

経路分枝数を用いた詰め将棋解図について

岡部文洋

f_okabe@jcom.home.ne.jp

概要

詰め将棋等 AND/OR 節点で構成されるゲームグラフに一般化できるパズル問題を少ない探索量で解く為、ルートから展開候補となっている末端節点までの経路に付随する未解決分枝の数（経路分枝数）の利用を検討した。経路分枝数は従来詰め将棋を高速で解くにあたり利用されている証明数、反証数とよく似た挙動をしめし、かつ問題視されていた局面の合流の2重カウントの影響が極めて小さいという特徴をもつ。今回、探索打ち切りの閾値として証明数、反証数を用いたプログラムと経路分枝数を用いたほぼ同一のプログラムを作製し、解に至るまでの探索量の比較をおこなった。その結果、多くの例題で経路分枝数を用いた探索は証明数、反証数を用いたものと比較して同等以下の探索量で解に到達できることがわかった。

Application for solving tsume shogi problem

by route branch number

Fumihiko Okabe

Abstract

This paper describes the use of the number of unsolved branches that connected a route between a start node and one of unsolved end node (route branch number) under solving a puzzle, which have AND/OR node graph, like tsume shogi problem. The route branch number has behavior such as proof number and disproof number. Moreover, it does not need consideration of double count caused by Directed Acyclic Graph like a proof number and disproof number. This time it was found that the search which use Branch Number (BNS: Branch Number Search) was solved by a few searches than the search which use proof number or disproof number by an experiment.

1. はじめに

詰め将棋を解くということは問題局面を始点としてすべての末端節点を詰み局面（解決済み節点）とするひとつのAND/ORゲームグラフを完成させる行為である。探索の過程では始点と複数の、詰みあるいは不詰みとして解決された末端節点および未解決末端節点を含む未完成のAND/ORゲームグラフが生成され、不詰みの末端節点についてはもっとも近いOR節点で枝刈り、未解決の末端節点については更なる展開が繰り返されることになる。ここで展開が繰り返される未解決末端節点はすべてが詰みとして収束するわけではなく、膨大な探索の結果、最終的には不詰めとして収束する節点も多く含まれているのが一般的である。よって詰め将棋をより少ない探索量で解く為には最終的に詰みとして収束す

る可能性の高い末端節点をいかに優先的に展開していくかが鍵となっている。

近年、詰め将棋等のパズルを解くにあたって証明数や反証数の概念が導入された。[1] たとえば、解となるべきAND/ORゲームグラフが完全な木構造（ゲーム木）をしている大部分の詰め将棋の場合、証明数は探索過程においてこのAND/ORゲーム木を完成させるために最低限解決すべき未解決末端局面数を表し、反証数は不詰めを証明するためのAND/ORゲーム木を完成するために最低限解決すべき未解決末端局面数を表す。よって任意の節点に於いて、証明数、反証数の値が小さいものほどその節点以下のAND/ORゲーム木の完成度が高く、より少ない探索量での完成が期待できる。この点を根拠として、それぞれのOR節点（証明数では詰め

方手番、反証数では玉方手番)において、証明数ないし反証数のもっとも小さい分枝を選択し続けて到達する未解決末端節点から優先的に展開していくP N*, PDSやdf-pnといったアルゴリズムは解を得るまでの探索量の劇的な削減効果をもたらした。[4][6]

2. 証明数、反証数の限界

しかしながら詰め将棋は本来、異なる手順の末同一局面にいたる合流や手順中に同一局面にもどってしまうループを含む有向グラフに一般化される。合流が生じた場合、証明数や反証数の2重カウントが多発し、本来持つべき意味合いが損なわれる結果となる。特にこのような分岐合流が多重化しているような例では証明数、反証数が指数関数的な増加をみせるため、分岐点で後の合流を察知するなんらかの特殊な処理を施さないと一部の詰め将棋において解図が著しく困難な例があることが報告されている。このような例に対して併用するハッシュ局面表に直前の局面へのポインタを持たせ、特殊な条件を満たした場合これを辿ること[6]や特定の駒の動きをした場合に先回りして局面表を調べること[8]により分岐点にて後の合流を特定し、AND節点において通常は子節点の証明数の和を証明数とするところ子節点の証明数の最大値に置き換えるという処理方法が紹介されている。しかしながらこれらの一般化された処理についてはあまり知られていない。

3. 経路分枝数について

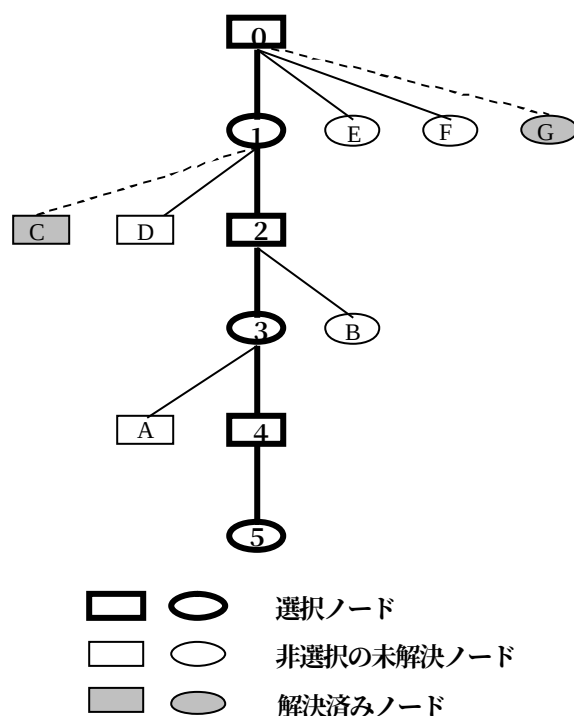
今回、このような詰め将棋における証明数、反証数の代替として始点から任意の未解決末端節点に至る経路に付随する着手選択枝数に着目した。一般的な有向グラフにおいて始点から任意の未解決末端節点に至る経路を選んだ場合、経路に付随する各節点において選択経路以外に複数の分枝を有することになる。詰め将棋においては、

経路 : 問題局面から末端局面に至る手順。

分枝数 : 手順中に含まれる未解決分枝の総和と定義することが可能である。ここでの未解決分枝とは分枝以下に形成されているAND/ORゲームグラフが完成されていない、すなわち少なくとも以下に1つ以上の未解決末端節点を有する分枝を言う。ここである詰め将棋が詰みということは問題局面(始点)から任意の最終詰み局面(解決済み末端節点)にいたる経路中のAND節点(玉方手番)における分枝数が0であることを意味し、不詰みということは任意の最終不詰み局面にいたるOR節点における分枝数が0であることを意味する。一般的には

AND節点の未解決分枝数が少ない経路を包含するAND/ORグラフほど詰みを証明する上での完成度が高く、OR節点の未解決分枝数が少ない経路を包含するAND/ORグラフほど不詰みを証明する上での完成度が高いといえる。よってこれらの経路の分枝数が少ない未解決末端節点を積極的に見つけ出し優先的に展開していくこともまた少ない探索量で問題を解決に導くと考えられる。以上の考察より本稿において経路分枝数というグラフ探索における指標を提案する。【図1】は探索過程での経路分枝数の概念を示すものである。AND/ORグラフにおける経路分枝数はOR節点での分枝数(obn:以下OR分枝数と呼ぶ)およびAND節点分枝数(abn:以下AND分枝数と呼ぶ)で構成されるものとする。図の中で、四角で示したものがOR節点、楕円で示したものがAND節点を表す。記号については0が探索の始点となるべきルート節点、5が一末端未解決節点、A,B,D,E,Fは非選択の未解決節点、C,Gは詰み、不詰みいずれかの結論が出た解決済み節点とする。アルファベットで表した前記各節点の下にはその節点自体を始点とするAND/ORグラフが生成されていることもあり、付随する経路分枝数(abn,obn)を有しているとする。

【図1】 経路分枝数の概念



太い直線で示した節点0～5を結ぶ経路が選択経路、その他の直線は非選択の分枝、破線は解決済みの節点に至る、事実上無視すべき分枝である。ここで提案する経路分枝数とは直感的には図におけるルート節点と任意の末端節点とを結ぶ経路に散在する未解決分枝の数を意味する。すなわちOR節点の分枝であれば0-E、0-F、2-B、および節点5が不明(未解決)の場合における4-5が対象となりOR分枝数 (obn) は4、AND節点の分枝であれば1-D、3-A、および節点5が未解決であればそれに加えたものが対象となり、AND分枝数は3となる。

以上をまとめると、ある始点と複数の未解決末端節点を含む詰め将棋における有向AND/ORゲームグラフにおいて、各節点における経路分枝数 (abn、obn) は以下の規則により定義するものとする。

【経路分枝数 (abn、obn) 定義】

abn : AND分枝数

obn : OR分枝数

(末端節点)

未解決 obn = 1, abn = 1

詰み abn = 0, obn = ∞

不詰み abn = ∞, obn = 0

(始点、中間節点)

OR節点 abn = (選択子節点の abn)

obn = (選択子節点の obn)

+ (非選択の未解決分枝数)

AND節点 obn = (選択子節点の obn)

abn = (選択子節点の abn)

+ (非選択の未解決分枝数)

4. 経路分枝数の特性

以上より、経路分枝数は従来の証明数、反証数の定義より、非選択の節点以下の情報を省略したのみの形に見える。これが証明数、反証数の代替となるべき指標になるのかについて検証する。

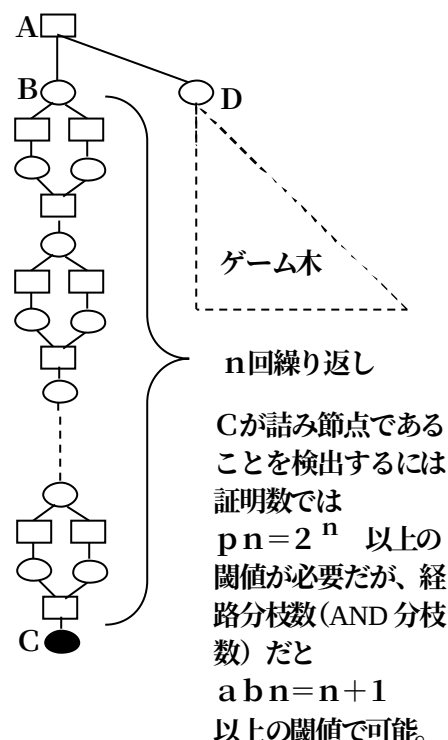
証明数、反証数における非選択節点以下の情報は、合流やループの無い完全な木構造のAND/ORゲーム木の解を想定した場合、有意義である。しかしながら合流が存在したゲームグラフを想定した場合、必ずしもそうとはいいきれない。分枝の合流により多重に証明数、反証数がカウントされる可能性が残されているからである。

例えば【図2】で示される詰め将棋のゲームグラフに於いてA→B→・・・→Cが作為であったとする。AND分枝数ではabnの閾値がn+1以上となれば解けるのに対し、証明数を用いた場合、pn

の閾値が2のn乗以上となった時点で解ける。よってnが大きくなった場合、証明数を用いた探索では解図が困難である。

しかしながらD以下の木構造のゲーム木に解が存在する場合、証明数を用いた探索のほうがB以下を早めに展開しない分、少ない探索量で解を得ることが期待できる。よって、証明数、反証数を用いた場合と経路分枝数をもちいた場合との解に至るまでの探索量の違いは作為手順とその変化で構成されるAND/ORゲームグラフに依存すると考えられる。

【図2】 AND分枝数と証明数の挙動の違い



5. 解図プログラムへの実装

今回、詰め将棋探索プログラムを作製し、経路分枝数を用いた多重反復深化法 (ここではBNS (Branch Number Search) と呼ぶ) と証明数、反証数を用いた従来の多重反復深化法 (df-pn 法) とで探索能力の比較をおこなった。経路分枝数利用の実装において以下の点に留意している。

5.1 経路選択のルール

経路分枝数は証明数、反証数によく似た挙動しめすのでこれらを利用した既存の探索アルゴリズムにそのまま適用することが可能と考えられる。

まず、探索の任意の段階において次に展開すべき

末端節点は3節での考察よりルートおよび経路中の OR 節点においては最小のAND分枝数を持つ子節点、また経路中の AND 節点では最小の OR 分枝数を持つ子節点を選択し続けて到達する末端節点である。そのような節点に到達するために分枝選択は一般的には以下となる。

OR節点

$abn = \text{MIN}$ (子節点の abn) となる子節点

AND節点：

$obn = \text{MIN}$ (子節点の obn) となる子節点

また、各節点においてOR (AND) 分枝数が同一の子節点が複数あり選択に迷う場合、対として保持しているAND (OR) 分枝数の少ない選択が、一般的に探索量の削減に効果があると考えられ、実験プログラムの分枝選択ルールもそのようにした。

次に選択分枝に与える多重反復深化の閾値であるが今回の実験では以下とした。

n ：探索節点

$obn(n)$ ：節点 n のOR分枝数

$abn(n)$ ：節点 n のAND分枝数

$c1, c2, \dots, ck, cn, \dots, cu$ ：子節点

ただし子節点間で以下の関係を持つ。

OR節点の場合：

$abn(c1) \leq abn(c2) \leq \dots \leq abn(ck),$

$abn(cn) = \dots = abn(cu) = \infty$

AND節点の場合：

$obn(c1) \leq obn(c2) \leq \dots \leq obn(ck),$

$obn(cn) = \dots = obn(cu) = \infty$

ABN, OBN ：節点に与えられた閾値

ABN', OBN' ：子節点に与える閾値

OR節点：

$ABN' = \min(abn(c2)+1, ABN)$

$OBN' = OBN - k + 1$

AND節点：

$ABN' = ABN - k + 1$

$OBN' = \min(obn(c2)+1, OBN)$

(戻り条件)

$OBN \leq obn(n)$ or $ABN \leq abn(n)$

(反復深化条件)

$OBN > obn(n)$ and $ABN > abn(n)$

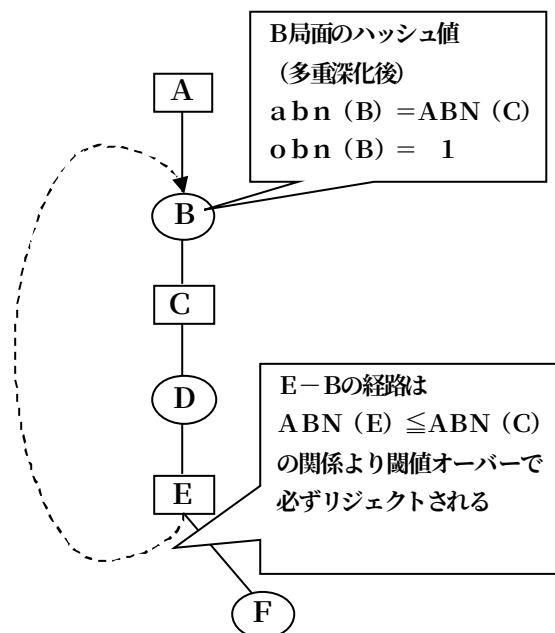
5.2 履歴判定

詰め将棋においては王手の千日手は詰め方が回避しなければならないという特別なルールがある。

詰め将棋での千日手をプログラム上で認識させるのはハッシュ局面表を用いるのがよい。しかしながら探索中の局面のハッシュ値に親局面から与えられた閾値をそのまま利用するのはうまく動作しないことが知られている。[7]

今回の実装ではハッシュ局面表を用いて千日手回避を以下のように組み込んだ。まず【図 3】に示すような単純な千日手による局面ループにおいてはある探索局面から多重深化させる際、探索局面ハッシュの abn の値を、一時的に選択した子局面に与える abn 閾値とする。 obn としてはOR節点では常に1を与えておく。たとえば図におけるAND節点Bにおいては節点Cを探索の際に与える閾値 $ABN(C)$ を $abn(B)$ としてあたえておく。これはOR節点も含めたすべての節点について同様にあつかう。こうすることにより探索手順中に含まれているすべての局面について詰め方手番 (OR節点) においては避けるように、また玉方手番 (AND節点) では優先して選択する方向で働く。多重深化を終え、コントロールが節点Bに戻った時点で改めて abn, obn を再計算の上局面のハッシュ値を再設定して処理を終えるようにする。

【図 3】単純なループの例



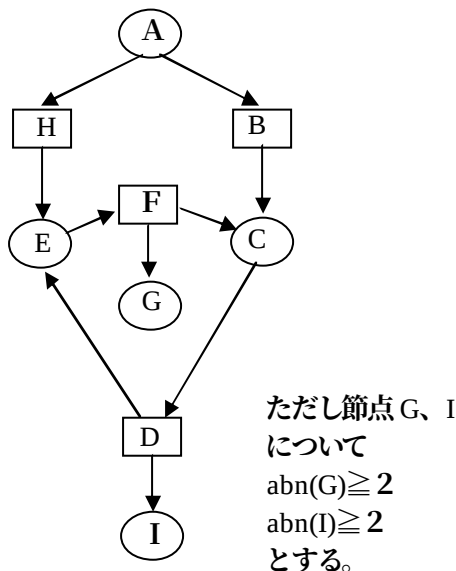
このような実装にしたのは GHI (Graph History Interaction) 問題による結果誤認の防止を強く意識したものである。[2]

さらに手順中に遭遇した局面と比べ同一盤面で且つ持ち駒が少なくなっている局面であることを認識した場合も同様に扱うことにより探索量を削減することが可能である。

また多重深化を用いた詰め将棋解図アルゴリズムでは前述した千日手による局面ループのほか、閾値を更新しながら特定の局面を特定の手順で展開と戻りを繰り返すハッシュでは捕捉しきれない展開履歴依存のループに遭遇することがある。

【図4】はハッシュで捕捉できない展開履歴依存のループの例である。

【図4】ハッシュで捕捉できないループ例



見かけ上は侵入節点として C と E の 2 箇所を持つ C、D、E、F の局面ループであるがそれらの侵入経路が共通の AND 節点より端を発している場合に生じる。図の例ではまず AND 節点 A を始点として obn の閾値 2 で $A \rightarrow B \rightarrow C \rightarrow D$ と展開する。その時点で D の obn が 2 であることより C、B の obn が 2 に変化し、コントロールが A に戻る。次に obn の閾値 3 で $A \rightarrow H \rightarrow E \rightarrow F \rightarrow C$ と展開される。ここで C には obn = 2 が記録されているので F の obn は 3 となり、E、H の obn に 3 が記録されつつコントロールが A にもどる。更に obn の閾値 4 で $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ と展開していく。ここでは E には obn = 3 が記録されているため、D、C、B の obn が 4 に変

化してふたたびコントロールが A に戻ることになる。以下 $A \rightarrow H \rightarrow E \rightarrow F \rightarrow C$ と $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ との展開を交互に繰り返しつつ、abn = 1 の状態のままで obn の値を徐々に増加させていくが G や I 以下は全く展開されない事態に陥る。今回の実装ではこのような展開履歴依存のループを回避するため、以下のような仕組みをほどこした。まずハッシュに探索局面以下の探索深さを記録しておく。もし前述のような展開履歴依存のループに遭遇した場合、節点 C、E で記録された探索深さ情報が節点 A において累積された形で見えてくる。よって探索深さが非現実的な値となったとき (たとえば最長手順詰め将棋の作為手数を上回った場合など) 閾値を (次善着手の obn + 1) から (親局面の obn 閾値) に引き上げることで回避するようにした。今回の実験プログラムでは節点以下の探索深さとして 1600 とし、これを超えた時点で事実上逐次展開に移行させるようにしてループを回避した。しかしながら今回の回避手法はきわめて暫定的なものでループ発生後閾値の切り替えまでに多少無駄な探索を行う可能性もある。

5.3 探索量削減について

経路分枝数は、証明数、反証数に性質がよく似ていることから、これらを用いての詰め将棋の探索に利用されていた多くの探索量削減の工夫をそのまま利用することが可能である。今回の実装においては既存より報告のある証明駒、反証駒、および持ち駒の優越関係の利用を施して探索量の削減を図った。[6] さらに持ち駒による合いについては任意の 1 手のみ展開し、この手が詰まない間は他の合い駒については一切展開しないようにした。

5.4 局面表の更新

今回の実験プログラムでは蓄積盤面数 7 M の局面表を使用し、それぞれの盤面単位にリスト形式で持ち駒ごとの局面情報を蓄積させる形とした。また探索量が増加してきた場合、局面廃棄を行わなければならない。廃棄の手法として Small Tree GC [6] に準じた方法を用いた。すなわち各局面データに節点以下の探索深さの情報を保持させた上で節点以下の探索深さが浅く、詰み情報のない盤面をより優先して廃棄する方式としている。探索深さの浅い情報は優越関係と証明駒が使えぬ詰み情報を除いては再利用の確率の低い局面データと考えられるからである。ただし詰みデータがハッシュ局面表の容量を圧迫してきた場合、ある時点で探索深さの浅いものよりまとめて廃棄するようにした。

6. 実験結果

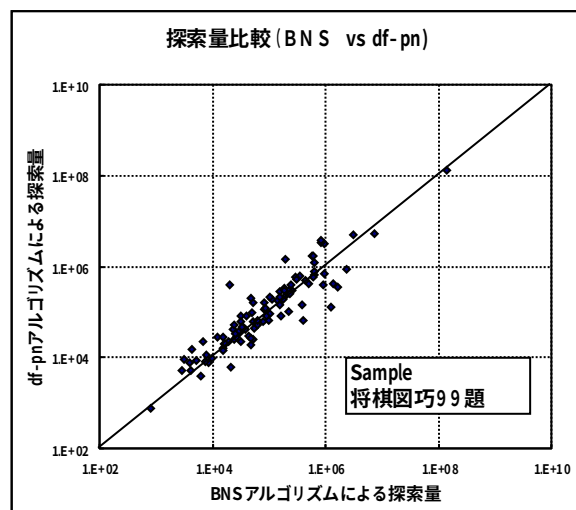
探索量はプログラムの実装に大きく依存するため閾値パラメータ（証明数、反証数と経路分枝数）以外の部分は同一のプログラムを使用した。

ここでは「将棋図巧」99題（補正図含む）[3] および現存する作為手数310手以上の超長編完全作41題 [9] を実験サンプルとして取り上げる。

6.1 将棋図巧

最初の実験サンプルとしてコンピュータ詰め将棋プログラムのベンチマークにしばしば用いられる【将棋図巧】を取り上げる。これらの問題の作為は9手～611手で平均して40手前後であり、作為中に証明数の2重カウントとなる玉方の着手非限定についても特に多重化した形ではあまり含まれていないと考えられる。【図5】は将棋図巧のうち、補正図を含めて詰ますことができるもの99題に対してBNS法を用いた場合の解を得るまでの探索量とdf-pn法を用いた場合での探索量とを比較した結果である。これを見ると $X=Y$ の直線を中心にはほぼ対称的にプロットが分布しており、これらの手法間では探索量に有意差はあまりないといえる。

【図5】 将棋図巧でのBNS法と df-pn 法との探索量の比較



6.2 超長編問題

次に超長編問題について比較をおこなった。一般的に作為が長手数になると玉方に非限定着手を含む機会が増えてくると考えられる。今回は現存する作為手数310手を超える完全作41題をサンプルとして用いた。その結果、実験プログラムにおいて実用時間内で解くことができたのは表1となった。尚、比較用の df-pn アルゴリズムでは2重カウント対策

は用いていない。

表1：超長編（310手以上）完全作41題中BNS法と df-pn 法とで解に至った問題数

BNS	35題
df-pn	25題(注1)

表2は今回の実験プログラムでBNS法でのみで解くことができた主な作品と解図までの探索量である。これらには作為中に玉方着手非限定で同一局面に至る手順が数多く含まれている作品として知られているものが多い。

表2：今回の実験においてBNSでのみ解図できた超長編作品とその探索量（注2）

	作者	名前	作為	探索量
1	田島 秀男		345	50191654
2	田島 秀男	夫婦馬	423	23613760
3	添川 公司	天国と地獄	431	5468096
4	田島 秀男	乱	451	200186714
5	田島 秀男	まだら	569	84528197
6	橋本 孝治		587	14988487
7	相馬 康幸	風神	589	629429527
8	馬詰・摩利	Fairway(修正図)	611	84528197
9	今村 修	ゴゴノソラ	651	33467625
10	添川 公司	桃花源	767	4023437

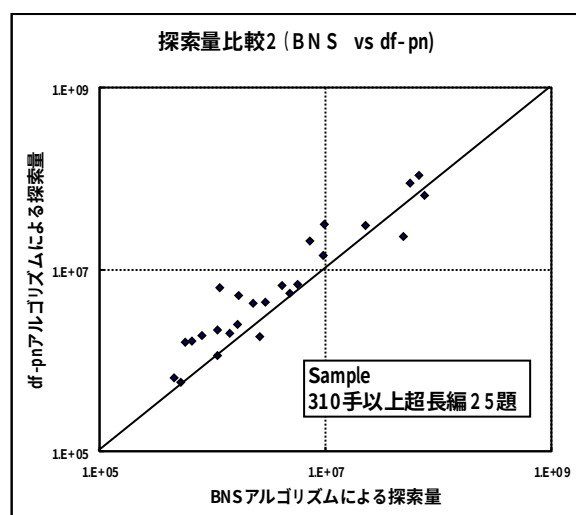
例題作品の出典および発表時期 [9]

	出典	発表
1	詰棋めいと27号	2000年2月
2	詰棋めいと31号	2002年12月
3	近代将棋	1995年6月
4	詰バラ	1999年10月
5	詰棋めいと29号	2001年5月
6	詰バラ	2001年3月
7	詰バラ	1985年4月
8	夢銭河	2000年3月
9	近代将棋	1995年8月
10	近代将棋	1985年12月

（注1）（注2） df-pn アルゴリズムにてハッシュ局面表にポイントをもたせ、合流点から辿って分岐点を特定し、証明数の2重カウント対策を施した場合、300手以上の超長編問題すべて（発表当時に存在したもの）について解けることは長井氏より報告されている。[5][6]

たとえば表中の Fairway(修正図)などは作為の中に玉の王手回避移動の非限定が70回近く現れる作品であり、証明数、反証数では十分な2重カウント対策をしないと極めて解きにくい問題である。この結果は経路分枝数を用いた探索（BNS）では局面の合流による2重カウントの影響が極めて限定的であり、このような問題に対する解図能力を高めていることを証明しているといえる。

【図6】超長編問題（310手以上）でのBNS法とdf-pn法との探索量の比較



【図6】は今回の実験プログラムにおいて BNS 法、df-pn 法ともに解けた310手以上の超長編問題25題についての解に至るまでの探索量の比較である。この中に最長手順詰め将棋「ミクロコスモス」も含まれ、今回の実験プログラムではBNS法で約5700万局面、df-pn 法で約8900万局面の探索で解に至っている。ここでの結果をみるとわずかながら左上に偏りが出てきているのが見て取れる。これも超長編詰め将棋に含まれる非限定作為の影響によりBNS法はdf-pn法と同等以上の解図能力を示しているといえる。

7. まとめ

経路分枝数を用いた探索は詰め将棋の問題を解くにあたり証明数、反証数を用いた探索と同様に解に至るまでの探索量を減らすために有効であることがわかった。また実験を通じて特に作為に玉方着手非限定を多く含む超長編詰め将棋については、証明数、反証数を用いた場合のように2重カウントを防止する対策を採らずともかなりの解図能力が発揮された。以上より経路分枝数を用いた探索はAND/ORゲームグラフ解析の帰着できる詰め将棋の解図に対し

て有力な手法の一つであると言える。

8. 最後に

今回の実験で作製したプログラムは残念ながらすべての詰め将棋について解くことができるまでは未だ至っていない。膨大な探索量に起因するもの、全く原因不明で解けないものがいくつも残されており、改めて詰め将棋というパズルの奥の深さ、および難解な作品を作り上げた作者の方々の力量に驚嘆するしだいです。

現代親しまれている囲碁、将棋、チェス、オセロといったゲームの中での変化はゲーム木というよりゲームグラフの解析に一般化されます。

その意味で人工知能の分野において今後もゲームグラフの解析はさらに重要な位置を占めていくと考えられ、今回の報告が今後のゲームグラフ探索研究の発展の役に少しでも立つことができれば幸いです。

作成にあたっては文献やネット上での多くの公開情報を参考といたしました。これらの情報の公開を積極的にしていただいた多くの研究家の方々に感謝いたします。

参考文献

- [1] L. Victor Allis, Maarten van der Meulen, H. Jaap van den Herik, Proof-number search, Artificial Intelligence, v.66 n.1, p.91-124, March 1994
- [2] Akihiro Kishimoto and Martin Müller: A general solution to the graph history interaction problem, 19th National Conference on Artificial Intelligence, published by AAAI Press, 644-649, 2004
- [3] 門脇 芳雄、詰むや詰まざるや、平凡社、1975
- [4] 脊尾 昌宏、共謀数を用いた詰将棋の解法、コンピュータ将棋の進歩2, pp.1—21、共立出版、1998
- [5] 長井 歩、コンピュータ詰将棋の最新成果、コンピュータ将棋会誌, Vol.13、2000
- [6] 長井 歩、df-pn アルゴリズムと詰将棋を解くプログラムへの応用、アマ4段を超える コンピュータ将棋の進歩4, pp.96-114、共立出版、2003

[7]岸本 章宏、I S将棋の詰め将棋解答プログラム
について、コンピュータ将棋会誌 Vol.16、2004

[8]柿木 義一、詰め将棋プログラムにおける証明数
の2重カウント対策の一手法、コンピュータ将棋会
誌 Vol.17、2005

[9]加藤 徹、詰将棋おもちゃ箱
超長編作品のプログラムによる解図と検討
<http://www.ne.jp/asahi/tetsu/toybox/kenkyu/chotesu.htm>,1999