

コンピュータ将棋における相手考慮時間を含めた時間制御の考察

芝 世式^{†1}

概要: 本研究はコンピュータ将棋における対戦勝率を向上させる手段として考慮時間を有効活用する手法を提案するものである。具体的な実装を行いその効果のほどを確認した後、世界コンピュータ将棋選手権等において有効性を示した。

キーワード: コンピュータ将棋, 木探索, 探索時間

Effective Time Control with Opponent Consideration Time on Computer Shogi

SEIJI SHIBA^{†1}

Abstract: In this research, we propose a method to effectively utilize consideration time as a means to improve rating in computer shogi. After conducting concrete implementation and confirming its effect, we showed effectiveness in the World Computer Shogi Championship and so on.

Keywords: Computer Shogi, Tree search, Search time

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きたが、近年は人間の最上位者を上回る実力を身につけてきたことが明らかになっている。しかしながら、新しい多くの試みが常に導入されコンピュータ同士の対戦においてその強さというものは年々向上の一途を辿っている。著者は一昨年秋の電王トーナメントにおいて準優勝した *shotgun*、昨年春の世界コンピュータ選手権において優勝した *Hefeweizen* および今年春の世界コンピュータ選手権において準優勝した *Kristallweizen* において、過去にない斬新な手法で探索時間を制御することにより時間的優勢を築く手法を実装しその勝敗に対する有効性を示した。本発表はその手法導入における一段階を詳細に示すものである。コンピュータ将棋ソフトの名称は *shotgun* がアメリカンフットボールにおけるショットガン攻撃に由来しており相手ディフェンスに対して攻撃の手段を複数事前準備を行うパス攻撃を意味している。*Hefeweizen* はドイツ語で酵母入りビールのことであり、さわやかな口当たりと共に先を見通せない不透明感を意味している。*Kristallweizen* は酵母を濾過し透明度を上げたものである。

まず、コンピュータ将棋において探索時間をどのように使うかは非常に重要な問題であると共にその適切な解は全くと言って得られていない。[1] 木探索において探索時間はその探索深さに対応し、探索が有効に働くためにその前

後の手も同様の深さの探索を行う必要があると考えられていた。この仮説によりコンピュータ将棋においては終局手数を見込んで、持ち時間をその手数で除算を行うことにより一手当たりの平均的な消費時間と概算を行うのが通例である。具体的には10分の持ち時間で100手で終了予定であれば片方が指す手は50手であるから、平均一手12秒と算出される。しかしながら、プロ棋士の対局などで明らかのように人間の将棋の約半数はほぼノータイムで指される。本手法の発案の原点はここにある。つまり、消費時間があまり必要でない局面が存在し、その局面においては比較的浅い探索結果でも問題ないということである。もちろん、これにより自分の持ち時間が後半に残すことが可能となる。中盤や終盤で時間を残すことが勝率向上に繋がることは想像に難くないが、実際に多くの試行で確認されている。[2]

2. 時間と探索の進行

本モデルでは現在 *Bonanza*[3]登場以来主流である $\alpha\beta$ 探索を元にした全幅探索を反復深化させるアルゴリズムのプログラムを対象とする。計測に用いたものは技巧2および *Kristallweizen* である。図1に技巧2による本将棋初期局面における探索深さと探索ノード数の関係を示す。探索ノード数は枝狩りが無い場合、探索枝の階乗で増加すると考えら、一般的には将棋では70~90とされている。*Bonanza* においてはこれを大幅に枝狩りすることにより2.3~2.8に抑えることができたとされる。技巧2においては図1の1スレッドのプロットを近似することによりその係数を約1.6であると算出した。*Kristallweizen* においても同様である。

次に近年一般的になってきた多コアのCPUにおける探

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

索速度の問題である。従来木探索を行うアルゴリズムは多コアに向いていないとされており歴史的には Young Brothers Wait Concept(YBWC)や Lazy SMP などの手法が実用化されているが2スレッドで2倍の探索速度が得られるわけではない。スレッドが増えるほど無駄が多いと言える。図1においてスレッド数が多いものが探索ノードが増えているがこれは Lazy SMP のアルゴリズム上多くのスレッドが独立的に探索を行うためである。16スレッドにおいて5~6倍程度の探索ノード数となっている。

次に図2に探索深さと探索時間の関係を示す。本実験では負荷増による CPU クロックの低下が起これないように気を配ったため凡そ単位時間当たりの探索ノード数は安定

されていて初期設定時間や高スレッド数でのメモリバスのボトルネックなどは考えられる。探索深さと探索ノード数の関係では比較的对数関数に近似されていたが、探索時間は初期設定時間を含むために0付近で増加がみられる。特にスレッド数が多い場合に初期時間がとられることが分かる。探索深さが進むにつれ傾きが一致していく様子が見られる。16スレッドにおいて探索の高速化率を算出したところ2.5~3倍程度の高速化が行われていた。探索ノード数の増加率と積算すると15倍程度とスレッド数にほぼ等しくなる。つまり15倍の演算能力のうち5~6倍分を無駄にし、2.5~3倍の高速化が達成できていると考えられる。

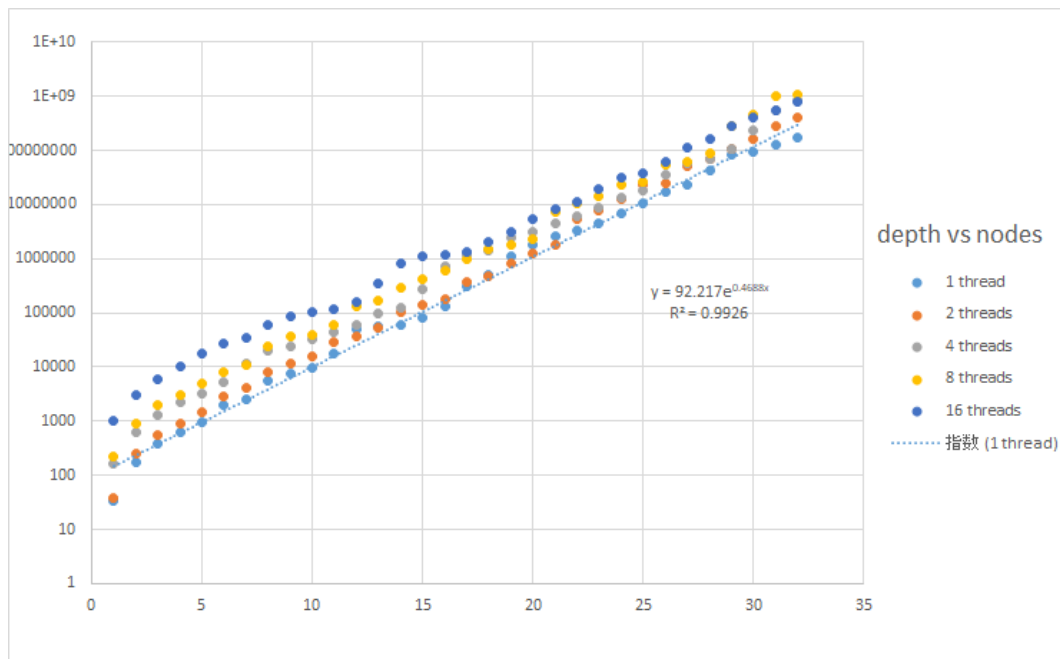


図1 技巧2における探索深さと探索ノード数の関係

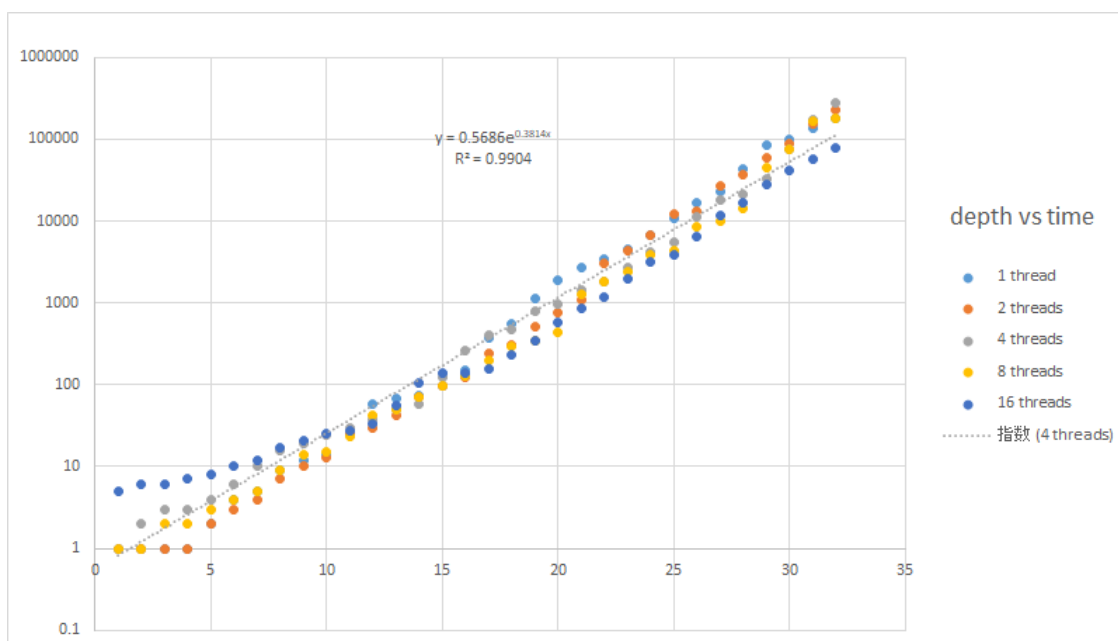


図2 技巧2における探索深さと探索時間の関係

3. Multi Ponder

著者はコンピュータ将棋の対戦で Ponder の問題が軽んじられていると感じてきた。Ponder とは対戦時の相手の考慮時間において相手の指し手を予想し仮に置くものである。これにより予想手から先の局面において自分の指し手の考慮に入ることが可能となる。Ponder が当たった場合はこれにより考慮時間が有効になるが、当たらなかった場合は全く無駄となり自分の手番の時間から考慮を始めることとなる。事実 Ponder のヒットが対戦時の勝敗に多く寄与することが大会での上位者は経験的に感じているようである。しかしながら、準備段階で十分にこれらについて考察されている開発者は皆無であった。理由は以下のようなものである。対戦の試行を行う際にはほぼ大半の開発者が一台の計算機に二つのエンジンを立ち上げ、交互に探索を行うことにより指し手を進めている。多コアのマシンにおいてはこれを複数並列で行っている例も少なくはない。当然これらの対戦において Ponder は全く考慮されておらず、疎かになっているという程度ではない。計算機リソースに対する対戦数を考えると多くの開発者にとって Ponder の考慮は後回しになるようである。

著者は本件に別の視点を与えてみた。Ponder を複数設定した状況を想定する。つまり、演算能力を分散させた投機的実行である。この場合、相手の考慮時間において複数の探索を行いヒットする可能性は飛躍的に増大する。大半の手がヒットするような状況では非常に時間的優位に立てるのではないかと考えた。

前述のような評価より 4 スレッドの探索は 1 スレッドの探索に比べて 2 倍の速度が出ていない。つまり、概念的には 4 択で必中に近い状態を仮定すると相手考慮時間内に探索の半分強が完了することになる。加えて前述の探索深さと探索時間の関係よりその相手考慮時間内の 1 スレッドの探索は相手の探索深さと同等か 1 手浅い程度であることが分かる。

前回の報告において 5 択で 90% 以上の的中率を誇る手法を提案している。[4] 図 3 に予想候補手 8 手の場合の的中率確率密度を示す。手番は 60 手目のみを抽出している。図より技巧 2 の第一候補(PV1)は 50% 以上の確率で指し手を的中させていることが分かる。また、PV0 は 8 候補で的中しなかったばあいであるが、技巧 2 および技巧 2+ では 5% 未満である。

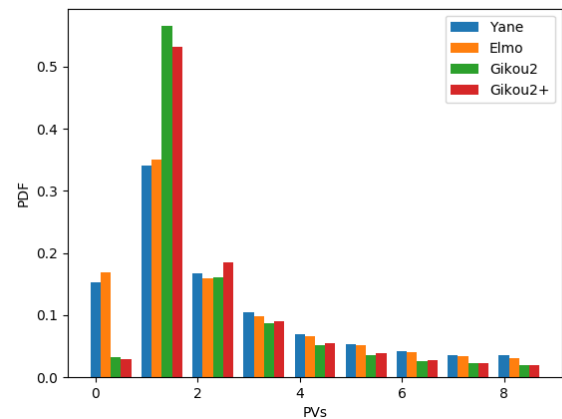


図 3 予想候補手と確率密度 (60 手目)

4. 戦果について

以上の Multi Ponder を用いた考慮時間モデルを用いた結果を実戦で示す。本来は再現性のあるデータ収集が必要であるが既知の探索エンジンでは予想手の的中が容易であるための中の見込みが全く分からない未知の将棋エンジンと対戦する必要がある。色々悩ましい問題であるが御容赦頂きたい。

まず、昨年秋の第 5 回電王トーナメントにおいては準優勝という結果のみならず、対局の中盤において相手の二倍以上の持ち時間を残すことに成功している。特に持ち時間の三分の一も消費せずに勝利するケースが少なくない。[5]

次に、floodgate において覆面エンジンとして開発途中のものを特殊な設定で対戦させてみた。[6] この対戦においては自身の考慮時間を一切使用しないという条件設定である。具体的には通信遅延などがあるためサーバ側で 1 秒未満が切り捨てられており、全指し手 1 秒未満であるということである。Multi Ponder から外れた場合は 0.2 秒で指す設定になっており記録上 0 秒である。対戦相手は floodgate の歴代のエンジンの中でも片手内に入るレーティング 4200 以上の強豪であったが、結果考慮時間 0 のまま勝利をあげることができた。Multi Ponder の概念が無い場合信じられない成果であることが分かる。

最後に、第 28 回および第 29 回世界コンピュータ将棋選手権である。[7] 第 28 回の一次予選においては上記の考慮時間 0 のモデルで対戦を行った。通信遅延等も発生したため時折考慮時間が計上されているが概ね考慮時間を使うことなく戦うことができた。7 勝 1 敗の 2 位通過である。それ以降二次予選および決勝においては十分に探索を行ったが持将棋の局面を除き考慮時間を相手より 5~10 分程度は残して勝ちをあげている。第 29 回の大会においては Multi Ponder の概念を踏まえた対策を練られたソフトも多く出場していたが、本研究を踏まえた時間制御を取り入れた我々

61	5459	1	1
62	8316	1	1
63	5905	1	1
64	6590	1	1
65	6961	1	1
66	2786	1	1
67	3495	1	1
68	15308	1	1
69	14987	1	1
70	12060	1	1
71	33334	1	1
72	23104	1	1
73	10887	1	1
74	23578	1	1
75	23573	2	1
76	12445	1	1
77	28763	1	1
78	8502	19	6119
79	14027	1	1
80	12055	1	1
81	19548	1	1
82	53786	1	1
83	26481	4	2
84	31017	1	1
85	9865	1	1
86	9904	1	1
87	5305	1	1
88	51074	1	1
89	46207	1	1
90	33179	1	1
91	33548	16	3394
92	33041	1	1
93	42395	1	1
94	53123	1	1
95	27041	1	1
96	90038	1	1
97	32102	1	1
98	39377	1	1
99	73145	18	41995
100	72216	1	1
101	16292	1	1
102	12369	1	1
103	19464	1	1
104	119900	1	1
105	80847	2	3