

## ゲーム Hex における必勝手順の検証プログラム

三島 健<sup>†1</sup> 櫻井 英俊<sup>†1</sup>  
吉元 昭裕<sup>†1</sup> 野下 浩平<sup>†1</sup>

ゲーム Hex では、具体的な必勝手順を求めるために、様々な数学的技法が提案されている。8×8 や 9×9 など大きい盤面に対する必勝手順を導くためには、これらの技法に基づいたシステムで人間とコンピュータの共同作業ができるものを実装する必要がある。本稿では、与えられた局面に対してユニオン連結性を判定するプログラム A と B を提示する。プログラム A は局面表を用いた AND-OR 探索に基づくものであり、Hex 特有の着手選択アルゴリズムにより強化した。プログラム B は、さらに仮想準連結などを組み込んで改良したものである。これらのプログラムは、実行時間と探索節点について評価する。ベンチマーク問題として、Noshita による 7×7 と 8×8 の必勝手順に現れる約 40 種類の難しい問題を選んだ。我々の実験結果によると、これらの局面の正しさは、非常に速く検証できた。このことは、我々のプログラムが実用に使えるということを意味する。着手選択（順序付けと先行排除）は、人間の熟練者と同程度に正確であると評価できる。本稿の実験結果をみると、我々のプログラムは、8×8 以上の必勝手順の検証・発見のための対話システムを開発する際に最も基本的な道具として使えることが分かる。

### Testing Connectivity for Verifying Winning Strategies in Hex

KEN MISHIMA,<sup>†1</sup> HIDETOSHI SAKURAI,<sup>†1</sup>  
AKIHIRO YOSHIMOTO<sup>†1</sup> and KOHEI NOSHITA<sup>†1</sup>

In the game of Hex several mathematical techniques have been proposed for constructing explicit winning strategies. If we try to construct them for relatively large boards of 8×8, 9×9 or more, it is necessary to implement a system which enables human-computer cooperation based on those techniques. This paper presents two programs A and B for testing whether, for a given position, a desired union-connection can be attained. Program A is based on AND-OR searching with a transposition table, which is enhanced by Hex-dependent move-selection algorithms. Program B is further improved by incorporating, among others, virtual semi-connections. We evaluate the efficiency of those

programs in terms of the running time as well as the number of searched nodes. For the set of benchmark problems, we have selected about forty hard positions from Noshita's winning strategies for 7×7 and 8×8. Our computing experiment shows that the correctness of those positions has been verified very quickly in that our programs can be used for practical purposes. The quality of the move-selection (ordering and preclusion) can be evaluated to be as accurate as human expert players. Our results show that our programs can be used as the most basic tool for building an interactive system to verify/find winning strategies for 8×8 or more.

### 1. はじめに

ゲーム Hex は Nash により  $N \times N$  の盤面において先手必勝手順の存在が証明された。しかし、具体的な手順を示すことは長年の研究課題である<sup>6)</sup>。7×7 以下の大きさの盤面においては、仮想連結<sup>1)</sup> による盤面分割の方法を用いて必勝手順が求められている<sup>8)</sup>。2004 年 Noshita は、ユニオン連結 (union-connection) の概念を導入し、これに基づいて 8×8 (初手 54) に対する完全な必勝手順を示した<sup>7)</sup>。2006 年には三島ら<sup>9)</sup> が  $\sigma$  連結と  $\sigma$  拡張を提案し、これによって 8×8 (初手 63) と 9×9 (初手 55) の十分に簡潔な必勝手順を示した。このように現在では、9×9 に対する必勝手順が人の手で導き出せる範囲に入ってきている。

コンピュータを用いた必勝手順の研究も行われている。2000 年に Enderton がコンピュータによって 6×6 のすべての初手における勝利プレイヤを示した<sup>3)</sup>。2005 年には Hayward らが 7×7 のすべての初手における勝利プレイヤを示した<sup>5)</sup>。

大きな盤面における必勝手順を導くためには、人の手で盤面分割を行い、コンピュータでその細部の検証を行うなどの共同作業が重要になってきている。2006 年に Hayward らはコンピュータによる必勝手順検証用のプログラムを作成し、2001 年に Yang らが示した 7×7 の必勝手順<sup>8)</sup> の正しさの検証をしている<sup>4)</sup>。

今後 (様々な初手における) 8×8 や 9×9 の必勝手順や、10×10 以上の大きな盤面において必勝手順が導かれていくことが予想される。その際、人間の手によって分割された盤面に対して、(コンピュータなどによって) 石どうしの連結性を証明することは必須になってくる。また、この部分は、計算量という点において大きな部分を占める。そこで、本稿で

<sup>†1</sup> 電気通信大学大学院電気通信学研究科

Department of Computer Science, The University of Electro-Communications

は、ユニオン連結の成否を出力する連結判定のプログラム（プログラム A、プログラム B）を作成した。さらに、プログラム A と B の性能を評価するため、Noshita<sup>7)</sup> の示した必勝手順とそこで用いられているユニオン連結の正しさの検証を行った。

プログラム A と B の基本的なアルゴリズムは、ハッシュ表を用いた AND-OR 探索であるが、Hex 特有の手法を取り入れることによって、探索の高速化を図っている。プログラム A、B とともに、着手の順序付けのために、Hex に依存した多くの評価項目を線型結合した評価関数を用いている。各々の探索節点では、着手候補を限定する手法（先行排除）を用いることによって、探索を行う必要のない着手を取り除いている。また、基本的な仮想連結の連結パターンを登録して、パターンの照合により、探索の効率向上を図っている。さらに、プログラム B では、仮想準連結<sup>1)</sup> に基づく連結判定法を組み込んでいる。

プログラム A の有用性を評価するため、実例として、Noshita の示した  $7 \times 7$  の必勝手順の正しさの検証を行う。Noshita の示した証明木の（後手局面の）節点には、ユニオン連結に基づいて分割された盤面が付与されている。これらの分割された部分盤面を入力として与えた。その結果、すべてのユニオン連結においてその正しさを確かめることができた。入力として与えたユニオン連結の総数は 23 個であるが、1 分未満の計算時間ですべての検証を行うことができた。

次にプログラム A と B の性能比較を行う。そのために、Noshita の示した  $8 \times 8$  の盤面における必勝手順で、主要なユニオン連結の正しさを検証した。プログラム B の方がプログラム A に比べて、おおむね短い実行時間で検証を終了している。実行時間の差が顕著に現れたものでは、200 倍近い差が現れたものがあった。

計算時間について、本稿のプログラムは、必勝手順の検証・発見の実用に足ると判断できる。また、着手生成の良さについても、着手の順序付けと先行排除とともに、人間の熟練者と同じ程度に正確に行っている。

本稿の貢献は、人間・コンピュータの共同作業によって必勝手順を導いていけることを、具体的な実験データをもって示したことである。特に、 $8 \times 8$  以上の必勝手順を検証するための最も基本的な道具で、実用性能を持つものを実現したことは、本稿の大きい貢献であるといえる。コンピュータによって完全に自動的に必勝手順を発見することは、 $7 \times 7$  でさえ難しい。Hayward ら<sup>5)</sup> は、アルゴリズムを説明しているが、その詳しい実現方法については述べていない。彼らのプログラムでは、 $7 \times 7$  の必勝手順（すべての初手）を求めるのに約 615 時間かかっている。また、 $8 \times 8$  以上の盤面については言及していない。 $8 \times 8$  では、盤面の状態数は、単純計算で、 $7 \times 7$  の  $3^{64-49} \simeq 1.4 \times 10^7$  倍となり、コンピュータに

よって必勝手順を求めることは不可能とされている<sup>6)</sup>。Yang らによる必勝手順<sup>8)</sup> の正しさは、2006 年にコンピュータによって検証されている<sup>4)</sup>。 $8 \times 8$  以上の必勝手順の検証は、先行研究で行われたことがない。これまでの研究から分かるように、必勝手順の発見・検証の両方の側面において、 $7 \times 7$  で利用された方法を  $8 \times 8$  以上で適用していくことは難しいといえる。 $8 \times 8$  以上における必勝手順を求めていくためには、新しい証明技法の開発はいうまでもないが、人間によって分割された（部分）盤面における連結性をコンピュータで検証していくなど、人手・コンピュータの共同作業が必要になってきている。本稿の 5 章では、プログラム A と B の性能評価実験を行っている。 $7 \times 7$  と  $8 \times 8$  の盤面において、プログラム A と B とともに連結性を証明するために十分実用的な早さで検証を終えている。これらの実験結果は、人間・コンピュータの共同作業により  $8 \times 8$  以上の必勝手順を導くためのシステムが実現可能であることを示しているといえる。

## 2. 基本概念

この章では、Hex の基本的な事項を説明する。

Hex は 2 人で行う完全情報ゲームである。正六角形が敷き詰められた盤を用い、先手と後手のプレイヤーが交互に自分の色の石を盤上に置く。先手は黒石、後手は白石を用いる。図 1 のように、先手は盤の上下を黒石で連結させれば勝利となる。また、後手は盤の左右を白石で連結させれば勝利となる。

本稿では盤上での座標を  $(x, y)$  と表現する。 $x$  は列の番号、 $y$  は行の番号である。例をあげると、図 1 の白石 a の座標は  $(2, 3)$  と表現する。

仮想連結 (virtual connection)<sup>1)</sup> とは、空セルの領域  $S$  内において、相手から着手を行っても、自分の石  $x$  と他の自分の石  $y$ 、もしくは端との連結が保証されるという概念である。

図 2 は先手の仮想連結の例である。図 2 の例では石 a と上端、石 b と石 c、石 d と下端が仮想連結によって連結している。石 a と上端の仮想連結、石 b と石 c の仮想連結をいず

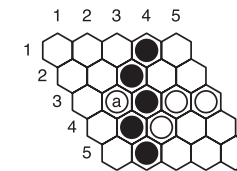


図 1 先手の勝利

Fig. 1 The first player wins.

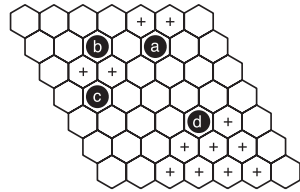


図 2 先手の仮想連結の例  
Fig. 2 Examples of virtual connection.

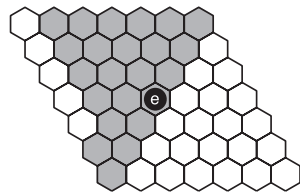


図 3 先手のユニオン連結の例  
Fig. 3 An example of union-connection.

れも一間跳びと呼ぶ。また、石 d と下端の仮想連結を台形レンマもしくは *T*-lemma と呼ぶ。図中の '+' で示されたセルは、それぞれの仮想連結に必要な空セルの領域である。

仮想準連結<sup>1)</sup>とは、空セルの領域 *S* 内において、自分から着手を行えば、自分の石 *x* と他の自分の石 *y*、もしくは端との連結が保証されるという概念である。

ユニオン連結 (union-connection)<sup>7)</sup>とは、1 つ以上の連結の候補を持ち、空セルの領域 *S* 内において相手から着手を行っても、その連結の候補のうち少なくとも 1 つが成り立つという概念である。仮想連結は、連結の候補が 1 つのユニオン連結である。

図 3 は先手のユニオン連結の例である。図 3 の例では色のついた領域において石 e と上端、もしくは上端と下端のどちらかの連結が成り立つ。また、*AB*-property<sup>7)</sup> や  $\sigma$  連結と  $\sigma$  拡張<sup>9)</sup> など、ユニオン連結を拡張した概念が提案されている。

先行排除は *must-play* と呼ばれるプレイヤーの着手を限定する手法である (たとえば文献 2), 5)。その考え方は他のゲームでも古くから知られている。

図 4 を例にとって説明する。この盤面から後手 (白) が着手を行うが、その際、探索の効率のために先手が即座に勝利できるような着手を省く。そこで、もう 1 手先手の石を盤に置き、先手が勝利できるような場合を考える。

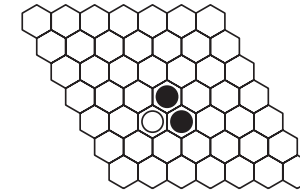


図 4 後手局面  
Fig. 4 A position of the second player's turn.

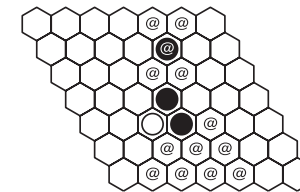


図 5 先手の勝利盤面 1  
Fig. 5 A winning position (1).

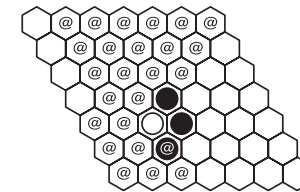


図 6 先手の勝利盤面 2  
Fig. 6 A winning position (2).

先手が (5, 2) に着手を行った場合、図 5 の '@' で示される領域を用い、先手が勝利する。よって、後手は図 5 の '@' で示された領域に着手を行わなければならない (それ以外の着手は先手に (5, 2) に着手されることによって先手の勝利となる)。

先手が (3, 6) に着手を行った場合、図 6 の '@' で示される領域を用い、先手が勝利することになる。こちらも後手は図 6 の '@' で示された領域に着手を行わなければならない。

後手は図 5 と図 6 の 2 つの場合に対応しなくてはならないため、これらの図で示された領域のうち、共通する領域のみを着手として残す。それが図 7 の '@' で示された領域である。

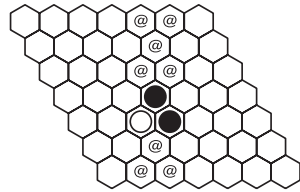


図 7 先行排除により残った後手の着手  
Fig. 7 The second player's surviving moves.

図 4 の盤面では、先行排除によって図 7 で示したように着手を限定することができる。なお、図 7 の '@' すべては、様々なユニオン連結における先行排除の組合せによって取り除くことができる。

### 3. 連結判定プログラム

本稿では、2つのプログラムを使用している。それぞれをプログラム A、プログラム B と呼ぶ。

プログラム A と B は、入力として与えられた部分盤面においてユニオン連結が成立しているか否かを判定する。ユニオン連結は複数の連結の候補を持つ可能性があるため、連結目標が複数設定できる。また、連結目標を任意の石や端に設定することができる。

#### 3.1 仮想連結パターンの利用

プログラム A と B では基本的な仮想連結をあらかじめデータとして登録しておく。目標の連結の成立を判定する際、石が直接接続しているかどうかだけでなく、仮想連結を用いた接続の判定を行う。

図 8、図 9、図 10 はいずれも色のついた領域を用い、黒石と上端が連結する仮想連結である。プログラム A と B では、一間跳びと *T*-lemma に加え、これらの仮想連結（計 5 種類）の連結パターンを使用している。平行移動、線対称を考慮すると、 $7 \times 7$  では、図 8、図 9、図 10 の形の仮想連結は、それぞれ 6 個、4 個、14 個のパターンがある。 $8 \times 8$  では、それぞれ 8 個、8 個、18 個のパターンがある。後手も同数である。なお、計算時間との兼ね合いを考慮して、これらより大きいパターンは利用していない。

#### 3.2 仮想準連結の利用

プログラム B は A と比べて、いくつかの能率向上の工夫を組み込んでいるが、特に重要なものは仮想準連結の利用である。石の群 *X* と、石の群 *Y* があるとす。このとき、*X* と

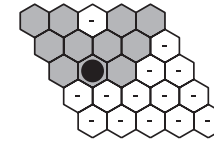


図 8 仮想連結 1  
Fig. 8 A virtual connection (1).

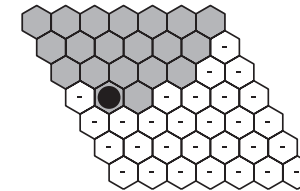


図 9 仮想連結 2  
Fig. 9 A virtual connection (2).

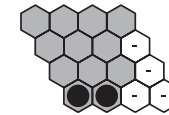


図 10 仮想連結 3  
Fig. 10 A virtual connection (3).

*Y* をつなぐ仮想準連結が 2 つ以上あり、それらの領域が重複しない場合、*X* と *Y* が連結していると判定する。

具体例を示す。図 11 は後手局面である。この盤面では、領域 a と領域 b の 2 つの仮想準連結が成立している。この後、後手が領域 a に着手を行うと、先手は領域 b に着手を行う。また、後手が領域 b に着手を行うと、先手は領域 a に着手を行う。どちらの場合も先手の勝利となる。よって、上端と下端は連結していると判定できる。

#### 3.3 探索法

探索は AND-OR 探索によって行う。ユニオン連結が成立するなら 1、不成立なら 0 を返す。着手候補を評価関数によって評価し、評価値の高い手から順に着手する。着手候補は空セルに打つすべての手に対し、先行排除による着手の限定を行ったものである。

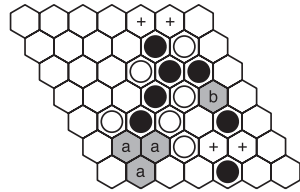


図 11 仮想準連結を用いた連結判定  
Fig. 11 Testing connectivity by virtual semi-connection.

### 3.4 評価関数

着手の順序付けを決める評価関数を説明する。着手の評価項目は次の 5 個である。

- (1) 着手の座標に対する静的および動的な評価。
- (2) 目標の連結を達成した着手の記憶による評価。キラリスト技法の変形である。
- (3) 2 手連続着手して連結を達成できるか否かの評価。
- (4) 一間跳びの切断を守る着手の評価。
- (5) ランダムウォークによる着手の評価。たとえば深度 5 のランダムウォークを 200 回行い、連結の達成回数を勘定する。

これらの評価項目の値を線形結合し、評価値によって着手の順序を決める。線形結合の係数は計算実験により定める。

### 4. 必勝手順とその検証

プログラム A と B の性能評価のため、Hex の必勝手順の検証を行う。Hex の先手必勝手順を示す証明木（解の木）とは次のようなものである。

- 後手局面では、着手可能なすべての手を着手する。
- 先手局面では、1 手のみを着手する。

Hex の必勝手順を求めた多くの論文では、必勝手順を簡潔に記述するため、次のような工夫を行っている。

- 先行排除による後手着手の限定
- ユニオン連結などによる着手のパターン化

Noshita が示した  $7 \times 7$  の盤面の必勝手順の木（図 12）を例にして説明する。根節点の子には 7 つの先手局面がある。すなわち後手の着手候補が 7 手存在する。しかし根節点の盤面には 48 セルの空きがあるため、着手候補も 48 手存在する。根節点で先行排除を行う

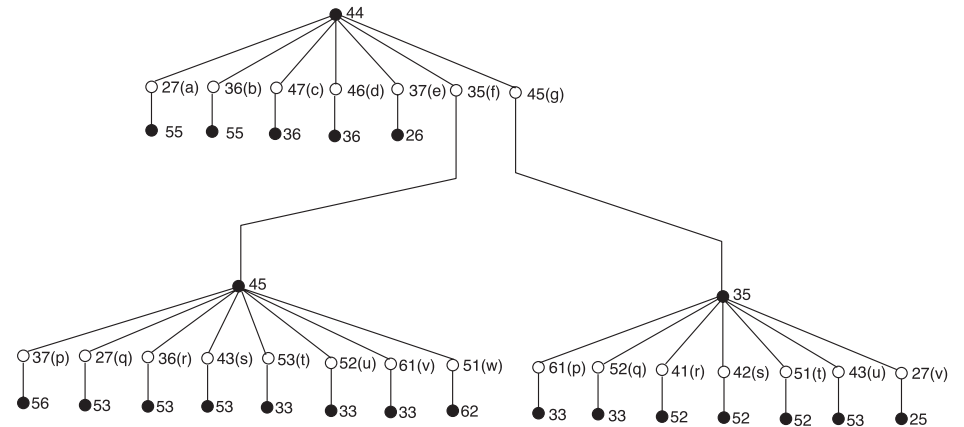


図 12 Noshita が文献 7) で示した  $7 \times 7$  の必勝手順  
Fig. 12 Noshita's winning strategy for  $7 \times 7$ .

と、48 手中 41 手が排除できるため、着手候補から取り除かれている。その他の内部節点の後手局面でも先行排除により、後手の着手候補を限定している。

葉節点の盤面では、実際には上端と下端が連結することが必ずしも明示的に書かれていない。しかしユニオン連結が成立する複数の領域に対する（通常は簡単な）論理計算により、先手の勝利を示している。

本稿のプログラムを用いた必勝手順の検証方法は次のとおりである。内部節点の後手局面では、先行排除により後手の着手を限定している。そこで、先行排除の正しさを検証する。排除された着手の集合によって定まる（一般には）複数の盤面それぞれにおいて、後手から着手を始めて先手勝ちを示す。

葉の節点は後手局面である。この局面では、盤面分割などを用いて先手必勝であることを示している。そこで、まずプログラムを用い、部分盤面内で目標の連結が成立することを確認する。次に部分盤面どうしの連結を手で統合して、その葉が先手必勝であることを示す。

### 5. 性能評価実験

計算実験により、本稿のプログラムの性能評価を行った。使用したコンピュータの CPU は Intel(R) Pentium(R) 4 の 3.06 GHz、メモリは 512 Mbytes である。

5.1 プログラム A の性能評価

2005 年に Noshita が示した  $7 \times 7$  の必勝手順<sup>7)</sup> 全体を、本稿のプログラム A を用いて検証する。この必勝手順 (図 12) では 20 個の葉を含む 23 個の節点に対して検証を行う必要がある。

実験の結果、ほとんどの盤面について、1 秒程度で連結の判定を行うことができた。必勝手順全体が正しいことは合計 1 分未満で検証できた。このことはこの必勝手順の簡単さの程度を示していると解釈できる。

各盤面の空セル数と連結の判定にかかった時間 (time)、探索節点数 (nodes) は表 1 のとおりである。表の盤面の番号は、文献 7) のものである。表の空セル数とは、プログラムに入力として与えた部分盤面における空セルの数である。一般的な傾向としてはこの値が大

表 1  $7 \times 7$  の検証結果  
Table 1 Verifying the positions for  $7 \times 7$ .

検証した盤面	空セル数	time (sec)	nodes
Figure 9	20	1	0
Figure 10	20	1	0
Figure 14	12	1	0
Figure 15	22	1	57
Figure 22-1 <sup>*1</sup>	18	1	7,755
Figure 22-2	18	1	7,100
Figure 23	8	1	90
Figure 24	19	1	924
Figure 11	22	1	186
Figure 12	22	1	3,293
Figure 13-L <sup>*2</sup>	18	1	133
Figure 13-R	8	1	9
Figure 17	24	1	9,149
Figure 18	16	1	15,006
Figure 19	24	1	85
Figure 20	18	1	0
Figure 21	10	1	10
Figure 26	25	16	182,448
Figure 27	24	8	119,246
Figure 28-L	14	1	609
Figure 28-R	14	1	649
Figure 29-L	19	1	919
Figure 29-R	25	1	279

\*1 盤面の番号の横に-1 や-2 とあるものは、一部の石の配置を変えて検証を行ったものである。

\*2 盤面の番号の横に-L や-R とあるものは、盤面分割を用いて検証を行ったものである。

きいほどプログラムによる計算量が大きくなる。nodes が 0 となっている盤面は、探索開始と同時に、仮想連結により勝利が確定した盤面である。

例として、図 13 における先手の勝利を検証する。この盤面では先手の勝利を示すために、図 14 と図 15 のような盤面分割を行っている。そこで、各部分盤面を入力として本稿のプログラムに与え、連結を判定した。その結果、図 14、図 15 とともに 1 秒で目標の連結が確認できた。これにより、図 13 の上端と下端が連結することが確認できた。

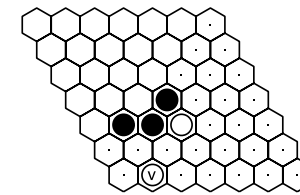


図 13 Figure 29<sup>7)</sup>  
Fig. 13 Figure 29<sup>7)</sup>.

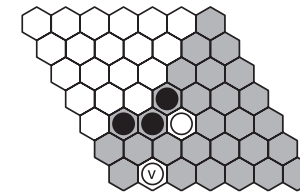


図 14 Figure 29-1  
Fig. 14 Figure 29-1.

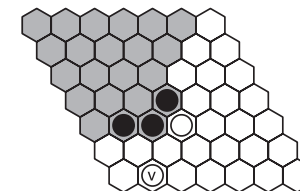


図 15 Figure 29-2  
Fig. 15 Figure 29-2.

5.2 プログラム A とプログラム B の性能比較

2005 年に Noshita が示した  $8 \times 8$  の必勝手順<sup>7)</sup> に使われているユニオン連結の図の一部について、本稿のプログラム A とプログラム B を用いて連結を判定する。それにより、両プログラムの性能を比較する。これらのユニオン連結を選んだ理由は、必勝手順の中で特に難しく、性能評価に適しているからである。

実験の結果、ほとんどのユニオン連結は数分以内で連結の判定を行うことができた。計算時間などは表 2 のとおりである。多くの盤面でプログラム B の方が計算時間が短くなることが分かった。特に Figure 33 では大きな差が出た。

プログラム A と B の差が大きかった Figure 33 (図 16) を例にあげて説明する。

図 16 は黒石 a と下端、黒石 b と下端、もしくは上端と下端が連結するユニオン連結である。この盤面に対し、両プログラムは先行排除によって図 17 のように後手着手を限定する。‘@’ で示したセルが先行排除後に残った着手である。この先行排除は人間の熟練プレイ

ヤ(筆者の 2 人)と同じ結果である。

図 18 は、図 16 から着手を進めた盤面である。白(3,7)に対して黒(5,6),白(4,8)と進む。黒は(5,6)を 1 番目の着手に選択したが、実は正解の着手は(4,6)である。プログラム B では 2 番目にこの正解を選んでいる。この盤面の状況においてプログラム A では先行

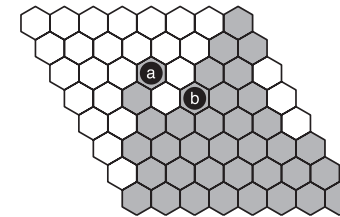


図 16 Figure 33<sup>7)</sup>  
Fig. 16 Figure 33<sup>7)</sup>.

表 2  $8 \times 8$  のユニオン連結の検証結果  
Table 2 Verifying the union-connections for  $8 \times 8$ .

検証した盤面	空セル数	プログラム A		プログラム B	
		time (sec)	nodes	time (sec)	nodes
Figure 32	32	20	38,383	158	27,284
Figure 33	35	56,685 *3	38,002,570	303	47,698
Figure 34	26	4	13,225	4	103
Figure 36-L	22	21	22,582	6	264
Figure 36-R	23	401	435,543	8	3,871
Figure 38-L-A *4	26	4	4,217	8	103
Figure 38-R-A	10	1	595	0	3
Figure 38-L-B	26	3	3,143	11	127
Figure 38-L-X	20	10	51,181	0	1
Figure 38-R-X	12	1	26	0	3
Figure 39-L-A	25	2	796	2	413
Figure 39-R-A	20	175	679,145	0	25
Figure 39-L-B	25	1	295	2	413
Figure 39-R-B	18	84	471,356	0	3
Figure 39-L-C	25	3	1,688	2	373
Figure 40	34	194	388,078	38	708

\*3 プログラム A の評価関数を調整することにより、検証時間 6,351 秒、探索節点数 3,700,794 で検証を行うことができる。

\*4 盤面の番号の横に-A, -B, -C, -X とあるものは、AB-property<sup>7)</sup>を展開して検証を行ったものである。例をあげると、Figure 38-L-A とは、Figure 38 の AB-property を展開した盤面 A を盤面分割したうちの片側の部分盤面である。

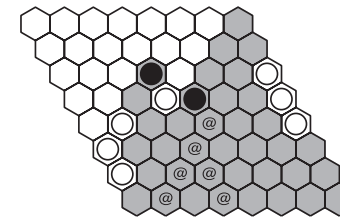


図 17 先行排除後に残った着手

Fig. 17 The remaining moves after the precluding analysis.

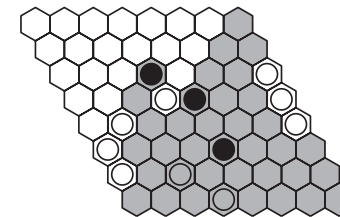


図 18 図 16 から着手を進めた盤面

Fig. 18 A position in the descent of Fig. 16.

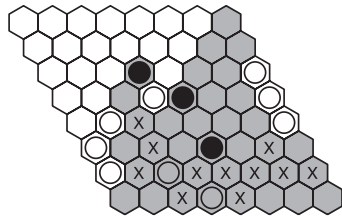


図 19 プログラム B の先行排除  
Fig. 19 The precluding analysis for Program B.

排除ができず、盤上の空セル全 32 手の着手が残る。プログラム B では仮想準連結を組み込んだ部分がいきて、多くのセルを先行排除できた。その結果、図 19 中の 'X' で示したセルへの着手、11 手のみが黒の着手として残る。この先行排除は熟練者と同等、もしくはそれ以上の結果である。これにより着手 (5, 6) が短時間で不正解であると判定できた。図 16 全体では、プログラム A が 56,685 秒かかったのに対し、プログラム B は 303 秒で連結を判定できた。このように仮想準連結を用いたアルゴリズムにより、計算時間の大幅な短縮ができた。

表 2 に示すとおり、特に難しいと思われるユニオン連結の連結が実用的な計算時間で判定できたといえる。

## 6. おわりに

ゲーム Hex において、大きな盤面における必勝手順を導くためには、人の手で盤面分割を行い、コンピュータで細部を検証するなどの共同作業が非常に重要になってきている。そこで、本稿では、部分盤面における連結性を判定する 2 つのプログラム、プログラム A と B を作成した。両プログラムでは、本稿でとりあげた規模の盤面に対して人間の熟練者と同程度の着手選択や先行排除ができた。また、計算時間については、 $8 \times 8$  の必勝手順に対して実用的な時間内で連結判定ができるようになった。

プログラム A を用いて、Noshita が文献 7) で示した  $7 \times 7$  の盤面における必勝手順をすべて検証することができた。各々の部分盤面は多くても十数秒で検証が終わり、必勝手順全体の検証時間の合計は 1 分未満になった。 $8 \times 8$  の必勝手順は、判定が難しいユニオン連結を検証した。それらのほとんどのユニオン連結は数分以内で連結を判定することができた。プログラム A では判定に十数時間かかるユニオン連結があったが、プログラム B では 5 分

程度で判定することができた。両プログラムともに、先行研究では明らかにされていない  $8 \times 8$  以上の盤面に対して、検証を行うことができた。

今後の課題は、もっと大きい盤面に適用するためのプログラムの改良である。前述のように、(人間による解析の複雑さや探索空間の規模という観点から) 特に連結判定が難しいと思われるユニオン連結を、非常に短い実行時間で検証することができた。今後、 $8 \times 8$  の様々な初手に対する必勝手順を発見し検証するために、実用に足る性能を持っているといえる。もっと大きい  $9 \times 9$  や  $10 \times 10$  の盤面に対しては、さらなる改良が必要であろう。その方法として、評価関数の性能向上や、別の探索法(たとえば、df-pn<sup>10)</sup>)の利用などが考えられる。また、多様な観点からプログラムの改良を検討するために、さらに多くの実験を行う必要がある。他の課題として、ユニオン連結による(先手の勝利を証明する)盤面分割を、プログラムで自動検出することがあげられる。

謝辞 丁寧に査読して下さった査読者の方々に感謝の意を表します。

## 参考文献

- 1) Anshelevich, V.: A Hierarchical Approach to Computer Hex, *Artificial Intelligence*, Vol.134, No.1, pp.101-120 (2002).
- 2) Björnsson, Y., Hayward, R., Johanson, M. and Rijswijk, J.v.: Dead Cell Analysis in Hex and Shannon Game, in *Graph Theory in Paris, Proc. Conference in Memory of Claude Berge (GT04 Paris)*, Birkhäuser, pp.45-60 (2007).
- 3) Enderton, B.: Answers to infrequently asked questions about the game of Hex. <http://www.cs.cmu.edu/People/hde/hex/hexfaq/>
- 4) Hayward, R., Arneson, B. and Henderson, P.: Verifying Hex Strategies, *CG*, p.10 (2006).
- 5) Hayward, R., Björnsson, Y., Johanson, M., Kan, M., Po, N. and Rijswijk, J.v.: Solving  $7 \times 7$  Hex with Domination, Fill-in, and Virtual Connections, *Theoretical Computer Science*, Vol.349, No.2, pp.123-139 (2005).
- 6) Herik, H.J.v.d., Uiterwijk, J.W.H.M. and Rijswijk, J.v.: Games Solved: Now and in the Future, *Artificial Intelligence*, Vol.134, No.1, pp.277-311 (2002).
- 7) Noshita, K.: Union-Connections and Straightforward Winning Strategies in Hex, *ICGA Journal*, Vol.28, No.1, pp.3-12 (2005).
- 8) Yang, J., Liao, S. and Pawlak, M.: On a Decomposition Method for Finding Winning Strategy in Hex Game, *International Conference on Application and Development of Computer Games in the 21st Century*, pp.96-111 (2001).
- 9) 三島 健, 櫻井英俊, 野下浩平: Hex の必勝手順に対する新証明技法とその応用, *Proc. 11th Game Programming Workshop*, pp.136-142 (2006).



- 10) 長井 歩, 今井 浩: df-pn アルゴリズムの詰将棋を解くプログラムへの応用, 情報処理学会論文誌, Vol.43, No.6, pp.1769-1777 (2002).

(平成 20 年 3 月 21 日受付)

(平成 20 年 9 月 10 日採録)



三島 健 (学生会員)

2005 年電気通信大学情報工学科卒業。2007 年同大学院情報工学専攻博士前期課程修了。現在, 同博士後期課程に在籍。アルゴリズムの計算量解析, 組合せゲームの理論と実験に関する研究に従事。



櫻井 英俊 (学生会員)

2006 年電気通信大学情報工学科卒業。2008 年同大学院情報工学専攻博士前期課程修了。同年日立ソフトウェアエンジニアリング株式会社入社。



吉元 昭裕

2008 年電気通信大学情報工学科卒業。同年株式会社ソフメディア入社。



野下 浩平 (正会員)

1943 年生。1966 年東京大学工学部計数工学科卒業。現在, 電気通信大学情報工学科教授, 工学博士。アルゴリズムの計算量解析, 組合せゲームの理論と実験に興味を持つ。