のの, ほぼ遜色ない結果が得られている。ただし, 手数を
要したほうの 5 試行ほどは, 完成できないということはな
かったものの, 人間の場合に比べ大きく劣っており, 都合
の悪い配石が与えられた場合の柔軟な対応などでは人間の
熟練者に及ばないことが見て取れる。

一方で図 15, ドミノ 6 段のテンプレートでは, AI が人間
を大幅に上回る性能を示している。平均手数は AI が 21.96
手, 人間は 25.98 手であった。これはぷよぷよには登場し

ない形であるため, ぷよぷよ特有の勘や感覚のようなもの
が働かず, 単純な先読み能力で勝負せざるをえなくなるこ
とで, AI よりも大きく劣ってしまうのだと推測している。

図 16 は各試行ごと, つまり配石が同じ場合の, 人間の
完成手数を横軸, AI の完成手数を縦軸にプロットしたも
のである。全体に正の相関があり, 人間にとって難しい配
石の場合には AI にとっても難しいという関係があること
が見て取れる。

結果 1-2：テンプレート群の組合せによるロバスト化

図 13, 14 から, AI には苦手とする配石があり, 30 手
を超えるような場合もみられることが分かった。それぞれの
テンプレート群はすでに柔軟な形を許容するようにそれぞ
れ 10 種, 12 種用意されているが, さらにこれを組み合わせ
ることで苦手な配石を克服することができないかを実験
した。500 試行で 30 手を超えた場合の数を計測したとこ

る、階段 8 連鎖では 6.6%が 30 手を超過、挟み込み 8 連鎖では 9.6%が超過してしまったのに対し、組み合わせた場合にはこれは 3.4%に抑えられた。このことから、複数しかも本質的に異なるテンプレートを組み合わせることには一定の効果があると考えている。

結果 1-3：速度に関する考察

3.6 節で述べたとおり、本手法では 1 手を着手するために最大 223・782・テンプレート数の計算量が必要である。C#, シングルスレッド、CPU は Intel Core i5-2310 (2.90 GHz) の環境で 1 手あたりの平均探索時間を計測したところ、階段 8 連鎖の場合は 0.63 秒、挟み込み 8 連鎖の場合は 0.47 秒、組み合わせた場合は 0.71 秒であった。これは人間の思考時間と比べても十分短く、問題にはならないことが分かった。組み合わせた場合の探索時間が組み合わせる前のものの単純な合計よりも大幅に抑えられているのは、本質的に異なるテンプレート群であるために、序盤を過ぎるとどちらかのテンプレート群以外には初手からマッチしなくなるためである。

結果 1-4：読み深さの変更

ゲームによっては、計算時間が許容できないほど長くなる可能性もありうる。その場合は、仮に配石が予告されており深い手数先の読みが可能であっても、一定の深さで探索を打ち切らなければならないこともありうる。そこで、そのような場合の性能の劣化度合いの参考とするために、あえて次の手のみ（深さ 1）の探索を行って、結果 1-1 の深さ 3 の場合と平均手数を比較してみる。

階段 8 連鎖は、3 手読みが平均 23.40、1 手読みでは 24.84 となった。挟み込みではそれぞれ 24.06 と 26.96、ドミノ型では 21.96 と 24.10 であり、やはり平均して数手の劣化が見られる。これが致命的かどうかはゲームに依存するが、深さ 3 以上でない本手法がまったく機能しないというようなことはないことが確認できた。

6.2 ポテンシャル最大化法との組み合わせによる連鎖生成能力の検証

提案手法は、8 連鎖までの部分を空間的に効率良く作成することができる。よって、定型部の完成後に、従来型の連鎖構成法を組み合わせるより連鎖を延ばせば、最初から従来型の方法を用いる場合に比べて最終的な連鎖数・攻撃力が向上することが見込まれる。攻撃力（おじゃまぶよの落下数）は連鎖数を N 、 i 連鎖時に消去したぶよの数を c_i としたとき式 (6) で求めることができる。実戦では攻撃力 30 程度の差があれば事実上勝敗が定まる。

$$3N(N - 1) + \sum_{i=1}^N i(c_i - 4) \quad (6)$$

以下の実験では、ゲーム開始時には定石形を目指し、合致度が 0.9 を超えた時点でポテンシャル最大化法に切り替え

表 1 連鎖生成能力の比較

Table 1 Comparison of the ability for chain generation.

	組合せ AI	ポテンシャル最大化 AI
平均連鎖数	10.99	8.80
攻撃力	438.9	301.0
連鎖開始手数	41.9	41.2

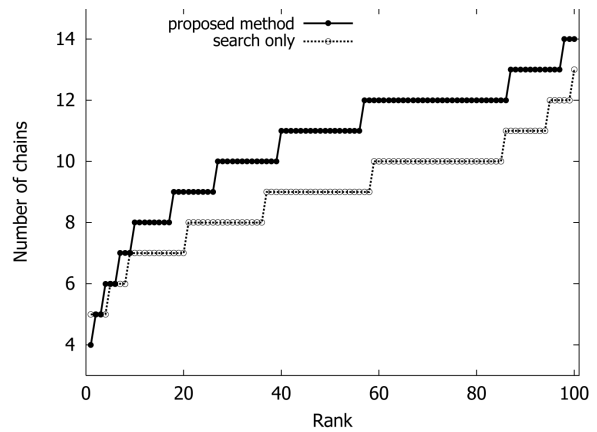


図 17 提案手法と従来手法の連鎖数比較

Fig. 17 Comparison of chain length by the proposed method and by search only.

ると非常に単純な組合せ手法を用いることにする。テンプレート群には前節で用いた階段 8 連鎖のものをを用いた。

結果 2-1：単体での連鎖数・攻撃力・攻撃のタイミング

まず、1 人ゲームでどこまで連鎖を延ばせるかを「組み合わせた AI」「ポテンシャル最大化のみの AI」について計測した。乱数の種は毎回変え、ただし両者には同じ配石を与えている。表 1 は 500 ゲームした場合の、連鎖数・攻撃力（相手に降らせるおじゃまぶよの数）・最終的な攻撃開始までの手数、の平均値を表したものである。攻撃開始のタイミングはほとんど同じであるにもかかわらず、連鎖数で 2 以上、攻撃力で 100 以上の大きな差をつけて、提案手法と従来手法の組合せによる AI が優れている。図 17 に、それぞれの AI の 100 試行分について、連鎖数を少ない順に並べたものを示す。全体的には提案手法が優れているが、1 割ほどの例では 8 連鎖すら成功していない。これは、いったん 8 連鎖が完成したものの、その後のポテンシャル最大化アルゴリズムの実施中に都合の悪い配石がきたことで自滅してしまった場合が多いことを確認している。これはどちらかといえばポテンシャル最大化アルゴリズムのチューニングに関わる部分である。図 18 に連鎖完成の一例を示す。完成例では定型部分の 3 列 4 行目の 13 □になるであろう部分が 7 □になっているが、これはポテンシャル最大化アルゴリズムの読みの中で変化したものである。

結果 2-2：提案手法と従来手法の対戦

最後に、提案手法 AI（定石形 + ポテンシャル最大化）を従来手法 AI と対戦させた。500 戦した結果、提案手法側の 374 勝 126 敗（.748）となり、Elo レーティングで 190



図 18 連鎖完成図例

Fig. 18 Example of a 15-chain created by AI.

点ほどの大幅な向上が確認できた。

本論文の主眼は落下型パズルゲーム一般における関連性行列を用いた定石形配置法の提案であるが、定石形を用いることでぷよぷよという高度なゲームでの AI の大幅な性能向上を達成したことで、その有用性と他のゲームへの適用の有望さを示すことができたと考えている。純粋にぷよぷよの強さを考えた場合には、連鎖構成法だけでなく駆け引きが重要であり、それは別の研究として進めていきたい。

7. まとめと今後の課題

本論文では、落下型パズルゲームを対象クラスとして一般化し、そこでしばしば用いられる概念である定石形をコンピュータに利用させる方法を提案した。盤の 2 つのマスにある石の種類が同じか異なるかが重要な落下型パズルゲームの特徴に留意し、関連性行列という形で効率的に状態と定石形を表現することができることを示した。さらに、落下型パズルゲームの代表格であるぷよぷよを例に、定石形のデータを手作業で効率的に作成するための半自動化手順を提案し、実際に 23 種類の定石形を約 2 時間で作成できることを示した。ぷよぷよを用いた性能評価では、定型連鎖の構成効率が熟練者に迫る効率であること、従来の連鎖構成法であるポテンシャル最大化法と組み合わせることで、従来手法単体の場合よりも良好な性能が得られることを示した。今後は、多くのテンプレート行列を効率的に構成する方法、棋譜からのテンプレート自動抽出、ロバストな最善手選択法の考案、機械学習による評価関数の調整などを提案していきたい。またぷよぷよに限って言えば、駆け引き部分を洗練させることで、人間のチャンピオンに挑戦できる水準にもっていくことを目指す。

参考文献

[1] PojeMaster, available from (<http://www.vector.co.jp/soft/win95/game/se095051.html>).
 [2] JAIST CUP 2011, available from (http://www.jaist.ac.jp/jaistcup2011/poje_details.html).

[3] 松金輝久, 武永康彦: 一般化ぷよぷよの NP 完全性, 数理解析研究所講究録 (2005).
 [4] Gelly, G., Wang, W., Munos, R. and Teytand, O.: Modification of UCT with Patterns in Monte-Carlo GO, Technical Report RR-6062, INRIA (2006).
 [5] 清 慎一, 川嶋俊明: 「局所パターン」知識主導型の囲碁プログラミングの試み, 第一回ゲームプログラミングワークショップ予稿集 (GPW'94), pp.97-104 (1994).
 [6] 保木邦仁: 局面評価の学習を目指した探索結果の最適制御, 第 11 回ゲームプログラミングワークショップ, pp.78-83 (2006).
 [7] Puterman, M.L.: *Markov Decision Processes*, John Wiley & Sons, New York (1994).
 [8] 中家啓文, 飯田弘之, 小谷善行: 棋譜データによる定跡ファイル作成手法, 全国大会講演論文集, 第 52 回平成 8 年前期 (2), pp.63-64 (1996-03-06).
 [9] 中村貞吾: 着手記号列の出現頻度に基づく囲碁棋譜からの定型手順獲得, 情報処理学会論文誌, Vol.43, No.10, pp.3030-3039 (2002).
 [10] Wikipedia, 落ち物パズル, 入手先 (<http://ja.wikipedia.org/wiki/落ち物パズル>), (参照 2012-02-20).



富沢 大介 (学生会員)

1988 年生。2010 年東京工業高等専門学校専攻科電気電子工学科卒業, 同年北陸先端科学技術大学院大学情報科学研究科博士前期課程入学。現在に至る。



池田 心 (正会員)

1975 年生。1999 年東京大学理学部数学科卒業, 2000 年東京工業大学総合理工学研究科知能システム科学専攻博士前期課程修了, 2003 年同博士後期課程修了, 博士 (工学)。同年京都大学学術情報メディアセンター助手, 2010 年北陸先端科学技術大学院大学准教授。ゲーム, 進化計算, エージェントシミュレーション, パズル等の研究に従事。計測自動制御学会, 進化計算学会各会員。



シモン ビエノ

1980 年生。2004 年グランゼコール Ecole Centrale de Nantes 大学大学院 (フランス) ロボット制御理論修士課程修了, 2011 年 Lille 大学 (フランス) 計算機科学博士後期課程修了, 博士 (Computer Science)。2012 年北陸先端科学技術大学院大学研究員。