Vol. 47 No. 12 情報処理学会論文誌 Dec. 2006

引き分けの証明向きのGHI問題対処法とその3x4将棋への応用

柴 原 一 友† 但 馬 康 宏† 小 谷 善 行†

本論文では将棋を解く研究への布石として，引き分けを効率的に証明するための GHI 問題回避法を提案し，3x4 の将棋の解を求めた．オセロや囲碁に比べ，将棋は小さな盤面に対する解を求める研究は少ない．将棋は局面がループするため，引き分けを含めた解を得ることが困難である．そこで，詰将棋で効果をあげた PDS 探索を用いて，効率的に引き分けを証明する方法を提案する．実験の結果，3x4 将棋のすべての解を得た．

A New Method of Dealing with GHI Problem for Proving Draw and Application to 3x4 Boards of Shogi

Kazutomo Shibahara,† Yasuhiro Tajima† and Yoshiyuki Kotani†

We obtained all the solutions of 3x4 Shogi by using a new proof number search method to deal with GHI problem in terms of proving draw effectively. It is a stepping stone to solve Shogi. There are few researches about solution of small Shogi compared with othello and Go. It has difficulty to obtain solutions, especially draw, because search trees of Shogi have many cycles. We propose the method that efficiently prove the draw by using PDS search which has beneficial effects in tsume Shogi. As a result, it solved all initial positions of 3x4 Shogi.

1. はじめに

ゲーム研究の最終目標の 1 つは，そのゲームの解を得ることである．近年チェッカーの解を得る研究[14] も行われており，解を得る研究は着々と進んでいる．また，オセロ，囲碁，Hex などでは，盤面を小さくした場合における解を求める研究がなされている[17]．小さな盤面に対して解を得ることは，通常のゲームの解を得る前段階としても重要なことである．その結果によって，通常の盤面における解の予測の精度を向上させることも可能となる．また，将棋は駒を同じ位置に再移動させることが容易なことから，局面のループが発生しやすい．そのため，将棋の解を得る探索は，倉庫番ゲームや裸玉問題の解を得る探索などのループを多く持つ木の探索の向上に役立つと考えられる．

本論文では証明数探索手法の 1 つである PDS 探索[11] を用いて，引き分けの証明に効果的な GHI 問題[1] の回避方法を提案している．証明数探索手法は詰将棋で多大な効果を発揮した手法であり，将棋を解くうえでも有効である．しかし，将棋のように局面がループする手順が存在するゲームの場合，1 度得た局面の解の情報を探索中で再利用すると，場合によっては正確な解を得ることができない．この問題は GHI 問題と呼ばれており，いくつかの回避法が提案されている．本論文では GHI 問題を引き分けの証明に特化した形で回避する方法を提案し，その方法を使用して 3x4 将棋の問題の解を求めた．

2. 3x4 将棋のルール設定

3x4 将棋は，後藤らがその解を求める実験を行っている[2]．本論文ではその設定で実験を行った．具体的なルール設定は次のとおりである．

- 盤面は縦 4 マス，横 3 マス
- 自陣 1 段目にだけ自分の駒を配置
- 1 つの局面に対し，駒が成れる段は 1 段目から 3 段目までの 3 種類
- 解は，先手必勝，後手必勝，引き分け（同一局面が 4 回現れる．千日手とも呼ばれる）の 3 種類
- 連続王手の千日手は禁止としない

初めから後手玉に利きがある局面と，盤面を左右反転したときに重複する局面を除外したとき，3x4 将棋の駒配置の総数は 79 となる．駒が成れる段は 3 種類あるため，3x4 将棋の局面は全部で $3 \times 79 = 237$ 局

† 東京農工大学
Tokyo University of Agriculture and Technology

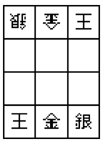

図 1　3x4 将棋の例
Fig. 1　Example of 3x4 Shogi.

面となる．

千日手とは，お互いが負けないように最善を尽くす場合，対局の中で同一局面が何度も現れる（ループする）状態となり，決着がつかなくなることを指す．また，通常の将棋においては，連続王手の千日手は禁止されている．これは，一方の手番が王手をかけ続けて千日手に陥る状態のことであり，この場合，王手をかけ続けた側の負けとなる．しかし，連続王手の千日手は検出することが困難なため，今回の実験では連続王手の千日手は特に禁止としない．この設定も後藤らの論文と同一である．3x4 将棋の盤面の例を**図 1** に示す．

3. 引き分けの証明法

詰将棋の解は詰むか詰まないかの二値である．これに対し，将棋の解は必勝か必敗か引き分けの三値である[4]．詰将棋においてもループする局面に至る可能性はあるが，その局面は連続王手の千日手となることから，詰まない局面として分類することで二値化している．詰将棋における詰むか詰まないかを，必勝か，必勝でないかとして置き換えた場合，ループ手順に至ってしまう局面に関しては，必勝でない局面として扱うことで，詰将棋の場合と同様に二値化できる．実際に探索する場合には，問題局面からの手順の中で，同一の局面が 2 度現れた場合に必勝でない局面（引き分け）と判断すればよい．この探索を用いて引き分けを証明するには，まず先手必勝か先手必勝ではないかを調べ，先手必勝ではないと判明した場合は，後手必勝か後手必勝ではないかを調べる．後手必勝でもないと判断された場合は，その局面は先手必勝でも後手必勝でもないので，引き分けであると判明する．これに関連する証明に関しては 5.1 節で示している．

4. 関 連 研 究

4.1 小さな将棋の研究

小さな将棋の解を求める研究は少ない．一次元の将棋の解に関する研究と[4),10)]，3x3，3x4[2)]，4x4[16)] の将棋を解く研究があるだけである．しかし，3x4 は全局面数の 56%しか解を得られていない．特に，千日手と思われる局面に対して解を得ることが難しく，引き分けに関しては，$\alpha\beta$ 探索によってわずかに 4 局面だけ解を得ている．

引き分けとなる局面の数を n とするとき，その局面から得られる引き分けのルートは最大で n!となる．もちろん，すべての局面に対しリンクがあるわけではないため，ここまでの数にはならないが，最大について考えるとき，引き分けの局面数が x 倍になれば，引き分けのルートの最大数は (x・n)!となってしまう．3x4 を解くことがいかに難しいかがうかがえる．ちなみに，3x3 と 3x4 それぞれにおいて，飛王金の駒配置の場合，3x4 の可能な局面の総数は 3x3 の約 10 倍程度となることが示されている．また，予備実験の結果，3x4 将棋の一部の問題は同一局面が現れるまでに 20 万手以上要するものもあることが分かっている．詰将棋において，現存する問題における最長手順が 1,525 手であることから考えても，計算に要する時間は途方もないことが予想される．また，さらに盤面の小さい 3x3 に対する実験では 3 問だけ解を得られず，後の追加実験で 30 分 PDS 探索や $\alpha\beta$ 探索を動かしても解を得られていない．

4.2 PDS 探索

本論文において，3x4 の将棋の解を得るうえで PDS 探索[11)] を使用している．より優れた手法として df-pn[13)] がある．本手法は df-pn を用いても実装可能であるが，証明数などの閾値の制御が PDS 探索に比べ煩雑であることから，ここでは新しい手法の導入が容易な PDS 探索を使用する．PDS 探索は各局面が保持する証明数と反証数の 2 つの値を使用して探索する手法である．証明数とはその局面が必勝であることを証明するために，証明しなければならない局面数であり，反証数とはその局面が必勝でないことを証明するために，証明しなければならない局面数を意味している．その局面が，その手番側の勝ちとなる場合，証明数は 0 に，反証数は ∞ となる．負けとなる場合は，証明数が ∞ に，反証数は 0 となる．新しく探索しようとしている未展開の局面は両者とも 1 になる．探索中で親局面の証明数や反証数は，子供局面の証明数，反証数を用いて，それぞれの意味を満たすように算出される．PDS 探索では，証明数が小さな局面から探索していく．証明数が同値である場合は，反証数が小さい局面から探索する．

PDS 探索では開始局面（探索を開始する局面，ルート局面）において証明数と反証数の閾値を 1 に設定する．開始局面の証明数と反証数がともに閾値を超えた場合に探索は一時的に終了し，さらに閾値を増やして探索を行う．繰り返し同じ局面を探索することになる

が，前回の探索の情報をハッシュ表に登録しておくことで，無駄な再計算を排除している．また，PDS 探索などの証明数を用いた探索は，ハッシュ表の利用が必須となる．部分木の情報が欠損し続けるような状況の場合，探索アルゴリズムはその部分木が欠損するたびに再構築しなおす場合があり，探索が終了しないことがある．

4.3 GHI 問題

証明数探索においてループする手順は大きな問題となる．手順の中に同じ局面が 2 度現れたとき，必勝でないとして扱うことで処理しようとする場合，GHI 問題と呼ばれる問題を引き起こすことが知られている．GHI 問題とは同じ局面であっても，両者の局面の解がその局面に至る手順に依存して異なっているとき，探索中にそれらを同一の局面として扱うことによって，探索の解が保証されなくなる問題のことである．ある局面がループに陥るためには，それ以前の手順に同一局面が現れていることが必須となる．そのため，手順の中に同一局面があるかないかで，その局面の解が異なってしまう場合がある．PDS 探索などの証明数探索は，4.2 節で示したように，判明している局面の解をハッシュ表に保存して再利用することが必須となっているため，ループした局面を必勝でないと登録した場合，この情報を再利用することで間違った解を返す可能性がある．手順によっては必勝（詰み）となる局面を，画一的に必勝でない（不詰）としてしまうからである．

GHI 問題の例を**図 2** に示す．図 2 において，ノード A と B と C は同一局面であるとする．このとき，ノード A を通過してノード B にたどりついたとき，ノード B は手順上に 2 度同一局面が現れたために，必勝でないと判断される．同時にノード A も必勝でないと判断されるため，以降の探索でノード A 以下が探索されることはなく，他の手順によって至るノード C でも同様の判断がなされるため，ノード A やノード C 以下の探索木は十分に構築されない．そのため，ノード A 以下に実際は必勝の手順があった場合，発見することができなくなる．この結果，実際には必勝であるにもかかわらず，必勝でないと判断される可能性が生まれる．

将棋の局面で発生する GHI 問題の例を**図 3** に示す．この局面は後手番であり，先手必勝となる局面である．手順としては，△ 3 一玉，▲ 1 一飛成までとなる．しかし，候補手の探索順番によっては，△ 3 一玉，▲ 1 三飛，△ 2 一玉，▲ 1 四飛とたどり，同一局面へと戻ってきてしまう．そのため局面がループ手順に陥

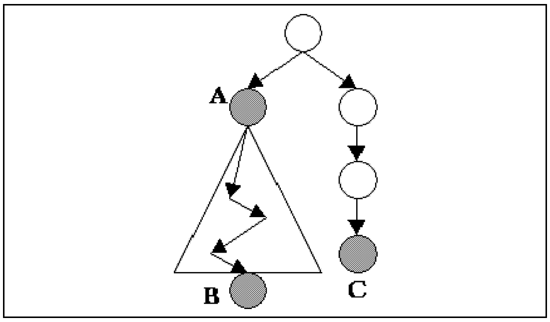

図 2 GHI 問題の例
Fig. 2 Example of GHI problem.

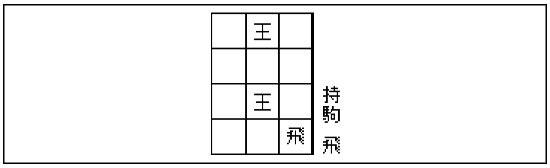

図 3 将棋の局面における GHI 問題の例
Fig. 3 Example of GHI problem in Shogi position.

ると誤って判断してしまい，先手必勝と判断できなくなってしまう．

GHI 問題は，証明数探索を使用する場合，回避することが難しい問題となる．証明数探索はハッシュ表の利用を必須としているため，単純にループ局面の登録を避けることは難しい．4.2 節で示したように，情報の欠損は探索の停止につながるからである．また，ループに至る手順が存在するときに，その手順の最終手（ループ局面に至る最後の手）を無視するなどの対処では GHI 問題を避けることはできない．無視することなどの特定の局面に対する特別な対処が，そのまま手順の異なる同一局面の差異を生み出すため，GHI 問題が発生してしまうからである．仮にその局面の情報にうまく対処できたとしても，その親局面もまた GHI 問題を発生させてしまう．

この問題を回避する方法は過去にいくつか提案されている[1),3),18)]．しかしその大半は，間違った解を返す可能性を減らすものであったり，すべての局面に対し，手順と局面をハッシュに登録することで GHI 問題を回避するかわりに探索の効率を大幅に下げてしまったり，PDS 探索や df-pn 探索に対して実装してその性能を確認していないものであった．その中で，文献 12) では，ループに関係ない詰，不詰の発見を優先し，定められた閾値の上限（∞-1）を超えても解が得られなかった場合に初めて，閾値を ∞ にしてループを不詰にする設定で探索を実行する方法を提案し，実装している．∞-1 の探索でループ以外の証明が完了すれば，

不詰を発見できるが，この方法は不詰の証明能力に関してはそれほど高くないことや，閾値の限界を超えるまで行う探索の明らかな無駄が指摘されている[8)].

GHI 問題に対する最も有効な方法として提案されているのが，ループに依存して得られている解を別のハッシュ表に登録し，その局面に至る手順もハッシュ化してあわせて保存することで，手順の異なる局面を別局面として扱う方法である[8),9)]．また，その局面の結果に影響されている手順依存の局面についてもあわせて，手順と局面をハッシュ化して登録する．手順非依存な局面に関しては手順を登録しないことで高速化を図っている．すでにループ局面として登録された情報に至った場合は，登録された手順のハッシュを調べ，違う手順で至っているならば新規に探索を行う．この際，探索の無駄を削減するために，シミュレーション[6)]を行っている．シミュレーションとは，すでに登録されている同一局面を証明した手順を再利用することで，新たに調べようとしている異なった手順で至った局面の証明を効率的に行う方法である．

この方法を使用することで，詰将棋において不詰を正しく証明することができることが示されている．しかし，このアルゴリズムを将棋を解くために使用する場合，いくつかの問題点がある．詰将棋を解くことと，将棋を解くことは，性質が若干異なっている．詰将棋では攻め手は相手に王手をかける手だけを指すことができ，受け手はその王手から，逃れる手だけを指すことができる．しかし，将棋を解く場合はその制限がない．そのため，将棋を解く問題は，詰将棋を解く問題に比べてループする局面が現れやすい．その結果，手順の違う同一局面の登録量が詰将棋に比べて多くなり，メモリを圧迫する可能性が高い．また，詰将棋よりも手の制限が緩い分，最長手数も必然的に長くなってしまう．先述のように，3x4 の将棋においても同一局面が現れない手順の長さはゆうに 20 万手を超えてしまう．よって，探索では 20 万手を超える情報を保持しなければならなくなるため，メモリの圧迫が激しい．より大きな盤面を解く場合も考慮すれば，メモリの圧迫は無視できない問題である．一般に，各局面における候補手数の平均を w，探索の深さを d としたとき，その局面の探索量は最低でも $w^{\frac{d}{2}}$ 程度になることが知られている．単純に考えれば，3x4 将棋における d の値は 20 万に至る場合があることから，予想探索量は w^{100000} となる．先手必勝や後手必勝に至る候補手については 20 万に至ることはなく，また，同一局面が現れた場合に探索を終了できるため，単純にこの計算量になることはないが，非常に大きな探索量になることが予想される．さらに，先述の GHI 対策方法では，手順をハッシュ化して保持しなければならないため，確保する乱数の量からくるメモリの圧迫もある．文献 12) では，手順の乱数は 97 手分に制限し，それ以降は使いまわしをしている．詰将棋においては，97 手も進むと同一局面は現れにくいため，衝突の不安はないが，将棋を解く場合は同一局面が生まれやすく，ハッシュの衝突に関しても不安が生まれてくる．また，同一局面の証明に対しシミュレーションを使用していても，その局面を証明するための探索木を構築しなければならない点は変わらない．そのため，同一局面が頻発する状況では，そのオーバヘッドが懸念される．以上のことから，この方法は将棋を解くために利用することは難しいと考えられる．

5. 小さな将棋の探索方法

5.1 探 索 方 針

文献 8) では，問題を木の形でとらえている．そのため，同一局面が木の中で数多く発生する．しかし，グラフでとらえた場合，同一局面は 1 つしか現れない．よって，木で解くよりも，グラフの方が引き分けを効率的に扱える．たとえば**図 4** のような探索木があった場合，グラフでとらえると**図 5** のようになる．このように，グラフでとらえた場合に調べるべきノード数は木の場合と比べ，ループに関わる局面数が多ければ多いほど少なく済む．そこで，詰将棋で多大な成果を発揮した木探索を使用して，グラフ的に引き分けを証明する方法について提案する．なお，以降ルート局面とは最終的に解を得たい問題の局面のことであり，末

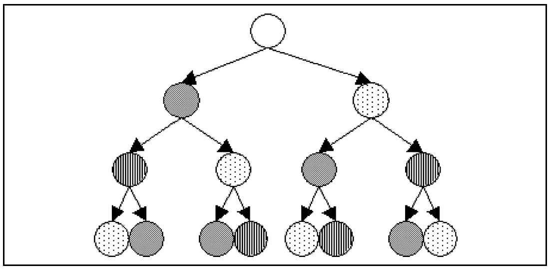

図 4 探索木の構造
Fig. 4 Structure of search tree.

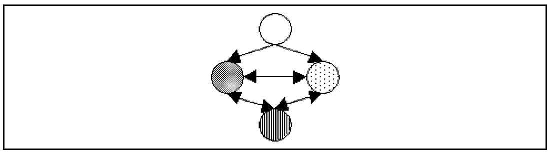

図 5 グラフの構造
Fig. 5 Structure of graph.

端局面とは，合法手が存在しない局面のことである．これらのことは探索木を縮約して得られるグラフにおいても同様である．

ループした局面を必勝でないとハッシュ表に登録し，探索を終えた際に問題となるのは，その登録した局面である．この局面以下の探索が不十分となっていることから，得られた情報が間違っている可能性が生まれ，GHI 問題が発生する．そこで，探索を終了したときにこの登録が誤っていないかを検査する．検査した結果，登録した情報がすべて正しいと判明した場合，探索で得られた解は正しいといえる．この検査のために行う探索に対し，グラフ的な考え方を使用する．この考え方が正しいことを証明するために，将棋を有向グラフとしてとらえたうえで，いくつかの定義と定理を示す．

以降の話は，先手の必勝であるか必勝でないかを判断する問題についてとする．後手の必勝か必勝でないかを判断する問題に対しては，先手と後手を入れ替えて考えればよい．

5.1.1 必勝局面，必敗局面，サイクル局面の定義

まず，次のように言葉や記号の意味を定める．

- $\boldsymbol{P}$：局面の集合
- $\boldsymbol{S}$：先手が手番の局面の集合
- $\boldsymbol{G}$：後手が手番の局面の集合
- $\boldsymbol{C}_p$：ある局面 p の子供局面の集合
- 必勝局面：先手が必ず勝つことができる手順が存在する局面
- 必敗局面：後手が必ず勝つことができる手順が存在する局面
- $\boldsymbol{W}$：必勝局面の集合
- $\boldsymbol{L}$：必敗局面の集合

将棋における末端局面は必勝局面か必敗局面のいずれかに分類される．

定義 1 必勝局面集合 $\boldsymbol{W}$ と必敗局面集合 $\boldsymbol{L}$ は次の式で定義される．ただし，候補手の存在しない末端局面における必勝局面集合と必敗局面集合をそれぞれ $\boldsymbol{W}_1$，$\boldsymbol{L}_1$ とする．ここで i を 1 以上の自然数であるとするとき，次の集合を定義する．

$$\begin{aligned}\boldsymbol{W}_{i+1} = \{x | x \in \boldsymbol{W}_i \vee \\ (x \in \boldsymbol{S} \Rightarrow \boldsymbol{C}_x \cap \boldsymbol{W}_i \neq \phi \wedge \\ x \in \boldsymbol{G} \Rightarrow \boldsymbol{C}_x \subset \boldsymbol{W}_i)\}\end{aligned}$$

$$\begin{aligned}\boldsymbol{L}_{i+1} = \{x | x \in \boldsymbol{L}_i \vee \\ (x \in \boldsymbol{S} \Rightarrow \boldsymbol{C}_x \subset \boldsymbol{L}_i \wedge \\ x \in \boldsymbol{G} \Rightarrow \boldsymbol{C}_x \cap \boldsymbol{L}_i \neq \phi)\}\end{aligned}$$

ここで，ある 1 以上の自然数 f に対し，$\boldsymbol{W}_{f+1} = \boldsymbol{W}_f$ となるとき，$\boldsymbol{W} = \boldsymbol{W}_f$ であり，$\boldsymbol{L}_{f+1} = \boldsymbol{L}_f$ となるとき，$\boldsymbol{L} = \boldsymbol{L}_f$ である．

先手番においてはすべての子供局面が必敗局面ならば，先手は必敗局面を選ばなければならないため，その局面は必敗局面となり，1 つでも必勝局面へつながる手がある場合は，先手は必勝局面を選択できるため，その局面は必勝局面となる．後手番においてはその逆で，1 つでも必敗局面があれば，必敗局面となり，すべてが必勝局面となる場合，親局面は必勝局面となる．

次に，サイクル局面という言葉を次のように定義する．

定義 2 サイクル局面の集合 $\boldsymbol{D}$ は，次の式で定義される．

$$\boldsymbol{D} = \boldsymbol{P} - \boldsymbol{W} - \boldsymbol{L}$$

サイクル局面とは，必勝局面でも必敗局面でもない局面のことである．このとき次のことが成り立つ．

定理 1 局面の分類

$$\boldsymbol{P} = \boldsymbol{W} \cup \boldsymbol{L} \cup \boldsymbol{D}$$

また，$\boldsymbol{W}$，$\boldsymbol{L}$，$\boldsymbol{D}$ は互いに素である．

証明：サイクル局面の定義より，必勝局面と必敗局面の和集合と，サイクル局面とは互いに素である．また，必勝局面と必敗局面とを同時に満たす局面が存在しないことは，それぞれの定義から明らかである☆．（証明終）

ここで，$\boldsymbol{D}$ に関する性質を証明する．

定理 2 局面集合 $\boldsymbol{D}$ は，次の性質を持つ．

$$\begin{aligned}\boldsymbol{D} = \{x | \boldsymbol{C}_x \cap \boldsymbol{D} \neq \phi \wedge \\ x \in \boldsymbol{S} \Rightarrow \boldsymbol{C}_x \subset (\boldsymbol{D} \cup \boldsymbol{L}) \wedge \\ x \in \boldsymbol{G} \Rightarrow \boldsymbol{C}_x \subset (\boldsymbol{D} \cup \boldsymbol{W})\}\end{aligned}$$

証明：先手番のサイクル局面について考える．サイクル局面が必勝局面を子局面に持つと仮定したとき，サイクル局面は必勝局面の定義を満たすことになり，サイクル局面の定義に矛盾する．よって，サイクル局面は必勝局面を子局面として持たない．また，サイクル局面の子局面がすべて必敗局面であると仮定するとき，サイクル局面は必敗局面の定義を満たすことになり，サイクル局面の定義に矛盾する．よって，サイクル局面の子局面は，必敗局面以外の局面を持つ．以上より，先手番のサイクル局面の子局面は 1 つ以上のサイクル局面を持ち，必勝局面を持たない．次に後手番について考える．サイクル局面が必敗局面を子局面に持つと仮定したとき，サイクル局面は必敗局面の定義を満たすことになり，サイクル局面の定義に矛盾する．よって，サイクル局面は必敗局面を子局面として持た

☆ 文献 4) で，局面の定義などは異なるが，将棋の結論は先手必勝，後手必勝，引き分けのいずれかただ 1 つになることが示されている．

ない．また，サイクル局面の子局面がすべて必勝局面であると仮定するとき，サイクル局面は必勝局面の定義を満たすことになり，サイクル局面の定義に矛盾する．よって，サイクル局面の子局面は，必勝局面以外の局面を持つ．以上より，後手番のサイクル局面の子局面は１つ以上のサイクル局面を持ち，必敗局面を持たない．（証明終）

サイクル局面は，先手番の場合，サイクル局面以外につながる手はすべて負ける手であるため，サイクル局面への手を選ばなければならない．逆に後手番では，必勝局面となる手ばかり存在するため，後手はサイクル局面へとつながる手を選ばなければならない．

また，先手も後手も自分が負けないように指し続けたとき，どちらも勝つことができずに局面がループしてしまう局面を引き分け局面と定義するとき，次のことが成り立つ．

定理 3 $\boldsymbol{D}$ に属する局面は引き分けとなる．

これはサイクル局面の性質などから，自明である．サイクル局面は，必勝局面や必敗局面になりえず，ループする手順を選ばざるをえないので，千日手（引き分け）となる．よってサイクル局面の性質を用いることで，木ではなくグラフで局面の引き分けを証明することができる．

5.1.2 探索を用いたサイクル局面の証明

探索中にサイクル局面を判別することは容易ではない．サイクル局面は再帰的に定義されるため，いくつかの局面を調べただけでは，判断できないからである．そこで次に，実際の木探索の中でサイクル局面を証明する方法について述べる．探索の最中に，手順の中に同一局面が２度現れたとき，この局面はサイクル局面である可能性がある．そこで，この局面をサイクル局面候補と呼び，その局面の情報を保存しておき，探索中でその局面は必敗局面として扱う．**図 6** のような部分グラフを持つ局面を考える．ここで，必勝局面と必敗局面は，最終的にその結果に至る局面であり，現時点では判明していないものとする．

ここで，図 6 におけるサイクル局面候補の登録例を**図 7** に示す．ここでは簡略化のため，サイクル局面候補の登録を先手番局面だけに限定している．サイクル局面候補は，探索中に現れる繰り返された局面を登録していくことから，探索されたノードで構成されるグラフに存在するサイクル中に必ず１つ存在するように登録される．ただし，サイクル局面の可能性がある局面を不足なく登録することから，サイクルに関わらない局面も登録される．そのほか，探索の仕方や探索順序に依存して，登録される局面は変化する．また，n 個のサイクルが絡みあう場合，n 個以下のサイクル局面候補が登録される．図の２つのサイクル局面候補 a, b は実際にサイクル局面となる局面であり，c, d は実際には先手必勝となる局面ではあるが，サイクル局面候補として登録されている．

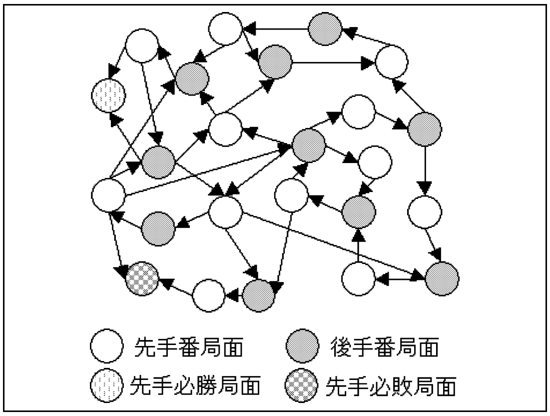

図 6 グラフの例
Fig. 6 Example of graph.

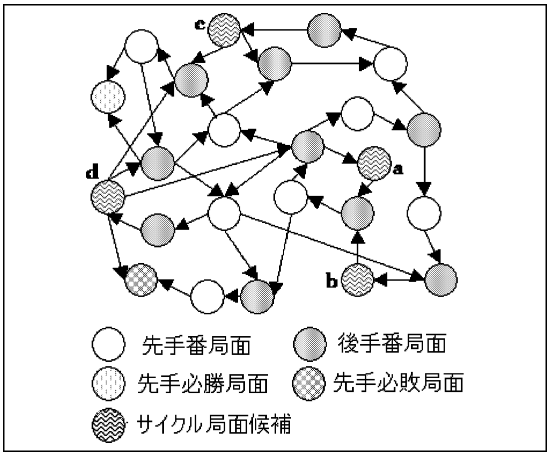

図 7 サイクル局面の例
Fig. 7 Example of cycle position.

探索が終了したときのサイクル局面候補の集合を $\boldsymbol{D}_c$ とするとき，ここでサイクル局面候補の持つ不確実性に影響を受ける局面として，サイクル局面候補影響局面を考え，サイクル局面候補影響局面の集合 $\boldsymbol{D}_e$ を次のように定義する．

定義 3 $\boldsymbol{D}_{e_1} = \phi$ とし，1 以上の自然数 f に対し，$\boldsymbol{D}_{d_f} = \boldsymbol{D}_{e_f} \cup \boldsymbol{D}_c$ であるとするとき，i が 1 以上の自然数とする次の集合を定義する．

$$\begin{aligned}\boldsymbol{D}_{e_{i+1}} = \{x | x \in \boldsymbol{D}_{e_i} \vee (&\boldsymbol{C}_x \cap \boldsymbol{D}_{d_i} \neq \phi \wedge \\ &x \in \boldsymbol{S} \Rightarrow \boldsymbol{C}_x \subset (\boldsymbol{D}_{d_i} \cup \boldsymbol{L}) \wedge \\ &x \in \boldsymbol{G} \Rightarrow \boldsymbol{C}_x \subset (\boldsymbol{D}_{d_i} \cup \boldsymbol{W}))\}\end{aligned}$$

ここで，$\boldsymbol{D}_{e_{f+1}} = \boldsymbol{D}_{e_f}$ のとき，$\boldsymbol{D}_e = \boldsymbol{D}_{e_f}$ となる．

これはサイクル局面の性質をサイクル局面候補とサイクル局面候補影響局面に置き換えたものである．また，サイクル局面候補影響局面に関する定義をもう1つ行う．

定義4 あるサイクル局面候補影響局面 $p_e \in \boldsymbol{D}_e$ に対する，影響局面集合 $\boldsymbol{E}_{p_e}$ を次のように定義する．$\boldsymbol{E} \subseteq \boldsymbol{D}_c$ を満たす，ある集合 $\boldsymbol{E}$ を考える．ここで，$\boldsymbol{D}_c$ を $\boldsymbol{E}$ に置き換えたときに，ある局面 p_e がサイクル局面候補影響局面となるような，最小の集合 $\boldsymbol{E}$ を，局面 p_e の影響局面集合 $\boldsymbol{E}_{p_e}$ と呼ぶ．

影響局面集合とは，そのサイクル局面候補影響局面の結果に影響を与えているサイクル局面候補の集合である．ここで次の定理が成り立つ．

定理4 サイクル局面候補影響局面がサイクル局面となる条件：

$$\exists p_e \in \boldsymbol{D}_e(\boldsymbol{E}_{p_e} \subseteq \boldsymbol{D}) \Rightarrow p_e \in \boldsymbol{D}$$

これはサイクル局面とサイクル局面候補影響局面の定義から自明である．したがって，サイクル局面候補すべてがサイクル局面であることを示せれば，サイクル局面候補影響局面もまたサイクル局面と判明する．図7における，サイクル局面候補影響局面は**図8**のようになる．探索中でサイクルを検出し，ノードBとノードCがサイクル局面候補として登録されている．ここで，それぞれの親局面は子供局面を1つだけ持ち，それがサイクル局面候補となっているため，探索はこの局面をサイクル局面候補影響局面と判断する．これをさらに親局面をたどり続けることで，ノードAはサイクル局面候補影響局面と判断される．ここでノードAの影響局面集合の要素は，ノードBとノードCである．ノードAの親局面は3つの子供局面を持っており，1つはサイクル局面候補，もう1つはサイクル局面候補影響局面であると判明しているが，最後の1つが判明していないため，この時点ではサイクル局面候補影響局面と判断できない．

ここで，次の定理が成り立つ．

定理5 サイクル局面の証明：$\boldsymbol{D}_d = \boldsymbol{D}_e \cup \boldsymbol{D}_c$ とするとき，次の式が成り立つ．

$$\begin{aligned}&\forall p_c \in \boldsymbol{D}_c(\boldsymbol{C}_{p_c} \cap \boldsymbol{D}_d \neq \phi \wedge \\ &x \in \boldsymbol{S} \Rightarrow \boldsymbol{C}_{p_c} \subset (\boldsymbol{D}_d \cup \boldsymbol{L}) \wedge \\ &x \in \boldsymbol{G} \Rightarrow \boldsymbol{C}_{p_c} \subset (\boldsymbol{D}_d \cup \boldsymbol{W})) \\ &\qquad \Leftrightarrow \boldsymbol{D}_d \subseteq \boldsymbol{D}\end{aligned}$$

証明：左辺の $\boldsymbol{D}_c$ の要素に関する式が満たされるとき，定義3の $\boldsymbol{D}_e$ の式を用いて，$\boldsymbol{D}_c$ と $\boldsymbol{D}_e$ の和集合を考えるとき，次の式が導かれる．

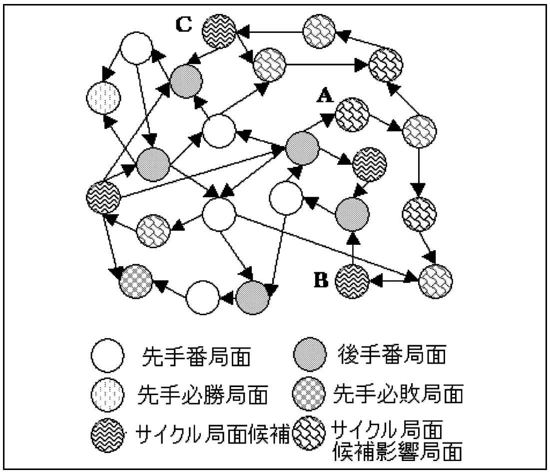

図8 サイクル局面候補影響局面の例
Fig. 8 Example of affected position by candidate of cycle position.

$$\begin{aligned}\boldsymbol{D}_d = \boldsymbol{D}_c \cup \boldsymbol{D}_e = \{x | &\boldsymbol{C}_x \cap \boldsymbol{D}_d \neq \phi \wedge \\ &x \in \boldsymbol{S} \Rightarrow \boldsymbol{C}_x \subset (\boldsymbol{D}_d \cup \boldsymbol{L}) \wedge \\ &x \in \boldsymbol{G} \Rightarrow \boldsymbol{C}_x \subset (\boldsymbol{D}_d \cup \boldsymbol{W})\}\end{aligned}$$

よって，$\boldsymbol{D}_d$ はサイクル局面の性質を満たすため，右辺は成立する．右辺が成り立つとき，左辺が成り立つのはサイクル局面やサイクル局面候補影響局面の定義から自明である．（証明終）

すべてのサイクル局面候補がサイクル局面であることが示されれば，定理4から，サイクル局面候補影響局面もすべてサイクル局面であることが分かる．よって，問題局面が必勝局面，必敗局面，サイクル局面候補，サイクル局面候補影響局面のいずれかに分類できれば，問題局面の解を得ることができる．

5.1.3 実装上でのサイクル局面の証明

実際の探索においては，サイクル局面候補は解の二値化のために必敗局面として処理していく．$\boldsymbol{D}_c$ の中に必敗局面があると判明した場合，そのサイクル局面候補を影響局面集合に含むサイクル局面候補もまた，連鎖的に必敗局面と判明する可能性がある．このとき，必敗局面と判明したサイクル局面候補をサイクル局面候補から外し，残ったサイクル局面候補だけを考えれば，これらの局面は定理5を用いてサイクル局面であることが証明できる．必敗局面であると判明したサイクル局面候補の影響によって，サイクル局面候補やサイクル局面候補影響局面は必敗局面になることはあるが，必勝局面になることはない．そのため，必勝か必勝でないかを判断するためには，サイクル局面候補がすべて必敗局面かサイクル局面であることを示せばよい．

5.2 アルゴリズムの設計

引き分けの証明には 3 章で示した方法を用いる．つまり必勝か必勝でないかを証明する探索を，先手，後手それぞれの場合について行う．必勝か必勝でないかを証明するためには，まずルート局面からの探索を実行する．

5.2.1 ルート局面からの探索

ここでは，ルート局面を開始局面として探索を実行する．探索中の手順に同一局面が現れた場合に，その局面をサイクル局面候補とする．サイクル局面候補を必敗局面として探索を行うことでループ手順を避け，ルート局面の解とサイクル局面候補のリストを得る．ここで，ルート局面が必勝局面であると判明した場合，必勝であることが判明する．必勝局面と必敗局面は，定義 1 から，それぞれ子供局面に含まれる必勝局面，必敗局面の状態にだけ依存して確定し，現時点で不確定な要素であるサイクル局面候補やサイクル局面候補影響局面の解に影響されないからである．ルート局面の解が必勝でない場合は，検査探索を実行する．

5.2.2 検査探索

検査探索とは，得られたルート局面の解を確定するために，サイクル局面候補やサイクル局面候補影響局面の解を証明する探索である．検査探索では，サイクル局面候補やサイクル局面候補影響局面の解を証明するために定理 5 を用いる．定理 5 ではサイクル局面候補影響局面やサイクル局面候補を用いて証明を行うため，これまでに得られた探索の情報をそのまま使用し，サイクル局面候補を開始局面とした証明数探索を実行すればよい．なぜならば，これまでに得られた情報はすべて必勝局面，必敗局面，サイクル局面候補，サイクル局面候補影響局面のいずれかに分類できるからである．

このとき，5.1.3 項で示したように，サイクル局面候補は必敗局面であっても問題ない．ここで，定理 5 の式において，$\boldsymbol{D}_d$ を $\boldsymbol{L}$ に置き換えた場合，左辺の式は，すべての p_c が必敗局面の定義を満たす場合に，という形に変形できる．よって，サイクル局面候補とサイクル局面候補影響局面を必敗局面として扱って探索した場合に，すべてのサイクル局面候補が必敗局面であることを示せば，サイクル局面候補とサイクル局面候補影響局面はサイクル局面もしくは必敗局面であると判明する．ルート局面はサイクル局面候補，サイクル局面候補影響局面，必勝局面，必敗局面のいずれかに判明していることから，ルート局面が必勝であるか必勝でないかが判明する．しかし，サイクル局面候補が 1 つでも必勝局面であると判明した場合，サイクル局面候補やサイクル局面候補影響局面の解は定理 5 を用いて判別できない．サイクル局面候補，サイクル局面候補影響局面の中に必勝局面が含まれてしまうからである．この場合は再探索準備処理へと移る．

5.2.3 再探索準備処理

サイクル局面候補のうちの 1 つが必勝局面であった場合，必勝局面となる可能性を持つサイクル局面候補，サイクル局面候補影響局面が生まれるため，保存された情報は誤りを含んでいる．正確には，ある局面 x の影響局面集合に属している局面（サイクル局面候補）が必勝局面である場合に，局面 x は必勝局面である可能性が生じる．そのため，必勝局面と判明したサイクル局面候補を影響局面集合の要素として持たない局面はいまだに信頼できる情報と見なし，必勝局面を影響局面集合の要素として持っている局面の情報だけ破棄すれば情報の誤りを排除できる．しかし，影響局面集合を保存しておくのは難しく，また必勝局面であると判明したサイクル局面候補を影響局面集合の中に含んでいるサイクル局面候補もまた必勝局面である可能性を有するため，連鎖的に多数のサイクル局面候補の必勝が証明されることがある．これらの処理は煩雑になり，多大な計算量を失うことが予想されるため，本論文において提案する手法では，サイクル局面候補，サイクル局面候補影響局面をすべて削除する．前述のように必勝局面，必敗局面はサイクル局面候補の影響を受けていない確定された情報であるため，削除する必要はない．保有している情報から誤りを排除したうえでルート局面からの探索に戻り，問題局面を再探索する．

5.3 実装

探索には PDS 探索を用いている．また，優越関係や証明駒[15]，反証駒[13] などの高速化技法も導入している．このアルゴリズムの擬似コードを**図 9**，**図 10** に示す．図 9，図 10 において，pds_search は PDS 探索を実行する関数である．2 つの引数をとり，1 つ目は探索を実行する局面，2 つ目は同一局面が手順の中に現れたときに，その局面をどう扱うかである．はじめに，通常の PDS 探索を用いて問題局面の探索を行う．このとき，千日手と思われる局面は先手側が負けになるとして探索する．つまり，手順の中で同一局面が現れた場合，その局面を必敗局面としてハッシュ表に登録する．同時に，その局面の情報をサイクル局面候補として別のテーブルに保存しておく．また，サイクル局面候補影響局面もしくはサイクル局面候補である局面と，そうでない局面とを分けるフラグをハッシュ表の局面情報につけておく．

```
hikiwake_pds(){
  result=hikiwake_pds_search(問題局面,
      引き分けは先手の負け);
  if(result が先手必勝ではない){
    ハッシュ表におけるサイクル局面情報を削除;
    result2=hikiwake_pds_search(問題局面,
        引き分けは後手の負け);
    if(result!=result2)result=引き分け;
    else result=後手必勝;
  }
  return result;
}
```

図 9 引き分けを証明するアルゴリズムの擬似コード 1
Fig. 9 Pseudo code of algorithm to prove draw - No.1.

```
hikiwake_pds_search(開始局面, 引き分けは X 側の負け){
  while(1){ // pds_search は PDS 探索
    // 第 1 引数は開始局面
    // 第 2 引数はループ局面をどちらの負けと扱うか
    result=pds_search(開始局面,
        引き分けは X 側の負け);
    if(開始局面がサイクル局面候補,
      もしくはサイクル局面候補影響局面である){
      for(各サイクル局面候補 A に対して){
        result2=pds_search(A, 引き分けは X 側の負け);
        if(result2 が X 側の必勝である場合)break;
      }
      if(すべてのサイクル局面候補が証明されている)
        break;
      else {
        ハッシュ表に存在する全サイクル局面候補,
          サイクル局面候補影響局面を削除;
        continue;
      }
    }
    else break;
  }
  return result;
}
```

図 10 引き分けを証明するアルゴリズムの擬似コード 2
Fig. 10 Pseudo code of algorithm to prove draw - No.2.

探索が終了したとき，問題局面の解 S を得る．解が必勝局面であった場合，この解 S はサイクル局面候補の影響を受けていないため，先手必勝が解であると判明する．問題局面がサイクル局面候補もしくはサイクル候補影響局面であった場合は検査探索を実行する．このときに使用する探索は問題局面の探索で使用したものと同一である．ただし，探索木のルート部分ではハッシュ表の情報を参照しない．ルート部分はサイクル局面候補であるため，必敗局面であるという情報が入っているからである．その結果，サイクル局面候補がすべて必敗局面であることが示された場合，サイクル局面候補影響局面は，必敗局面かサイクル局面であることが判明する．一方，サイクル局面候補が 1 つでも必勝局面であると判明した場合，サイクル局面候補，サイクル局面候補影響局面の情報をハッシュから削除したうえで問題局面を再探索し，新たな解 S を得る操作へと戻る．

最終的な結果として，先手必勝であることが示された場合は問題局面は先手必勝であると返し，後手必勝であることが示されれば問題局面は後手必勝であると返す．両者とも否定された場合，問題局面は引き分けであると返す．

5.4 アルゴリズムの動作例

図 8 を部分グラフに持つ局面の探索においてこの方法を適用し，得られた解が必勝局面，必敗局面のいずれでもない場合，存在する 4 つのサイクル局面候補に

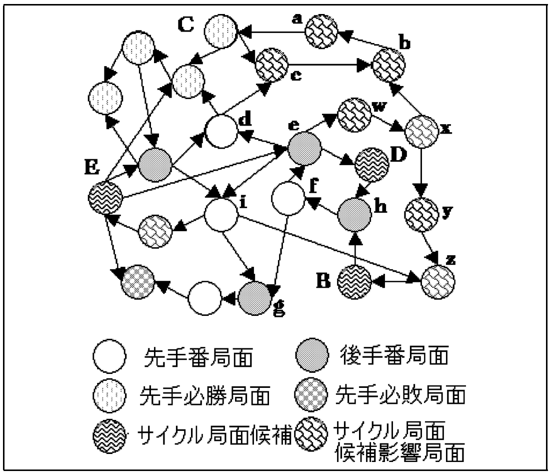

図 11　提案手法の実行例
Fig. 11　Example of execution of the proposed method.

対して検査探索を実行する．サイクル局面候補 C の検査探索を行った場合，この局面の解は先手必勝局面となることが分かる．このとき，サイクル局面候補が必敗局面でないことが明らかとなったため，再探索を実行する必要がある．この時のグラフの状態を**図 11** に示す．その後の再探索においては，局面 a，b，c や w，x，y，z などのサイクル局面候補影響局面に関する情報はすべて削除され，再探索によって新たな結果を得ることになる．このとき，先手必勝局面であると判明した局面 C の情報は保存されたままとなる．このことが，後で示すアルゴリズムの停止性に関わってくる．再探索の中で，サイクル局面候補影響局面と判断されていた局面 a，b，c は局面 C の影響を受けていた局面であったため，局面 C が先手必勝局面となるにともない，すべて先手必勝局面となる．

ここで，再探索の結果，D，B，E が再びサイクル局面候補として保存されたとする．このとき，局面 w，x，y，z はサイクル局面候補 B の影響を受けているため，サイクル局面候補影響局面のままとなる．ただしこれは，サイクル局面候補 B が再探索においても再びサイクル局面候補となったと仮定した場合であり，状況によってはサイクル局面候補にならない場合もある．また，局面 c が先手必勝局面であると判明することにより，局面 d もまた先手必勝局面となる．

ここで，サイクル局面候補 B の検査探索が行われたときを考える．局面 g が先手必敗であることが探索によって判明するとき，局面 i はサイクル局面候補影響局面となり，局面 e，f，h もまたサイクル局面候補影響局面と判明する．この結果，局面 B は唯一の子供局面 h がサイクル局面候補影響局面であると判明するため（内部的には必敗局面であると判明するため），局面 B の検査探索は終了する．局面 D も同じ局面 h を唯一の子供局面としているため，検査探索は問題なく終了する．最後の局面 E の検査探索では，局面 E が先手必勝となることが明らかなため，再探索が実行されることになる．

なお，以上で示した動作は，あくまで説明用の例であり，実際のシステムがこの部分グラフを探索した際に，同一の動作が起こることを保証してはいない．たとえば局面 E などは，再探索の結果，サイクル局面候補として登録されず，先手必勝と判明する可能性は高い．

5.5 停止性

プログラムの停止性について次のことが成り立つ．

定理 6　PDS 探索の停止性が保証されているとき，提案手法は有限回で終了する．

証明：プログラムに存在するループは PDS 探索内と図 10 の hikiwake_pds_search 関数内の while 文と for 文だけである．PDS 探索と hikiwake_pds_search 関数内の for 文に関しては，局面の数が有限であることと，PDS 探索の停止性が保証されているという仮定から，無限にループすることはない．ここで，while の終了条件はすべてのサイクル局面候補が必勝局面ではないと証明されるか，問題局面が必勝局面か必敗局面のいずれかに判明することである．ここで，問題局面をルート局面とする探索木において，さまざまな状態における，問題局面から必勝局面，必敗局面のいずれも含まない手順で到達できる，必勝局面にも必敗局面にも分類されていない局面の数の集合を $\boldsymbol{A}$ とする．このとき，$\boldsymbol{A}$ は 0 以上の自然数で構成される集合となることから，$\boldsymbol{A}$ は順序集合である．また，局面の総数は有限であることから，$(\boldsymbol{A},<)$ はその要素を用いた無限に減少する列を作れない．ここで，j 回目の while 文のループにおいて必勝局面，必敗局面に分類されていない局面数を返す関数を Ψ^j とするとき，Ψ^j が 0 となったときに確実にループは終了する．問題局面が必勝局面，必敗局面のいずれかに分類されるからである．ここで，hikiwake_pds_search 関数内の for 文の終了条件は，すべてのサイクル局面候補が必勝局面でないと証明されるか，いずれかのサイクル局面候補が必勝局面であると証明されることである．前者は while 文の終了条件となるため，プログラムは終了する．後者においては，確実に 1 つ以上のサイクル局面候補，もしくはサイクル局面候補影響局面が必勝局面であると判明する．サイクル局面候補やサイクル局面候補影響局面は PDS 探索によって判別されるため，PDS 探

索の性質から問題局面から必勝局面，必敗局面を経ずにたどれる手順を持っている．for 文はループが繰り返されるたびに必ず実行されることから，$\Psi^{j+1} < \Psi^{j}$ となる．終了条件に関連付けられた関数 Ψ がループごとに単調減少し，無限の減少列を作れないことから，提案手法は有限回で終了する．（証明終）

5.6 特徴と欠点

この探索手法の重要な点は，サイクル局面の証明を行う際に，それまでに得たサイクル局面候補影響局面の情報をそのまま使用できる点である．これにより，再探索が起こらない場合，探索木を証明するのに比べ，ループの証明が格段に速くなる．また，ループする局面の探索は 1 度サイクル局面候補に登録されれば，再探索する場合とサイクル局面の証明をする際を除き，その局面以下を探索する必要はない．また，同一局面の情報を複数保存する必要や，手順を保存する必要がないため，メモリの利用効率が良い．また，ループを複数のサイクル局面候補やサイクル局面候補影響局面で分割して証明を行うため，1 つの探索における深さを低く抑え，メモリを効率的に使用することができる．より大きな盤面の解を求めることを考えた場合，膨大になる局面数を処理するために，メモリの利用効率は重要な要素となる．

一方で，再探索が大量に発生した場合における，探索時間の増加が欠点となる．ただし，将棋を解く場合は詰将棋を解くことと異なり，必ずしも高速に解が判明する必要はない．実戦で使用するよりも，問題の性質を調べる意味合いが強いからである．また，将棋を解く場合は詰将棋を解く場合に比べ，ループする局面がきわめて多いと予想される．そのため，詰将棋においては非効率的になる可能性があるが，将棋を解く場合においては，ループの処理に強い本手法は，単純な木探索よりも解が早く得られると予想される．

6. 3x4 の解を得る実験

6.1 実 験 環 境

実験環境を次に示す．

- ハッシュサイズ：2^{22}
- ハッシュはオープンアドレス法
- 1 問あたりの制限時間：10 分
- 深さは 2^{10} に達した場合に終了
- サイクル局面候補の登録の最大数：400,000
- Pentium4 3.2 GHz，メモリ 2 GB

時間や深さを小さく制限しているのは，ある程度のリソース制限化でどの程度問題が解けるのかを示すためである．この結果は，3x4 に限らずに，より大きな問題に対する探索の傾向を推し量る要素となる．探索時間や探索深度は，通常の詰将棋を解くうえで十分な量として設定している．また，ハッシュのガーベジコレクションについては行っていない．

証明数探索ではループによる証明数の無限カウント問題があることが知られている[7]．手順の中で同一局面が 2 度現れるとき，最初に現れた局面と後で現れた局面とで多重に証明数をカウントすることで，無限に証明数が膨れ上がることがある．その結果，駒の少ない簡単と思われる局面においても解が得られない場合があった．文献 7) ではその問題に対する解決法が示されている．具体的には，局面が現れる最短手順の長さを保存しておき，それより深い位置で現れた同一局面は，その親の局面の証明数計算に加えないという方法である．これにより，証明数や反証数の増加を抑えることができるが，値が小さくなった分，探索が深くなりやすいという問題がある．先述のように，将棋を解く場合，探索の深さが 200,000 を超えることがある．これだけ深い探索を実行する場合，メモリの圧迫が生まれるため好ましくない．そこで，証明数の無限カウント対策を，探索開始直後から行わずある程度経ってから行うようにした．PDS 探索は証明数と反証数の閾値を持っており，探索の進行度を表す基準となりうるので，その閾値があらかじめ設定した無限カウント対策実行閾値を超えた場合に証明数の無限カウント対策を行う．無限カウント対策実行閾値は，探索木によって最適値は異なるが，ここでは 100,000 と設定した．

6.2 3x4 の解を得る実験の結果

実験の全結果を**表 1** に示す．結果について先手必勝数などを集計したものを**表 2** に示す．表 1 において，No. とは問題に一意に振られた番号のことであり，再探とは再探索が行われた回数であり，再探時間とは再探索にかかっている時間（ms）である．

実験の結果，87%の局面について解を得ることができた．文献 2) では，2 問後手必勝があると示されているが，実際にはその後手必勝は引き分けであると判明した．この食い違いは文献 2) の間違いであると考えられる．文献 2) で示されているアルゴリズムで実験したところ，後手必勝となる問題の解を得ることはできなかった．今回の実験結果では後手必勝となる局面は判明していない．引き分けに関しては，4 局面しか発見されていなかったが，73 局面に関して引き分けを証明することができた．引き分けと判断された局面におけるサイクル局面の証明に関しては，再探索の回数は平均で約 8.00 回であり，サイクル局面候補の検査回数の平均は 1 問あたり約 747 回，最大値の平均

表 1　3x4 将棋の実験結果
Table 1　Detail of experiment about 3x4 Shogi.

No.	段	駒配置	結果	時間：ms	再探	再探時間
1	1	飛玉－	引分	5201	11	4368
2	1	角玉－	引分	1286	8	988
3	1	角玉飛	先必	92852	0	0
4	1	金玉－	引分	1282	6	646
5	1	金玉飛	先必	394	0	0
6	1	金玉角	先必	11640	0	0
7	1	金玉金	引分	5239	7	2636
8	1	銀玉－	引分	10109	11	8870
9	1	銀玉飛	先必	10565	0	0
10	1	銀玉角	不明	708847	13	550383
11	1	銀玉金	不明	61180	8	56268
12	1	銀玉銀	引分	1727	5	161
13	1	桂玉－	先必	432	2	0
14	1	桂玉飛	先必	1178	0	0
15	1	桂玉角	引分	89650	5	89576
16	1	桂玉金	先必	177	2	0
17	1	桂玉銀	不明	597295	6	1078066
18	1	桂玉桂	先必	0	0	0
19	1	香玉－	先必	14	0	0
20	1	香玉飛	先必	46	0	0
21	1	香玉角	先必	1351	0	0
22	1	香玉金	先必	150	0	0
23	1	香玉銀	先必	43	0	0
24	1	香玉桂	先必	7402	0	0
25	1	香玉香	先必	174	0	0
26	1	歩玉－	引分	7256	14	6607
27	1	歩玉飛	先必	3875	0	0
28	1	歩玉角	引分	166884	10	19292
29	1	歩玉金	先必	4389	0	0
30	1	歩玉銀	先必	18201	6	27342
31	1	歩玉桂	先必	8538	0	0
32	1	歩玉香	先必	783	0	0
33	1	歩玉歩	先必	6577	0	0
34	2	飛玉－	引分	723	5	378
35	2	角玉－	引分	2081	7	948
36	2	角玉飛	先必	1	0	0
37	2	金玉－	引分	1277	6	646
38	2	金玉飛	先必	146	0	0
39	2	金玉角	先必	3951	0	0
40	2	金玉金	引分	5240	7	2637
41	2	銀玉－	引分	12538	11	10557
42	2	銀玉飛	先必	155	0	0
43	2	銀玉角	先必	18463	0	0
44	2	銀玉金	不明	71383	4	58794
45	2	銀玉銀	引分	2067	5	392
46	2	桂玉－	先必	3	0	0
47	2	桂玉飛	引分	7611	7	3103
48	2	桂玉角	先必	44	0	0
49	2	桂玉金	引分	418	4	92
50	2	桂玉銀	先必	307	0	0
51	2	桂玉桂	先必	0	0	0
52	2	香玉－	先必	7	0	0
53	2	香玉飛	先必	55	0	0
54	2	香玉角	先必	2	0	0
55	2	香玉金	先必	63	0	0
56	2	香玉銀	先必	66	0	0
57	2	香玉桂	先必	777	0	0
58	2	香玉香	先必	147	0	0
59	2	歩玉－	先必	5	0	0
60	2	歩玉飛	先必	8815	0	0
61	2	歩玉角	先必	51	0	0
62	2	歩玉金	先必	1744	0	0
63	2	歩玉銀	先必	434	0	0
64	2	歩玉桂	先必	10	0	0
65	2	歩玉香	先必	606	0	0
66	2	歩玉歩	先必	344	0	0
67	3	飛玉－	引分	3743	7	2577
68	3	角玉－	引分	4274	7	3196
69	3	角玉飛	先必	0	0	0
70	3	金玉－	引分	1282	6	649
71	3	金玉飛	先必	178	0	0
72	3	金玉角	先必	91170	4	39694
73	3	金玉金	引分	5261	7	2637
74	3	銀玉－	引分	9284	7	4492
75	3	銀玉飛	先必	226	0	0
76	3	銀玉角	先必	181811	2	0
77	3	銀玉金	不明	132974	4	132686
78	3	銀玉銀	不明	41554	5	10481
79	3	桂玉－	先必	2	0	0
80	3	桂玉飛	引分	5808	5	2722
81	3	桂玉角	先必	30	0	0
82	3	桂玉金	引分	419	4	93
83	3	桂玉銀	不明	600000	4	158798
84	3	桂玉桂	先必	0	0	0
85	3	香玉－	引分	5872	9	4624
86	3	香玉飛	先必	50	0	0
87	3	香玉角	先必	3	0	0
88	3	香玉金	先必	82	0	0
89	3	香玉銀	先必	105	0	0
90	3	香玉桂	先必	614	0	0
91	3	香玉香	先必	184	0	0
92	3	歩玉－	引分	2788	6	1227
93	3	歩玉飛	引分	89559	12	65122
94	3	歩玉角	先必	523	0	0
95	3	歩玉金	不明	61109	8	49661
96	3	歩玉銀	不明	570724	10	950982
97	3	歩玉桂	引分	763	4	115
98	3	歩玉香	先必	1157	0	0
99	3	歩玉歩	先必	518	0	0
100	1	玉－角	引分	2341	7	1984
101	1	玉－金	引分	2906	8	1840
102	1	玉－銀	引分	16812	13	14698
103	1	玉－桂	先必	654	2	0
104	1	玉－歩	引分	7998	13	6623
105	1	玉飛－	引分	5932	10	5017
106	1	玉飛角	不明	173554	16	157668
107	1	玉飛金	先必	49134	2	0
108	1	玉飛銀	引分	111232	12	38501
109	1	玉飛桂	不明	204428	7	124636
110	1	玉飛歩	引分	249752	17	231882
111	1	玉角－	先必	55	0	0
112	1	玉角金	先必	14	0	0
113	1	玉角銀	先必	21	0	0
114	1	玉角桂	先必	55	0	0
115	1	玉角歩	先必	5361	2	0
116	1	玉金－	引分	2212	5	650
117	1	玉金角	引分	34876	7	13532
118	1	玉金金	引分	5404	5	1726
119	1	玉金銀	不明	61312	8	51524
120	1	玉金桂	先必	41	0	0
121	1	玉金歩	先必	10686	4	7904
122	1	玉銀－	引分	14056	11	10895
123	1	玉銀角	不明	310459	14	607362
124	1	玉銀金	不明	96012	10	127374
125	1	玉銀銀	引分	3913	5	656
126	1	玉銀桂	不明	600000	8	1196000
127	1	玉銀歩	先必	19867	4	33382
128	1	玉桂－	先必	0	0	0
129	1	玉桂角	引分	193767	5	7423
130	1	玉桂金	先必	23	0	0
131	1	玉桂銀	不明	600001	12	1199632
132	1	玉桂桂	先必	0	0	0
133	1	玉桂歩	先必	0	0	0
134	1	玉香－	引分	5095	6	2796
135	1	玉香角	不明	600001	28	1163908
136	1	玉香金	先必	16661	0	0
137	1	玉香銀	先必	17086	2	0
138	1	玉香桂	先必	20208	0	0
139	1	玉香歩	不明	600001	6	1178657
140	1	玉歩－	引分	9274	19	7614
141	1	玉歩角	引分	62575	9	57791
142	1	玉歩金	引分	224230	10	212062
143	1	玉歩銀	不明	661846	8	305367
144	1	玉歩桂	不明	600001	12	1141512
145	1	玉歩歩	不明	161158	18	317986
146	2	玉－角	引分	2559	7	1210
147	2	玉－金	引分	2903	8	1839
148	2	玉－銀	引分	19825	13	16669
149	2	玉－桂	先必	4	0	0
150	2	玉－歩	先必	2	0	0
151	2	玉飛－	引分	3027	8	2159
152	2	玉飛角	先必	14127	0	0
153	2	玉飛金	先必	46568	2	0
154	2	玉飛銀	先必	59939	2	0
155	2	玉飛桂	先必	538	0	0
156	2	玉飛歩	引分	80247	15	71004
157	2	玉角－	先必	19	0	0
158	2	玉角金	先必	10	0	0
159	2	玉角銀	先必	14	0	0
160	2	玉角桂	先必	1	0	0
161	2	玉角歩	先必	172	0	0
162	2	玉金－	引分	2239	5	650
163	2	玉金角	先必	59333	0	0
164	2	玉金金	引分	5404	5	1726
165	2	玉金銀	不明	128864	8	180774
166	2	玉金桂	引分	419	4	93
167	2	玉金歩	先必	5963	0	0
168	2	玉銀－	引分	14755	13	12839
169	2	玉銀角	不明	126088	0	0
170	2	玉銀金	不明	122701	12	155863
171	2	玉銀銀	引分	12874	6	1717
172	2	玉銀桂	先必	4	0	0
173	2	玉銀歩	先必	305	0	0
174	2	玉桂－	先必	1	0	0
175	2	玉桂角	先必	12622	2	0
176	2	玉桂金	先必	43	0	0
177	2	玉桂銀	不明	23087	5	7500
178	2	玉桂桂	先必	9	0	0
179	2	玉桂歩	先必	10	0	0
180	2	玉香－	先必	116	0	0
181	2	玉香角	先必	5612	0	0
182	2	玉香金	先必	9763	0	0
183	2	玉香銀	先必	9023	0	0
184	2	玉香桂	先必	1295	0	0
185	2	玉香歩	先必	1069	0	0
186	2	玉歩－	先必	198	0	0
187	2	玉歩角	引分	38813	9	25010
188	2	玉歩金	不明	497455	34	953448
189	2	玉歩銀	引分	45082	8	28598
190	2	玉歩桂	不明	534852	6	831946
191	2	玉歩歩	引分	43486	6	41317
192	3	玉－角	引分	6828	9	5072
193	3	玉－金	引分	2902	8	1839
194	3	玉－銀	引分	13308	8	8839
195	3	玉－桂	先必	3	0	0
196	3	玉－歩	引分	2239	6	409
197	3	玉飛－	引分	3844	6	2044
198	3	玉飛角	先必	1858	0	0
199	3	玉飛金	先必	429	0	0
200	3	玉飛銀	先必	635	0	0
201	3	玉飛桂	先必	319	0	0
202	3	玉飛歩	先必	1698	0	0
203	3	玉角－	先必	4	0	0
204	3	玉角金	先必	7	0	0
205	3	玉角銀	先必	13	0	0
206	3	玉角桂	先必	4	0	0
207	3	玉角歩	先必	0	0	0
208	3	玉金－	引分	2234	5	667
209	3	玉金角	先必	105034	4	46712
210	3	玉金金	引分	5405	5	1728
211	3	玉金銀	不明	117668	6	117642
212	3	玉金桂	引分	418	4	94
213	3	玉金歩	先必	29373	4	18714
214	3	玉銀－	引分	17591	11	15428
215	3	玉銀角	先必	221412	2	0
216	3	玉銀金	不明	306894	4	514998
217	3	玉銀銀	引分	22619	6	10069
218	3	玉銀桂	引分	1054	4	135
219	3	玉銀歩	不明	600002	6	1037702
220	3	玉桂－	先必	1	0	0
221	3	玉桂角	先必	6233	0	0
222	3	玉桂金	先必	44	0	0
223	3	玉桂銀	先必	67	0	0
224	3	玉桂桂	先必	8	0	0
225	3	玉桂歩	先必	13	0	0
226	3	玉香－	引分	9101	11	7289
227	3	玉香角	先必	3538	0	0
228	3	玉香金	先必	4460	0	0
229	3	玉香銀	先必	3004	0	0
230	3	玉香桂	先必	54	0	0
231	3	玉香歩	先必	2377	0	0
232	3	玉歩－	引分	2156	6	758
233	3	玉歩角	不明	371884	6	324488
234	3	玉歩金	引分	56928	10	27645
235	3	玉歩銀	引分	505566	10	335064
236	3	玉歩桂	先必	58	0	0
237	3	玉歩歩	先必	10034	0	0

は約 204 回であった．不明局面と判断された原因の内訳は，時間超過が 7 問，サイクル局面候補登録数の超過が 3 問，探索深度の超過が 21 問であった．たいていの詰将棋では十分に解を得ることができる探索深度を用意しても，解けない問題が 21 問も存在していることが分かる．

6.3　不明局面の再実験

不明局面の解を得るために再実験を行った．その際

表 2 3x4 将棋の実験結果

Table 2 Result of experiment about 3x4 Shogi.

問題数	先手必勝	後手必勝	引き分け	不明
237	133	0	73	31

に行った変更は次の 4 つである．

- 探索時間，最大探索深度の増加
- 再探索時の未証明局面の証明数，反証数削除
- 検査探索の並列化
- 優越関係の一部不使用

2 つ目の処理は，探索中に証明数や反証数が増加しすぎるのを防ぐためである．3 つ目の並列化に関して，そもそも先述の実験では，サイクルの証明の際，サイクル局面候補がサイクル局面でないと判明した時点で再探索処理に移っている．この場合，探索がサイクル局面候補に満遍なく行きわたらず，サイクル局面候補の登録が偏り，サイクル局面候補の登録数が膨大になったり，探索が深くなりすぎたりすることがあった．これを避けるために，問題によっては PDS 探索の閾値の変更ごとにサイクル局面候補の検査対象を切り替えることで，並列的に探索を行っている．

4 つ目は優越関係[15] の適用の制限である．優越関係とは，同一盤面で持ち駒が異なる局面を比較し，その優越性を用いて，証明数，反証数の再利用を行う方法である．盤面が同一で持ち駒が異なる A，B の 2 局面があり，A の攻め側の持ち駒が，B の攻め側の持ち駒を真に包含する場合，A は B を優越しているという．このとき，B が攻め側の勝利となるならば，A もまた攻め側の勝利となる．また，B の証明数は A の証明数以上になることから，B の証明数を参照する際に，B を優越する局面についても調べ，その最大値を B の証明数とする．反証数についても同様の理論が適用できる．

しかし，必勝必敗でない情報を使用する場合，証明数の無限カウントが発生する可能性がある．手順の中で A の後で B が現れた場合，B は A の証明数を参照してしまうため，B と A の訪問を繰り返すことで，B 以下は探索されずに証明数が上昇し続けてしまう．詰将棋においても同様の問題は発生すると思われるが，この状況が発生するのは，駒をただ捨てした場合であり，この部分の探索は詰めに影響する手順にはならないため問題が発生しないものと思われる．しかし，引き分けを証明するためには，B 以下の探索も必要になる場合がある．そこで，必勝必敗の判明していない局面の優越関係を利用しない設定についても一部の問題に対して行っている．優越関係を利用しないという設定は，6.2 節でも適用できるが，実験した結果得られた解の個数は減少した．この方法はあくまで証明数の無限カウントを避けるものであり，優越関係の持つ利点の恩恵を受けられなくなるため，探索の性能は低下してしまう．よって，無限カウントの影響で解くことができない問題に対してだけ使用するほうが望ましいといえる．

表 3 3x4 将棋の追加実験結果

Table 3 Result of additional experiment about 3x4 Shogi.

No.	段	駒配置	結果	ノード	時間 (s)	再探	再探時間
10	1	銀 玉 角	引分	30909738	4540.4	28	3262.2
11	1	銀 玉 金	引分	21338655	6477.5	32	5118.2
17	1	桂 玉 銀	引分	16480505	5299.0	10	4957.2
44	2	銀 玉 金	引分	18399717	3264.3	23	2644.7
77	3	銀 玉 金	引分	10798276	10647.8	21	9538.4
78	3	銀 玉 銀	引分	2032084	1937.7	6	10.9
83	3	桂 玉 銀	引分	37263843	3989.0	30	3154.7
95	3	歩 玉 金	引分	1006724	83.7	13	32.4
96	3	歩 玉 銀	引分	54117302	9775.5	28	6799.7
106	1	玉 飛 角	引分	4329485	2966.9	14	2839.8
109	1	玉 飛 桂	引分	857221	79.8	9	78.9
119	1	玉 金 銀	引分	18079496	4371.3	18	3037.3
123	1	玉 銀 角	引分	86060953	9862.6	81	8828.4
124	1	玉 銀 金	引分	49954877	13849.2	32	13816.3
126	1	玉 銀 桂	引分	25841883	5433.3	38	5431.0
131	1	玉 桂 銀	引分	72487	5.6	4	0.1
135	1	玉 香 角	引分	7197196	586.9	19	562.1
139	1	玉 香 歩	引分	51683749	7571.3	44	6762.0
143	1	玉 歩 銀	引分	20194880	1661.4	12	1137.9
144	1	玉 歩 桂	引分	26434769	2014.5	78	1983.5
145	1	玉 歩 歩	引分	2795154	182.6	10	177.8
165	2	玉 金 銀	引分	34416710	5986.5	29	5539.9
169	2	玉 銀 角	引分	1971004	236.4	6	17.0
170	2	玉 銀 金	引分	19445292	3358.4	28	2702.2
177	2	玉 桂 銀	引分	27587759	6370.1	24	2427.3
188	2	玉 歩 金	引分	556300	44.7	15	23.8
190	2	玉 歩 桂	引分	24422053	1921.5	55	1364.4
211	3	玉 金 銀	引分	16261065	12067.0	31	12000.0
216	3	玉 銀 金	引分	6993226	4360.3	13	4297.2
219	3	玉 銀 歩	引分	156496834	15799.2	55	14374.7
233	3	玉 歩 角	先必	20354974	991.7	4	462.4

表 4 3x4 将棋の最終結果

Table 4 Final result about 3x4 Shogi.

問題数	先手必勝	後手必勝	引き分け
237	134	0	103

新たに得られた結果を**表 3** に示す．なお，この結果は解くことを目的に実験されているため，条件を問題ごとに変更しており，ノード数や時間の単純な比較は意味を持たない．同時に，これらの問題が必ずしも難解な問題であるというわけではない．

実験の結果，不明であった 31 題の解を得た．結果は 1 問が先手必勝であり，残りはすべて引き分けであった．最終的な結果を**表 4** に示す．

7. 考　察

3x4 将棋の解は半数以上が先手必勝となり，後手必勝は存在しなかった．3x4 のような盤面では先手が一手で後手の動きを抑えることができるためと考えられる．

実験の結果，判明した引き分けの数は大幅に増大した．文献 2) で $\alpha\beta$ 探索を用いた探索では判明しなかった問題についても，問題によっては数秒から数十秒で

解を得ている．この探索が引き分けを効率的に発見していることが分かる．再探索に関しては平均では約 8.00 回となっており，少ない回数で探索を終了しているが，再実験で解けた問題の場合はさらに多い．再探索が多くなると，探索量も多くなることに加え，ハッシュ表からデータを削除する処理もまた大きくなる．再探索の数をいかに減らせるかが，重要となるだろう．再探索の回数を低く抑えられるならば，詰将棋においても有効に働く可能性が高い．

再探索の量を下げるほかの方法としては，再探索時に破棄するサイクル局面候補やサイクル局面候補影響局面の再利用についても考えられる．あるサイクル局面候補の解が間違っていたと判明しても，それに影響されていないサイクル局面候補影響局面などは解の変更が起こりえない．ただし，それらを選び出して残す方法は決して簡単ではなく，逆に処理が重くなることも予想される．

最初の実験で不明局面となってしまった原因の大半は，探索の深さが深くなりすぎていることであった．これは，証明数の無限カウント対策が探索の深さを増やしてしまっただけでなく，サイクル局面を不詰としてしまっていることもその原因として考えられる．サイクル局面が多く現れる手順を証明しやすいと判断し，より深く探索してしまっている可能性がある．より大きな盤面の問題を解こうとする場合，探索の深さを可能な限り浅く保つことが重要となるだろう．できるかぎり浅い位置でサイクル局面候補を登録できるようにする方法について考えていくべきであろう．

また，より深く探索してしまっていることの弊害は，先手必勝の証明にも悪影響を及ぼしている．たとえば，角玉飛の配置で成れる段が 1 段目だけの設定のとき (問題番号 3)，先手必勝の証明に 90 秒以上を要している．この問題は文献 2) では数秒で解を得ている．探索がループによる結果に影響され，結果に関わらない探索に時間を費やしてしまっていることが原因と考えられる．引き分けを証明することに特化した探索手法ではあるため，やむをえない面もある．しかし，近年新規節点の証明数，反証数の値を探索と評価関数を用いて予測する方法が，有効な成果をあげており[5)]，これらの方法を用いることで，改善できると思われる．探索の初期ではループの処理を行わない方法などの対策も有効であると考えられる．

8. おわりに

本論文では，3x4 将棋の解を得ることを目的とし，引き分けの証明に特化した形で，メモリの効率を良く探索する証明アルゴリズムを提案し実験を行った．結果，全 237 題すべての解を得ることができた．その内訳は先手必勝が 134，引き分けが 103 であった．今後の展望としては，考察で示した修正案の適用，王手千日手への対応，4x4 将棋への適用や実際に親しまれている 5x5 将棋への適用などがあげられる．また，既存の GHI 問題対処法にこの考え方を組み込むことで，探索性能の向上を図ることも有効であると考えられる．

参考文献

1) Breuker, D.M., Herik, H.J.v.d., Uiterwijk, J.W.H.M. and Allis, L.V.: A solution to the GHI problem for best-first search, *Theoretical Computer Science*, Vol.252, No.1-2, pp.121–149 (2001).

2) 後藤智章，柴原一友，乾 伸雄，小谷善行：小さな将棋の解，第 8 回ゲームプログラミングワークショップ，pp.25–32 (2003).

3) 橋本 剛，作田 誠，飯田弘之：完全に解を保証する必死探索について，第 6 回ゲームプログラミングワークショップ，pp.1–8 (2001).

4) 林忠一郎：将棋の結論とグラフ理論，コンピュータ将棋協会誌，Vol.10, pp.70–75 (1997).

5) 金子知適，田中哲郎，川口和紀，川合 慧：新規接点で固定深さの探索を併用する df-pn アルゴリズム，第 10 回ゲームプログラミングワークショップ，pp.1–8 (2005).

6) Kawano, Y.: Using similar positions to search game trees, *Games of No Chance*, Vol.29, pp.193–202 (1996).

7) Kishimoto, A. and Müller, M.: Df-pn in go:An application to the one-eye problem, *Advances in Computer Games*, Vol.10, pp.125–141 (2003).

8) Kishimoto, A. and Müller, M.: A solution to the ghi problem for depth-first proof-number search, *7th Joint Conference on Information Sciences*, pp.489–492 (2003).

9) Kishimoto, A. and Müller, M.: A Solution to the GHI Problem for Depth-First Proof-Number Search (Long version), *Information Sciences*, Vol.175, pp.296–314 (2005).

10) 松原 仁：将棋の数学的性質，将棋とコンピュータ，pp.1–16 (1994).

11) Nagai, A.: A new AND/OR Tree Search Algorithm Using Proof Number and Disproof Number, *Complex Games Lab Workshop*, pp.40–45 (1998).

12) 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 (2003).

13) 長井 歩，今井 浩：df-pn アルゴリズムの詰将

棋を解くプログラムへの応用，情報処理学会論文誌，Vol.43, No.6, pp.1769–1777 (2002).

14) Schaeffer, J., Björnsson, Y., Burch, N., Kishimoto, A., Müller, M., Lake, R., Lu, P. and Sutphen, S.: Solving Checkers, *IJCAI-05*, pp.292–297 (2005).

15) 脊尾昌宏：詰将棋を解くアルゴリズムにおける優越関係の効率的な利用について，第 5 回ゲームプログラミングワークショップ，pp.129–136 (1999).

16) 柴原一友，但馬康宏，小谷善行：小さな将棋の解を得る手法の提案，第 10 回ゲームプログラミングワークショップ，pp.134–147 (2005).

17) Herik, H.J.v.d., Uiterwijk, J.W.H.M. and van Rijswijck, J.: Games Solved Now and in the Future, *Artificial Intelligence Journal*, Vol.134, pp.277–312 (2002).

18) Werf, E.C.D.v.d., Herik, H.J.v.d. and Uiterwijk, J.W.H.M.: Solving Go on small boards, *International Computer Games Association Journal*, Vol.26, No.2, pp.92–107 (2003).

(平成 18 年 1 月 31 日受付)
(平成 18 年 9 月 14 日採録)

柴原　一友（学生会員）

2004 年東京農工大学大学院工学府博士前期課程修了．現在，同大学院工学府博士後期課程在学中．ゲームに関連した人工知能の研究に従事．

但馬　康宏（正会員）

1996 年電気通信大学大学院電気通信学研究科博士前期課程修了．同年石川島播磨重工業（株）入社．2001 年電気通信大学大学院博士後期課程修了．同年東京農工大学工学部助手．現在に至る．博士（工学）．文法推論，計算学習理論，およびその応用の研究に従事．電子情報通信学会，人工知能学会各会員．

小谷　善行（正会員）

東京農工大学大学院教授．共生科学技術研究院システム情報科学部門・情報工学科．人工知能，知識処理，自然言語処理，知識獲得，ゲームシステム，ソフトウェア工学，教育工学の研究に従事．電子情報通信学会，人工知能学会等会員．コンピュータ将棋協会副会長．最近では，多量データからの知識獲得に興味を持っている．