非常に手数の長い詰将棋問題を解くアルゴリズムについて

伊藤琢巳[†] 河野泰人[††] 野下浩平[†††]

最良優先探索に基づいて設計した三つの詰将棋プログラムが611手詰という非常に手数の長い問題「寿（ことぶき）」をほぼ同時に解いた．寿は長手数問題の代表とされる問題で，古典詰将棋の中で最も有名なものである．詰将棋を解くプログラムは，速度と解答率の両面でこの3年間に急速に進歩した．手数の長い問題はこれまで100手台の問題が数題解けているだけであったので，詰手数が数百手を超える長手数問題を解くことは，詰将棋を解くプログラムの研究において最大の課題であった．それで，本稿の結果は，詰将棋プログラム技術の大きい進歩を示す．長手数問題を解くためのいろいろな技法，特に，探索法と評価関数に関する工夫，凝った先読み法，模倣の概念，近似ハッシュ法を考案し改良したが，これらは今後の応用可能性が大きい．また，非常に手数の長い問題を解くための困難な点，特に，よく似た手順や合流する手順を認識することの困難さを指摘するが，これは今後の研究課題を示す．

On the Algorithms for Solving Tsume-Shogi with Extremely Long Solution-Steps

TAKUMI ITOH,[†] YASUHITO KAWANO[††] and KOHEI NOSHITA[†††]

Our three Tsume-shogi (Japanese chess mating) programs based on best-first searching have solved the celebrated classic problem named Kotobuki that has six hundred and eleven solution-steps. Kotobuki is representative of all the Tsume-shogi problems with extremely long steps. Though Tsume-shogi programs have been enormously improved in terms of the speed as well as the solving power for the last three years, they have not yet solved a problem with more than two hundred steps so far. Thus solving problems with very long steps has been a research subject. The result of this paper implies the notable advance in Tsume-shogi programs. Various techniques concerning searching frameworks, evaluation functions, limited look-ahead, simulation and approximate hashing, are devised or improved, which can be useful in further applications. Some hard aspects to cope with long solution-steps, concerning similar and confluent sequences of moves, are also discussed, which are left to be challenged.

1. はじめに

よく知られているように詰将棋を解くことは，一般のゲーム木の探索として定式化できる[8),11)]．特に，詰むかどうかを判定し，詰む場合にはその解を求めることは，AND/OR木の探索問題になる．本稿が対象とする木は深さが数百になるが，このような非常に深い木を探索することは組合せ的問題として他の分野でも従来あまり例がない．本稿では，事例研究として詰将棋を取り上げ，その探索問題を解くための様々な方法を提示するとともに，このような問題を解く上での困難な点を指摘する．

一般にAND/OR木の探索では，先手のどれか一つ（OR）詰みに至る着手を探し，後手のそれに対するすべて（AND）の着手を調べ，詰みに至ることを示す．詰将棋問題の解手順（解答）としては，詰みに至るまでの着手の列（手順）のなかで，特に先手と後手双方が最強（先手最短後手最長）に応じたものを出力する．なお，詰将棋の用語やルールについては，本稿で必要な範囲で簡単に説明するが，詳しくは，文献6)，12)などを参照されたい．

超長手数の詰将棋とは，詰手数が非常に長い詰将棋問題である．現在までに作られた超長手数の問題のうち主要なものを表1にまとめる．この表は門脇芳雄が

† NTTソフトウェア研究所 広域コンピューティング研究部
Global Computing Lab., NTT Software Labs.
†† NTTコミュニケーション科学研究所
NTT Communication Scicence Labs.
††† 電気通信大学電気通信学部情報工学科
Department of Computer Science, The University of Electro-Communications

図1 伊藤看寿「寿」611 手詰
Fig. 1 Kanju's Kotobuki (611 solution-steps).

作成したリスト（1994）からの抜粋である．

筆者の3人はそれぞれ独立に詰将棋プログラムを作った．三つのプログラムは，94年5月，河野，伊藤，野下の順に相前後して611手詰の問題「寿（ことぶき）」を解いた．詰将棋プログラムはこの3年間に急速に進歩したが，手数の長い問題はそれまで100手台の問題を少数解いただけであり，長手数の問題を解くことが研究課題であった[2),9)]．現在のところ特定の超長手数問題を一つ解いただけであるが，詰将棋プログラムの技術の進歩を象徴するという点でこの結果は大きい．

寿の問題図を**図1**に載せる[4)]．この問題は，伊藤看寿の「将棋図巧」の第百番で，200年間長手数の記録をもっていたものであり，長い手数を実現するための問題作成技法として，龍で玉を追い回す“龍追”などを採用したもので，古典詰将棋の頂点に立つ作品とされ，人間にとって解くのに非常に難しいものである[4)]．詰将棋プログラムが解を出力した時には，その解の探索中に後手のすべての逃げ方に対する詰み（変化手順と呼ぶ）を検査しているので，この問題が実際に詰む問題であることがコンピュータで検証できたことになる．なお，この問題は終盤収束部に余詰があり，その解は629手詰になる．三つのプログラムともに余詰解の方を出力した．ここで余詰があるとは，詰みに至る手順の中で先手の着手として，他の手を選択しても詰めることができることをいう[6),12)]．つまり，先手の選択により詰み至る手順が，一意に決まらないことをいう．問題に余詰がある場合，その問題は不完全である．なお，後手の着手の選択によって詰みに至る手順は変化手順といい，基本的には作品の完全性を損ねない点に注意されたい．

今回の成果は，門脇芳雄による寿の部分問題によるところが大きい．94年3月に門脇が寿を簡単化した手数の短い部分問題を五つ提案した[13)]．このうち300手詰前後の部分問題を河野が解いたことにより寿を解ける可能性が示された．その結果，本稿で示す工夫を実現することにより，最終的な問題を解いた．

三つのプログラムは**表1**のほかの問題を解くことを試みたが，現在までどれも解けていない．このように，以前には試みすらしなかった超長手数問題がプログラムで解ける見込みがでてきたと同時に，その難しさもわかってきた．

本稿の次章以降では，三つのプログラムについて概要をそれぞれ説明し，超長手数問題を解くための課題などについて述べる．

2. 野下のプログラム

詰将棋プログラムの探索の枠組として，深さ優先探索と最良優先探索の2種類あるが，手数の長い問題に対してはもっぱら最良優先探索を用いる[10)]．伊藤と河野のプログラムは，展開のための節点を選択するのに，展開の都度ミニマックス則により探索木の中から先手後手ともに最強の道を計算する（動的展開）．一方，野下のプログラムT3は展開の際に計算した評価値の順に節点を展開する（静的展開）．なお，T3は深さ優先探索に基づくプログラムT2[2),9)]と別物である．一般に，二つの方法で同じ評価関数を用いるならば，動的展開の方が静的展開よりも探索節点数が少ないとされている（ある種の定式化のもとで証明もある）[8)]．動的展開のプログラムにより長手数の問題がかなり解けることを伊藤が示したこと[1)]をうけて，静的展開との差を実験的に調べるため作成したのがT3である．三つのプログラムはおおむね後手玉の自由度（玉の逃げ方の個数[1),9)]）に基づいた評価関数を用いているとはいえ，実際には独立して作成したもので，評価関数の細部はかなり異なっている．T3では，評価関数の計算にかける時間を多くすれば，動的展開のプログラムとは異なる振舞いを示すこと（問題によって得意不得意があること）が実験的に確かめられた．

T3のアルゴリズムの特徴を次にまとめる．

（1） 静的展開の最良優先探索による．ただし一部に探索の進行に応じて動的に評価値を変更する．例えば後手の一つの手に詰みがある場合，その他の後手の手の優先度を上げるようにする．

（2） 節点の評価関数の計算には，アルファベー

表 1 超長手数問題リスト
Table 1 List of Tsume-shogi problems with extremely long steps.

順位	手詰	作者	「命名」	発表誌	時期
1	1519	橋本孝治	ミクロコスモス	詰パラ	1986.6
2	941	山本昭一	メタ新世界	詰パラ	1982.7
3	873	奥薗幸雄	新扇詰	近代将棋	1955.1
4	789	橋本孝治	イオニゼーション	近代将棋	1985.12
5	767	添川公司	桃花源	近代将棋	1985.12
6	729	天変地異	バベルの塔	詰パラ	1981.10
7	611	伊藤看寿	寿	将棋図巧	1755
8	589	相馬康幸	風神	詰パラ	1985.4
9	551	黒川一郎	竹生島	風ぐるま	1953.9
10	525	山崎　隆	赤兎馬	詰パラ	1979.7
11	515	山本昭一	メガロポリス	詰パラ	1981.1
12	481	変幻自斉	勇馬伝説	自作集	
13	465	七条兼三		近代将棋	1980.4
14	433	三上　毅	扇の舞	詰パラ	1971.3
15	431	添川公司	天国と地獄	近代将棋	1993.6
16	421	黒川一郎	翁	詰パラ	1957.2
17	403	河原泰之	SWING II	詰パラ	1993.5
18	393	添川公司	呪われた夜	詰パラ	1981.2

タ法による数手先読みを行う．時間をかけるオプションでは通常 5 手，最大 7 手読むことがある．先読みによる葉（後手番）の値は玉の自由度による．ただし合（先手の飛角香による玉への効きを途中でさえぎるために打つ駒）は効きなどを考慮して個々の場合に応じて特別扱いをする．また先読みの評価値の計算法の変形，桂打ちや馬の移動の優先順位の向上など，各種ヒューリスティクスによる評価関数の評価法の選択が可能である．

（3） 後手番の節点の展開は逐次的に行う．これは，記憶領域の節約のためではあるが，同時に無駄合（解手順に影響しない無駄な合駒）の逐次的な判定に用いる（合は玉の近くから順に調べその結果を後の無駄合の判定に用いる）．ゲーム木のミニマックス則により，この方法による記憶領域節約の効果は非常に大きい．

（4） 評価関数の計算結果や節点の探索結果（詰・不詰）をハッシュ表により登録・検索する（基本となる方法は文献 2) を参照されたい）．さらに，河野と小山による近似ハッシュ法[5]を拡張して，飛，角，香とその合のある局面に対する近似ハッシュ法を組み込んでいる．この近似ハッシュ法では，飛，角，香の位置を消し，合の種類を消した局面の符号をキーにする．ハッシュ表のアフレに際しては，適当な優先順位にしたがって不要そうなものから順に忘れ，それを再利用する．

（5） ミニマックス則による計算により探索中に不要になった節点は再利用する．

寿を解くのに探索した節点の総数は約 1280 万であり，所要時間は約 65 時間である．時間をかける先読みを用いているので，1 秒当りの探索節点数は約 60 と著しく遅い．ここではワークステーション Sun Sparc IPX を用いた．記憶領域の大きさは節点数で 80 万（約 28 MB）である．

3. 伊藤のプログラム

伊藤のプログラムは，動的展開による最良優先探索を行うプログラムであり，長い手数の問題を解くために開発された[1]．現在のところ 30〜100 手の問題であれば約 40%解くことができる．なお，使用した計算機は，Sparc Station10 である．

本章では，最良優先探索を用いた詰将棋を解くアルゴリズムについて概説し，その後に，寿を解くための改良点について述べ，最後に超長手数問題がうまくとけない理由について考察を行う．

3.1 最良優先探索

ここでは，最良優先探索による詰将棋を解くアルゴリズムについて概説する．伊藤のプログラムは，動的評価による最良優先探索を行っている．動的評価とは，探索を行う度に探索木のルートから評価の一番良い枝を選択し，探索の結果を探索木の末端からルートまで逆向きに反映することをいう．伊藤のプログラムでは，詰みがありそうかどうかの評価基準に玉の自由度（玉の逃げられる数）を用いている．具体的な評価基準は以下のように与えられる．

末端局面
　詰み　：0
　不詰　：∞

未探索：1

中間局面

先手番：Min（子節点の玉の自由度）

後手番：Σ（子節点の玉の自由度）

この評価基準によると，探索木のルートの評価値が 0 になれば詰，∞ になれば不詰であり，0 に近いほど詰の可能性が高いものと評価される．後手で，Max（玉の自由度）でなくΣ（玉の自由度）を用いることによって，玉の逃げられる数を計算している．河野のプログラムでも基本的には同様の評価基準によって探索が行われる．

超長手数を解く場合にこの評価方法では問題がおきることが今回明らかになった．この問題点については後述する．

3.2 寿の解法

寿を解く際，前述したように門脇によって寿の部分問題が与えられた[13]．この問題は 287 手から 545 手までの 5 題用意されており，詰手数の短い順に計算機で解くことを試みた．その結果，287 手の問題のみ解くことができた．伊藤のプログラムの性能限界を知るため，できる限りの記憶領域（900 万局面分）を確保し再度実験を行った．その結果，伊藤のプログラムを何ら改造すること無しに，545 手の部分問題まですべて解くことができることが判明した．その際にかかった計算時間は約 5000 秒，消費した記憶領域は約 760 万局面分であった．この結果，動的展開を行う最良優先探索を用いた場合，時間と記憶領域さえ確保すれば寿が解ける可能性があることが示された．

次に，寿を解くために行った改良点について述べる．今回のアイデアは，詰/不詰の評価をできるだけ正確に行うことである．より正確な評価を行うことによって，不必要と思われる部分木の探索をなるべく行わず，その結果，より少ない記憶領域で解答を求められる効果を狙った．具体的には，探索木の末端における未探索局面の評価を単純に 1 とせず，深さ優先の先読みによって玉の自由度を評価しその結果を未探索局面の評価値とする，ハイブリッド探索法を採り入れた．ハイブリッド探索を行うことによって，先読みによる計算時間の増加があるものの，記憶領域はかなり節約できることがわかっている[3]．

ハイブリッド探索を行った結果，計算時間は 25 万秒（約 70 時間），探索した局面は 300 万局面で寿を解くことができた．

3.3 超長手数問題を解く際の問題点

表 1 に示した超長手数問題は寿を除いてどれも解けていない．長い手数の詰将棋を解くためのプログラムにもかかわらず，これらの問題が解けない理由について考察を行う．

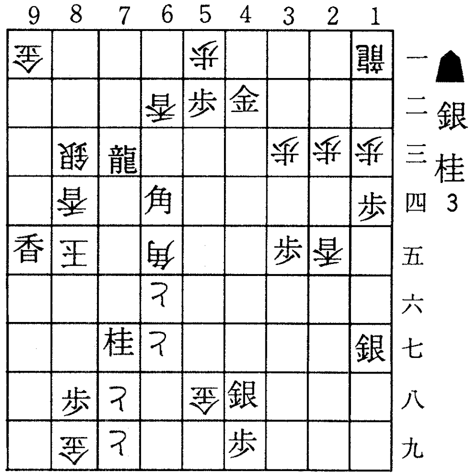

図 2　「寿」中間図（259 手目）
Fig. 2　Kotobuki (at 259th Step).

長い手数の問題が解けない理由の一つに，探索時の評価の行い方がある．3.1 節で述べたように，後手の評価は Σ（玉の自由度）となっている．後手番で Max（玉の自由度）ではなく Σ（玉の自由度）としているのは，詰将棋の場合，何通りにも逃げられるのは詰めにくいという直観的な評価を表現したものであり，実際にも最大値ではなく和をとった方が速く解けることが多い．

ところが，後手番で Σ（玉の自由度）を用いることによって，探索がうまく行われない場合がある．それが，成/不成の非限定や，駒の移動順序の非限定などという着手の非限定である．具体的な例として，寿の 260 手目があげられる．この例では，7 七に打った桂馬を 6 六，6 七，7 八にいる後手番 3 枚の “と金” でとるという手順を 3 回繰り返すが，と金はどの順番で動かしてもよい．さらに，その後 7 七桂打ちとそれを “と金” でとるという手順を 3 回繰り返す．

これを Σ で表現すると，最初の後手番の評価が 3 となり，さらに，それぞれを探索していくと，もう一度おなじ局面で二つに分岐してしまうことになる．したがって，最初に 7 七桂を打った局面の評価は，一つ “と金” をはがした後の評価の 6 倍となってしまう（**図 2**）．

その結果，正解手順の探索優先順位が非常に低くなってしまい，正解手順が探索されなくなる（**図 3**）．また，仮に解けた場合でも，同じような手順を何度も繰り返し探索してしまうので，計算時間や記憶領域を

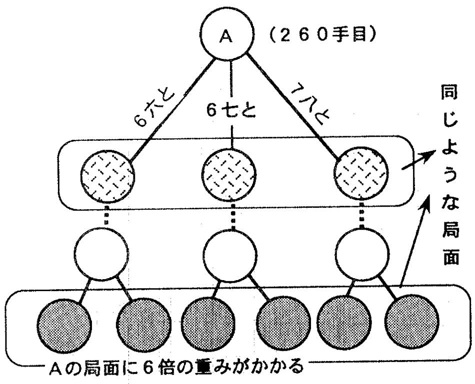

図3 動的展開の問題点
Fig. 3 A difficulty of dynamic expansion.

多量に消費してしまう.

超長手数問題の場合，例外なく長い手数の似たような手順を繰り返すことで手数を稼いでるので，後手番で Σ（玉の自由度）を用いると，うまく解けない場合が多い.

上記のような問題を引き起こすもう一つの理由には，後手番の手の探索方法がある．後手番の手は，詰む場合にはどれを選択しても詰んでしまうので，本来ならばどこから探索してもよい．しかし，実際には一番詰みにくい手から優先して手を選択している.

後手番の手を選択する場合に，一つの手だけを集中的に探索して，その手が詰むまで他の候補手を選択しないという方法によって，局面の評価値が異常に大きくなるという現象をある程度くいとめることができる.

4. 河野のプログラム

河野の詰将棋プログラムは最良優先探索によるものである．このプログラムはもともと短手数詰将棋問題の自動創作プログラムのサブルーチンとして開発が始められた．まず試作品が作られ各種の性能テストが行われた後，単独の詰将棋プログラムとしても良い結果を出すよう，大きな修正が加えられ現在のものとなっている．試作品は探索速度が速く比較的長い手数の問題も解ける反面，記憶領域の消費量が多いなどの欠点があった.

本章では，河野が行ったプログラムの改良と，超長手数問題の難しさについて述べる.

4.1 河野プログラムの改良

上記の試作プログラムで数多くの詰将棋問題を解き性能テストを行った結果から，以下の点に気がついた.

（1） 合駒処理は近似局面判定の特殊形である.

（2） 短手数問題と長手数問題では合駒処理の方法を少し変えた方が良い.

（3） 近似局面の利用は，解手順中によく似た局面が何度も現れる超長手数問題を解く際に有効になることが予想される.

そこで改良版では，合駒処理アルゴリズムを近似局面判定アルゴリズムで置き換え，同時にアルゴリズムの処理対象を合駒以外にもひろげた．その結果，多くの問題に対して消費する記憶領域が少なくてすむようになり，解答率も向上した.

以下，局面間の近似関係の定義と利用法について簡単に説明する[5).]

局面 P が局面 Q を f-simulate するとは，大雑把にいって，局面 P の解手順に従って局面 Q を詰めることができることをいう．ここで“解手順に従う”とは，先手は P の解手順をもとに選択関数 f が生成する複数候補手の中から手の選択が可能で，後手の着手は Q を読み進めた各局面において新しく生成することを意味する.

このように二項関係 f-simulate を定義すると，局面 P が詰むかどうかに関わらず P f-simulate Q ならば Q は詰むことが証明される.

この二項関係を有効に利用するために，詰将棋の問題が作るゲーム木の後手の節点全体に優先順位（priority）をつける．この優先順位は，局所的に定義された優先順位のゲーム木全体への自然な拡張によって得られる．そして優先順位に従って探索することにより条件つきの解を求めることを優先し，初めに探索から除外した優先順位の低い節点を探索に加える際には，既に解が得られた節点との間で上記二項関係の成否を調べる．もし二項関係が成立していれば優先順位の低い側の節点について解の探索をする必要がないので，重複計算が避けられ記憶領域の節約と高速化につながる.

上記の手法で，応用上優先順位の設定が最も重要である．一般的には，後手の最強応手の集合に最も高い優先順位を与え，計算が重複しそうな手の優先順位は異なる値に設定する.

寿の場合，本手順中の 7 七桂打ちに対する“同と”という後手の着手が非限定になっている（動ける“と金”が複数枚あり，どの駒で取ってもよい）．そのため，普通に探索を行うと 100 手近い同一の変化手順を十数回も繰り返し読む必要がある.

河野のプログラムでは，寿を解くために“焦点打ちの駒”を取る手に対する優先順位を異なる値に設定することにより大量の重複計算を避け，短時間での解答と記憶領域の節約に成功した．ワークステーション

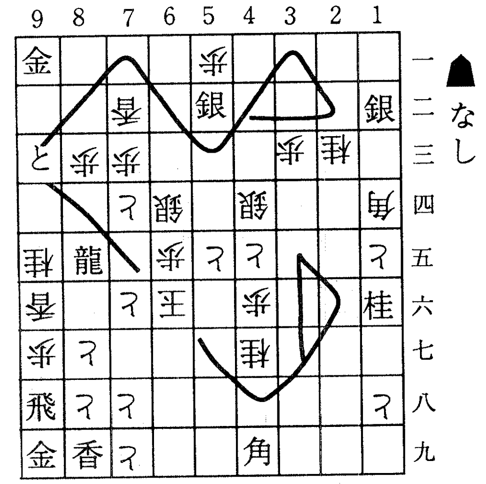

図 4 黒川一郎「竹生島」551 手詰
Fig. 4 Kurokawa's Chikubujima (551 solution-steps).

図 5 「竹生島」149 手目
Fig. 5 Chikubujima (at 149th step).

HP9000 model735/99 MHz を使用した場合，寿を解くための所要時間は約 8 分，消費した記憶領域は約 200 MB であった．

4.2 超長手数問題の難しさ

一般に，解くために高度なヒューリスティクスが必要な超長手数問題に対しては，人間のエキスパートの思考に近いアルゴリズムが不可欠であろうと言われている．

寿以外の超長手数問題をプログラムで解くには，エキスパートの思考と詰将棋を解くアルゴリズムとの違いを検討する必要がある．そこでエキスパートにとって非常に易しい超長手数問題を取り上げ，その問題の詰将棋プログラムにとっての難しさを分析することにより，現在の詰将棋プログラムに欠けている要素について考察する．

黒川一郎作「竹生島」は，“龍追”の中では最も構造の簡単な部類に属する．**図 4**（問題図）では，分かりやすさを考慮して玉の軌道を太線で記してある．門脇芳雄によれば，この問題はエキスパートなら 5 分程度で解けるとのことである．詰将棋プログラムから見た難しさを説明する前に，解手順を簡単に説明する．

この問題で先手は，龍で玉を追いながら最終的に軌道上の一点を封鎖することを目指す．封鎖するのは 5 七の位置で，そのために 6 九に桂馬を打ちたい．そのためには邪魔駒 7 九と，7 八とを消去しなければならず，さらにこれらの駒を消去するためには左下にある後手の駒をすべて消去する必要がある．そこで解手順では，龍で玉を追いながら 1 周するごとに 1 枚ずつ左下の後手の駒を消去する．8 周で隅の邪魔駒をすべて掃除し終り，9 周目の途中で桂馬を打ち，10 周目に入ったところで玉は逃げ道を失って詰む．

この問題が解けない最大の原因は，現在の詰将棋プログラムが“見かけ上の分岐”を正しく認識できない点にあると考えられる．

図 5 は竹生島 149 手目の局面である．この時玉には 8 二に逃げる手と 8 四に逃げる手がある．詰将棋のルールにより後手はもっとも長手数になるような逃げ方をする必要があるので本手順は 8 二玉である．しかしこう逃げた後，図の a で示した一本道のコースを辿って 30 手後には 8 四に逃げた手順に合流する．このように，ダグ（サイクルのない有向グラフ）で表現した場合，後ですぐに合流するような後手番でのグラフの分岐のことを“見かけ上の分岐”と呼ぶことにする．

後手の着手非限定も“見かけ上の分岐”の代表例である．最良優先の詰将棋プログラムでは攻め方の着手を選ぶ際の評価関数として“玉の自由度（=分岐数）”を使い，分岐数が大きいほど詰み難い局面であると判断している．そのため，プログラムは“見かけ上の分岐”を持つ局面を実際以上に詰み難いと誤認する．

竹生島では本手順上に“見かけ上の分岐”が 7 箇所もあり，エキスパートが一本道であると判断する手順を，コンピュータは多数の分岐をもつ詰め難い手順と勘違いしてその先を深く読もうとしない．

こうした超長手数問題中に現れる“見かけ上の分岐”はプログラムにとって最も厄介な存在であり，何らかのうまい対策が見つからない限り超長手数問題に対する詰将棋プログラムの実力が今後大きく向上することはないと思われる．

5. あとがき

寿は解けたとはいえ，現在の詰将棋プログラムは超長手数問題を短時間で解くには，まだ十分の能力をもっていない．しかし，はじめにで述べたように，少し前までは超長手数の問題領域は，まったく未知の世界であった（詰将棋は人間の作ったものではあるが計算量が十分大きい）．このような領域の問題群に対して，アルゴリズムの一般的な工夫というこれまでの戦略が基本的にそのままの形で使えることを示唆したという点は，本稿の最大の成果であると判断できる．

本稿で示した技術的な成果の主なものとしては，探索法と評価関数に関する諸種の工夫，凝った先読み法，模倣と優先順位の概念，近似ハッシュ法の考案ないしは改良があげられるが，これらは今後の応用可能性が大きい．また，非常に手数の長い問題を解くための困難な点，特に，よく似た手順や合流する手順を認識することの困難さを指摘したが，これらは今後の研究課題を示す．

なお，本稿は情報処理学会夏のプログラミングシンポジウム（94 年 7 月）で発表した報告に基づいている．本稿の執筆の後，いくつかの進歩が見られたので付記する．野下は，本稿で指摘したよく似た手順の繰返しの発見法に新しい工夫を導入し，超長手数問題リスト 17 位の 403 手詰の問題を解いた（寿は約 40 万節点の探索により 1 時間程度で解く）．ごく最近，脊尾昌宏（東京大学）は，共謀数（conspiracy number）の概念を応用して，問題リストの数題を新しく解いた[7]．その中には 3 位 873 手詰の新扇詰が含まれる．

参考文献

1) 伊藤：最良優先方式による解探索アルゴリズム，第 44 回情報処理学会全国大会論文集，No.3-49 (1992).
2) 伊藤，野下：詰将棋を速く解く二つのプログラムとその評価，情報処理学会論文誌，Vol.35, No.8, pp.1531–1539 (1994).
3) 伊藤：ハイブリッド探索法を用いた詰将棋を解くプログラム，第 48 回情報処理学会全国大会論文集，No.2–215(1994).
4) 門脇 編：詰むや詰まざるや，平凡社東洋文庫 (1975).
5) Kawano, Y.: Using Similar Positions to Search Game Trees, to appear in *Proc. of Combinatorial Game Workshop* (1994).
6) 小山，河野：名作詰将棋における感性の定量評価，情報処理学会論文誌，Vol.35, No.11, pp.2338–2346 (1994).
7) 脊尾：C^*アルゴリズムによる AND/OR 木の探索および詰将棋プログラムへの応用，情報処理学会人工知能研究会，人工知能 99-14, pp.103–110 (1995).
8) Nilsson, N.: *Problem Solving Methods in Artificial Intelligence*, McGraw-Hill (1971).
9) 野下：詰将棋を解くアルゴリズムについて，電子情報通信学会研究会報告，COMP91-56, pp.29–37 (1991).
10) 野下：詰将棋を解くアルゴリズム，*Computer Today*, Vol.57, pp.14–18 (1993).
11) Pearl, J.: *Heuristics*, Addison-Wesley, Reading (1984).
12) 松原：将棋とコンピュータ，共立出版 (1994).
13) 小谷，飯田 編：コンピュータ将棋協会資料集，Vol.8, CSA (1994).

(平成 7 年 3 月 9 日受付)
(平成 7 年 9 月 6 日採録)

伊藤　琢巳（正会員）

1964 年生．1987 年茨城大学工学部情報工学科卒業，1989 年同大学修士課程修了．同年，NTT に入社，NTT ソフトウェア研究所に所属，現在に至る．人工知能，なかでもゲームにおける問題解決などに興味を持つ．コンピュータ将棋協会会員．

河野　泰人

1967 年生．1989 年大阪大学理学部数学科卒業．1991 年名古屋大学理数大学院修了．同年 NTT 入社．現在，NTT コミュニケーション科学研究所研究員．数理論理学の研究に従事．電子情報通信学会会員．

野下　浩平（正会員）

1966 年東京大学工学部計数工学科卒業．現在，電気通信大学情報工学科教授．工学博士．アルゴリズム解析，組合せゲームなどの研究に従事．CSA などの会員．