# 長手数の詰将棋を解くプログラムを改善するヒューリスティックス

長　井　　歩

詰将棋を解くアルゴリズムの進歩を影で支えるヒューリスティックスに着目する．適切なヒューリスティックスは個々のゲーム特有の事情をうまく吸収し，探索アルゴリズムの本来の性能を引き出す意味で重要な存在である．本研究では，既に提案されているヒューリスティックスだけでなく，効果がありそうなヒューリスティックスをいくつか提案し，効果的な組み合わせを調べる．実験の結果，df-pn アルゴリズムで詰将棋を解く際に，ある 3 つのヒューリスティックスを導入すると実行時間を約 77%に抑えられた．

# Heuristics to improve solving Tsume-Shogi

Ayumu Nagai[*1]

We focus on heuristics which support to solve Tsume-Shogi problems. Ideal heuristics can settle troubles concerning individual domain-specific games and draw out latent abilities of the search algorithm. In this paper, we suggest some heuristics and find out the best combination among them. Our experimental results show that three heuristics in combination reduced execution time down to around 77%.

## 1. はじめに

計算機によって詰将棋を解くアルゴリズムは，証明数や反証数を使った探索法によって大きく進歩した．証明数のみを使った先駆的な研究である脊尾による C* アルゴリズム[14] や，証明数と反証数を対等に利用した df-pn アルゴリズム[17] の開発により，詰将棋を解くプログラムは大きく進歩した．一方で，合駒処理を代表とするヒューリスティックスの導入も詰将棋を解くプログラムのパフォーマンスを向上させる大きな要因となってきたことも忘れてはならない．実際のゲームに適用する場合，探索アルゴリズムの中に盛り込むのが困難な，それぞれのゲームに特有の事情がしばしば存在する．詰将棋の場合，合駒処理がその典型的な例であろう．このようなゲーム特有の事情を，適切なヒューリスティックスの導入によって吸収することにより，探索アルゴリズムが本来持つ潜在的なパフォーマンスを引き出すことができる．

本研究は，詰将棋を df-pn アルゴリズムで解く際の適切なヒューリスティックスについて議論する．既に提案されているヒューリスティックスだけでなく，効果がありそうなヒューリスティックスをいくつか提案し，効果的な組み合わせを探す．

*1 群馬大学大学院工学系研究科
Department of Computer Science, Gunma University
E-mail: nagai@cs.gunma-u.ac.jp

## 2. 証明数・反証数

証明数や反証数[2] は，AND/OR 木の各節点に対して定義される値である．詰将棋に適用した場合の証明数 (反証数) の定義は，その節点が詰 (不詰) であることを示すために，詰 (不詰) であることを示さないといけない先端節点の個数の最小値である．

節点 $n$ の証明数を $\mathbf{pn}(n)$，反証数を $\mathbf{dn}(n)$ で表すと，具体的な証明数・反証数の計算法は以下のように再帰的に行われる．

(1) 節点 $n$ が先端節点
  (a) $n$ が詰み
    $\mathbf{pn}(n) = 0$
    $\mathbf{dn}(n) = \infty$
  (b) $n$ が不詰
    $\mathbf{pn}(n) = \infty$
    $\mathbf{dn}(n) = 0$
  (c) $n$ が詰みか不詰か不明
    $\mathbf{pn}(n) = 1$
    $\mathbf{dn}(n) = 1$
(2) 節点 $n$ が内部節点
  (a) $n$ が OR 節点
    $\mathbf{pn}(n) =$ 子節点の $\mathbf{pn}$ の最小
    $\mathbf{dn}(n) =$ 子節点の $\mathbf{dn}$ の和

(b) $n$ が AND 節点

$\mathbf{pn}(n)$ = 子節点の $\mathbf{pn}$ の和

$\mathbf{dn}(n)$ = 子節点の $\mathbf{dn}$ の最小

## 3. 既存のヒューリスティックス

ある局面にて指せるすべての手の重要度が等しいと仮定することができるなら，定義通りに証明数や反証数を計算すればよい．しかし実際には，すべての手を等しい重要度で扱うと悪平等になってしまう場合がある．

詰将棋においては合駒処理がその典型である．王手のかかっている局面で合駒する手が複数あるとして，それらの合駒の手のうち，どの合駒を指してもその先の展開が同じになることがしばしばある．合法手としては多数あっても実質的には高々1手にしかすぎない．このような場合，AND 節点の証明数として，定義通りにすべての合駒の手の証明数を加算すると，問題の真の難易度を反映しなくなってしまう．

そこで脊尾は，実際の解けにくさの度合をなるべく正確に証明数に反映させる工夫として次のような工夫を導入している[15]．先に無駄合以外の応手をすべて探索し，それらの手がすべて詰むことが判明したあと，無駄合と予想される手を後手玉に近い位置から順に探索する．その際，詰ますために本当に必要な駒 (証明駒[16]) の情報をハッシュ表に登録しておくと効果的である．また，無駄合と予想される手は証明数にカウントせず，無駄合以外の合駒は1種類だけカウントする．証明数を利用する詰将棋プログラムの多くは脊尾の工夫を導入していると考えられる．

df-pn アルゴリズムをはじめとする証明数探索で詰将棋を解く場合，証明数・反証数の二重カウント[8][10]がしばしば問題となる．探索木が厳密には木ではなく，DAG を形成する箇所があるために，定義通りに証明数・反証数を計算すると，特定の節点の証明数や反証数を二重にカウントしてしまうという問題である．例えば，この問題に対する対策として，本論文のプログラムでは局面表に親節点へのポインタを格納することによって DAG を検出する手法[6] を実装している．このアイデアを拡張した手法が source node detection algorithm(SNDA)[5] である．類似の手法として，現在探索中の局面に至るパス中の分岐数に着目した経路分岐数探索[11] や weak proof-number search[9] がある．

4章に述べるヒューリスティックス C(あの手にはこの手ハッシュ) は鶴岡の工夫[12] や context killer heuristic[3] に似ている．違いを挙げるとすると，上記の2手法は $\alpha\beta$ 法にて用いられ，直前の (複数の) 手が合致すれば，かなり高い優先度で登録された手が探索される．それに対し，ヒューリスティック C は df-pn のような証明数探索で用い，証明数・反証数があくまでも主役であり，証明数や反証数が同一の手が複数ある場合にそれらの指し手の優先順位を決めるためだけに用いる．

同様に，killer heuristic[1] や history heuristic[7] にも似るが，証明数・反証数が同一の複数の手の優先順位を付けるためだけに用いる点が大きく異なる．

ここで河野の simulation[4] との類似性を議論する．河野の simulation は，それまでの探索で効果的だった一連の手順の応酬を記憶しておき，現在注目のよく類似した局面に対してもその一連の手順を試す手法である．これに対し，4章に述べるヒューリスティックスは一連の手順という形では記憶しない．ある局面にて指せる複数の手の間の優先度や，証明数・反証数のカウント法などに終始し，一連の手順を扱うことはない．その代わり，実装した数々のヒューリスティックスのいずれかがいつも効果を発揮すれば，結果的に一連の手順の応酬を誘導するように機能することはあるかもしれない．我々の現在の実装では，河野の simulation は導入していない．オーバーヘッドのわりに効果が薄いため[13] である．

## 4. 提案ヒューリスティックス

我々は前章で挙げた脊尾の合駒対策など以外にも次のヒューリスティックスを提案する．これらのヒューリスティックスの中には，高性能な詰将棋プログラムでは既に採り入れられているものもあると考えられるが，著者の知る限り，個々のヒューリスティックスの実際の効果について包括的に論じられたことはない．

### 4.1 ヒューリスティックス A(成って王手駒を取る手を優先)

王手をかけている駒を取る際に，成る手も成らない手も可能な場合，成る手のみをカウントする．成る手が詰むことが判明したら，不成をカウントする．一般的には成れるチャンスがあるなら成っておいた方が良いケースが多いためである．攻め方の手に対し同様の工夫をすることは多いと思うが，玉方の手に対しても似た工夫を行う．高性能な詰将棋プログラムでは既に盛り込まれている可能性も高いと考えられる．

### 4.2 ヒューリスティックス B(同種の駒で王手駒を取れる手は 1 手だけカウント)

王手をかけている駒を同種の駒で取る手が複数ある場合，それらの取る手のうち1手だけカウントする．その手が詰まない限り別の取る手は考慮しない．具体

例として，図 1 はメタ新世界 (941 手詰) の 316 手目の局面で，正解は▽２八同とであるが，王手をかけている２八の角を取れる「と金」は３つも存在する．どのと金で取っても，先の展開は同じである．つまり，取る手の合法手としては３手あっても実質的には１手にすぎない．そこで AND 節点の証明数の計算の際に１手だけカウントする．

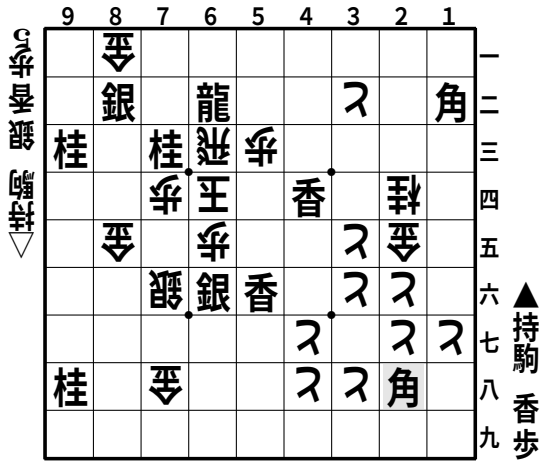


**図 1** メタ新世界の 316 手目，▲２八同角の局面．▽同とが正解だが，どのと金で取るかには任意性がある．

### 4.3 ヒューリスティック C (あの手にはこの手ハッシュ)

基本的なアイデアは，OR 節点で詰む手を発見したときや，AND 節点で不詰の手を発見したときに，その手を専用のハッシュに登録し，他の類似局面[⋆1]においてその手を優先的に探索する．ハッシュ関数のキー (入力) には，直前の相手の手を利用する．

具体例として，図 2 は「寿」の 259 手目▲７七桂に対し，260 手目▽同ととすべきところを▽９六玉と逃げた変化局面であるが，▲９七銀から 100 手前後の長い手順になるが詰む．詰むことが判明した時に，「▽９六玉」なるキーに「▲９七銀」をハッシュ登録する．図 2 と殆ど同じ局面は 346 手目，432 手目，520 手目に繰り返し現れるが，ハッシュを引くことにより「▲９七銀」を優先的に調べることができ，探索の効率が上がる．実際の利用方法としては，OR 節点で探索候補である王手のうち証明数最小のものが複数ある場合に，ハッシュにヒットした王手の優先度を上げている．例えば図 2 の場合「▽９六玉」の直後の「▲９七銀」の優先度が高くなる．

また，「▽９六玉」なるキーに「▲９七銀」を登録する方法であるが，我々の実装では移動手については，移動元と移動先のマスの情報のみを利用しており，打つ手に関しては駒の種類と打つマスの情報を利用している．例えば図 2 の例の場合，「▽９六玉」は「85 から 96 への移動手」，「▲９七銀」は「９七銀打」という情報に置き換えて登録している．

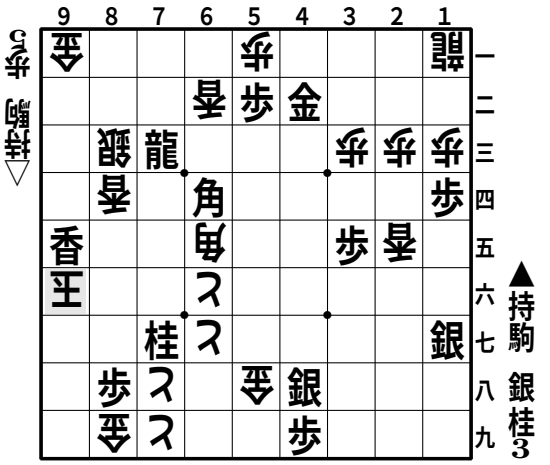


**図 2** 「寿」の 259 手目▲７七桂に対し，▽同ととすべきところを▽９六玉と逃げた変化局面．▲９七銀から 100 手近い手順になるが簡単に詰む．ハッシュの利用により，この結果を再利用できる．

### 4.4 ヒューリスティックス D (repeating moves)

基本的なアイデアは，長手数の詰将棋の特徴を利用することにある．長手数の詰将棋の詰み手順中には，殆ど同一の手順が繰り返し含まれることがしばしばある．そこで，現在探索中の局面に至るまでの手順中に同じ盤面 (持駒は違う) の局面が出現している場合，以前の局面で指した手と同じ手の優先度を上げる．この場合の優先度も，証明数最小の王手が複数ある場合に利用する．

### 4.5 ヒューリスティックス E (探索木の小さい手を優先)

基本的なアイデアは，OR 節点 (AND 節点) におい

⋆1 ここで言う「類似局面」とは，直前の手がハッシュ関数のキーの手に一致し，かつハッシュに登録されている手が合法手になるという意味で類似した局面である，と表現しているにすぎない．直接盤面上の駒の配置を見ているわけではないので，実は似ても似つかぬ局面同士の可能性もある．しかし局所的には似ている可能性が高いことに期待している．その意味ではかなり広義の類似局面である．

て証明数 (反証数) 最小の手が複数ある場合に，あまり探索していない方の手を優先的に探索することである．

例えば OR 節点にて王手が２つあり，現在共に証明数が２になっているとする．一方の王手はほんの数局面調べただけで証明数２を得ており，もう一方の王手は玉を追い回す手があり，非常に多くの局面を探索した挙げ句に証明数２を得ているとする．この場合，前者の王手を探索した方が少ない労力で詰み・不詰などの結論を得られる可能性が高い．

脊尾の工夫として，初めて探索する AND 節点にて，受けの手の数が証明数のしきい値を越えている場合，本来ならその AND 節点の探索はすぐに打ち切るべきであるが，受けたあとの１手詰だけは調べた方が良いという工夫がある．(我々のプログラムもこの１手詰チェックを導入している．) 探索木の小さい手を優先するということは，結局のところ，すぐ先に待っている詰み・不詰を見逃さないという意味で脊尾の工夫の延長線上にある．

### 4.6 ヒューリスティックス F (王手の反証数を 10 方向だけカウント)

基本的なアイデアは，OR 節点で反証数を計算する際に，同一方向からの王手については１手しかカウントしないことである．ここでいう方向とは，上下左右斜めの８方向に加え，桂馬による王手の２方向を加えた 10 方向のことである．

本来，OR 節点における反証数はすべての王手の和にて定義する．しかし AND 節点における証明数と同様に，合法手としては多数あっても，本質的には１つの手にすぎない状況がある．例えば飛び駒を打つ手については，近付けて打つ手や離して打つ手があるが，１手しかカウントしていないプログラムが多いと思われる．

ヒューリスティックス F はその考えをさらに発展させたものである．OR 節点，つまり攻め方手番のときに，角で王手するのと銀で王手するのとでは，その先の展開が似ていることがしばしばある．同様に飛車で横から王手するのと，金で横から王手するのとでは，その先の展開が似ていることがしばしばある．それは「角銀ゼット」「飛車金ゼット」という用語があることからも裏付けられる．そこで，同じ方向からの王手は各方向ごとに１手しかカウントしない．

## 5. 実 験 結 果

合駒対策など既存のヒューリスティックスを導入したプログラムを基準に，さらに A〜F まで 6 種類の提案ヒューリスティックスを一つずつ導入した場合のパフォーマンスを調べる．

実験環境は Intel Core i7-860 (2.8GHz) である．メモリは 4GB 搭載しているが，局面表に 1GB だけ使用する．

表１は実験対象とする問題である．問題１は将棋図巧のうち，不詰の１題を除く 99 題の問題をすべて解くのにかかる時間を以って評価する．提案ヒューリスティックスには手順の繰り返しの多い長手数詰将棋を意識しているものもあるが，そうでない詰将棋への影響を知るため，将棋図巧も対象とした．その他の問題の選択基準は，長手数詰将棋の中でも特に解答時間のかかるものを選んだ．

| 問題 | 作品名 |
|---|---|
| 1 | 将棋図巧 (不詰の第 73 番を除く全作品) |
| 2 | 田島作品 (345 手) |
| 3 | 呪われた夜 |
| 4 | 乱 |
| 5 | 赤兎馬 |
| 6 | アルカナ |
| 7 | メタ新世界 |
| 8 | ミクロコスモス |
| 9 | 回転木馬 |
| 10 | メガロポリス (修正図) |

**表 1** 問題番号と対応する作品名

#### ヒューリスティックスを 1 つだけ導入

表２は提案ヒューリスティックスを導入しない場合と，１つだけ導入した場合の実行時間とその比である．提案ヒューリスティックスを導入しない場合と，１つだけ導入した場合の実行時間の比較である．表２からヒューリスティックス B を導入するのがベストであることが分かる．ただし注意すべき点として，長手数詰将棋でない将棋図巧のような問題に対しては３割ほどパフォーマンスが落ちていることである．

#### 2 つ目のヒューリスティックスを導入

表３は提案ヒューリスティックス B のみを導入した場合と，さらにもう１つ導入した場合の実行時間とその比である．表 3 からヒューリスティックス B と C を導入するのがベストであることが分かる．ただし注意すべき点として，長手数詰将棋でない将棋図巧のような問題に対しては６割ほどパフォーマンスが落ちていることである．

#### 3 つ目のヒューリスティックスを導入

表４は提案ヒューリスティックス B,C を導入した場合と，さらにもう１つ導入した場合の実行時間とその

| | 導入ヒューリスティックス | | | |
|---|---|---|---|---|
| 問題 | なし | A (比) | B (比) | C (比) |
| 1 | 150 | 176 (1.18) | 197 (1.32) | 155 (1.03) |
| 2 | 208 | 203 (0.98) | 209 (1.01) | 212 (1.02) |
| 3 | 275 | 263 (0.96) | 216 (0.79) | 268 (0.97) |
| 4 | 5862 | 5313 (0.91) | 5502 (0.94) | 4861 (0.83) |
| 5 | 1862 | 2095 (1.12) | 1717 (0.92) | 1144 (0.61) |
| 6 | 222 | 212 (0.95) | 178 (0.80) | 209 (0.94) |
| 7 | 5876 | 5911 (1.01) | 1905 (0.32) | 5611 (0.95) |
| 8 | 124 | 123 (1.00) | 124 (1.00) | 129 (1.04) |
| 9 | 341 | 221 (0.65) | 370 (1.08) | 232 (0.68) |
| 10 | 2878 | 2799 (0.97) | 546 (0.19) | 2914 (1.01) |
| 平均 | 1780 | 1732 (0.97) | 1096 **(0.84)** | 1573 (0.91) |

| | 導入ヒューリスティックス | | | |
|---|---|---|---|---|
| 問題 | なし | D (比) | E (比) | F (比) |
| 1 | 150 | 158 (1.06) | 167 (1.11) | 150 (1.00) |
| 2 | 208 | 184 (0.89) | 211 (1.01) | 374 (1.80) |
| 3 | 275 | 270 (0.98) | 304 (1.10) | 277 (1.01) |
| 4 | 5862 | 5367 (0.92) | 5913 (1.01) | 7196 (1.23) |
| 5 | 1862 | 1380 (0.74) | 3353 (1.80) | 2334 (1.25) |
| 6 | 222 | 223 (1.00) | 223 (1.00) | 187 (0.84) |
| 7 | 5876 | 6465 (1.10) | 6203 (1.06) | 6565 (1.12) |
| 8 | 124 | 125 (1.01) | 110 (0.89) | 105 (0.85) |
| 9 | 341 | 353 (1.04) | 307 (0.90) | 399 (1.17) |
| 10 | 2878 | 2970 (1.03) | 3354 (1.17) | 2577 (0.90) |
| 平均 | 1780 | 1750 (0.98) | 2014 (1.10) | 2016 (1.12) |

**表 2** 実行時間 (単位は秒)．提案ヒューリスティックスを導入していないものと，一つ導入したものの比較．

| | 導入ヒューリスティックス | | | |
|---|---|---|---|---|
| 問題 | B | B+A (比) | B+C (比) | B+D (比) |
| 1 | 197 | 147 (0.75) | 317 (1.61) | 141 (0.71) |
| 2 | 209 | 242 (1.15) | 206 (0.99) | 204 (0.97) |
| 3 | 216 | 243 (1.13) | 180 (0.83) | 218 (1.01) |
| 4 | 5502 | 5126 (0.93) | 4938 (0.90) | 5876 (1.07) |
| 5 | 1717 | 1534 (0.89) | 1440 (0.84) | 2043 (1.19) |
| 6 | 178 | 159 (0.89) | 143 (0.81) | 168 (0.95) |
| 7 | 1905 | 2057 (1.08) | 1893 (0.99) | 1879 (0.99) |
| 8 | 124 | 120 (0.97) | 132 (1.07) | 121 (0.98) |
| 9 | 370 | 387 (1.05) | 257 (0.70) | 241 (0.65) |
| 10 | 546 | 549 (1.01) | 450 (0.82) | 583 (1.07) |
| 平均 | 1096 | 1056 (0.98) | 996 **(0.95)** | 1147 (0.96) |

| | 導入ヒューリスティックス | | |
|---|---|---|---|
| 問題 | B | B+E (比) | B+F (比) |
| 1 | 197 | 199 (1.01) | 117 (0.60) |
| 2 | 209 | 228 (1.09) | 294 (1.40) |
| 3 | 216 | 190 (0.88) | 164 (0.76) |
| 4 | 5502 | 4595 (0.84) | 8909 (1.62) |
| 5 | 1717 | 6372 (3.71) | 3751 (2.18) |
| 6 | 178 | 151 (0.85) | 154 (0.86) |
| 7 | 1905 | 1781 (0.93) | 1816 (0.95) |
| 8 | 124 | 106 (0.85) | 123 (0.99) |
| 9 | 370 | 463 (1.25) | 328 (0.89) |
| 10 | 546 | 387 (0.71) | 298 (0.55) |
| 平均 | 1096 | 1447 (1.21) | 1595 (1.08) |

**表 3** 実行時間 (単位は秒)．提案ヒューリスティックス B を導入したものと，それ以外にもう一つ導入したものの比較．

比である．表 4 から，ヒューリスティックス A を導入したものが僅かに改善している．つまり，ヒューリスティックス A,B,C を導入するのがベストであることが分かる．

### 4 つ目のヒューリスティックスを導入

表 5 は提案ヒューリスティックス A,B,C を導入した場合と，さらにもう 1 つ導入した場合の実行時間とその比である．表 5 からヒューリスティックス A,B,C 以外に導入しない方が良いことが分かる．

### 3 ヒューリスティックス A,B,C を合わせた効果

表 6 は提案ヒューリスティックスを全く導入しない場合と，A,B,C を導入した場合の実行時間とその比である．表 7 は同じく局面展開数とその比である．表 6 と表 7 から提案ヒューリスティックス A,B,C の導入により，実行時間を約 77%に抑えるパフォーマンスの向上が見られる．局面展開数は約 75%に抑えられた．

ただし注意すべき点として，長手数詰将棋でない将棋図巧のような問題に対しては 4〜6 割ほどパフォーマンスが落ちていることである．その理由は，ヒューリスティックス C の発動条件がいい加減であることに起因すると思われる．意図としては，勝てる手を発見したときにその手を記憶しておき，その後出現した類似局面に対して記憶していた手を優先的に探索するというものである．しかしこの場合の「類似局面」は直接盤上の駒を見ているわけではなく，連続する 2 手 (直前の手と，現在の局面で或る手を指せるかどうか) にしか着目していない．ということは，局所的には類似していても大局的には全然違う局面を「類似局面」と錯覚することは十分起こり得る．このようないい加減さを含んでいる．さらに，我々のプログラムで将棋図巧を解く場合，実は第 8 番の問題に全実行時間の 6〜7 割を費している．つまり第 8 番の実行時間に大いに左右されている．この詰将棋の特徴は，角の遠打を 2 度行い玉方の龍を微妙に移動させるという趣向の問題で，要は盤面を広く見なければならない．ヒューリスティックス C のいい加減な「類似局面」の概念は，将棋図巧第 8 番には逆効果なのであろう．

### DAG 検出の効果

実は 3 章にも述べたように，実験に用いたプログラムは局面表に親節点へのポインタを格納することによって DAG を検出する手法[6] を実装している．これも一種のヒューリスティックスと言えるかもしれない．そこで最後に，この DAG 検出の効果を調べる．提案

| | 導入ヒューリスティックス | | | |
|---|---|---|---|---|
| 問題 | BC | BC+A (比) | BC+D (比) | BC+E (比) |
| 1 | 317 | 247 (0.78) | 214 (0.67) | 260 (0.82) |
| 2 | 206 | 221 (1.07) | 139 (0.67) | 202 (0.98) |
| 3 | 180 | 184 (1.02) | 178 (0.99) | 176 (0.98) |
| 4 | 4938 | 4862 (0.98) | 4155 (0.84) | 4932 (1.00) |
| 5 | 1440 | 1625 (1.13) | 2364 (1.64) | 5109 (3.55) |
| 6 | 143 | 148 (1.03) | 157 (1.09) | 142 (0.99) |
| 7 | 1893 | 1893 (1.00) | 1937 (1.02) | 1579 (0.83) |
| 8 | 132 | 126 (0.95) | 136 (1.03) | 108 (0.81) |
| 9 | 257 | 168 (0.65) | 227 (0.88) | 272 (1.06) |
| 10 | 450 | 432 (0.96) | 466 (1.04) | 323 (0.72) |
| 平均 | 996 | 991 **(0.96)** | 997 (0.99) | 1310 (1.17) |

| | 導入ヒューリスティックス | |
|---|---|---|
| 問題 | BC | BC+F (比) |
| 1 | 317 | 190 (0.60) |
| 2 | 206 | 196 (0.95) |
| 3 | 180 | 134 (0.75) |
| 4 | 4938 | 3576 (0.72) |
| 5 | 1440 | 3597 (2.50) |
| 6 | 143 | 133 (0.92) |
| 7 | 1893 | 1762 (0.93) |
| 8 | 132 | 132 (1.00) |
| 9 | 257 | 273 (1.06) |
| 10 | 450 | 238 (0.53) |
| 平均 | 996 | 1023 (1.00) |

**表 4** 実行時間 (単位は秒)．提案ヒューリスティックス B,C を導入したものと，それ以外にもう一つ導入したものの比較．

| | 導入ヒューリスティックス | | | |
|---|---|---|---|---|
| 問題 | ABC | ABC+D (比) | ABC+E (比) | ABC+F (比) |
| 1 | 247 | 201 (0.81) | 259 (1.05) | 690 (2.79) |
| 2 | 221 | 217 (0.98) | 200 (0.90) | 283 (1.28) |
| 3 | 184 | 193 (1.05) | 184 (1.00) | 156 (0.85) |
| 4 | 4862 | 6355 (1.31) | 4839 (1.00) | 5320 (1.09) |
| 5 | 1625 | 3330 (2.05) | 2732 (1.68) | 1956 (1.20) |
| 6 | 148 | 150 (1.02) | 154 (1.04) | 135 (0.91) |
| 7 | 1893 | 2025 (1.07) | 1749 (0.92) | 1891 (1.00) |
| 8 | 126 | 150 (1.19) | 136 (1.08) | 146 (1.16) |
| 9 | 168 | 346 (2.06) | 219 (1.31) | 257 (1.53) |
| 10 | 432 | 477 (1.10) | 292 (0.68) | 267 (0.62) |
| 平均 | 991 | 1344 (1.26) | 1076 (1.07) | 1110 (1.24) |

**表 5** 実行時間 (単位は秒)．提案ヒューリスティックス A,B,C を導入したものと，それ以外にもう一つ導入したものの比較．

ヒューリスティックス A,B,C を導入した上で，DAG 検出も導入したものとしないものとの比較結果を表 8 と表 9 に載せる．表 8 は実行時間で評価し，表 9 は局面展開数で評価している．表 8 と表 9 から，DAG 検出は実装しておいた方が多くの問題で解答速度が上がることが分かる．DAG 検出をしないと，平均で 7 割ほど実行時間も局面展開数も悪化することが分かる．これは証明数の二重カウントに起因する解答速度の低下と推定できる．

| | 導入ヒューリスティックス | |
|---|---|---|
| 問題 | なし | ABC (比) |
| 1 | 150 | 247 (1.65) |
| 2 | 208 | 221 (1.06) |
| 3 | 275 | 184 (0.67) |
| 4 | 5862 | 4862 (0.83) |
| 5 | 1862 | 1625 (0.87) |
| 6 | 222 | 148 (0.67) |
| 7 | 5876 | 1893 (0.32) |
| 8 | 124 | 126 (1.02) |
| 9 | 341 | 168 (0.49) |
| 10 | 2878 | 432 (0.15) |
| 平均 | 1780 | 991 (0.77) |

**表 6** 実行時間 (単位は秒)．提案ヒューリスティックスを導入していないものと，A,B,C を導入したものの比較．

| | 導入ヒューリスティックス | |
|---|---|---|
| 問題 | なし | ABC (比) |
| 1 | 62809365 | 89115745 (1.42) |
| 2 | 86329858 | 89448568 (1.04) |
| 3 | 119709995 | 77577265 (0.65) |
| 4 | 2032235611 | 1804430053 (0.89) |
| 5 | 711056176 | 608205660 (0.86) |
| 6 | 99950596 | 65505585 (0.66) |
| 7 | 1967327456 | 703305307 (0.36) |
| 8 | 52112106 | 53218535 (1.02) |
| 9 | 123117775 | 61714511 (0.50) |
| 10 | 1117865921 | 185176303 (0.17) |
| 平均 | 637251486 | 373769753 (0.75) |

**表 7** 局面展開数 (単位は回)．提案ヒューリスティックスを導入していないものと，A,B,C を導入したものの比較．

| | 導入ヒューリスティックス | |
|---|---|---|
| 問題 | ABC | ABC without DAG 検出 (比) |
| 1 | 247 | 183 (0.74) |
| 2 | 221 | 231 (1.05) |
| 3 | 184 | 182 (0.99) |
| 4 | 4862 | 8432 (1.73) |
| 5 | 1625 | 2265 (1.39) |
| 6 | 148 | 142 (0.96) |
| 7 | 1893 | 10769 (5.69) |
| 8 | 126 | 118 (0.93) |
| 9 | 168 | 425 (2.53) |
| 10 | 432 | 465 (1.08) |
| 平均 | 991 | 2321 (1.71) |

**表 8** 実行時間 (単位は秒)．DAG を検出[6] による効果．ただし提案ヒューリスティックス A,B,C を導入．

## 6. ま と め

df-pn アルゴリズムで長編詰将棋を解く際に，3 つのヒューリスティックスを導入すると実行時間を約 77%に抑えられた．その 3 つのヒューリスティックスとは，(1) 王手をかけている駒を取る際に，成る手も成らない手も可能な場合，成る手のみをカウントする．成る

| | 導入ヒューリスティックス | |
|---|---|---|
| 問題 | ABC | ABC without DAG 検出 (比) |
| 1 | 89115745 | 65887434 (0.74) |
| 2 | 89448568 | 98685925 (1.10) |
| 3 | 77577265 | 82625073 (1.07) |
| 4 | 1804430053 | 3501729960 (1.94) |
| 5 | 608205660 | 885798800 (1.46) |
| 6 | 65505585 | 69915593 (1.07) |
| 7 | 703305307 | 3587785889 (5.10) |
| 8 | 53218535 | 50729714 (0.95) |
| 9 | 61714511 | 152952592 (2.48) |
| 10 | 185176303 | 198937222 (1.07) |
| 平均 | 373769753 | 869504820 (1.70) |

**表 9** 局面展開数 (単位は回). DAG を検出[6] による効果. ただし提案ヒューリスティックス A,B,C を導入.

手が詰んだら不成をカウントする. (2) 王手をかけている駒を同種の駒で取る手が複数ある場合, それらの取る手のうち 1 手だけカウントする. (3) 攻め方手番で詰む手を発見したときや, 玉方手番で不詰の手を発見したときに, その手を登録し他の類似局面においてもその手を優先的に探索する.

ただしこれら 3 つのヒューリスティックスはあらゆる種類の詰将棋に対して効果的というわけではなく, 一部の詰将棋に対しては逆効果になることもある. また, 証明数・反証数の二重カウントを防ぐための DAG 検出を導入しないと, 平均 7 割の解答時間の増加を招くことが分かった.

## 参 考 文 献


1) Selim G. Akl and Monroe M. Newborn. The Principal Continuation and the Killer Heuristic. *ACM Annual Conference Proceedings*, pp. 466–473, 1977.
2) L.V. Allis, M.vander Meulen, and H.J. vanden Herik. Proof-Number Search. Technical Report CS 91-01, University of Limburg, Maastricht, Netherlands, 1991. Also available as *Artificial Intelligence*, Vol.66, pp. 91–124, 1994.
3) J.Hashimoto, T.Hashimoto, and H.Iida. Context Killer Heuristic and its Application to Computer Shogi. In *Proceedings of Computer Games Workshop*, pp. 39–48, 2007.
4) Y.Kawano. Using Similar Positions to Search Game Trees. In *Games of No Chance, MSRI Publications*, Vol.29, 1996.
5) A. Kishimoto. Dealing with Infinite Loops, Underestimation, and Overestimation of Depth-First Proof-Number Search. In *Proceeding of the 24th AAAI Conference on Artificial Intelligence (AAAI-10)*, pp. 108–113, 2010.
6) A. Nagai. *Df-pn Algorithm for Searching AND/OR Trees and Its Applications*. PhD thesis, Department of Information Science, University of Tokyo, Japan, 2002.
7) J.Schaeffer. The History Heuristic and Alpha-Beta Search Enhancements in Practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.11, No.1, pp. 1203–1212, 1989.
8) M.Seo. The C* Algorithm for AND/OR Tree Search and its Application to a Tsume-Shogi Program. Master's thesis, Department of Information Science, University of Tokyo, Japan, 1995.
9) Toru Ueda, Tsuyoshi Hashimoto, Junichi Hashimoto, and Hiroyuki Iida. Weak Proof-Number Search. In *Computers and Games 2008 (CG 2008)*, LNCS 5131, pp. 157–168, 2008.
10) 伊藤琢巳, 河野泰人, 野下浩平. 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. 情報処理学会論文誌, Vol.36, No.12, pp. 2793–2799, 1995.
11) 岡部文洋. 経路分岐数を用いた詰め将棋解図について. 第 10 回ゲームプログラミング・ワークショップ (GPW2005), pp. 9–16, 2005.
12) 鶴岡慶雅. 将棋プログラムの現状と未来. 情報処理, pp. 817–822, 2005.
13) 脊尾昌宏. 個人的な対話.
14) 脊尾昌宏. 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, No. 99-14, pp. 103–110, 1995.
15) 脊尾昌宏. 共謀数を応用した詰め将棋プログラムについて. ゲーム・プログラミング ワークショップ '95, pp. 128–137, 1995.
16) 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. In *Game Programming Workshop in Japan '99*, pp. 129–136, 1999.
17) 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.