着手の不可逆的性質を利用したDFPN探索による Yonmoqueの求解

中川 憲[1,a)] 荒井 光[2,b)] 保木 邦仁[3,c)] 村松 正和[1,d)]

概要：Yonmoque とは、1996 年にギフトボックス社の山本光夫によって考案された二人零和有限確定完全情報ゲームである。本稿は、このボードゲームの求解を目的とする。この解析のための Depth First Proof Number(DFPN) 探索アルゴリズムをベースとした改良手法について報告する。本研究では、ゲームルール上で許可されている着手の不可逆的性質に着目する。この性質を活用することで、Graph History Interaction 問題として知られている誤評価問題を回避し、探索結果の確実性を保証しながら解析を行うことが可能となった。結果として、Yonmoque が先手番必勝であることを証明した。

キーワード：Yonmoque、着手の不可逆的性質、Depth First Proof Number 探索、Graph History Interaction 問題

Yonmoque is solved by DFPN search exploiting the irreversible property of moves

Nakagawa Ken[1,a)] Arai Hikaru[2,b)] Hoki Kunihito[3,c)] Muramatsu Masakazu[1,d)]

Abstract: Yonmoque is a two-player zero-sum finite game with perfect information invented in 1996 by Yamamoto Mitsuo of Gift Box Co.Ltd. We have solved this game by developing a technique on the basis of Depth First Proof Number (DFPN) search. In the technique, we focused on the irreversible property of moves, and carefully avoided the graph history interaction problem which destroys the validity of the DFPN search. We have succeeded the analysis of the initial board, and concluded that the blue player (the first player) can always win in this game.

Keywords: Yonmoque, Irreversible property of a moves, Depth First Proof Number Search, Graph History Interaction Problem

1. はじめに

近年、コンピュータを用いたゲームの求解例が相次いで報告されている。例えばチェッカーは 2007 年に Schaeffer ら [6] によって解かれ、また田中哲朗によってボードゲームの一種である「シンペイ」[12]、「どうぶつしょうぎ」[11] などが完全解析されている。

本研究は、二人零和有限確定完全情報ゲームの一種である Yonmoque [13] の求解を目的とする。すなわち、初期局面からの全ての応手を考慮した必勝手順を求める。Depth First Proof Number (DFPN) 探索 [1], [3], [9] を基本とす

1 電気通信大学大学院 情報理工学研究科 情報・通信工学専攻
Department of Communication Engineering and Informatics, Faculty of Informatics and Engineering, The University of Electro-Communications
2 電気通信大学 情報理工学部 情報工学科
Department of Computer Science, Faculty of Electro-Communications, The University of Electro-Communications
3 電気通信大学先端領域教育研究センター
The Center for Frontier Science and Engineering, The University of Electro-Communications
a) n1131083@edu.cc.uec.ac.jp
b) ha911119@jed.uec.ac.jp
c) hoki@cs.uec.ac.jp.
d) muramatu@cs.uec.ac.jp

る手法を用いるが，Yonmoque のゲームの性質を用いて探索結果の確実性を保証する解析手法の提案を行う。その結果、Yonmoque は先手必勝であることを証明するとともに、考えられる初手全てを先手必勝、後手必勝の２つに分類した。

2. Depth First Proof Number 探索

DFPN 探索アルゴリズムとは、AND/OR 木と呼ばれるゲーム木において定義される**証明数**および**反証数**を利用してルートノードの値を反復深化深さ優先探索で求めるアルゴリズムである。本節では、本研究で実装した DFPN 探索アルゴリズムと関連する基礎知識について簡単に説明を行う。

2.1 AND/OR 木

AND/OR 木とは、詰め将棋などの詰み手順探索によく用いられている木構造の一種である [8]。AND/OR 木を構成するノードは、**AND ノード**と **OR ノード**の 2 種類に区別され、各々のノードは**真**または**偽**のいずれかの**値**を持つ。2 人ゲームを解く場合では、攻め手の勝ちとなる局面を**真**、負けを**偽**として**値**を与える。子ノードが存在する内部ノードの**値**は以下のように定義される。

- AND ノードが真 ⇔ 子ノードが全て**真**
- OR ノードが真 ⇔ 子ノードのいずれか 1 つ以上が**真**

2.2 証明数/反証数

証明数と**反証数**とは、AND/OR 木の各ノードにおいてノードの計算コストの尺度を示す評価値の一種である [1]。**証明数**は”そのノードが**真**の値であること”を証明するために最低限評価が必要な子ノードの個数、**反証数**は”そのノードが**真**の値でないこと”を証明するために最低限評価が必要な子ノードの個数を表している。現在までに展開されているゲーム木における葉を**末端ノード**と言い、そのうちゲームが終了して勝敗が決定したノードを**終端ノード**と呼ぶ。終端ノードは必ず末端ノードであるが、逆は必ずしも成り立たない。**終端ノード**における**証明数** $p(n)$ と**反証数** $d(n)$ は表 1 のように定義する。

表 1 終端ノードにおける定義

	n の**値が真**	n の**値が偽**
証明数 $p(n)$	0	∞
反証数 $d(n)$	∞	0

終端ノードでない**末端ノード** における**証明数** $p(n)$ と**反証数** $d(n)$ はそれぞれ 1 と置く。

末端ノードでないノードを **内部ノード** と呼ぶ。**内部ノード**については、ノード n の子ノードの集合 $C(n)$ を用いて**証明数** $p(n)$ と**反証数** $d(n)$ を表 2 のように定義する。

表 2 内部ノードにおける定義

	n が AND ノード	n が OR ノード
証明数 $p(n)$	$\sum_{i\in C(n)} p(i)$	$\min_{i\in C(n)} p(i)$
反証数 $d(n)$	$\min_{i\in C(n)} d(i)$	$\sum_{i\in C(n)} d(i)$

内部ノードにおいては、ゲームの詰みやチェックメイトなどの存在を探索する際、詰みを狙う攻め手側（OR ノード）では有効な攻め手（子ノード）が 1 つでも存在すれば良いのに対して、それを防ごうとする受け手側（AND ノード）では全ての応手（子ノード）が無効であると示さなければならない。

2.3 DFPN 探索

DFPN 探索アルゴリズムは、証明数と反証数の概念を用いて、子ノードの中でより短時間でルートノードの値の評価が得られると予想されるものを優先的に展開する深さ優先アルゴリズムである。証明数または反証数の値がより小さいノードほど、より少ない時間で値の評価が得られるものと期待できる。このアルゴリズムでは以下の条件を全て満たす AND ノード n を有力な候補として展開を行う。

- ノード n が持つ証明数は、兄弟ノードの中で最小である
- ノード n が持つ証明数は、親ノードに与えられた計算リソースを上回らない
- ノード n を除く兄弟ノードの反証数の合計は、親ノードに与えられた計算リソースを上回らない

また OR ノードでは、AND ノードの条件の証明数と反証数を入れ替えた条件を満たすノードについて展開を行なう。

この条件に基づいて各ノード n について次式の閾値 $P(n)$ および $D(n)$*1を定義し、**展開条件** $P(n) > p(n)$ 且つ $D(n) > d(n)$ を定める。ここで、n_{parent} はノード n の親ノード、n_2 は $C(n_{\mathrm{parent}})$ の中で 2 番目に証明数 (OR ノードの場合は反証数) が小さなノードである。

- n が AND ノードの場合

$$P(n) = \min\left(p(n_2)+1, P(n_{\mathrm{parent}})\right) \quad (1)$$
$$D(n) = D(n_{\mathrm{parent}}) - d(n_{\mathrm{parent}}) + d(n) \quad (2)$$

- n が OR ノードの場合

$$P(n) = P(n_{\mathrm{parent}}) - p(n_{\mathrm{parent}}) + p(n) \quad (3)$$
$$D(n) = \min(d(n_2)+1, D(n_{\mathrm{parent}})) \quad (4)$$

ただし、n がルートノードである場合には $P(n) = D(n) = \infty$ とする。

あるノードがこの**展開条件**を満たしたときにのみ、そのノードの子ノードを展開して探索を行うものとする。

*1 この閾値 $P(n)$ と $D(n)$ はノード n に許可された計算リソース量を表している

3. Yonmoque

Yonmoque[13] とは、ギフトボックス社の山本光夫によって考案された二人零和有限確定完全情報ゲームの一種である。本研究はこの Yonmoque を題材とした。この章では Yonmoque のゲームルールの説明を行う。

3.1 ルール

本ゲームの主旨は、1 枚のゲーム盤（図 1）と専用の駒 12 個を用いて 2 人のプレイヤーが勝ち負けを競い合うことである。Yonmoque の駒はリバーシの駒のように片面が青色、もう片面が白色となっている。

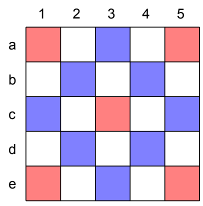

図 1 ゲーム盤

初期状態では、青側のプレイヤーと白側のプレイヤーはそれぞれ駒 6 個を自らの**手駒**として所持するものとする。青側のプレイヤーを先手番として交互に着手を行い、後述の勝敗条件が満たされるまでこれを繰り返す。次にルール上で認められている合法な着手と勝敗条件について述べる。

3.1.1 合法手

各プレイヤーは、自分の手番において次の 2 種類の着手のいずれかを 1 回行う。**パス**は認められない。

配置着手

手駒から 1 つの駒を自分の色を上に向けて盤上の任意のマスへ配置する。**手駒**がない場合や、配置先のマスへ既に他の駒が置かれている場合にはこの着手を行うことができない。

移動着手

盤上に置かれている自分の色の任意の駒を 1 つ移動させる。駒の移動できる範囲は隣接する上下左右斜めの 8 方向のマスであるが、移動させたい駒の置かれているマスの色が自分の駒の色と一致する場合に限り、同一の色のマス上を斜めに複数マス移動できる (図 2)。ただし、移動過程に他の駒が存在する場合は、それらを飛び越えるような移動は認められない。更に、駒の移動先において、相手の駒を自分の駒で挟む状態となった場合、挟まれた相手の駒を全部裏返して自分の駒とする。

また、元々の Yonmoque には無いルールであるが，ゲームが無限に続くことを防止するため、その対局において過去に現れた局面と**同一局面**が現れるような着手を**禁じ手**とする (同一局面反復の禁止)。ただし**同一局面**とは、現在の手番の色、盤上の駒の配置、各々のプレイヤーの手駒の個数が全て一致することと定義する。

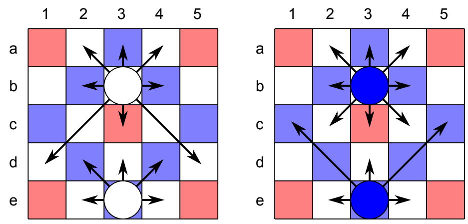

図 2 駒の移動可能範囲

3.1.2 勝敗条件

前述の着手を交互に繰り返し、**移動着手**が行われた結果、上下左右斜めのどれかの方向に 4 つの自分の色の駒を並べた側のプレイヤーを**勝ち**とする。ただし、**移動着手**で 5 つの自分の駒が並んだ場合には並べた側のプレイヤーの自滅**負け**とする。一度の**移動着手**で 4 つの駒が並ぶ列と 5 つの駒が並ぶ列が同時に出現した場合、自滅**負け**を優先する。また、自分の手番であるにも関わらず、前項で述べた合法な着手を行うことができない場合にもそのプレイヤーの**負け**とする。**配置着手**が行われた手番では勝敗は付かない。

4. 課題

4.1 Graph History Interaction 問題

広大な探索空間を持つ問題を対象として解析を行う上で transposition table を用いることは効率的な探索を行う上で必須と言えるが、しばしば探索結果の破綻を引き起こすことが知られている。Graph History Interaction(GHI) 問題と呼ばれるこの問題は、1983 年に Palay [10] によって指摘された。あるノード n の下位のノードの展開を繰り返した結果、一見 n と同じ状態を持つノードが再度現れることがある。これらの 2 ノードを同一のノードと見なすと GHI 問題が発生する。以下に例を用いて GHI 問題の説明を行う。

図 3 では、ルートノード a から順々に展開を行い、ノード f へと至る経路 $a \to b \to c \to d \to e \to f$ と、f の**同一局面**ノード f' へ至る経路 $a \to g \to h \to i \to j \to f'$ を表している。ここで、ノード f は同一局面反復の禁止ルールにより**このノード以下にはノード** a, b, c, d, e **と同一局面のノードが現れてはならない**と言う禁じ手の情報を潜在的に保持していることに注目する。一方、これに対してノード f' の禁じ手は a, g, h, i, j である。局面の状態は等しいノード f と f' であるが、この差異を無視して前章で述べたようにノード f とノード f' を同一のノードとして扱って transposition table への登録を行うと、GHI 問題の原因となる。

例えば、ノード f の値が真 (先手の勝利) と判定されたとする。このとき、ノード f では後手側に c と言う逃げ道

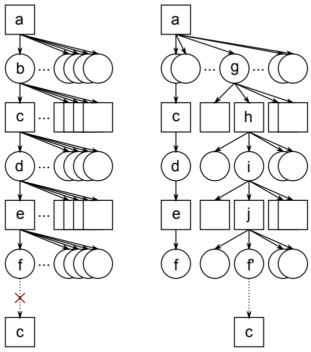

図 3　GHI 問題

が存在しない。一方、ノード f' では c と言う変化を選択できる自由がある。従って後手側の応手の制限が異なるため、ノード f の解析で得られた必勝手順をノード f' へ必ずしも同様に適用できるとは限らないのである。

ここでの GHI 問題の本質は、局面が潜在的に保持している情報が辞書へ登録する際に失われることによって引き起こされる問題と言える。

4.2　従来の解決手法

GHI 問題を解決する代表的な従来手法としては、以下のものがある。

ナイーブな手法 [4], [5]
その局面に至るまでの手順情報のすべてを各ノードに情報として与え、手順が異なる局面をそれぞれ区別する手法。この手法の欠点としては、まず手順情報を全て記憶することによるメモリリソースの増大が挙げられる。さらに、手順が少しでも違う局面を全て別個の局面として扱うために辞書の再利用性が著しく低下してしまう。これにより、初期局面や初期の手順が異なる問題について transposition table を共有して使用できない。

手順のハッシュを用いた手法 [4], [5]
ナイーブな手法と同様に、Zobrist ハッシュ [14] などのハッシュを用いて各ノードへ手順情報を格納する手法。Zobrist ハッシュとは、着手手順を基にして得られるハッシュの一種である。一般的な手法であるが、低確率ながらもハッシュの衝突によって誤答が発生する可能性が存在する。また、メモリリソースの問題を解決できるが、再利用性の問題を解決できない。

再探索手法 [7]
サイクルを含むノードとそうでないノードを区別する。サイクルを含まないノードの探索結果は信頼できるものとし、サイクルを含むノードの探索結果ではある程度の誤答を許容する。誤答を許容した解については辞書を用いずに再探索を行い、改めて真偽を確かめる。

上 2 つの手法は、いずれも計算リソースの問題を抱えていたり、解析結果の確実性を保つことができない欠点があるので、本研究の目的である求解には不向きである。再探索法に関しては、アルゴリズムが複雑であり、Yonmoque の解析においてはより簡便な方法が有効である。次節ではこの提案手法について解説する。

5.　提案手法

5.1　着手の不可逆的性質

ある着手 M によって**局面の状態**が α から β へと遷移したとする。ここでの**局面の状態**とは、盤上の駒の配置、手駒の個数、現在の手番のことと定義する。ここで、同一反復禁止を除き、ルールで許可されているいかなる操作を複数回行っても状態 β から α への遷移ができないとき、**着手 M は不可逆的性質を持つ**と定義する。

不可逆的性質を持つ着手の具体例として、本研究の題材である Yonmoque の着手の例を図 4 へ示す。図左の状態 α における青番の手駒は 3 個なのに対し、着手 M' 後の状態 β では 2 個である。Yonmoque のルールでは盤上にある駒の色は目まぐるしく変化する一方、手駒を増加させるような着手は存在しないため、状態 β の局面から以後いかなる着手を繰り返しても状態 α への遷移は不可能である。したがって、図の着手 M' は不可逆的性質を持っていると言える。

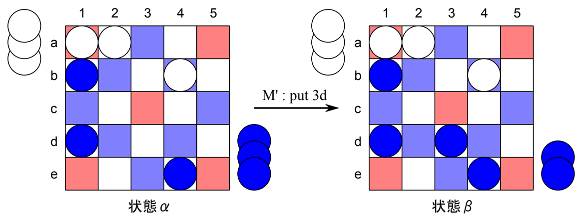

図 4　不可逆的性質を持つ着手の例

一般に、Yonmoque において配置着手は不可逆的性質を持つ。

その他の二人零和有限確定完全情報ゲームにおける不可逆的性質を持つ着手の例としては、チェスのポンの前進や敵駒の捕獲、リバーシの任意の着手などが挙げられる。

5.2　不可逆的性質を持つ着手の GHI 問題における性質

ある局面 n に至るまでに現れた局面の集合を $H(n)$ とする。$H(n)$ の中で n から（同一局面禁止のルールを無視して）到達可能な局面の集合を $\bar{H}(n)$ とする。

定理 5.1　不可逆的性質を持つ着手により生成された局面 n では、$\bar{H}(n) = \emptyset$ である。

証明. $\bar{H}(n)$ が空でないと仮定し、$a \in \bar{H}(n)$ を一つとる。すると、同一局面禁止のルールを無視すれば、n から a へ到達することができ、さらに a から n の直前の局面 $\mathrm{prev}(n)$ に到達することができる。これは $\mathrm{prev}(n)$ から n への着手が不可逆的性質を持つことに矛盾する。 □

系 5.1　不可逆的性質を持つ着手により生成された局面 n と n' が同一の局面であるとき、n の真偽値 $= n'$ の真偽値である。

GHI 問題を回避して正確な解を得るための研究は岸本らの詰め将棋に関する論文 [7] を始めとして、数多く行われているが [2], [4], [5]、あまりに単純なためか、この性質については触れられていないようである。

Yonmoque においては、**配置着手** はすべて不可逆的性質を持つ。そのため、**配置着手** 直後の局面を「正しい」結果を持つものとして transposition table へ登録することとする。また、結果を求めて transposition table を参照するのも**配置着手**直後のみとする。

特に Yonmoque の対局においては、序盤に同性質を有す**配置着手**が多く現われ、終盤に**移動着手**が多く現われる傾向がある。このことから、この手法を Yonmoque へ用いると、ルートノードに近い上位のノードの情報を数多く transposition table へ登録できるため、この手法がより有効に働くだろうと予想できる。

5.3 実験用アルゴリズム

着手の不可逆性を利用する transposition table として以下のものを作成して 2 章の DFPN 探索へ適用し、実験を行なった。

5.3.1 識別子とハッシュ

盤上の状態を保持する局面 S の**識別子**を下図のように定義する。

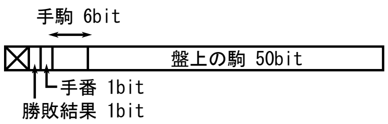

図 5　局面の識別子

この**識別子**から各 bit に対応する乱数値を用いて Nbit のハッシュを生成し、これを 2^N のサイズを持つ transposition table における局面 S のハッシュ値 $hash(S)$ とした。

5.3.2 transposition table

以下の 3 種類の transposition table を用いる。

証明テーブル

不可逆的性質を持つ着手の直後のノード n で、$p(n) = 0, d(n) = \infty$ を満たす内部ノード n の識別子、**コスト値**を保持する。探索の過程で不可逆的性質を持つ未知のノード x が現れた際、$hash(x)$ を用いてテーブルの参照を行い、識別子が一致するならばノード x とノード n は**等価である**として $p(x) = 0, d(x) = \infty$ を得る。

反証テーブル

不可逆的性質を持つ着手の直後のノード n で、$p(n) = \infty, d(n) = 0$ を満たす内部ノード n の識別子、**コスト値**を保持する。探索の過程で不可逆的性質を持つ未知のノード x が現れた際、$hash(x)$ を用いてテーブルの参照を行い、識別子が一致するならばノード x とノード n は**等価である**として $p(x) = \infty, d(x) = 0$ を得る。

参考テーブル

以上 2 つの条件を満たさない全ての内部ノードの証明数、反証数、識別子、**コスト値**を保持する。原則として、このテーブルが保持する値をゲーム木のノードへと適用することは行わない。同じ証明数または反証数を持つ子ノードが複数個現れた際の最有力ノードを決定する手段として利用する。

以上の条件を満たすノード n について、新たにテーブルへ情報を記憶する際、既に同じハッシュ値を持つノード m の情報がテーブル上に格納されていた場合（**衝突**）には、よりコスト値が大きなノードを優先するものとした。ただし、ノード n の**コスト値**を “ノード n の下位に存在するノードの総数” と定義した。

6. 実験

6.1 実験内容

まず初期盤面から 4 手配置着手を行った局面における必勝必敗を得ることを目的として次の実験を行った。

初期状態から 4 手配置着手を行った局面として全 75,900($= {}_{25}C_4 \cdot {}_4C_2$) 局面を用意し、これらから回転および反転によって同じ局面と見做すことができるものを除外した 9,664 局面の全てを対象として、前節のアルゴリズムを用いて探索を行った。この操作は、ルートノードからある程度進んだ子節点からの探索を個別に行うことで、並列的に探索を行うと共に、探索に必要なメモリリソースを抑えるために行ったものである。この実験には CPU Intel(R) Xeon(R) E31275, 物理メモリ 16.0GB を始めとした計算機 4 台を使用した。

6.2 実験結果

この解析の結果、初期局面から 4 手進んだ状態の全 9,664 局面のうち先手番必勝となったのは 8,953 局面、後手番必勝となったのは 711 局面であった。これらの結果を基に深さ 4 の全展開木を作成して得られた、表 3 へ初手青番による配置着手 [1a] から 3 手目までの先手番必勝手順について示す。表 3 において、たとえば最上段の 1a→1b→1d,2a,2b,2c,2d,3a,3c,3e,4a,4b,4e,5b,5c,5d は、先手初手 [1a] 後手 [1b] の局面に対し、3 手目先手番が [1d] または [2a]、[2b]、· · ·、[5d] のいずれかの着手を行った局面がすべて先手番必勝であることを示している。このとき、4 手目の後手番は配置着手ならば先手番必勝は覆らない。4 手目に移動着手を含む場合については、次節で説明する。

初手 [1a] 以外の着手を行なった場合の解析結果は付録に

示す。先手青番が [1b] および [2c]、[3c] への着手を行なった場合は青番の必勝 (付録 A.1)、これら以外の着手では白番の必勝 (付録 A.2) となる解析結果が得られた。ただし 4 手目までは移動着手を含まないと仮定している。

表 3 初手 1a における先手番必勝の全着手候補

初手	2 手目	先手番必勝となる 3 手目の候補
1a	1b	1d,2a,2b,2c,2d,3a,3c,3e,4a,4b,4e,5b,5c,5d
	1c	1b,1d,1e,2a,2b,2e,3a,4a,4b,4c,4d,4e,5a,5b,5c,5d,5e
	1d	2b,4a,5d
	1e	1b,2a,2b,2c,3a,3b,4a,4b,4e,5b,5d
	2b	1b,1e,2a,2c,2d,2e,3b,3d,3e,4b,4c,4d,4e,5a,5b,5c,5d,5e
	2c	1b,1e,2a,2b,2d,2e,3b,3d,4a,4e,5b,5c,5d
	2d	1b,1e,2b,2c,2e,5d
	2e	1d,2a,2b,4a
	3c	1b,1c,1d,1e,2a,2b,2e,3a,3e,4a,4d,4e,5a,5b,5c,5d,5e
	3d	1b,1d,2b,2c,3b,5d
	3e	1b,1d,1e,2a,2b,2c,3b,4a,4b,5b,5d,5e
	4d	2b,2c,3b,5e
	4e	1d,2b,4a
	5e	1b,1c,1d,2a,2b,2c,2d,2e,3a,3b,3c,4a,4b,5b

6.3 移動着手を含む場合

上の結果から、青番の初手が [1a] または [1b]、[2c]、[3c] のいずれかである場合には青番の必勝である可能性が高いことがわかった。よって、初手を [3c] に絞り、移動着手まで含めて考慮して必勝証明を行った。すると、前の実験で得られた transposition table を用いることで、結果としては青番の勝利を証明することができた。

さらに、前の実験で得られた**反証テーブル**と**参考テーブル**のみを用いて再探索を行なうことで終局までの手順を 1 つ求めた。このとき、白番で終局までの手順がなるべく長くなるような着手を選択する手順を選んだ。これによって得られた初手 1c に対する手順 (29 手詰み) を図 6 へと示す。

7. 検証

前章で得た青番の必勝手順について、その信憑性を確かめるために、先手青番を我々の必勝手順、後手白番を WYonmoque:AI レベル 5(最大) として 200 局の対局実験を行った。WYonmoque とは、立教大学の小副川によって開発された Yonmoque の対局プログラムであり、Yonmoque の考案元であるギフトボックス社から強い AI プログラムとして紹介されている [13]。これらの 200 局の対局を行なった結果、すべての対局において我々のプログラムである先手番の勝ちが確認された。

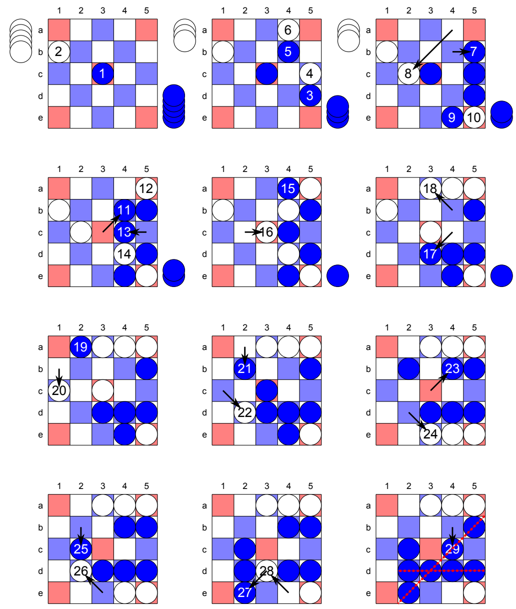

図 6 青番初手 3c からの手順

8. まとめ

本研究では、二人零和完全確定情報ゲームである Yonmoque について初手からの必勝手順解析を求めることを目的として、既存の Depth First Proof Number 探索アルゴリズムに対して, 着手の不可逆的性質を用いて GHI 問題として知られている問題を回避して安全な解析が可能であることを示した。結果として Yonmoque の初手からのゲーム木探索に成功し、このゲームが先手番必勝であること、またその具体的な詰み手順を求めた。さらに、実際の対局を用いた検証実験では、解析結果の手順を使用した実際の対局においてすべて先手番が勝つことを示し、この解析結果に対して一定の信頼性を得た。

本研究で用いた手法は、Yonmoque に限らずリバーシやチェスなど多様な完全情報ゲームに対して適用可能なものであると考えられる。さらにこの手法だけを用いるのではなく、安全な transposition table を作成するための条件の一要素として扱うことで、既存の GHI 対策手法と組み合わせる形での実装も可能であると思われる。今後の展望としては、本研究の手法を活用して Yonmoque 以外の様々なゲーム木の解析をより深く進めることが期待される。

謝辞 対局による検証実験を行うにあたって、小副川氏制作の Yonmoque の対局ソフトウェア WYonmoque を提供して頂いた山本光男氏、そして対局実験に協力して頂いた村松研究室の皆様へここに心より感謝の意を表する。

参考文献

[1] L.V.Allis, M.van der Meulen, and H.J.van den Herik: "Proof-Number Search", *Artifical Intelligence*, 66(1), pp. 91-124, 1994.

[2] 金子知適, 田中哲朗, 山口和紀, 川合慧: ”新規節点で固定深さの探索を併用する df-pn アルゴリズム”, *The 10th Game Programming Workshop*, pp.1-8, 2005.

[3] A. Kishimoto and Martin Muller: ”Df-pn in Go: An Application to the One-Eye Problem”, *Advances in Computer Games 10*, pp. 125-141, 2003.

[4] A. Kishimoto: ”Correct and Efficient search Algorithms in the Presence of Repetitions”, *Ph. D. thesis*, Department of Computing Science, University of Alberta, 2005.

[5] A. Kishimoto and Martin Muller: ”A General Solution to the Graph History Interaction Problem”, *Proc. of 19th National Conference on Artifical Intelligence (Association for the Advancement of Artificial Intelligence '04)*, pp. 644-649, 2004.

[6] J. Schaeffer, N. Burch, Y. Björnsson, A. Kishimoto, M. Müller, R. Lake, P. Lu, and S. Sutphen, “Checkers Is Solved”, *Science* **14** Vol. 317 no. 5844 pp. 1518-1522, 2007.

[7] 岸本章宏: ”不詰を正しく証明するアルゴリズム”, 情報処理学会シンポジウム論文集, pp. 1-8, 2004.

[8] 小谷善行, 岸本章宏, 柴原一友, 鈴木豪: ”ゲーム計算メカニズム”, コロナ社, 2010.

[9] 長井歩, 今井浩, ”df-pn アルゴリズムと詰将棋を解くプログラムへの応用”, 情報処理学会論文誌, 43(6), pp. 1769-1777, 2002.

[10] A. J. Palay: ”Searching with Probabilities”, *Ph. D. thesis*, Carnegie Mellon University, 1983.

[11] 田中哲朗: ”「どうぶつしょうぎ」の完全解析”, Information Technology Center, The University of Tokyo, 2010.

[12] 田中哲朗: ”ボードゲーム「シンペイ」の完全解析”, 情報処理学会研究報告. ゲーム情報学, pp. 65-72, 2006.

[13] 山本光男: “Yonmoque Game ヨンモクゲーム”, http://www.logygames.com/yonmoque/ (2013/02/04 アクセス)

[14] A. L. Zobrist: ”A New Hashing Method with Application for Game Playing”, *Technical Report.* 88, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1969.

付　録

A.1 先手番必勝となる手順

表 A·1 初手 1b,2c,3c における先手番必勝の全着手候補

初手	2 手目	先手番必勝となる 3 手目の候補
1b	1a	1d,2d,2e,3d,4a,4d,4e
	1c	1a,1d,1e,2a,2b,2c,2d,2e,3a,3b,3d,3e,4a,4b,4c,4d,4e,5b,5c,5d
	1d	1e,2d,2e,3c,4a,4d,4e
	1e	1a,1c,1d,2a,2b,2c,2e,3b,3c,3d,3e,4a,4b,4c,4d,4e,5a,5b,5c,5d,5e
	2a	1a,1d,1e,2d,2e,4a,4e,5b
	2b	1a,1d,2a,2c,2d,2e,3a,3c,4a,4b,4c,4d,4e,5a
	2c	1a,1c,1d,2d,2e,3c,3d,3e,4a,4b,4c,4d,4e
	2d	1a,1c,1d,2c,2e,3b,3e,4a,4b,4d,4e
	2e	1c,1d,1e,2b,2c,4a,4b,4d,4e,5b
	3a	1a,1c,1d,1e,2a,2b,2c,2d,2e,3b,3c,3d,3e,4a,4b,4c,4d,4e,5a,5b,5d,5e
	3b	1a,1c,1d,1e,2b,2c,2e,3c,3d,3e,4a,4b,4c,4d,4e,5a,5b,5c,5d,5e
	3c	1a,1c,1d,1e,2a,2b,2c,2d,2e,3a,3b,3e,4a,4b,4c,4d,4e,5a,5b,5c,5d,5e
	3d	1a,1c,1d,1e,2a,2c,2d,2e,3a,3b,4a,4b,4e,5b,5d,5e
	3e	1a,1c,1d,1e,2a,2b,2c,2d,2e,3a,3b,4a,4b,4d,4e,5a,5b,5c,5d,5e
	4a	1d,1e,2d,2e,4d,4e,5b
	4b	1d,2c,2d,2e,3a,3b,3d,4a,4e,5a,5e
	4c	1d,2a,2c,2d,2e,3a,3c,4a,4e,5a,5e
	4d	1c,1d,2b,2c,2e,3a,3b,3c,4a,4b,4e,5e
	4e	1c,1d,1e,2a,2b,2c,2e,3b,4a,4b,5b,5e
	5a	1a,1c,1d,1e,2a,2b,2c,2d,2e,3c,3d,3e,4a,4d,4e,5d,5e
	5b	1a,1c,1d,1e,2b,2c,2d,2e,4a,4d,4e,5a,5e
	5c	1a,1c,1d,1e,2a,2b,2c,2d,2e,3a,3b,3d,4a,4b,4d,4e,5a,5b,5d,5e
	5d	1d,1e,2a,2b,2c,2d,2e,3b,3c,3d,4a,4e,5e
	5e	1a,1c,1d,1e,2a,2b,2c,2d,2e,3a,3b,3c,4a,4b,4e,5a,5b
2c	1a	1d,1e,3b,4a,4c,4d
	1b	1a,3e,4b
	1c	1b,1d,2a,2e,4b,4c,4d
	2a	1d,4e
	2b	1a,1b,1d,1e,2e,3a,3b,4c,4d
	3a	1b,1d,1e,2a,2e,3e,4a,4b,4d,4e,5b
	3b	1a,1b,1d,1e,3a,4a,4b,4d,5a,5b
	3c	1b,1c,1d,2a,2e,3a,3b,3d,3e,4b,4d,5b,5c,5d
	4a	1c,1d,2d,2e,4d
	4b	1b,1c,1d,1e,2a,2e,3e,4c,4d,4e,5a,5b
	4c	1a,1b,1c,1d,1e,2a,2e,4a,4b,4d,4e,5b,5c,5d
	5a	1a,1b,1c,1d,1e,2a,2b,2d,2e,3d,3e,4a,4c,4d,4e,5d
	5b	1b,1c,1d,2b,2e
	5c	1a,1b,1c,1d,1e,2a,2b,2d,2e,3a,3e,4a,4b,4c,4d,4e,5a,5b,5d,5e
3c	1a	1d,2e,3e,4a,4d,4e,5b,5c,5d,5e
	1b	1a,1d,3e,5d
	1c	2a,2e
	2b	1b,2a,2d,3e,4b,4d,4e,5c,5d
	2c	1b,1d,2a,2b,2d,2e,3a,3e,5b,5c,5d

A.2　後手番必勝となる手順

表 A·2　初手 1c,2b における後手番必勝の全着手候補

2 手目まで	3 手目	後手番必勝となる 4 手目
1c→1b	1b	1d,2a,2c,2d,3d,4a,4b,4c
	1c	1b,1d,2a,2d,3b,3d,4a,4b,4c,4e,5b,5c,5d
	1d	3d,4b
	1e	3b,3d,4b,4c,4e,5b,5d
	2a	1b,1d,2d,3b,3d,4a,4b,4c
	2c	1b,1e,2a,2d,2e,3b,3d,3e,4b,4c,4d
	2d	1e,2c
	2e	3b
	3a	1b,1d,2a,2c,2d,2e,3d,3e,4a,4b,4c,4e,5d
	3b	1b,2a,2c,2d,3d,4b,4c,4d,5a,5b,5c
	3c	1b,2a,4e,5d
	3d	4e
	3e	3b,4b,4d,4e
	4a	2d,4c
	4b	3b,5a
	4c	5d
	4d	4e,5d
	4e	1b,2a,2c,3b,4b,5d
	5a	2c,2d,2e,3d,4c,4e,5d
	5b	2c
	5c	2c,2d,4d,5d
	5d	1b,2a,2c,2d,3b,4e
	5e	1d,1e,2c,2d,2e,3b,3d,4a,4b,4c,5a,5b
2b→1a	1a	1d,2d,2e,3b,4a,4b,4c,4d,4e,5b,5d
	1d	2d
	1e	1a,1d,2c,2d,3a,3b,4a,4b,4c,4d,4e,5b,5c,5d
	2a	1a
	2b	1a
	2c	1a,1e,2d,2e,3a
	2d	1a,1e,2e,3b,4a,4e
	2e	1a,1e,2c,4a,5d
	3a	1a,1d,1e,2a,2b,2c,2d,2e,3b,3e,4a,4b,4d,4e,5a,5b,5c,5d,5e
	3b	1d,1e,2d,2e,5b
	3c	1a,2c,2d,2e,3a,3e,5b,5c,5d
	3d	1d,1e,2e,3b,4a,5d
	3e	1d,1e,2c,2d,2e,3a,3b,4a,4b,4c,4e,5b,5c,5d,5e
	4a	2d,2e,4d
	4b	2d,2e,3a,4a
	4c	1e,2d,2e,3a,3b,4a,5b
	4d	1e,2e,3b,4a,5d,5e
	4e	2c,2e,3a,3b,4a,4b,5b,5d,5e
	5a	1e,2a,2c,2d,2e,3a,3b,4d
	5b	2c,2d,2e,3a,4a
	5c	1d,1e,2c,2e,3b,4a,5a,5d,5e
	5d	1e,2c,2d,2e,3b,4a
	5e	1a,1e,2c,2d,2e,3a,3b,4a,4b,5b