ボードゲーム「シンペイ」の完全解析

田　中　哲　朗†

「シンペイ (SIMPEI)」は高橋晋平氏が考案し株式会社バンダイが 2005 年 7 月に発売したボードゲームである．縦横斜めに駒を並べることを目標とする点は，n 目並べの多くのバリエーションと共通しているが，盤面を「上の世界」と「下の世界」の 2 つに分けている点や，挟んだ駒を自由に移動できる点に特徴があり，高いゲーム性を有している．この点が評価されて，2006 年度の GPCC(Games and PuzzlesCompetitions on Computers) の課題問題に選ばれた．

「シンペイ」は二人完全情報零和ゲームなので，すべての局面の理論値 (勝ち，負け，引き分けのいずれか) を決定することが可能である．

本論文では，後退解析 (Retrograde analysis) をベースにしたプログラムを用いてすべての局面の理論値を求めた．そして,「シンペイ」の公式ルールの初期配置が後手必勝であること，1 手目を自由に置くことが許されれば先手必勝であることを確かめた．また，勝ちに要する最長手数が 49 手であること,「シンペイ」のゲームにツークツワンク (ZugZwang) が存在することや，単純なサイクルが存在し，その周期は 1,3,4 の 3 通りしかないことなど，いくつかの興味深い性質を求めることができた．

Complete analysis of a board game “SIMPEI”

Tetsuro TANAKA†

“SIMPEI” is a board game, which was designed by Simpei TAKAHASHI. It was released in Jul. 2005 by BANDAI. Although it is similar to other n-stones-in-a-row games, it has two unique features. The first one is the two separated worlds in a board, the upper world and the lower world. And the second one is to move in free the opponents piece which is clipped by one player’s pieces. This game is selected one of the problems of this year in the GPCC(Games and Puzzles Competitions on Computers).

Because “SIMPEI” belongs to perfect information two player zero-sum games, in a theoretical sense, all states in the game can be decided as winning, losing or in draw. We practically analyzed all game states with a program based on retrograde analysis.

In this paper, we show the result of the analysis. We found that the second player can always win in the “SIMPEI” official rule. And we present some other interesting features of the game.

1. はじめに

「シンペイ」☆は高橋晋平氏が考案し株式会社

† 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
ktanaka@ecc.u-tokyo.ac.jp

☆ 株式会社バンダイの登録商標.

バンダイが2005年7月に発売したボードゲームである．

ルールは以下のようになっている．

道具　図1のようなボード1面と，赤と青☆の駒それぞれ4個を使ってプレイする．駒はボード上の25箇所の穴のそれぞれに1つまで置くことができる．ボード上の交点は4×4の上の世界と3×3の下の世界に分かれている．図1では大きな丸が上の世界，小さな丸が下の世界を表してる．

進行　プレイヤー2人で遊ぶゲームである．それぞれのプレイヤーは自分の色の駒を4つずつ手に持ってプレイを開始する☆☆．最初の8手までは手持ちの駒を盤面上の空マスに置いていく．盤面のどこに置いても良いが，最初の1手は，図2の位置に置く必要がある．9手目以降は盤面上の自分の駒を1つ選んで斜めの線に従って，1つだけ動かす．動かす先は空マスである必要がある．元々上の世界にあった駒の場合は下の世界に，下の世界にあった駒は上の世界に移ることになる．図3のように自分の駒が動かせない場合は，パスして相手の手番に変わる．

挟む　置いた駒，あるいは移動した駒によって挟まれた状態(リバーシと同様に縦横斜めに挟む)の敵の駒は，元の場所から空いてる空欄に移動する．ただし，この際は上の世界と下の世界は独立なものとして扱う．

ゲームの目標　上の世界，あるいは下の世界で，自分の手番で縦横斜めに駒を3つ並べると勝ちになる．ただし，4つ並べるのは勝ちにはならない(図4左)．また，上の世界と下の世界は駒を並べることに関しては独立なので，図4右のような並べ方をしても勝ちにはならない．

縦横斜めに駒を並べるのを目標とする点は，n目並べの多くのバリエーションと共通しているが,「上の世界」と「下の世界」という2つを設定している点や，挟んだ駒を自由に移動できるという特徴があり，高いゲーム性を有している．この点が評価されて，このゲームは2006年度のGPCC(Games and PuzzlesCompetitions on Computers)☆☆☆の課題問題に選ばれた．

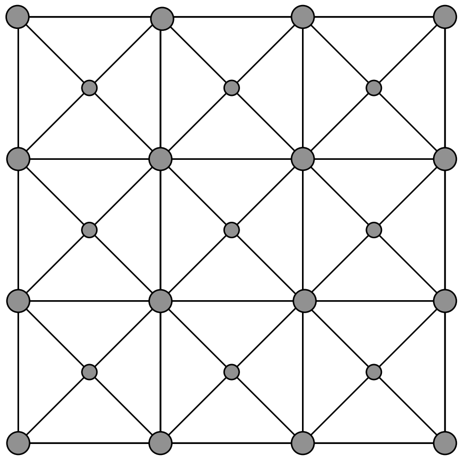

図1　シンペイのボード

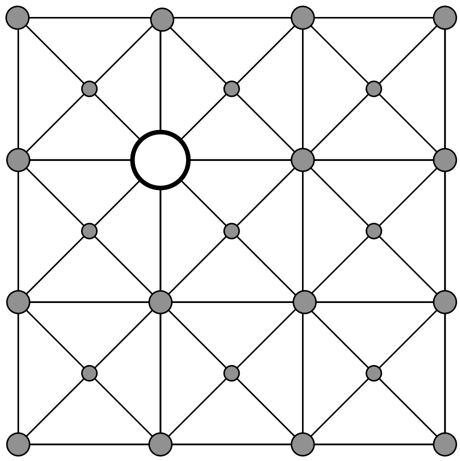

図2　最初の1手

リバーシのように，局面が単調に進行していくゲームではないので，数手進むと元の局面に戻るというケースがある．このようなケースは将棋の千日手のルールのように実質的に引き分け☆4にする場合や，囲碁のコウのルールのように着手禁止とする場合など，さまざまあるが，シンペイの公式ルールでは現在のところ該当する規定がない．本論文中では便宜上，引き分けとして扱うことにする．

2. 解析の方針

ゲームの複雑性を評価する指標として，一般に

☆ 論文中では代わりに白と黒を用いる．
☆☆ 便宜上，論文中では先手を白とする．
☆☆☆ 情報処理学会プログラミング・シンポジウムの分科会として1972年から活動している[1]．毎年ゲームやパズルを計算機で解く課題を決めて，翌年に結果を発表する．
☆4 連続王手の場合を除く．

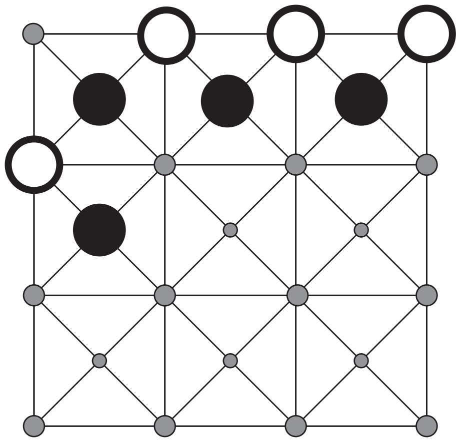

図 3　動かす手がない場合 (白番)

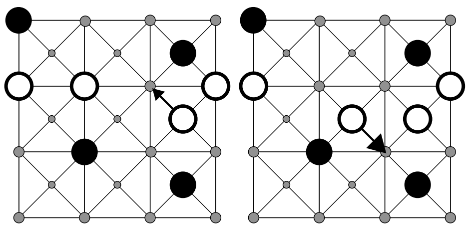

図 4　勝ちでない例

ゲーム木のサイズと総局面数という 2 つの指標が用いられる．シンペイに関しては，手数の上限がなく，また人間のエキスパートによる棋譜もないため，ゲーム木のサイズの予測は難しい☆．

一方，総局面数に関しては，上限を簡単に計算することが可能である．盤面上に駒が n 個ある時の局面数を B_n とすると，以下のような上限が存在する．

$B_0 = 1$
$B_1 \leq {}_{25}C_1 = 25$
$B_2 \leq {}_{25}C_1 \times {}_{24}C_1 = 600$
$B_3 \leq {}_{25}C_2 \times {}_{23}C_1 = 6900$
$B_4 \leq {}_{25}C_2 \times {}_{23}C_2 = 75900$
$B_5 \leq {}_{25}C_3 \times {}_{22}C_2 = 531300$
$B_6 \leq {}_{25}C_3 \times {}_{22}C_3 = 3542000$
$B_7 \leq {}_{25}C_4 \times {}_{21}C_3 = 16824500$
$B_8 \leq {}_{25}C_4 \times {}_{21}C_4 = 75710250$

上の数字の計算の際には，手番の情報を考えていないが，これは手番の情報を考えなくても盤面の情報だけで局面を決めることが可能であるという事実に基づいている．7 手目終了までの局面では，盤面上の駒の個数だけから次の手番が決まる．また 8 手目以降はゲームの手番に関する対称性☆☆により，次の手番を白と固定して良い☆☆☆．次の手番が黒の場合は，盤面の白黒の駒を入れ替えた盤面を考えれば良い．

以上のように，総局面数は 96,691,476 以下におさえられることがわかる．また，上の数え方では，図 5 のように対称な局面もすべて含んでいるので，これによる重複を取り除くと局面数は更に減る．通常の PC でも主記憶上に置ける容量なので，すべての局面の解析をするという方針でプログラムを作成することにする．

3. 解析の概要

本節では，解析のために作成した C++言語のプログラムの概要を述べる．

3.1 盤面の表現と全局面の列挙

ゲーム中の局面は，各マスの状態を 8 ビット (C++の char 型) で表して配列としてアクセスする表現方式を基本として使う．この配列は，番兵 (sentinel) を含めてサイズ 51 の配列で，盤面上のマスとの添字の対応は，図 6 のように定義する．ここに現れない添字は盤外となり，配列には特別な値 `EDGE` が書かれている．

この添字を用いると，3 連ができるかどうか，および挟んでいるかどうかのチェックは，どこを起点にしても添字の差が $\{-10,-9,-8,-1,1,8,9,10\}$ の 8 つの方向を調べれば良いし，9 手目以降の移動先は，添字の差が $\{-5,-4,4,5\}$ の 4 つのマスを調べれば良い．

この表現はメモリ使用量が多いので，25 のマスをそれぞれ 2 ビットで表して，C++言語の long long int で表現する表現方式も併用する．この際に，図 5 で示される 8 つの対称形に関して，それぞれを適用した上で，long long int に変換した値を 8 つ求めて，数として比較した時の値が最小の値を用いるという正規化を行っている．

この正規化により，通常の PC でも十分メモリに収まるサイズになったため，C++言語の `hash_map` を用いて，全局面の数え上げをおこなった．その結果を図 1 に示す．図 5 の対称性のために，見積もった上限の $\frac{1}{8}$ 程度の数で収まっ

☆ 後述するように解析の結果，初期局面からの最小証明木のノード数は 11128 と分かった．

☆☆ 連珠のように黒白によって合法手が違うゲームでは成り立たない．

☆☆☆ 本論文でも駒数 8 の盤面は特に説明がなければ，白番とする

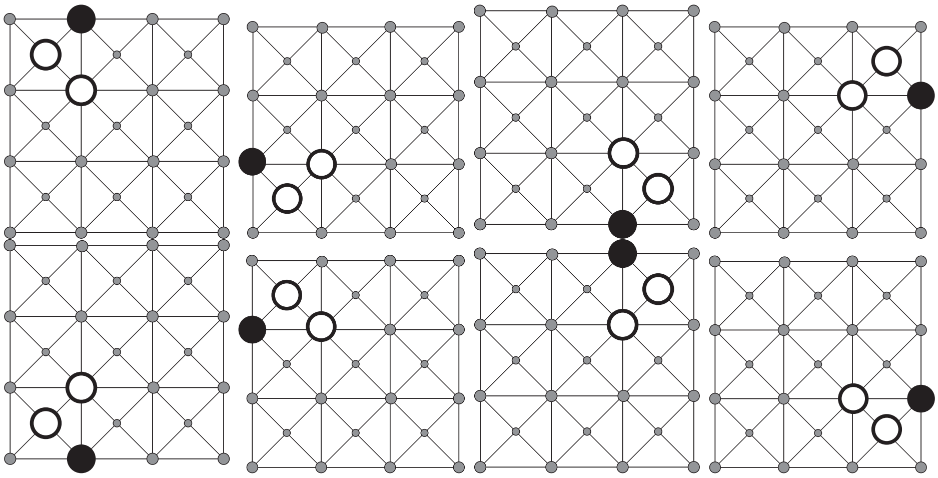

図 5　対称性

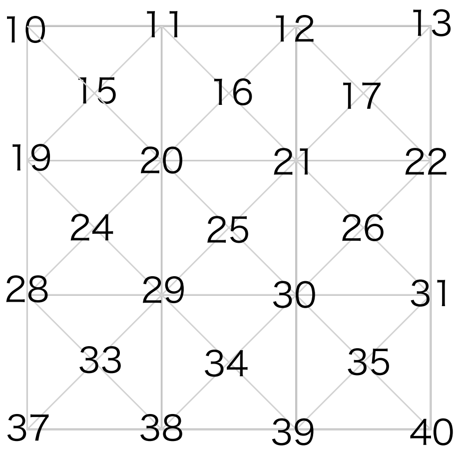

図 6　盤面上のマスの添字

ている．

なお，ここで求めた局面は，初期配置から到達不可能なものも含んでいる．到達不可能な配置としては図 7 のような例がある．右側が到達不可能であることは，勝敗をつけずに 2 手さかのぼるかことが可能かどうかを考えれば分かるだろう．

初期局面の勝敗を決定するという立場からは，初期配置から到達不可能な局面を入れることに意味はないが，今後のルールの変更の可能性も考慮して，以降ではこのような局面も含めて議論することにする．

3.2　後退解析による局面の勝ち負けの確定

すべての局面を数え上げた後は，後退解析 (retrograde analysis) により，すべての局面の勝ち負けを求めることができる．後退解析は以下のようにおこなう．

表 1　駒数ごとの局面数

駒数	局面数
0	1
1	6
2	87
3	915
4	9713
5	67016
6	444336
7	2106976
8	9473115

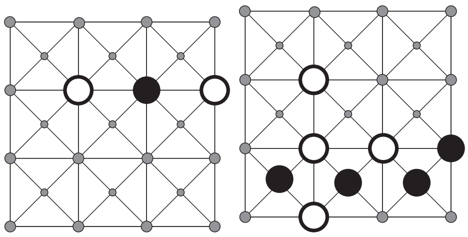

図 7　初期配置から到達不可能な配置

(1)　勝負のついた局面の集合から開始する．

(2)　勝負のついた局面の一手前の局面を求める．

- (手番のプレイヤーの) 負け局面から一手前の局面は (手番のプレイヤーの) 勝ち局面
- 勝ち局面から一手前の局面の勝敗が未確定の時は，そこから可能な手がすべ

表 2　駒数ごとの局面の勝ち負け

駒数	勝ち	引き分け	負け	計
0	1	0	0	1
1	4	0	2	6
2	83	0	4	87
3	701	5	209	915
4	7529	74	2110	9713
5	51951	532	14533	67016
6	347243	3428	93665	444336
7	1781756	10776	314444	2106976
8	6985915	111206	2375994	9473115

て勝ち局面に移行する時は，負け局面とする．

(3)　操作を繰り返して，勝ち局面の集合も負け局面の集合がそれ以上増えなくなったら終了する．

大きな問題を後退解析で解くためには，多数の局面をなるべく少ないビット数で表し，ディスク上でもアクセスできるようにアクセスを局所化したり，並列化するなど様々なテクニックが必要となるが[2)]，ここでは局面数が十分主記憶に収まるため，特に工夫をしないで実装した．

Opteron 252(2.6GHz) x 2, メモリ 12GB のマシンで 12 分ほど実行したところ，プログラムが終了した．結果を図 2 に示す．

4.　解析結果の検討

この節では解析結果の中で人間にとって興味深いと思われるいくつかの話題を取り上げる．

4.1　初期局面の勝敗

シンペイの現在のルールの初期配置である図 2 は次の手番，すなわち黒の勝ちであることがわかった．なお，黒が勝つ手は図 8 の手のみに限られている．

この後，黒が最短の勝ちを目指しても 17 手が必要となる．また，黒の勝ちを証明するのに必要な最小の証明木を求めたところ，そのサイズは 11128 ノードと分かった．

一方，図 1 の駒が全くない状態で先手が 1 手目を自由に選択できるようにルールを変更したとすれば，白が勝てる．図 9 の二つの手が白が勝つための 1 手目となる．証明木の小さいのは右の方で証明木のサイズは 1081 となっている．

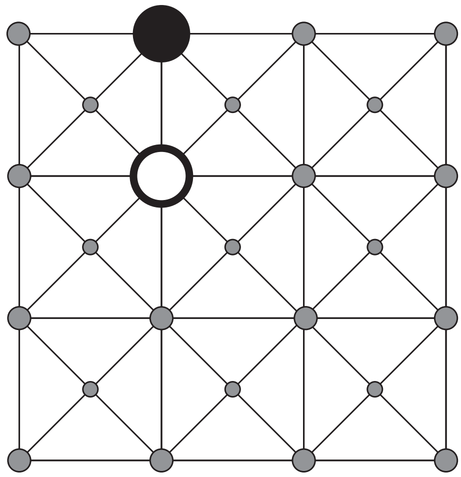

図 8　黒が勝つための 2 手目

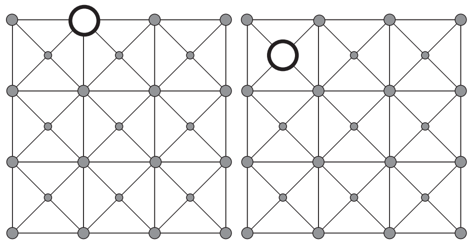

図 9

4.2　分岐数最大の局面

普通の局面では分岐数 (合法手の数) は 10-20 手程度だが，一度に複数の相手の駒を挟める場合は分岐数が増える．分岐数が最大となる局面を求めたところ，図 10 が見つかった．この時の分岐数は 2403 である．この局面は次の手番である白の勝ちとなっている．

4.3　ツークツワンク (ZugZwang)

ツークツワンクはチェスで使われている用語である．自分がパスをすることが許されれば勝ちだが，手を進めなくてはいけないために負けるという局面をいう．

手番を入れ替えた後のプレイが自然におこなえるように，白黒の駒数が同じ局面に限って，ツークツワンク局面の検出をおこなった．その結果，表 3 のようにいくつかのツークツワンク局面を検出することができた．

その中には，図 11 のように手番を入れ替えた局面が元の局面と対称性を考慮すると同一である局面も含まれている．

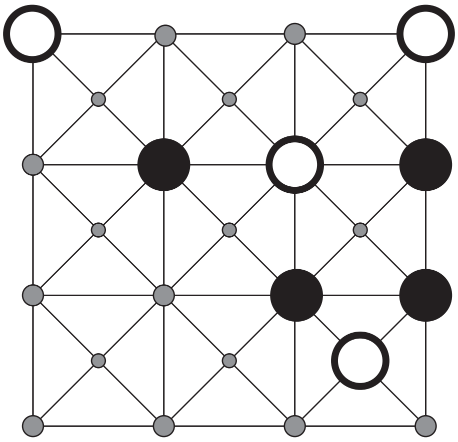

図 10　分岐数最大の配置

表 3　ツークツワンク局面の数

駒数	局面数 (自己対称)
0	0(0)
2	0(0)
4	49(3)
6	4464(44)
8	14362(96)

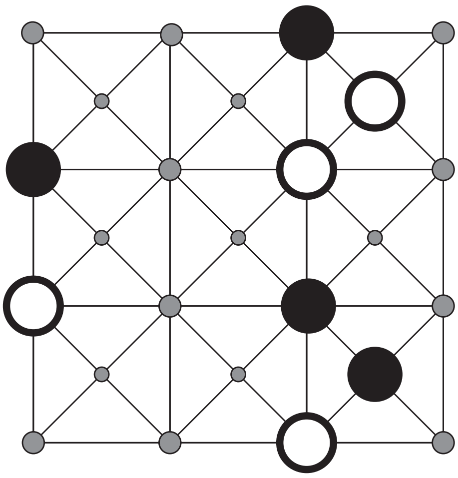

図 11　自己対称な zugzwang 局面

4.4　勝ち負け以外の局面

表 2 のように，シンペイには後退解析によっても勝ち局面にも負け局面にもたどり着かない局面が存在することがわかった．一度このような局面に入ると，白も黒も勝つことができなくなる．

表 4　単純なサイクルの周期

周期	数
1	1
3	9
4	1724

局面数は限られているので，どちらも負けないように着手を続けると，有限回で過去に出現したのと同一の局面に到達する．このような時に関する規定がルールにはないので，延々とプレイを続けることが可能だが，ここでは，仮に引き分けとして考えることにする．

どちらも負けない手が 1 手だけで，そのような手の繰り返しでのみで，自分自身に戻る単純なサイクルが存在するかどうかが興味の対象となる．このようなサイクルを検出して数え上げるプログラムを作成して実行した．その結果表 4 にあるような数のサイクルが見つかった．まったく同じ盤面に戻るためには，かならず偶数回の手番が必要になるが，ここでは手番を取り替えて対称な局面は同一局面としたために，周期 1, 3 のサイクルが存在している．このうち，周期 1 のサイクルを図 12 に示す．図 12 の左側の局面で白が負けない手は，右の状態に移行する手しかないが，右側は左側の黒白を入れ替えた盤面の対称形になっている．

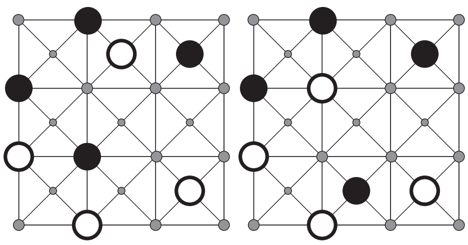

図 12　周期 1 のサイクル

4.5　詰めシンペイ問題の抽出

すべての局面から特徴的な勝ち局面を探し出して，次の一手問題を作ることを考える．問題として成立することを重視して，次に勝つ手が 1 つしかないものを問題として考える．

この中から，面白い問題と成りうる要素を探してみることとする．

詰将棋の問題では，100 手を超えるような長手

数の問題が作成されているが，シンペイの場合を調べてみることにも意味がある．調べてみたところ，勝つまでの手数の最大値は 49 だった．この中には勝つ手が一つだけものものもあった．このうちの一つを図 13 に示す．図 13 左の局面から白が勝つ手は，右の局面に移る手だけである．

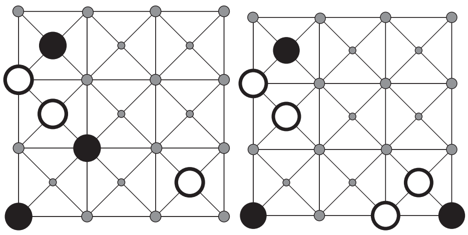

図 13　勝つまでの手数が 49 手必要な局面

勝つ手が一つで，その手の後で相手の合法手の数が最大となる局面を求めたところ，図 14 の局面が得られた．この局面は白勝ちの局面だが，勝つ手は矢印の動きしかない．

次の手で，黒は白の 4 つ駒を挟んで好きな所にとばすことができるが，どの局面に移行しても白勝ちになる．ただし，これを人間が手で確かめるのは困難かもしれない．

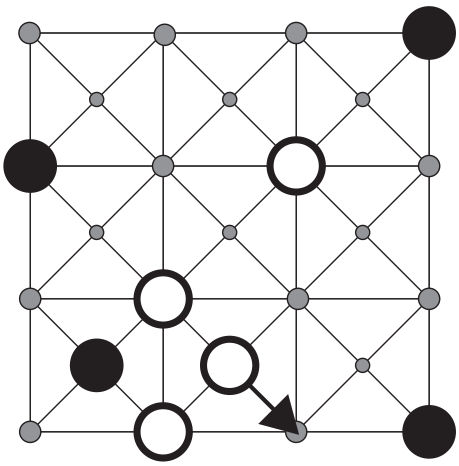

図 14　4 つ挟ませる手が正解の局面

5. ま と め

本研究では，シンペイのすべての局面の勝敗を求める解析をおこなった．その結果，シンペイが後手必勝であることを示すだけでなく，勝ちに要する最長手数が 49 手であること,「シンペイ」のゲームにツークツワンク (ZugZwang) が存在すること，単純なサイクルが存在し，その周期は 1,3,4 の 3 通りしかないことなど，いくつかの興味深い性質を求めることができた．

この結果を元に，後手で勝つためのプログラムを作るのは容易である．一方，人間相手に先手で勝ちやすいプログラムを作る場合は，OM-search[3] のように相手プレイヤーに関するモデルを使った探索手法が有効と考えられる．

シンペイが後手必勝だという事実が判明したという事実は，人間にとってのゲームの面白さを減らす結果につながるかもしれない．トッププレイヤーが皆，証明木を丸暗記してしまえば，競技は成り立たないだろう．証明木のサイズは 11128 ノードなので，難しいことは確かだが不可能とは言えない．

ゲームの寿命を伸ばすために図 15 のような引き分け局面からスタートするようにルールを変更するという対策が考えられる．ただし，それでゲームの面白さが保たれるかどうかは今後の検証が必要となるだろう．

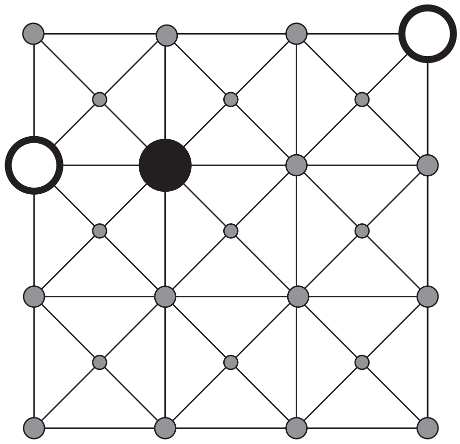

図 15　石数 3 の引き分け局面

参 考 文 献

1) 小谷善幸, 南雲夏彦, 飯田弘之, 竹内郁雄, 一松信 : プログラミングシンポジウム GPCC のゲームとパズル, 情報処理学会研究会資料, 1999-GI-1, pp. 55 – 61 (1999).

2) J. Romein and H. Bal: Solving the Game of Awari using Parallel Retrograde Analysis, IEEE Computer, Vol. 36, No. 10, pp. 26 – 33(2003).

3) Iida, H., Uiterwijk, J.W.H.M, Herik, H.J. van den, and Herschberg, I.S: Potential Applications of Opponent-Model Search; part 1: the domain of applicablity, ICCA Journal, Vol. 16, No. 4, pp. 201 – 208(1993).