



Title	局面ネットワークを利用したHexゲーム戦略の提案
Author(s)	本庄, 将也; 鈴木, 育男; 山本, 雅人 他
Citation	情報処理北海道シンポジウム講演論文集, 2010, 37-42
Issue Date	2010-10-02
Doc URL	<a href="https://hdl.handle.net/2115/51049">https://hdl.handle.net/2115/51049</a>
Rights	ここに掲載した著作物の利用に関する注意 本著作物の著作権は情報処理学会に帰属します。本著作物は著作権者である情報処理学会の許可のもとに掲載するものです。ご利用に当たっては「著作権法」ならびに「情報処理学会倫理綱領」に従うことをお願いいたします。
Type	journal article
File Information	Hokkaidosympo20103742.pdf



## 局面ネットワークを利用した Hex ゲーム戦略の提案

本庄将也\* 鈴木育男 山本雅人 古川正志

(北大情報科学)<sup>†</sup>

## 1 はじめに

Hex は簡単なルールの 2 人用ボードゲームである。六角形を敷き詰めた菱形の盤面を使い、先にこの菱形の対角を自分の駒でつなげたプレイヤーが勝利する (図 1)。

Hex は二人零和有限確定完全情報ゲームに分類され、チェスや囲碁のように戦略的な奥深さがあるゲームとして知られている。ゲームの特徴としては、既存研究から、必ず勝敗が付くこと (no draws)、先手に必勝戦略があることがわかっている [Browne 00]。この先手が有利であることから、後手が希望すれば、先手の 1 手目を後手の 1 手目として置き換えることができるルール (スワップルール) が存在する。このルールを適用すると、後手に必勝戦略が存在する。

必勝戦略が存在が証明されてからは、必勝戦略を求める研究が行われている。必勝戦略を判定する問題のクラスは PSPACE-完全 [Browne 00] であるので、直接すべての局面について調査することは難しいが、様々な工夫により現在では  $8 \times 8$  のすべての初手における勝利プレイヤーが示されており [Henderson 09]、 $9 \times 9$  も多くの初手における勝利プレイヤーが示されている。

また、コンピュータ Hex の研究も行われている。有名なものには Anshelevich が開発した、数手先の局面の情報をもつ仮想リンク (Virtual Connections) を導入した Hexy [Anshelevich 00] というプログラムがあり、現在の Hex コンピュータに大きな影響を与えている。他には、囲碁の探索アルゴリズムとして有名になった、モンテカルロ木探索の改良である、UCT 探索を用いた Hex コンピュータ [Arneson 08] が登場し、研究が進められている。

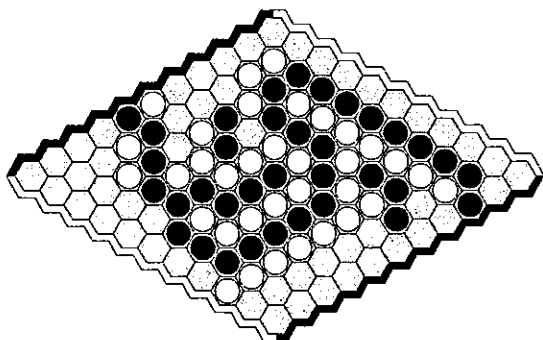


Fig. 1  $11 \times 11$  の Hex の終了図 (黒プレイヤーが勝利)

Hex の任意の盤の状態は、自身と対戦相手の 2 つのネットワーク  $N_B(V, E)$  と  $N_W(V, E)$  によって表現可能なことが知られている。各ネットワークは、セルを表すノード集合  $V$  と盤面の端を表す辺ノード  $(s, t)$ 、セルや盤面の端の隣接関係を表すリンク集合  $E$  によって構成される。ネットワーク表現では、勝利条件は自分のネットワークの辺ノード  $(s, t)$  間にリンクが存在すること、または、相手のネットワークの辺ノード  $(s, t)$  間のパスがないこととなる。つまり、自分の辺ノード  $(s, t)$  間の最短経路長が短くなる、または、相手の辺ノード  $(s, t)$  間の最短経路長が長くなるようなネットワーク形状になるように手を打つ戦略が有効であると考えられる。

しかし、最短経路が 1 つしかない場合や、すべての最短経路に共通するノードが含まれる場合は、相手の手でそのパスが使えなくなってしまう可能性が高いため、最短経路の種類が多さや最短経路に続いて短いパスについても考慮しなければならない。

本研究では、辺ノード  $(s, t)$  間の短いパスに多く寄与するノードの指標として  $k$ -寄与度を定義し、 $k$ -寄与度が高いノードを手として選択する戦略と Hex の熟練者の戦略の関係性を調査する。また、調査結果から得られた、熟練者の手の特徴を反映させた局面評価関数  $E_k$  を作成し、性能評価を行う。

## 2 関連研究

## 2.1 Hex のネットワーク表現

Hex の任意の盤の状態は黒プレイヤーと白プレイヤーの 2 つのネットワーク  $N_B$  と  $N_W$  に変換することができる [Rijswijk 03]。黒プレイヤーのネットワーク  $N_B$  は次の手順で作成できる。また、白プレイヤーのネットワーク  $N_W$  も同様の手順で作成できる。

1. セルをノードとする。ノード集合を  $V$  とする。
2. 盤上で隣接しているノード間にリンクを追加する (図 3)。
3. 向かい合う 2 つの黒のボードの辺をダミーのノード  $s, t \in V$  と設定し、ノード  $s, t$  と隣接するノードとの間にリンクを追加する。
4. 黒のセルはリンクで接続されているノード同士を完全グラフにし、そのセルのノードとリンクをネットワークから除去する。
5. 白のセルはそのノードとリンクをネットワークから除去する。

例えば、図 2 は図 4 の黒プレイヤーのネットワークと図 5 の白プレイヤーのネットワークに変換される。

\* honjyo@complex.eng.hokudai.ac.jp  
<sup>†</sup> 札幌市北区北 14 条西 9 丁目北海道大学大学院情報科学研究科

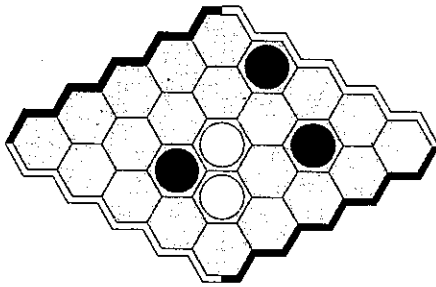


Fig. 2 5 × 5 の Hex の例

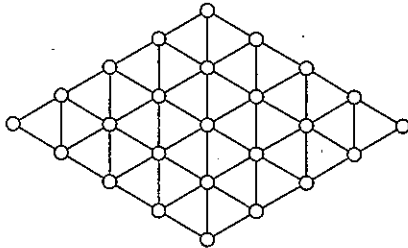


Fig. 3 図2の状態をネットワーク表現に変換する過程 ((2)の状態)

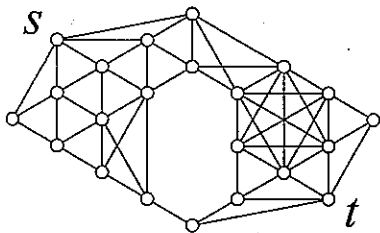


Fig. 4 図2の黒プレイヤーのネットワーク

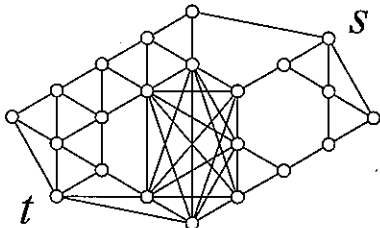


Fig. 5 図2の白プレイヤーのネットワーク

2.2 仮想リンク (Virtual Connections)

仮想リンク (Virtual Connections)[Anshelevich 00]とは、Anshelevich が開発した、最良手を指せば必ず接続できる2つのセル間にリンクを追加する方法である。

本研究では、ネットワークに先読みの情報を付加するために使用する。

2.2.1 仮想リンクの定義

自分の色の隣接しているセル (グループ)  $x$  と  $y$ 、またはグループ  $x$  と空きセル  $y$ 、または空きセル  $x$  と  $y$  が存在したとき、たとえ相手が先に手を打ったとしても、自分の  $x$  と  $y$  の接続を防ぐことができないならば、 $x$  と  $y$

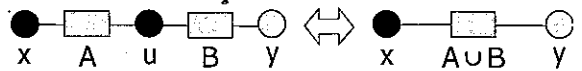


Fig. 6 The AND deduction rule

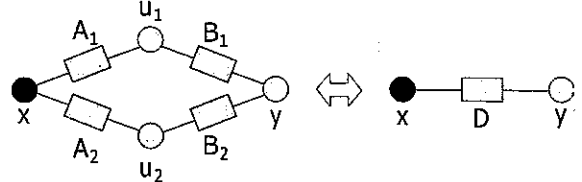


Fig. 7 The OR(2) deduction rule

は仮想リンクを形成する。仮想リンク  $V$  は  $(x, A, y)$  と表記される。  $x, y$  は仮想リンクが作られる両端セル、もしくはグループであり、  $A$  は仮想リンク形成に必要なセル集合 (キャリア) を表す。

2.2.2 仮想リンクの発見手法

仮想リンクの多くは次の2つのルールを繰り返し適用することにより発見できる。

AND Deduction Rule

仮想リンク  $(x, A, u)$  と仮想リンク  $(u, B, y)$  があるとき、  $u$  が自分のセルかグループであり、  $A \cap B = \emptyset$  ならば  $(x, A \cup B, y)$  もまた仮想リンクである (図6)。

$u$  が自分のセルで  $(x, u)$  間と  $(u, y)$  間の接続が保証されているため、  $(x, y)$  間の接続も当然保証される。

OR(n) Deduction Rule

仮想リンク  $(x, A_k, u_k)$  と仮想リンク  $(u_k, B_k, y)$  ( $k = 1, 2, \dots, n$ , for  $n > 1$ ) があるとき、すべての  $k$  に対して、  $u_k$  が空きセルであり、以下の条件 (図2)~(図3) をすべて満たすならば、  $(x, D, y)$  もまた仮想リンクである (図7)。

$$x \cap B_k = \emptyset \text{ かつ } y \cap A_k = \emptyset \quad (1)$$

(for all  $k = 1, 2, \dots, n$ )

$$A_k \cap B_k = \emptyset \quad (for \text{ all } k = 1, 2, \dots, n) \quad (2)$$

$$\bigcap_{k=1}^n C_k = \emptyset \quad (3)$$

ただし、

$$C_k = A_k \cup u_k \cup B_k \quad (4)$$

$$D = \bigcup_{k=1}^n C_k \quad (5)$$

である。

これらは、すべての  $u_k$  に対して  $(x, u_k)$  間と  $(u_k, y)$  間の接続が保証されている。さらに、相手に  $u_l$  ( $l \in \{1, 2, \dots, n\}$ )

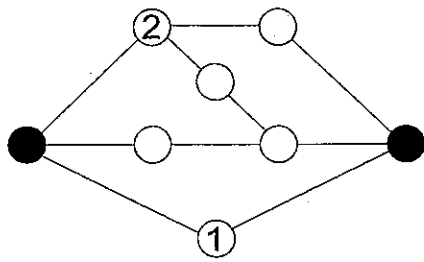


Fig. 8 And ルールと Or ルールでは見つけれない黒の仮想リンクの例

に手を打たれたとしても、 $k \neq l$ な  $u_k$  に手を打てば AND Deduction Rule を適用できる状態になるので、 $(x, y)$  間の接続は保証されている。

ただし、この2つのルールを繰り返し適用しても発見できない仮想リンクが存在することも知られている。例えば、図8の状態の左端の黒ノードと右端の黒ノードの接続を考える。相手がノード1に打ってきた時はノード2に打てば仮想リンクができ、相手がノード2に打ってきた時はノード1に打てば接続する事ができる。しかし、図8の状態に AND Deduction Rule と OR(n) Deduction Rule を適用しても、仮想リンクを1本も発見できない。

### 3 $k$ -寄与度

$k$ -寄与度とはノード  $s$  とノード  $t$  の接続にノード  $u \in V(u \neq s, u \neq t)$  がどの程度寄与しているかを示す指標である。 $k$ -寄与度を定義するために、まず  $k$ -最短経路集合を定義する。

#### $k$ -最短経路集合 $S_k$

ノード  $(i, j)$  間の最短経路長を  $d(i, j)$  としたとき、経路長が  $d(i, j) + k$  のノード  $(i, j)$  間の経路をノード  $(i, j)$  間の  $k$ -最短経路集合  $S_k$  とする。

$k$ -最短経路集合  $S_k$  を用いて、ノード  $s$  とノード  $t$  の接続に対するノード  $u$  の  $k$ -寄与度  $\delta_{stk}(u)$  を次の式で定義する。

$$\delta_{stk}(u) = \frac{\sum_{i=0}^k \sigma_{st_i}(u)}{\sum_{i=0}^k \sigma_{st_i}} \quad (6)$$

$\sigma_{stk}$  はノード  $(s, t)$  間の  $k$ -最短経路集合  $S_k$  の要素数で、 $\sigma_{st_i}(u)$  はノード  $(s, t)$  間の  $k$ -最短経路集合  $S_k$  の経路にノード  $u$  が含まれる回数である。ノード  $(s, t)$  間の  $k$ -最短経路集合  $S_k$  の要素数が0の場合  $\delta_{stk}(u) = 0$  とする。

ノード  $(s, t)$  間の  $k$ -最短経路集合  $S_k$  は、ノード  $(s, t)$  間の経路長と接続の冗長性の指標であり、Hex では、相手

にノード  $(s, t)$  間の接続を妨害されたとき、ノード  $(s, t)$  間の経路長がどう変化するかを予測することができると考えられる。

## 4 熟練者が選択するノードとそのノードの $k$ -寄与度の関連調査実験

仮想リンクを付加したネットワークについて、熟練者が選択するノードと、そのノードの  $k$ -寄与度の関係を数値計算実験で調査する。

### 4.1 実験条件

#### 4.1.1 使用する熟練者の手のデータ

使用する棋譜データは Richard Rogalie's Gamerz.NET server 上でされた対戦のうち、以下の条件を満たす 22 ゲーム分の棋譜とする [Browne 00]。

- ゲーム盤は  $11 \times 11$  とする。
- 対戦プレイヤーはこのサーバ上の上位 5% 以内のプレイヤーである。
- 先手は必ず黒色の駒を使用する。
- ゲームは勝敗がつくまで行われる。つまり、どちらかのプレイヤーが両端を繋いだか、勝利が明らかになるまで続けられる。
- このゲームに既知の戦略的な手筋が見られる。

この棋譜の勝利プレイヤーの手を熟練者の手のデータとして使用する。

#### 4.1.2 使用するネットワーク

勝利したプレイヤーから見て、自分の盤面と相手の盤面に仮想リンクを付加した状態から 2.1 の手法で得られる 2 つのネットワークを用いて実験する。仮想リンクサーチアルゴリズムには H-search アルゴリズム [Anshelevich 02] を使用した。

#### 4.1.3 仮想リンク付きネットワークにおける $k$ -最短経路集合

例えば、ノード  $(s, t)$  にノード  $a$  とノード  $b$  から構成される仮想リンクが存在する場合 (図9)、通常の方法では、ノード  $(s, t)$  間の 0-最短経路集合は仮想リンクを使う経路となり、ノード  $(s, t)$  間の 1-最短経路集合はノード  $a$  を通過する経路とノード  $b$  を通過する経路となる。しかし、ノード  $a$  を通過する経路とノード  $b$  を通過する経路は、仮想リンクを構成するノードに含まれているので、1-最短経路としては機能していない。

このような経路を除外するために、 $k$ -最短経路集合を求めるときに、以下のルールを適用する。

- 経路の通過ノードと仮想リンクの構成ノードの集合を  $S_n$  とする。新しくノード  $(i, j)$  間の  $s$  ( $s \geq 1$ )-最短経路が見つかったとき、その経路の  $S_n$  である  $S_{n_{new}}$

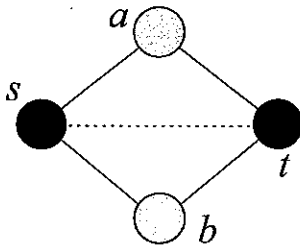


Fig. 9 ネットワークに仮想リンクを加えた例。破線が仮想リンクであり、この仮想リンクはノード a, b(キャリア)から成る

とノード  $(i, j)$  間の  $t < s$  なすすべての  $t$ -最短経路の  $S_n$  の集合  $S_{S_n}$  について、 $S_{n_{new}} \subseteq S_{n_c} (S_{n_c} \in S_{S_n})$  を満たす  $S_{n_c}$  が1つも存在しないならば、新しい経路として受理する。

4.1.4  $k$ -寄与度の評価方法

$k$ -寄与度によるノードのランク付けをしたときにランクが上位のノードほど重要なノードであると考え、勝利した熟練者の手が上位のノードとどの程度一致するかで評価する。

ランク関数  $R_\delta(k, u, N)$

ランク関数  $R_\delta(k, u, N)$  をネットワーク  $N$  のすべてのノードの  $k$ -寄与度を計算し、ノード  $u$  の  $k$ -寄与度の順位を返す関数とする。その値域は  $1 \leq R_\delta(k, u, N) \leq n$  ( $n$ : ネットワークのノード数) の整数である。

評価  $Q$

評価  $Q$  は  $R_\delta(k, u, N)$  に熟練者の手  $v$  が代入された時の  $R_\delta(k, v^*, N)$  を正規化した値であり、次の式で表される。

$$Q = 1 - \frac{R_\delta(k, v^*, N) - 1}{n - 1} \quad (0 \leq Q \leq 1) \quad (7)$$

評価  $Q$  は、ある盤面において  $k$ -寄与度がランク付けした結果と熟練者が選んだ手がどれ程違うか評価している。  $Q = 0$  のときに熟練者は最下位のランクのノードを選択し、  $Q = 1$  のときに1位のランクのノードを選択している。

4.2 実験結果

まず、22 ゲーム分の自分と相手の仮想リンクを付加したネットワークを用いた時の1-寄与度の評価  $Q$  をターン数ごとにプロットした2つのグラフを図10, 図11に示す。また、  $k = 0$  の場合は図10, 図11より  $Q$  が低

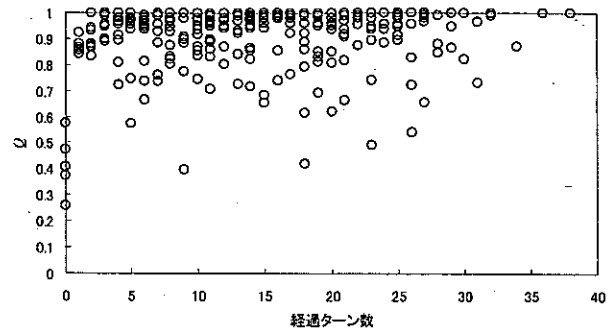


Fig. 10 自分のネットワークの  $k=1$  の熟練者が選択したノードの評価  $Q$

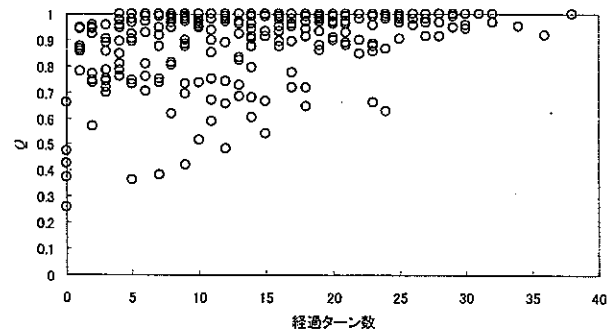


Fig. 11 相手のネットワークの  $k=1$  の熟練者が選択したノードの評価  $Q$

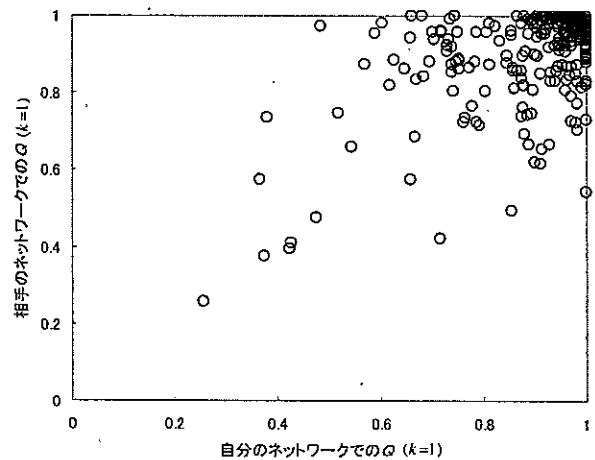


Fig. 12 自分のネットワークでの評価  $Q$  と相手のネットワークでの評価  $Q$  の相関図

い場合が多く、  $k = 2$  の場合は図10, 図11と大きな違いは見られなかった。

次に、熟練者の選択するノードの、自分のネットワークでの評価  $Q$  と相手のネットワークでの評価  $Q$  の相関図を図12に示す。

### 4.3 考察

図 10, 図 11 より, 多くの場合, 評価  $Q$  は高い位置に集中しているがわかる。これより, 1-寄与度が高いノードは, 多くの場合, 戦略的に重要なノードであることが言える。しかし, 評価  $Q$  が低くても選択する価値のある手が存在することがわかる。これは, H-search と 1-寄与度を用いた評価  $Q$  の組み合わせでは発見できない良い手の特徴が存在することを示している。

また, 図 12 より, 自分のネットワークと相手のネットワークの両方で評価  $Q$  の高いノードを選択する傾向があることがわかる。さらに, 自分のネットワークより相手のネットワークの評価  $Q$  を重視している傾向が見られるが, 僅かな差であるので, 自分のネットワーク上で手を考えること, つまり, 攻めることと, 相手のネットワーク上で手を考えること, つまり, 守ることは, どちらも重要であると考えられる。

### 5 ノード $(s, t)$ 間の $k$ -最短経路集合 $S_k$ を用いた局面評価

調査により得られた, 次の特徴を反映させた評価方法を提案する。

- 最短経路長だけでなく,  $k$ -最短経路集合も考慮する。
- 攻めることと守ることの両方を重視する。

#### ノード $(s, t)$ 間の $k$ -最短経路集合 $S_k$ を用いた局面評価関数 $E_k$

ネットワーク  $N_B$  と  $N_W$  の  $k$ -最短経路  $S_{k_B}$ ,  $S_{k_W}$  を用いて評価  $E_k$  を次のようにする。

$$E_{k_W} = d_W(s, t) - \sum_{i=0}^k x^{i+1} n_{i_W} \quad (8)$$

$$E_{k_B} = d_B(s, t) - \sum_{i=0}^k x^{i+1} n_{i_B} \quad (9)$$

$$E_k = \frac{E_{k_W}}{E_{k_B}} \quad (10)$$

$n_k$  は  $S_k$  の要素数を表す。  $x$  は経路長  $l$  の経路の価値と経路長  $l+1$  の経路の価値の比を表す係数である。例えば, 経路長  $l+1$  の経路  $n$  本の価値と経路長  $l$  の経路 1 本の価値が等しいとするならば,  $x = 1/n$  となる。  $E_{k_B}$ ,  $E_{k_W}$  がともに負にならないような値を設定する。

この評価手法では,  $E_k = \infty$  のときは黒の勝利であり,  $E_k = 0$  のときは白の勝利である。

### 6 局面評価関数 $E_k$ の性能調査実験

$k$ -最短経路集合を考慮する評価関数と, 考慮しない評価関数のどちらが良いか調べるために実験を行う。

$k$ -最短経路集合を考慮しない評価関数は以下の評価関数とする。

#### ノード $(s, t)$ 間の最短経路長を用いた評価関数 $E$

ネットワーク  $N_B$  と  $N_W$  のノード  $(s, t)$  間の最短経路長を  $d_B(s, t)$ ,  $d_W(s, t)$  したとき, 評価  $E$  を次のように定義する。

$$E = \frac{d_W(s, t)}{d_B(s, t)} + r \quad (0 \leq E \leq \infty) \quad (11)$$

$E = \infty$  のときは黒の勝利であり,  $E = 0$  のときは白の勝利である。  $r$  は, 同じ評価値の手が存在する場合にランダムに選択させるための小さな値の乱数である。

### 6.1 実験条件

#### Hex の設定

コンピュータ Hex においてよく用いられる盤面のサイズは  $11 \times 11$  であるが,  $11 \times 11$  の評価は計算時間が多くかかり, また, 評価関数の有効性を確かめるためには,  $7 \times 7$  の盤面であれば十分と考えられるので, 本実験では  $7 \times 7$  の盤面を使用した。また, スワップルール (後手が希望すれば, 先手の 1 手目を後手の 1 手目として置き換えることができるルール) は使用しない。

#### 状態探索

現在の状態から 1 手先までゲーム木探索を行い, 一番良い評価の局面になるノードを選択する。

黒プレイヤーの具体的な処理の流れを以下に示す。白プレイヤーも同様の処理を行う。

1. 現在の状態に H-search から得られる仮想リンクを追加した後, ネットワーク  $N_B$ ,  $N_W$  を生成する。
2. 手を打てるノードをリストに格納する。
3. リストからノードを取り出し, そのノードに手が打たれた局面に差分 H-search から得られる仮想リンクを追加した後, ネットワーク  $N'_B$ ,  $N'_W$  を作る。
4. ネットワーク  $N'_B$ ,  $N'_W$  の評価値  $e$  を評価関数から求める。
5. 探索した中で一番良い評価値  $e$  ならば候補ノードとする。
6. リストを確認し, リストが空ならば候補ノードに手を打つ。空でなければ (3) に戻る。

#### 使用する評価関数

評価関数は  $k$ -最短経路集合を考慮しない評価関数である  $E$  と,  $k$ -最短経路集合を考慮した評価関数である  $E_0$ ,  $E_1$ ,  $E_2$  の 4 種類を使用する。また, 今回の実験では, “経路長  $l$  の経路の価値”  $\gg$  “経路長  $l+1$  の経路の価値” と仮定し,  $E_k$  の係数  $x$  を違う経路長に影響のない  $x = 0.00001$  とした。

Table 1  $E$  と 100 ゲーム対戦させたときの  $E_k$  の勝率

	$E_0$	$E_1$	$E_2$
$E_k$ が先手	0.95	0.94	0.97
$E_k$ が後手	0.50	0.55	0.56

Table 2  $E_k$  同士,  $k$  を変えて対戦させた結果 (先手が勝利したときは○, 後手が勝利したときは×)

	$E_0$ (後手)	$E_1$ (後手)	$E_2$ (後手)
$E_0$ (先手)	×	×	×
$E_1$ (先手)	×	×	×
$E_2$ (先手)	×	×	×

## 6.2 実験結果

$E$  と  $E_k$  を 100 ゲーム対戦させた結果を表 1 に,  $E_k$  同士,  $k$  を変えて対戦させた結果を表 2 に示す.

## 6.3 考察

表 1 より,  $E_k$  が先手のときは, ほぼ必ず  $E_k$  が勝利し,  $E_k$  が後手のときは, 半分以上  $E_k$  が勝利してことがわかる. Hex では先手に必勝戦略があることを考慮すると,  $E_k$  は  $E$  より, かなり良い手を発見できていると考えられる. また,  $k$  を大きくすると  $E_k$  が先手のときは勝率はほぼ変わらず,  $E_k$  が後手のときは  $k$  が大きくなるにつれ勝率は少し増えた. これより,  $k$  は大きいほうがより正確な評価になると考えられる. また,  $E_k$  が先手の時に勝率が 1 になっていないので, 必勝戦略を見つけることは出来ていないこともわかる.

$E_k$  同士の対戦は,  $k$  の値によらず, すべて後手が勝利した (表 2). 棋譜を見ると, 序盤は  $k$  の値の変化による選択ノードの変化が起らず, 勝利が確定した後の手の打ち方が違うだけであった. よって, この手の流れでは後手が有利な状況が生まれやすい評価関数であることが言え,  $k$  の値の違いによる優位性はわからない.

## 7 まとめ

本研究では, Hex の任意の局面が黒と白の 2 つネットワークとして表現出来ることから, 有効な戦略はノード  $(s, t)$  間の経路長を効率よく下げることと最短経路以外の迂回路も考慮することであると考えた. 次に, この仮説と熟練者の手筋が一致するか検証するために, ノード間の  $k$ -最短経路上でのノードの媒介性を表す指標である  $k$ -寄与度を提案し,  $k$ -寄与度と熟練者の手を比較することにより,  $k$ -寄与度が高いノードを選択する傾向があることがわかった. 得られた結果を元に,  $k$ -最短経路集合を考慮した評価関数  $E_k$  を提案し,  $k$ -最短経路集合を考慮しない評価関数より優れていることを実験により示した.

今後の課題としては, この評価関数の苦手とする展開を解析し, 更新することや,  $k$ -寄与度を評価関数に取

り込む方法の考案などが挙げられる.

## 参考文献

- [Browne 00] Cameron Browne. Hex Strategy: Making the Right Connections. A K Peters Ltd, illustrated edition edition, May 2000.
- [Henderson 09] Philip Henderson, Broderick and Arneson, Ryan B Hayward. Solving  $8 \times 8$  Hex. IJ-CAI'09: Proceedings of the 21st international joint conference on Artificial intelligence, pp.505-510. 2009
- [Anshelevich 00] Vadim V. Anshelevich. The game of hex: An automatic theorem proving approach to game programming. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 189-194. AAAI Press / The MIT Press, 2000.
- [Arneson 08] B. Arneson, R. Hayward, and P. Henderson. Wolve 2008 wins Hex Tournament. ICGA Journal, 32(1):49-53, March 2009.
- [Rijswijk 03] Jack van Rijswijk. Search and evaluation in hex. Technical report, 2003.
- [Anshelevich 02] V. Anshelevich. A hierarchical approach to computer hex. Artificial Intelligence, Vol. 134, No. 1-2, pp. 101-120, January 2002.