

不完全情報ゲームにおける 適応的モンテカルロ木探索手法の提案

大佐賀 猛^{1,a)} 野中 秀俊¹ 吉川 毅¹ 杉本 雅則¹

受付日 2014年8月25日, 再受付日 2014年10月17日,
採録日 2014年11月25日

概要: モンテカルロ木探索は完全情報ゲームを対象として提案された手法であるが, 不完全情報ゲームに対しても有効であることが知られている. これに対し, モンテカルロ木探索を再帰的に繰り返し行う手法や情報集合を用いて木を探索する手法などが提案されている. しかし, ゲームの状態によってはこのような工夫を行うよりも通常のモンテカルロ木探索を行う方が性能が良い場合がある. 本報告では, 探索の回数や深さを適応的に変化させる適応的モンテカルロ木探索を提案する. 本報告における評価の対象問題として, 既存の2人零和不完全情報ゲームのモデルである Synthetic Tree のパラメータを再定義し, より多様な不完全情報ゲームに対応できる汎用的なモデルを導入した. このモデルを用いた評価実験により, 提案手法は既存手法よりも高い性能を示すことができた.

キーワード: 不完全情報ゲーム, モンテカルロ木探索, UCB, 適応的モンテカルロ木探索

Adaptive Monte Carlo Tree Search for Incomplete Information Games

TAKESHI OSAGA^{1,a)} HIDETOSHI NONAKA¹ TAKESHI YOSHIKAWA¹
MASANORI SUGIMOTO¹

Received: August 25, 2014, Revised: October 17, 2014,
Accepted: November 25, 2014

Abstract: Monte Carlo Tree Search (MCTS) is a search method for perfect information games and can be effectively used for incomplete information games too. Several extensions for the MCTS method were proposed, for example recursive MCTS or MCTS using information sets. However, these methods may not improve performance compared to the original MCTS. In this paper, we propose Adaptive Monte Carlo Tree Search (AMCTS) that changes the number of search times and search depth. We use a Synthetic Tree with redefined parameters as a target problem for the evaluation of our proposed method as this kind of problem corresponds to a variety of incomplete information games. The proposed method achieved higher performance than conventional methods.

Keywords: incomplete information games, Monte Carlo Tree Search (MCTS), Upper Confidence Bound (UCB), Adaptive Monte Carlo Tree Search

1. はじめに

モンテカルロ木探索 (Monte Carlo Tree Search, 以下 MCTS) はモンテカルロ法とゲーム木探索を組み合わせた手法で, 完全情報ゲームのうち, 特に囲碁などの状態数が

多いゲームで有効な手法であるとされている. カードゲームなどの不完全情報ゲームでは, 一般に観測できない情報があるため, 状態数がさらに多くなる傾向があり, これらに MCTS が適用されるようになってきている.

不完全情報ゲームに初めて MCTS を適用した研究として Perfect Information Monte Carlo (PIMC) 探索 [1] がある. これは観測できない状態の中からランダムにサンプリングを行うことによって状態を定め, 完全情報ゲームとし

¹ 北海道大学
Hokkaido University, Sapporo, Hokkaido 060-0814, Japan
^{a)} osaga@main.ist.hokudai.ac.jp

てゲーム終了までシミュレーションを繰り返し行うアルゴリズムである。このアルゴリズムを用いてブリッジやハーツなどで強いプレイプログラムが作られた。

Furtak らは PIMC 探索におけるプレイアウト中の方策を様々な方策に変更した手法, Imperfect Information Monte Carlo (IIMC) 探索 [2] を提案している。この手法は、通常の PIMC 探索よりも報酬を獲得する能力 (以下、性能と呼ぶ) が向上しているが、計算時間が増大していることが問題としてあげられている。また、深く探索する手法は不完全情報が多いゲームをプレイする場合には逆効果になる場合がある [3]。

本研究では様々なゲームに対応するために、ゲームの状態に応じてプレイアウトの方策や探索の回数を適応的に変更させる手法、適応的モンテカルロ木探索 (Adaptive Monte Carlo Tree Search, 以下 AMCTS) を提案する。AMCTS はゲームの状態に応じて十分な回数、十分な深さの探索を行うためにプレイアウト中の方策をランダムに選び着手を決定する方法と PIMC 探索によって着手を決定する方法を使い分ける手法である。

本手法の有効性を検証するために Long らが提案した 2 人零和不完全情報ゲームのモデルである Synthetic Tree [4] においてパラメータの再定義を行い、より一般的な不完全情報ゲームに対応できる汎用的なモデルを構築した。このモデルを利用して評価実験を行った結果、AMCTS は PIMC 探索と比べて約 2% 高い勝率を示し、かつ IIMC 探索の計算時間と比べて最大約 65% 短縮されることを示した。

2. 研究背景

2.1 UCB (Upper Confidence Bound)

UCB アルゴリズムは、多腕バンディット (Multi-Armed Bandit) 問題に対して提案された手法で、式 (1) で定義される UCB 値が最も高いスロットマシンが選択される。

$$UCB(s) = \bar{x}_s + \sqrt{\frac{2 \ln n}{n_s}} \quad (1)$$

s はスロットマシン、 \bar{x}_s はスロットマシン s において観測した報酬の平均値、 n は全体の試行回数、 n_s は s を選んだ回数である。まだ一度も選ばれていないスロットマシンの UCB 値は最大にする。また UCB1-tuned [5] は UCB の戦略に加え報酬の分散を考慮した戦略である。UCB1-tuned 値は式 (2), (3) で表される。

$$UCB1\text{-tuned}(s) = \bar{x}_s + \sqrt{\frac{\ln n}{n_s} \min\{1/4, V_s(n_s)\}} \quad (2)$$

$$V_s(n_s) = \left(\frac{1}{t} \sum_{i=1}^t x_{s,i}^2 - \bar{x}_{s,t}^2 + \sqrt{\frac{2 \ln n}{t}} \right) \quad (3)$$

$x_{s,i}$ はマシン s の i 回目の試行で得られた報酬、 $\bar{x}_{s,t}$ はマシン s を t 回試行したときの得られた報酬の平均値である。UCB1-tuned は理想的な戦略との報酬差に対する理論的な

Algorithm 1 PIMC 探索 (InfoSet I , int N)

```

1: for  $i = 0$  to  $M(I)$  do
2:    $val[i] = 0$ 
3: end for
4: for  $i = 0$  to  $N$  do
5:    $x = \text{Sample}(I)$ 
6:   for  $m = 0$  to  $M(I)$  do
7:      $v = \text{PerfInfoValue}(x, m)$ 
8:      $val[m] += v$ 
9:   end for
10: end for
11: return  $\arg \max_m \{val[m]\}$ 

```

評価式は与えられていないが、評価実験によって UCB よりも性能が高いことが示されている。

本研究の戦略決定のアルゴリズムには、UCB1-tuned を使用した。

2.2 Perfect Information Monte Carlo (PIMC) 探索

Perfect Information Monte Carlo (PIMC) 探索は MCTS を不完全情報ゲームに適用する際に、ゲーム全体の状態を特定できないという問題を解決するために提案された手法である。

PIMC 探索は特定できない状態においてとりうる状態を列挙 (サンプリング) する手法である。PIMC 探索の擬似コードを Algorithm 1 に示す。情報集合 I はゲーム木のノードの集合であり、 N はプレイアウト回数である。 $M(I)$ は I から選択できる行動の数である。 $\text{Sample}(I)$ は I からとりうる状態から 1 つの状態をランダムにサンプリングする関数である。 $\text{PerfInfoValue}(x, m)$ は状態 x から行動 m を行い状態遷移した後、それぞれのプレイヤーが完全情報でプレイしたときに得られる報酬を返す関数である。

2.3 Imperfect Information Monte Carlo (IIMC) 探索

完全情報ゲームで MCTS の性能を向上させるために MCTS を再帰的に適用する手法が提案されている [6]。これを不完全情報ゲームに一般化した手法が Imperfect Information Monte Carlo (IIMC) 探索 [2] である。IIMC は PIMC 探索の PerfInfoValue を FinishedGameValue に変更している点が異なる。

FinishedGameValue の擬似コードを Algorithm 2 に示す。MakeMove(x, m) は状態 x のときに行動 m を行ったときに到達する状態 y を返す関数である。ComputeMove はランダムなプレイヤーやルールベースプレイヤー、より洗練されたプレイヤーなどのアルゴリズムなどを適用することによって行動選択を行う関数である。文献 [2] では ComputeMove に PIMC 探索を適用したものについて議論している。今後、IIMC 探索とは ComputeMove に PIMC 探索を適用したものとする。

Algorithm 2

FinishedGameValue(Node x , Move m , Player P)

```

1:  $y = \text{MakeMove}(x, m)$ 
2: while  $y$  is not a terminal node do
3:    $y = \text{MakeMove}(y, \text{ComputeMove}(P, I(y)))$ 
4: end while
5: return value of  $y$  in view of the player to move in  $x$ 
    
```

3. 評価モデル

文献 [4] では、不完全情報ゲームが持つ特徴を 3 つにまとめ、様々な特徴において PIMC 探索の性能を評価するために Synthetic Tree というモデルを提案している。本実験では提案手法を評価するための 2 人零和不完全情報ゲームのモデルとして Synthetic Tree を用いた。

Synthetic Tree は以下の 3 つのパラメータによって特徴付けられる。

- Leaf Correlation(lc): 確率 lc で終端ノードの組が同じ報酬になる。確率 $1 - lc$ で終端ノードの組が異なる報酬になり、どちらか片方がランダムに 1 の報酬が与えられ、もう片方に -1 の報酬が与えられる。
- Bias(b): 上記の lc で同じ報酬と判定された終端ノードの組に対して、確率 b で報酬 1、確率 $1 - b$ で報酬 -1 が与えられる。
- Disambiguation Factor(df): それぞれのプレイヤーが行動を選択するたびに、確率 df でそのプレイヤーが保有する情報集合 (以下、世界と呼ぶ) の要素数を半分にする。この操作が起きたとき、再帰的にこの操作を繰り返す。たとえば、確率 df を満たし世界の数を半分にした後、また確率 df を満たしたとき、さらにそのプレイヤーの世界の数が半分になる。

lc , b は実際のゲームをランダムにプレイして到達した終端ノードの親ノードのすべての報酬の値を記録することで得られる。 df は、行動ごとにとりうるゲーム全体の状態の数を計算し、行動を選択するたびに前の世界の候補の数と比べることにより得られる。これにより、実際のゲームを Synthetic Tree のパラメータで表現することができる。

各プレイヤーの世界の候補の数が 2、木の深さが 2 の場合の Synthetic Tree の例を図 1 に示す。丸で示されているルートノードがチャンスノードと呼ばれるノードであり、そこから確率的に子ノードへと遷移する。上向き三角、下向き三角はそれぞれのプレイヤーを示しており、下向き三角については影の色はそのプレイヤーから見て同じ状態に見える状態を示している。

文献 [4] のモデルでは報酬が 1, -1 の 2 つしか与えられないため、以下のように Bias の再定義を行った。

Bias の再定義

平均 b , 標準偏差 $1/R$ の正規分布を想定する。 R は正

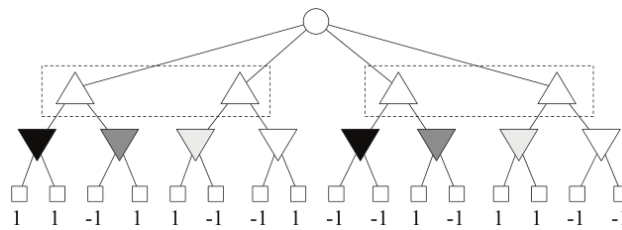


図 1 世界 $W = 2$, 木の深さ $d = 2$ の Synthetic Tree

Fig. 1 Synthetic Tree ($W = 2, d = 2$).

の整数とする。この正規分布からサンプル i を 1 つ取り出す。取り出した i を $0 \leq i \leq 1$ とするために、 $i < 0$ のとき $i = 0$, $i > 1$ のとき $i = 1$ とした。取り出した i が以下の式を満たす最小の正の整数 r が終端ノードに報酬として与えられる。

$$i \leq \frac{2(r+1)-1}{2R} \tag{4}$$

また、確率 lc により同じ報酬が与えられると判定されたノードの組に対しては報酬 r , 確率 $1 - lc$ により異なる報酬が与えられると判定された終端ノードの組に対しては $R - r$, r を与える。

Long らの Bias の定義では 1, -1 の報酬しか表現できなかったが、新しい bias の定義により 0 から R までの整数の報酬を表現でき、Synthetic Tree はより一般的なモデルとなった。

4. 提案手法

本章では提案手法である、適応的モンテカルロ木探索 (Adaptive Monte Carlo Tree Search, AMCTS) のアルゴリズムについて述べる。まずゲームの状態に応じて適応的にプレイアウト回数を設定する方法と、プレイアウトの方策を設定する方法について述べる。その後、アルゴリズムの全体の流れについて述べる。

4.1 適応的アルゴリズム

AMCTS は IIMC 探索のプレイアウトの方策をゲームの状態に応じてランダムプレイアウトと PIMC 探索を使い分けるアルゴリズムである。また、プレイアウト回数もゲームの状態に応じて変更する。まず、プレイアウト回数の設定について述べ、その後、プレイアウトの方策の設定について述べる。

4.1.1 プレイアウト回数の設定

現在の手番の状態における世界の候補の数 W , 行動の数 A に対して、プレイアウト回数 N は式 (5) で設定する。

$$N = WAC \tag{5}$$

C は定数であり、ゲームに応じて設定する。式 (5) により、世界の候補が少ない状態や、行動の選択肢が少ない状態に性能を落とすことなくプレイアウト回数を削減することが

できる。

プレイアウト回数は完全情報ゲームと不完全情報ゲームで意味が異なる。完全情報ゲームの場合は木を構築しながら探索を行うため、プレイアウト回数を無限回行うと最適解に到達するということが示されている [7]。一方、不完全情報ゲームの場合は状態のサンプリングを行いながら探索を行うため木の構築を行わない。そのため現在の手番の状態からの評価しか行わないため、プレイアウト回数を増やしても最適解に到達する保証はない。また、状態は一意に定まらないため、求めた最適解に到達できないことがある。よって PIMC 探索や IIMC 探索のプレイアウト回数はそれぞれの状態における行動を適切に評価するように設定する必要がある。各行動を適切に評価するために必要なプレイアウト回数を C とすると、各世界における行動の数だけ C 回のプレイアウトが必要であることが分かる。

以上の考察により、必要なプレイアウトの回数 N は W , A に比例すると考えられるため、式 (5) を用いた。

4.1.2 プレイアウトの方策の設定

プレイアウトの方策はまずランダムに N 回探索を行う。式 (6) を満たすときにプレイアウトの方策を PIMC 探索として再度探索を行う。

$$v(a^*) - v(a_i) < \frac{1}{\sqrt{W}} (\sigma^* + \sigma_i) \quad (6)$$

a_i は現在の手番から行える i 番目の行動、 a^* はランダムプレイアウトで評価値が一番高い行動、 σ_i は行動 i のプレイアウトで得られた報酬の標準偏差、 σ^* は a^* のプレイアウトで得られた報酬の標準偏差、 $v(a_i)$ は a_i の評価値である。

プレイアウト時に PIMC 探索が必要となる場合は、現在の手番の状態から選択できる最適解の経路上である子ノードのその他の終端ノードの評価値が低い場合である。その場合 PIMC 探索で探索することによってより高い報酬を獲得できる行動を探索することができ、子ノードの評価値は終端ノードの報酬が高いノードを選択した回数が多いほど高くなる。このことからプレイアウトに PIMC 探索を使うべき状態は最も評価値が高いノードと他のノードとの評価値の差 ($v(a^*) - v(a_i)$) が僅差、かつ評価値が高いノードと低いノードが混在している、つまりプレイアウトで得られた報酬の標準偏差 ($\sigma_a^* + \sigma_i$) が大きいノードが存在する状態である。また、真の状態と異なっている状態が多いほど、様々な終端ノードの報酬があるため、評価値の標準偏差が大きくなる可能性があり、標準偏差の和に $\frac{1}{\sqrt{W}}$ をかけた。以上の理由により、PIMC 探索による再探索の基準を式 (6) で与えた。

1 回のランダムプレイアウトにかかる計算量を $O(1)$ とした場合、プレイアウトの方策として PIMC 探索を適用した際に 1 回のプレイアウトにかかる計算量は、PIMC 探索中のプレイアウト回数を M とすると、 $O(M)$ である。そのため、ランダムプレイアウトよりも良い着手が見つかる可

能性があるときのみ PIMC 探索を行うことにより、すべての状態に対してプレイアウト時に PIMC 探索を行う IIMC 探索よりも計算量を削減できる。

4.2 AMCTS のアルゴリズム

現在の手番の状態 s において選択できる行動の集合を A_s とする。集合 A_s の要素 $a_i \in A_s$ の行動の評価値を $v(a_i)$ とする。以下に AMCTS の詳細なアルゴリズムを示す。

step1 初期化

現在の状態からとりうる状態の集合 S を求め $s \in S$ としたとき s からとりうる行動を列挙する。列挙した行動は A_s に追加する。 A_s のすべての要素の行動の評価値を $v(a_i) = 0$ に初期化する。

step2 プレイアウト回数の計算

$|S|$, $|A_s|$ に応じたプレイアウト回数 N を式 (5) で求める。

step3 状態サンプリング

S の中からランダムに 1 つサンプリングする。

step4 ノード選択

UCB1-tuned によって探索すべき $a_i \in A_s$ を選択する。

step5 プレイアウト

step4 で選択した行動 a_i を行い、状態遷移させたのちランダムプレイアウトを行う。プレイアウトの結果を $v(a_i)$ に加算する。

step6 繰返し

step3–5 を N 回繰り返す。

step7 探索終了判定

すべての $a_i \in A_s$ に対して式 (6) を評価して 1 つでも満たす場合は step8 を行う。満たさない場合は step9 を行う。

step8 再探索

A_s のすべての要素に関して $v(a_i) = 0$ にした後、step3–5 を N 回繰り返す。ただし、step5 のプレイアウト中の方策は相手の行動はランダムで選択し、自分の行動を PIMC 探索によって選択する。

step9 探索終了

$\arg \max_a v(a_i)$ を満たす a_i を選択する。

5. 評価実験

5.1 実験設定

実験は Bias を再定義した Synthetic Tree を用いて行った。 $lc = 0.3$, $b = 0.75$, $W = 8$, $d = 10$, $|A| = 2$ とし、報酬の範囲を 0 から 4 までの整数とした。 df の値を 0.1 ずつ変更し、Synthetic Tree を作成した。この Synthetic Tree に対して、AMCTS, PIMC 探索, IIMC 探索それぞれと完全情報でプレイする PIMC 探索との対戦を行った。それぞれ先攻後攻を入れ替え、計 3000 回ずつ対戦を行っ

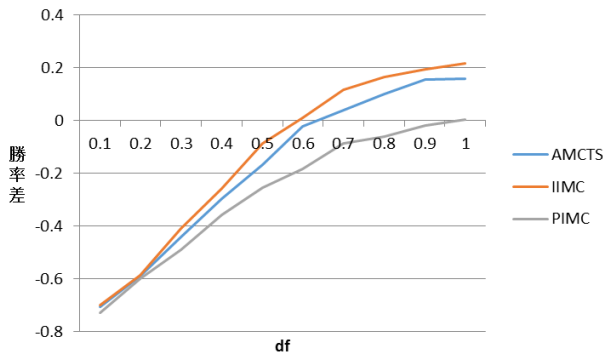


図 2 N 固定勝率差の比較

Fig. 2 Comparison of the difference of winning rate fixed N .

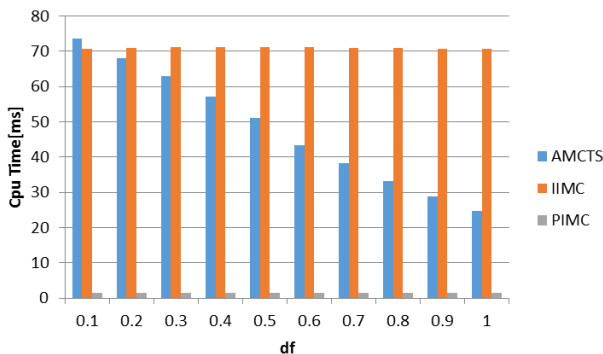


図 3 N 固定計算時間の比較

Fig. 3 Comparison of the search time fixed N .

た. PIMC 探索のプレイアウトはランダムプレイ, IIMC 探索のプレイアウトは PIMC 探索を使用した. 実験はプレイアウト回数を固定する実験と, 1つの着手を決めるために使える計算時間を制限する実験の2種類行った. プレイアウト回数を固定する実験は十分な時間が与えられた場合の報酬獲得能力と計算時間の関係と比較するために行った実験である. また, 計算時間を固定した実験は制限時間内での報酬獲得能力を比較するために行った実験である. プレイアウト回数は $N = 400$ とし, プレイアウト中の PIMC 探索のプレイアウト数は $M = 50$ とした. AMCTS のプレイアウト数は式 (5) より $W = 8$, $|A| = 2$ のとき $N = 400$ となるように $C = 25$ とした. 計算時間 t は $0.05 \leq t \leq 1.5$ [ms] の範囲で 0.05 ずつ変化させた. 計算時間が t を超えた場合はその時点の評価値を基に着手を決定した. 計算時間を制限する実験は df 以外のパラメータは変化させず, $df = 0.2, 0.5, 0.8$ の3種類の Synthetic Tree で実験を行った. これはハーツや大貧民などのゲームは序盤は不完全情報の量が多く, 終盤になるにつれて情報が明らかになっていくことから, ゲームの序盤, 中盤, 終盤を想定した設定である. 以上の2つの実験の1試合あたりの各プレイヤーの勝率と計算時間を記録した.

5.2 実験結果

プレイアウト回数を固定した実験結果は図 2, 図 3 のよ

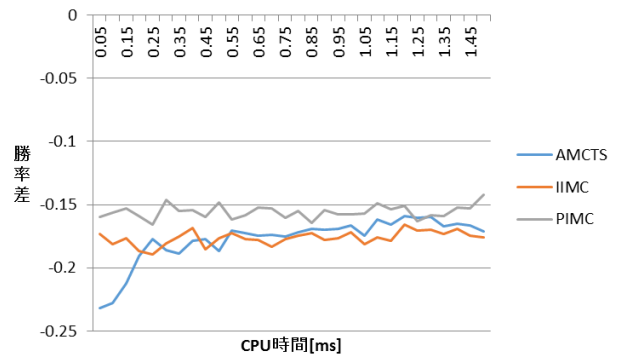


図 4 計算時間 (CPU 使用時間) 制限の勝率差比較 ($df = 0.2$)

Fig. 4 Comparison of the difference of winning rate that limited calculation time ($df = 0.2$).

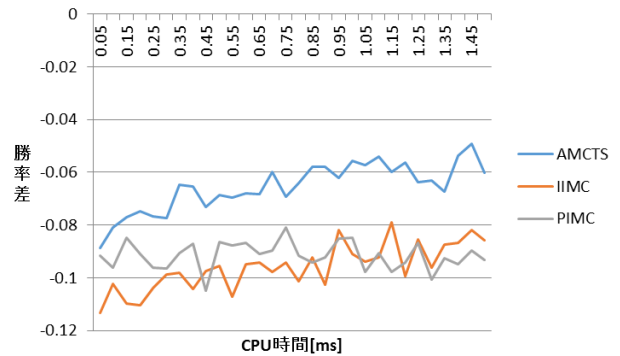


図 5 計算時間 (CPU 使用時間) 制限の勝率差比較 ($df = 0.5$)

Fig. 5 Comparison of the difference of winning rate that limited calculation time ($df = 0.5$).

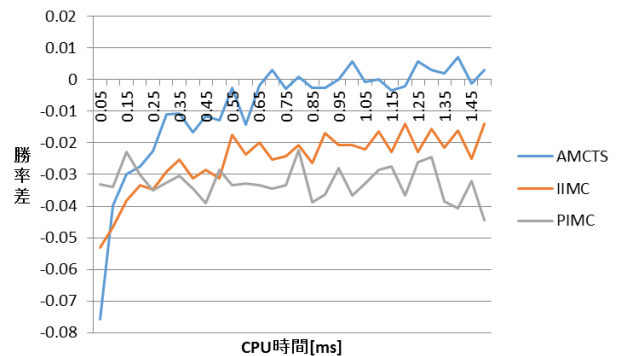


図 6 計算時間 (CPU 使用時間) 制限の勝率差比較 ($df = 0.8$)

Fig. 6 Comparison of the difference of winning rate that limited calculation time ($df = 0.8$).

うになった. 図 2 は df を横軸にとったときの 1 試合あたりの各プレイヤーの完全情報でプレイする PIMC 探索との勝率の差を表している. 図 3 は df を横軸にとったときの 1 試合あたりの計算時間を表している. 図 2 より df が高いとき, つまり不完全情報が少ないときに AMCTS と IIMC 探索の性能が PIMC 探索よりも上回っていることが分かる. AMCTS と IIMC 探索はほぼ同程度の勝率を記録したが, 図 2, 図 3 より少ない計算時間で同じ性能を示していることが分かる. また図 3 より, df が大きくなるにつれ

て計算時間が短くなっており、最大で約 65%短くなっていることが分かる。

計算時間を制限した実験結果は図 4, 図 5, 図 6 のようになった。同じ計算時間で性能を比較した場合, 図 5, 図 6 より, df が大きい値では AMCTS は高い性能を示した。ただし図 4 より, df が低いとき, つまり不完全情報が多いときは AMCTS は PIMC 探索に性能が劣っている。これらの結果についての考察は 6.3 節で述べる。

6. 考察

図 2 より AMCTS が PIMC 探索よりも高い性能 (報酬を獲得する能力) を示すことができ, 計算時間を制限した場合図 4, 図 5, 図 6 より df の値が低いときを除き AMCTS が PIMC 探索と IIMC 探索よりも高い性能を示すことができた。

その要因は式 (5), (6) である。これらの式についての考察を行う。

6.1 プレイアウト回数の考察

プレイアウト回数は性能を大きく左右する要因となる。プレイアウト回数を増やすことによって, 多くの状態における行動の評価を行うことができ, 評価値がより信頼できるものになる。既存の手法ではプレイアウト回数を事前に十分与えていた。しかしその方法ではゲーム全体で考えたときにゲームのすべての状態で適切なプレイアウト回数であるとはいえない。その理由は 2 つあげられる。

まず 1 つ目の理由は世界の候補数が増えることである。サンプリングする際, 世界の候補数に比例してサンプリングを行う必要がある。サンプリングの回数はプレイアウト回数と等しく, 候補数 W が小さい場合, プレイアウト回数を減らすことができる。

2 つ目の理由は現在の手番から選択できる行動の数がゲームの状態によって変化することである。行動の数に比例したプレイアウト回数が必要である。ゲーム終了までに遷移しうる状態においても行動の数が変化したが, どの状態に遷移するかは不明であるため, 考慮していない。

今回の実験では $A = 2$ で固定であったが, df が大きいとき, つまり W が小さくなりやすいとき, 図 3 より AMCTS の計算時間が短縮されているのが分かる。

6.2 プレイアウトの方策についての考察

プレイアウトの方策をランダムとすると探索 1 回あたりにかかる時間が短い, 最適解を見つけにくいという特徴がある。しかし, 不完全情報が多い状態のときにはすべての状態を網羅的に探索するのは効率が良くない場合がある。そのような場合に対してはランダムプレイアウトは有効に機能する。一方でプレイアウトに PIMC 探索を用いた場合はランダムの方策よりも最適解を見つけやすいが,

1 回の計算に時間がかかり限られた時間の中で状態を評価することは困難である。よってこれらの探索手法を使い分けることが非常に効果的であると考えられる。たとえば囲碁における定石では, 石を置く順番が最終的な勝ち負けに大きく影響する。このような状況では, PIMC 探索によるプレイアウトが有効であると考えられる。また不完全情報の量, 状態の評価値によってプレイアウトの方策を変更する必要がある。図 5, 図 6 より, AMCTS は式 (6) の方策の切替えがうまく機能しており, 高い性能を示している。また図 2 より, AMCTS はプレイアウトの方策をすべて PIMC 探索を行っている IIMC 探索と比べてほぼ変わらない性能を示している。以上より, AMCTS のプレイアウトの方策をランダムでよいところはランダムで探索を行い, 探索回数を増やすべきところでは PIMC 探索を使い分けることにより, 短い計算時間で探索スコアを向上させることに成功したと考えられる。

6.3 実験結果の考察

図 2 より AMCTS は PIMC 探索よりも高い性能を示しているが, AMCTS と IIMC 探索と比べるとほぼ同じ性能を示した。IIMC 探索の性能を上回るためには式 (6) を改良するか, プレイアウトの方策を新たに追加する必要がある。IIMC 探索は不完全情報が多いときにはうまく性能を発揮できないため, 図 2 より df が極端に小さいときには AMCTS の性能が悪いことが分かる。そのため AMCTS の追加の方策としては不完全情報に強いアルゴリズムが適していると考えられる。

計算時間については図 2, 図 3 より df に比例して減少しておりかつ, 性能も向上していることから, 式 (5) の有効性が示されたが, 図 4, 図 6 より AMCTS に十分な計算時間が与えられなかった場合, 性能が悪くなることがあることが分かる。これは AMCTS がプレイアウトの方策を PIMC 探索とする場合, 以前の探索した結果を破棄して新たに探索を始めるからであると考えられる。これを解決するためには以前の探索の結果を破棄するのではなく, 以前の探索の結果と新たな探索の結果をうまく組み合わせる必要がある。その際にはランダムプレイアウトの結果と PIMC 探索の結果の重み付けが重要になると考えられる。

7. まとめ

本報告では不完全情報ゲームをプレイするためのアルゴリズムである AMCTS を提案した。この手法は IIMC 探索を応用したアルゴリズムであり, プレイアウト中の方策をゲームの状態に応じて変更することによって適切な探索を行うアルゴリズムである。またプレイアウト回数をゲームの状態に応じて変更し, 計算時間の削減を行った。実験より, 提案手法が IIMC 探索や PIMC 探索と比べて性能を向上させることに成功した。しかし, 十分な計算時間が与

えられなかった場合は PIMC 探索に劣ることがあり、また IIMC 探索と比べて計算時間は改善されているが、十分な計算時間を与えられた場合は AMCTS は IIMC 探索の性能を上回ることができなかったため改善していく必要がある。

本報告の評価実験では人工的な不完全情報ゲームのモデルである Synthetic Tree で提案手法の評価を行った。今後は木の深さや、枝の数からも状態を判断しハーツや大貧民といった実際のゲームに対して評価実験を行いたい。その際に Synthetic Tree で用いた各パラメータが実際のゲームの特徴を適切に表現しているかどうかを確認する必要がある。

参考文献

- [1] Ginsberg, M.: GIB: imperfect information in a computationally challenging game, *Journal of Artificial Intelligence Research* 14, pp.303-358 (2001).
- [2] Furtak, T. and Buro, M.: Recursive Monte Carlo Search for Imperfect Information Games, *Computational Intelligence in Games (CIG)* (2013).
- [3] Cowling, P.I., Powley, E.J. and Whitehouse, D.: Information Set Monte Carlo Tree Search, *IEEE Trans. Computational Intelligence and AI in Games*, Vol.3, No.2 (2012).
- [4] Long, J.R., Sturtevant, N.R., Buro, M. and Furtak, T.: Understanding the Success of Perfect Information Monte Carlo Sampling in Game Tree Search, *Proc. 24th AAAI Conference on Artificial Intelligence* (2010).
- [5] Auer, P., Cesa-Bianchi, N. and Fischer, P.: *Finite-time Analysis of the Multiarmed Bandit Problem*, Kluwer Academic Publishers, Manufactured in The Netherlands (2002).
- [6] Cazenave, T.: Nested Monte-Carlo Search, *Proc. 21st International Joint Conference on Artificial Intelligence* (2009).
- [7] Kocsis, L. and Szepesvari, C.: Bandit Based Monte-Carlo Planning, *Machine Learning: ECML* (2006).



大佐賀 猛

2013年北海道大学工学部情報エレクトロニクス学科コンピュータサイエンスコース卒業。現在、北海道大学大学院情報科学研究科コンピュータサイエンス専攻修士課程在学中。



野中 秀俊 (正会員)

1983年東京大学工学部計数工学科卒業。1985年同大学大学院工学系研究科修士課程修了。同年北海道大学大学院工学研究科博士後期課程入学。同年北海道大学助手。1996年北海道大学助教授。現在、北海道大学情報科学研究科准教授。博士(工学)。知能情報学、ヒューマンインタフェースの研究に従事。



吉川 毅 (正会員)

1993年北海道大学工学部情報工学科卒業。2001年北海道大学大学院工学研究科システム情報工学専攻博士後期課程修了。博士(工学)。現在、北海道大学大学院情報科学研究科助教。知能情報学、数理情報科学の研究に従事。人工知能学会、日本応用数理学会各会員。



杉本 雅則 (正会員)

1990年東京大学工学部航空学科(宇宙工学専修)卒業。1995年同大学大学院工学系研究科博士課程修了。博士(工学)。同年より文部省大学共同利用機関学術情報センター(現国立情報学研究所)研究開発部助手。1999年より東京大学情報基盤センター助教授、2002年より同大学大学院新領域創成科学研究科基盤情報学専攻助教授、2008年より同大学大学院工学系研究科電気系工学専攻准教授、2012年より北海道大学大学院情報科学研究科コンピュータサイエンス専攻教授、2014年より同研究科情報理工学専攻教授となり現在に至る。