

モンテカルロ法を用いたボードゲームの対戦プログラム 双対グラフ上の高速プレイアウト手法の提案 Monte Carlo Methods for a Board Game – Efficient Playout Method on Dual Graph –

情報工学専攻 神野 雅俊
JINNO Masatoshi

1. はじめに

ボードゲームは古くから多くの人に遊ばれ、今もなお老若男女に親しみがある。現在有名なボードゲームとして囲碁、チェス、オセロなどが挙げられ、人と人に限らず、人とコンピュータが対局できるソフトウェアも多く開発されている。近年は、強いソフトウェアを製作するために、様々なアルゴリズムが研究され、その強さは日進月歩の勢いである。記憶に新しい人間とソフトウェアとの対局として、2013年3月から4月にかけて行われた「第2回将棋 電王戦」がある。この将棋棋戦は、プロ棋士5人とコンピュータソフト5本が対局する団体戦であり、コンピュータは3勝1敗1分けという好成績を残している。

本研究は、ボードゲーム Hex に対するモンテカルロ法をベースとしたアルゴリズムに用いる新たな高速プレイアウト手法を提案する。提案するプレイアウト手法は、盤面を双対グラフとして捉えたデータ構造を利用し、勝利に必要なパスの探索を行うことで高速化を図っている。

2. Hex

2.1. Hex のルール

Hex は図1のような六角形のセルが敷き詰められた盤面で2人で遊ぶゲームである。両プレイヤーは各自の色を持ち、先手は黒、後手は白となる。盤面上の何も置かれていないセルに、交互に各自の色の石を打ち、先手は盤面の左上と右下を、後手は盤面の右上と左下を繋げることで勝利となる。図1は白の勝利したゲームである。

図1は、盤面のサイズが 5×5 の Hex である。本論文では、記述の無い限り、盤面のサイズを 11×11 とする。これは、国際コンピュータゲーム協会 (International Computer Games Association;

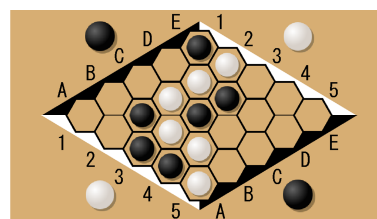


図 1. 5×5 の盤面・白の勝利したゲーム

ICGA) が主催する、ボードゲームプログラム同士を戦わせる世界大会「ICGA トーナメント」のルールに基づいている。

2.2. Hex の特徴

Hex は、デンマークの数学者 Piet Hein によって1942年に考案されたゲームである。また、Piet Hein とは別に、アメリカの数学者 John Nash が1948年に Nash という名前で同じゲームを独立に考案した。その後、科学雑誌に取り上げられたことをきっかけに有名になった。

Hex には、以下の特徴がある、

- 引き分けが存在しない、
- 先手必勝である、
- 必勝手順は不明である。

先手必勝であることは、Nash が strategy stealing の手法を用いて証明した。しかしこの手法では、先手必勝のゲームであることを証明できるが、その手順については手がかりが得られない。

Hex は先手必勝であるため、パイルールという、先手と後手の優劣の差を小さくするためのルールを2.1節のルールに付与することがあるが、本論文にはこのルールを適用しない。

3. モンテカルロ法

ボードゲーム AI に適用されるモンテカルロ法は、評価関数を必要とせず、囲碁などのボードゲームで多大な成果をあげている手法であり、Hex に

図4, 5は, UCTのゲーム木のモデル図になる.
 図4の左のノードを展開すると, 図5のようになる.

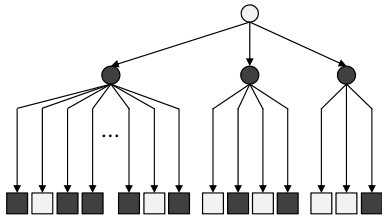


図4. UCTのゲーム木のモデル. 左のノードが閾値に達したため展開する.

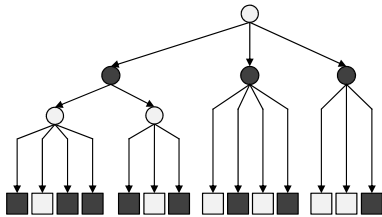


図5. UCTのゲーム木のモデル. 左のノードは展開された

3.4. All Moves As First (AMAF)

シミュレーションの精度は, プレイアウト数に大きく影響され, プレイアウト数が多いほど精度は高くなることが知られている. しかし, プレイアウトの回数を増やすと計算時間が増えてしまう. そこで, All Moves As First (AMAF) が考案された. AMAFは, 1度のプレイアウトの結果を, 木探索で辿ったノード以外に, ランダムゲームシミュレーションの終局盤面に到達できるノードにも反映させる手法である.

例えば, 図6の盤面から, 黒C3, 白E1, 黒D1, 白E5のゲーム木探索を経てランダムゲームシミュレーションを行い, 図7の盤面になったとする. AMAFを用いていないUCTであれば, 白E5, 黒D1, 白E1, 黒C3のノードのみ結果を反映するが, AMAFを用いた場合は, これら4つのノードの兄弟ノードに図7の局面に至るようなノードに対しても結果を反映する. 具体的には, 黒D1の兄弟ノードの1つ黒A2に打つノードなどになる. このようにして1度のプレイアウトの結果を多くのノードに反映させることで少ないプレイアウト数でゲーム木の展開が早くなり, より高精度な探索が可能になる.

4. プレイアウト

本研究は, 新たな高速プレイアウト手法を提案する. 本章では, Hexにおいて主流とされているプレイアウトと提案する高速プレイアウト手法について記述する.

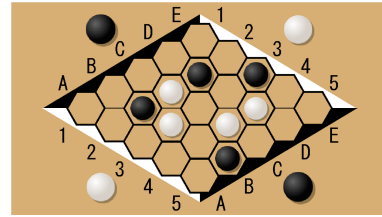


図6. 現在の盤面

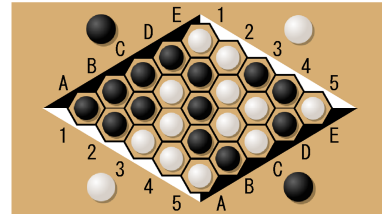


図7. 図6の盤面から, 黒C3, 白E1, 黒D1, 白E5のゲーム木探索, プレイアウトを行った時点の盤面

4.1. 既存研究

Hexにおいて主流とされているランダムゲームシミュレーションは, ゲームの進行と同様に行うものである. 具体的には, 空のセルからランダムに1つ選び黒を置く, 続いて空のセルからランダムに1つ選び白を置く, ということを続ける. しかし, シミュレーションの終了は, どちらかが勝利したときではなく, 空のセルがなくなったときとなる. 1手置くごとに終了判定をするよりも, 空セル数の回数だけ乱数を生成し, 1度終了判定をする方が平均的な処理時間は短いからである.

4.2. 高速プレイアウト手法

既存の主流とされるプレイアウトは, 空のセル全てに対して乱数生成を行い, その後勝敗判定をする. このため, 勝敗に関係ない処理が多いという問題を内包している. この問題に対して, 勝敗判定を行いながら, ランダムゲームシミュレーションを行えるプレイアウト手法を提案し, プレイアウトの高速化を図る.

勝敗判定は, セルとセルの間を辿り盤面のどこに行き着くかを調べる手法をとっている. 図8は, 空セルがなくなりゲームが終了した盤面である. 盤面を左端から見て, 左側が黒, 右側が白となるようにセルとセルの間を縫うように進む. 進み続けると, 盤面の下もしくは上に抜ける. このとき, 下に抜けたときは黒が勝利しており, 上に抜けたときは白が勝利している.

これを, ゲームが終了していない盤面に当てはめる. 例えば, 図9の勝敗判定を開始すると, B2

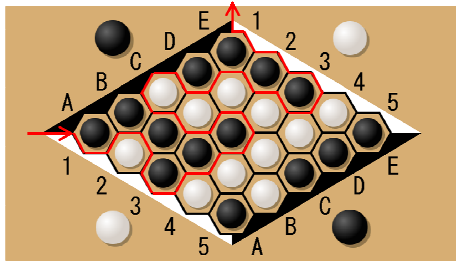


図 8. 勝敗判定方法．上に抜けているので白の勝利と判断できる．の空セルに到達する．ここで，空セルに 0.5 の確率で黒を，0.5 の確率で白を置き，その後，勝敗判定を続行する．この手法により，ランダムゲームシミュレーションを従来の手法よりも少ない処理数で行うことができる．

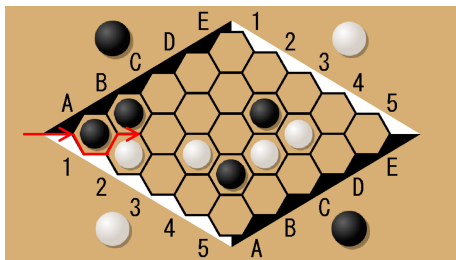


図 9. 高速プレイアウト手法．勝敗判定中に空セルが出現したとき，0.5 の確率で黒もしくは白を置く．

5. 計算機実験

5.1. 高速プレイアウト手法の評価

提案する高速プレイアウト手法がどの程度実用性があるかを調べる．実験方法は，従来のプレイアウトを用いた UCT と高速プレイアウト手法を用いた UCT を 1000 局対局させ，勝率を調べる．次の手を決定するまでの計算時間は 20 秒とする．

実験結果の結果，次のようになった．数値は先手の勝率 [%] を示す．

表 1. 2 つのプレイアウト手法による UCT の勝率

先手 \ 後手	従来	高速
従来	53.2	36.8
高速	74.6	53.1

従来のプレイアウトによる UCT に比べ，高速プレイアウト手法を用いた UCT は，手番に依らず高い勝率を得ることができることが読み取れる．

5.2. 高速プレイアウト手法を適用した AMAF の評価

AMAF を適用した UCT の実用性を測る実験を行った．従来のプレイアウトを用いた UCT と高速プレイアウト手法を用いた UCT に対して，AMAF

の有無により勝率がどのように変化するかを調べる．AMAF を適用していない対戦プログラム同士の対局は 1000 局，AMAF を適用した対戦プログラム同士の対局数は 500 局，AMAF を適用した対戦プログラムと適用していない対戦プログラムの対局数は 200 局行った．表中の数値は先手の勝率 [%] を示す．

表 2. AMAF を適用した UCT を含む 4 つの AI の勝率

先手 \ 後手	従来	従来 AMAF	高速	高速 AMAF
従来	53.2	20.5	36.8	35.0
従来 AMAF	97.0	-	99.5	43.8
高速	74.6	17.0	53.1	43.5
高速 AMAF	94.0	48.8	91.0	-

AMAF を適用していない UCT との対局結果から，AMAF を適用することで，どちらのプレイアウト手法による UCT でも勝率が上がることがわかる．また，AMAF を適用した UCT が後手であれば，先手が AMAF を適用した UCT であっても勝ち越す結果が得られている．Hex は先手に有利なゲームであるにもかかわらず，なぜこのような結果が得られたのかは不明である．考えられる要因の 1 つとして，高速プレイアウト手法を AMAF と併用する際，従来のプレイアウト手法とは異なり，プレイアウトの結果を反映する手に偏りがある．この偏りが高速 AMAF の強さを安定させないため，先手が負け越す結果となったと考える．

6. 結論

本研究において，UCT を基礎とした Hex の対戦プログラムを構築し，提案する高速プレイアウト手法が従来のプレイアウト手法に比べて有用であることを実験により確かめた．AMAF を適用した UCT では，AMAF を適用していない AI に対して勝率を大きく伸ばしている．しかし，AMAF を適用した UCT が対局相手の場合，先手が勝ち越せていないため，さらなる実験と考察が課題となる．

参考文献

- [1] 小谷善行, 岸本章宏, 芝原一友, 鈴木豪, 「ゲーム計算メカニズム」, コロナ社, 2010.