不確定性を持つ問題を解くためのAND/OR木探索——衝立詰将棋を題材として

作田 誠† 飯田 弘之††

不完全情報2人ゲームである衝立将棋の詰め問題を題材として，不確定性を持つ問題をAND/OR木探索として決定論的に解く手法を導入する．このAND/OR木は詰将棋など通常の完全情報2人ゲーム問題において生成されるものに比べてより一般化されており，節点はそれぞれ不確定性を持つ．探索アルゴリズムとして，局面表を活用した全幅深さ優先反復深化探索（ID），および証明数探索の深さ優先版の一種であるPDSを実装し動作を調べた．またPDSを不確定性のあるAND/OR木の探索に特化させたUPDSを開発しパラメータを変化させて動作を調べた．さらに，UPDSにおいて実行時にパラメータを順次変化させて解けるまで繰り返し探索を行うdpUPDS，および，UPDS，IDにおいて不確定性の探索閾値を小さな値から順次増加させて解けるまで繰り返し探索を行うuidUPDS，uidIDというバリエーションを考案・実装し動作を調べた．dpUPDS，uidUPDS，uidIDは解答能力が高く，特にuidUPDSはすべての問題を解いた．しかし，確実に最適解を得る効率の良い探索法は課題として残る．

AND/OR-tree Search for Solving Problems with Uncertainty——A Case Study Using Screen-shogi Problems

Makoto Sakuta† and Hiroyuki Iida††

This paper explores a deterministic approach to solving problems with uncertainty, using Screen-shogi problems, which are the mating problems of Kriegspiel-like Shogi variant. Our programs resolve a search space into an AND/OR tree and solve a problem by searching the tree. The AND/OR tree, each of which node has an uncertainty, is more generic than that in solving a problem of a two-person complete-information game such as Tsume-Shogi. The search algorithms are implemented using a full-width depth-first iterative deepening (ID), and using PDS, which is one of the depth-first variants of the proof-number search. In addition, UPDS, which is a specialized version of PDS for an AND/OR tree with uncertainty, is proposed and examined with changing the parameters. Moreover, some search variations are developed and examined. One variation is dpUPDS, which performs iteration with changing the parameters of UPDS. The others are uidUPDS based on UPDS and uidID based on ID. Both variations perform iteration with increasing the search threshold of uncertainty. The solving abilities of dpUPDS, uidUPDS and uidID are high. Especially, the uidUPDS program has solved all problems. However, an efficient algorithm that always finds an optimal solution even for a hard problem still needs to be investigated.

1. はじめに

ゲームやパズルを題材とした探索による問題解決は人工知能研究の重要分野で，とりわけ，詰将棋問題を題材としたAND/OR木探索アルゴリズムの進歩は顕著である．代表的な例として，野下による深さ優先反復深化法のプログラムT2によって短手数問題では人間エキスパートの能力を陵駕し，伊藤・河野・野下らの最良優先探索プログラムによって100手を超す長手数の問題も解けるようになった[5),9)]．最近では脊尾あるいは長井による局面表を活用した証明数探索の深さ優先版アルゴリズムによって，500手以上の超長手数の難問題をも非常に少ない探索節点数と短時間で解けるようになった．

しかし，これまでのAND/OR木探索の応用分野はほとんどが完全情報2人ゲームを題材とする問題（以下，完全情報2人ゲーム問題と表す）であった．不完全情報ゲームの問題解決は一般に完全情報ゲームの問題解決より複雑であり，新たなチャレンジとして注目

† 静岡大学大学院理工学研究科
Graduate School of Science and Engineering, Shizuoka University
†† 静岡大学情報学部
Faculty of Information, Shizuoka University

されている．今回我々は詰将棋の玉方の応手・駒が初期局面以外分からない「衝立詰将棋」という不完全情報問題に取り組み，これを AND/OR 木探索に帰着させて解くことに成功した．探索空間の AND/OR 木は完全情報２人ゲームのものと比べてより一般化されている．本稿ではまず衝立詰将棋問題について説明し，次にこの不完全情報問題を AND/OR 木に帰着させる方法に触れる．そして探索手法として，全幅深さ優先探索の反復深化法，証明数探索の深さ優先版およびそれを不確定性を持つ AND/OR 木に特化させた探索法を示す．さらに，難問題を解くために上記探索手法の数種のバリエーションを提案し，各探索法によってテスト問題を解かせ，それぞれの性能を評価する．

2. 衝立詰将棋

2.1 衝立将棋と詰め問題

変則将棋の１つに衝立将棋というゲームがある．ゲームの対局者は２人だが，ほかに審判が必要となる．盤を２つ，駒を１組用意し，２つの盤の間に衝立をはさんでお互いの盤が見えない状態にして片方の駒だけを並べ，普通の将棋と同じように対局する．審判は両者の指し手を見て，１人が指し手を指したら「指しました」と発言しゲームを進行させていく．審判の発言は両対局者に知らされる．審判は必要ならば駒を取って取った側に渡したり，王手ならば「王手です」と発言する．また審判は，王手を放置したり，駒が行けないところに行ったというような反則手をチェックし，もし反則ならば「反則... 回目です」と発言する．反則のときは続けて別の指し手を選ぶ必要がある．反則の回数は普通８回まで許され，その許容回数を超えてしまうと反則負けとなる．どちらかの玉が詰んだ場合，審判が「詰み」を宣告し終局となる．衝立詰将棋は，この衝立将棋を基にした詰将棋の変種のパズルである[6]．

衝立詰将棋では２つのエージェント，攻め方と玉方とがある☆．問題の初期局面は詰将棋と同じように与えられるが，攻め方は玉方の応手を知ることができず，初期局面以外は自分側の駒しか見ることができない．一方，玉方はすべての駒が見えている状態にあり，つねに最善手を選んでくると考えてよい．なお，詰将棋問題と同じく多くの問題では攻め方の玉は存在しないが，一部に攻め方の玉も存在する問題があり双玉問題といわれる．

問題の目標は，玉方がどのような応手をしようとも攻め方が連続王手で玉方の玉を詰ませることである．ただし，攻め方は以下のような反則手を指してみることを許されている．

(1)　飛（竜）・角（馬）・香が玉方の駒を飛び越える．
(2)　玉方の駒があるマスへの駒の重ね打ち．
(3)　打歩詰め．
(4)　指し手の後，攻め方の玉が王手になる，あるいは王手を放置している（双玉問題のとき）．

許される反則の回数は上限があり何も断りがなければ８回である．攻め方が選んだ指し手が反則の場合，その手が反則であることだけが攻め方に通知される．反則の種類は通知されない．反則数が+1 され，攻め方は続けて別の王手候補を選ばなければならない．攻め方が選んだ指し手が反則でない場合，その指し手は必ず王手でなければならない．もし王手でない合法手を選んでしまったら，攻め方の失敗すなわち問題解決の失敗となる．

なお，最善応手手順として，第１にできるだけ手数（攻め方の王手の手数とそれに対する玉方の防ぎ手の手数の合計）が少なく，第２に反則数がより少ないものが選ばれる．それらが同じなら詰み局面での攻め方の残り持駒数がより多いものを選ぶ．ただし，攻め方の残り持駒がなくなるような問題が正しい問題とされる．

2.2 簡単な問題例と解答

図 1 は簡単な問題例で，これは与えられる確定初期局面である．解答を以下に示す．まず攻め方は 1.☗2 三銀と打って王手する．玉方はこれに対して 2.☖1 三玉と 2.☖3 一玉という２通りの逃げ方が可能だが攻め方はこれらを識別できない．したがって攻め方は両方の局面に対して王手または反則手となる指し手を選ぶ必要がある．そこで攻め方は 3.☗1 三角と打ってみる．もしこれが反則手と通知されれば，攻め方は玉方が 2.☖1 三玉と指したことが分かるので，3.☗1 四金と打って詰みとなる．もし 3.☗1 三角が反則手と通知されなければ，攻め方は玉方が 2.☖3 一玉と指したことが分かる．これに対し玉方は 11 通りの指し手が可能である．そのうちの２つは 3.☗1 三同桂と 3.☗1 三

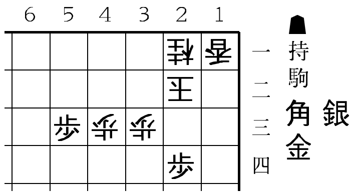

図 1　衝立詰将棋の問題例
Fig. 1　A sample Tsuitate-Tsume-Shogi problem.

☆ 反則や王手を攻め方に告げる役割をする審判エージェントもあると考えることもできるが，審判による通知は自動的に与えられるとするのが自然である．

同香で，これらの指し手は玉方の指し手の後 1 三にある角が消えることから，攻め方は玉方がどちらかの指し手を指したことを確定でき，3.☗3 二金と打って詰みとなる．他の 9 通りの指し手は攻め方には識別できない．玉を逃げる指し手 4.☖4 一玉，4.☖4 二玉が 2 つと 2 二に合駒を打つ手が 7 つある．ここで攻め方は 5.☗3 一角成としてみる．もしこれが反則手と通知されれば，攻め方は玉方が 2 二に合駒を打ったことを確定できるので 5.☗3 二金と打って詰みとなる．もし 5.☗3 一角成が反則手と通知されなければ，攻め方は玉方が 4.☖4 一玉または 4.☖4 二玉を選択したと確定できる．ここで玉方は 2 通りの応手が可能である．1 つは 6.☖3 一同玉と馬を取る手だが，攻め方は 3 一の馬が消えることからこの指し手を確定でき，7.☗3 二金と打って詰みとなる．もう 1 つは 6.☖5 一玉と逃げる手だが，攻め方は 3 一の馬が消えないことと他の可能性がないことからこの指し手を確定でき，7.☗5 二金と打って詰みとなる．

解答手順は 1.☗2 三銀 2.☖(—) 3.☗1 三角 4.☖(—) 5.☗3 一角成 6.☖(同) 7.☗3 二金まで 7 手詰めとなる．最後の 2 手は 6.☖(—) 7.☗5 二金でもかまわない．ここで (—) はこの玉方の指し手の後何も駒が取られなかったことを，(同) はこの玉方の指し手によって直前の指し手と同じ位置 3 一の攻め方の駒が取られてしまったことを表す．

3．不完全情報問題に対する決定論的問題解決

3.1　不確定性パラダイム

不完全情報問題を AND/OR グラフ（木）探索に帰着させ決定論的に解くための考え方・手法（不確定性パラダイムと名付けている）を導入する．決定論的な問題解決とは，考えられうるすべての場合について対処できる解答手順を見つけることで，その手順にはわずかな抜け道も許されない．

ここで，対象とする問題に関与する複数のエージェントがあるとして，Solver を問題をできるだけ解こうとするエージェント，Opponent を問題をできるだけ解かれまいとするエージェントとする．2 人ゲームを題材とする問題の場合は Solver と Opponent が 1 つずつ存在する．一般に探索による問題解決においては，問題のある状態を表す節点と状態に対する何らかの行動あるいは選択を表す枝とによって探索空間が構成される．ここではゲーム研究分野でよく使われる用語を使って，状態を局面，行動を着手ということにする．

不確定性パラダイムとは，不確定な状態をあたかも確定状態であるかのように取り扱う考え方である．Solver にとって識別することのできない状態（局面）の集合をまとめて 1 つのものとして取り扱い，**メタ局面**（metaposition）と定義する．メタ局面を構成する局面の数は**不確定性指標**（uncertainty index）と見なすことができる．また，メタ局面に対する状態遷移の選択を**メタ着手**（metamove）と定義する．

メタ着手が Solver にとって確定着手である場合，あるメタ局面に対しあるメタ着手を実行することはメタ局面を構成するすべての局面に対してその確定着手を実行することに相当する．

メタ着手が Solver にとって識別できない複数の着手の集合である場合，あるメタ局面に対しあるメタ着手を実行することはメタ局面を構成するすべての局面に対してそれぞれの着手を実行することに相当する．その結果メタ局面を構成する局面数は一般に増加する．これを**メタ局面の拡散**と呼ぶ．

もし探索中に Solver が何回かのメタ着手の間メタ着手およびメタ局面に対する手がかりを何も得ることができなければ，拡散が繰り返して生じメタ局面の不確定性は組合せ論的に爆発してしまう．しかし解決可能な問題には**観測**（observation）という Solver を助ける手がかりがあり，メタ局面を構成する各局面がそれぞれの観測結果に適合するかどうかに応じてメタ局面はより不確定性の小さい（すなわち構成局面数の少ない）いくつかのメタ局面に分裂する．これにより Solver はメタ局面がどの局面から構成されている可能性があり，どの局面からは構成されていないのかを識別することができる．しかし Solver はこの分裂を受動的に受け入れることしかできず，分裂したすべてのメタ局面を解決する必要がある．したがって，探索空間においては仮想の AND 節点が存在することになるので，これを**メタ局面の AND 分裂**と呼ぶ．

ここで探索空間全体を考えてみると，Solver の手番の確定着手の節点では Solver が任意の着手を選択できるため OR 節点になり，Opponent の手番の確定着手の節点では Opponent によって任意の着手を選択されてしまうため AND 節点になる．これらと上述したメタ局面の AND 分裂とをあわせて，結局すべての不完全情報問題の探索空間は AND/OR グラフ（木）に帰着される．これを**不確定性 AND/OR グラフ（木）**と呼ぶ．問題解決はこの AND/OR グラフ（木）を探索することによって実行できる．この探索を**不確定性パラダイム探索**（Uncertainty Paradigm Search）と名付ける．以下では UPS と略記する．

3.2　関 連 研 究

従来から，Mastermind[7),8)] など多くの可能な状態

から何らかの手段でただ1つの状態に絞り込むことを目的とするsingle-agentパズル問題では,「候補の集合」を探索空間の節点として問題解決されている．我々も，いくつかの金貨の中から重さの違う贋金貨を見つける「贋金貨問題」を不確定性パラダイムの考え方で解決するプログラムを作成した[12]．これらのsingle-agentパズル問題では,「候補の集合」(＝我々の用語では，メタ局面）を容易に直接表現できるため，候補の集合を探索空間の節点とする考え方は大体のところきわめて自然なものである．結局，これらsingle-agentパズル問題に限れば，不確定性パラダイムという考え方は目新しいものではない．しかし，不確定性パラダイムはsingle-agentパズル問題だけでなく，2人ゲーム問題，さらにn人ゲーム問題にいたるまで任意の不完全情報問題に対して適用可能なように一般化・定式化したもので，その中には候補の集合（＝メタ局面）および着手の集合（＝メタ着手）が容易には直接表現できないものもあり，そこでは我々の考え方・手法がクローズアップされてくる．また，一般にsingle-agent問題では可能な候補の集合は探索の進行にともなって小さくなる一方であるが，2人以上のゲームを題材とする問題では，候補の集合が探索中にダイナミックに増減するという特徴を持つ興味深い問題領域が存在する．

さらに，我々の考え方は非決定性有限状態機械を等価な動作をする決定性有限状態機械に変換する考え方と同様である．ただ，我々が提案するのは探索による問題解決のための考え方・手法で，探索空間を決定性有限状態機械に変換した後の探索手法までも含めて議論の対象としている点で独自性がある．非決定性有限状態機械の全体が初めから見えているわけではなく，探索の進行につれてごく一部分一部分が明らかになっていくにすぎず，問題解決された時点でも探索空間全体のごく一部しか見えないことがほとんどである．

また，我々の手法は展開型のゲームにおける情報集合を探索空間の節点としたものととらえることができる．しかし展開型のゲームではあくまでもゲーム木の節点を局面とし弧を着手としているので，1つの情報集合の各局面に対して複数の弧（着手）でつながる後続局面が存在することになる．一方，メタ局面をゲーム木の節点，メタ着手をゲーム木の弧としてとらえる我々の手法では節点と子節点の間は必ずたった1つの弧によってつながっている点で異なる．

不完全情報ゲームの分野でも類似する手法が知られている．Frankらの研究[4]ではゲーム木探索において同じ情報集合にある状態をまとめて“vector”として扱い，1つの節点として探索を行う．ただし，対象が実際のゲームのミニマックス木であり確率を考慮した探索を行うことと，vectorが固定長で取り扱われるのが異なる点である．我々の手法は確率をまったく考慮せず決定論的な問題解決に焦点を絞ったものなので，適用可能領域は限定されるがAND/OR木探索になる．また，節点であるメタ局面を構成する局面数がダイナミックに増減するのが面白い点である．

チェス系の不完全情報ゲームはあまり研究されていない．チェス版の衝立将棋に相当するKriegspielにおいて，Fergusonはking，bishop，knight対相手kingのみという終盤問題をゲーム理論でいう多段階再帰ゲームとして研究し，一般的には（初期状態でkingがbishopとknightに利きをつけて守っていれば）確率1でbishop，knightのある側が勝つことを示した[3]．またCiancariniらは知識ベースのアプローチによりking，pawn対kingという終盤をプレイするプログラムを報告している[2]．しかし，Kriegspielにおいては今回の我々の手法のように単なる探索によって決定論的に問題解決をした報告はない．また，衝立将棋をプレイする，あるいは衝立詰将棋を解くコンピュータプログラムに関する研究論文はこれまでなかった．

4. 衝立詰将棋を解くプログラム

4.1 不確定性パラダイム下での解釈

衝立詰将棋問題の探索空間を「局面を節点・指し手を枝とする形」で表すと，攻め方にとって識別できない局面の集合（情報集合）は通常複数の局面を含む．これは不完全情報問題の特徴である．しかし，衝立詰将棋の正しい問題においては決定論的な解が存在することが保証されているので，不確定性パラダイムの下でAND/ORグラフ（木）探索によって解くことができる．

衝立詰将棋における2つのエージェント，攻め方はSolver，玉方はOpponentに相当する．攻め方にとって，攻め方のメタ着手は確定着手のうちの1つ，玉方のメタ着手はいくつかの着手をまとめたものになる．つまり，攻め方の手番のあるメタ局面において可能な着手はメタ局面を構成するすべての局面に対して王手あるいは反則手になる着手であり，その中の1つの着手がそのメタ局面に対するメタ着手になる．一方，玉方の手番のあるメタ局面において可能な着手はメタ局面を構成する各局面に対するすべての合法着手であり，そのメタ局面に対するメタ着手はそれら合法着手をすべてまとめたものとなる．

攻め方がメタ局面を構成する局面を判別する材料となる観測として以下のものがある．

(1) 攻め方のメタ着手が王手だったか反則手だったか.
(2) 局面が詰んだか.
(3) 玉方のメタ着手の後で駒が取られたか，取られたとしたらどこのマスで取られたか.
(4) 攻め方のメタ着手の後で駒を取ったか，取ったとしたらどの駒種を取ったか.
(5) 玉方のメタ着手の後で逆王手がかかったか（双玉問題のとき).

4.2 メタ局面の実装

我々は衝立詰将棋問題を不確定性パラダイムの下で解くプログラムを実装したが，そこではメタ局面を局面の配列の形で表現している．これはメモリ資源を多く必要とするが単純で容易に実装できる方法であり，衝立詰将棋には最も適していると思われる.

あるメタ局面に対して何らかのメタ着手を実行した後，その中に複数の同一局面が生じる場合がある．この場合，メタ局面内のすべての局面を調べて重複している局面を削除する必要がある．これをメタ局面のコンパクト化と呼ぶ．我々の実装では各局面のコードの算術和をとる方法によってメタ局面のコード化を行っている．コンパクト化を行わないと実際には同一であるメタ局面が違ったコードを持ってしまう場合がありまずいので，コンパクト化はコード化の前に必須のものとなっている.

5. 局面表を使った UPS

5.1 UPS における AND/OR 木の特徴

衝立詰将棋では，UPS での AND/OR グラフは同一メタ局面が頻繁に生じないことから木と見なして探索してよい．以下では AND/OR 木探索に絞って UPS を議論する．詰将棋のような手番が順次交代するような完全情報 2 人ゲーム問題の探索空間もまた AND/OR 木と見なせるが，この AND/OR 木と UPS で生ずる AND/OR 木との間にはいくつか特徴的な相違点がある．両者の相違点を**表 1** にまとめるが，UPS の AND/OR 木の方がより一般性がある．UPS では，Solver の手番の節点は基本的には OR 節点であり，そこから出る各枝は OR 条件でつながっている．しかし枝の中に AND 分裂するものがあるので，1 節点に AND 枝と OR 枝が混在することになる．以下では，この UPS における AND/OR 木探索において局面表を有効活用したいくつかの探索法について述べる．ここで局面表とは transposition table のことで，実際には要素はメタ局面を表す.

表 1 完全情報 2 人ゲーム問題の AND/OR 木と UPS の AND/OR 木の相違点
Table 1 AND/OR tree in a two-person complete-information problem and that in UPS compared.

	親子関係	AND 弧と OR 弧	その他
完全情報 2 人ゲーム	AND 節点と OR 節点が交互	分離	
UPS	AND と OR が必ずしも交互でない	1 節点に混在している場合がある	各節点が不確定性を持つ

5.2 深さ優先の全幅反復深化探索

これは局面表を利用する深さ優先の全幅反復深化法（iterative deepening）で，以下では ID と略記する．C ライクな擬似コードを**図 2** に示す．ただし，これは

```
int IDsearch( 初期局面 ) {
  初期メタ局面 = 初期局面;
  for ( 手数 = 1;   ; 手数 += 2 ) {
    for ( 反則数 = 0;
          反則数 <= 最大許容反則数; 反則数++ ) {
      r = semekata( 初期メタ局面, 手数, 反則数 );
      if ( r == 詰み ) return r;
    }
  }
}

int semekata( メタ局面, 手数, 反則数 ) { // 攻め方手番
  反則数を考慮したメタ着手リストの生成;
  if ( メタ着手リストが空 ) return 不詰;
  for ( すべてのメタ着手について ) {
    次メタ局面リスト
      = itteSusumeS( メタ局面, 反則数, メタ着手 );
    r1 = 詰み;
    for ( すべての次メタ局面について ) {
      if ( 王手後の次メタ局面 )
        r2 = gyokukata(次メタ局面, 手数 - 1, 反則数);
      else // 反則後の次メタ局面
        r2 = semekata(次メタ局面, 手数, 反則数 - 1);
      if ( r2 != r1 ) {
        r1 = 未解;  break;
      }
    }
    if ( r1 == 詰み ) return 詰み;
  }
  return 未解;
}

int gyokukata( メタ局面, 手数, 反則数 ) { // 玉方手番
  次メタ局面リスト = itteSusumeG( メタ局面 );
  if ( 次メタ局面リストが空 ) return 詰み;
  if ( 手数 == 0 ) return 未解;
  r1 = 詰み;
  for ( すべての次メタ局面について ) {
    r2 = semekata( 次メタ局面, 手数 - 1, 反則数 );
    if ( r2 != r1 ) {
      r1 = 未解;  break;
    }
  }
  return r1;
}
```

図 2 衝立詰将棋問題の ID 探索の擬似コード
Fig. 2 Pseudo-code of the ID search for a TTS problem.

最短手数・最小反則数の詰みを見つけるためのもので，最適解も不詰も見つけない．中心となる関数 IDsearch において，反復は最短手数かつ最小反則数の解を見つけ出すため二重ループで行う．外側のループでは手数による反復深化，内側のループでは許容反則数による反復深化を行う．これにより最小手数・最小許容反則数の解が得られることが保証される．攻め方の手番の節点では，メタ着手の生成を行い，OR 条件で各メタ着手について調べる．関数 itteSusumeS で，メタ着手の実行によるメタ局面の拡散・観測によるメタ局面の AND 分裂・メタ局面のコンパクト化を処理しており，次のメタ局面になりうる「次メタ局面リスト」が返される．ここで，各「次メタ局面」に対して AND 条件で，王手か反則手かで場合分けして探索を進める．玉方の手番の節点では，関数 itteSusumeG で，メタ着手の実行・観測によるメタ局面の AND 分裂などを処理しており，次のメタ局面になりうる「次メタ局面リスト」が返される．ここで，各「次メタ局面」に対して AND 条件で探索を進める．

解の手数および許容反則数を見つけた後，最善応手手順を含む最適解を求めるためすべての OR 節点（攻め方の手番）において多重反復深化をする再探索を行う．最善応手手順の決定では，OR 節点においては最も良い手順を持つメタ局面が，AND 節点（AND 分裂）においては最も悪い手順を持つメタ局面が選ばれる．根節点で選ばれる手順が最善応手手順となる．また，最適解とは解決木のすべての OR 節点が最善な着手を持つ解を意味する．手順は以下のとき「良い」と見なす．

(1) その手順の手数が別の手順の手数よりも少ない．
(2) 手数が同じなら，その手順中の反則数が別の手順中の反則数よりも少ない．
(3) 手数と反則数が同じなら，その手順の詰み局面における攻め方の残り持駒数が別の手順の残り持駒数よりも多い．

5.3 PDS

AND/OR 木の探索については近年多くの革新的で高機能な探索アルゴリズムが研究されてきている．Allis が証明数・反証数という概念を導入して開発した証明数探索[1]，Seo らが開発した局面表を利用し証明数を閾値とする多重反復深化を行い深さ優先ながら最良優先探索と近似的に同様に振る舞う C*（PN* と改称された）[14]，また，Nagai が C*を基にして開発した証明数および反証数を閾値とする多重反復深化を行う PDS（Proof-number and Disproof-number Search）[10]，さらにその改良版の df-pn[11] などがある．UPS も AND/OR 木の探索であるので，これらの高効率な探索法を利用することができる．その中で，我々の先の実験で完全情報 2 人ゲーム問題である詰将棋において PDS が優れた結果を出した[13] ので，これを UPS に適用することとした．

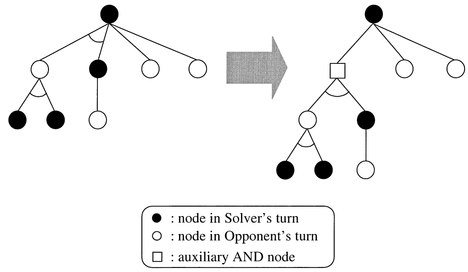

図 3 AND/OR 木の等価的変換
Fig. 3 Transformation of general AND/OR tree.

ただし，表 1 に示したように UPS では AND 枝と OR 枝との混在節点があり，そのままでは PDS を適用できない．これは補助節点を導入して節点を AND 節点と OR 節点に分離することにより対応できる（**図 3** 参照）．また，PDS では negaPDS 記法といって，OR 節点と AND 節点におけるプログラムコードを共有する記法を使っている．しかし，これは AND 節点・OR 節点が交互に現れることを前提としており，UPS では表 1 にあるように AND 節点と OR 節点が必ずしも交互に現れないため適切でないので，OR 節点と AND 節点のプログラムコードを場合分けする必要がある．

5.4 UPDS

証明数・反証数を利用する探索は探索節点数を最小化することにより探索コストを最小化することを目的としている．そこでは未展開である各節点を証明するあるいは反証するためのコストが平均的に等しいことを仮定している．しかし UPS では各節点が不確定性指標（実際には局面数）という属性を持っており，一般に節点の不確定性が大きいほど探索コストは大きくなる．不確定性指標（局面数）が N のメタ局面節点の子節点の展開に必要な計算量を通常の確定局面の子局面への展開に必要な計算量と比較してみると，メタ着手の生成が N 倍，メタ着手の実行が N 倍，メタ局面のコンパクト化が $N \log N$ 倍などになっている．どの部分が律速かに依存するが，$N \log N$ 倍あるいは N 倍の展開コストを要することが分かる．

計算量とは別に，不確定性指標（局面数）と証明しやすさ・反証しやすさとの間にある関係を考えてみる．衝立詰将棋の場合，攻め方の手番のメタ着手はメタ局面を構成するすべての局面に対して王手あるいは反則手になる着手であり，積集合を求めることになるので，

局面数が大きいほど集合が空集合になりやすい，すなわち反証されやすい．また，詰みメタ局面とはメタ局面を構成するすべての局面が詰みになる場合なので，局面数が小さいほど詰みになりやすい，すなわち証明されやすい．

以上のことから，未展開節点の証明数・反証数の初期値として通常の 1 でなく，

$$\mathrm{proof_{init}} = \min(\lceil (N/D)^{\gamma_p} \rceil, \mathrm{IPD_{max}})$$
$$\mathrm{disproof_{init}} = \min(\lceil (N/D)^{\gamma_d} \rceil, \mathrm{IPD_{max}})$$

を与えるような探索を考える．ここで γ_p，γ_d および D は定数パラメータである．γ_p，γ_d は不確定性の爆発を抑えようとする度合いを与える．これらの値が大きいほど探索はより不確定性が小さい方向に進むようになる．D は γ_p，γ_d の効果を抑えるための正定数で，値が大きいほど効果が抑えられる．また，$\mathrm{IPD_{max}}$ は初期値の上限値で，今回は 50 に固定されている．上記の初期値設定関数を使う探索はコスト推定関数を使う PDS である PDS*の一種とも考えられるが，PDS を不確定性パラダイムの探索用に特化させたものといえるので，この探索を UPDS（Uncertainty Paradigm PDS）と名付けた．衝立詰将棋の場合は，先に述べた計算量の目安と証明しやすさの関係から，$\gamma_p \geq 1$，$\gamma_d \leq 1$ が適していると予想できる．ただし，γ_p にあまり大きな値をとると，各節点内の不確定性の爆発は抑えられるようになる一方，探索がより節点数が多い方へと誘導されてしまう危険性がある．なお，$\gamma_p = \gamma_d = 0$ で UPDS は純粋な PDS と同じになる．

5.5　難問題解決のための探索のバリエーション

5.5.1　dpUPDS

当初 UPDS の探索において衝立詰将棋のすべての問題について最も適切なパラメータ (γ_p,γ_d,D) 設定があると考え，それらを見つけようとした．しかしいろいろなパラメータ設定で問題を解かせてみて，問題の性質によって適切なパラメータ設定が変ってくることが分かった．ある種の問題は不確定性が小さく完全情報の詰将棋に似た性質を持つ．それらに対しては $\gamma_p \sim 0$，$\gamma_d \sim 0$ の値が適切と思われる．また別の種の問題では探索において不確定性が爆発しやすく，解を見つけるためにできるだけ不確定性を抑える方向に探索を進める必要がある．それらに対しては $\gamma_p > 1$，$\gamma_d \sim 1$ の値が適切と思われる．

そこで実行時にあらかじめ決めてあるパラメータセット (γ_p,γ_d,D) の中の値を順次設定して問題が解けるまで探索を反復する方法を考案し，dpUPDS（UPDS with dynamic parameters）と名付けた．パラメータの設定値の指針は特に設けず，絨毯爆撃的な変更を行っているため，かなり時間を消費する場合がある．

5.5.2　uidUPDS および uidID

5.4 節で触れたように，大きな不確定性をもつメタ局面は反証されやすいにもかかわらず計算量は大きくなる．見込みが少なくかつ計算量が多い節点以下を探索するのはほとんどの場合大きな無駄になる．そこで，不確定指標を閾値とする反復深化を行う UPDS を考案し，これを uidUPDS（UPDS with iterative deepening of uncertainty index）と名付けた．uidUPDS は不確定性の小さい解を持つ問題に対しては非常に有効に働くことが期待できるが，逆に不確定性の大きい解を持つ問題に対しては反復深化で無駄な探索を繰り返し時間を浪費する懸念がある．

ID においても同様に，手数・反則数の反復深化の外側でさらに不確定指標を閾値とする反復深化も行う探索 uidID（ID with iterative deepening of uncertainty index）が考えられる．証明数探索において最適解が求められない問題に対して効率的に最適解を求めることが期待できる．

6. 実 験 結 果

実験は Pentium II 450 MHz，RAM：384 MB Windows 98 という環境で行った．用いた問題[6)] は全部で 39 問ある．まず PDS および UPDS でパラメータ (γ_p, γ_d) を (0,0) から (1.8,0.6) まで単調変化させ問題を解かせて動作を調べた．パラメータ D は 4 に固定した．PDS，UPDS では問題解決は複数の段階からなる．最初の段階は 1 つの解を求めるもので，残りの複数の段階は最適解を求めるためのものである．実験では各段階に最大 2 時間を与えた．2 時間で最初の段階の解が求まらないものは未解とした．**表 2** に結果のまとめを示す．比較のため ID による結果も示している．ただし ID では解答時間の制限は特に設けなかった．なお，本稿の時間データは秒単位である．

ID は 28 題解けたが，比較的難しい問題にはきわめて多くの時間を要しており，一般に ID よりも UPDS 系の探索の方がずっと速く問題を解く．UPDS 系では PDS が 23 題しか解けず最も悪く，未解問題も含めた解答時間が最も多くかかっている．しかし，その他のパラメータセットではどれがよいか判別できない．表には示せていないが，パラメータセットによって解ける問題の種類がいろいろ変わってきて問題ごとに別の適したパラメータセットがあるように見える．注目すべき点は (γ_p, γ_d) を小さい値にした場合，探索節点数が少なくなり，探索における平均不確定性指標は大きくなっている．逆に (γ_p, γ_d) を大きい値にした場合，

表 2 ID とパラメータを変えた UPDS による実験結果
Table 2 Results of ID and UPDS with several sets of parameters.

		ID	PDS	UPDS				
(γ_p,γ_d,D)		–	(0,0,–)	(0.5,0.2,4)	(0.8,0.3,4)	(1.1,0.4.4)	(1.5,0.5,4)	(1.8,0.6,4)
solved		28	23	27	28	27	27	26
unsolved		11	16	12	11	12	12	13
time solved	total	1599739	6369	17465	29880	12872	28566	23084
	average	57134	277	647	1067	477	1058	888
time all	total	–	122395	105991	109632	99253	115246	109784
	average	–	3138	2718	2811	2545	2955	2889
average depth		–	29.92	29.40	29.55	29.35	29.40	31.39
average uncertainty		4.15	11.09	3.96	3.61	3.60	3.16	3.02
average node count		2921297	150256	415857	446240	440545	502017	532583

time solved：解けた問題のみの時間集計，time all：すべての問題の時間集計（時間は秒単位）

表 3 各探索法によって解けた問題数と解答時間（全 39 問）
Table 3 Results of solving 39 TTS problems.

	ID	uidID	dpUPDS	uidUPDS
solved	28	36	32	39
unsolved	11	3	7	0
total time	1599739	355784	164647	165088
average time	57134	9883	5145	4233

探索節点数は多くなる一方，探索における平均不確定性指標は小さくなっている．これは，パラメータ調整によって探索節点数を抑えることと不確定性の爆発を抑えることのバランスをとっていることを示しており，UPDS のねらいが実現されていることが分かる．

次に，ID，uidID，dpUPDS，uidUPDS について全問を解かせた結果を**表 3** に示す．uidUPDS はすべての問題を解くことができた．ただし，uidUPDS では 35 題は $\gamma_p = 0.8$，$\gamma_d = 0.3$，$D = 4$ として解け，残りの 4 題は試行錯誤によりパラメータを人手で調整した．その 4 題のうち 3 題は，解決木における節点の不確定性指標（局面数）の最大値が 40 以上のもので，探索に制限を加えて探索空間のごく一部から答えを得ようとする uidUPDS 法の意図が働きにくく，探索空間中のかなり多くの節点を調べる必要があったと思われる．つまり，不確定性の観点から適するパラメータの選択に敏感であったと考えられる．もう 1 題は解決木における節点の最大不確定性指標は 9 と小さいが，上記 3 題を除いた 36 題中で有効探索空間の大きさ（ここでは局面表の登録数で考える）が最も大きいものである．つまり，探索節点数の観点から適するパラメータの選択に敏感であったと考えられる．表 2 と表 3 から，探索バリエーション（uidID，dpUPDS，uidUPDS）の方が元となった探索（ID，PDS，UPDS）よりも解答能力が高いことが分かる．問題の解決木における手数と反則数の合計と解決木における節点の最大の不確定性指標のプロットによる解決木の分布を**図 4** に示す．ここで，All とはすべての探索法で解かれた問題，ID+uidUPDS とは ID と uidUPDS で解かれた問題などを表す．手数と反則数の合計（解決木の高さに相当）が大きくかつ最大不確定性指標が比較的大きい問題が難問で，探索バリエーションでしか解くことができなかった．

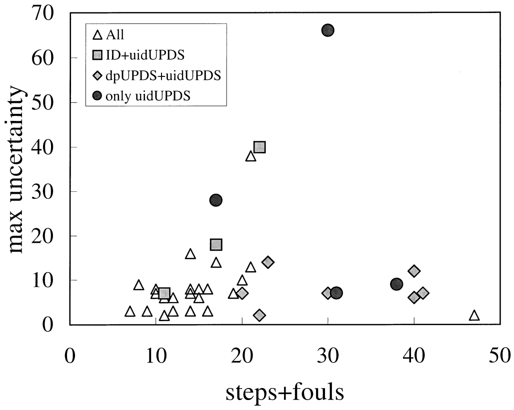

図 4 問題の解決木の分布
Fig. 4 Distribution plot of solution trees.

解答手順の手数・反則数は 7 手反則 0 から 43 手反則 4 までで，解決木におけるメタ局面の最大局面数は 66 であった．また，全幅探索である ID での各データを基に探索にかかわる集計データを示しておく．探索節点数の平均は 2921297・最大は 23382373，探索メタ局面のうちの最大局面数の平均は 987.7・最大は 10572，探索メタ局面の平均局面数の平均は 4.15・最大は 10.85 であった．

また，手数と反則数の合計に対する解答時間のプロットを**図 5** に示す．一般的に ID の方が UPDS 系の探索よりも多くの時間がかかっており，特に手数の長い問題に対しては 10 日間以上の時間を要するものもあった．43 手反則 4 の問題を除き，ID では手数と反則数の合計が 20 前後以上の問題は解けていない．

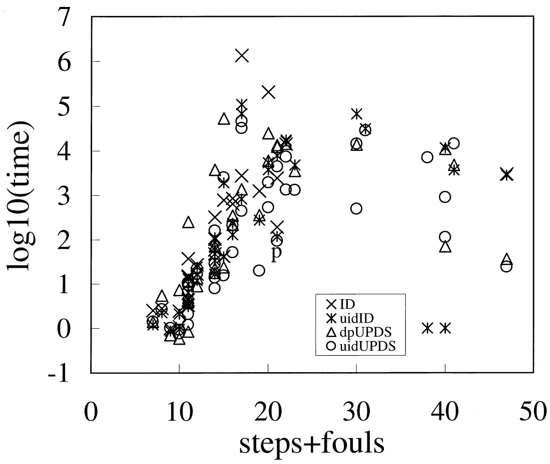

図 5　各問題の探索法別の解答時間
Fig. 5　Time for solving problems.

dpUPDS と uidUPDS では，両者が解けた問題については解答時間に大きな差は見られなかった．一般に uidID は dpUPDS, uidUPDS よりも解答に時間がかかっている．

最善応手手順を含む最適解を求めることにおいては，しらみつぶし探索を行う ID に確実性がある．uidUPDS を含む証明数探索系は最適解を見つけることを保証しない．証明数を使う探索においても近似的に最適解を見つけられるようなアルゴリズムを実装して実験したが，うまく最適解を見つけられなかった問題が数題ある．uidID は ID よりもずっと多くの 36 題を解くことができたが，そのうち最適解を求めていない問題が 2 題あり，残念ながら最適解を求めるには適していないことが分かった．

参考のため，用いた問題のうち 2 題をあげる．まず**図 6** は解答手順の面白い問題の 1 つである．解答手順は 1.☗1 三歩成 2.☖(—) 3.☗2 二歩成 4.☖(同) 5.☗1 二歩 6.☖(—) 7.☗1 一歩成 8.☖(同) 9.☗3 一竜：銀 10.☖(同) 11.☗1 二銀まで 11 手詰めである．コロン (:) の後の駒で攻め方が取った駒を表す．5 手目の 1 二歩によって 4 手目が 2 二同香であったことを確認し，すぐ 7 手目の 1 一歩成で玉を元に戻すのが面白いところである．4 手目が 2 二同銀の場合，5 手目の 1 二歩は打歩詰めの反則となり，5.☗1 二歩 × 5.☗2 二と：銀 6.☖(同) 7.☗3 二竜 8.☖(—) 9.☗1 二銀まで早詰みとなる．× はその指し手が反則であることを表す．この問題はコンピュータにとって比較的簡単な問題と考えられ，すべてのプログラムで解くことができた．たとえば ID プログラムでは正解を 16.8 秒で解答した．

次にプログラムで解くのが難しかった問題を**図 7** に示す．これは 27 手反則 3 の問題で，解決木中の節点の最大不確定性指標が 66 になる．解決木において，下記の手順の後の途中メタ局面を**図 8** に示す．

1.☗5 三銀 2.☖(—) 3.☗5 一飛 4.☖(—) 5.☗3 二飛 6.☖(—) 7.☗6 二飛成 × 7.☗5 二飛引成 8.☖(—) 9.☗6 二銀成：香 10.☖(—) 11.☗7 一成銀：銀

9 手目に香を取ったことから 8 手目に香が合駒されたことが分かる．一方，7 手目の 6 二飛成が反則で 5 二飛引成で駒が取れていないことから，6 手目で 4 二に A で示す合駒をしたことが分かるが，これが 3 一金の移動合も含め 6 通り考えられる．したがって，図 8 のメタ局面の不確定性指標は 6 である．今 5 二の竜で 8 二玉が王手になっており，これの応手として，何も駒が取られなかった（12.☖(—)）とすると，合駒が E, F 地点に各 5 通りと玉が G に逃げる手とで合計 11 通りある．結局，12 手目の後では $6 \times 11 = 66$ 通りの可能局面が生じる．この問題は uidUPDS プログラムで人手でパラメータ調整したときのみ解くことができた．解答にも 14480 秒かかっている．解決木中に不確定性の大きな節点があることが一番の難しい要因

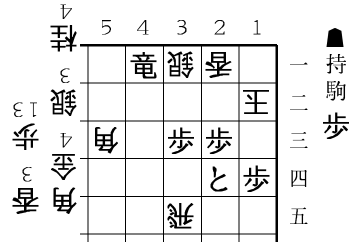

図 6　面白い解答手順の問題例
Fig. 6　A problem with a gimmicky sequence.

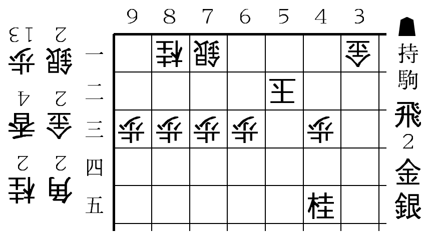

図 7　難しい問題
Fig. 7　One of the hardest problems.

図 8　解決木中の一変化
Fig. 8　A variation position in the solution tree.

といえる．

7. まとめ

不完全情報問題の一種である衝立詰将棋問題に対して不確定性パラダイムという手法により問題解決を単なる AND/OR 木探索に帰着できた．人工知能分野で研究・開発されてきた完全情報の AND/OR 木に対する高効率な探索手法は本手法においても有効であるが，探索節点数を少なくすることとメタ局面内での不確定性の爆発を抑えることとの間にはトレードオフがある．我々は PDS の不確定性パラダイム版である UPDS を考案し，探索の際の節点数と不確定性のバランスをパラメータによって制御できることを示した．証明数探索系の探索法は最良優先的に進むため，しらみつぶし探索では不確定性の爆発により解けない問題でもうまく解決できる場合が多く，非常に有望な手法である．また，反証する（解がないことを証明する）ことにおいても非常に高い能力を示している．

衝立詰将棋の問題はすべて解けたが，最善応手手順を含む最適解を見つけられない問題が数題あった．詰将棋において PDS を使って最善応手手順を求めるアルゴリズムが発表されており[15]，衝立詰将棋に応用することが考えられる．今後，本稿で使用した手法を Kriegspiel の終盤問題への適用，さらに他の問題領域への適用を図っていく．その際，必要ならば決定論的解決の制約を少し緩めるような拡張が考えられる．

参考文献

1) Allis, L.V.: Searching for Solutions in Games and Artificial Intelligence, Ph.D. Thesis, University of Limburg, The Netherlands (1994).
2) Ciancarini, P., Libera, F.D. and Maran, F.: Decision Making under Uncertainty: A Rational Approach to Kriegspiel, *Advances in Computer Chess 8*, van den Herik, H.J. and Uiterwijk, J.W.H.M. (Eds.), pp.277–298, Drukkerij Van Spijk B.V., Venlo, The Netherlands (1997).
3) Ferguson, T.S.: Mate with bishop and knight in kriegspiel, *Theoretical Computer Science*, Vol.96, pp.389–403 (1992).
4) Frank, I. and Basin, D.: Search in games with incomplete information: A case study using Bridge card play, *Artificial Intelligence*, Vol.100, pp.87–123 (1998).
5) 伊藤琢巳，野下浩平：詰将棋を速く解く 2 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).
6) 加藤　徹：カピタン文献集 1，衝立将棋編，(1995).
7) Knuth, D.E.: The Computer as Master Mind, *Journal of Recreational Mathematics*, Vol.9, pp.1–6 (1976-77).
8) Koyama, K. and Lai, T.W.: An Optimal Mastermind Strategy, *Journal of Recreational Mathematics*, Vol.25, No.4, pp.251–256 (1993).
9) 松原　仁（編）：コンピュータ将棋の進歩，共立出版 (1996).
10) Nagai, A.: A New Depth-First-Search Algorithm for AND/OR Trees, M.Sc. Thesis, Department of Information Science, The University of Tokyo, Japan (1999).
11) Nagai, A. and Imai, H.: Application of df-pn+ to Othello Endgames, *Proc. Game Programming Workshop in Japan '99*, Hakone, Japan, pp.16–23 (1999).
12) Sakuta, M.: Deterministic Solving of Problems with Uncertainty, Ph.D. Thesis, Shizuoka University, Japan (2001).
13) 作田　誠，飯田弘之：高性能な AND/OR 木探索アルゴリズムの詰め将棋問題による比較，静岡大学情報学研究，Vol.5, pp.15–22 (1999).
14) Seo, M., Iida, H. and Uiterwijk, J.W.H.M.: The PN*-search algorithm: Application to tsume-shogi, *Artificial Intelligence*, Vol.129, pp.253–277 (2001).
15) 山田　剛，松原　仁（著），松原　仁（編）：計算機による逆算式詰将棋創作の支援，コンピュータ将棋の進歩 3，5 章，pp.60–81，共立出版 (2000).

(平成 12 年 10 月 25 日受付)
(平成 13 年 11 月 14 日採録)

作田　誠（正会員）

2001 年静岡大学大学院理工学研究科後期課程修了．博士（工学）．現在，同大学院研究生．ゲーム・パズルを題材とした探索・推論・学習等に興味を持つ．

飯田　弘之（正会員）

1962 年生．将棋プロ棋士六段．1994 年東京農工大学大学院博士後期課程修了．博士（工学）．科学技術振興事業団博士研究員，オランダマーストリヒト大学客員教授等．現在，静岡大学情報学部助教授．ゲーム情報学，エンターテイメントコンピューティング等に興味を持つ．