オセロ求解に向けた単純な縦型探索をベースにする探索方法の研究

森田悠樹[†1] 橋本剛[†2] 小林康幸[†1]

コンピュータゲームプレイヤの研究ではゲームの解明も盛んに行われており, 近年ではチェッカーが解かれて大きな話題となった. 次のターゲットはオセロが有力である. 証明数系の探索がオセロにも有効で特に WPNS が良い結果を出すことが報告されている.

本稿では単純な 2 種類のαβ探索と WPNS の比較実験を行った. パブリックドローと呼ばれる定石から作った残り 19 から 25 石の難解な局面を使っている. その結果, ソートを行わないαβ探索の性能はかなり落ちるものの子ノードの数でソートしたαβ探索は WPNS と同等の性能を出すことが示された. また WPNS の探索における弱証明数計算の占める割合のデータを示した. それによりオセロ完全探索に向けた今後のより難解な局面の探索で WPNS など証明数系探索が主役になりにくいことが予想された.

また新たに分枝数を閾値とするαβ法を提案し実験を行った.

A research of search methods based on simple depth-first search towards solving the game of othello

Yuki Morita,[†1] Tsuyoshi Hashimoto[†2] and Yasuyuki Kobayashi[†1]

Solving games are one of the hottest topics in the field of computer game player research. Solved the game of Checkers attracted tremendous interest recently. It is generally understood that next target is Othello. Search techniques like the proof-number search are effective and the WPNS gets especially good results are reported.

This paper performed comparative experiments between two kinds of simple alphabeta search and the WPNS. Abstruse positions from 19 to 25 stones to put generated from a opening book called the Public Draw are used. The results show that no sort alphabeta search are much inferior to others however alphabeta search sorted by numbers of child nodes performed equivalent quality to that of WPNS. Proportion of weak-proof-number calculation to total search time are also shown by experimental data. By the results, search techniques like proof-number search as WPNS are unexpected to play a leading role to solve more abstruse positions towards perfect search of Othello.

We also proposed a new alphabeta search using branching number as threshold and performed experiments.

1. はじめに

コンピュータゲームプレイヤは, 人工知能の黎明期から盛んに研究されている. 計算能力の向上や最新の手法を通して, いくつかのコンピュータエージェントの技術はオセロやチェスなどの複雑なゲームにおいて最強の人間の実力を上回るなど成功を収めている. またゲームの解明も盛んに行われており, 6x6 オセロの解明[1] などが有名だが, 証明数探索[2] の出現により五目並べ[3] や連珠, Qubic[4] などいくつかの有名なゲームは完全に解決された. 近年ではチェッカーが解かれて大きな話題となったが[5], ゲームの人気, 知名度, 複雑さを考えると次のターゲットはオセロが有力である.

チェッカーの解明では証明数探索（pns[2] と dfpn[6]）とαβ探索が使われた. 詰め将棋などでの成功[7],[6] からも詰みなどで突然終わるゲームに証明数系の探索が有効であることは知られているが, 決まった深さまで探索が必要なオセロでも意外に証明数系の探索が有効であることと 2 重カウント問題を引き起こす合流が多いことが報告されており, これに対応するために WPNS と呼ぶ手法も開発された[9]. それによるとオセロでは残り 10 石程度から証明数系の探索がαβ探索の性能

†1 島根大学大学院総合理工学研究科
Graduate School of Science and Engineering, University of Shimane
†2 松江工業高等専門学校情報工学科
Department of Information Engineering, College of Technology of Matsue

を上回っている．だが 8) では, 残り 20 数石以上からノード選択のための証明数計算に時間がかかり過ぎて性能がかなり落ちてしまう事が報告されており, オセロの完全解析に証明数系の探索を使うためには新たな工夫が必要となる．

本稿では, オセロ完全解析を目標とし, 証明数探索の良さを残しつつノード選択に多くの時間を消費しない新しい探索法の提案を行う．ノード選択に多く時間を割かないためには, dfpn などよりもシンプルな縦型探索にする方法が適すと思われる．オセロ等二人零和完全情報ゲームの基本はαβ法であるので, ここではαβ法による求解性能を検証し, 証明数系でオセロにおいて最も性能が良いとされる WPNS と比較を行う．

まずは 2 章で WPNS を紹介する．

次に 3 章で 2 種類のαβ探索（子ノード数によるソートの有無）と WPNS の性能比較実験を行う．

4 章で新たに分枝数を閾値とするαβ法を提案する．

2. WPNS (Weak Proof Number Search)

証明数は AND ノードで（反証数は OR ノードで）子ノードの証明数（反証数）をすべて足すため, 異なるパスから局面の合流があると値を過剰に高くしてしまう 2 重カウントが問題となる．将棋やチェスだけでなく, オセロでも合流が多く発生し 2 重カウントが問題となることが報告されている.[10)]

そこで, 過剰見積もりの原因となる AND ノードで証明数をすべて足す方式ではく, 証明数よりも少し「弱い」値, 弱証明数（Weak Proof Number）が提案され, これを用いた探索 Weak Proof Number Search (WPNS) がオセロや詰め将棋の実験で df-pn や経路分枝数探索[12)] を上回る結果を出した[9)]．表 1 に弱証明数の定義を示す．本稿では証明数系探索の代表としてこの WPNS を実験の比較対照にする．

3. αβ法と WPNS の比較検証

αβ法による探索について実験により検証を行う．勝ち負けの読み切りが目的なので, 深さに制限を設けず必ず終端局面まで読ませる．評価関数は使わず, 反復深化も行わない．トランスポジションテーブルも使わない．

勝敗の結論が早めに出ることを期待して, ノード選択の際は合法手が一番少ない子ノードを選ぶことにした．以下この方法を子ノードソートαβ法（ソートαβ）と呼ぶ．ノード選択に関しては必ずヒューリスティック値最小のノードを選び終端局面まで深さ優先で探索するため, 山登り法に近いイメージとなる．それにより単純な手法ではあるが証明数探索と同様に証明木の小さいノードが選ばれやすくなると予想される．

また, 比較のためソートをまったく行わないαβ探索（無ソートαβ）の実験も行う．実験に使ったプログラムは盤面左上より順に手を生成しているので, その順で探索が行われる．

3.1 実　　験

オセロプログラムを作成しソートαβ,WPNS, 無ソートαβを実装して実験し, 探索時間を比べた.WPNS プログラムは 9) と同一で, 残り 7 石の局面からはαβ探索（ソート無し）を行っている．

3.2 実 験 環 境

実験は以下の環境で行なった．

- CPU intel Core2Duo
- メモリ 2GB
- 言語 C++

実験は残り手数 19,21,23,25 手の局面（図 1〜6）を用意し, それぞれの探索時間を調べた．これらの局面はパブリックドロー[13)]（おそらくオセロの結論ではないかと予想されている定石手順）の局面から Zebra（世界最強といわれるオセロプログラム）[14)] 同士を対戦させて得られたかなり難解と思われる局面である．

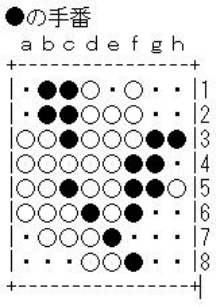

図 1 残り 19 石局面
Fig. 1 A position of 19 stones to put

3.3 実 験 結 果

実験結果を表 2 に載せる．探索は 24 時間経過したら打ち切っている．

結果として, ソートαβと無ソートαβでは探索時間に大きな差が出た. 全ての局面において無ソートαβより探索時間が早い. 同じαβ法でも子ノードの選び方で探索時間がかなり変わることがわかる．WPNS とソートαβとの比較であるが, こちらは局面によって WPNS が早い場合とソートαβが早い場合があり, ここの例では両者はおおまかに言って同程度の性能を

表 1 弱証明数（Weak Proof Number）の定義
Table 1 Definition of the Weak Proof Number

節点の種類	弱証明数	弱反証数
先端 勝ち	0	∞
負け	∞	0
引分け	1	1
内部 AND ノード	MIN（子接点の弱証明数）	MAX（子接点の反弱証数）＋ (子接点の数) -1
OR ノード	MAX（子接点の反弱証数）＋ (子接点の数) -1	MIN（子接点の弱証明数）

表 2 探索時間の比較
Table 2 Comparison of Search time

		time		
図 No.	残りマス	無ソートαβ	WPNS	ソートαβ
1	19	52 秒	8 秒	3 秒
2	21	32 分 34 秒	43 秒	55 秒
3	23	11 時間 14 分	7 分 01 秒	11 分 5 秒
4	23	2 時間 20 分	8 分 12 秒	6 分 22 秒
5	25	24 時間以上	32 分 8 秒	41 分 56 秒
6	25	24 時間以上	24 時間以上	1 時間 17 分

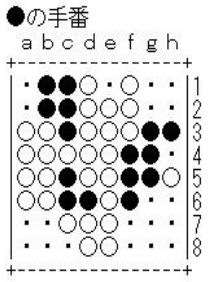

図 2 残り 21 石局面
Fig. 2 A position of 21 stones to put

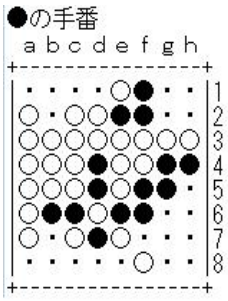

図 4 残り 23 石局面 2
Fig. 4 A position of 23 stones to put #2

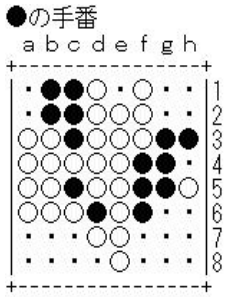

図 3 残り 23 石局面 1
Fig. 3 A position of 23 stones to put #1

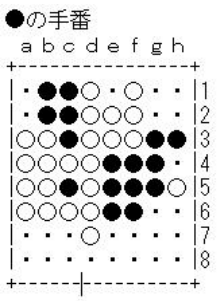

図 5 残り 25 石局面 1
Fig. 5 A position of 25 stones to put #1

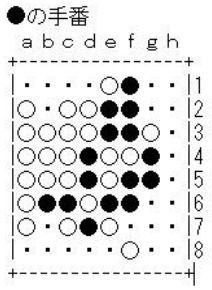

図 6 残り 25 石局面 2
Fig. 6 A position of 25 stones to put #2

示している.

3.4 WPNS ノード選択の計算時間

文献 8) では証明数系探索だと残り石が多く難しい局面になるほど子ノード選択に費やす計算コストの割合が大きくなり過ぎると推察されているが, 具体的なデータはない. ここでは WPNS が子ノード選択の計算にどのくらい時間をかけているか調べた. df-pn と同様に WPNS も全子ノードの弱証明数と弱反証数をトランスポジションテーブルから取得して局面の弱証明数, 弱反証数と閾値を計算し, 次に探索する子ノードを決定して探索を繰り返している. この子ノード決定までの計算時間を証明数計算時間として計測し, 全探索時間における割合を調べる. 残りマス 7 以下の局面からはαβ探索を行っているので, 子ノード決定までの計算時間以外で大きな割合を占めているのは末端のαβ探索であると考えられる. 結果を 3 に示す.

表 3 総探索時間と証明数計算時間の比較
Table 3 Comparison of total search time and proof-number calculation time

局面 No.	探索時間	計算時間
1	8 秒	2 秒
2	43 秒	15 秒
3	7 分 1 秒	3 分 12 秒
4	8 分 12 秒	4 分 6 秒
5	32 分 8 秒	15 分

図 6 は WPNS で 24 時間以内に解を得られなかったので除外した. 深さが深くなるほど, ノード選択の計算時間が増えることがわかる. 残りマス 20 手あたりから, 計算時間に探索時間の約半分の時間を費やしている. 深さが深くなるとノード数が増えるのでその分計算時間も増えると考えられる. 時間がかかり過ぎるため今回計測できなかったが, 残りマスが増えてさらに難しくなるほど証明数計算に時間がかかり求解が困難になると予想される. オセロ完全探索のためにはノード選択のための計算が簡単な方法にすることが必要だろう.

3.5 考　　察

WPNS は同じ局面の実験で dfpn や BNS を圧倒している[9] が, ソートαβは単純な方法であるにも関わらず残り 19〜25 手で WPNS と同程度の性能を出した. 11) によると残り 10 手程度まではαβ法が早いものの, 残り 10 石以上だと df-pn や WPNS が早くなるとされている. 確かにこの結果からは無ソートαβ法より WPNS の方が早いが, 子ノードの選び方によって, 残り 19 石以上あたりからαβ法の性能が WPNS などを上回る可能性が示された.

またオセロでは残り石数が多くなると証明数系の探索ではノード選択のための計算にかかる時間の割合が増大することが示された. オセロ完全探索に向けてさらに残りマスの多い局面を探索する際, 証明数計算の時間は末端局面までの距離に対して指数的に増大すると思われるので, WPNS など証明数系の探索よりはαβ探索の方が早く解に達するだろうと予想される.

だが, 残り 10 石以上で df-pn や WPNS の性能が良くなったように, 証明数探索のような工夫をαβ探索に加えることで性能が良くなる可能性は大いに考えられる. 次章ではαβ探索をベースに証明数的な工夫を加える方法について考える.

4. 分枝数を閾値とするαβ法

WPNS では証明数計算に時間を費やすことになり完全解に向けた主役とはなりにくいことがわかった. ソートαβは単純なため, たまたま選ばれたノードが結論の出にくい難しいノードでも結論がでるまで探索することになる. しかし, ソートαβは残り 19〜25 石で WPNS と同等の性能を発揮している. よってここではαβ法をベースにより証明数探索に近い新たな探索方法を考える.

証明数系の探索はわかりやすい局面を先に深く調べ, 難しそうな局面の探索を後回しにすることで詰め将棋などの分野で成功を収めている. 通常のαβ探索では探索深さを閾値とするが, これと似たイメージの探索を行うには閾値を探索深さ以外にすることが考えられる. 探索深さ以外を閾値に使うαβ探索としては局面の実現確率を閾値に使う実現確率探索[15] が有名である. ここではソートαβに似たふるまいがうまく働くことを期待して分枝の数が少ないノードを優先させるために分枝数を閾値とする手法を提案する. 分枝数を探索に使う手法には経路分枝数探索[12] があるが, この経路分枝数の最小ノードを探索するのではなくαβ探

索の閾値に使うのである.

その実装方法を説明する. 経路分枝数探索と同様にAND 分枝数と OR 分枝数を独立して計算する. 経路分枝数探索ではそれぞれが証明数, 反証数に相当する. 以下簡単のため AND 分枝数について説明する. ルートから分枝数を子に引き継いでいくが, OR ノードでは値をそのまま子に引き継ぐ. AND ノードでは選択した子ノード以外の分枝の数, すなわち子ノード数 -1 を閾値から引く.

図 7 は AND 分枝閾値計算の具体例である. 一番上のノード閾値が 5 であるとする. 一番左の子ノード（黒塗り）を選択していると, 他に分枝は 2 なので次の子ノードには $5-2=3$ が閾値として渡される. 次は OR ノードなので閾値 3 がそのまま渡され, 次の選択子ノードには $3-1=2$ が渡される. このようにして閾値が 0 になるまで縦に探索が行われる. 同様にして OR 分枝閾値も計算する.

ルートの閾値は反復深化で増やしていく.

以上が分枝数を閾値とするαβ法の概要である.

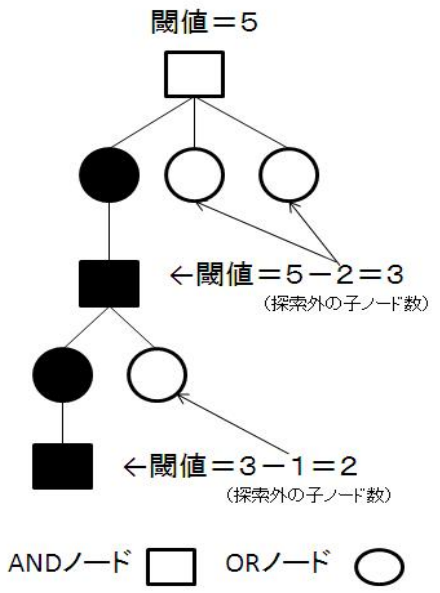

図 7 AND 分枝閾値計算例
Fig. 7 An example of AND branching threshold calculation

5. 実　　験

3.3 章と同じ局面で実験を行ったが, 残念ながら良い結果は出なかった. 残り 19 石の図 1 を解かせても一回の反復深化で時間がかかりすぎてしまった. 閾値の初期値を手動で決めて解かせると解は得られるが, 初期値が大きすぎると普通のαβと同じ結果になり, 適当に選んでも普通のαβより良くなることはなかった.

6. ま と め

WPNS と子ノードの数でソートするαβ, ソートしないαβの 3 種類でオセロ局面を解く実験を行った. パブリックドローから作った難解な残り 19〜25 石の局面でソートαβが WPNS と同等の性能を示した. また WPNS の全探索における証明数計算の占める割合を調べた結果, 証明数計算のコストが残り石が増える難しい局面ほど増えることが示された. 以上より, オセロ完全探索に向けて今後より残り石の多い局面を探索させる際には WPNS など証明数系の探索手法が主役になることは難しく, αβ法をベースにした手法が有力になると予想される.

また分枝数を閾値とするαβ法の提案を行ったが, 良い結果を得られなかった.

7. 今後の予定

まず分枝数を閾値とするαβ法がなぜうまくいかなかったか解析を行う予定である. まだバグの可能性もあり, 詳しい検証次第では良い結果をもたらす可能性もあると考えている. 今回は分枝数を閾値としたが, オセロでは経路分枝数探索ではなく WPNS が良い性能を示したので WPNS と似た方式のαβ探索を考案して試してみたい.

詰め将棋など証明数系の探索が得意とする分野で分枝数を閾値とするαβ法がどのような結果を示すかも実験を行う予定である.

また, オセロにはすでに Zebra[14] などオープンソースの強いプログラムが存在するので, それらをうまく使って今回得られた知見も活かしより強力なソルバーの作成にもチャレンジしてみたい.

参 考 文 献

1) Joel Feinstein: Perfect play in 6 × 6 Othello from two alternative starting positions, `http://www.feinst.demon.co.uk/Othello/6x6sol.html`
2) Allis,L.V.,van der Meulen,M.,and van den Herik,H.J.: Proof-number search, Artificial Intelligence,Vol.66,Issue.1,pp. 91-124, 1994.
3) Allis,L.V.,van den Herik,M.P.H. Huntjens: Go-Moku solved by new search techniques, Comput. Intelligence：An Internet. J. 12（1）pp.7-24,1995.

4) Allis,L.V.,P.N.A. Schoo: Qubic solved again, in：H.J. van den Herik, Allis, L.V.(Eds.), Heuristic Programming in Artificial Intelligence 3, The Third Computer Olympiad, Ellis Horwood, Chichester, pp.192-204,1992.
5) Schaeffer, J., Bjornsson, Y., Burch, N., Kishimoto, A., Muller, M., Lake, R., Lu, P., and Sutphen, S.: Checkers Is Solved, Science, 10.1126/science.1144079, published online July 19, 2007.
6) 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, 43(6), pp.1769-1777, 2002.
7) Seo, M., Iida, H., Uiterwijk, J.W.H.M.: The PN*-search algorithm: Application to tsume-shogi. Artificial Intelligence 129(4), pp.253-277, 2001.
8) 橋本 剛: オセロ完全解析を目指して, LA シンポジウム会誌, LA シンポジウム, Vol.51, pp.19-24, 2008.
9) Toru Ueda, Tsuyoshi Hashimoto, Junichi Hashimoto and Hiroyuki Iida: Weak Proof-Number Search, Lecture Notes in Computer Science, Proceedings of CG2008, Springer, pp.157-168, 2008.
10) 上田徹, 橋本剛, 橋本隼一: オセロの定石読切りと問題点に関する考察, 第 12 回 ゲームプログラミングワークショップ, pp.132-135, 2007.
11) 上田徹: DAG を有する問題を解くための超効率的 AND/OR 木探索アルゴリズム, 北陸先端科学技術大学院大学修士論文, 2008.
12) 岡部文洋: 経路分枝数を用いた詰将棋解図について, 第 10 回 ゲームプログラミングワークショップ, pp.9-16, 2005.
13) 奥原 俊彦: パブリックドロー全変化, `http://www.amy.hi-ho.ne.jp/okuhara/pubdraw.htm`
14) Gunnar, A.: ZEBRA, `http://radagast.se/othello/`
15) Yoshimasa Tsuruoka, Daisaku Yokoyama and Takashi Chikayama: Game-tree Search Algorithm based on Realization Probability, ICGA Journal, Vol. 25, No. 3, pp. 145-152, 2002.