

# プレイアウト数増加に伴うモンテカルロ木探索の振舞い

竹内聖悟<sup>1,a)</sup>

**概要:** ゲームにおいて探索は重要であり、アルファベータ探索では深さと最善手の関係などが調査されている。一方で、モンテカルロ木探索についてこのような調査は知られていない。本稿では、モンテカルロ木探索の性能改善につながる知見の獲得を目的として、プレイアウト回数増加に伴い最善手選択や勝率などがどのように変化するかを調査する。コンピュータ囲碁を対象として、モンテカルロ木探索における深さと最善手の関係について調査し、最善手変化率や勝率についてアルファベータ探索と同様の結果を得た。また、コンピュータ囲碁ではディープニューラルネットワークを用いるプログラムも多く、ディープニューラルネットワークの有無で差があるかについて調査した。さらに、モンテカルロ木探索は最善手へのプレイアウト数によって指手を選択するため、プレイアウト数に着目した調査を行い、最善手へのプレイアウト数の影響を調査した。

## Behavior of Monte-Carlo Tree Search with Increasing Playouts

SHOGO TAKEUCHI<sup>1,a)</sup>

**Abstract:** Searching is important in games, and the relationship between depth and the best-changes has been investigated in alpha-beta search. On the other hand, such a study on Monte-Carlo Tree search is not known. In this paper, we investigate the best-move and the winning rate change as the number of playouts increases. We investigated the behavior of computer go programs using Monte-Carlo tree search and obtained similar results in terms of the best-change rate and winning rate. Additionally, we investigated the difference between computer go using and not using deep neural networks. We also investigated the influence of the number of playouts in Monte Carlo tree search since the number of playouts to the best move determines the move selection.

### 1. はじめに

モンテカルロ木探索 (Monte-Carlo Tree Search: MCTS) は囲碁で提案され、多くのゲームで使われている [1]. AlphaGo では MCTS と Deep Neural Network (DNN) の組み合わせとその学習フレームワークが提案され、囲碁プロ棋士に勝利を収めた [6]. 更に、学習においてゲームの知識を用いない Alpha Zero アルゴリズムは囲碁だけでなく、将棋やチェスでも成功を収めた [7].

一般的に、探索の深さが増すほどにプログラムは強さを増す。アルファベータ探索を用いて深く探索した時の最善手や評価値の変化については、チェスにおける調査が行われてきた。例えば、Heinz は探索深さと最善手の変更の関係

を調べ [4], Guidらは探索深さと局面評価値の関係について調査した [3]. 前者では、深く読むほど最善手の変更する割合は減少する傾向が、後者では、勝っている時は深く読むことで更に評価値が高く、負けている時は低くなる傾向が確認された。

このような探索の調査は MCTS 単体でも、MCTS と DNN の組み合わせにおいても確認されていない。探索の性質について知ることによって探索性能の改善につながることもあり、このような調査は重要である。例えば、MCTS での探索では勝率の高さとプレイアウト数に基づいて探索する指手がされる。このため、勝率以外の条件は全て同じ、2つの局面が与えられた時、勝率の違いによって探索の振る舞いが変わることがある。また、MCTS では最終的な指手の選択は勝率ではなく、プレイアウト数が最も多い手が選ばれる。このように最善手に対するプレイアウト数が大きな影響をもつ探

<sup>1</sup> 高知工科大学 情報学群  
School of Information, Kochi University of Technology

<sup>a)</sup> takeuchi.shogo@kochi-tech.ac.jp

索であるので、こに着目した調査を行う。例えば、最善手の更新確率が高い状況や低い状況の性質がわかれば、探索の延長や打ち切りなどの制御につながる。このような MCTS の改良につながるような知見を得ることを目的として、調査を行う。

本稿では、囲碁プログラムを対象として、MCTS のプログラムと MCTS に DNN を用いるプログラムを使い、調査を行う。まず、アルファベータ探索で行われた調査の結果が MCTS においても再現するかを調べ、DNN の有無の影響について調査を行う。さらに、探索に大きな影響を持つ、最善手へのプレイアウト数に基づいた調査を行う。

## 2. 関連研究

### 2.1 アルファベータ探索における深さについての調査

Heinz は、チェスプログラムを用いて、探索深さに対する最善手の変化について、最善手変更率や新規最善手率などを調査した [4]。最善手変更率 (Best-Changes rate) とは、深さ  $d$  に対して得られる数値で、深さ  $d$  における最善手と  $d-1$  における最善手を比較した時に最善手が変わっていた割合を示す。新規最善手率 (Fresh Best rate) も深さ  $d$  に対して定まる値で、深さ 1 から  $d-1$  までに現れていない最善手が深さ  $d$  において見つかった割合を示す。深く読むほどに最善手変更率や新規最善手率は低くなるという結果が得られている。

Steenhuizen は、Heinz らの研究が問題集の局面で行われており、詰みなど勝ちのある局面が多く含まれていることに着目した。序盤の局面を対象にチェスプログラムを用いて実験を行い、最善手変化率が深さを増しても一定とならないことなどを示した [8]。

Guid らは局面の影響の他に評価値の影響があると考え、評価値によるグループを作り実験を行った。グループごとに最善手変化率に差があることや、評価値の符号を入れ替えたグループ間では同じような変化率の振る舞いになることなどを示した。更に、勝勢や敗勢の時は、深く読むほどに評価値の絶対値が増加する傾向があることを発見した [2][3]。

最善手変更については、一回の探索中に多く起こるのであれば難しい局面と考えることができる。Sato らは、評価関数調整のためのメタパラメータ最適化の研究において、最善手変更の値をメタパラメータとして用いている [5]。

### 2.2 モンテカルロ木探索

モンテカルロ木探索は囲碁で成功し、幅広い応用がされている [1]。探索資源の割り振りに関して、これまで良い結果を得た箇所とまだ良く調べていない箇所とのトレードオフがあるが、これには Upper Confidence Bound (UCB) を木探索へと適用した Upper Confidence bounds applied to Trees (UCT) が使われる。UCT では探索節点の選択に次の式で表される UCB 値を用いる。

$$UCB-Value_i = \frac{w_i}{n_i} + c\sqrt{\frac{\log n}{n_i}} \quad (1)$$

なお、指手  $i$  について、 $w_i$  はこれまでの探索で勝った数、 $n_i$  はプレイアウト数を表し、 $n$  は  $i$  の親局面のプレイアウト数を表す。

$\frac{w_i}{n_i}$  は勝率を表し、勝率が高い局面ほどこれまでに良いところを選びやすく、 $\sqrt{\frac{\log n}{n_i}}$  は分散に近いもので、これまで選ばれていない箇所ほど選びやすいことに相当する。

### 2.3 モンテカルロ木探索と Deep Neural Network

囲碁においては AlphaGo[6]、チェスと将棋では Alpha Zero[7] がディープニューラルネットワークを用いて成功している。局面評価や方策へとディープニューラルネットワークを用い、これらを MCTS と組み合わせている、以降のプログラムも同様の使い方をしている。

MCTS との組み合わせの際には、UCB 値の代わりに p-UCT 値を用いる。局面評価値に相当するバリューネットワークの値を  $v(s_i)$ 、指手に対するポリシーネットワークの値を  $p_i$  とした時に次の式で表される。

$$pUCT-Value_i = v(s_i) + p_i c \frac{\sqrt{n}}{1+n_i} \quad (2)$$

## 3. 提案手法

アルファベータ探索における研究と同様の振る舞いをするかを調べる他、MCTS に DNN を用いることで差が生じているかを調べる。MCTS においては最終的な指手の選択はプレイアウト数で決まること、MCTS 中の UCT (p-UCT) においても指手へのプレイアウト数が使われるなど、指手へのプレイアウト数が大きな役割を果たしていると考えられるため、プレイアウト数に着目した調査を行う。

アルファベータ探索での調査では、反復深化が進み深さ  $d$  の探索が終わるたびに、最善手と評価値を記録していた。MCTS では探索の制御に深さをうけないため深さ  $d$  の代替値が必要となる他、最善手に対するプレイアウト数に着目するため、その記録も取る。この調査では探索深さの代わりに、プレイアウト数を元にして調査を行う。具体的には、深さの代替値  $d'$  を考え、プレイアウト回数が  $100 \times 2^{d'}$ 、または  $1000 \times 2^{d'}$  に達した時に最善手や勝率、試行回数などの記録を取る。

得られたデータから、次の項目について調査を行う。

- (1) 最善手変更率と新規最善手率について、探索が進むにつれて減少していくか
- (2) 勝率によりグループ分けした時、プレイアウト回数が多くなるにつれ、勝勢なら更に勝率が高く、敗勢なら更に勝率が低くなるか
- (3) 敗勢の時と勝勢の時とで最善手変更率などに差が出るか (アルファベータ探索では絶対値が等しいグループは似た挙動を示した)

- (4) DNN を使うプログラムとそうでないプログラムとで差が出るか  
 (5) 最善手に対するプレイアウト数の違いで、振舞いに差が出るか

1, 2, 3 については、アルファベータ探索における先行研究の追試に相当する。それ以外は、MCTS が持つ特徴からの調査である。

## 4. 実験

囲碁プログラムを対象として実験を行った。囲碁プログラムは以下のものを利用した（いずれも 2019 年 3 月に取得）：Leela Zero<sup>\*1</sup>, ELF<sup>\*2</sup>, Ray<sup>\*3</sup>, Pachi (11.00 Retsugen)<sup>\*4</sup>。最新の Pachi では DNN を利用しているが、DNN の有無の違いを見ることを目的としているため、バージョンの古いものを利用した。なお、Pachi, Ray とともに、指手に対するレーティングやパターンに基づくボーナスなどが使われており、探索中の指手選択では純粋な UCB 値からは変更がされている。一方の DNN を用いる ELF と Leela Zero では p-UCT が使われている。

実験局面には Kiseido Go Server の棋譜を用いた<sup>\*5</sup>。両者が六段、または片方が七段以上のアマチュアの棋譜が集められており、2018 年 10 月から 12 月の棋譜でコミ 7.5 目の 1,673 棋譜を使い、棋譜中の 10 手目以降の局面を対象として、20 手毎にランダムに 1 局面を取り出し、約 12,000 局面を得た（プログラムごとに多少のずれがある）。

実験環境には、CPU: AMD Ryzen Threadripper 2990WX 32-Core Processor, メモリ: 128GB, GPU: GeForce RTX 2080 Ti のワークステーションを用いた。

実験においては、DNN を用いるプログラムとそうでないプログラムとでは一回のプレイアウトにかかる時間が 10 倍以上異なるため、DNN を用いるプログラムでは 100 回から 25,600 回まで、プレイアウト回数が 2 倍になる度に最善手と勝率、最善手へのプレイアウト数を記録した。DNN を用いないプログラムでは、1,000 回から 128,000 回までで同様の記録を取った。

以下、探索の初回とは、DNN を用いるプログラムはプレイアウト回数 100 の時、DNN を用いない場合はプレイアウト回数が 1,000 の時を指し、探索の中盤とは、DNN を用いるプログラムはプレイアウト回数 1,600 の時、DNN を用いない場合はプレイアウト回数が 16,000 の時を指し、探索の最終回とは、DNN を用いるプログラムはプレイアウト回数 25,600 の時、DNN を用いない場合はプレイアウト回数が 128,000 の時を指す。

<sup>\*1</sup> <https://github.com/leela-zero/leela-zero>

<sup>\*2</sup> <https://github.com/pytorch/ELF>

<sup>\*3</sup> <https://github.com/kobanium/Ray>

<sup>\*4</sup> <https://github.com/pasky/pachi/tree/pachi-11.00-retsugen>

<sup>\*5</sup> <https://www.u-go.net/gamerecords/>

### 4.1 勝率やプレイアウト数の分布

まず、集めたデータについての基礎的な情報をプログラム毎に示す。勝率と最善手へのプレイアウト数の分布を、図 1 にまとめる。それぞれ探索の初回と最終回のみを示す。

上に DNN を用いないプログラム、下に DNN を用いるプログラムの結果をまとめたが、図 1(a), 1(b) からは、DNN を用いるプログラムでは勝率 0.0 と 1.0 に、そうでないプログラムでは勝率 0.5 に、それぞれピークがあるという明確な差が確認出来る。最善手へのプレイアウト数についても、図 1(d) に示すように、DNN を用いるプログラムではほぼ全プレイアウト回数とその最善手の試行に使われたケースに、そうでないプログラムでは全体の 0.5 よりも少ないケースにピークがある。

### 4.2 最善手変更率、勝率

アルファベータ探索のチェスプログラムで行われた実験について、MCTS の囲碁プログラムで同様の結果が得られるか調べる。

まず、最善手変更率と新規最善手率について調査した。結果を表 1 にまとめた。表中の BC は最善手変更率、FB は新規最善手率を表し、PO 数はプレイアウト (Playout) 数を表す。

Pachi, Ray, ELF ではアルファベータ探索と同様に、プレイアウト回数が増加するに伴い、割合が減少する傾向がある一方で、Leela Zero ではその傾向が得られなかった。また、DNN の有無による差について見ると、プレイアウト回数が大きく異なるために単純な比較はできないが、DNN を用いるプログラムは最善手変化の割合が明確に低く、DNN による探索改善の効率が伺われる。

表 2 にプレイアウト回数に対する勝率の関係を示す。なお、「勝勢」は、探索最終回の勝率が [0.75, 1] となった探索において、各プレイアウト回数で得られた勝率の平均値を得たもので、「敗勢」は、勝率が [0, 0.25] となった探索で得られたものである。

Guid らの研究では、勝勢の時と敗勢の時は、深さが増すにつれそれぞれ評価値が増加と減少する傾向が得られていた。MCTS の囲碁プログラムの結果である表 2 においても同様の結果が確認できる。このことは、プレイアウト数の異なる勝率を単純に比較することで問題が起り得ることを示唆している。

なお、勝率 0.5 を中心として対称となるペアでの最善手変化率などの振る舞いに着目したが、全体的にはペアであることに意味がある結果は得られなかった。例として、Pachi の結果を表 3 に示す。勝率が高いほど最善手変化率が低くなる傾向がある。

### 4.3 プレイアウト数に基づくデータ

最善手へのプレイアウト数に着目した調査を行う。

図 2 に勝勢時と敗勢時のプレイアウト回数と勝率の関係

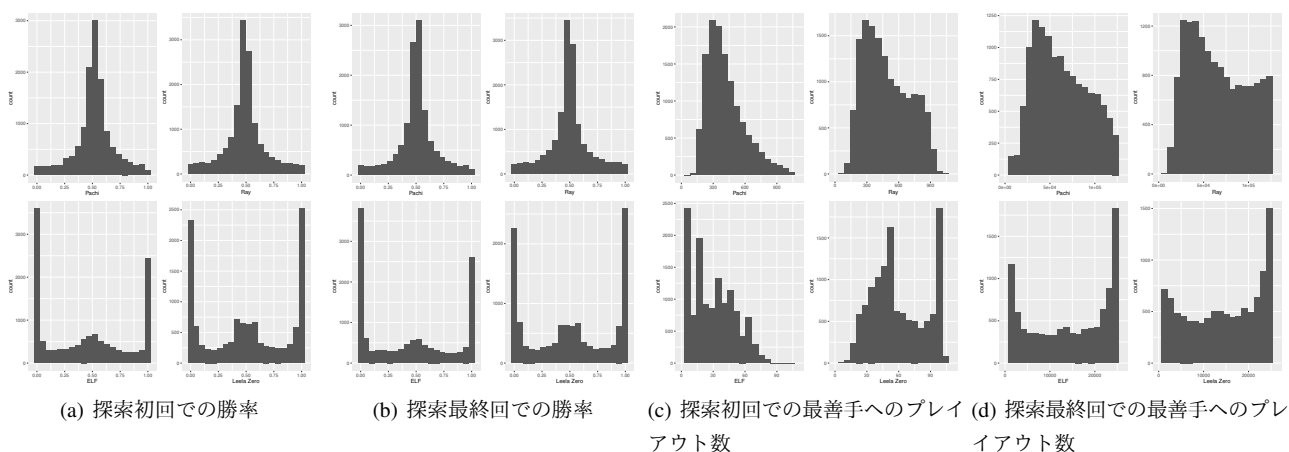


図1 勝率と、最善手へのプレイアウト数の分布. (左上: Pachi, 右上: Ray, 左下: ELF, 右下: Leela Zero)

表1 最善手変更率 (BC) と新規最善手発見率 (FB)

PO 数	Pachi		Ray		PO 数	ELF		Leela Zero	
	BC	FB	BC	FB		BC	FB	BC	FB
2000	0.358	0.358	0.214	0.214	200	0.182	0.182	0.054	0.054
4000	0.359	0.321	0.218	0.186	400	0.118	0.109	0.055	0.046
8000	0.347	0.292	0.211	0.164	800	0.112	0.094	0.060	0.049
16000	0.335	0.261	0.212	0.150	1600	0.094	0.076	0.062	0.049
32000	0.299	0.237	0.215	0.156	3200	0.079	0.062	0.064	0.050
64000	0.274	0.214	0.209	0.144	6400	0.069	0.049	0.075	0.058
128000	0.253	0.194	0.201	0.138	12800	0.074	0.051	0.068	0.051
					25600	0.080	0.054	0.076	0.056

を、最善手へのプレイアウト数でグループ分けした結果を示す。x軸はプレイアウト回数、y軸は勝率を表し、破線は最善手へのプレイアウト数の割合が高い場合、点線は低い場合を表す。最善手へのプレイアウト数が多いほど最善手変更率は低く、プレイアウト数が少ないほど変更率が高くなる傾向がどのプログラムにも現れた。特に ELF は傾向が顕著になっている。

まず、プレイアウト回数が少ない時の勝率から最終的な勝率をどの程度予想できるかを調べるため、プレイアウト回数の異なる勝率同士の比較を行った。結果を図3にまとめる。探索初回（または中盤）の勝率を x 軸に、探索最終時の勝率を y 軸にプロットしたもので、最終的な最善手へのプレイアウト数について高いもの赤、低いものを黄色で色付けた。多くの点は  $y = x$  上にあることや、探索の初回よりも中盤の方がよりその傾向が強くなることが確認できる。また、ELF は探索の初期からは予測精度が低いと考えられる結果が得られた。

最善手へのプレイアウト数に着目すると、Pachi や Ray ではプレイアウト数が高いものほどグラフの上方にあり、最終的な勝率が高くなっている傾向がある。一方の ELF や Leela Zero ではプレイアウト数については顕著な傾向は見られなかった。

次に、最善手へのプレイアウト数の変化について調べる

ため、最善手へのプレイアウト数が全体に占める割合の比較を行い、結果を図4にまとめた。探索初回（または中盤）のプレイアウト数率を x 軸に、探索最終時のプレイアウト数率を y 軸にプロットしたもので、最終的な勝率について高いもの赤、低いものを黄色で色付けた。ほとんど相関は見られない。

Pachi では最終的なプレイアウト数が高い点ほど図の上方にあり、前の結果と合致する。Leela Zero については、図の下方で  $y = x$  の右下に勝率の高い点が、図の右下に勝率の低い点が集中しているという特徴が見られた。探索初期の最善手に対するプレイアウト数が全体の半分以下で、最終的なプレイアウト数の割合が初期より低いケースにおいて、勝率が高い点が多いこととまた、探索初期の最善手に対するプレイアウト回数が半分以上だが、最終的な最善手へのプレイアウト数の割合が半分以下となるケースにおいて、勝率の低い点が多いことが読み取れる。

続いて、最善手へのプレイアウト数で条件を分けた時の、最善手変更率の結果を図5にまとめる。x軸に全体のプレイアウト回数、y軸に最善手変更率をプロットしている。図中の実線、破線、点線は、全データでの結果、最善手へのプレイアウト回数が全体に対して 0.75 以上のデータ、0.25 以下のデータに対応する。最善手へのプレイアウト数については、探索初回と探索最終回のそれぞれで図5(b), 5(c)にまと

表2 勝率の変化

PO 数	Pachi		Ray		PO 数	ELF		Leela Zero	
	敗勢	勝勢	敗勢	勝勢		敗勢	勝勢	敗勢	勝勢
1000	0.136	0.876	0.126	0.860	100	0.063	0.928	0.053	0.943
2000	0.133	0.876	0.125	0.862	200	0.055	0.936	0.053	0.944
4000	0.133	0.876	0.125	0.865	400	0.050	0.944	0.052	0.945
8000	0.133	0.875	0.124	0.867	800	0.046	0.949	0.052	0.946
16000	0.129	0.875	0.124	0.870	1600	0.043	0.952	0.051	0.947
32000	0.124	0.876	0.123	0.872	3200	0.042	0.954	0.051	0.948
64000	0.120	0.878	0.123	0.875	6400	0.041	0.955	0.051	0.949
128000	0.118	0.879	0.122	0.877	12800	0.041	0.955	0.050	0.950
					25600	0.041	0.956	0.050	0.951

表3 勝率でグループ分けした、最善手変化率の変化 (Pachi)

PO 数	[0, 1/6]	[1/6, 1/3]	[1/3, 1/2]	[1/2, 2/3]	[2/3, 5/6]	[5/6, 1]
2000	0.221	0.328	0.366	0.369	0.355	0.280
4000	0.232	0.355	0.369	0.376	0.363	0.270
8000	0.311	0.364	0.348	0.357	0.347	0.226
16000	0.419	0.388	0.348	0.318	0.317	0.195
32000	0.444	0.328	0.302	0.299	0.275	0.163
64000	0.368	0.284	0.287	0.275	0.244	0.104
128000	0.335	0.312	0.278	0.247	0.212	0.068

めた。

いずれのプログラムでも、最善手にプレイアウトが集中している時ほど最善手変更率が低く、最善手にプレイアウトがあまり集中していない時は最善手変更率が高くなる傾向が得られている。探索初回の最善手プレイアウト数が高い場合は、探索が進むに連れて、変更率がやや高くなる傾向が見られる。

#### 4.4 結果のまとめと考察

調査した結果をまとめると次のようになる。

- (1) 最善手変更率と新規最善手率について、Leela Zero 以外のプログラムは探索が進むにつれて減少していく
- (2) 勝率によりグループ分けした時、プレイアウト回数が多くなるにつれ、勝勢なら更に勝率が高く、敗勢なら更に勝率が低くなった
- (3) 敗勢の時と勝勢の時とで最善手変更率などに差は出るが、グループ間で振る舞いが似ることはなかった
- (4) DNN の有無で、勝率や最善手へのプレイアウト数など基礎的なデータから差が生じた
- (5) 最善手に対するプレイアウト数の違いによって、最善手変更率に差が生じた

アルファベータ探索での先行研究との比較結果 (1), (2) について述べる。最善手変更率および新規最善手率は Leela Zero 以外のプログラムで同じ傾向の結果が得られた。深さが増すほどに最善手変化率が低下する点については、探索節点の展開に伴う節点数の指数的增加による効率減が原因と考えられる。また、勝率でグループ分けした場合に深く読むほど、勝勢では勝率が高く、敗勢では低くなる結果が得

られた。勝勢の局面で深く読めば勝ちに近づくことから自然な結果であり、アルファベータ探索と同じ結果が得られるのは自然と言える。このことは、プレイアウト回数が多い指手と少ない指手とでは勝率を単純に比較すると正しくないケースが存在することを示している。

この実験では、(3) のようにアルファベータ探索で確認された評価値の対称性効果が確認できなかった。Pachi の結果 (表 3) はその例で、最善手変更率も新規最善手率が勝率グループを 0.5 で折り返したペアで挙動が異なる。Pachi の最善手へのプレイアウト数と勝率の間に相関があるとすれば、プレイアウト数が高いほど最善手変更率が低くなる結果から説明ができる。図 3 を見ると、Pachi の最善手へのプレイアウト数と勝率の間には相関があるように見え、説明ができる。また、MCTS では探索時に UCB 値など勝率とプレイアウト数を利用して探索手を選んでいるため、勝率が対称の関係にある時、探索の挙動は非対称になることが、グループ間で振る舞いが似ることのなかった原因として考えられる。

(4) の DNN の有無の影響については、勝率やプレイアウト数の分布から明確に差があった。DNN を用いるプログラムは最善手変化率が低く、ネットワークの組み合わせによる探索の精度が高いことが伺われる。

(5) については、MCTS において最善手へのプレイアウト数が必要な影響を持つと考えて行った調査で、予想した結果が得られた。最善手へのプレイアウト数の割合が低い局面は比較的難しい局面と考え、探索延長を行う、その逆にプレイアウト数の割合が高い局面では探索を打ち切るなどの、探索制御への利用が考えられる。

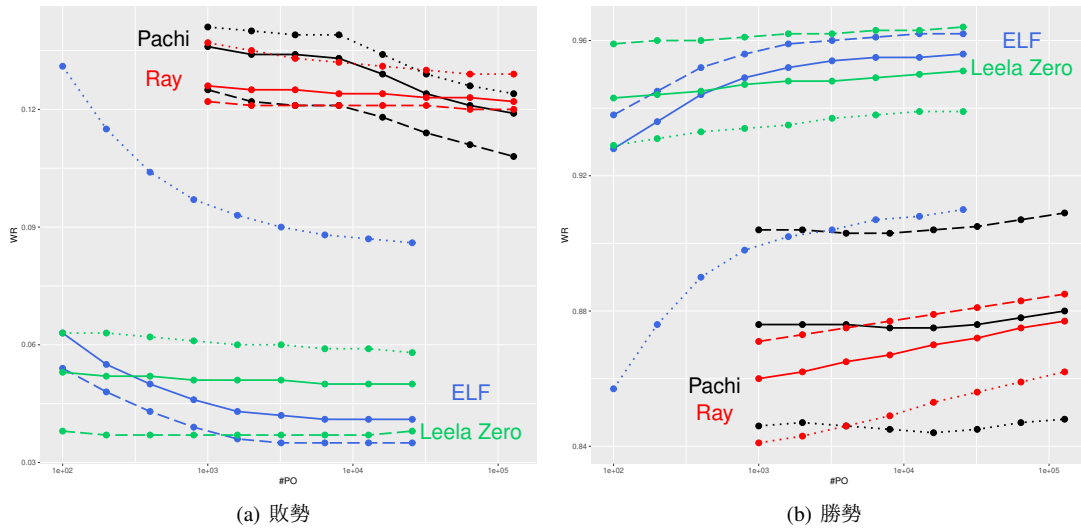
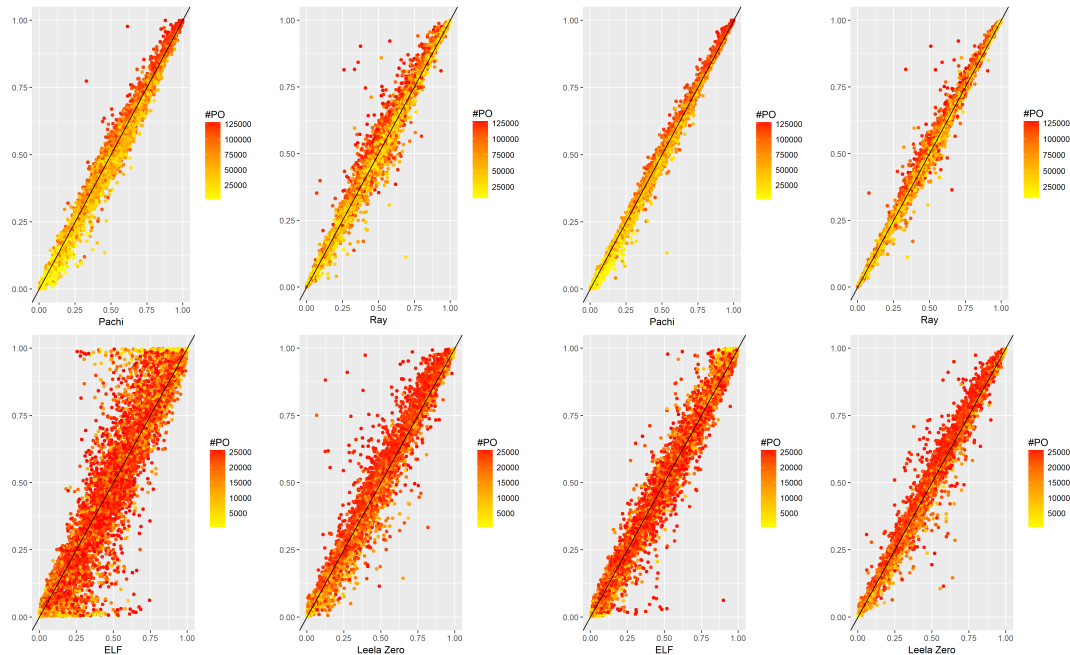


図2 最善手へのプレイアウト数で条件分けした時の、プレイアウト回数に対する勝率の変化。(a), (b) はそれぞれ敗勢と敗勢の時の変化。最善手へのプレイアウト回数が全体に対して占める割合が0.75以上のデータを破線, 0.25以下は点線で, 全データは実線でそれぞれ示している。



(a) 初回対最終回, 最終回の最善手へのプレイアウト数で色分け (b) 中盤対最終回, 最終回の最善手へのプレイアウト数で色分け

図3 各プログラムの、プレイアウト回数の異なる勝率の比較。最善手へのプレイアウト数で色分けを行った。(左上: Pachi, 右上: Ray, 左下: ELF, 右下: Leela Zero)

## 5. おわりに

MCTSについて、DNNを用いる場合も含め、プレイアウト回数の増加に伴い、最善手変更の割合や評価値についての振舞いがどのようになるかを調査した。コンピュータ囲碁における実験から、アルファベータ探索での研究と同様の結果とそうでない結果が得られた。MCTSでは最善手へのプレイアウト数が指手選択に大きな影響を持つため、これに着目した実験を行い、最善手へのプレイアウト数が少

ない時に最善手変更割合が高くなることなどが確認できた。また、DNNを用いるプログラムとそうでないプログラムとの間に大きな差があることも確認できた。

今後の課題としては、MCTSでは最善手へのプレイアウト数に着目したように、アルファベータ探索においても探索ノード数に着目して調査を行うこと、今回得られた知見を元にMCTSの改善を行うことが挙げられる。

謝辞 本研究はJSPS 科研費 17K12807の助成を受けたものです。

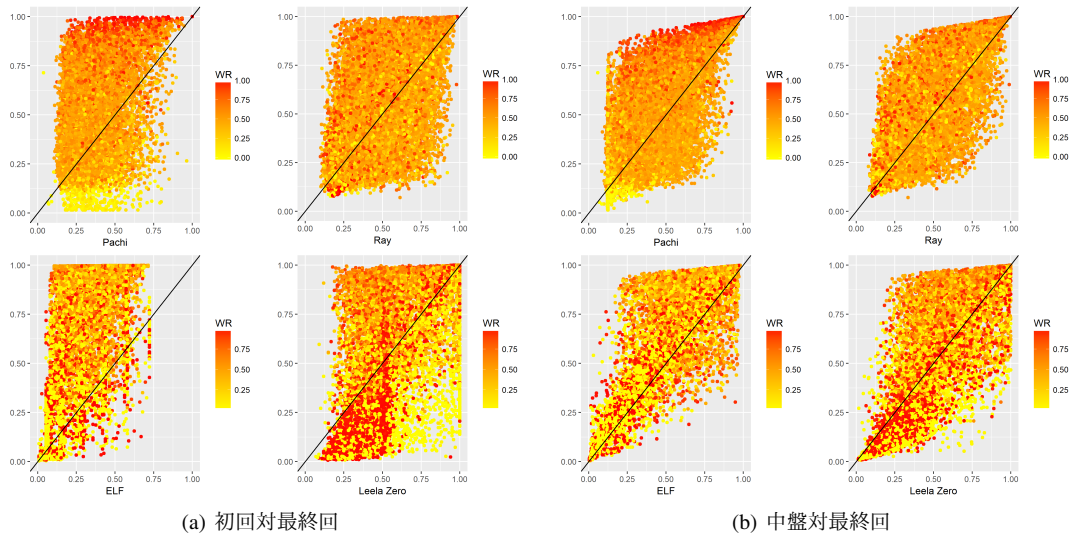


図 4 最善手への試行割合の変化。(左上: Pachy, 右上: Ray, 左下: ELF, 右下: Leela Zero)

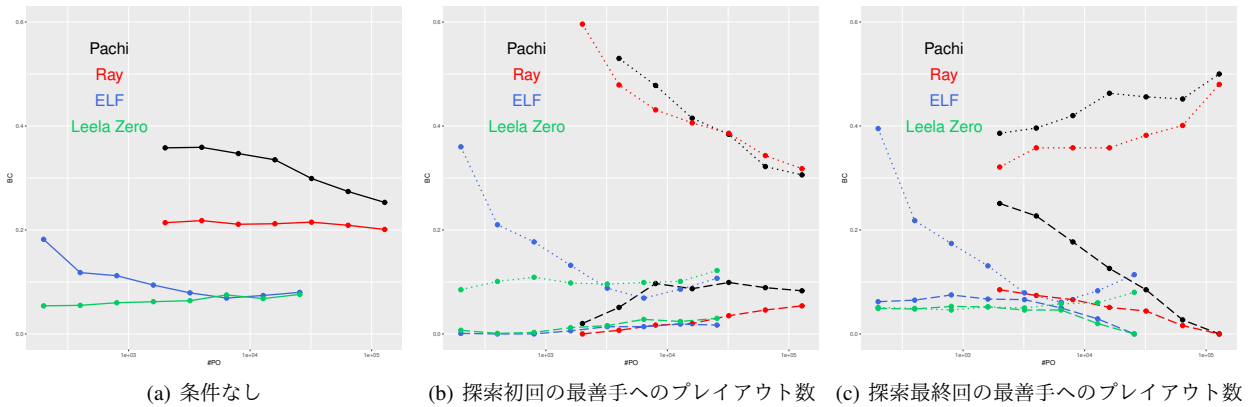


図 5 最善手へのプレイアウト数で条件分けした時の、プレイアウト回数に対する最善手変化率。(a) は条件なし, (b) と (c) では、探索初回と最終回の、最善手へのプレイアウト数を用いる点異なる。なお、図中の実線、破線、点線は、全データでの結果、最善手へのプレイアウト回数が全体に対して 0.75 以上のデータ、0.25 以下のデータに対応する。

参考文献

[1] C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, march 2012.

[2] M. Guid and I. Bratko. Factors affecting diminishing returns for searching deeper. *ICGA Journal*, 30:75–84, 06 2007.

[3] M. Guid and I. Bratko. Influence of search depth on position evaluation. In *Advances in Computer Games*, pp. 115–126. Springer, 2017.

[4] E. Heinz. DARKTHOUGHT Goes Deep. *ICCA Journal*, 21(4):228–229, December 1998.

[5] Y. Sato, M. Miwa, S. Takeuchi, and D. Takahashi. Optimizing objective function parameters for strength in computer game-playing. In *Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-2013)*, pp. 869–875, 2013.

[6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 01 2016.

[7] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[8] J. R. Steenhuisen. New results in deep-search behaviour. *ICGA Journal*, 28(4):203–213, 2005.