

最善手の予測に基づくゲーム木探索の分散並列実行

金子 知適^{1,a)} 田中 哲朗^{2,b)}

受付日 2012年2月20日, 採録日 2012年9月10日

概要: 本稿では, 主記憶を共有しない多数の計算機を用いて, ゲーム木探索を並列に行う新たな手法を提案する. 提案する枠組みでは, 各節点における最善手や次善手を予想しながらマスタのゲーム木を成長させ, 最終的に葉にスレーブが1台ずつ割り当てられる. 各スレーブは, 割り当てられた葉を独立に探索する. この手法ではマスタやスレーブ間の通信はほぼ最小限に抑えられており, 通信遅延の大きな環境でも頑健に動くこと, また元のプログラムの変更をほとんど必要としないことが利点である. 多数の計算機を用いることによる強さの向上の度合いについて, オープンソースプログラムのGPS将棋を用いた実験を行い, 8スレーブの分散探索では4並列のメモリ共有探索に近い強さを確認した.

キーワード: ゲーム木探索, 分散処理, 将棋

Distributed Game-tree Search Based on Prediction of Best Moves

TOMOYUKI KANEKO^{1,a)} TETSURO TANAKA^{2,b)}

Received: February 20, 2012, Accepted: September 10, 2012

Abstract: This paper presents a new game-tree search method on distributed systems. In this method, a master tree grows based on the prediction of the best moves at each internal node until when each slave is assigned a leaf to search. Then, all slaves independently conduct a game-tree search. This framework has two major advantages; (1) it can work under an environment with relatively high latency, as it requires little communication between a master and slaves. (2) it works without significant modification of a base search program. In the experiments with an open source shogi program, the presented method worked effectively. The strength of the presented search with 8 slaves was almost comparable to that of a shared memory search with 4 threads.

Keywords: game tree search, distributed computing, Shogi

1. はじめに

本稿では計算機クラスタなど主記憶を共有しない多数の計算機を用いて, ゲーム木探索を並列に行う手法を提案する. 計算資源を有効に活用することで, ゲームプログラムの強さはさらに向上すると期待される. コンピュータ将棋の分野では2010年4月に情報処理学会が日本将棋連盟に

挑戦状を送る*1など, トッププロとの対局に興味が集まっている. 歴史的な対局に最強のコンピュータプログラムで臨むためにも, 計算資源を有効に活用する研究は重要である.

提案手法では, 探索対象局面をルートとするゲーム木を短い時間内に生成してマスタが管理し, そのゲーム木の葉のそれぞれにおいて, 1台のスレーブが時間の許す限り独立に探索を行う. 既存手法と比較して, マスタやスレーブ間の通信が非常に少ないため, 通信遅延の大きな環境でも頑健に動くことと, 元のプログラムの変更をほとんど必要としないという利点の実現している. また, マスタが保持するゲーム木を作成する際には, 各節点で有力と予想され

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, Meguro, Tokyo 153-8902, Japan

² 東京大学情報基盤センター
Information Technology Center, The University of Tokyo,
Bunkyo, Tokyo 113-8658, Japan

a) kaneko@graco.c.u-tokyo.ac.jp

b) ktanaka@mail.ecc.u-tokyo.ac.jp

*1 <http://www.ipsj.or.jp/50anv/shogi/index2.html>

る指手のみを展開することで、重要な部分木に計算資源を効果的に投入することを目指している。

実際に将棋を対象に実装を行い、手法を評価し、(1) マスタのゲーム木の生成において、正解手（長時間の逐次探索で選ばれる手）を成長させる精度、(2) メモリ共有並列探索との対戦成績を通じた強さの2点でそれぞれ有効性を確認した。情報処理学会が日本将棋連盟に挑戦した際のプログラム「あから 2010」においてもこの手法が採用され、その実用性が支持されている。

2. 関連研究

主記憶を共有するマルチコアのコンピュータ上で、複数のコアを活用してゲーム木探索を効率的に行う研究では、PVS (Principal Variation Splitting) [4] をはじめさまざまな手法が提案され、実際にコンピュータ将棋の分野でも広く使われるようになってきている。

主記憶を共有しない疎結合並列計算機におけるゲーム木探索手法にも、YBWC [2], APHID [1], TDSAB [3] など古くからの研究がある。しかし、それらの手法は実装が複雑であり、また局面表を共有するために高速なネットワークを必要とするなどの点で、コンピュータ将棋プログラムと現在の汎用的な環境を対象とした場合の性能は定かでない。将棋における報告は文献 [7], [8] を参照されたい。著者らは、比較的低速なネットワークでも動作し、かつ実装も容易な手法を目指しており [9], 本稿ではその拡張を提案し評価する。

計算資源を活用する他の方法として、複数のプログラムで指手を検討し投票により次の一手を選ぶ手法も研究されている [5]。しかし現在のところ、勝率の向上に関してはゲーム木探索を並列に行う方が効果的であるようである。

3. ゲーム木探索の分散並列実行

提案手法の概要は、探索対象の局面をルートとするゲーム木を短い時間に生成することと、そのゲーム木の葉節点においてそれぞれを1台のスレーブが時間の許す限り探索することの2点である。スレーブの探索開始後、マスタは、スレーブからの報告に基づきゲーム木の各節点の評価値を適宜更新する。図1に簡単な例を示す。木の各節点は、ゲームの1つまたは複数の局面に対応する。葉節点は実線で囲まれており、各葉節点には1台のスレーブが割り当てられ、時間の許す限り探索を行う。葉の評価値は、担当するスレーブが最も最近にマスタに報告した値が用いられる。点線で囲まれた節点の評価値は、子節点のMinMax値により定められる。背景に網掛けを施した一部の葉節点は、複数の局面をまとめて表現した仮想的な節点である。図1の例では、ルート節点のAが初期局面に対応し、この局面の最善手を求めたい。初めにルートで有力な指手を▲7六歩と▲2六歩と予想したのでそれぞれの先（節点B

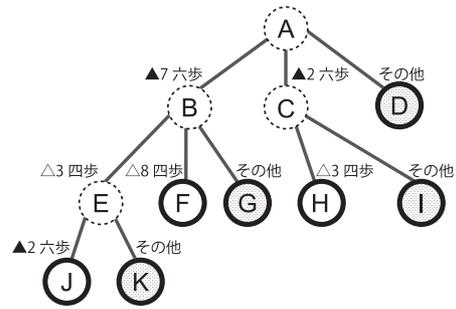


図1 マスタのゲーム木と葉節点へのスレーブの割当て

Fig. 1 A master's game tree where a slave is assigned to each leaf.

とC)を複数のスレーブで深く読むこととし、また残りの指手はまとめて（節点D）スレーブ1台が探索することを表している。

この探索中に必要な通信は、基本的に評価値と最善手が更新されるたびに、あるいは1秒に1回程度、スレーブからマスタに情報を送るだけである。そのため、通信遅延の大きな環境でもある程度頑健に動くことができる。またスレーブの主要な機能は、与えられた局面で探索を行い評価値と最善応手手順 (Principal Variation; PV) を報告することであるため、分散実行を前提としていないプログラムをほとんど変更なしに用いることができるという利点もある。

3.1 ゲーム木の作成と計算資源の割当て

逐次のゲーム木探索手法でも有力な展開を深く読むことが強さにつながることから [6], 分散並列探索においても有力な展開に多くの計算資源を割り当てるのが効果的と期待される。そこでマスタのゲーム木を作成する際に、短時間の探索を行うことで有力な指手を予想する。ゲーム木作成の具体的な手順を、図2のゲーム木成長という再帰の手続きに示す。引数の「節点」は探索対象で、初めの呼び出しではマスタのゲーム木の根が与えられる。また「スレーブのリスト」は、その節点以下の部分木を担当するスレーブの一覧である。まず、手持ちのスレーブSが1つであれば即座に、探索開始手続き（引数はスレーブと節点）により、そのスレーブをその節点に割り当て探索を開始する。また局面で合法手が一手しかなければ、その指手に対応する節点を作成し、あらためてその節点でスレーブの割当てを行う。それ以外の場合は、次の段落で説明するような短い探索を行って、最大n個の有力な指手を得る。それらの指手を担当するn'個のグループと「その他」を担当する1つにSを分割し、各指手に対しては再帰的にゲーム木成長を行う。また「その他」に対応する節点を作成し、1つのスレーブで探索開始する。最終的にはすべてのスレーブが探索開始という手続きにより、1つの節点を割り当てられて探索を始める。

```

ゲーム木成長(節点  $N$ , スレーブのリスト  $S$ ) {
  if ( $|S| = 1$ ) {
    探索開始( $S$  内の唯一のスレーブ,  $N$ )
    return
  }
  合法手生成
  if (合法手が 1 つだけ) {
    子節点作成
    ゲーム木成長(子節点,  $S$ )
    return
  }
   $n' \leftarrow \min(|S|, n)$ 
  合法手全体から有力な順に  $n$  個の指手を得る
  上位  $n$  手を  $m_1, m_2, \dots, m_{n'}$ , それ以外を  $M^*$  とする
  対応する子節点  $c_1, c_2, \dots, c_{n'}$  と  $c^*$  を作成
   $S$  を  $n' + 1$  群  $S_1, S_2, \dots, S_{n'}, S^*$  に分割
  for  $i$  in  $1 \dots n'$ 
    ゲーム木成長( $c_i, S_i$ )
  探索開始( $S^*, c^*$ )
}
    
```

図 2 マスタのゲーム木を成長させる方法
Fig. 2 Growth of a master's game tree.

```

指手推薦(合法手の集合  $M$ , スレーブのリスト  $S$ ) {
  if ( $|M| \leq |S|$ ) {
    各指手を 1 つのスレーブで  $a$  秒探索
  } else {
    合法手をヒューリスティックにより並び替え (*)
    上位  $|S| - 1$  の各指手をスレーブ 1 つで  $a$  秒探索
    残りの指手をまとめてスレーブ 1 つで  $a$  秒探索
  }
   $\min(|M|, |S|)$  個の指手の評価値が求まる
  評価値の良い順に推薦
}
    
```

図 3 固定秒数の探索に基づく指手の並べ替え
Fig. 3 Move ordering based on search results with in a certain seconds.

有力な指手を得るには図 3 に示すような、固定の a 秒の探索を行う。合法手の数 ($|M|$) がスレーブの数 ($|S|$) よりも少ない場合は、それぞれの指手を 1 つのスレーブで探索する。そうでない場合は何らかのヒューリスティックに基づいて指手に仮の順位をつけ (*), 有力と予想される指手から順にスレーブを割り当て探索し、最後の 1 つのスレーブが残りの指手をまとめて探索する。ある指手の評価を請け負ったスレーブはその評価値を返し、その他の指手をまとめて請け負ったスレーブは、その中で最も良かった指手と評価値を返す。総合して、スレーブの数だけ、指手と評価値のペアが求まる。その他の指手をまとめて請け負ったスレーブからは、1 つの指手と評価値しか得られないので、複数の有力な指手がその他に分類されないことが望ましい。そのために、仮の順位付け (*) の精度もある程

度重要となる。本稿では、親節点で固定秒数の探索をしたときの最善応手手順をまず優先し、他は実現確率 [6] の高い順に仮の順位とした。著者以外による応用で、静止探索の評価値を用いた動作報告もある。

3.2 探索の効率とオーバーヘッド

共有メモリ環境で同じ CPU 資源を投入した場合と比較して、提案手法の得失は以下のとおりである。マスタのゲーム木とスレーブの割当てが決まって以降は、提案手法はスレーブ間で通信を行わないため、同期のオーバーヘッドはない。また各スレーブ内では、余計な仕事は発生しないため、ベースプログラムの探索と同じく最大限の効率が出る。提案手法でスレーブ数を 100, 1,000 と増やす場合を考えると、この 2 点がスケラビリティの障害にならないことは大きな利点である。

一方、 $\alpha\beta$ ウィンドウや局面表をスレーブ間で共有しないため、分散探索全体では探索のオーバーヘッドが生ずる。 $\alpha\beta$ 探索が最もうまく働いた場合と比較すると、投入したスレーブの計算資源の平方根に相当する強さの実現が 1 つの目標となる。また、ゲーム木が決まるまでの間は固定秒数の探索を行うため、スレーブ数の \log 程度の時間も消費する。

具体的に本稿の実験では分割の幅 n を 2 としたので、(合法手がつねに 2 より大きければ) マスタの作るゲーム木は二分木に「その他の指手」のための節点加わったものとなる。したがって、 $2^{D+1} - 1$ の数のスレーブを用意すると深さ D の木の葉のそれぞれにスレーブを配置できる。この場合に、スレーブ 1 つのみで探索する場合と比較すると、幸運な場合で D 手深いところから探索を始めた場合と同様の効果がある。これは十分なアドバンテージである。不運な場合はルート節点でその他の指手の評価が最善となった場合で、このときの利点は、有力な 2 手を探索しないで済むことしかない。

3.3 プロトコルと実装

スレーブが行う通信は、局面と探索条件を受け取って最善手や評価値を報告することであるので、さまざまな実現方法が考えられる。今回は汎用性を重視して、USI*2 という、GUI と思考エンジンのやりとりを定めたプロトコルを拡張して用いた。局面の表記や探索の途中経過の報告は USI に準じた。拡張した部分には、実現確率とともに合法手を作成する指示 (`genmove_probability`) や、指定の指手を無視する探索条件の表記 (`ignore_moves`) などがある。

マスタ部分は perl を用いて実装し、分量は 2,500 行程度

*2 <http://www.glauringchess.com/shogi/usi.html>,
<http://www.geocities.jp/shogidokoro/usi.html>

であった*3。図2や図3は再帰的な形で表現したが、実際にはイベントドリブンに記述する必要がある。また、ルートに近い節点からゲーム木が成長するためスレーブの探索開始には時間差がある。つまり、有力な手の推薦のための探索結果の報告と、メインの探索の評価値の報告が混ざることにも起こる。

スレーブはssh経由で起動し、標準入出力を通じてマスタと通信する。通信においては、最善手の報告 (bestmove) とマスタからの探索終了の命令 (stop) の競合に注意が必要である。USIの文法ではstopの対象の探索は明示されないため、探索開始命令 (go) と最善手報告の回数を数えておき、前者が後者より多い場合のみ終了命令に反応させるなどの解決策が必要となる。他に、特定の一手以外は負けという局面も注意が必要である。通常の対局プログラムであれば、その一手を即座に指すことで思考時間を節約するが、分散探索のスレーブとして用いる場合は探索を続けて評価値を更新し続ける必要がある。

4. GPS将棋を用いた実験結果

GPS将棋を用いて提案手法を評価した結果を報告する。実験で用いたハードウェアは、以降断りがなければ、CPUがOpteron 2376で8つのコアを持ち、オペレーティングシステムが64ビットのLinux (Debian) である。GPS将棋はrevision 2445, OSL部分はrevision 4219のソースコードを用いた。

4.1 重点的に探索すべき指手の予想の精度

提案手法が用いる、スレーブの役割分担を司るゲーム木について、その良さを実験的に評価する。以降の実験では、有力な候補手の選択 (図3) には1秒の探索結果を使い、さらに上位2手に集中的に計算資源を割り当てた。そこで、各節点で読むべき二手が、1秒の探索で正しく推薦されているかを調べたい。どの指手が正解であるといいきることは難しいが、おおむね一致するはずの指手として、次の一手問題集「ラクラク次の一手」[10], [11]の正解の指手と、トップレベルのプログラムどうしの対局で指された指手*4との比較を行った。

具体的には、指手を有力と思われる順に並べた場合に、正解の指手が何番目に登場するかを調べた。指手に順序をつける指標には、提案手法の1秒の探索の評価値 (図3) と、比較対象として指手の実現確率 [6] の両者を用いた。図4に問題集を分析した結果を掲載する。横軸が指手の推薦順位で、縦軸がその順位の指手が正解であった頻度を%で表

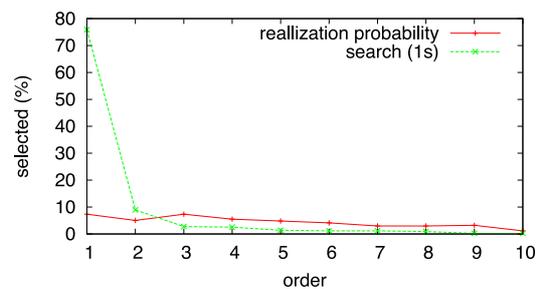


図4 「ラクラク次の一手」問題集の正解手の予測
Fig. 4 Rate of agreement in “rakuraku” problem sets.

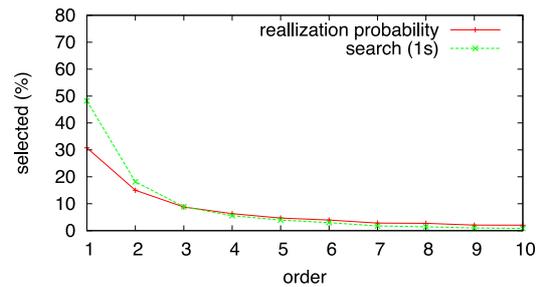


図5 floodgateの棋譜一手の指手予測 (GPS将棋以外の指手)
Fig. 5 Rate of agreement in floodgate game records (w/o GPSShogi).

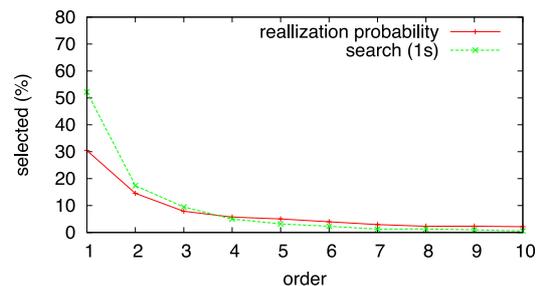


図6 floodgateの棋譜一手の指手予測 (GPS将棋の指手)
Fig. 6 Rate of agreement in floodgate game records played by GPSShogi.

記している。第1手が正解であった頻度は、1秒の探索で整列した場合は70%を超えたのに対し、実現確率で整列した場合は10%未満にとどまり大幅な差がついた。この結果には、問題集では意外な手が正解になる局面が集まっているという傾向が影響している可能性がある。

図5はプログラムどうしの対局のうちGPS将棋以外のプログラムが指した手の分析、図6は同じくGPS将棋が指した手の分析である。GPS将棋と他のプログラムの指手では、GPS将棋の指手の方が若干予想しやすいという自然な結果になっている。両方のグラフに共通する性質として、問題集での結果よりも1秒の探索と実現確率の差が縮まっていることが読み取れる。また、順位1と2の指手で7割程度を占める一方で、8手目以降もなかなか0になることはない。

図7は、1秒の探索を2秒、4秒と増やした場合の結果をまとめたものである。棋譜はGPS将棋以外のプログラ

*3 <http://gps.tanaka.ecc.u-tokyo.ac.jp/cgi-bin/viewvc.cgi/trunk/gpsshogi/sample/perl-cluster/usi.pl?root=gpsshogi&view=log>

*4 http://wdoor.c.u-tokyo.ac.jp/shogi/view/latest-table.cgi?filter=floodgate-10800-60&show_self_play=1 (持ち時間3時間, 秒読み1分)

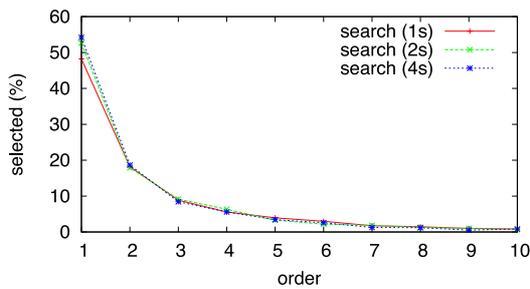


図 7 予測探索の秒数を変化させた場合の比較 (floodgate の GPS 将棋以外の指手)

Fig. 7 Rate of agreement by search within various time periods (floodgate game records, w/o GPSShogi).

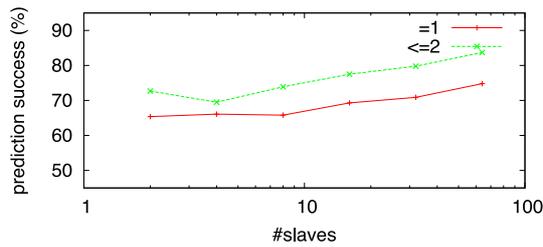


図 8 予測探索に参加する slave の数と予測成功率 (ラクラク)

Fig. 8 Rate of agreement for "rakuraku" problem sets, when moves were searched in parallel.

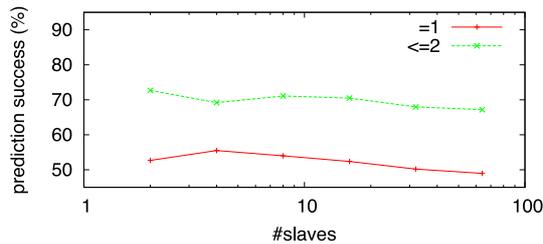


図 9 予測探索に参加する slave の数と予測成功率 (floodgate, 1s)

Fig. 9 Rate of agreement for floodgate game records, when moves were searched in parallel (1s).

ムのものを用いた。グラフから、第 1 手目の正解率が数ポイント向上するもののグラフの概形には変化がないことが読み取れる。この数ポイントがどの程度強さに影響しうるのは現状では分からないが、仮に効果を持つ場合でも予想のための探索の秒数を増やすことはゲーム木の確定を遅らせるというデメリットを持つため、トレードオフを慎重に検討する必要がある。

続いて、1 秒の探索に参加するスレーブの数の影響を調査した。これまでのこの節の実験では数百のスレーブがある状況を想定して、それぞれの手を独立に 1 秒の探索を行った結果である。一方、現実には合法手の数が利用可能なスレーブの数を超える場合があり、その場合に残りの指手は「その他の手」とまとめられて 1 つのスレーブで探索される。そのような状況を想定して、スレーブの数を 2, 4, 8, 16, 32, 64 と変化させて同様の実験を行った。

図 8 はラクラク次の一手に対する結果である。横軸がス

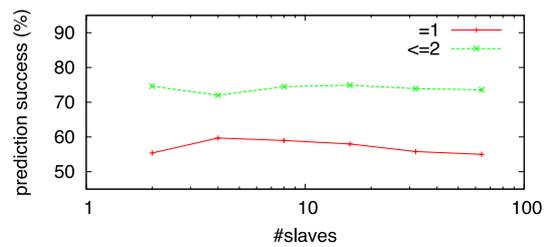


図 10 予測探索に参加する slave の数と予測成功率 (floodgate, 4s)

Fig. 10 Rate of agreement for floodgate game records, when moves were searched in parallel (4s).

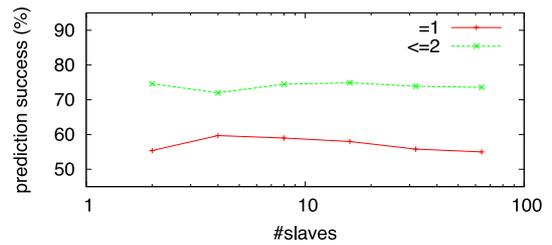


図 11 予測探索に参加する slave の数と予測成功率 (floodgate, GPS 将棋の指手)

Fig. 11 Rate of agreement for floodgate game records played by GPSShogi, when moves were searched in parallel.

レーブの数、縦軸が予測に成功した頻度を%で表す。第 1 手目の予想が正解手だった場合と、第 1 手または 2 手目の予測に正解手が含まれていた場合の 2 通りを示した。どちらの場合も、右肩上がりのグラフになっていて、たくさんのスレーブが参加するほど予測が正確になるという傾向を示している。

一方、コンピュータ将棋の対局の棋譜を用いると図 9 のように若干の右肩下がりとなり、参加するスレーブが増えるほど予測が乱れるという、問題集の場合とは異なる傾向となった。4 秒考えさせた場合の図 10 も、GPS 将棋が指した指手を予想させた場合の図 11 も傾向はあまり変わらない。問題集と異なり、近い評価値の指手が多数あるような局面では、スレーブの数が最良の指手の予想に影響するようである。

4.2 メモリ共有並列探索に対する強さの比較

さらに、提案した手法の総合的な性能を評価するために、メモリを共有する並列探索 (PVS) を行うプログラムとの対局を行った。提案手法を用いるプログラムは、各 1 コアずつを用いるスレーブを 8 体参加させた。一方、対局相手であるメモリを共有する並列探索を行うプログラムは、それぞれ 1 (=逐次)、2, 4 コアを使う 3 種類を試した。思考時間は 1 手 15 秒とし、定跡や手番は乱数で決めた。実験を単純に保つため、相手が思考中の時間を用いた予測読みは行わなかった。評価関数など、探索以外の条件はすべて同じである。

表 1 に勝敗を掲載する 8 スレーブを用いた提案手法のプ

表 1 勝敗表 (勝-負-引分)

Table 1 Self-play results (win-loss-draw).

提案手法の並列度	メモリ共有並列		
	1	2	4
8	71-28-1	57-29-2	43-50-1

GPS 将棋用クラスタ :

- Intel X5570 2.93GHz, 8 cores 1 台
- Intel X5470 3.33GHz, 8 cores 1 台
- Intel X5365 3.00GHz, 8 cores 1 台
- AMD Opteron 2376 2.3GHz, 8 cores 4 台
- AMD Opteron 280 2.4GHz, 4 cores 1 台
- Intel Core2 Quad Q6700 2.66GHz, 4 cores 1 台

InTrigger クラスタ (科学研究費補助金「特定領域研究」情報爆発時代に向けた新しい IT 基盤技術の研究):

- AMD Opteron 2380 2.4GHz, 8 cores 2 台
- Intel E5410 2.33GHz, 8 cores 4 台

東京大学情報理工学系研究科クラスタ :

- Intel Xeon 2.80GHz, 2 cores 33 台

図 12 クラスタ版 GPS 将棋が利用した機器

Fig. 12 Hardware used by GPSShogi.

プログラムは、逐次のプログラムと 2 コアのメモリ共有並列探索にそれぞれ 7 割 2 分と 6 割 5 分の勝率で勝ち越し、計算資源を投入するほど強くなることを示した。また、4 コアのメモリ共有並列探索とは若干負けが先行するものの互角に近い成績である。このことから提案手法で 8 スレーブの探索を行った強さは、メモリを共有して 4 並列の探索を行う場合に近い強さに相当すると推定される。一方、提案手法で 8 スレーブで探索した場合でも 4 並列のメモリ共有探索に勝ち越せなかったことから、8 並列のメモリ共有並列探索に強さは及ばないと想像される。これは提案手法では、 $\alpha\beta$ ウィンドウや局面表を共有しないため、予想された結果である。

コンピュータ将棋においてクラスタ環境での探索はまだ研究段階であるため、今回限られた条件ながら 8 スレーブのクラスタ環境の探索が逐次の探索よりも実際に強くなったこと、さらにメモリを共有する探索の 4 並列に近い強さまで達したことは大きな意義があると著者らは考えている。

4.3 あから 2010 の探索記録の分析

平成 22 年 10 月 11 日に清水市代女流王将と「あから 2010」の対局が行われ、86 手で清水女流王将が投了、あからの勝ちとなっている。あから 2010 に参加した各プログラムのクラスタの探索においては、本稿で提案した手法を実装したフレームワークが共通して用いられた。最後に、このときの GPS 将棋の記録を簡単に分析して紹介したい。

まず図 12 に、クラスタ版 GPS 将棋が利用した機器を掲載する。個々の機器の能力には差があり、速い機器は 1 秒

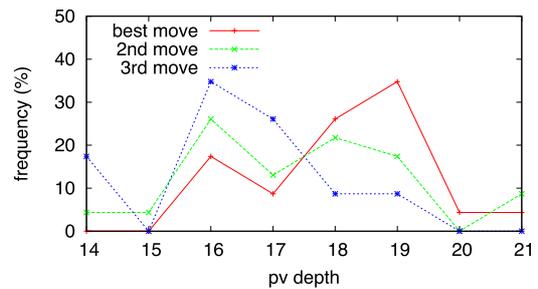


図 13 GPS 将棋のクラスタ版プログラムの PV の長さの分布
Fig. 13 Distribution of the length of PV reported by GPSShogi on PC clusters.

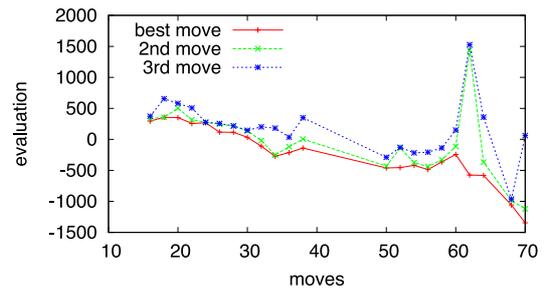


図 14 GPS 将棋のクラスタ版プログラムの評価の推移
Fig. 14 Evaluation by GPSShogi on PC clusters.

間の探索節点数が 100 万を超える一方で、遅い機器は 14 万程度にとどまっている。そこで、なるべく速い機器をゲーム木のルート近くに、遅い機器は先端に用いることが性能を出すために重要である。GPS 将棋の場合は、一手読みを深くすると 2 倍から 2.5 倍程度の時間がかかる。つまり、一手深いところに割り当てる機器は 2 倍程度遅くても実用に足ると期待される。そこでルート (のその他の手、以下同様) と深さ 1 の 2 つの節点にはそれぞれ 8 コアの機器をメモリ共有並列探索によりフルに使うプロセスを割り当て、深さ 2 以降の節点には 4 コアのプロセスを可能な限り割り当てた (残っている 8 コアの機器にも 4 コアを使用するプロセスを 2 つ割り当てた)。その結果、深さ 3 までのすべての節点と深さ 4 の節点の半分程度に 4 コアを使用するプロセスを割り当てることができた。最も遅い 2 コアの機器が担当する節点をなるべく深く先に伸ばすことで、探索で得られた PV が一定の深さを持つことが期待される。マスタプロセスは Xeon 5365 上で動作させた。50 台以上の機器を接続しても CPU 使用率は 10%程度であった。

コンピュータ側の手番での思考記録を、最善手、次善手、3 番目の手について分析して以下に示す。候補手が 3 つある理由は、本稿の提案手法では、ルートで深く掘り下げた 2 つの手と「その他の手」について評価値と PV が求まるためである。

まず図 13 に、GPS 将棋のクラスタ版プログラムが候補としてあげた指手の PV の長さのヒストグラムを掲載する。評価値が最も良い第 1 の候補手 (赤) が PV が長く、3 番目の候補手 (青) が短い傾向がある。これは、3 番目の候

表 2 GPS 将棋のクラスタ版の思考

Table 2 Best, second best and third best moves reported by GPSShogi on PC clusters.

手数 棋譜	候補手 1		候補手 2		候補手 3	
	評価	深さ	評価	深さ	評価	深さ
16 △ 2 二飛	295	18 △ 2 二飛	335	18 △ 4 四角	374	17 △ 9 二香
18 △ 7 二銀	354	19 △ 4 二銀	354	18 △ 9 二香	655	17 △ 7 二金
20 △ 4 二銀	351	18 △ 4 二銀	503	21 △ 5 二金左	580	19 △ 6 四歩
22 △ 5 四歩	256	18 △ 5 二金左	317	18 △ 5 四歩	507	18 △ 7 四歩
24 △ 5 三銀	266	18 △ 2 一飛	273	18 △ 9 四歩	275	18 △ 5 三銀
26 △ 4 四角	117	19 △ 6 四銀	245	16 △ 4 四角	256	17 △ 2 一飛
28 △ 同角成	114	19 △ 同角成	216	17 △ 5 二金左	217	16 △ 9 四歩
30 △ 6 四銀	33	19 △ 4 四銀	135	16 △ 6 四銀	145	16 △ 5 二金左
32 △ 4 四角	-109	18 △ 6 五銀	-19	19 △ 4 四角	201	17 △ 5 五歩
34 △ 6 五銀	-273	17 △ 6 五銀	-254	19 △ 5 五歩	181	16 △ 9 四歩
36 △ 同桂	-213	19 △ 同桂	-119	17 △ 7 七角成	35	14 △ 5 三角
38 △ 5 六銀	-140	20 △ 5 六銀	4	16 △ 7 六銀	349	14 △ 5 二飛
50 △ 7 四桂	-461	19 △ 7 四桂	-431	14 △ 6 四歩	-289	16 △ 3 五歩
52 △ 8 五桂	-455	16 △ 8 五桂	-138	15 △ 3 六銀	-130	16 △ 6 六桂
54 △ 7 七桂成	-420	21 △ 7 七桂不成	-375	21 △ 7 七桂成	-217	16 △ 5 五歩
56 △ 6 六桂	-486	16 △ 6 六桂	-438	18 △ 3 六銀	-208	19 △ 6 四歩
58 △ 5 二金打	-364	19 △ 3 六銀	-329	16 △ 5 六銀	-138	17 △ 3 五歩
60 △ 2 二飛	-243	16 △ 2 二飛	-115	17 △ 3 三飛	150	14 △ 3 三金
62 △ 同角	-575	17 △ 同角	1,439	16 △ 6 九金	1,527	14 △ 6 四歩
64 △ 6 九金	-581	18 △ 6 九金	-367	16 △ 3 七角	357	17 △ 3 三角
68 △ 5 六銀	-1,058	19 △ 7 九金	-987	19 △ 5 六銀	-963	16 △ 5 五角
70 △ 5 五角	-1,346	16 △ 5 五角	-1,121	19 △ 4 七銀成	60	16 △ 7 九金

補手は 1 台で読む場合が多いことを考えると妥当である。

図 14 は、同じく 3 つの候補手の評価値を縦軸に、対局の進行を横軸に描いたものである。範囲は、定跡を外れた 16 手目以降、GPS 将棋が形勢が開いたと判定した 70 手までとした。ただし、40 手目からしばらくは、途中で不調になった機器の切り離しに成功するまで読み筋が記録されていない。また、66 手目はソフトウェアの不具合で短い時間で候補手を決定してしまったため、分析から取り除いた。また記録のとり方の問題から、第 2、第 3 の候補手の評価に関しては指手が決められた時点のものではなく、第 1 の候補手の評価が最後に更新された時点での評価である。

全体として、先手が少し指しやすそうという評価の序盤から、後手勝勢の終盤に移している。また、第 1 の候補手の評価と、第 2、第 3 の候補手の評価に近い局面と大きく異なる局面の両方があることも読み取れる。そこで、将来の対局では、第 2、第 3 の候補手や評価値を活用することも有用と期待される。たとえば、第 1 の候補手以外も投票したり、2 つの手で意見が割れたときでも 2 つの手の評価値の差が少ないと全員が考えていれば思考時間を延長しないなどの方法が考えられる。

表 2 に、棋譜で指された指手と GPS 将棋のクラスタ版が選んだ 3 つの候補手の評価値と読みの深さをまとめる。

実際に棋譜で指された指手には 下線 を引いてある。まず、合議で選ばれた棋譜の指手の多くは GPS 将棋の第 1 の候補手であった。それ以外の場合でも第 1 の候補手と近い評価の指手のようである。例外的に、あからが選んだ指手が GPS 将棋のクラスタ版の第 3 候補手にも入らなかった局面としては、18 手目の △ 7 二銀と、58 手目の △ 5 二金打があげられる。前者は GPS 将棋の評価関数が穴熊を好むため選びにくく、また後者は攻め好きな GPS 将棋の棋風とは異なっていたと考えられる。他に特徴的な局面としては 34 手目 △ 6 五銀のところ、激指が推奨していた △ 5 五歩が 20 点差未満の僅差で第 2 候補手にあがっていることがあげられる。第 1 の候補手以外に第 2 候補手も投票が行われていれば結果が変わっていたかもしれない。

5. おわりに

メモリを共有しないクラスタ環境においてゲーム木探索を並列に行う枠組みを提案し、評価した。この枠組みでは、マスタが作成したゲーム木の葉節点をスレーブが独立に探索して評価値をマスタに報告する。ゲーム木の作成においては、各節点で利用可能なスレーブが複数ある限り有力な候補手の節点を作成し、分割したスレーブをそれぞれに割り当て再帰的に作成を続ける。GPS 将棋を用いた実験が

らは各節点の上位2手に集中的に資源を割り当てる単純な仕組みを用いた場合でも強さは向上し、8スレーブの分散探索は4並列のメモリ共有探索に近い強さであった。情報処理学会が日本将棋連盟に挑戦した際のプログラム「あから2010」でもこの手法が採用されており、その記録から、性能が異なる機器を50台程度集めた場合も期待どおり動作したことが読み取れる。

コンピュータ将棋プログラムの分野では、分散環境での並列探索の実証的な研究は始まったばかりであり、今後の発展が期待される [3], [7], [8]。今回の提案手法には (1) 枠組みが単純であり実装にあたって元のプログラムの変更が少ない、(2) 汎用的なネットワーク環境で動作可能であるという利点があるため、他の分散探索手法の研究が将来進んだ場合でも、提案手法は依然有力な選択肢であり続けると期待される。

参考文献

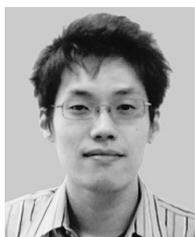
- [1] Brockington, M.G.: Asynchronous Parallel Game-Tree Search, Ph.D. Thesis, Department of Computing Science, University of Alberta (1997).
- [2] Feldmann, R.: Game Tree Search on Massively Parallel Systems (1993).
- [3] Kishimoto, A.: Transposition Table Driven Scheduling for Two-Player Games, M.Sc. Thesis, University of Alberta (2002).
- [4] Marsland, T.A. and Popowich, F.: Parallel Game-Tree Search, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.7, pp.442-452 (1985).
- [5] Obata, T., Sugiyama, T., Hoki, K. and Ito, T.: Consultation Algorithm in Computer Shogi - A Move Decision by Majority, *Computers and Games - 7th International Conference (CG2010)*, LNCS, No.6515, pp.156-165, Springer-Verlag (2011).
- [6] Tsuruoka, Y., Yokoyama, D. and Chikayama, T.: Game-tree Search Algorithm based on Realization Probability, *ICGA Journal*, Vol.25, No.3, pp.145-152 (2002).
- [7] 浦 晃, 横山大作, 近山 隆: 投機を用いた並列ゲーム木探索の効率化, 第15回ゲームプログラミングワークショップ, pp.134-141 (2010).
- [8] 横山大作: 「激指」におけるゲーム木探索並列化手法, *人工知能学会誌*, Vol.26, No.6, pp.648-654 (2011).
- [9] 田中哲朗, 金子知適: 将棋プログラムの大規模並列実行, *情報処理学会研究報告*, Vol.GI-24, No.2, pp.1-8 (2010).
- [10] 日本将棋連盟書籍 (編): ラクラク次の一手 基本手筋集, 日本将棋連盟 (2002).
- [11] 日本将棋連盟書籍 (編): ラクラク次の一手2 基本手筋集, 日本将棋連盟 (2003).



田中 哲朗 (正会員)

1965年生まれ。1987年東京大学工学部計数工学科卒業。1992年同大学大学院博士課程修了。博士(工学)。東京大学工学部助手、東京大学教育用計算機センター助教授を経て、現在は東京大学情報基盤センター准教授。日本ソ

フトウェア科学会, ACM 各会員。



金子 知適 (正会員)

1997年東京大学教養学部卒業。2002年同大学大学院総合文化研究科博士課程修了。博士(学術)。2002年同大学院総合文化研究科助手。2007年助教。2012年准教授。