多腕バンディットアルゴリズムのMCTSへの応用と性能の分析

今川　孝久[1]　金子　知適[1]

概要：UCT は囲碁や General Game Playing などの応用で効果を発揮している探索手法で，多腕バンディット問題のアルゴリズムである UCB1 をモンテカルロ木探索 (MCTS) に応用したものである．多腕バンディット問題には，KL-UCB, Thompson Sampling, UCB1-Tuned など UCB1 より優れるとされる様々なアルゴリズムが提案されてきている．そこで本研究では、UCB1 に変えてそれらのアルゴリズムを MCTS に用いることを提案し，実際の性能について P-game と呼ばれる仮想的なゲーム木を用いて分析した．実験の結果，UCB1 よりも KL-UCB, Thompson Sampling, UCB1-Tuned が MCTS においても優れることが，期待通りに確認された．ただし，各アルゴリズムの差よりも各仮想的なゲーム木の性質に因る性能の違いの方が大きいことも同時に確認されている．本稿で用いた P-game は，広く探索アルゴリズムの性能の評価で用いられているが，MCTS の評価に用いる場合は，木の作り方に注意を払う必要がある可能性がある．

Applying Multi Armed Bandit Algorithms to MCTS and Those Analysis

Takahisa Imagawa[1]　Tomoyuki Kaneko[1]

Abstract: UCT is a search method which is effective in such as Go and General Game Playing, and it is a application of UCB1, an algorithm of multi-armed bandit proplem to Monte-Carlo tree search (MCTS). In multi-armed bandit problem, various algorithms better than UCB1 have been proposed, such as KL-UCB, Thompson Sampling, UCB1-Tuned. In our research, the other algorithms instead of UCB1 are applied to MCTS and, it's effectiveness are analized by P-game, a virtual game. The result of the experiments show that KL-UCB, Thompson Sampling and UCB1-Tuned are better than UCB1 as it is expected, but also difference of effectiveness caused by properties of P-game trees is larger than difference between the algorithms. P-game is widely used for evaluating effectiveness of search algorithms, but we may need to take care of creating P-game tree when evaluating MCTS.

1. はじめに

モンテカルロ木探索 (MCTS) は，囲碁などの探索空間が大きなゲームでも，概ね効果的に動作する優れた探索手法である．MCTS の 1 種である UCT [7] は，多腕バンディット問題のアルゴリズムである UCB1 [2] を木探索に応用したものであり，特に広く用いられている．しかし，チェス等のゲームでは $\alpha\beta$ 探索の方が適するなど，UCT は万能ではない．例えば，有利さが偏っていると性能が出難いこと [3], [10] 等が知られている．また，多腕バンディット問題では，Regret という指標で評価した際に，UCB1 よりも優れたアルゴリズムが提案されている [1], [5], [8].

UCT の性能はどのような局面で出やすいのかは未解決な研究課題である．我々は，UCB1 以外のアルゴリズムを MCTS に応用すると UCT の性能と比較してどれくらい改善するのか，また，そのような応用には探索局面に対し得手不得手があるのか，もしくは，似た傾向になるのかに興味がある．

[1] 東京大学大学院総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo

本研究では，UCB1 以外の手法の MCTS への応用を考え，どれだけ性能が改善するかを調査，分析をする．探索対象としては P-game[7], [9] を用いた．P-game は乱数によってゲーム木が決まるもので，統一の枠組みでパラメータが異なる木を大量に作ることが出来，深さや分岐数といったパラメータも調節が容易であるという利点がある．

調査の結果，多くの P-game 木に対してアルゴリズムによる改善が見られたが，同時にその改善の程度より，木の違いによる性能の変化の方が大きいという知見を得た．そこで，木の特徴量と各アルゴリズムの性能との関係性を調査した．

2. 背景

2.1 多腕バンディット問題と各種アルゴリズム

多腕バンディット問題は次のような問題である．各アームに対して，利得の確率分布が決められていて，あるアームを選ぶとその利得が確率的に得られる．しかし，その確率分布は分からないとする．そのような状況下で，決められた回数だけアームを選んだ場合の利得の総和を最大化するという問題である．この問題での性能を議論するのに Regret という量がよく用いられる．Regret は最善手を引きづづける場合と比べて，利得の観点でどれだけ損をしたかという量である．

UCB1 [2] では，有望さとその評価の不確かさを組み合わせた評価基準である UCB 値を定義し，それが最大となるアームを選ぶ．具体的には，全アームの試行回数を s とし，取りうるアームの集合を A，アーム j を選んだ場合の平均利得を $\hat{\mu_j}$ その試行回数を n_j とすると

$$\arg\max_{j \in A} \left(\hat{\mu}_j + \sqrt{\frac{2\log(s)}{n_j}} \right)$$

というアームを選択する．

KL-UCB [5] は，KL 情報量を元にして評価基準を定めた方法である．具体的には，

$$d(p,q) \equiv p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$$

とおくと

$$\arg\max_{j \in A} \max \left(q \in [0,1] : d(\hat{\mu}_j, q) \leq \frac{\log(t)}{T_j(t)} \right)$$

のアームを選ぶ．Regret の上限は UCB1 よりも低く抑えられることが理論的に示されている．また，ベルヌーイ利得の場合で 1 が出る確率が低い場合に Regret が UCB1 よりも低く抑えられることが実験で示されている．

Thompson Sampling [1] は各アームで予想される利得の確率分布からそれぞれサンプリングして，その値が最も高かったアームを選ぶ手法である．利得の確率分布はアーム j の勝ち数，負け数をそれぞれ α_j, β_j としてベータ分布 $B_e(\alpha_j + 1, \beta_j + 1)$ を用いる．Regret の上限は KL-UCB と同等と証明され，また KL-UCB と同様の実験で Regret が KL-UCB や UCB1 より低いことが示されている [6]．

UCB1-Tuned [8] は UCB1 の第 2 項を少し改変し，分散も考慮する手法で，

$$V_j = \hat{\sigma}_j^2 + \sqrt{\frac{2\log(s)}{n_j}}$$

として

$$\arg\max_{j \in A} \left(\hat{\mu}_j + \sqrt{\frac{\log(s)}{n_j} \min(\frac{1}{4}, V_j)} \right)$$

のアームを選ぶ手法である．Regret の理論的な保証が無いものの，UCB1 より Regret が低く抑えられることが実験的に示されている．

2.2 モンテカルロ木探索 (MCTS)

MCTS はシミュレーションを行い，その結果に基づき，木を探索する手法である．UCB1 を MCTS に応用した手法が UCT [7] である．UCT では探索木のどの葉からシミュレーションするかの決定を多腕バンディット問題の変種とみなして，UCB1 をその決定に用いる．UCT ではまず，根から葉に到達するまで，UCB 値が最大の手を繰り返し選んでいき，葉に到達したら，シミュレーションを行う（これをプレイアウトと呼ぶ）．シミュレーションの際はゲームが終了するまでランダムに手を選ぶ．利得は一般に勝ちが 1 で負けが 0，引き分けは 0.5 として与えられる．尚，そのノードを訪問した回数が閾値を超えていたら探索木を成長させる．そして，シミュレーション結果（利得）を親，先祖に伝えて，訪問数や平均利得を更新する．それを繰り返し行うことで評価の精度を高めていく．そして，既定数プレイアウトを行ったら探索を終了し，最終的な回答として手を選ぶ．囲碁プログラムでは回答として最多訪問数の手を選ぶのが一般的である．

MCTS ではシミュレーションを開始する節点が訪問する毎に異なる可能性があるため，葉以外の節点では，利得の分布は訪問する毎に異なる．その点が多腕バンディット問題と異なる．本研究では，UCB1 以外も MCTS に応用し，実験からその有効性を議論する．

2.3 P-game

P-game は元々前向き枝刈りについての分析をするために考案された仮想ゲーム [9] でそれが改変されて，MCTS の評価に利用されている [7]．P-game 木は MinMax 木であり，木の中の各辺（エッジ）には，それぞれ数値をランダムに割り当てて，木を作成する．以後この数値をエッジスコアと呼ぶこととする．エッジスコアはその辺を辿る（手を選ぶ）ことで Max プレイヤがどれだけ勝ちに近づくか

を表す値である．このゲームでは，初期局面（根節点）から終端局面（終端節点）に至るまでのパスのエッジスコアの総和（スコアと呼ぶ）が正だとその終端局面は Max プレイヤの勝ち，負だと負け，0 だと引き分けとして勝敗を決める．

3. P-game の改変

P-game は，初期局面でどんな手を選んでもスコアの MinMax 値が正（理論上勝ち），もしくは負（理論上負け）のゲームが出来る可能性がある．本研究では，初期局面の最善手のスコアの MinMax 値が 0（理論上引き分け）となるように，全ての終端節点のスコアを一律に加減をした．但し，初期局面に於いて，2 つ以上の最善手が出来る可能性があり，その場合は除いた．つまり，本研究での対象とした P-game 木では最善手以外をとると理論上負けになる．本研究と同様に P-game を改変し，理論上引き分けとなるただ 1 つの最善手と，理論上負けとなる複数の次善手がある P-game は [4] にて利用されている．（但し， [4] では，必ず 1 つの手のエッジスコアは 0 であるが，本研究の場合はそうとは限らない．）

また，本研究の P-game 木は全ての節点で分岐数が同じで，全ての終端節点までの深さが同じとした．

4. 実験

本研究では，MCTS での探索の失敗度合いの指標として最終誤答時刻という量を採用する．本節では，まず，その妥当性について検証した．次に，最終誤答時刻の観点で UCB1 よりも KL-UCB，Thompson Sampling，UCB1-Tuned が優れていることを示した．更に，Δ' という木の特徴量を定義し，Δ' と最終誤答時刻の関係を調査した．

4.1 最終誤答時刻と誤答率

本研究では，探索の失敗度合いの指標として Regret は用いない．それは，最後に正しい手を選べるならそれまでの試行錯誤での失敗は不利益にならないためである．代わりに，最終誤答時刻という量を採用する．本節では，最終誤答時刻より一般的な指標と思われる誤答率との類似点について述べる．

まず，誤答率について説明する．本研究の MCTS では，最終的な回答として最多訪問数の手を選ぶとした．誤答率とは，回答として最善手を選べない場合を誤答として，探索回数に占める，誤答数の割合であると定める．今までに行ったプレイアウトの数を時刻と呼ぶことにして，各時刻での誤答率は，現時刻で探索終了と仮定すると最善手を選べない場合をその時刻で誤答として算出する．

実際に誤答率についての実験を行った．実験では分岐数 4 深さ 6 の P-game 木を作成し，UCB1, KL-UCB (KL), Thompson Sampling (TS)，UCB1-Tuned (Tuned) をそれぞれ MCTS に応用して探索し，各時刻での誤答率を計測した．また，参考のため $\alpha\beta$ 探索での誤答率も計測した．$\alpha\beta$ 探索では，時刻は探索節点数とし，誤答率は読み切らない限り，全て誤答として算出した．尚，$\alpha\beta$ 探索のノードの探索順は等確率で決めた．本研究の P-game では，Max プレイヤの手にはそれぞれ $[0, 127]$ のエッジスコア，Min プレイヤの手には $[-127, 0]$ のエッジスコアを一様な確率で割り当てた．探索は各 P-game 木に対し，1000 回行なっている．誤答率は 2 の累乗の時刻で測った．また，最終誤答時刻は 1000 回探索の平均をとった．各探索では，プレイアウトは 5461 回行った．

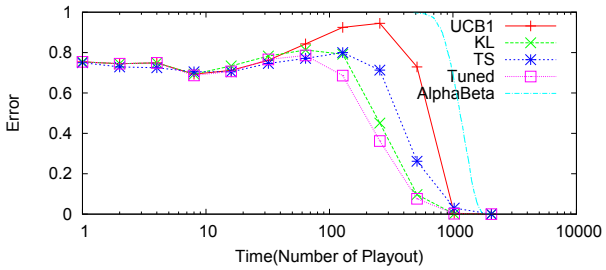

図 1 誤答率が上昇する木での誤答率と時刻

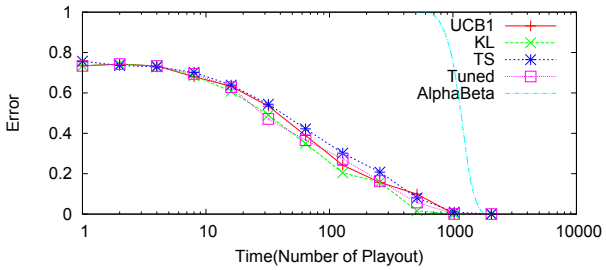

図 2 誤答率が緩やかに低下する木での誤答率と時刻

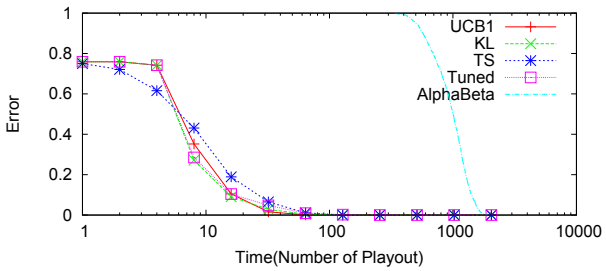

図 3 誤答率がすぐに低下する木での誤答率と時刻

誤答率と時刻のグラフを描画した結果，**図 2** では KL-UCB と UCB1-Tuned が少し誤答率が低く抑えられている．また，**図 1** では UCB1 の誤答率が 9 割近くまで上昇したのに対して他のアルゴリズムでは微増した程度だった．更に，**図 3** では，Thompson Sampling だけが他と異なり滑らかに誤答率が減少したが，他はほとんど差が付かなかった．

次に最終誤答時刻について説明する．最終誤答時刻とは「現時点で探索終了と仮定すると最善手を選べないような時刻の内の最大値」で，常に最善手を選べた場合は 0 と定

める．

誤答率と最終誤答時刻には以下の関係がある．誤答率は毎時刻計測すると仮定すると「最終誤答時刻をまだ過ぎていない探索の割合 $\geq$ 誤答率」が成り立つ．特に誤答率が 0 の時，その割合は 0 になる．

実際に各木での UCB1，KL-UCB，Thompson Sampling，UCB1-Tuned の最終誤答時刻を計測すると図 2 の木ではそれぞれ，205.738，126.482，181.74，146.822，となり，図 1 の木では 675.619，287.155，415.429，251.636，となり，図 3 の木では 10.325，8.494，10.496，8.73 となった．

総括すると，最終誤答時刻は誤答率が速く下降する場合は小さく，誤答率が上昇する場合は上昇度合いの大きいほど大きな値で上昇度合いの小さいほど小さな値となった．以上から最終誤答時刻は誤答率と類似の傾向があり，最終誤答時刻は探索の失敗度合いの良い指標になっていると期待できる．

4.2 MCTS アルゴリズムの比較

本節では最終誤答時刻の観点で UCB1 よりも KL-UCB，Thompson Sampling，UCB1-Tuned が優れていることを示す．但し，他のアルゴリズムは UCB1 より優れているが，UCB1 の方が優れた結果となった木もあることや概して各アルゴリズムは木の特徴によって，似たように性能が落ちることも示す．

実験での木の分岐数－深さは 4-6, 4-8, 8-6, 2-18 として 4.1 節と同様に UCB1，KL-UCB，Thompson Sampling，UCB1-Tuned の各アルゴリズムを MCTS に応用し，P-game を探索した際の最終誤答時刻を計測した．P-game 木は 100 本作成し，探索は各木で 100 回行った．プレイアウト回数は木の節点数つまり，分岐数を b 深さを d として，$\frac{b^{d+1}-1}{b-1}$ とした．図 1 等の結果からこの回数だけプレイアウトすれば誤答率は 0 になると期待できる．実験の他の設定は 4.1 と同じとした．尚，最終誤答時刻は 100 回探索の結果の平均をとっている．

まず，分岐数 4 深さ 6 の木で最終誤答時刻を測定した．そして，各木の結果を横軸を UCB1 での，縦軸をその他のアルゴリズムでの最終誤答時刻としてプロットした．このグラフのプロットポイント 1 つは 1 本の木での結果に対応する．この木ではプレイアウトは 5461 回行った．結果を**図 4** に示す．

計測の結果，最終誤答時刻において，UCB1 が他のアルゴリズムを下回った木がいくつかあったものの，全体的に UCB1 よりも他のアルゴリズムの方が下回った．実際に，100 本の木の内で各アルゴリズムが UCB1 の最終誤答時刻を下回る回数を調査した．KL-UCB の最終誤答時刻が UCB1 を下回ったのは，91 回であった．100 本全ての木での最終誤答時刻の平均をとると UCB1，KL-UCB それぞれ 193.7979，114.8263 で，平均値でも KL-UCB の方が

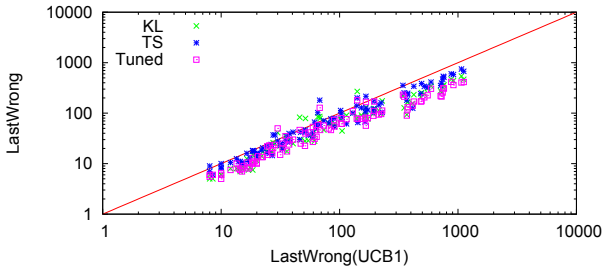

図 4 分岐数 4 深さ 6 の木での最終誤答時刻の平均（UCB1 と他のアルゴリズム）

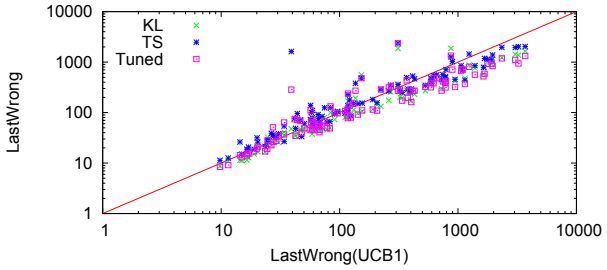

図 5 分岐数 4 深さ 8 の木での最終誤答時刻の平均（UCB1 と他のアルゴリズム）

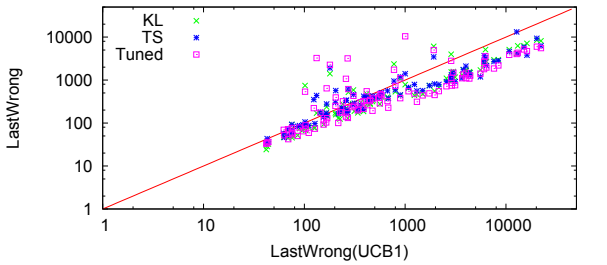

図 6 分岐数 8 深さ 6 の木での最終誤答時刻の平均（UCB1 と他のアルゴリズム）

下回った．Thompson Sampling では，UCB1 を下回った回数は 45 回であったが，平均値は 167.0017 となり UCB1 を下回った．UCB1-Tuned では 85 回下回り，平均もまた 113.1694 で下回った．

木の大きさを変えた場合の影響を調査するため，今度は分岐数 4 のまま，深さ 8 に増やした場合のグラフを描く．結果を**図 5** に示す．この場合のプレイアウト回数は 87381 回である．

最終誤答時刻は全体的に UCB1 よりも小さくなった．同様に UCB1 の最終誤答時刻を下回った数と全ての木で平均の最終誤答時刻を比べる．下回った数は KL-UCB では 79 回，全ての木での平均は UCB1 で 457.5645，それに対して KL-UCB では 296.3221 となり平均値でも下回った．Thompson Sampling では，46 回であったが平均は 394.2873 で UCB1 を下回った．UCB1-Tuned では 72 回で平均は 281.0496 となり，平均値も下回った．

今度は分岐数 8 深さ 6 のデータを取得した．この場合のプレイアウト回数は 299593 である．

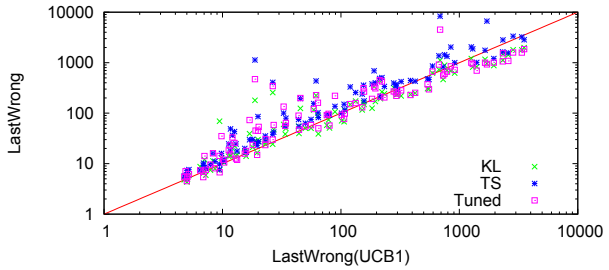

図 7　分岐数 2 深さ 18 の木での最終誤答時刻の平均（UCB1 と他のアルゴリズム）

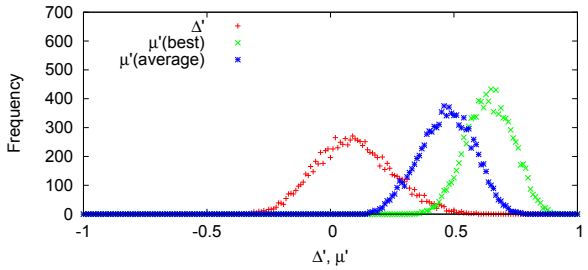

図 8　分岐数 4 深さ 6 の木での Δ' と μ'（最善手の及び，手の平均）のヒストグラム

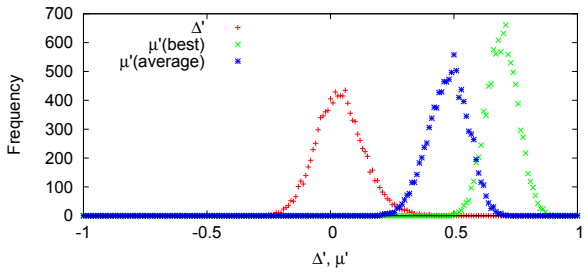

図 9　分岐数 8 深さ 6 の木での Δ' と μ'（最善手の及び，手の平均）のヒストグラム

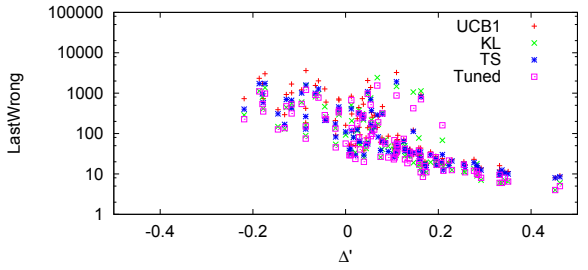

図 10　分岐数 4 深さ 6 の木での最終誤答時刻の平均と Δ'

結果を図 **6** に示す．この場合でも図 4 と同様の傾向となった．また，同様に UCB1 の最終誤答時刻を下回った回数及び，全ての木での最終誤答時刻の平均を調査した．KL-UCB，Thompson Sampling，UCB1-Tuned で下回った回数はそれぞれ 85，78，79 回となり，全て半数以上となった．平均をとるとそれぞれ 1154.7144，1116.5467，1093.2141 となり，UCB1 の平均 2332.7852 を全て下回った．

分岐数 2 深さ 18 の木でも同様に最終誤答時刻を計測し，結果を図 **7** に示す．この場合もまた図 4 と同様の傾向となった．また，同様に UCB1 の最終誤答時刻下回った回数及び，全ての木での最終誤答時刻の平均を調査した結果，UCB1 に対して KL-UCB，Thompson Sampling，UCB1-Tuned が下回った回数はそれぞれ 47，11，41 となり，UCB1 の平均 395.5003 に対して，それぞれ 287.6293，597.723，328.9482 となった．下回った回数は半数以下ではあるが，Thompson Sampling 以外の平均値は UCB1 の平均値を下回った．

総括すると，UCB1 と他のアルゴリズムの成績の違いを調査した結果，図 5，図 6，図 7 から，概ね UCB1 よりも良い成績を示した．特に KL-UCB と UCB1-Tuned は多くの木で UCB1 を上回る成績となった．しかしながら、UCB1 で最終誤答時刻が大きい木は KL-UCB や UCB1 − Tuned においても大きい傾向にある．次節ではこの点について詳しく述べる．

4.3　木の特徴と MCTS の性能の関係

前節で P-game 木の違いによって各アルゴリズムの最終誤答時刻は大きく影響されることを確認した．本節では Δ' という木の特徴量を定義し，その値と最終誤答時刻の関係について調査した．

Δ' は，${\mu'}_i$ を根の子節点 i からシミュレーションした場合の利得の期待値とし，根の子節点の集合を $\mathbb{K}$ として，$\Delta' \equiv {\mu'}_{i^*} - \max_{i\in\mathbb{K}, i\neq i^*} {\mu'}_i$ と定め，$\arg\max_{i\in\mathbb{K}, i\neq i^*} {\mu'}_i$ の手を次善手と呼ぶこととする．Δ' は式 2.1 等が平均利得を用いていることから，探索木を 1 手分だけ成長させる場合に最善手を見つけ出すおおよその難しさを表していると期待できる．Δ' は負にもなり得るが，その場合も同様に負の方向に大きいほど次善手の最善手との間違えやすさを表していると期待できる．

実際に各分岐数・深さで P-game 木を作成した場合の Δ' の分布を調査するため，P-game 木を 10000 本作成し，ヒストグラムを作成した．また，参考のため，最善手の μ' 及び，手の μ' の平均も計測してヒストグラムにした．分岐数-深さは 2-4,2-6,2-18,4-4,4-6,4-8,8-6 で計測した．その内 4-6，8-6 の結果を掲載する． その結果，図 **8** よりも図 **9** の方が Δ' の分散が小さくなるように，分岐数，深さとも大きいほど分散が小さくなる傾向を示した．平均も同様の傾向を示した．

相関関係を見るため，Δ' を横軸として，最終誤答時刻の平均を縦軸（対数）として片対数グラフを作成した．データは 4.2 節のものと同一である．分岐数 4 深さ 6 の時の結果を図 **10** に示す．

結果はほぼ一直線上になった．また，各アルゴリズムでの改善度合いが目立たない，つまり木の間での最終誤答時刻の差に比べて，各アルゴリズムでの差が小さいと言える．ばらつき度合いは Δ' が小さいほど大きくなった．

同様に，分岐数 8 深さ 6，分岐数 2 深さ 18 の木でも測定

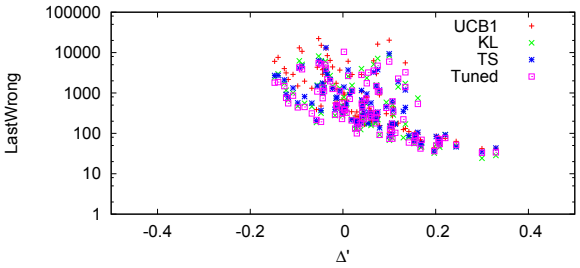

図 **11**　分岐数 8 深さ 6 の木での最終誤答時刻の平均と Δ'

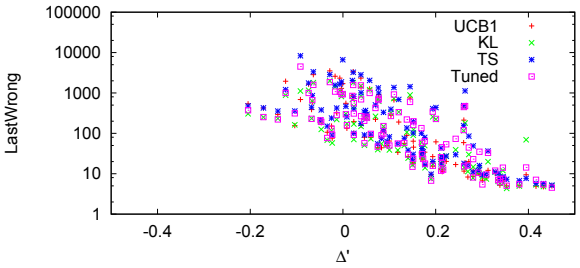

図 **12**　分岐数 2 深さ 18 の木での最終誤答時刻の平均と Δ'

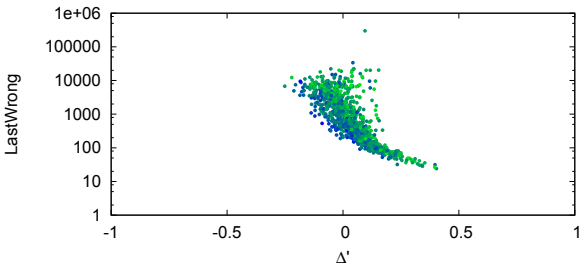

図 **13**　分岐数 8 深さ 6 の木での最終誤答時刻の平均と Δ'

した結果をそれぞれ図 **11**，図 **12** に示す．各木での最終誤答時刻は Δ' が大きい時は，ばらつきが小さいものの，全体的に図 10 と比べてばらつきが大きくなった．

総括すると各アルゴリズムで Δ' と最終誤答時刻には負の相関が見られたが，木が大きい場合には相関の度合いが低くなった．

今度は UCB1 のみを用いて，1000 本の木に対して，最終誤答時刻を計測した．その際，次善手の μ' と次善手の子の μ' の最小値の差を元にして色分けした．この値は探査時の次善手の平均利得の下がりやすさを表していると期待できる．値が最小値の時 RGB で 0，255，0，最大値の時に 0，0，255 となるように線形に変化させた．色分けによってばらつきを説明出来るかを検討する．実験の他の設定は 4.1 節のものと同じとした．

グラフを図 **13** や図 **14** に示す．同じ Δ' でも最終誤答時刻が低いほうが青で高い方が緑になっていることが多いが，青も緑も入り混じっている．

5.　おわりに

本稿の主要な成果は，モンテカルロ木探索 (MCTS) に

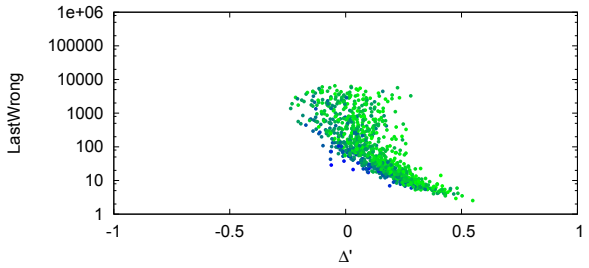

図 **14**　分岐数 2 深さ 18 の木での最終誤答時刻の平均と Δ'

応用した場合でも KL-UCB と UCB1-Tuned, Thompson Sampling の順にどれも UCB1 より優れることを，P-game を用いた実験により明らかにしたことにある．但し，実験に用いた P-game の木により各アルゴリズムの性能にばらつきがあり，また木の違いによる成績の変化が大きいことにも注意を払う必要がある．

最善手からシミュレーションした際の利得の期待値とその他の手からの場合の期待値の最大値との差 (Δ' と呼ぶ) が MCTS の性能に大きな影響を与えることが分かった．Δ' 以外にも性能に影響する要因はありそうであるが，P-game でアルゴリズムの性能評価を行う場合は Δ' を揃えたほうが良さそうであると結論づけられる．

参考文献

[1] Agrawal, S. and Goyal, N.: Analysis of Thompson sampling for the multi-armed bandit problem, *arXiv preprint arXiv:1111.1797* (2011).

[2] Auer, P., Cesa-Bianchi, N. and Fischer, P.: Finite-time Analysis of the Multiarmed Bandit Problem, *Machine Learning*, Vol. 47, No. 2-3, pp. 235–256 (2002).

[3] Baudiš, P.: Balancing MCTS by Dynamically Adjusting the Komi Value, *ICGA Journal-International Computer Games Association*, Vol. 34, No. 3, p. 131 (2011).

[4] Furtak, T. and Buro, M.: Minimum Proof Graphs and Fastest-Cut-First Search Heuristics., *IJCAI*, pp. 492–498 (2009).

[5] Garivier, A. and Cappé, O.: The KL-UCB algorithm for bounded stochastic bandits and beyond, *arXiv preprint arXiv:1102.2490* (2011).

[6] Kaufmann, E., Korda, N. and Munos, R.: Thompson sampling: An asymptotically optimal finite-time analysis, *Algorithmic Learning Theory*, Springer, pp. 199–213 (2012).

[7] Kocsis, L. and Szepesvári, C.: Bandit based monte-carlo planning, *Machine Learning: ECML 2006*, Springer, pp. 282–293 (2006).

[8] Kuleshov, V. and Precup, D.: Algorithms for multi-armed bandit problems, *arXiv preprint arXiv:1402.6028* (2014).

[9] Smith, S. J. and Nau, D. S.: An analysis of forward pruning, *AAAI*, pp. 1386–1391 (1994).

[10] 今川孝久，金子知適：難しさが手番で異なる局面でのモンテカルロ木探索の性能の改善，ゲームプログラミングワークショップ 2013 論文集，pp. 162–169 (2013).