

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} 田 中 哲 朗^{†2}

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手法は古くから研究されているが、従来の手法はいずれもプログラムの大幅な改変を必要とした。本研究では、元のプログラムの変更をほとんど要しないような単純さを保ちながらも、全体の強さを向上させる分散並列探索の枠組みを提案する。提案する枠組みでは、各節点における最善手を予想しながら再帰的にマスターのゲーム木を成長させ、各葉にスレーブが一つずつ割り当てられるまでゲーム木を成長させた後に残りの探索をスレーブに任せる。GPS 将棋を用いた実験からは上位 2 手に集中的に資源を割り当てる単純な仕組みでも効果的に機能し、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実現した。

Distributed Game-Tree Search Based on Prediction of Best Moves

TOMOYUKI KANEKO^{†1} and TETSURO TANAKA^{†2}

Parallel game-tree search techniques on distributed systems have been intensively researched. However, one has to drastically modify his program to adopt such techniques. This paper presents a simple but effective framework of parallel game-tree search on distributed systems, which requires few modifications on a sequential implementation. In this framework, the master tree grows based on the prediction of the best moves at each internal node until when each slave is assigned a unique leaf in the master tree. Then, each slave independently conducts a game-tree search. In the experiments with GPS-Shogi, the presented search worked effectively even when one simply assigned most computer resources to the top two moves at each node. The strength of the presented search with 8 slaves was almost comparable to that of a shared memory search with 4-threads.

1. はじめに

本稿では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手法を提案する。ゲームプログラムの強さを向上させるうえで、クラスタの計算資源を活用することは有効な手法の一つである。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がいよいよ迫っていると期待されている。このような状況でトッププロに勝つ日を早めるために、従来の開発の延長での強さの向上だけでなく、クラスタ等の計算資源の活用する探索手法が模索されている。

本研究で提案する枠組みは、(1) 各節点の最善手を予想して有力な指し手を分割することで再帰的にマスターのゲーム木を成長させ、(2) 各葉にスレーブが一

つずつ割り当てられるまでゲーム木が成長したら残りの探索をスレーブに任せる。というもので、元のプログラムの変更をほとんど要しない点が特徴である。

単純な枠組みにもかかわらず、GPS 将棋を用いた実験では 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ることが確認できた。また、情報処理学会がプロ棋士に挑戦したプログラムである「あから 2010」に参加したプログラムにおいて、この枠組みがクラスタ上で探索を行う際に採用された。

2. 関連研究

主記憶を共有するマルチコアマシン上でスレッドを複数用いて、ゲーム木探索を効率的に行う手法としては PVS (Principal Variation Splitting)⁴⁾ をはじめさまざまな手法が考案され、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。

メモリを共有しない疎結合並列計算機上のゲーム木探索手法も YBWC²⁾、APHID¹⁾、TDSAB³⁾ など古くから研究されている。しかし、それらの手法はプログラムの大幅な改変を必要とするため、それらの手法が特定の環境で使えるかどうかを評価するためにはコスト

^{†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
kaneko@graco.c.u-tokyo.ac.jp

^{†2} 東京大学情報基盤センター
Information Technology Center, The University of Tokyo
ktanaka@mail.ecc.u-tokyo.ac.jp

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

がかかる。実際、コンピュータ将棋はチェスとは異なる性質の探索空間を持ち、また現在の強いプログラムは、将棋を題材に開発された手法^{5),12)}を含んでいることから、ある時点でコンピュータチェスで有効とされる手法も現在のコンピュータ将棋でどの程度の性能が出るかどうかは試してみないと分からない側面がある。

クラスタ等の計算資源を活用する他の方法としては、プログラムを複数走らせて投票により次の一手を決める手法^{6),7)}なども研究されている。しかし投入する計算資源と比較すると、得られる勝率の向上は、並列のゲーム木探索手法の効果には及ばないようである。

著者らは過去に疎結合並列計算機上のゲーム木探索を行った結果を報告している^{8),9)}。本稿はかつて発表された手法を洗練させたうえで、評価を充実させたものである。手法の具体的な差としては、1秒の探索で有力な指手を得る際に以前は一つのスレーブで行っていた点を、利用可能な全てのスレーブで行うこととした。これは探索時間後すぐに中止を命令された場合にも利用可能な評価値の質を高めるためである。また、再帰的な分割において幅を厳しく制限した。これは、性能に差が大きなハードウェアを混合して使用する場合に、中途半端に分割された探索木から推薦された浅いPVが全体の最善応手手順として選ばれてしまう問題を防ぐためである。ルート付近の「その他の手」の探索に性能の良い機器を割り当てる運用を同時に行うことで、この問題を緩和することができる。さらに、リーフの評価値が更新されるたびに探索木全体の評価値を更新している。これにより、探索中いつでも候補手を返すことができるようになっている。マスターの負担が増えているため、マスターは ruby から perl に書き直し、基本的な処理はスレッド並列で動作する。

3. 探索木の作成と計算資源の割り当て

3.1 アルゴリズム

提案する手法の基本的なアイデアは、探索木を分割してスレーブに割り当てることである。たとえば、ある局面をスレーブ数2で探索する場合には、その局面の最善手と予想される指手を何らかの方法でまず求める。仮に76歩が有力だとすると、一つのスレーブに76歩を指した後の局面を探索させ評価値を求め、もう一つのスレーブに元の局面で76歩は合法手でなかったと仮定して「それ以外の手」の評価値を求めさせる。スレーブ数が3の場合は、最善手と次善手の両方を予想し、それぞれが76歩と26歩だとすると、76歩を指した後の局面、26歩を指した後の局面、「それ以外の手」と分担する。何手までを並

```

探索木割り当て (節点  $N$ , スレーブのリスト  $S$ ) {
  if ( $|S| = 1$ ) {
    節点  $N$  にスレーブを割り当て
    return
  }
  合法手生成
  if (合法手が 1 つだけ) {
    子節点作成
    探索木割り当て (子節点,  $S$ )
    return
  }
  合法手全体から有力な順に  $n$  個の指し手を得る
  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  $(c_1, c_2, \dots, c_n)$  を作成
  スレーブ  $S$  を分割  $(S_1, S_2, \dots, S_n, S^*)$ 
  for  $i$  in  $1 \dots n$ 
    探索木割り当て ( $c_i, S_i$ )
  節点  $N$  の  $M^*$  の手を  $S^*$  に割り当て
}

```

図1 計算資源の割り当て方法

列に探索するかについては柔軟な制御を行う余地があるが、後で述べる実験では2手までに固定してデータを集めた。すなわち、スレーブ数が100の場合は、76歩を指した後の局面に50スレーブ、26歩を指した後の局面に49スレーブ、「それ以外の手」に1スレーブを割り当てる。この場合、76歩や26歩を指した後の局面でも同様の手続きを取り、最善手や次善手にスレーブを割り当てる。

図1に、上記の説明を一般的な形で表現した「探索木割り当て」という再起関数を掲載する。手順中、合法手全体から有力な順に一部を得る部分では、1秒間の探索を行う手法が有力である。その後の行には上位 n 手を分割して探索する手続きが記述されている。その際、合法手全体を M として $n+1 \leq |M|$ を満たすように局面に応じて分割数 n を小さくしないと、計算資源を有効に活用できない。

マスターが作成した探索木のリーフを割り当てられたスレーブは、マスターから停止指令を受け取るまで制限なく探索を続け、途中経過をマスターに定期的に報告する。正確には、最善手や評価値が変わった場合は即座に報告し、他は1秒毎程度に探索した節点数等を報告する。マスターはリーフの情報の更新があるたびに、自身が持つゲーム木のミニマックス値を再帰的に更新する。この更新により、探索開始後は常に現時

```

指手の推薦 (合法手の集合  $M$ , スレーブのリスト  $S$ ) {
  if ( $|M| \leq |S|$ ) {
    それぞれの節点にスレーブを割り当て  $a$  秒探索
  } else {
    合法手を短い時間で有力な順に並び替える
    上位  $|S| - 1$  手を各スレーブ 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
  }
   $|S|$  個の指手の評価値が求まる
  有力な順に並び替えて推薦
}

```

図 2 固定秒数の探索に基づく指手の並び替え

点での最善手を把握することができる。

図 2 に固定の a 秒の探索を行って、有力な指手を列挙する手順を掲載する。指手の数がスレーブの数よりも少ない場合は、各指手を一つのスレーブで探索する。そうでない場合は何らかの方法で指手を並び替え、有力な順にスレーブを割り当て、最後の一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探索結果として評価値を返す。またその他の指し手をまとめて請け負ったスレーブは、探索結果として評価値と最善手を返す。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。

この時、再帰的に分割する幅 n はスレーブの数以下であるため、推薦する指手の数が合法手の数よりも少ないことは問題ない。一方、スレーブの数が合法手の数よりも少ない場合に、正しく上位 n を推薦できない場合もある。そのような状況は、初めの並び替えの精度が悪く複数の有力な指手を最後の一つのスレーブが担当してしまった場合に起こる。

この初めの並び替えの方法としては、親節点で固定秒数の探索をした時の最善手順 (以下 PV) が利用可能であれば、その手を最優先し、他の順序はクライアントに任せる。GPS 将棋の場合は実現確率を用いたが、静止探索の評価値を用いた動作報告もある。

3.2 オーバーヘッド

マスターの探索木とスレーブの割り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。各スレーブ内では、余計な仕事は発生しないため、最大限の効率が出る。一方、 $\alpha\beta$ ウィンドウやトランスポジションテーブルをスレーブ間では共有しないため全体の探索のオーバーヘッドは、通常のメモリ共有並列探索よりも大きい。また、探索木が決まるまでの間には固定秒数の探索を

行うため、スレーブ数の \log 程度の時間を消費する。

分割の幅 n を 2 とした場合には (合法手が常に 2 より大きければ)、マスターの作る探索木は二分木に「その他の探索」のための節点を各内部節点に加えた形となる。 2^{D-1} の数のスレーブを用意するとそのような深さ D の木の葉の全てにそれぞれスレーブを配置できる。スレーブ一つと同等の性能を持つ逐次の探索を行う場合と比較すると、もっとも幸運な場合は $D-1$ 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。もっとも不運な場合はルート節点でその他の手が最善手となった場合で、この時は有力な 2 手を探索しないことで探索が早く終了する程度の利点しか生まれない。この場合でもルートの指手推薦に必要な固定秒数を除けば、逐次の探索と比較した不利益はない。

3.3 プロトコルと実装

上記の枠組みは基本的に局面を渡して最善手や評価値を得ることで動作するため、通信の仕組みには自由度がある。今回は汎用性を重視して GUI と思考エンジンのやりとりを定めた USI^{*1} というプロトコルを基本に、拡張して用いた。局面の表記や探索の途中経過の報告には USI プロトコルをそのまま用いた。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、指定の指手を無視して探索する指示の方法 (ignore_moves) などがある。

マスター部分は perl を用いて実装した。行数は 2500 程度であった。^{*2} リモートのクライアントを ssh で起動し、標準入出力を通じてやりとりするため、クライアントの交換は簡単となっている。

さらに図 1 や図 2 は再帰的な形で記述したが、実際にはイベントドリブンに記述する必要がある。ルートに近い節点から探索木が決まってゆくためスレーブの探索開始には時間差があり、マスターが持つゲーム木のある節点では有力な手の推薦のための探索結果が報告され、別の節点では既に探索が開始されて評価値が報告されるということが起こりうる。

クライアントを対応させるうえで注意が必要な点もいくつか存在する。一つは、クライアントからの最善手の報告 (bestmove) とマスターからの探索終了の命令 (stop) に競合が発生し得ることである。たとえば、探索開始命令 (go) と最善手報告の回数を数えておき、前者が後者を上回っているときのみ終了命令に反応させるなどの解決策がある。探索開始や終了の命令にど

*1 <http://www.glaurungchess.com/shogi/usi.html>
<http://www.geocities.jp/shogidokoro/usi.html>

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

の局面に対する指示ががついていればこの問題は発生しなかったが、USI は並列探索を念頭においたプロトコルではないため、このような考慮はされなかったと想像される。他の重要な注意点は、合法手が複数あるものの特定の一手以外は詰みで負けという局面における振り舞いである。通常の対局のためのプログラムであれば、そのような状況ではその一手を即座に指すことで思考時間を節約することが望ましい。しかし、分散探索においては最善手ではなく評価値が必要とされるため、そのような状況でも探索を続けて深い探索に基づく評価値を求める努力を続ける必要がある。

4. 実装と GPS 将棋における実験と結果

GPS 将棋を用いて提案手法を評価した結果を報告する。以降、断りがなければ実験で用いたハードウェアは CPU に Opteron 2376 を搭載した機器で 8 コアを持つ。オペレーティングシステムは 64 ビットの Linux (Debian) である。GPS 将棋は OSL 部分が revision 4219 相当、GPS 将棋部分が revision 2445 相当のソースコードを用いた。

4.1 最善手の予想

以降の実験ではスレーブの割り当てのための候補手の並び替えには、1 秒の探索を利用し、さらに上位 2 手に集中的に計算資源を割り当てる方法を取った。そこでまずは、1 秒の探索に基づく最善手の予想の正確さを評価した。計算資源を投入すべき有力な二手を 1 秒の探索で正しく推薦しているかどうかが強さに大きく影響するためである。

深く読むべき指手とは何かについて十分な定義を与えることは難しいが、深く読むべき手と概ね一致するはずの指し手として、次の一手問題集である「ラクラク次の一手」^(10,11) の正解の手と、インターネット上でトップレベルのプログラムにより行われた持ち時間 3 時間秒読み 1 分の対局の棋譜^{*1}の指手を分析した。

具体的な方法としては、局面の合法手を全て列挙し全ての指手の評価値を求め、手番のプレイヤーにとって良い評価の順に整列させたうえで、正解や棋譜の指手が何番目に登場するかを調べた。指手の評価値を求める方法としては、通常の条件で 1 秒の探索を行った。また比較対象として、探索の代わりに指手の実現確率⁵⁾の高さを用いた場合も評価した。図 3 に「ラクラク次の一手」を分析した結果を掲載する。横軸が指手を評価値で整列させた場合の順位で、縦軸がその順位

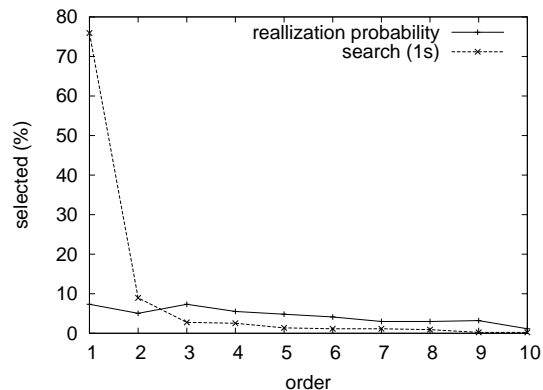


図 3 「ラクラク次の一手」問題集の正解手の予測

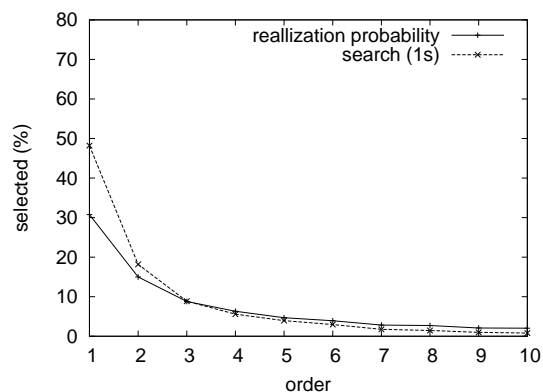


図 4 floodgate の棋譜一手の指手予測 (GPS 将棋以外の指手)

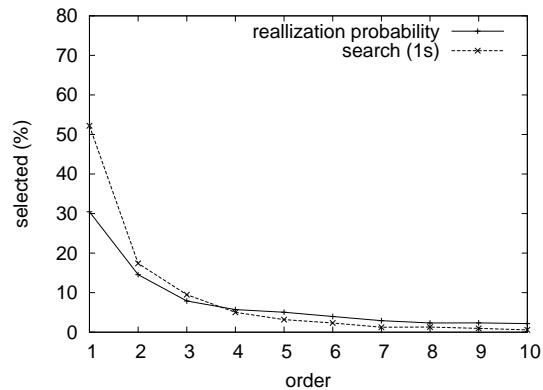


図 5 floodgate の棋譜一手の指手予測 (GPS 将棋の指手)

の指手が正解であった頻度を%で表記している。第一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%を越えたのに対し、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 4 はインターネットの対局のうち GPS 将棋以外

*1 http://wdoor.c.u-tokyo.ac.jp/shogi/view/latest-table.cgi?filter=floodgate-10800-60&show_self_play=1

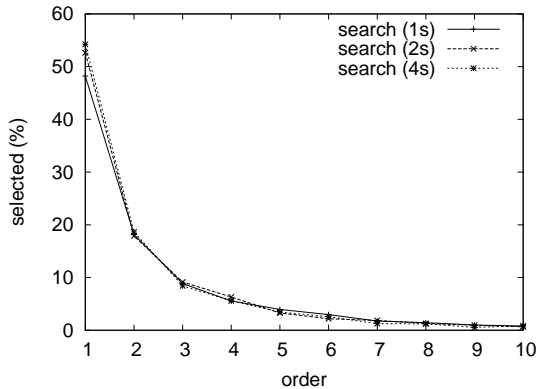


図 6 予測探索の秒数を変化させた場合の比較 (floodgate の GPS 将棋以外の指手)

表 1 並列に探索した場合の一致率の変化

ラクラク 次の一手	floodgate							
	1s		2s		4s			
2	65.4	72.7	56.7	75.7	55.1	74.1	55.4	74.5
4	66.1	69.5	59.2	72.1	58.5	71.6	59.6	71.9
8	65.8	73.9	58.2	74.0	57.5	73.9	59.0	74.4
16	69.3	77.5	55.4	73.4	56.1	73.8	58.0	74.9
32	70.9	79.8	53.4	71.2	54.1	72.2	55.8	73.8
64	74.8	83.7	52.8	70.6	53.1	71.1	55.0	73.5

のプログラムが指した手の分析, 図 5 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し手では, GPS 将棋の指し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, 問題集と比べると 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 で 7 割程度を締める一方で, 8 手目以降の裾野も長く, なかなか 0 になることはない。

図 6 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ムのものを用いた。グラフから, 第一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か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探索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。

最後に, 1 秒の探索に参加するスレーブの数を制限する実験を行った。これまでのこの章の実験では数百のスレーブがある状況を想定して, それぞれの手にスレーブを割り当てて 1 秒の探索を行った。一方, 現実の探索では前章で説明したように合法手の数がある時点で利用可能なスレーブの数を越える場合は, 溢れた分の指し手は「その他の手」とまとめられて一つのス

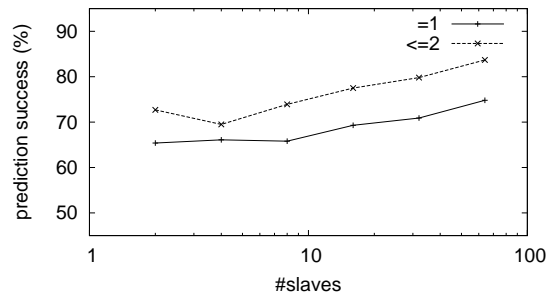


図 7 予測探索に参加する slave の数と予測成功率(ラクラク)

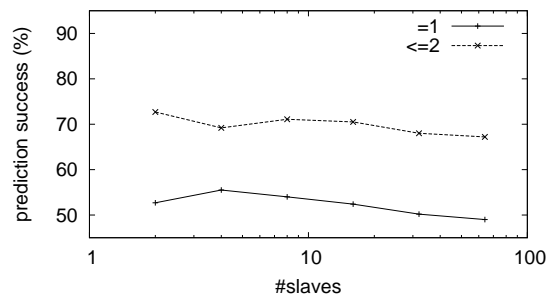


図 8 予測探索に参加する slave の数と予測成功率(floodgate)

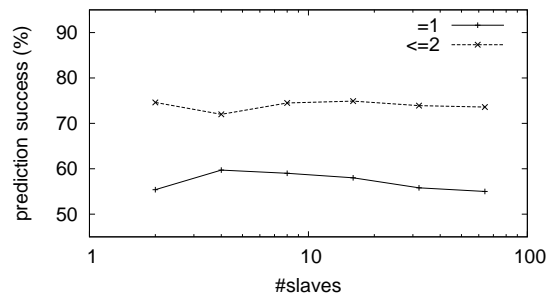


図 9 予測探索に参加する slave の数と予測成功率(floodgate, 4s)

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 7 はラクラク次の一手に対する結果である。横軸がスレーブの数, 縦軸が予測に成功した頻度を%で表す。第一手目の予想が正解手だった場合と, 第一手または二手目の予測に正解手が含まれていた場合の二通りを示した。どちらの場合も, 右肩上がりになっている。沢山のスレーブが参加するほど予測が正確になる傾向を示している。

一方, コンピュータ将棋の対局の棋譜を用いると図 8 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

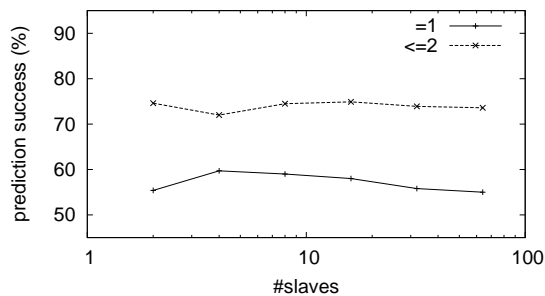


図 10 予測探索に参加する slave の数と予測成功率 (floodgate, GPS 将棋の指手)

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

異なる傾向となった。4 秒考えさせた場合の図 9 も、GPS 将棋が指した指手を予想させた場合の図 10 も傾向はあまり変わらない。原因は分析中であるが、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を増やすと様々な指手に目移りするようである。

4.2 メモリ共有並列探索に対する強さの比較

続いて、提案した手法の性能を測定するために、メモリを共有する並列探索を行うプログラムとの対局を行った。提案手法で戦うプログラムは各スレーブに 1 コアずつを割り当て、全体で 8 スレーブ用いた。一方、対局相手の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。持ち時間は 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を用いた予測読みは行わなかった。評価関数等探索以外の条件は全て揃えてある。

表 2 に勝敗を掲載する 8 プロセスを用いた提案手法のプログラムは、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、 $\alpha\beta$ ウィンドウやトランスポジションテーブルを共有できないため、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には大きな意義があると考えている。

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、この時の GPS 将棋の記録を簡単に分析して紹介したい。

まず図 11 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、速い機器は秒間の探索節点数が 100 万を越える一方で、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 の二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二つ割り当てた)。その結果、深さ 3 までの全ての節点と深さ 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。なお 50 台以上の機器を接続してもマスタープロセスの CPU 使用率は 10%程度であった。

棋譜のコンピュータ側の手番の局面について思考記録を分析した。清水女流王将の手番を分析から外した理由は、相手の手を予測する探索の際は思考時間が十分でない場合があるためである。まず図 12 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。候補手が 3 つある理由は、本稿の提案手法では、ルートでスプリットした 2 つの手それぞれと「その他の手」の合計 3 つの指し手について無限の $\alpha\beta$ ウィンドウで探索した評価値と PV が求まるためである。大まかな傾向として、評価値がもっとも良い第一の候補手がもっとも PV が

- 東京大学 GPS 所有クラスタ:
- 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 11 クラスタ版 GPS 将棋が利用した機器

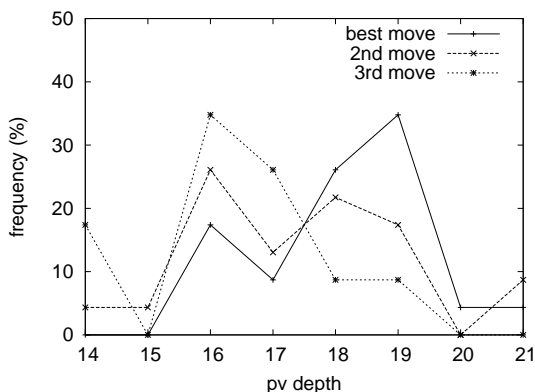


図 12 GPS 将棋のクラスタ版プログラムの PV の長さの分布

長い。しかし、その差はそれほど大きくはなく、これだけのデータでははっきりした結論は出せないものの安定した深さになっていると期待される。

図 13 に、同じくコンピュータ側の手番について、GPS 将棋のクラスタ版が選んだ三つの候補手の評価値の推移を描いた。GPS 将棋の定跡を外れた 16 手目以降で、GPS 将棋が形勢が開いたと判定した 70 手までを載せている。ただし、40 手目からしばらくは、途中で不調になった istbs173 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十分読んだと勘違いして候補手を決定してしまったため、分析から取り除いた。また記録のとり方が不十分で、第二、第三の候補手の評価に関しては指し手が決められた時点のものではなく、第一の候補手の評価が最後に更新された時点での評価になってしまっていることに注意されたい。全体として、先手

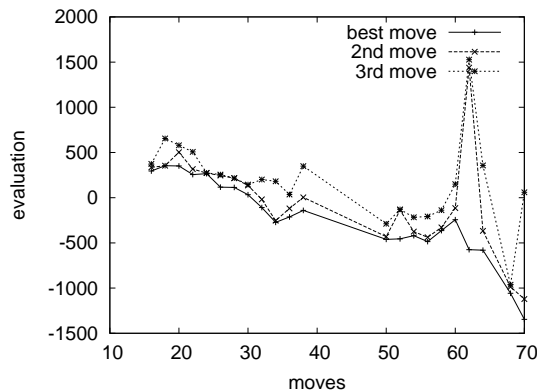


図 13 GPS 将棋のクラスタ版プログラムの評価の推移

少し指しやすそうという評価の序盤から、後手勝勢の終盤に推移していることが分かる。また、第一の候補手の評価と、第二、第三の候補手の評価が近い局面と大きく異なる局面があることも読み取れる。今回の対局で用いられた合議の枠組みでは、意見が合う時と合わない時で思考時間を変化させて時間を有効に活用する取り組みが行われたが、このような第二、第三の候補手や評価値を活用することでより効果的な合議が可能になると期待される。たとえば、第一の候補手以外も投票したり、二つの手で意見が割れた時でも二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 3 に、棋譜で指された指手と GPS 将棋のクラスタ版が選んだ三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し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 将棋の第一の候補手であった。それ以外の場合でも第一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三候補手にも入らなかった局面としては、18 手目の 72 銀と、58 手目の 52 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 65 銀のところ、激指が推奨していた 55 歩が 20 点差未満の僅差で第二候補手にあがっていることが挙げられる。第一の候補手以外に第二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、マスターが作成した探索木のリーフをス

表 3 GPS 将棋のクラスタ版の思考

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	22 飛	295	18	22 飛	335	18	44 角	374	17	92 香
18	72 銀	354	19	42 銀	354	18	92 香	655	17	72 金
20	42 銀	351	18	42 銀	503	21	52 金左	580	19	64 歩
22	54 歩	256	18	52 金左	317	18	54 歩	507	18	74 歩
24	53 銀	266	18	21 飛	273	18	94 歩	275	18	53 銀
26	44 角	117	19	64 銀	245	16	44 角	256	17	21 飛
28	同角成	114	19	同角成	216	17	52 金左	217	16	94 歩
30	64 銀	33	19	44 銀	135	16	64 銀	145	16	52 金左
32	44 角	-109	18	65 銀	-19	19	44 角	201	17	55 歩
34	65 銀	-273	17	65 銀	-254	19	55 歩	181	16	94 歩
36	同桂	-213	19	同桂	-119	17	77 角成	35	14	53 角
38	56 銀	-140	20	56 銀	4	16	76 銀	349	14	52 飛
50	74 桂	-461	19	74 桂	-431	14	64 歩	-289	16	35 歩
52	85 桂	-455	16	85 桂	-138	15	36 銀	-130	16	66 桂
54	77 桂成	-420	21	77 桂不成	-375	21	77 桂成	-217	16	55 歩
56	66 桂	-486	16	66 桂	-438	18	36 銀	-208	19	64 歩
58	52 金打	-364	19	36 銀	-329	16	56 銀	-138	17	35 歩
60	22 飛	-243	16	22 飛	-115	17	33 飛	150	14	33 金
62	同角	-575	17	同角	1439	16	69 金	1527	14	64 歩
64	69 金	-581	18	69 金	-367	16	37 角	357	17	33 角
68	56 銀	-1058	19	79 金	-987	19	56 銀	-963	16	55 角
70	55 角	-1346	16	55 角	-1121	19	47 銀成	60	16	79 金

レーブが独立に探索して評価値をマスターに報告する。このように枠組みを単純に保つことで、実装にあたって元のプログラムの変更が少ないという利点がある。探索木の作成においては、利用可能なスレーブが複数ある限り各節点において有力な候補手を再帰的に分割し続ける。GPS 将棋を用いた実験からは各節点の上位 2 手に集中的に資源を割り当てる単純な仕組みを用いて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。

これまで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で強さを向上させた報告はほとんどなかったが、このような単純な手法でも計算資源を投入すれば強さが向上することが分かってきた。そこで、分散環境での並列探索において、より本格的な手法(たとえば文献³⁾)を用いた場合にさらにどの程度強さを向上できるかどうかに興味深い状況となっている。

参 考 文 献

- 1) Brockington, M.G.: Asynchronous Parallel Game-Tree Search, PhD 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., Member, S. and Popowich, F.: Parallel Game-Tree Search, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.7, pp. 442-452 (1985).
- 5) 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).
- 6) 杉山卓弥, 小幡拓弥, 斎藤博昭, 保木邦仁, 伊藤毅志: 将棋における合議アルゴリズム -局面評価値に基づいた指し手の選択-, ゲームプログラミングワークショップ, pp.59-65 (2009).
- 7) 小幡拓弥, 杉山卓弥, 保木邦仁, 伊藤毅志: 将棋における合議アルゴリズム: 既存プログラムを組み合わせて強いプレイヤーを作るか?, ゲームプログラミングワークショップ, pp.51-58 (2009).
- 8) 田中哲朗, 金子知適: コンピュータ将棋の不遜な挑戦: 4. 大規模クラスタシステムでの実行 - GPS 将棋の試み -, 情報処理, Vol.51, No.8, pp.1008-1015 (2010).
- 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, 2, Vol.GI-24, 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).
- 12) 長井 歩, 今井 浩: df-pn アルゴリズムと詰将棋を解くプログラムへの応用, 情報処理学会論文誌, Vol.43, No.6, pp.1769-1777 (2002).