評価値を用いて展開制御したゲーム木に基づく並列探索

横山 秀[1,a)]　金子 知適[1]

概要：廉価なネットワークで接続されたグリッドコンピューティング環境を前提としたゲーム木探索の並列化手法である P-GPP の改善手法を提案する。P-GPP では，ワーカ計算機に特定局面を分担させ非同期に探索を行う。マスタ計算機は内部にゲーム木を保持し，その葉にワーカ計算機を対応付けることで分担局面を決定する。実現する確率の高い局面をゲーム木に追加することで効率的な分担ができる。局面の実現確率を計算するのに，P-GPP はゲーム木の兄弟局面間での評価値の順位を用いたモデリングを行った。提案手法では，より正確なモデリングを行うため，評価値の順位だけでなく，差を利用する。これを実装してチェスの対局実験を行ったところ，勝率においては P-GPP から顕著な進歩は見られず同等程度の成績だったものの，応用に有望な指標を示した。

Parallel Searching based on Game-Tree Expanding Controlled by Evaluation-Value

Shu Yokoyama[1,a)]　Tomoyuki Kaneko[1]

Abstract: P-GPP is a parallel game-tree search method by utilizing many computing nodes connected in low-cost network systems. In P-GPP, the master program manages a game-tree and distributes positions in the tree to workers. Then, each worker asynchronously searches the best move and evaluation for its assigned position. This paper presents a new method for concentrating more workers on the promising moves to improve playing strength. In our method, the difference of evaluation values between siblings is applied for choosing the promising moves, in addition to the rank of evaluation value. We implemented our method in chess. The game results didn't show progress clearly in the winning probability, but promising results were observed in some statistical properties.

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が作成されている [2]。より強いプログラムを実現するためは，この木探索を多数の計算機を用いて並列に行い，ゲーム木を深く探索することが有効である。非共有メモリ環境での並列化手法として，著者らはプ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，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。ワーカ計算機は割り当てられた局面をそれぞれ非同期に探索して評価値をマスタに報告する。マスタはマスタゲーム木に報告された評価値を保存し，最終的にマスタゲーム木に min-max 原理を適用することで，指し手を決定する。

マスタゲーム木は，両プレイヤーの着手により盤面が変化したとき，新局面を根とし，また葉の数がワーカ数と一致するように展開されなければならない。図 2 では，図 1 のゲーム木が展開する様子を簡略化した記法で示した。ゲーム木の展開時には，有望な局面を重点的に展開して待機中

[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

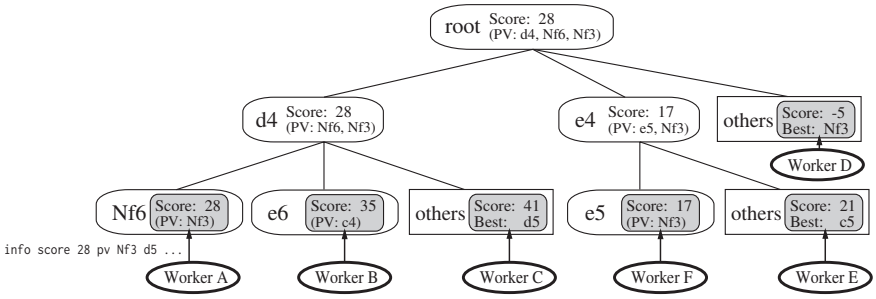
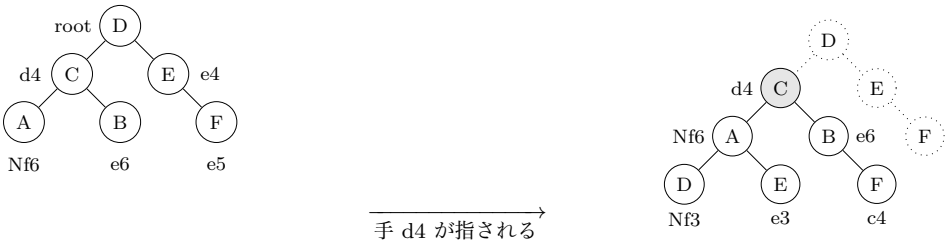


図 1: P-GPP 方式並列探索のマスタゲーム木。この木は初期盤面を根としており，6 個の葉それぞれに一台ずつワーカ計算機が割り当てられている。“others” と書かれている葉は，兄弟節点で分担されている手以外の全ての合法手を探索する。



(a) 図 1 の簡略表記

(b) 局面変化後のマスタゲーム木

図 2: (a): マスタゲーム木を図示するときは，以降はこの図のように，“others” を親節点と同一視して描く。このとき，全てのワーカ計算機は，図中の節点と対応している。 (b) 手 d4 が指された場合の，マスタゲーム木の拡張例。ワーカ C の節点が新たな根となる。ワーカ D・E・F は探索を継続する必要がなくなるので，手 d4 以下の新たに展開された葉に再割り当てされる。

のワーカ（計算資源）を割り当てることで並列化効率の向上が期待できる。一度ワーカが割り当てられた局面の探索は，その局面に変化しないことが確定し，マスタゲーム木から取り除かれるまで継続される。将来の盤面に出現する局面が早い段階からマスタゲーム木に追加されていれば，そのぶん同局面を長期間探索することができる。このため P-GPP では，直前の探索までで得られているマスタゲーム木の各節点の評価値を順位付けし，高順位の節点以下を優先してワーカを割り当てる。

本稿では，評価値の順位に替わって，最善手との評価値の差を利用したマスタゲーム木展開手法を提案し，チェスプログラムを実装することで有用性を検証する。

2. 関連研究

2.1 min-max 木探索の並列化

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。

2.2 ゲーム木展開制御手法

評価値を基準としてゲーム木を展開する操作は，横山大作ら [10] の，評価関数とモンテカルロ木探索を組み合わせた手法の将棋への適用で行われている。この手法では，プレイアウトを行った回数に応じて評価値の確率分布を定め，Bayesian Approach [11] で提唱されている QSS (Q step size) アルゴリズムを用いて展開するノードを決定している。

また，鶴岡ら [12] の将棋ゲーム木の探索範囲を限定する手法では，駒の取り返しといった着手の特徴ごとに，その着手が既存棋譜中で行われた割合を統計的に求める。これを基に着手が実際に行われる確率（Transition Probability: **遷移確率**）を推定し，ある局面が実際に指される確率（Realization Probability: **実現確率**）を，局面までの手順に沿って遷移確率を掛け合わせることで求め，実現確率がしきい値よりも高いノードのみを展開する。

3. 実現確率を最大化するゲーム木展開

P-GPP 方式の並列化効果は，現局面よりも先の局面をマスタゲーム木に追加し予め探索しておくことから得られるため，追加した局面が実際に指される可能性が高いほど効果が高い。そこで，新しい葉を展開したときに得られる効用を，その葉局面がもつ実現確率によって評価し，葉の深さで重み付けられた実現確率の合計が最大となるようにゲーム木を展開する。ただし，図 1 で “others” と示されている葉は，同時に複数の兄弟局面を担当する葉だから，深さは 1 小さいものとして扱う。つまり，マスタゲーム木を図 2a に示した簡略表記において，節点の集合を V とし，節点 $v \in V$ の深さを d_v，v に到達する確率（v の実現確率）を p_v としたときの，マスタゲーム木全体の実現深さ期待値（式 1）を最大化するように展開を行う。

$$\sum_{v \in V} d_v p_v \tag{1}$$

マスタゲーム木の展開は，図 3 のようにゲーム木に追加したときの実現確率が最大であるような葉を選択し，これを木に追加することで行う。なお，マスタゲーム木への葉の追加は，局面の変化によりマスタゲーム木の傘下から外れて取り除かれ，対応するワーカ計算機が待機状態となった葉の個数ぶんだけ行うので，一手着手後のマスタゲーム木の展開で 2 個以上の葉を追加する場合もある。このときは，1 個の葉を追加した後の木を対象に，改めて上記した葉の選択を行う。これを追加する葉の個数ぶんだけ繰り返す。

この操作で実現深さ期待値（式 1）が最大化できるのは，ある親節点 v_0 から子節点 v_c を展開したときは，親節点の “others” 葉に対応する実現確率が p_{v_c} だけ減る一方で，$d_{v_c} = d_{v_0} + 1$ であることから，木全体の実現深さ期待値の増分は p_{v_c} となることによる。

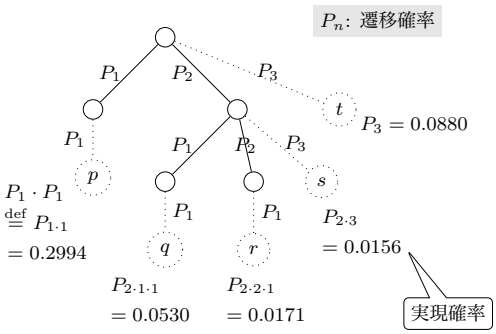


図 3: P-GPP 方式での，マスタゲーム木に追加される葉の選択。p から t の追加候補についてそれぞれ実現確率を算出し，確率最大である p が追加される。ここで実現確率は，兄弟節点内での評価値の順位 n に対応した遷移確率 P_n を木に沿って掛け合わせることで算出されており，この P_n は木中全ての階層で単一である。

各着手の遷移確率は，前節で述べた鶴岡らによる着手の特徴を用いる手法とは異なり，マスタゲーム木における兄弟ノード間での評価値の順位によって定める。評価値の順位と遷移確率との関係は，ワーカプログラムによって熟練者の既存棋譜中の局面をあらかじめ探索し，ワーカが局面を第何位の手と評価するかの統計を取ることで算出する。

上記の手法を用いることで現在のマスタゲーム木のどの部分を展開するかを決定した後，その部分（ここでは親局面という）の子孫局面のうち，どの局面をマスタゲーム木に追加するかは，親局面を担当していたワーカが直前の探索までで行っていた深さによって選ぶ。ワーカ内部の探索では，有望でない局面は早期に枝刈りされ深く探索されないから，逆に深く探索された局面は有望であるとの予測に基づき，探索深さを簡易的な指標として利用する。この操作を行うために各ワーカは，各手番の終了時にマスタに対し，深く探索していた局面の上位いくつかを報告する通信を行う。

この方式はゲームごとにルールを考察して指し手を分類・調整する必要がなく，評価関数による逐次探索プログラムが確立している様々なゲームに容易に応用可能であるという利点を持つが，評価値の情報を，その順位のみに着目して利用している点に改善の余地がある。具体的には，兄弟ノードの持つ評価値との関係が使われていない点が問題である。例えば，最善手の評価値と次善手の評価値が近く，どちらを指してもほぼ同等に有利であるような局面にあっては，両着手の遷移確率は順位によらず同程度と考えるのが自然である。

4. 提案手法

そこで本稿では，Coulom [13] による，Bradley-Terry モデルに基づき，囲碁の着手の特徴（フィーチャー）ごとに

打たれる確率を棋譜学習し，これらを組み合わせて着手の確率を求める方法を応用し，兄弟節点間の評価値の差から遷移確率を算出する手法を提案する。Bradley-Terry モデルでは，プレイヤー i と j の間の対戦での勝率は，それぞれのプレイヤーの「強さ」を正の値で表した（強いほど大きい）ものをそれぞれ γ_i, γ_j としたとき，

$$(i \text{ が } j \text{ に勝つ確率}) = \frac{\gamma_i}{\gamma_i + \gamma_j}$$

となる。提案手法では，これを多数プレイヤーに一般化し

$$(i \text{ が } n \text{ 人中で勝つ確率}) = \frac{\gamma_i}{\gamma_1 + \gamma_2 + \cdots + \gamma_n} \qquad (2)$$

としたうえで，ある局面の合法手の評価値差が所属する階級を各「プレイヤー」と考え，学習する棋譜中に現れる各局面をそれぞれ一つの勝負と考え，棋譜で指された手を「勝ち」，その他を「負け」とする。なお Coulom は手の特徴を「プレイヤー」とし，これらのチーム戦と考えることで手が複数の特徴を持つ場合に対応したが，本稿では，ある手が複数の階級に属することはなく，常に個人戦になる点が簡素化されている。

Coulom によれば，それぞれの階級に対応する γ の値は，最小化最大化 (Minorization-Maximization) アルゴリズムによって，反復計算することで求めることができる。この反復計算の式は，先述した簡素化のもとでは，以下のようになる。

G ： 棋譜中に現れる局面の集合
L_j ： 局面 j がもつ合法手の集合
$c(m)$ ： 手 m が所属する評価値差階級

$$\gamma_i \leftarrow \frac{W_i}{\sum_{j \in G} \frac{C_{ij}}{E_j}} \qquad (3)$$

なお，

$$W_i = \left|\{j \in G \mid f(\text{棋譜手}) = i\}\right|$$
$$C_{ij} = \left|\{m \in L_j \mid c(m) = i\}\right|$$
$$E_j = \sum_{m \in L_j} \gamma_{c(m)}$$

である。

5. 棋譜からの学習

全ての合法手を調べることとすると，指される可能性の極めて低い手までも探索しなければならないなど非効率なので，棋譜中の局面を探索し，指すべき手の上位 10 個として出力された手のみを，評価値差の階級を付す対象とした。その他の手は，まとめて一つの「ランク外」階級とした。「ランク外」階級とする手は，探索結果が 10 個全て出力された局面の L_j にのみ加えた [♡1]。また，詰みが見えており評価値による評価がなされない局面は除外した。

既存棋譜中の局面の探索には，オープンソースのチェスプログラム Stockfish DD[♡2] を利用した。Stockfish は最強クラスのチェスプログラムの一つであり，後述する対局実験でもワーカ計算機のプロセスとして利用した。

1996–1997 年に行われた Deep Blue 対 ガルリ・カスパロフの 12 試合全 1,091 局面を Stockfish に探索させ，結果を分析した。探索は各局面に対してそれぞれ新たな Stockfish プロセスを作成して 1 スレッドのみで行い，上位 10 個の手を出力するように設定し，300 万ノードを探索するまで続けた。なお，Intel Xeon E5-4650 プロセッサ環境においては，1 局面あたりの探索に平均 1.5 秒要した。これらの指定のため，Stockfish に以下の UCI (Universal Chess Interface) コマンドを発行した。

```
setoption name MultiPV value 10
position startpos moves [局面]
go nodes 3000000
```

最善手との評価値差の階級を 10 センチポーンずつ，0, 1–10, 11–20, ... , 71–80, 81– の 9 段階に区切り，「ランク外」と併せて 10 種の階級とした。各 γ_i の初期値を適当な同じ非負の値とし，式 3 を 1000 回反復する学習を行った。この結果を，評価値差 0 の階級の γ_0 を 1 とした相対値で表したのが図 4 である。概ね，評価値差が開くにつれて値が小さくなっている，つまり棋譜手と合致しにくくなっていることが確認できる。

この結果を使って，例えば図 2 中の root 局面（MAX ノード）の子では，d4 が最善手であり，兄弟間での評価値の差はそれぞれ 0 (d4), 11 (e4), 33 (others) となっている。“others” は，階級 31–40 の手と「ランク外」の手とを扱っていると近似することとして，d4 が選ばれる確率は式 2 に当てはめて

$$P(\boxed{\text{d4}}) = \frac{\gamma_0}{\gamma_0 + \gamma_{11\text{–}20} + \gamma_{31\text{–}40} + \gamma_{\text{外}}} \approx 0.729$$

となるし，

$$P(\boxed{\text{others}}) = \frac{\gamma_{31\text{–}40} + \gamma_{\text{外}}}{\gamma_0 + \gamma_{11\text{–}20} + \gamma_{31\text{–}40} + \gamma_{\text{外}}} \approx 0.101$$

である。

なお，図 5 では，最善手として得られた着手の評価値と，棋譜手（熟練者の手）との評価値との階級ごとに，棋譜手の評価順位の構成比を示した。これによれば，低順位と評価された棋譜手でも，評価値差が小さいものがあることが確認できる。

[♡1] 合法手がちょうど 10 個の場合，他に合法手がないにもかかわらず「ランク外」手が加えられることになり正確でないが，稀なケースであり，考慮しないこととした。

[♡2] https://stockfish.s3.amazonaws.com/stockfish-dd-src.zip 2013 年 11 月 30 日公開版

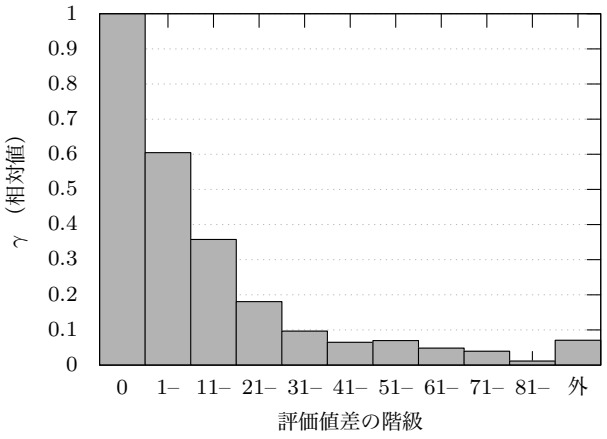


図 4: 評価値差の階級と γ 値（選ばれやすさ）との関係

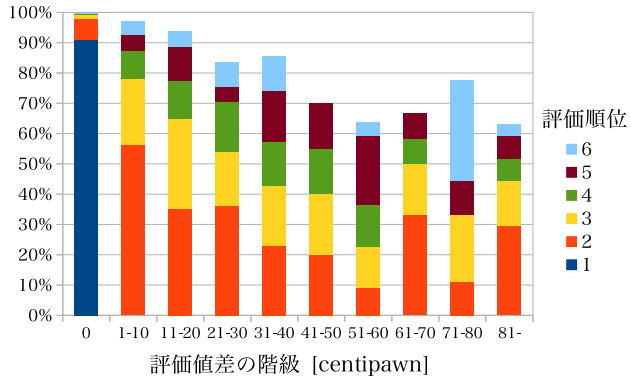


図 5: 最善手と棋譜手との評価値差の階級と，評価順位の関係。探索の結果，ある階級の評価値差を得た局面が，各評価順位に分類される割合を 2 位から 6 位まで示した（棋譜手が上位 10 位に入らなかった「ランク外」の局面は分母に算入されていない）。なお，評価順位 1 のときの評価値差は常に 0 となる。

6. チェスでの対局実験

6.1 実装

P-GPP の対局実験に使用したプログラムを改変し，提案手法である，兄弟節点間の評価値差を基準とする遷移確率計算を行うようにした。ただし，これは兄弟節点全ての評価値が分かっている局面に対してのみ使用することができる。よって，マスタゲーム木に複数の葉を追加するとき，一つ目の葉をゲーム木に追加した後の実現確率の計算では，新規追加葉が兄弟に存在する部分の遷移確率は，従来手法 (P-GPP) と同じく順位に基づいて算出した。

ワーカプログラムは，予備実験で用いた Stockfish DD と同一のもので，探索に関係ない多少の拡張・変更を行った。マスタプログラムは，これらと TCP 通信を行い，テキストベースのプロトコルである UCI (Universal Chess Interface) で情報をやりとりするもので，boost/asio ライブラリを用いた C++で実装されている。

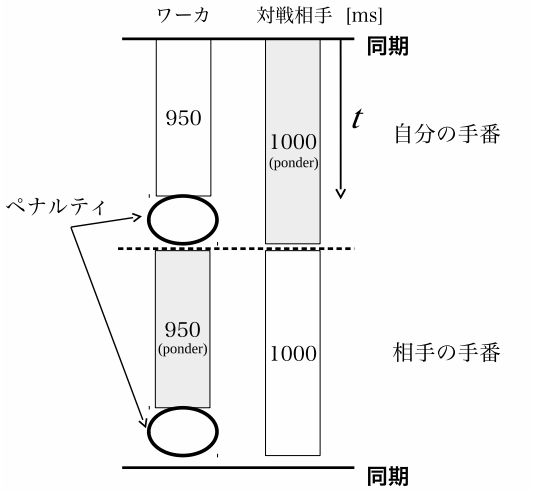


図 6: 両プレイヤーの動きの模式図。一手番が終わるごとに同期し，各々は固定時間だけ思考する。

個々のワーカプログラムは，均一な CPU コア (Intel Xeon E5-4650) 上で逐次探索プログラムとして，コア数を上回らない数のプロセスを稼働させた。一つのワーカあたり，32 MiB(約 210 万エントリ) の局面表 (Transposition Table) を持つ。

6.2 対局設定

ワーカプログラムと同じ逐次プログラムを対戦相手とし，本プログラムは参加するワーカ計算機の数を変えて並列化の効果を調べた。対局にランダム性をもたせるため，序盤の着手は定跡ファイル performance.bin [♡3] に記録された定跡を，チェスプロトコル変換プログラムの PolyGlot の機能でランダムに指すようにした。両者交互に，同じ回数だけ先手を持たせた。

対局の思考時間は一手あたり 1 秒に固定した。ただし，並列化によって生じる通信オーバーヘッドを，実際に大規模な計算クラスタを用いない単一の計算機上での実験で再現するため，並列化プログラムの思考時間には一手あたり 50 ms のペナルティを与え，実際の思考時間は 950 ms とした（図 6）。マスタ・ワーカ間で行われる通信は，着手決定時に 2 KiB ほど行われるのみであるから，Ethernet のような低速通信環境を前提としても，十分なペナルティであると考えられる。

提案手法を実装しての実験は，ワーカ数 8 および 16 は 1,000 試合，ワーカ数 32 では 100 試合行った。なお，P-GPP による対戦は，全て 1,000 試合である。

6.3 結果と分析

本実験での対局結果を表 1 に示す。なお，勝率は引き分けを 0.5 勝として計算したものである。既存手法の勝率と

[♡3] http://wbec-ridderkerk.nl/html/download.htm

表 1: 対戦結果

ワーカ数	提案手法				P-GPP			
	勝	敗	分	率	勝	敗	分	率
8	223	111	666	55.60%	224	105	671	55.95%
16	273	89	638	59.20%	267	82	651	59.25%
32	29	8	63	60.50%	313	64	623	62.45%

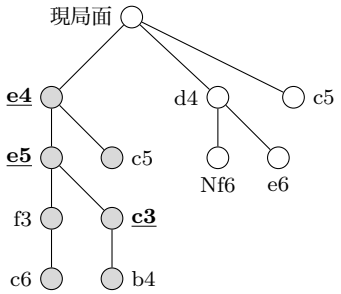


図 7: **引継ワーカ数**: このマスタゲーム木が作られた後に指された実際の着手が e4 だった場合，灰色で塗られた 7 個の節点はゲーム木の展開後も残留し，このまま同じ局面を探索し続けることができる。このようなワーカの数をいう。 **最深一致長**: さらに対局が ”(e4 →) e5 → c3 → f6 → f3 . . .” と進行した場合，ゲーム木中の節点は，最大深さ 3 まで一致している（下線部）。これを最深一致長とする。

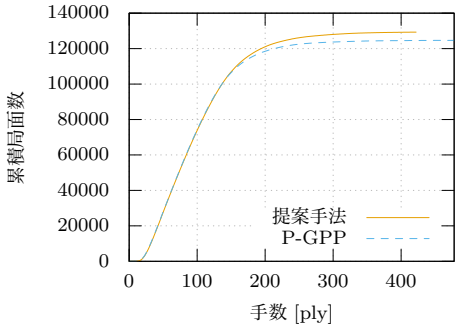


図 8: 総ワーカ数 16 での対局結果中の局面の手数（累積値）

比較したが，ほとんど差はみられなかった。ワーカ数 32 は対局数が少ないため，参考としての値である。

また本手法を導入したことにより，対局中に使用されたマスタゲーム木の性質がどのように変化したかを調べるため，手数が 100 手 (ply) 以内のものを対象としてマスタゲーム木の分析を行った。本稿の対局実験では投了や引き分けの合意を採用せず，ルールに従い結果が確定するまで対局を続けたため，勝敗がほぼ決定した後の終盤や引き分けを成立させる過程で，取るべき手が一つしかないか極めて限られている局面が現れる。手数制限は，これらの並列化の必要性が薄い一方で，同型反復により長期になり平均値に与える影響が大きい局面を計算から除外するためである。典型的なチェスの対局が投了までに要する手数は 80 手 (40 moves) ほどである [14] から，分析対象を 100 手以内に限定しても，プログラムの強さに影響する探索局面の

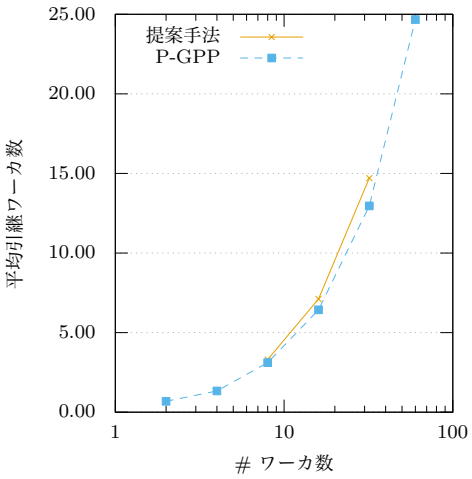


図 9: 1 局面あたりの平均引継ワーカ数

大部分は対象になっていると考えられる。平均ワーカ数 16 の対局（各手法それぞれ 1,000 局）を例にすると，図 8 に局面の手数の累積値を示したように，分析の対象となる局面数は約 7.4 万，全体の約 58%である。

また，ある局面のマスタゲーム木の節点のうち次局面を探索するために展開したゲーム木にそのまま引き継がれた節点の個数（「引継ワーカ数」）や，マスタゲーム木と続く実際の対局進行とが一致した最長部分の深さ（「最深一致長」）について，同様に平均をとった。これらの計算例を，図 7 に示した。

これらの指標は双方とも，一度マスタゲーム木に追加され探索が行われた局面の情報が，後に効果的に利用されたかを表しているから，数値が高いほうが効率的な探索が行われているといえる。対局に参加した総ワーカ数を横軸に対数表示でとり，局面ごとの引継ワーカ数（図 9），および最深一致長（図 10）を示した図によれば，総ワーカ数 16 の対局で P-GPP を上回っている。特に最深一致長を示した図 10 に注目すると，P-GPP でワーカ数を 3 台増やし 19 とした場合に相当する結果が得られている。

生成マスタゲーム木じたいの深さ（最深葉の深さ）について，平均を図 11，分布を図 12 に示した。総ワーカ数 16 の対局については，P-GPP が平均 4.17 であるのに対して提案手法では 4.66 となり，またその他の総ワーカ数の対局でも同様に，提案手法のほうが高い値であった。また，同じく総ワーカ数 16 の対局について，引継ワーカ数の分布を調べた図 13 をみると，P-GPP では 10 で頭打ちになっている一方で，提案手法では 5–7 付近の低い値が若干多いが，11 以上となった割合も大きい。このことから提案手法は P-GPP に比して，狭く深い探索，つまり最善手を指す遷移確率を相対的に高く見積もり，有望な変化に多くの計算資源を投機的に割り当てる傾向をもつといえる。

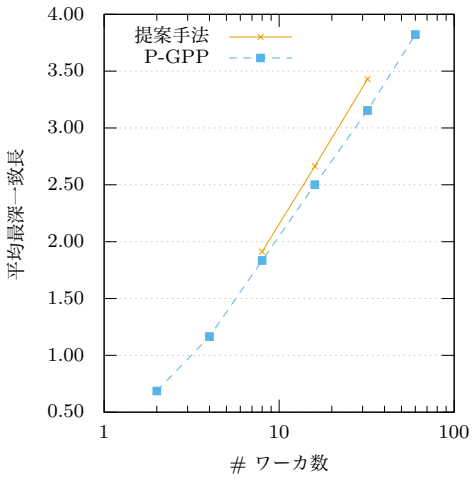


図 10: 1 局面あたりの，最深一致長の平均

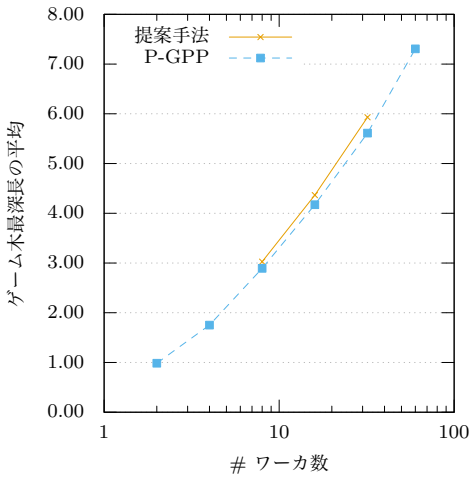


図 11: 1 局面あたりの，マスタゲーム木中の最も深い葉の深さ

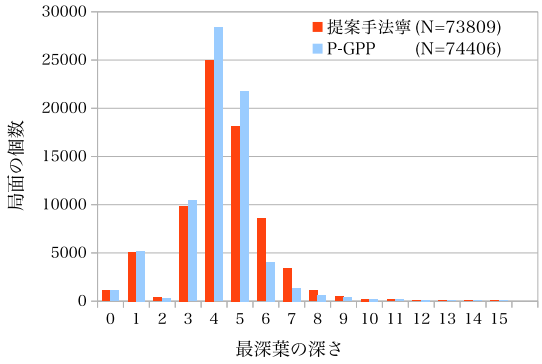


図 12: 総ワーカ数 16 での，マスタゲーム木中の最も深い葉の深さの分布

7. まとめ

実現する確率の高い局面の探索を優先的にワーカ計算機に分担させ，各ワーカが非同期に探索した結果を min-max

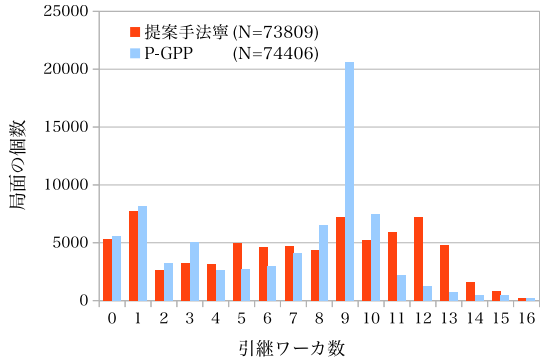


図 13: 総ワーカ数 16 での，引継ワーカ数の分布

原理によって取りまとめることで並列化を行うゲーム木探索手法である P-GPP を，局面実現確率の計算において評価値差を考慮することで改善する手法を提案した。提案手法をチェス対局を題材として実装し，逐次プログラムとの対局成績を既存手法 (P-GPP) と比較したところ，十分に有意といえる差は顕れなかった。しかし，マスタゲーム木の概形，つまりワーカ分担の戦略は P-GPP と異なっていることが確認されたうえ，指標のうえでは P-GPP よりも優れた分担並列化が行われていることがうかがわれた。よって，今回行わなかった，例えば長い持ち時間といった対局条件のもとでは提案手法が最終的な勝率に寄与する可能性も検討する価値があるだろう。

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).

[10] 横山 大作, 喜連川 優：ベイジアンアプローチに基づくモンテカルロ木探索アルゴリズムの将棋への適用と評価. 情報処理学会論文誌 Vol. 55, No. 11, pp.2389–2398 (2014).
[11] Baum, E. B., Smith, W. D.: A Bayesian Approach to Relevance in Game Playing. Artificial Intelligence Vol. 97, pp. 195–242 (1997).
[12] Tsuruoka, Y., Yokoyama, D., Chikayama, T.: Game-tree search algorithm based on realization probability. ICGA Journal Vol. 25, No. 3, pp. 145–152 (2002).
[13] Coulom, R.: Computing Elo Ratings of Move Patterns in the Game of Go. ICGA Computer Games Workshop (2007).
[14] Shannon, C. E.: Programming a Computer for Playing Chess. Philosophical Magazine, Ser. 7, Vol. 41, No. 314, pp. 256–275 (1950).