

証明数を使った囲碁死活探索の効率化に関する実験

6Q-01

後藤 智章 阿部 能明 乾 伸雄 小谷 善行
 東京農工大学工学部 情報コミュニケーション工学科

1. 概要

詰将棋などにおいて、深さ優先探索に証明数を利用した探索の研究がされている。本稿では、証明数を使った探索で、囲碁の死活問題初手解答の実験を行った性能評価を示す。実験を行った探索法は、深さを閾値とした反復深化法、C*[1]、PDS[2]、df-pn[3]の4種類である。

2. 本稿の実験システム

本稿の実験は、本研究室で作成している囲碁プログラム「gochan」上で実現する。現在の「gochan」にある機能は、対局システムと連の認識である。

末端判定は、殺す側はパス以外の候補手が無くなると負け、生きる側は最初の局面での最も大きな連が打ち上げられると負けとした。一度「コウは負け」として探索した後、不詰であれば「コウは勝ち」として探索する。二回目の探索で答えが出た場合はコウとなる。コウの回数は考慮しない。候補手は、殺す側は矩形内の着手可能点、生きる側はそれとパスである。問題はあらかじめ周りを殺す側の石で埋めて矩形としたものを利用する。

深さを閾値とした反復進化法は、候補手の順番を着手可能点の中で周りのアキ数の多い順とし、生きる側のパスは最後の候補手とした。C*は証明数を、PDSは証明数と反証数を閾値として反復進化する深さ優先探索である。df-pnはPDSを改良したアルゴリズムであり、ルート局面では反復進化しないなどの特徴がある。本実験では、無限大を利用したdf-pnでのサイクルの処理の扱いが難しかったため、サイクルが存在した場合はアゲハマの増減によって末端（詰か不詰）とすることにした。しかし、この手法はループの区間の転置表に書かれる情報が信頼できず、誤る可能性がある。これは不完全な手法であるが、df-pnでのみ導入した。ハッシュキーには、手番、黒石、白石、コウ情報を用いる。転置表のアドレスはオープンアドレス法で決定する。

3. 実験

問題集とWEBから、最初から閉領域に近い形となっており、加工後の矩形内のアキ数が3~12となる問題100

題を集めて実験を行った。実験環境は、OSがWindows Me、CPUがCeleron 1.2GHzである。

100題を5分以内に正しく解けた数は、反復進化法とC*で95題、PDSとdf-pnで96題であった。解けなかった理由は、5分で解けない、死活判定法が不十分、サイクルやDAG（非循環有向グラフ）の影響などがあった。

全ての探索法で解けた問題92題での詰認識（コウであれば2回目の探索）における計算時間の順位該当数は表1のようになった。なお、精度はミリ秒である。

表1 詰認識における計算時間の順位該当数

	1位	2位	3位	4位
反復進化法	18	14	13	47
C*	83	6	3	0
PDS	17	48	24	3
df-pn	11	12	48	21

詰認識における計算時間はC*、PDS、df-pn共に反復進化法よりも速く、C*とPDSではC*の方が速かった（図1・図2）。また、PDSとdf-pnではPDSの方が若干速く、df-pnではDAGが存在すると計算時間が増大する問題が10題程度あった（図3）。

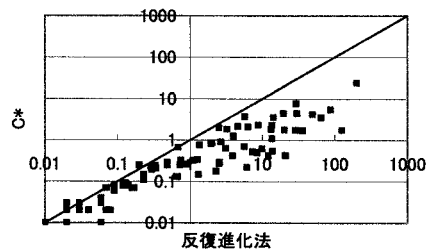


図1 詰認識における反復深化法とC*の計算時間(秒)

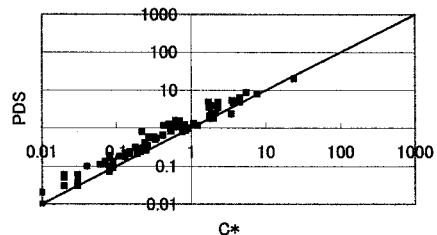


図2 詰認識におけるC*とPDSの計算時間(秒)

The Experiment about Increase in Efficiency of Life and Death of Go Game Search Using Proof Number.
 Gotoh Tomoaki, Abe Nobuharu, Inui Nobuo, Kotani Yoshiyuki.
 Tokyo University of Agriculture and Technology.

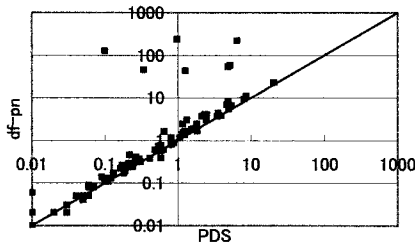


図3 詰認識におけるPDSとdf-pnの計算時間(秒)

転置表への登録数も計算時間と似たような結果となったが、C*とPDSの比較では計算時間の比較とは対照的に、PDSの方が少ないことが多かった(図4)。

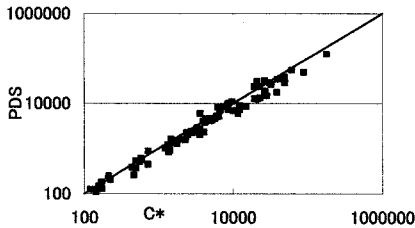


図4 詰認識におけるPDSとC*の転置表への登録数

本システムでは、コウになる問題では不詰認識を先に行う。コウであるために行った不詰認識は30題であった。不詰認識の計算時間の順位は表2のようになった。

表2 不詰認識における計算時間の順位該当数

	1位	2位	3位	4位
反復進化法	4	2	15	9
C*	2	2	12	14
PDS	24	5	0	1
df-pn	13	14	0	3

不詰認識が詰認識と異なったのは、C*と反復進化法では計算時間が同程度となった点である。ただし、転置表への登録数はC*のほうが少ない(図5)。また、C*はPDS・df-pnよりも明らかに遅かった(図6)。

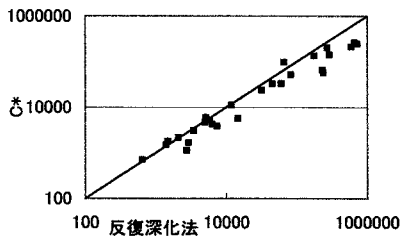


図5 不詰認識における反復進化法とC*の転置表への登録数

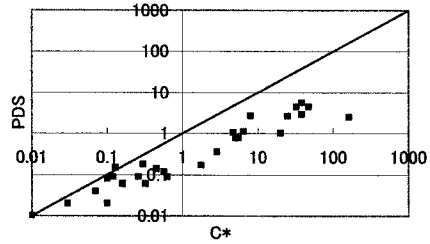


図6 不詰認識におけるPDSとC*の計算時間(秒)

C*の不詰認識はPDS・df-pnと比べて極端に遅かったため、コウが答えとなる問題での2回の探索の計算時間を合計すると、C*よりもPDS・df-pnの方が速くなった。また、コウは3割程度であったので、全体ではC*が最も高速となる問題が多いということになった。

このシステムでは、アキ数が計算量に指数関数的に比例する。アキ数と計算時間の関係を調べると、反復進化法よりも証明数を使った探索の方が、アキ数が多くなったときの計算時間の増大が少なかった(図7)。つまり、証明数が難しい問題であるほど有効であった。

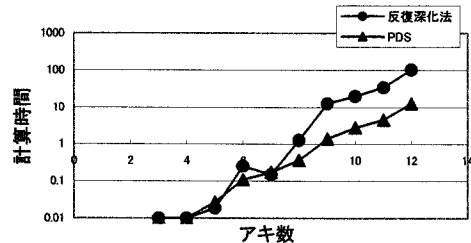


図7 アキ数による計算時間(秒)の平均値

4. 考察

実験の結果、死活問題においても証明数を利用した深さ優先探索が有効であることがわかった。PDSとC*を比べると、詰認識ではC*の方が高速に解け、不詰認識ではPDSの方が高速に解けた。実際のゲームで用いる場合は不詰認識が重要であるので、C*よりPDSの方が有効ではないかと考えている。また、df-pnを利用する場合は、サイクルやDAGを対処する工夫が必要であるだろう。

参考文献

- [1] 松原仁著:「コンピュータ将棋の進歩2」, 共立出版, 1998.
- [2] Ayumu Nagai : A new AND/OR Tree Search Algorithm Using Proof Number and Disproof Number, Complex Games Lab Workshop, pp. 40-45, 10 November 1998.
- [3] 長井歩, 今井浩:「df-pn アルゴリズムの詰将棋プログラムへの応用」, 情報処理学会 アルゴリズム75-2, 2000.