

ゲーム木に基づく並列探索での下位局面の分担

横山 秀^{1,a)} 金子 知適¹

概要: ゲーム木探索の並列化手法 P-GPP は、将来到達する可能性の高い有望な局面を予想し、ワーカ計算機に割り当てて非同期に探索させる手法である。これにより、先読みしていた局面が実際に現れた場合、長い時間その局面を探索していたことになり、pondering と同等の効果を得ることができる。また、有望度の順位が低い局面は、複数の兄弟局面をまとめてひとつのワーカで探索することで、合法手の分岐よりも少ないワーカ数での並列探索を可能にしている。提案手法では、有望度の順位が低い局面の扱いを改善し、より高い確率で pondering の効果が得られるような局面分担を行う。複数局面をまとめたワーカは、既存手法では兄弟局面に対して1つであったのに対し、提案手法では、複数のワーカを用いて均等に局面を分担する。この手法を評価するためのシミュレーション実験を行い、既存手法と同等以上の性能が期待できることを確認した。

Sharing Lower-Ranking Positions in Parallel Searching based on Game-Tree

SHU YOKOYAMA^{1,a)} TOMOYUKI KANEKO¹

Abstract: P-GPP is parallel game-tree search method. In P-GPP, promising moves which have a high possibility to play are predicted and assigned to the worker computers. Each worker asynchronously searches assigned moves in advance. When the predicted move is realized on the board, the worker in charge of the move takes advantage of additional thinking time. It is equivalent to the pondering effect. Moves which have low possibility to play are aggregated and assigned to one worker for each sibling group. We present an improvement method of worker allocation for low probability moves in order to enforce the pondering effect. In this method, multiple workers serve the group of low probability moves, through splitting equally the group. We evaluated this method in a simulation, promising results for improving play-strength were observed.

1. はじめに

チェスや将棋のような、精度の高い評価関数 [1] が作成されている完全情報ゲームは、min-max 原理に基づくゲーム木探索を行うことで強力なコンピュータプレイヤーが作成されている [2]。強いプレイヤーを実現するためには、対局の制限時間内でゲーム木を深く探索してより先の手を読むことが重要である。単一計算機の性能には限界があるため、木探索を多数の計算機を用いて並列に行うことが有効である。非共有メモリ環境での並列化手法として、著者らはブ

ロ棋士との対局で実績を挙げた GPS 将棋 [3] や、その改良手法であり、ゲームごとの特性に基づくパラメータの手動調整を排した P-GPP [4] を提案した。

P-GPP では、対局中に今後到達すると考えられる局面を予測し、これらの局面をあらかじめワーカ計算機に探索させておく。一台のマスタ計算機が図 1 に示したような **マスタゲーム木** を構築し、それぞれの葉（フロンティア・ノード）に対応する局面に対して各々一台のワーカ計算機を割り当てる。この際、有望度の低い兄弟局面は「その他」局面としてまとめて、一つのワーカに担当させる。これにより、限られたワーカ数でより多くの手を分担することができる。

ワーカ計算機はそれぞれ非同期に動作し、割り当てられ

¹ 東京大学総合文化研究科
Graduate School of Arts and Sciences, The University of Tokyo

^{a)} yokoyama@graco.c.u-tokyo.ac.jp

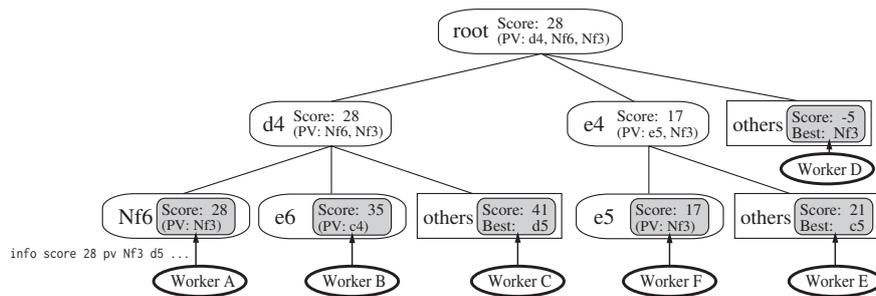


図 1: P-GPP 方式並列探索のマスターゲーム木とワーカの割り当て例。この木は初期盤面を根としており、6 個の葉それぞれに一台ずつワーカ計算機が割り当てられている。“others”と書かれている葉は、兄弟節点で分担されている手以外の全ての合法手を探索する。

た局面内での最善手を探索し、その評価値をマスに報告する。マスはマスターゲーム木に報告された評価値を保存し、最終的にマスターゲーム木に min-max 原理を適用することで、指し手を決定する。その後、新局面に対応するようマスターゲーム木を展開する。並列化の効果は、分担によりワーカあたりの探索がより深く行えるようになることから得られるのに加え、現局面より数手先の局面を探索しているワーカが存在することで、その局面が実際に指された際に担当ワーカから探索を引き継ぐことで、pondering と同様の効果が生じることによる。

本稿では、P-GPP 方式での、有望度が低い予測されてまとめられた「その他」局面に対するワーカ割り当てを修正し、低位の局面を手広く探索する手法を提案する。加えて、その有効性を予測するための実験を行い、その結果について述べる。

2. 関連研究

min-max 木探索の逐次実行では α - β 枝刈りなどの工夫を用いた探索の効率化がなされるが、それらのほとんどは探索順序に依存している。このため並列化の際には、無駄な探索が必然的に生じる。また、計算資源間での情報伝達が必要であることが、特に非共有メモリ環境では性能に影響する要因となる。

P-GPP 方式は Ethernet で構築された LAN のような、通信遅延の大きいネットワークで接続された計算機群を用いた非共有メモリ環境を前提としており、同期が必要な通信は両プレイヤーの着手時にのみ行われ、また通信量が少なく抑えられる。APHID [5] も各ワーカが非同期で探索を行うアルゴリズムだが、一手の思考時間中に局面の割り当てが変化するため、P-GPP よりも通信が頻繁に発生する。

YBWC [6] では、先行して実行される部分的な探索で得られた α - β 窓を利用して、待機していた残りの部分の探索を行うことで効率的な枝刈りを行っている。さらに浦ら [7] は YBWC を改良し、待機タスクに二段階の優先度をつけることで並列実行可能なタスクを増やし、待機する計算資源を削減している。また、TDSAB [8] では局面のハッシュ

テーブルに基づき探索を分散化している。これらの手法では、無駄な探索の軽減が可能である反面、計算資源間の通信が頻繁に発生する。

GridChess [9] は個々のワーカ計算機にゲーム木中の節局面を割り当てて Optimistic Pondering を行う並列化手法をとることで、あるワーカが一つの局面を長い時間探索できるようにした点で P-GPP と類似する。しかし、P-GPP がゲーム木に対して min-max 原理を適用し、複数のワーカから得られた評価値を総合して指し手を決定するのに対し、pondering の効果を得た根ワーカの結果のみを利用する点が異なる。また、ゲーム木の構造を一手の探索中に動的に変更する点も、P-GPP との差異である。

3. P-GPP 方式の分析

3.1 局面表を用いた pondering

P-GPP 方式の pondering では、予測局面が実現すると楽観的 (optimistic) に考えず、有望と予測した局面が実現しなかった場合でも先読みの効果が得られるように、全ての局面に対して探索ワーカを割り当てる。ただし、有望度の低い局面は兄弟ごとに「その他」としてまとめて、一つのワーカを割り当てる。各ワーカは内部に局面表 (ハッシュテーブル) を保持し、一度探索した局面変化を、局面表のエントリが溢れない限り記憶している。プレイヤーの着手により局面が変化した場合には、これまでに着手局面を担当していたワーカを用いて、指された局面を新たな起点として探索することで、局面表の情報を再利用することができる。

各ワーカは、担当する局面のなかから 1 つの最善手を見つけ出す探索を行うため、最善手とそれに連なる変化 (PV: Principal Variation) が重点的に探索され、局面表には PV に近い変化が多く記憶されると考えられる。それ以外の局面は、最善手でないことが判明した段階で枝刈りされる。よって、複数の局面をまとめて探索する「その他」ワーカがもたらす pondering の効果は、指された局面を PV として探索していたときに最も発揮されると考えられる。一つの手のみを担当するワーカは、pondering 開始時点で最善手

表 1: 局面表を用いた pondering の効果

pondering の種類	速度倍率の平均	平均時間 [ms]	局面数
(1)	46.4	303	7573
(2) すべて	23.3	420	7573
(2) PV 一致	35.1	315	4687
PV 不一致	4.2	592	2886
(3) pondering なし	1.0	703	7573

を一手に決め打ちして探索するのに対して、「その他」ワーカは、まとめられた候補手のなかから、最善手を pondering 中に動的に検出して探索を行う、といえる。

以上に述べた、局面表を用いた pondering の性能を計測する実験を行った。対象局面について、(1) 対象局面と同じ局面を pondering したワーカ、(2) 一手前の局面を探索する pondering を行ったワーカ、(3) pondering をしないワーカのそれぞれで、深さ 16 まで (深さ固定) の探索を行い、かかった時間を比較した。pondering の時間経過後、一度探索を停止し、得られた局面表の情報を利用して深さ固定の探索を行った。実験の諸条件は、[4] の P-GPP を用いた対局実験と同一であり、pondering 時間は対局での一手ぶんと同じ 950 ms とし、使用 CPU は Xeon E5-4650 の 1 コア 1 スレッドである。局面表のハッシュサイズは約 210 万エントリ (32 MiB) で、950 ms 間の訪問ノード数の半分程度である。P-GPP の対局実験で得られた棋譜の各局面を対象とした。なお、本実験では並列化を行っておらず、(2) は P-GPP で全ての合法手を 1 つの「その他」ワーカにまとめた場合に相当する。(1) は、一旦探索を中断して、局面表に残る途中経過以外を使用しない点を除けば、Optimistic Pondering と同等である。

実験結果を表 1 に示す。対象局面それぞれについて、pondering を行わなかった場合と比べての速度の向上率およびかかった時間を求め、これらを平均した。なお、(2) のケースについては、pondering 中に、対象となる局面を PV と判定していた場合と、そうでない場合とに分けて表示した。結果では、(1) の同じ局面の探索を二度繰り返すケースが最も pondering の効果があり、また、(2) の一手前の局面の探索を前置する実験では、pondering 中に次局面を PV として予測できていた場合に、高速化効果が強く見られた。

このことから、実際の局面が「その他」ワーカにまとめられていたとしても、その局面を PV と予測している限りは、単一のワーカによって探索された局面に準ずる pondering 効果が得られることが分かった。

3.2 非有望手の扱い

P-GPP では、探索開始時点までに得られている評価値を用いて局面の有望度 (実現確率) を予測する。有望度の高い局面に対しては、マスタゲーム木中のその局面を広く深く展開し、子孫局面を担当するワーカを増やすことで、

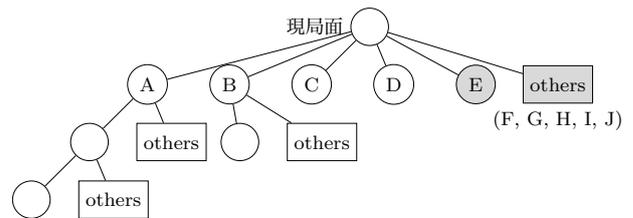


図 2: P-GPP 方式でのマスタゲーム木。現局面には合法手が 10 個あり、A から J までの英字が、有望度の高い順に割り振られている。

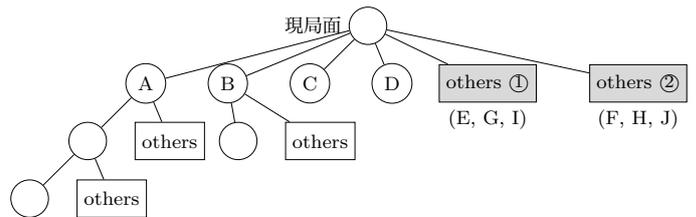


図 3: 提案手法でのマスタゲーム木。灰色の 2 ワーカの割り当てが変更されている。

有望な局面に多くの計算資源を割り当てる。割り当てるワーカの数には有望度が下がるほど減少するが、有望度の低い局面であっても、「その他」局面としてまとめられる低位の局面を除き、1 局面につき最低 1 つのワーカは割り当てられることとなる。図 2 の例では、A から J までの英字は有望度の順に付されているが、C, D および E の局面が 1 ワーカのみ割り当てを受け、F から J までは、「その他」("others")としてまとめられている。どの局面を重点的にマスタゲーム木に展開するかは、局面の有望さと、その局面に到達するまでの手番の数とで判断される。有望な手は、数手前の段階で、やや遠い未来の局面であるにもかかわらずマスタゲーム木に展開されて評価値が求められるのに対し、D や E のような有望度の低い局面は、指される可能性が生じる直前 (一手前) に「その他」葉から分離展開される。

有望度が低いために探索が進んでおらず評価値が得られていない局面を展開する際は、ワーカのもつ局面表の情報を利用する。局面表から、深くまで枝刈りされずに探索された記録のある局面を拾い出し、マスタゲーム木に加える。この情報は、評価値を得るための探索の途中経過として得られているに過ぎないため、正確性が低い。

図 4 の統計をみると、マスタゲーム木には有望度順位が 5 以下の局面が追加されている一方で、これらの低有望度の手を相手が指してくることは稀で、むしろ「その他」としてまとめられた手に該当するケースが多いことがわかる。

4. 提案手法

提案手法では、あるしきい値以下の有望度順位の局面には個別ワーカを割り当てず、残りの合法手を等分して分担

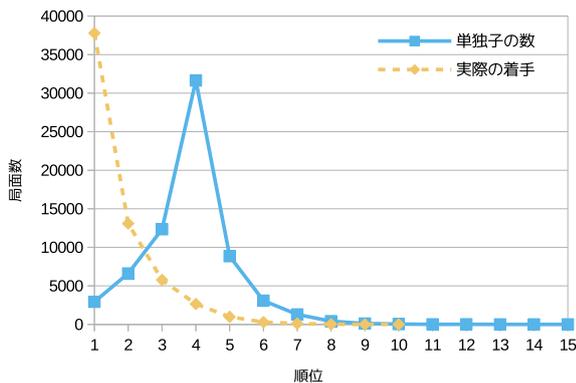


図 4: 16 ワーカーを使用した P-GPP 方式並列探索で相手の手番を探索したときの統計。青の実線は、現局面直下の局面を「その他」を除く何台のワーカー数で分担しているか、つまり、第何位の手までを個別に探索しているかを示す。橙の点線は、それらの子を有望度順に並べたとき、相手の指した手が何位に相当するかを示す。この他に、「その他」としてまとめられていた子が指された局面が、6,650 個ある。

する。これにより、正確性の低い情報に依拠して次局面を決め打ちするのではなく、探索の幅を広げ、複数の候補手のなかから PV として探索する局面を動的に決定する。探索の深さや精度が損なわれるが、次の手を PV として探索できる確率が高まるので、3.1 節の実験で得られたように、総合的には pondering の効果を得やすくなることが期待される。

図 2 と同じ局面に対するマスタゲーム木を、5 位以下の手について均等分担を行う提案手法を用いて作成した木が、図 3 である。4 位までの局面には変更がなく、5 位以下の 6 局面には、従前これらに割り当てられていた 2 つのワーカーが均等に割り当てられ、それぞれのワーカーが 3 局面ずつをまとめて探索する。これにより、あまり有望でない局面については、より手広く、広く浅く探索されることが期待される。

5. 評価実験

P-GPP 方式の並列化で用いるマスタゲーム木を、提案手法のマスタゲーム木に置き換える効果をシミュレーションする評価実験を、チェスの試合を題材に行った。なおこの実験では、提案手法をルート局面直下の子（深さ 1 の葉）のみに適用するものとした。

まず P-GPP 方式に、いくつかの小さな改善を加えた並列プログラムで対局実験を行い、対局中の全局面でマスタゲーム木の記録をとった。ワーカーとしては Stockfish DD を用い、16 並列とした。対戦相手は、逐次実行したワーカープログラムと同等とした。対戦結果は、全 1,000 試合で並列プログラムの 282 勝 82 敗 636 分（勝率 60.0%）となった。

表 2: シミュレーション実験の結果

既存手法での扱い	提案手法のシミュレーション		
	単独子	「その他」の PV	それ以外
単独子	3819	226	38
「その他」の PV	-	191	57
それ以外	-	119	86

次に、記録したマスタゲーム木のうち、提案手法の適用対象となる局面、つまり k 位以下の手は均等割当するとのしきい値 k に対して、マスタゲーム木に「その他」を含めた葉が $k+1$ 以上あるもののみを抽出した。これらの局面の、ルート局面直下の子のうち、 k 位以下の葉に対して、局面を均等に割り当て直した場合の探索を追加で行い、木を上書きした。

上記の実験をしきい値 $k=6$ で行い、従前の木と提案手法を模した木とが、実対局で指された手を予測できていたかを示したのが、表 2 である。マスタゲーム木作成の時点で有望と予測できており、単独の子として既に分割していた場合を「単独子」とし、「その他」にまとめられていた場合を、探索終了時に PV と評価していたか、そうでないか（「それ以外」）により分けた。既存手法で単独の子や PV として予測できていなかった局面を、新たに PV として予測できるようになった局面が 119 個あった。その逆で、予測を逸した局面も 95 個（38+57）あった。個数でみる限りでは、性能向上が期待できる局面のほうが上回った。

それぞれの分類を、3.1 節の実験で行った pondering と対応させると、「単独子」は (1) の同局面を先読みする pondering を行ったところに相当し、残りの二者は、(2) の一手前の局面を先読みする pondering を行ったことに相当する。既存手法で単独子でありながら、提案手法では「その他」としてまとめられてしまった局面は、行われた pondering が (1) から (2) に劣化してしまったことを意味する。しかし、そのうち 226 局面は、「その他」の中でも PV として評価された局面と一致しているため、性能の大きな障害にはならないと予測される。

今回のシミュレーション実験では、提案手法で割り当てが変更されたワーカーは、局面表ハッシュを持たない状態で探索を行っているのに対し、実際の対局実験中に探索を行っていた既存手法の「その他」ワーカーは、このワーカーがマスタゲーム木に導入されたときから継続して探索を行っており、局面表に情報が含まれた条件下で探索を行っている。棋譜手を既存手法で「その他」ワーカーの PV として予測していながら、シミュレーション下では PV と予測できなかった局面（57 局面）があるのは、このことも要因と考えられる。

6. まとめ

P-GPP 方式で並列化されたプログラムの対局経過をも

とに、並列化効果のもととなっている pondering の寄与をケース別に分析した。これをもとに、現状では強さの向上に資していないワークに、より価値のある pondering を行わせるための、ワーク割り当て手法を提案した。この手法により得られるマスタゲーム木によって、実対局中に既存手法で得られたマスタゲーム木を上書きするシミュレーション実験を行い、相手の手の予測精度を比較した。その結果、提案手法は、既存手法と同等かそれ以上の性能が期待できることがわかった。

参考文献

- [1] Hoki, K., Kaneko, T.: Large-Scale Optimization for Evaluation Functions with Minimax Search. *Journal of Artificial Intelligence Research* Vol. 49, pp. 527–568 (2014).
- [2] Campbell, M., Hoane Jr., A.J., Hse, F.h.: Deep Blue. *Artificial Intelligence* Vol. 134, pp. 57–83 (2002).
- [3] 金子 知適, 田中 哲朗: 多数の計算機を活用したゲーム木探索技術の進歩 —三浦弘行八段と GPS 将棋との対局を振り返って—. *情報処理* Vol. 54, No. 9, pp. 914–922 (2013).
- [4] Yokoyama, S., Kaneko, T., Tanaka, T.: Parameter-Free Tree Style Pipeline in Asynchronous Parallel Game-Tree Search. 14th International Conference on Advances in Computer Games, to be published in *ICGA Journal* (2015).
- [5] Brockington, M.G., Schaeffer J.: APHID Game-Tree Search. *Journal of Parallel and Distributed Computing*, Vol. 6, pp. 90–114 (1997).
- [6] Feldmann, R.: Game Tree Search on Massively Parallel Systems. Ph.D. Thesis, University of Paderborn (1993).
- [7] Ura, A., Yokoyama D., Chikayama T.: Two-level Task Scheduling for Parallel Game Tree Search Based on Necessity. *Journal of Information Processing*, Vol. 21, No. 1, pp. 17–25 (2013).
- [8] Kishimoto, A: Transposition Table Driven Scheduling for Two-Player Games. M.Sc. Thesis, University of Alberta (2002).
- [9] Himstedt, K.: Gridchess: Combining Optimistic Pondering with the Young Brothers Wait Concept. *ICGA Journal* Vol. 35, No. 2, pp. 67–79 (2012).