

# 経験的知識を用いた証明数探索の効率化

島野 拓也<sup>1,a)</sup> 金子 知適<sup>1</sup>

**概要:** 本稿では、詰碁に関する問題の探索を効率化するために証明数探索に過去の探索で得た経験的知識を問題を検く際に活用する手法を提案する。証明数探索と過去の経験的知識を用いた RAVE (Rapid Action Value Estimate) のそれぞれは、囲碁の研究で有効とされている。これらを組み合わせることでより効率的な探索が可能になると考えられる。今回の提案手法は、まず各場所に着手した回数と攻めが成功した回数を記録する。そして、次に試すときに過去に攻めが成功した回数の多い順に優先して探索を行う方法である。アルゴリズムの評価方法として、証明数探索が成功している問題を用いて、オリジナルの証明数探索と提案手法の比較実験を行った。

## Enhancement of Proof Number Search by Using Empirical Knowledge

SHIMANO TAKUYA<sup>1,a)</sup> KANEKO TOMOYUKI<sup>1</sup>

**Abstract:** In this paper, we propose a method of proof number search by using empirical knowledge in order to improve the efficiency in solving problems about Tsumego. Proof number search and RAVE (Rapid Action Value Estimate) using past empirical knowledge have individually been recognized as effective methods in the study of Go. It is expected that the combination of these method is also effective. We propose an enhancement of proof number search by adding more preference on more promising move estimated by the frequency of the destination of the moves as well as the probability of successful attack. To show the effectiveness of the proposed algorithm, we conducted experiments on capturing problems. The results show that the proposed method solved problems in smaller number of nodes than those needed by the original proof number search, on average.

### 1. はじめに

近年、コンピュータによる詰将棋解答プログラムの研究は非常に進んでおり、ほとんどの問題を高速に解くことができるようになった [1]。一方、詰碁は解けない問題も多く、詰将棋と比べると発展途上である。

コンピュータが詰碁を解く上で難しい点は、探索空間が大きいことや人間の感覚的な評価が強いため評価関数の作成が難解であることが挙げられる。これらの問題を解決し、探索の効率化を図ることが発展には必要不可欠である。だから、本稿では詰碁に関する問題の探索の効率化に着目した。

詰将棋や詰碁などで成功を挙げている探索手法の一つと

して証明数や反証数という概念を利用した探索アルゴリズムがある [1]。証明数の概念と証明数探索は Allis [2] が導入して、それを進化させた df-pn [3], df-pn<sup>+</sup> [4],  $\lambda$  df-pn [5] は詰碁でも成果を上げている。

しかし、囲碁における石の死活問題は複雑で、現実的な時間で解くことが難しい問題が数多く存在する。この問題を解決するために過去の経験的知識を利用することが有効な手段の一つであると予想される。その理由として、囲碁における UCT の RAVE [6] やチェスのヒストリーヒューリスティック [7] のように過去の経験的知識を利用した手法が探索の効率化において成功した事例があるからである。このことから、石の死活問題に対しても同じ手法が有効な可能性が高く、オリジナルの証明数探索と組み合わせることによって、探索の効率化が可能であると予想できる。

本稿では、証明数探索に過去の経験的知識というヒューリスティックな評価を導入し、有望そうな候補手を優先的

<sup>1</sup> 東京大学大学院総合文化研究科  
Graduate School of Arts and Sciences, The University of Tokyo

<sup>a)</sup> shimano@graco.c.u-tokyo.ac.jp

に探索するような手法を提案した. そして, 提案手法が有効であるのかをオリジナルの証明数探索と比較実験を行うことで検証した.

## 2. 関連研究

### 2.1 証明数探索

効率のよい探索をするために, 証明数探索では証明数と反証数という概念を用いて, それを各ノードの評価値とする. 証明数とは, 各ノードにおいて, そのノードが詰む局面であることを証明するために展開しなければならない子ノードの最小の数, 反証数は, 各ノードにおいて, そのノードが詰まない局面であることを証明するために展開しなければならない子ノードの最小の数を意味する. 具体的な証明数と反証数の定義を (1), (2) に示す. ここで, ノード  $n$  の証明数と反証数はそれぞれ  $pn(n), dn(n)$  と記す [8].

(1) ノード  $n$  が先端ノード

(a) 最終的な評価値が詰み

- $pn(n) = 0$
- $dn(n) = \infty$

(b) 最終的な評価値が不詰み

- $pn(n) = \infty$
- $dn(n) = 0$

(c) 最終的な評価値が不明

- $pn(n) = 1$
- $dn(n) = 1$

(2) ノード  $n$  が内部ノード

(a)  $n$  が OR ノード

- $pn(n) =$  子ノードの  $pn$  の最小値
- $dn(n) =$  子ノードの  $dn$  の和

(b)  $n$  が AND ノード

- $pn(n) =$  子ノードの  $pn$  の和
- $dn(n) =$  子ノードの  $dn$  の最小値

この証明数と反証数は, 次に展開するノードを決めるために用いられる. つまり, ルートノードから攻め側は証明数の最小のノードを, 受け側は反証数の最小のノードを辿り, 末端ノードに着いたらそのノードを展開し, 木全体の証明数と反証数の値を更新する. そして, ルートノードの証明数または反証数の値が 0 となるまで繰り返す探索アルゴリズムである. 攻め側は証明数が最小の手, 受け側は反証数が最小の手を優先的に探索することで, 探索の効率化が可能となる.

### 2.2 RAVE

RAVE (Rapid Action Value Estimate) [6] とは, UCT のプレイアウト中に選択された着手全てを現在の局面から一手目に着手されたものとする. そして, 現在の局面からその着手した全ての場所によって変化する全ての局面に対して勝敗を更新する手法である. 通常の UCT では, プレイアウトで得られた勝敗はその局面のノードのみを更新するに對

して, RAVE は, プレイアウトで着手した場所の分だけ勝敗を更新する.

### 2.3 ヒストリーヒューリスティック

ヒストリーヒューリスティック [7] は, コンピュータ将棋やコンピュータチェスにおいてよく用いられる手法の一つで, 着手の場所と評価を記録しておき, 探索の際にある着手が最善手であった割合 (回数) を示す. 囲碁では, 似た手法として UCT の RAVE が成功を収めている.

## 3. 提案手法

本稿の目的は, 有望そうな候補手を絞り優先的に探索することで, 探索の効率化を図ることである. ここでは, 詰碁で成果を挙げている証明数探索という探索アルゴリズムを利用することにする. そして, ヒューリスティックな工夫として, 証明数を計算する際に過去の経験的知識を導入する. 探索の際に各座標に着手した回数と, 攻めが成功した回数を記録しておき, 次に探索を行うときに, 過去に成功した回数の多いものを優先的に選ぶようにする手法である. まず, 攻め側のノード  $n$  の成功率は次のように定義する.

$$SuccessProbability(n) = \frac{SuccessTimes(n)}{MoveTimes(n)} \quad (1)$$

次に式 (1) の成功率から攻めの失敗率を次のように定義する.

$$FailureProbability(n) = 1 - SuccessProbability(n) \quad (2)$$

そして, 攻め側のノード  $n$  の証明数 ( $pn(n)$ ) の値を次のように更新する.

$$pn(n) = pn(n) * FailureProbability(n) \quad (3)$$

後は, オリジナルの証明数探索と同様に, ルートノードから攻め側は証明数が最小のノードを, 受け側は反証数が最小のノードを探索する.

このようにすることで, 攻めが成功する確率が高いほど証明数の値は 0 に近づくので, 良い手である可能性の高い候補手が優先的に探索されると期待される.

## 4. 実験

### 4.1 実験設定

提案手法の効果を測定するために, オリジナルの囲碁プログラムを実装した. そして, 提案手法とオリジナルの証明数探索を導入し, その 2 つの手法の比較実験を行った. 本稿の目的は, 探索の効率化を調べることなので, 詰碁のルールも簡易化し, 石を先に取った方が勝ちというルールとした. 実験は三路盤と四路盤で行い, 問題はオリジナルの 40 問を使用した. 本稿では, 四路盤の結果を紹介する. 比較実験の

評価方法は、それぞれの手法の探索の際の訪問ノード数の増減と詰みまでの解答時間の長さとした。

#### 4.2 実験に使用した局面

ここでは、実験に使用した局面をいくつか紹介する。証明数探索で解ける問題に限定したので、攻め側(黒石)が受け側(白石)の石を確実に捕獲できるような局面を設定した。攻め側(黒石)が先手とする。

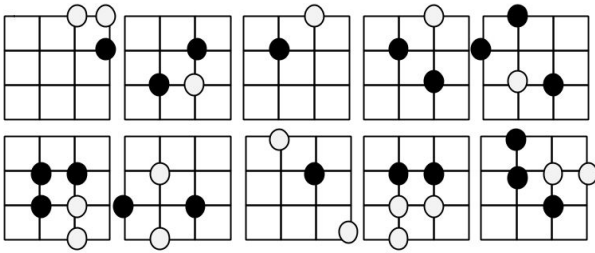


図 1 実験に使用した局面例

#### 4.3 比較実験

まず最初に、オリジナルの証明数探索と提案手法の探索の際の訪問ノード数の違いを比較した。実験結果は図 2 に示す。図 2 は、横軸にオリジナルの証明数探索のノード数を表し、縦軸に提案手法のノード数を表す。そして、問題ごとの訪問ノード数を散布図としてプロットしている。図 2 の結果より、多くの問題に対して、提案手法の方が訪問ノード数が少ないことが言える。よって、提案手法が有効な手法であると考えられる。最も効果が表れた問題では、提案手法の方が 1/2 以上の減少に成功している。しかし、逆に 5 倍以上増加してしまった問題もある。これは、問題の性質によって、提案手法が強い問題と弱い問題に別れてしまっていることが考えられる。提案手法が強い問題は、着手場所と成功率で重み付けをしているため、同じ場所で攻めが成功する分岐が多いものであると考えられる。逆に弱い問題は、どの着手場所でも同じような成功率で重みに差が生まれないようなものであると考えられる。また、手数が短い問題も成功率に差が生まれにくく、オリジナルの証明数探索とほとんど変わらない結果となった。

次に、詰みまでの解答時間の違いを比較した。実験結果は図 3 に示す。図 2 と同様に横軸と縦軸は、それぞれオリジナルの証明数探索と提案手法の解答時間をミリ秒で表す。図 3 の結果より、訪問ノード数と同様に多くの問題に対して、提案手法の方が解答時間が短いことが言える。そして、減少に成功した問題の数は訪問ノード数よりも多いことが言える。このことから、探索の際の訪問ノード数はほとんど変わらない、もしくは増加している問題でも解答時間の短縮されている問題が存在すると思われるが原因は不明であり、探索節点数と実行時間の関係は調査中である。

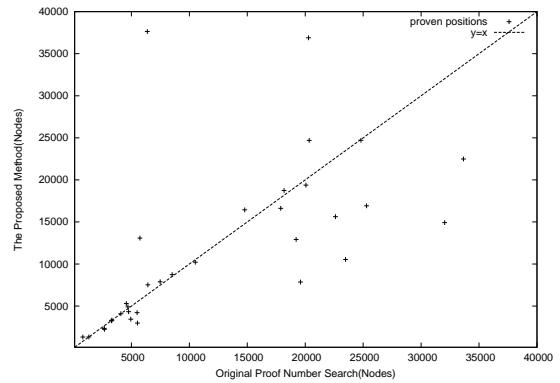


図 2 証明数探索と提案手法の訪問ノード数の比較

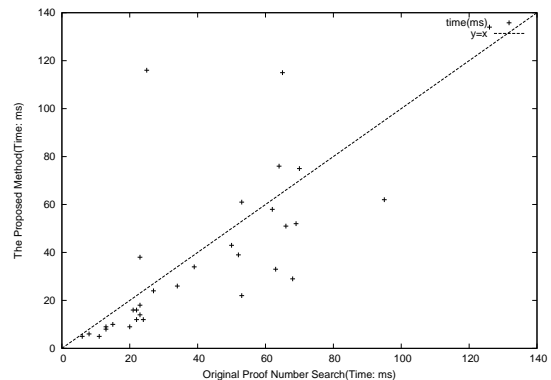


図 3 証明数探索と提案手法の解答時間の比較

これらの比較実験から、証明数探索の効率化に過去の経験的知識の利用は有効であると考えられる。しかし、探索接点数削減に失敗した問題もあるため問題の性質に依存する部分が強いと考えられる。つまり、失敗した問題の原因を究明することが必要な課題であると考えられる。

## 5. おわりに

本稿では、詰碁に関する問題を解くための探索アルゴリズム効率化を目的とし、既に詰碁の研究で成果を挙げている証明数探索と囲碁の UCT やチェスの研究で成果を挙げている過去の経験的知識に着目した。それぞれ囲碁関連の研究で成功していて、さらに経験的知識の性質は多くの探索アルゴリズムに対して、汎用性が高そうだと考え、これらを組み合わせる手法を提案した。

比較実験の結果から問題の性質によって、差はあるものの訪問ノード数と解答時間もオリジナルの証明数探索に比べ、期待通り、提案手法の方が実験の範囲では、平均して少ないことが観察された。このことから、証明数探索に対しても過去の経験的知識を利用することは有効であると考えられる。しかし、逆に訪問ノード数や解答時間が二倍以上増えてしまっている問題もあるため一概に提案手法が良いとは言い切れない。原因として、有望そうな候補手を絞る際に、最善手が含まれておらず、勝ちの手筋ではあるが、最善手に比べて非常に長くなってしまっている場合があると

いう可能性が予想される。また、勝ちの手筋が少ない問題に対しては、攻めの成功率ではほとんど差が出ないため、探索の効率化は難しいと考えられる。

今後の課題として、提案手法がどんな問題に弱いのか、その原因を詳しく調べる必要がある。また、既存の証明数探索より訪問ノード数や解答時間が長いものもあったため、解答までの手筋を調べ、最善手であるかを検証する必要がある。そして、候補手を絞る精度をさらに向上させることが探索アルゴリズムの効率化に繋がると考えられる。

今回は、詰碁に焦点を当てて、探索の効率化を図った提案手法は、攻めの成功率というどんなゲームでも存在する概念を評価値としている。つまり、囲碁に限らず多くのゲームで応用が可能であると考えられる。また、経験的知識も攻めの成功率だけではなく、様々な切り口から評価する方法があり、ゲームの性質によって変えることができ汎用的であると考えられる。これらの応用も今後の研究課題である。

#### 参考文献

- [1] A. Kishimoto and M. Mueller. Df-pn in go: An application to the one-eye problem. In *Advances in Computer Games 10*, pp. 125-141. Kluwer Academic Publishers, 2003.
- [2] L.V. Allis, M. van der Meulen, H.J. van den Herik, Proof-number search, *Artificial Intelligence* 66 (1) (1994) 91-124.
- [3] Nagai, A. and Imai, H.: Proof for the Equivalence Between Some Best-First Algorithms and Depth-First Algorithms for AND/OR Trees, *KOREA-JAPAN Joint Workshop on Algorithms and Computation*, pp. 163-170 (1999).
- [4] A.Nagai and H.Imai. Application of df-pn<sup>+</sup> to othello endgames. In *Game Programming Workshop in Japan '99*, pp. 16-23, Oct. 1999.
- [5] Yoshizoe,K., Kishimoto, A. and Miffler, M.: Lambda depth-first proof number search and its application to Go, *20th International Joint Conference on Artificial Intelligence(IJCAI-07)*, pp. 2404-2409 (2007).
- [6] S. Gelly and D. Silver. Combining online and offline knowledge in UCT. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 273-280, 2009.
- [7] Jonathan Schaeffer. The History Heuristic and Alpha-Beta Search Enhancements in Practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 11, pp. 1203-1212. ISSN 0162-8828 (1989).
- [8] 長井, 今井. 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):17691777, 2002.