

探索アルゴリズムに適した時間利用に関する研究

芝 世式^{†1}

概要：2020年11月に行われた世界将棋 AI 電竜戦において深層学習とモンテカルロ木探索を組み合わせた所謂 PV-MCTS 系のプログラムが優勝し、将棋の探索エンジンとしては同系のものが今後主流となると思われていた。しかしながら、2021年5月に行われた第31回世界コンピュータ将棋選手権においては決勝8チーム中1チームのみが同系エンジンであった。本研究はこの事象を持ち時間の観点から説明を試みる前段階として探索アルゴリズムの時間利用の適性を評価しようとするものである。

キーワード：コンピュータ将棋，強化学習，深層学習

1. はじめに

ゲームアルゴリズムや探索問題の研究題材としてチェスや囲碁，将棋などの題材が長年取り組まれているが，近年は人間の最上位者を上回る実力を身につけておりトップ選手のみならず多くのプレーヤーが AI を手本に学ぶようになってきている。

囲碁においてはモンテカルロ木探索（以下 MCTS^[1]）が登場して以来現在まで主流のアルゴリズムとなっている。その過程において古くはパターンマッチによる確率モデルなどから最新では深層学習のモデルが方策情報として使われてきたが主流となる基本的な探索アルゴリズムには変わりがなかった。

チェスにおいてはチェス AI 登場以来長くに渡り $\alpha\beta$ 探索型のエンジンがコンペにおいて優位とされてきた。2017年末 AlphaZero^[2]が登場しそれまでのチャンピオンとされた Stockfish^[3]に対しある条件下では MCTS アルゴリズムで勝ち越すことが示された。その追試とされた Leela Chess Zero^[4]プロジェクトが2019年の TCEC'15^[5]および2020年の TCEC'17において Stockfish に勝ち優勝することで実際にその実力を示した。しかしながら，その前後数回に渡っては Stockfish が優勝を誇っている。特に Stockfish 側の最新の更新では NNUE というコンピュータ将棋由来の浅いニューラルネットによる評価関数を導入することで劇的な向上が見られており Leela Chess Zero を大きく突き放す形となっている。それ故，チェスにおいては $\alpha\beta$ 探索に比べ MCTS が優位であるとは言えない状況である。

将棋においてもチェスと同様に $\alpha\beta$ 探索型のエンジンが長らく優位とされてきた。AlphaZero が登場したがソースやモデルなどの公開が無く，棋譜の公開も一部となったため MCTS が優位となる条件や対戦時の設定の公平性などについては疑問が残ったままであった。保木，山下，小林の3名の主導による AobaZero^[6]プロジェクトにより AlphaZero 将棋の追試が試みられたが，2020年7月に開催された電竜

戦予行演習2において上位の $\alpha\beta$ 探索エンジンを破り3位に入るなどの健闘を見せた。またレーティング計測により AlphaZero に準ずる程度まで強化学習が進んだと確認された。しかしながら，コンペにおいて $\alpha\beta$ 探索のエンジンを全て押さえて勝利するには至っていない。

また，加えてコメントすると AlphaZero の比較対象は2017年の elmo であり，同時期に設計された NNUE 型の評価関数モデルとの対戦は記録されていない。

2020年11月に開催された第一回電竜戦^[8]において MCTS 型の将棋エンジンがはじめてコンペで優勝するに至る。優勝した GCT では AlphaZero のモデルに多くの改良を加えている dlshogi^[7]をベースにしている。大会後公開された情報により多くの開発者がその強さの確認をおこない，今後のコンペにおいては MCTS 型のエンジンが優位ではないかとの意見が強かった。

しかしながら，2021年5月開催の第31回世界コンピュータ将棋選手権^[9]においては決勝進出8チームのうち7チームが $\alpha\beta$ 探索を用いており純粋な MCTS 型エンジンは PAL のみであった。その PAL は決勝で2位となり，優勝は $\alpha\beta$ 探索型の elmo であった。これにより探索アルゴリズムの優位は感嘆には語り難い状況にある。

本テーマはこのアルゴリズムの優位性を議論するパラメータとして思考時間を挙げ，問題を整理しようとするものである。

2. 持ち時間

前述の電竜戦および世界選手権では持ち時間が異なることが知られており，前述の結果について持ち時間の差が勝敗の主因となっているとの議論が上がった。

電竜戦の持ち時間は各10分に加え，一手2秒加算のフィッシャールールである。比較して世界選手権は持ち時間15分，一手5秒加算のフィッシャールールである。凡そ2倍程度の差があると考えてよい。

本議論に加えて floodgate^[10]のデータも参考に加える。Floodgate はインターネット上で誰でも自由に対戦できる対戦サーバである。世界選手権後の複数のチームが対戦を

^{†1} 岡山県立大学 情報工学部
Okayama Prefectural University
shiba@cse.oka-pu.ac.jp

行い、成績を残している。なかでも GCT および Daigorilla が選手権本番さながらのハイエンドインスタンスを用いた対戦を行っており、4400 台という高レートで拮抗していることを確認した。GCT は二次予選で 14 位であったが前述の電竜戦モデルより遥かに強化されたモデルを投入したとアピールされており MCTS 型としてはトップクラスであることは言うまでもない。二次予選 11 位となった私の二番紋りも同様である。Daigorilla も二次予選 2 位通過であり決勝では 7 位となったがトップクラスの実力であることは間違いない。これらが floodgate 上でほぼ拮抗したレーティングを示している。持ち時間としては各 5 分に加え、一手 10 秒加算のフィッシャールールである。単純比較は難しいが、例えば選手権決勝の平均手数である 180 手程度を想定した場合は各エンジンが思考するために使える時間が世界選手権 1350 秒、floodgate1200 秒、電竜戦 780 秒となる。つまり floodgate が中間的な値となる。

ここでひとつ仮説を立てる。持ち時間が短い場合は MCTS 型が有利であり、持ち時間が長い場合は $\alpha\beta$ 型が有利となる。Floodgate 程度であれば現行のクラウドで入手可能なハイエンドインスタンスで拮抗するというものである。

この検証は非常にコストがかかる。というのもハイエンドインスタンス自体が高コストである根本的な理由と共に各エンジンは同じ局面ではほぼ同じ手を指すことが考えられており長時間戦を多数回行うことは無駄が多い。選手権決勝の結果をみなくとも相性問題が勝敗を決する傾向にある。

本仮説を検証するためには $\alpha\beta$ 型および MCTS 型でそれぞれ適当数を持ち寄るリーグ戦などが好ましいように考える。

3. 低ノード数での比較

前述のように高レートでの比較にはコストがかかるために簡単に結論をだすことは難しい。そこで各エンジンにおいてダウンスケールした計測をおこなった。^[11]

これは別の興味もその動機のひとつである。それは人間と同程度の探索速度でもコンピュータは強いのかというテーマである。将棋の高段者であれば一秒間に数十局面の探索は可能であるとされており 11 手詰め程度の詰将棋は数秒で解かれる。

本実験は以下のように行われた。探索エンジンに dlshogi 互換の YaneuraOu モジュール (通称ふかうら王) の Intel MKL バージョン、評価モデルに GCT の世界選手権版および二番紋りの世界選手権版である。設定としては 1 スレッド 1 バッチサイズ、1 CPU および 4CPU 評価とした。所謂 MCTS の並列化アルゴリズム等の影響は受けけない上、計算機リソースの点からも平等と考えやすい。計測は floodgate にて 100 戦以上行いレート計測とした。

当初 4CPU 評価の GCT を投入した。一般的な 4 コア PC で 100nps 程度の速度であるが、これでレート 3452 と

floodgate レート基準の gikou2_1c を上回った。簡単に想定していた以上に前バージョンより強化されていると感じた。ちなみにこの gikou2_1c は Ryzen の 1 スレッドでおおよそ 700knps の探索速度によるエンジンである。つまり、探索速度にして 7000 分の一で勝ち越していることになる。

このため以後 1CPU で行った。GCT は 50nps 程度、二番紋りは 11nps 程度であることを確認した。モデルサイズは GCT が ResNet10 ブロック 192 チャンネル、二番紋りは ResNet40 ブロック 256 チャンネルであるから妥当な速度差と言える。

表 1 floodgate レートの一部

エンジン	レート	10	15	0.510	2021-05-07	N/A
dlshogi_wssc31_1t	3339	72	76	0.486	on line	N/A
gikou2_1c	3300	244	333	0.423	on line	N/A
Krist_483_473stb_1000k	3276	300	333	0.474	on line	N/A
nibanshibori_wssc31_1t	3275	53	57	0.482	2021-05-14	N/A
AobaZero_w3493_n_p800	3205	74	103	0.419	on line	N/A
AobaZero_w3483_n_p800	3193	88	134	0.397	2021-05-11	N/A

表 1 に計測結果を示す。GCT がレート 3339、二番紋りが 3275 となっている。レート基準の gikou2_1c および Kristallweizen の 1000k ノード制限および AobaZero の 800 プレイアウト制限などが近いレートである。

GCT は gikou2_1c をやや上回る結果となった。これは両者 CPU のシングルスレッドであるから純粋なエンジン比較として GCT の勝利と言って良い。ただし、gikou2 が上位と言えたのは 2016 年である。レートの根拠である対戦を個別にみると gikou2 に勝ち越しているが 1000k ノード制限の Kristallweizen には負け越していることを確認した。

二番紋りは gikou2 や Kristallweizen の両者ともに下回ることがレートとしては拮抗している。1 秒当たり 11 局面程度の評価でこのレートは驚異的と言える。gikou2 と比較すれば 6 万分の一以下の探索速度であり、一手 10 秒程度考えているとすれば深層学習モデルで評価されたのは僅か 110 局面である。

4. ノード比について

前節で低ノード数での比較を行った。この比較が高ノードでも可能かどうかの議論を進める。

最新モデルではないが以前 AobaZero を用いてプレイアウト数 200, 400, 800 の比較を行い探索数を倍にした際のレート変化を求めておりこれを 100~120 としている。^[12] 計測精度が低い二番紋りを 100nps 程度で floodgate で計測した際に 3600 程度と出たため概ね最新モデルでも有用に思われる。また、同時期確認した elmo および Kristallweizen でも 120 や 130 程度の数字が得られており、 $\alpha\beta$ 探索型でも MCTS 型でも大差ないように見受けられる。これはアルゴリズムが異なってもスケールの傾向が同じと言う意味で非常に興味深い現象である。

では、高ノードにした場合の比較をしてみよう。αβ型エンジンでは最高性能を誇るハードウェアとして現在 Ryzen Threadripper 3990X が挙げられるがこれは 64CPU128 スレッドを擁するものである。ざっくり評価で2の7乗スレッドである。

比べてMCTS型エンジンでは最高性能を誇るハードウェアは NVIDIA の A100 である。これを8枚搭載したインスタンスを用いてGCTは350knpsの探索速度が出るとしている。比べて二番絞りは100knpsであった。GCT側は高性能なハードウェアを十分に生かし切れていないが二番絞りは80%程度の負荷を確認しており、16枚のGPUにおいて200knpsの探索速度を動作確認している。これらは前述の1スレッドに対してGCTで7000倍、二番絞りで9000倍および18000倍となる。ざっくり2の13乗弱、13乗強、14乗強と求められる。

これは現行ハードウェアではMCTS型の方が有利となる。当然ながら消費電力や運用コストは遥かに大きいが無制限ルールにおいては勝利に近いと言える。

しかしながら、スケール効果が必ずしも得られるとは限らない。幾つかの懸念材料があり、それらの幾つかが今回の世界選手権で顕在化したものと考えられる。

ひとつ目はハードウェアの性能を生かし切っていない点である。GCTは上記で350knpsと示したが局面によっては三分の一も出ないことがあると計測されており、二番絞りで半分近くまで速度が落ちるケースがあった。これはMCTS型の並列探索性能によるものでGPUに同時に評価させる局面を与えられていないことによる。これには並列化効率を向上させるアルゴリズム的な改善が不可欠である。

ふたつ目はノード数拡大による弊害である。世界選手権においてやねうら王チームで発生したのは探索ノード数拡大によりメモリ溢れであった。同様のことが全てのエンジンで起こりえる。加えて演算誤差である。MCTS型エンジンでは局面評価を遡って反映させるが、この際評価局面数が増大するにつれて局面の評価値の誤差が増大する。この影響は少なからず存在することは既に確認されている。

しかしながら、二番絞りはもっとも大きなモデルを利用したため探索ノード数が最も少ないMCTS型エンジンであるためこの懸念からは最も遠いと考えられる。

二番絞りにおいては以上のようにハイパフォーマンス時の性能を目標に準備を行ってきたが選手権では十分な戦果を得ることができなかった。

5. おわりに

将棋の探索エンジン開発において現在αβ型のエンジンとMCTS型エンジンの実力が拮抗している。

持ち時間の長さによって差異が生じる可能性はあるがそれは僅かであると推察される。

MCTS型はまだ開発歴が浅く改善のポイントが複数残されており強化する余地がある。

二番絞りのような大型のモデルがハイパフォーマンスレンジにおいては有効と考えている。

参考文献

- [1] Rémi Coulom. “Efficient selectivity and backup operators in monte-carlo tree search”. In International conference on computers and games, pp. 72–83. Springer, 2006.
- [2] David Silver et al. “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm”, <https://arxiv.org/abs/1712.01815>
- [3] Stockfish, <https://stockfishchess.org/>
- [4] LCZero, <https://lczero.org/>
- [5] TESC, <https://tcec-chess.com/>
- [6] AobaZero プロジェクト, <http://www.yss-aya.com/aobazero/>
- [7] dlshogi, <https://github.com/TadaoYamaoka/DeepLearningShogi>
- [8] 竜戦, <https://denryu-sen.jp/>
- [9] 世界コンピュータ将棋選手権, <http://www2.computer-shogi.org/wcsc31/>
- [10] Floodgate, <http://wdoor.c.u-tokyo.ac.jp/shogi/floodgate.html>
- [11] 芝, “非 GPU 勢 DL 組”, <https://bleu48.hatenablog.com/entry/2021/05/12/075514>
- [12] 芝, “AobaZero で遊ぼう 11 (レーティング集計プレイアウト数について)”, <https://bleu48.hatenablog.com/entry/2020/03/21/144346>