

2 人対戦のハゲタカのえじきにおけるモンテカルロ木探索の性能比較

志村 伊生[†] 山口 文彦[‡]
 長崎県立大学[†] 長崎県立大学[‡]

1. はじめに

モンテカルロ木探索は、状態数が多く探索空間が膨大な問題において対応出来るよう、ゲーム木探索にモンテカルロ法を応用することでゲーム木探索における無駄を省き、有望な手について重点的に探索することが出来るアルゴリズムである。ゲーム AI に用いられるアルゴリズム研究において、このモンテカルロ木探索は、完全情報ゲームでありながら状態数が多いゲームである囲碁において大きな成果を上げた[1] ことで注目を集め、以降不完全情報ゲームや多人数プレイでのゲームなど、比較的難易度が高いゲームを対象に多角的なアプローチによる研究が行われている。しかし、モンテカルロ木探索はアルゴリズムの性質上、プレイヤーが順番に着手を行う逐次手番ゲームでの適用が多く、反対にプレイヤー全員が同時に着手を行う同時進行ゲームが対象となったモンテカルロ木探索の研究は少ない。

本研究では、同時進行で且つ不確定要素やバッティング要素を含むハゲタカのえじきを対象とし、モンテカルロ木探索を採用したゲーム AI を制作し、その性能を検証する。通常のモンテカルロ木探索は、逐次手番を前提としたアルゴリズムであるため、アルゴリズムの根幹的な部分は変えず、同時手番ゲームのために一部改変を行ったものを制作する。また、既に同時進行ゲームのためのモンテカルロ木探索として提案されている改良版のモデル、及びモンテカルロ木探索のシミュレーションにおいて、ハゲタカのえじきに含まれる要素の一つである偶然手番を重視した探索を行うモデルをそれぞれ制作する。そしてそれぞれの AI について、ランダムに行動を取るプレイヤーとの対戦を行い、その結果を比較する。

2. ハゲタカのえじき

ハゲタカのえじきは、2 人から 6 人でプレイ出来るドイツ発祥の対戦型カードゲームである。

プレイヤーは計 15 枚のハゲタカカードを特定のルールに従って獲得し、最終的にハゲタカカードに書かれている数字を得点として、一番得点を多く獲得したプレイヤーが勝ちとなる。

このゲームの特徴として、不確定な要素とバッティング要素を含むことが挙げられる。場に置かれるハゲタカカードは、プレイヤーの行動に左右されずランダムに決まる。またハゲタカカードを獲得する際のルールの中には、プレイヤー毎の行動についてバッティングが起こることで、本来獲得できていたはずのハゲタカカードが獲得出来なくなるといったことが起きる。同時進行ゲームであるため、自分が行動を取る際に相手の行動を観察できない点や、上記二点の要素があることによって、シンプルなルールであるにも関わらず決定的な戦略を見つけ攻略することが難しいゲームとなっている。

3. AI に実装するモデル

本研究で用いるモンテカルロ木探索のモデルは UCT と呼ばれており、これは探索に用いられる方策について、UCB と呼ばれる問題解決モデルを応用したものである。通常の UCT と、本研究に用いる UCT について、図 1 にその違いを示す。通常の UCT では、子ノードについて UCB1 値を比較し、UCB1 値が一番高いノードが次に探索するノードとして選択される。図 1 左の例では、A のノードが選択されることになる。一方、本研究用において実装した UCT では、次の状態に遷移するためには自分の行動と相手の行動の両方を考慮しなければならないため、その行動の全組み合わせにつ

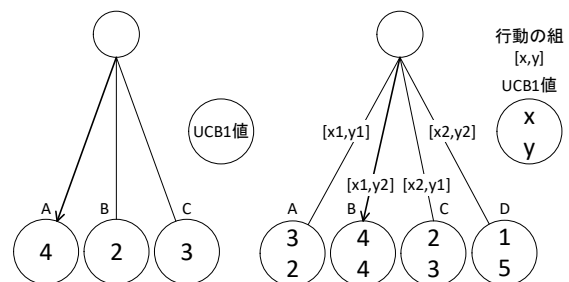


図 1. UCT の一例 (逐次手番と同時手番) [3]

Performance of monte carlo tree search methods in two-player "Hol's der Geier"

[†] Izumu Shimura; University of Nagasaki

[‡] Fumihiko Yamaguchi; University of Nagasaki

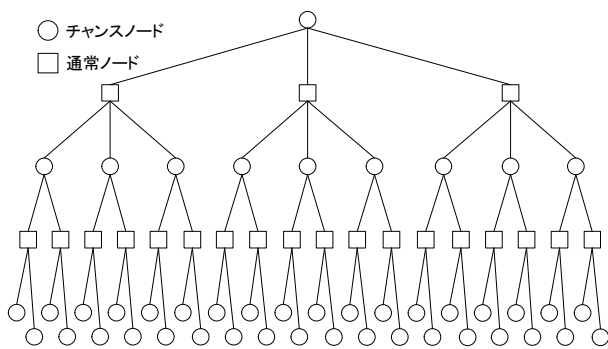


図 2. チャンスノードを含むゲーム木

いて子ノードが存在することになる。プレイヤー X とプレイヤー Y の行動をそれぞれ x, y とした場合、プレイヤー毎の UCB1 値を計算し、それぞれ UCB1 値が一番高い行動を保存、その行動の組に該当するノードを選択することになる。図 1 右の例では、B のノードが選択されることになる。この方法は、Decoupled-UCT (DUCT) [2] と呼ばれる方法と類似している。この方法を、本研究における基本的な UCT のアルゴリズムとして、Basic-UCT と呼ぶことにする。

また不確定な要素を含むゲームにおいてゲーム木を表現するには、プレイヤーの行動に左右されずに確率的に状態が遷移する部分を偶発手番としたとき、プレイヤーの行動によって次の状態に遷移する通常のノードに加え、偶発手番によって次の状態に遷移するチャンスノードで木を構成する必要がある。これは決定木や Expectiminimax など、いくつかの木探索アルゴリズムで使用されている方法である。チャンスノードと通常のノードについて、図 2 にその簡易的な位置づけを示している。この図を対象のハゲタカのえじきのゲーム進行に当てはめると、チャンスノードには裏にして伏せられている山札からハゲタカカードを一枚めくって場に出す動作で、通常ノードにはプレイヤーが持つ数字カードを場に出す動作となる。この時、通常のノードについては従来の UCT で探索を行えばよいものの、チャンスノードについては、実際の対戦場面の状況と同じく、一様な確率で子ノードの内一つのノードを選択することにする。

Basic-UCT の他に、Basic-UCT に対して特定の改良を加えたアルゴリズムについても実装を行う。一つ目に、Basic-UCT における着手決定について、純粹戦略的に行うところを、混合戦略的に着手を行うものを実装する。この方法を Basic-UCT(mix) と呼ぶことにする。具体的には、UCB1 値による探索及びノードの訪問回数による着手決

定について、一番大きな値のノードを選択するのではなく、それぞれの子ノードに付与されている値について、その割合を混合戦略として近似し選択する方法である。この方法を採用する理由は、本来相手の行動を見て自分の行動を決められない同時進行ゲームにおいて、ナッシュ均衡となる最適な戦略を取るには、混合戦略に基づいた確率的な着手を行う必要があるためである。これと同様の方法を不確定要素が含まれないゲームにおいて使用し、探索の深さを深くすることにより性能の向上が示唆されたことが分かっている [3]。

二つ目に、Basic-UCT における探索フェーズにおいて、先述した偶然手番及びチャンスノードの候補と、次状態でのプレイヤーの合法手を照らし合わせることで、合法手から算出される UCB1 値に対してバイアスを掛けるものを実装する。これにより、チャンスノードから遷移する要素をより重視した探索が行えることを期待する。この方法を Basic-UCT(chance) と呼ぶことにする。

4. アルゴリズムの評価

Basic-UCT と、上記で述べた 2 種類のアルゴリズムを実装した AI について、ランダムに行動をとるプレイヤーと 1000 回の対戦を行う。各アルゴリズムのシミュレーション回数、及び UCB1 値におけるハイパーパラメーターは全て統一し、それらを変化させた場合での対戦結果についても比較する。また各 AI がとった行動について記録し、不確定要素や相手の合法手に対しての行動傾向分析を行う。これはシミュレーション終了後における、各行動に付与された UCB1 値も考慮した分析を行う。

参考文献

- [1] Remi Coulom, Efficient selectivity and backup operators in Monte-Carlo tree search, International Conference on Computers and Games, pp. 72-83 (2006)
- [2] Lanctot, M., Wittlinger, C., Den Teuling, N. G. P., Winands, M. H. M., Monte Carlo Tree Search for Simultaneous Move Games: A Case Study in the Game of Tron, BNAIC 2013: Proceedings of the 25th Benelux Conference on Artificial Intelligence, (2013)
- [3] 大町洋, 池田心, 同時進行ゲームのためのモンテカルロ木探索, ゲームプログラミングワークショップ 2012 論文集, pp. 197-204 (2012)