Vol.2017-GI-37 No.7
2017/3/7

初期配置が指定された場合における高難易度数独問題の自動生成手法の提案および実装

座間 翔[1,a)]　篠埜 功[1]

概要：問題集などに掲載される数独の問題は、初期配置の数字が図や模様を描くように配置されているものがある。本研究ではそのような視覚的なデザインを考慮した問題制作を支援するため、初期配置の位置を問題制作者に指定させ、それに従って問題を自動生成する手法を提案する。この手法では、指定されたマスへの数字配置と配置した数字の変更によって問題を生成する。その際、マスに入る数字の可能性を絞り込むような数字配置手法や、解探索で解の求まらないマスを減らす数字変更基準を設け、唯一解を持つ問題生成の成功率を高める。また、唯一解の探索は人間が通常用いる解法を実装した解探索アルゴリズムにより行う。さらに、制作者による問題の難易度設定の補助として、より難易度の高い問題を自動生成する手法を考案し、実装した。

1. はじめに

数独とは、マスが敷き詰められた盤面に対して、数字を一つずつ当てはめていくパズルである。一般的に、数独の盤面は 9 × 9 のマスから構成されており、以下において、横につながっている 9 つのマスの集まりを行、縦につながっている 9 つのマスの集まりを列、盤面の内部を 9 つに分割する 3 × 3 のマスの集まりをブロックと呼び、それらを総じて領域と呼ぶ事にする。回答者は、あらかじめ盤面に与えられている初期配置の数字から、各空白マスに入り得る数字を判断し、盤面を埋めていくが、その際に、各領域内には 1 から 9 までの数字が 1 つずつ入るという制約を守らなければならない。**図 1** に数独の問題例を挙げる。

数独は世界的な人気を誇るペンシルパズルであり、数独の問題を掲載した専門雑誌や問題集は数多く発刊されている。これらの媒体に掲載される数独の問題は、初期配置が何らかの規則的なパターンを描いている事が多く、問題生成時に難易度の設定と視覚的なデザインとのバランスを取る事が難しい。

数独に関する研究は難易度判定に関するもの [1] や、問題の面白さに関するもの [2]、解盤面の探索に関するもの [3] など数多く行われているが、数独の問題生成に関する研究は未だ少ない。また、生成に関する既存研究として初期配置の位置を考慮した問題の自動生成手法に関する研究 [4] や、少数ヒント問題の生成手法に関する研究 [5] があるものの、生成可能な問題の幅が狭い、デザイン性を考慮していないといった点から、問題集などの制作支援としては不十分であると言える。そこで本研究では、問題集などの制作を支援するため、一定の問題生成能力と問題制作者による初期配置の位置指定を両立した、問題の自動生成手法を提案する。本手法では、生成した問題が人間が解く問題として適していることを判定するために、人間が実際に使用できる解法による解探索アルゴリズムを唯一解の解探索に用いた。さらに、問題制作者による難易度設定の補助として、難易度を考慮した問題生成手法を提案する。難易度を考慮可能な既存の自動生成ツールとして Number Place Generator Version2.0.2[6] などが挙げられるが、これは生成した問題を難易度でフィルタリングするという手法を用いており、特定の難易度の問題のみを生成できる手法ではない。そこで本研究では、解探索に用いた解法の難度に着目し、より高難易度の問題を自動生成する手法を実装した。

			9	6				3
	6				1			9
1			3				7	6
2	4							
				8				
	8	9		1		2		
7				9				
				3	8	4		
8			7			9		2

図 1　数独の問題例

1　芝浦工業大学
Shibaura Institute of Technology
a)　ma15046@shibaura-it.ac.jp

2. 提案手法

本章では、本研究で提案する問題生成手法を示す。この手法は、利用者の初期配置の指定に対して、数字配置、解探索、数字変更の 3 段階によって問題を生成する。

2.1 問題生成手法

既存の問題生成手法として、前田らは初期配置の位置を指定可能な問題生成手法を提案している [4]。この手法はデザイン性を考慮した問題生成が可能だが、初期配置数が 24 個以下の問題の生成成功率が非常に低いという問題点がある。また、初期配置数の少ない問題の生成手法として、那須らの手法 [5] やとん氏の手法 [7] がある。これらの手法を用いれば初期配置数が 17 個の問題も生成できるが、那須らの手法 [5] では初期配置の位置の指定は想定しておらず、とん氏の手法 [7] では初期配置の位置の変更を行う場合がある。よって、本研究ではデザイン性の考慮を可能としつつ、大量の問題を高速に生成できる手法を提案する。

本研究で提案する手法では、まず問題制作者に初期配置のマスを指定してもらい、それらのマスに対して数字配置を行う。指定されたマスの全てに数字を配置し終えると、その盤面が唯一解を持ち、数独の問題として成立しているかどうかを確かめる為に、解探索を行う。解探索の結果、盤面が唯一解を持っていたならば、問題生成を成功とし、盤面を問題として出力する。唯一解を見つけることができなかった場合は、とん氏の手法 [7] と同様に、あらかじめ設けた基準にしたがって配置した数字を変更し、新たな盤面を作り出して再び解探索を行う。基準を満たすような数字変更が行えなくなった場合には、配置された数字を全て取り消し、数字配置の段階からやり直す。アルゴリズム 1 に問題生成の手順を示す。

この手法は初期配置の指定によってデザイン性を考慮した上で問題を自動生成することができる。また、数字変更を行うことで、配置を最初からやり直すよりも効率的に盤面を生成することができる。数字配置、解探索、数字変更の各段階の詳細は次節以降に述べる。

2.2 数字配置

数字配置の手法として、始めに解盤面を生成し、その後に初期配置以外の数字を消して空白にするという解盤面生成方式による手法がある。先行研究 [8] ではこの手法で問題生成を行っていたが、解盤面の生成に時間がかかるという欠点を持っていた。よって、本研究では、初期配置として指定されたマスにのみ数字を配置する手法を採用する。また、配置する数字を決定する手法として、マスと数字をランダムに決定するランダム決定手法と、候補数字という概念に着目する絞り込み手法の 2 つを導入する。

Algorithm 1 . 問題生成手順

```
while true do
  数字配置実行
  解探索実行
  if 唯一解発見 then
    return 生成した盤面
  end if
  while 数字変更の余地あり do
    数字変更実行
    解探索実行
    if 唯一解発見 then
      return 生成した盤面
    end if
  end while
end while
```

2.2.1 候補数字

本研究の提案手法では、数字配置や数字変更の際に候補数字という概念に着目する。本論文では、あるマス X が持つ候補数字 C_X を以下のように定義する。ここで、I_X は X と同じ領域内における確定済みの数字である。

$$C_X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} - I_X$$

数独の盤面の一部を示した図 2 を例とすると、図中の X と書かれたマスは、色のついた同じ行、列、ブロックの領域内のマスに、1,2,3,4,5 の数字が既に確定している。この時、X のマスの候補数字は以下の通りである。

$$C_X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\} - \{1, 2, 3, 4, 5\} = \{6, 7, 8, 9\}$$

数独の制約から、1 つの領域内のマスには 1 から 9 の数字が 1 つずつのみ入るため、X のマスには 1,2,3,4,5 の 5 種の数字は入らない。そのため、X のマスは 1 から 9 の数字から 1,2,3,4,5 の 5 種を除いた 6,7,8,9 の 4 つの数字を候補数字として持つことになる。

マスに配置する数字はそのマスの持つ候補数字の中のいずれか 1 つであり、ランダム決定手法では候補数字の中からランダムに、絞り込み手法では他の空白マスの候補数字の個数に着目して 1 つの数字を選び出し、マスに配置していく。

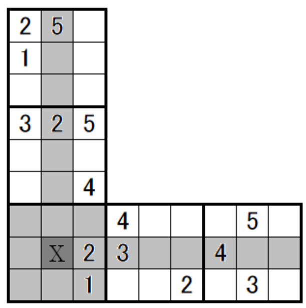

図 2 数独の問題の一部

2.2.2 ランダム決定手法

ランダム決定手法では、問題の制作者から指定されたマスの中から１つの空白マスをランダムに決定し、決定したマスが持つ候補数字の１つをランダムで配置する。あるマスへの数字配置によって、候補数字を持たない空白マスが発生してしまった場合には、配置した数字を取り消し、そのマスの候補数字から配置した数字を取り除く。それによって、そのマスの候補数字がなくなった場合には、全ての数字を取り消し、数字配置を最初からやり直す。アルゴリズム２にこの手順を示す。

ランダム決定手法は単純な処理で実現でき、数字配置にかかる実行時間が短く済むために、短時間で試行回数を多く確保できるという利点がある。しかし、4.1 節で述べるように、決定がランダムなために解探索に成功する盤面となる可能性は低くなる。

Algorithm 2．ランダム決定手法

```
while 空白の初期配置マスが存在 do
  cell ← ランダム決定した空白の初期配置マス
  num ← ランダム決定した cell の候補数字の１つ
  cell に num を配置
  if 数字の入らないマスが発生 then
    cell の候補数字から num を消去
  end if
  if cell の候補数字が無い then
    配置した数字を全て取り消し
  end if
end while
```

2.2.3 絞り込み手法

絞り込み手法は、各空白マスの持つ候補数字の数をなるべく小さくするように数字配置を行うことで、解探索に成功する可能性が高い盤面の生成を目指す。その際、とん氏の手法 [7] で数字変更に用いられていた、候補数字の数の２乗和を基準に用いる。絞り込み手法では、初期配置として指定されたマスの内、まだ数字が配置されていないマスについて、そのマスの持つ候補数字を仮置きする。そして、仮置きした盤面の各空白マスが持つ候補数字の数を数え、その２乗和の値を算出する。その後、盤面を仮置き前のものに復元する。これを全ての空白の初期配置マスとその候補数字について行い、候補数字数の２乗和の値が最も小さくなった配置を盤面へと反映する。この手順を全ての初期配置マスに数字を配置するまで繰り返す。なお、候補数字数の２乗和が最も小さくなるような配置が複数通りあった場合は、ランダムで１つを選び出し、盤面へ反映する。また、仮置きの際に、候補数字が１つも無いマスが出現してしまう配置であると分かった場合は、盤面を復元した後、仮置きした数字を配置したマスの候補数字から取り除く。これによってそのマスの候補数字がなくなってしまった場合は、既に配置した数字を全て取り消し、数字配置を最初からやり直す。アルゴリズム３にこの手順を示す。

絞り込み手法は、候補数字の数を減らすことによって各マスに入る数字が１つにまで絞り込みやすくできるため、4.1 節で述べるように、解探索に成功するような盤面を得られる可能性が高くなる。ただし、ランダム決定手法に比べると処理が複雑であり、実行時間が長くかかるため、一定時間での試行回数は少なくなる。

Algorithm 3．絞り込み手法

```
min ← 9 * 9 * 9
while 空白の初期配置マスが存在 do
  for cell ∈ 空白の初期配置マス群 do
    for num ∈ cell の候補数字群 do
      cell に num を配置
      if 数字の入らないマスが発生 then
        配置を取り消し
        cell の候補数字から num を消去
        continue;
      end if
      temp ← 盤面の候補数字の２乗和
      if temp < min then
        min ← temp
        配置するマスと数字の組の記録を更新
      else if temp = min then
        配置するマスと数字の組の記録に追加
      end if
      配置を取り消し
    end for
  end for
  if 候補数字が無い初期配置マスが存在 then
    配置のリセット
    min ← 9 * 9 * 9
  end if
  マスと数字の組の記録からランダムで１つを反映
end while
```

2.3 解探索手法

本研究の手法では人間が実際に用いる解法を解探索に取り入れることで、人間が解くのに適した問題を生成する。これにより、この手法で唯一解を発見できた盤面は、多段階の仮置きを要するような、人間が解く問題として不適切な問題ではないということが保証される。そこで、本研究の手法で実装した解法によって唯一解が発見できなかった盤面については、その盤面が実際に唯一解を持つかどうかの判定は行わず、解探索を失敗で終了する。

本研究の手法が解探索に用いる解法は次の６種 [9], [10] である。これらの解法を順次適用していき、唯一解の探索を行う。

- Singles[9]
- Intersections[9]
- NakedSubsets[9]
  - NakedPair
  - NakedTriple

  - NakedQuadruple
- HiddenSubsets[9]
  - HiddenPair
  - HiddenTriple
  - HiddenQuadruple
- BasicFish[9]
  - X-wing
  - SwordFish
  - JellyFish
- 浜田ロジック [10]

具体的な解探索手順を以下に述べる。まず、Singles の適用が可能であるかどうかを確認する。もし Singles の適用が不可能であったならば、次に Intersection の適用が可能かどうかを確認する。以後、同様に、NakedSubsets、HiddenSubset、BasicFish、浜田ロジックの順で解法の適用可否を確認する。もしいずれかの解法が適用可能であった場合には、その解法を適用して、再び Singles から適用可否を確認する。この手順を繰り返し、いずれの解法も適用できなくなったときに、盤面に解の求まっていない空白のマスが存在しなければ、解探索を成功とし、盤面を生成した問題として出力する。空白のマスが残っていた場合には、数字変更の段階へと移行し、異なる盤面を生成する。アルゴリズム 4 に解探索の手順を示す。

Algorithm 4 . 解探索手順

```
while true do
  if Singles 適用可能 then
    Singles 適用
    continue;
  else if Intersections 適用可能 then
    Intersections 適用
    continue;
  else if NakedSubsets 適用可能 then
    NakedSubsets 適用
    continue;
  else if HiddenSubsets 適用可能 then
    HiddenSubsets 適用
    continue;
  else if BasicFish 適用可能 then
    BasicFish 適用
    continue;
  else if 浜田ロジック適用可能 then
    浜田ロジック適用
    continue;
  else if 空白マスがない then
    return true;
  else
    return false;
  end if
end while
```

2.4 数字変更

数字配置によって生成した盤面が、本研究の手法で用いる解探索アルゴリズムによって唯一解を求めることができないものだったならば、盤面上に配置された数字を別の数字に変更することで異なる盤面を作り出し、唯一解を持つ問題の生成を試みる。本研究では、より解探索に成功する盤面を生成しやすいように、数字の変更を行う際の決定基準として、とん氏の手法 [7] で用いていたものと同様に、解探索によって解が求まらなかったマスの数が少なくなれば変更を採用するという空白マス数基準と、盤面の候補数字の個数の平方和が少なくなれば変更を採用するという候補数字基準 2 通りの基準を実装した。

2.4.1 空白マス数基準

空白マス数基準では、変更を行う前の盤面を解探索した結果、解の求まらなかったマスの数がいくつであったかを記録しておき、その上で次のように変更する数字を決定する。まず、初期配置の数字が配置されているマスを一つ選ぶ。次に、配置されている数字を取り消して空白に戻す。その後、現在配置されている数字を確定した数字として、空白にしたマスの持つ候補数字から取り消した数字以外のものを一つ選んで配置し直す。ここで、元々の数字以外に候補数字が無ければ、別の初期配置のマスを選び直す。そして、この数字変更によって得られた盤面を解探索し、唯一解を得られれば、盤面を問題として出力する。もし唯一解が得られなければ、解探索によって解を求めることができなかったマスの数を数え、その数が数字変更前に解の求まらなかったマスの数より少なければ、記録している解の求まらなかったマスの数を数字変更後のものに更新し、盤面の変更を保持する。変更前に解探索で解の求まらなかったマスの数と同じかそれより大きかった場合は、変更を取り消して変更前の数字を配置し直す。以降、唯一解を持つ盤面を得られるか、解の求まらないマスの数が減るような数字配置が出来なくなるまで、各初期配置のマスとその候補数字について、この手順の適用を繰り返す。アルゴリズム 5 にこの手順を示す。

盤面の解探索が成功した場合、盤面の全てのマスには数字が書き入れられており、空白マス数が 0 となっている。すなわち、解探索後の空白マス数が少なければ少ないほど、その盤面が解探索に成功する盤面に近いものと考えることができる。よって、解探索後の空白マス数を少なくしていく空白マス数基準による数字変更は、盤面を解探索に成功するような盤面へと近づけていくことに繋がるのではないかと考えられる。

2.4.2 候補数字基準

候補数字基準では、とん氏の手法 [7] と同様に、変更を行う前の盤面について、あらかじめ各空白マスの候補数字の数の 2 乗和を算出して記録しておき、それを基にして次のように変更する数字を決定する。まず、空白マス数基準と同様に、初期配置のマスを一つ選び、そのマスの数字を取り消して空白に戻し、元々の数字を除く候補数字の中から

Algorithm 5 . 空白マス数基準に基づいた変更手順

```
empty ← 9 * 9
while 盤面が変化 do
  for cell ∈ 初期配置マス群 do
    origin ← cell の元々の数字
    cell を空白にする
    for num ∈ cell の候補数字群 −origin do
      cell に num を配置
      解探索実行
      if 唯一解発見 then
        return 生成した盤面
      end if
      temp ← 盤面の空白マスの数
      if temp < empty then
        empty ← temp
        origin ← num
      end if
    end for
    cell に origin を配置
  end for
end while
```

1 つを選んで配置し直す。そして、その盤面について、各空白マスの持つ候補数字の数を求め、その 2 乗和を算出しておく。その後、解探索を行い、唯一解を得ることができれば、盤面を問題として出力する。もし唯一解が得られなければ、数字変更前と変更後の候補数字の数の 2 乗和の値を比較し、変更後の方が小さければ、記録した候補数字の数の 2 乗和の値を数字変更後のものに更新し、盤面の変更を保持する。変更前の方が小さければ、変更を取り消して変更前の数字を配置し直す。以降、空白マス数基準と同様、解探索に成功するか、基準を満たすような数字変更が行えなくなるまで、各初期配置のマスとその候補数字についてこの手順を繰り返す。アルゴリズム 6 にこの手順を示す。

Algorithm 6 . 候補数字基準に基づいた変更手順

```
cand ← 9 * 9 * 9
while 盤面が変化 do
  for cell ∈ 初期配置マス群 do
    origin ← cell の元々の数字
    cell を空白にする
    for num ∈ cell の候補数字群 −origin do
      cell に num を配置
      解探索実行
      if 唯一解発見 then
        return 生成した盤面
      end if
      盤面を解探索前の状況に復元
      temp ← 盤面の空白マスの持つ候補数字数の 2 乗和
      if temp < cand then
        cand ← temp
        origin ← num
      end if
    end for
    cell に origin を配置
  end for
end while
```

3. 高難易度問題の生成手法

数独の問題集は掲載されている問題の難易度別で発刊されているものや、難易度別に問題を掲載しているものが多く存在する。問題集の制作を支援するためには、生成する問題の難易度を制御できることが望ましい。自動生成する問題の難易度を制御する手法としては、何らかの基準に基づいて難易度を数値化して生成した問題にフィルターをかけ、それを通過した問題のみを出力するという手法が存在する。だが、本研究では指定された難易度の問題を自動生成する手法の実現を目標として、より高難易度な問題を自動生成する手法を考案した。

3.1 数独の難易度

数独の難易度の基準には、様々なものが存在する。例えば、成美堂出版の問題集 [11] では初期配置の数を基準として、初期配置が少ないほど難しい問題とし、初級編、初中級編、中級編、上級編の 4 つの難易度に分けて問題を掲載している。この基準は初期配置が少ないほど回答者が埋めなければならないマスの数が多く、その分だけ回答が困難になるという考えに基づくものである。しかし、初期配置数が少ない場合でもごく単純な解法のみで解ける問題も存在し、難易度の基準としては不十分ではないかという指摘がある [12]。

この点を考慮した基準として、解探索の手順や要した解法に着目した基準がある。松原ら [13] は、解探索時の推論を規則化し、解探索に要した推論規則から問題の難易度を推定している。また土出ら [12] は、解探索時の盤面の探索回数を基にした難易度基準を提案している。初期配置数に基づく基準は問題を解く前の状況から難易度を推定するのに対し、解法に着目した基準は問題を解き進めるプロセスを基に難易度を推定するため、より回答者の実感に即した難易度を算出することができるのではないかと考えられる。本研究では、解探索に浜田ロジックなどの比較的高度とされる解法を用いていることも踏まえ、解法に着目して高難易度問題の生成を行う。

3.2 既存の高難易度問題の自動生成手法

生成する問題の難易度を制御する手法としては、生成した問題に難易度でフィルターをかけ、望む難易度の問題のみを出力するという手法がある。この手法と関連のある手法として、石田 [2] は数独の問題としての面白さを評価する基準を定義し、回答者のレベルに合わせたフィルタリングを行うことで回答者が楽しめる問題を生成する手法を提案している。また、この手法を取り入れた問題生成ツールに、Number Place Generator Version2.0.2[6] が存在する。このツールでは、生成した問題の難易度を独自の基準によって数値化しており、問題生成時のオプションとして生

成する問題の難易度の範囲を指定することが出来る。

これらの手法やツールを用いれば望む難易度の問題を生成することができる。しかし、これらはあくまでも生成した問題に対してフィルターをかけているのであり、望む難易度の問題のみを生成している訳ではないため、フィルターを通過しなかった問題の生成時間が無駄になってしまう。そこで、本研究では問題の難易度を考慮した自動生成手法へのアプローチとして、より高難易度の問題が生成されやすい自動生成手法を提案する。

3.3 本研究での高難易度問題の自動生成手法

本研究の提案手法では配置した数字を変更して解探索を行い、唯一解が得られなければ、2 通りの基準に従って変更を採用する。2 通りの基準は前節で述べた通り、解探索の手順や用いる解法については考慮していない。そのため、基準に従った数字変更の結果、ごく単純な解法しか要しない配置となってしまう可能性がある。そこで、そのような可能性を低くするため、変更前と同じかそれ以上に高度な解法を用いていた場合にのみ変更を採用する、という条件を 2 通りの変更基準に付加する。これにより、たとえ基準を満たす数字変更だったとしても、単純な解法のみで解ける配置になってしまうものであれば、その変更の採用を防ぐことができるので、より高度な解法を要する問題を生成しやすくなるのではないかと考えられる。

具体的な手順は次の通りである。まず、数字変更を行い解探索をした際に、解探索に用いた解法の中で最も高度なものを記録しておく。そして、変更前の盤面の解探索に用いた最も高度な解法と比較し、変更後の解法が変更前のものと同じ解法であるか、より高度な解法であったならば、解探索に成功していた場合は盤面を問題として出力し、失敗していれば変更基準を満たしているかどうかの判定に移る。もし、変更前の方が高度な解法であった場合には、解探索の正否や変更基準を満たしているかどうかに関わらず盤面を変更前に戻し、数字変更を続行する。

なお、解法の高度さについては、Singles がもっとも簡単な解法であるとして、Intersections、NakedSubsets および HiddenSubsets、BasicFish、浜田ロジックの順でより高度な解法であるものとし、NakedSubsets と HiddenSubsets は同等の解法であるとした。例えば、変更前の盤面への解探索に Singles、Intersections、NakedSubsets を用いていた場合、数字変更を採用するには解探索に NakedSubsets、HiddenSubsets、BasicFish、浜田ロジックのいずれかを 1 度でも適用していなければならない。

4. 実験

この章では提案した手法の問題生成能力を測るために行った実験について述べる。実験環境の CPU は Intel®Xeon®Processor E3-1240 v3、OS は Windows 7、メモリは 8GB である。本研究の手法を実装した数独の問題生成プログラムを下記 URL で公開する。http://www.cs.ise.shibaura-it.ac.jp/2017-GI-37/

4.1 数字配置手法の比較実験

本研究で実装した 2 通りの数字配置手法と、先行研究 [8] で取り入れられた解盤面生成方式の手法について、実行時間や問題生成の成功率を比較するために実験を行った。

実験の詳細を以下に述べる。各手法に対して、特定の初期配置マスのパターンを入力として与えて、10 分間問題を生成し続け、生成できた問題数と生成の試行回数を比較した。ただし、先行研究 [8] の手法は解盤面の生成に失敗する確率が非常に高かったため、候補数字の少ないマスから優先して数字を配置するという改良を行い、数字配置の成功率を向上させた。

実験の結果、各手法で生成できた問題数と生成の試行回数を表 1 に、問題生成にかかった平均時間を表 2 にまとめる。表 1 より、解盤面生成方式による手法は、初期配置数が 23 個以下のパターンの問題を生成することができなかった。それに対し、本研究のランダム決定手法は、18 個以外のパターンにおいて解盤面生成方式による手法を上回る問題生成数となった。絞り込み手法は、初期配置数の少ないパターンにおいてランダム決定手法を上回る問題生成数となった。よって、本研究の 2 通りの数字決定手法は、解盤面生成方式による手法よりも有用であると分かった。また、表 2 より、問題生成の平均所要時間は問題生成数が多くなればなるほど短くなった。

本研究で取り入れた 2 通りの決定手法を比較すると、ランダム決定手法については、試行回数が非常に多く確保できる手法であり、初期配置が 24 個以上の配置の場合には絞り込み手法よりも多くの問題を生成できることが分かった。それに対して、絞り込み手法は試行回数が解盤面生成方式による手法よりも少なくなるものの、初期配置が 23 個以下ならばランダム決定手法よりも多くの問題を生成できることが分かった。よって、数字変更を考慮しない場合、初期配置数が 24 個以上ならばランダム決定手法が、23 個以下ならば絞り込み手法がより有用な手法であると言える。

4.2 問題生成能力の測定実験

本研究の提案手法が、様々な初期配置の指定に対して問題生成が可能であり、かつ、1 つの入力パターンに対して多くの問題が生成可能であることを示すために実験を行った。また、本研究の提案手法では、2 通りの数字配置手法と数字変更基準を組み合わせることで、4 通りの生成手法が得られるため、もっとも優れた手法がどれであるかを調査する。

実験の詳細を以下に述べる。各手法に対して、特定の初期配置マスのパターンを入力として与えて、30 分間問題を

表 1 各数字配置手法の 10 分間の問題生成数と試行回数

初期配置数	解盤面生成方式	ランダム決定手法	絞り込み手法
18	0/1020943	0/1743784	0/193514
19	0/1002921	1/1665775	0/184154
20	0/987930	30/1531432	899/164697
21	0/968809	149/1478804	1528/156920
22	0/926955	919/1354832	3525/165447
23	3/913464	3751/1435992	3758/156931
24	44/843411	10226/1794011	183/148079
25	219/848040	18536/2130673	208/139800
26	543/842844	9091/3705177	0/128753
27	1573/878500	13756/3950493	0/98890
28	8942/896961	14976/4742787	0/114857

表 2 各数字配置手法の 10 分間の問題生成の平均所要時間 (単位：ミリ秒)

初期配置数	解盤面生成方式	ランダム決定手法	絞り込み手法
18	-	-	-
19	-	572196	-
20	-	19684	666
21	-	4014	392
22	-	651	169
23	175489	159	159
24	13296	58	3268
25	2713	31	2881
26	1099	65	-
27	380	43	-
28	66	39	-

生成し続け、生成できた問題数と生成の試行回数を比較した。試行回数については、数字配置を開始してから、解探索に成功するまで、もしくは基準を満たすような数字変更が行えないことが判明するまでを 1 回の試行とした。

各手法で生成できた問題数と生成の試行回数を表 3 と表 4 に、問題生成にかかった平均時間を表 5 にまとめる。表 3、表 4 を表 1 と比較すると、数字変更を行うことで生成の成功率が上がっていることが分かった。特に、数字配置のみではほぼ問題を生成できなかった初期配置数が 19 個以下のパターンについて、数は少ないものの問題生成に成功している。また、表 5、表 4 を表 2 と比較すると、数字変更の導入によって問題生成が高速化したと分かった。

数字決定手法と変更基準の組み合わせごとに比較すると、全ての入力パターンにおいて、絞り込み手法+空白マス数基準の組み合わせが、問題生成数が最大となり、平均所要時間も最短となることが分かった。この組み合わせは、初期配置数が 20 個以上ならば 10000 問以上の問題を生成でき、初期配置数が 24 個を下回ると生成が困難になる前田らの手法 [4] と比較しても、有用な手法であると言える。ただし、初期配置数が 19 個以下のパターンに関しては、20 個以上のパターンに比べるとかなり少なく、未だに充分な生成能力を持っているとは言えない。

表 3 ランダム決定手法と数字変更を組み合わせた場合の問題生成数と試行回数

初期配置数	ランダム手法+空白マス数基準	ランダム手法+候補数字基準
18	4/17377	0/7818
19	62/18332	61/7359
20	743/20828	3867/7389
21	2168/24278	7116/8995
22	5391/33292	15369/16765
23	15386/45230	27136/27998
24	36770/77367	38548/49642

表 4 絞り込み手法と数字変更を組み合わせた場合の問題生成数と試行回数

初期配置数	絞り込み手法+空白マス数基準	絞り込み手法+候補数字基準
18	48/21305	0/18220
19	267/23220	116/19967
20	12164/32322	9741/23039
21	25772/44437	19902/28572
22	55063/87519	51216/74568
23	101059/125014	77692/104480
24	87591/138017	28376/109512

表 5 ランダム決定手法と数字変更を組み合わせた場合の問題生成の平均所要時間 (単位：ミリ秒)

初期配置数	ランダム手法+空白マス数基準	ランダム手法+候補数字基準
18	430563	-
19	28818	28766
20	2417	464
21	829	252
22	333	116
23	116	65
24	48	46

表 6 絞り込み手法と数字変更を組み合わせた場合の問題生成の平均所要時間 (単位：ミリ秒)

初期配置数	絞り込み手法+空白マス数基準	絞り込み手法+候補数字基準
18	37242	-
19	6712	15503
20	147	184
21	69	89
22	32	34
23	17	22
24	20	62

4.3 難易度の比較実験

本研究で提案した高難易度問題の自動生成手法によって生成される問題が、実際に高難易度であるかを確かめるため、実験を行った。本研究の高難易度問題の生成手法は、2通りの数字変更基準により高度な解法を用いるように条件を付加するため、数字配置手法と数字変更基準の組み合わせにより 4 通りの手法が得られる。これらに加え、既存の問題生成ツール Number Place Generator Version2.0.2[6] で生成される問題の難易度を測定し、比較した。問題の難易度の基準には、Number Place Generator Version2.0.2[6] で測定した難易度スコアを用いた。

実験の詳細を以下に述べる。各手法に対して、初期配置数 24 個のパターン 1 つを入力として与えて 10 問の問題を生成し、Number Place Generator Version2.0.2[6] で測定した難易度の平均値と標準偏差、および問題生成の平均所要時間を比較した。なお、Number Place Generator Version2.0.2[6] は Singles、Intersections、NakedSubsets、HiddenSubsets、BasicFish を用いて解探索を行うが、NakedSubsets の内の NakedQuadruple、HiddenSubsets の内の HiddenQuadruple、BasicFish の内の JellyFish、浜田ロジックを実装していないため、この実験においては本研究の手法もそれらを解探索に用いないこととする。

各手法で生成した 10 問の問題の難易度の平均値と標準偏差、問題生成の平均所要時間を表 7 にまとめる。表 7 より、ランダム決定手法と候補数字基準の組み合わせに高難易度化条件を付加した手法が、難易度の平均値が最も高くなると分かった。この組み合わせは Number Place Generator Version2.0.2[6] と比較しても、平均値が高く、標準偏差が小さいため、高難易度の問題を生成しやすい手法になっていると言える。ただし、平均所要時間に関しては Number Place Generator Version2.0.2[6] の 10 倍にもなっており、問題生成にかかる時間が長くなってしまうことが分かった。したがって、高難易度の問題を大量に生成するためには、さらなる高速化が課題となると言える。

表 7 各手法で生成した問題の平均難易度と標準偏差および生成の平均所要時間

手法	難易度平均	標準偏差	平均所要時間 (ミリ秒)
ランダム手法+空白マス数基準	6518	2239	267
ランダム手法+候補数字基準	8298	2730	533
絞り込み手法+空白マス数基準	4821	2666	30
絞り込み手法+候補数字基準	6264	4367	1724
Number Place Generator[6]	6606	3741	53

5. 考察

この章では、実験結果を元に、本研究の提案手法の問題生成能力について述べる。実験によって、本研究の提案手法が大量の問題を高速に生成できることを示したが、提案手法に導入した数字決定手法や数字変更基準がどのように問題生成能力へ影響を与えたのかを、実験の結果から考察する。

5.1 数字配置手法

本研究の提案手法に導入した数字配置手法は、先行研究 [8] に導入された解盤面生成方式による手法に比べて、問題生成能力を向上できていることを実験によって示した。解盤面生成方式による手法とランダム決定手法とを比較すると、ランダム決定手法の試行回数は解盤面生成方式の約 1.5 倍となっており、試行回数の大幅な増加が問題生成数の増加に繋がったと考えられる。また、絞り込み手法と比較すると、試行回数は解盤面生成方式よりも少ないものの問題生成数は上回っており、候補数字の絞り込みによって複数解を持つ盤面生成を防ぐことが、試行回数の減少以上に大きな影響を与えていると考えられる。

また、初期配置数によってランダム決定手法と絞り込み決定手法の問題生成数に差が生じていたが、これは初期配置が少ないほど問題として成立している可能性が低くなる、または解探索で実装している解法だけでは解を求められる可能性が低くなるために、試行回数を確保するよりも候補数字の絞り込みによって複数解を持つ配置や解を求められない配置を防ぐことの方が有効となったためだと考えられる。逆に初期配置数が多い場合は自然と候補数字数が少なくなるために、試行回数が多いランダム決定手法のほうが有効となったものと考えられる。

5.2 数字変更基準

数字変更の導入によって、数字配置のみよりも数字配置の回数が減少したものの、問題生成数が増加し、問題生成の平均所要時間が短縮したことを実験によって示した。数字変更の導入は、ある盤面から別の盤面への変更を数字配置をし直すよりも短い処理で行うことができるようになるため、これが問題生成の効率化に寄与し、素早く問題を生成できたと考えられる。また、生成できた問題数が増加したということから、変更時に着目した解探索後の空白マス数や盤面の候補数字数は、解探索に成功する盤面を生成するための指標として有用なものであると考えられる。

2 つの変更基準を比較すると、数字配置をランダム決定手法で行った場合、空白マス数基準は候補数字基準よりも問題生成数が少なくなった。これは、空白マス数基準に基づく数字変更が手詰まりに陥る可能性があるためだと考えられる。問題が解無しや複数解であったとしても、空白マ

スが残り数個になるまで解探索を進めることができる場合がある。そのような場合、空白マス数が既に少ないため、さらに空白を減らすような変更は難しくなるが、数字を一つ変えただけでは解の求まる盤面になるとは限らないため、手詰まりになってしまう。

それに対し、候補数字基準では候補数字数の 2 乗和を取ることで、候補数字数の多いマスが優先して絞り込まれやすくなり、解無しのマスが生じにくくなっている。さらに候補数字が絞り込まれるために複数解にもなりにくくなっており、結果として空白マス数基準を用いた場合のような手詰まりを防ぐことができたものと考えられる。

数字配置を絞り込み手法で行った場合、空白マス数基準との組み合わせが他の組み合わせよりも多くの問題を生成できた。これは、絞り込み手法によって得られた盤面は既に候補数字が絞り込まれているために解無しや複数解となる可能性が低く、空白マス数基準による変更をしても手詰まりに陥りにくくなり、欠点を緩和しつつ高い生成成功率と試行回数を確保できたためだと考えられる。逆に候補数字基準では変更の余地が少なく、数字変更がうまく機能しなくなったものと考えられる。

5.3 高難易度問題の生成

本研究の手法は、高度な解法を用いるような問題が難易度の高い問題であると想定し、より高度な解法を用いるように数字変更を行った。実際に高難易度の問題生成に成功していることから、解探索に用いる解法を制御することは問題の難易度制御に対して有効に働くものと言える。

さらに、ランダム決定手法と候補数字基準の組み合わせに高難易度条件を付加した手法が、他の手法よりも高難易度の問題を生成しやすくなっていることを実験によって示した。4.1 節で示したように、数字配置を絞り込み手法で行うことで問題生成の成功率を高めることができるが数字変更の余地が少なくなり、結果として高度な解法を用いる問題を生成しにくくなる。また、数字変更に空白マス数基準を採用した場合、Singles を多く適用でき、空白マス数が少なくそのような高度な解法を適用する余地のない盤面になりやすいため、空白マス数基準による数字変更と解法の高度化条件とは相性が良くないものと言える。したがって、ランダム決定手法による数字配置と候補数字基準による数字変更を組み合わせた手法は平均難易度が最も高くなり、逆に、絞り込み手法による数字配置と空白マス数基準による数字変更を組み合わせた手法は平均難易度が最も低くなったと考えられる。

本研究の手法の改善点としては、まず、数字配置に関する点がある。現状では数字配置の時点で問題生成に成功した場合、さらなる高難易度化を試みることはしない。よって、何らかの条件を付加するなどして、高難易度な問題になりやすい数字配置手法を考案することが挙げられる。

また、解探索に用いた解法以外にも難易度の指標を取り入れることが挙げられる。簡単な解法しか使わない問題が必ずしも簡単に解けるとは限らず、例えば、1 つの数字を決定するまでに何度も解法の適用によって候補数字を絞り込まねばならないような問題の場合、解法が単純なものでも回答者が難しく感じる可能性がある。現状では高度な解法を要さないが難易度の高い問題の生成は考慮していないため、解法の高度さ以外の指標を取り入れることで生成できる問題の幅が広がるものと考えられる。

6. 関連研究

本章では、推論規則と難易度判定に関する研究 [13]、問題生成の支援システムに関する研究 [4]、少数ヒント問題の自動生成に関する研究 [5]、数独の面白さの評価尺度に関する研究 [2]、解盤面の列挙と番号付けに関する研究 [3] の 5 つについて本研究との関連について議論する。

6.1 数独の推論規則と難易度判定に関する研究

松原らの推論規則と難易度の判定に関する研究 [13] は、数字を確定する為に用いた推論の数と、それぞれの推論の適用の困難さを難易度判定の基準とする事を提案した。そして、推論の方法を盤面の状況毎に規則化した 8 パターンの推論規則を提案した。松原らは、それらの推論規則によって数独の難易度を測定し、問題制作者が望むレベルの難易度の問題を自動生成することの実現につながるのではないかと推測している。

本研究で解探索に用いた Singles、Intersections、NakedSubsets、HiddenSubsets は、松原らの提案した推論規則と同様の解法であり、高難易度問題の自動生成手法で解法の高度さの決定には松原らの推論規則の考え方を参考にした。また、本研究では松原らが現状の推論規則の枠組みでは捉えられない解法の例として挙げている BasicFish(X-wing) と浜田ロジックを実装し、より幅広い問題生成を可能にした。

6.2 数独の問題作成支援システムの設計と開発

数独の問題作成システムを扱った研究として、前田らの研究 [4] がある。前田らは、数独の難易度と問題構造との間にある関係を解明することを課題として、制作中の問題についての様々な情報を提供して問題制作者の支援を行うシステムを開発した。システムから提供される情報は、各マスが持つ候補数字や、初期配置の数字が盤面に及ぼしている影響、生成する問題の推定難易度などであり、システムの利用者はこれらの情報を参考にしながら、盤面に数字を配置していくことで問題を作成できる。また、制作者支援のアプローチとして、前田らは候補数字の死活という概念に着目している。前田らは各マスの持つ候補数字を、その数字を入れる事によって問題が解無しとなってしまう Dead

候補と、その数字が配置された解が存在し得る Active 候補の 2 種類に分類し、全ての空白マスの Active 候補をただ 1 つに定めることを問題制作の目標であると定義した。そして、問題制作者が Dead 候補をマスに配置することを防ぐため、候補数字が Dead 候補と Active 候補のどちらであるかを表示する機能と、マスに数字を配置した際に、他のマスの Dead 候補と Active 候補がどのように変化するのかを確認できる機能をシステムに実装した。

さらに、候補数字の死活判定を応用し、初期配置の位置を指定可能な自動問題生成手法を考案している。これは指定された初期配置マスに対して、各空白マスの Active 候補が出来る限り少なくなる数字配置を行うという手法であり、全ての空白マスの Active 候補が 1 つに定まれば生成成功とする。この手法は、初期配置の位置を指定できるため、視覚的なデザイン性を考慮して問題生成を行う事が可能だが、初期配置数が 24 個以下の問題に関しては生成成功率が著しく低くなり、問題がほとんど生成できなくなってしまうという問題点を挙げている。

本研究でも数字配置の手法として候補数字に着目した絞り込み手法を採用しているが、絞り込み手法では候補数字の死活に関わらず個数のみを参照しており、また、個数の値を 2 乗することで候補数字の多いマスが優先して絞り込まれるような工夫を行っている。

6.3 数独の少数ヒント問題の生成に関する研究

プログラムによる自動問題生成に関する研究として、那須らの少数ヒント問題に関する研究 [5] がある。那須らの研究 [5] では、プログラムによる自動問題生成で初期配置が少ない問題を生成する事を目標として、2 種類の問題生成プログラムを開発し、実験を実施している。

那須らは候補数字の合計数に着目する数字配置アルゴリズムを導入した問題生成プログラムを開発した。このプログラムは、盤面の候補数字数の合計ができる限り小さくなるようにマスと数字を決定し、数字配置を行うというものである。このプログラムに対しても同様の実験を行った結果、初期配置が 20 個の問題については 61 問、19 個の問題については 32 問、18 個の問題については 7 問の生成に成功しており、この結果から少数ヒント問題の生成と候補数字の合計数の関連を示唆している。

那須らのプログラムと本研究の数字配置手法の違いとしては、初期配置の位置の指定の可否が挙げられる。那須らは初期配置の位置の決定も含めた少数ヒント問題の生成手法を提案している。また、那須らは解法による解探索に行き詰った場合の処理として、数字の仮置きによる深さ 1 のバックトラックを許容している。バックトラックを駆使した解探索は、高速な計算が可能であるコンピュータによる解探索手法としては非常に有効な手法であると言える。だが、この手法はいわゆる総当たり的な手法であり、また、人間が仮定の下に長く解探索を進めることは困難であると考えられる。本研究の手法で生成する問題は人間が解くことを前提としているため、解探索には人間が実際に利用できる解法を実装した手法を採用し、バックトラックによる解探索は行っていない。

6.4 ゲームにおける問題の評価および作成に関する研究

石田の研究 [2] では、問題を解くのに用いる解法の複雑さと解法による候補数字の絞り込みの回数に着目し、問題の面白さを測る尺度を考案している。石田は数独問題の自動生成について、人間が制作した問題に比べると面白さが劣る傾向にあると指摘し、問題の面白さを測る尺度を用いて、生成した問題を面白い問題と面白くない問題とを分別することで、回答者が面白いと思える問題を自動生成することができるようになるのではないかと提案している。

石田は数独を解く手順を、候補の絞り込みと数字の確定という 2 つの作業の繰り返しとみなし、各作業の複雑さと作業の繰り返し回数に着目した面白さの尺度を提案した。この尺度では、解法を複雑さでランク分けし、問題を解く上で要した解法の中でもっとも複雑なもののランクを作業の複雑さとし、盤面に解法を適用して候補を絞り込んだ回数を作業の回数としている。自動生成した問題の中からこの尺度で面白いと判断された問題とそうでない問題を選び出し、被験者にそれらの問題を解いてもらったところ、提案した尺度で面白いと判断された問題の方が人間も面白いと感じていたという結果を報告している。

本研究の提案手法では問題生成に際して面白さを考慮してはいないが、石田が着目した解法の複雑さと絞り込み回数という 2 点は、高難易度問題の自動生成手法における難易度の尺度を考える上でも重要な要素になるのではないかと考えられる。例えば、本研究の高難易度問題の自動生成手法は解探索に使用した解法の中でも最も複雑な解法に着目しているが、解探索に要した解法適用の回数も考慮に入れることで、簡単な解法のみで解ける高難易度問題を生成することができるのではないかと考えられる。

6.5 本質的に異なる数独解盤面の列挙と番号付け

井上ら [3] は数独の解盤面について、全ての本質的に同じ解盤面の集合からそれぞれ代表となる盤面を一つ選び出してできた集合を本質的に異なる解盤面の集合とした。

井上らは、ある解盤面に対して、数字を入れ替えた盤面、回転させて得られる盤面、同じブロックに属する行や列を入れ替えて得られる盤面、同じブロックに属する 3 つの行や列を入れ替えて得られる盤面を、元々の解盤面と本質的に同じ盤面であるとした。本質的に同じ解盤面は、その盤面を解に持つ問題の数が等しく、また、問題として成立しうる最小の初期配置数も等しいという性質を持つ。そのため、初期配置数が最小の問題を考えるような際には、本質

的に同じ解盤面の集合はその中の 1 つの盤面のみについて考慮するだけでよくなる。井上らの研究 [3] では、全ての解盤面の中から本質的に同じ解盤面を除くことで探索領域を絞り込み、本質的に異なっている盤面を高速に列挙する手法を提案している。さらに、その手法を利用して、列挙した本質的に異なる解盤面に対して通し番号を付けた。

本研究の提案手法では、問題生成の際にあらかじめ解盤面を生成することはしない。しかし、井上らの提案した解盤面の高速な列挙手法を応用することで、本研究の提案手法をより効率化できる可能性がある。例えば、ある盤面の初期配置が、いずれの解盤面とも一致しないことを照合できれば、数字配置の途中や解探索を行う前に解無しの盤面かどうかを判定することができるかもしれない。その際に、井上らの手法によって解盤面集合を高速に探索できれば、解の無い盤面に解探索を行う無駄を省くことができるものと考えられる。

7. まとめと今後の課題

本研究では、数独の問題集や数独の問題を掲載するパズル雑誌の制作者を支援を目的として、初期配置が指定可能であり、かつ大量の問題を高速に生成可能な問題生成手法を提案した。本研究の提案手法では、数字配置、解探索、数字変更の 3 段階によって問題生成を行い、その際に 2 種類の数字配置手法と数字変更基準を組み合わせて問題を生成し、人間が実際に用いる解法を導入した解探索アルゴリズムに従って唯一解を探索する。本研究の提案手法の問題生成能力を測定する実験を行った結果、初期配置が 20 個以上という条件下ならば大量の問題を高速に生成できることを示した。また、候補数字を絞り込む数字配置手法と解探索後の空白マス数を減らす数字変更基準を組み合わせることで、他の組み合わせよりも問題生成能力が高まることを示した。さらに、解探索に用いる解法の難易度に着目し、より高度な解法を用いるように数字変更を行うことで、より高難易度な問題を自動生成する手法を提案した。比較実験により、この手法は既存の問題生成ツールよりも高難易度の問題を生成しやすくなっていることを示した。

本研究の提案手法の今後の課題としては次のような点が挙げられる。

解探索の改良

本研究の手法では、解探索に人間が実際に利用可能である解法を導入しており、導入した解法には BasicFish や浜田ロジックなどの比較的高度とされる解法も含まれているが、その他にも様々な状況で適用可能な解法は存在する。例えば、X-Chain や XY-Chain[9] といった解法は浜田ロジックと同様に連鎖関係に着目する解法であるが、浜田ロジックよりも幅広い状況で適用可能であり、これらを解探索に導入することで、唯一解を求められる問題の幅が広がると考えられる。

初期配置数の少ない問題の生成

本研究の手法では、入力されたパターンの初期配置数が 19 個以下の場合、生成される問題数が 20 個以上の場合と比べて大きく減ってしまった。9 × 9 サイズの数独問題の最小初期配置数は McGuire によって 17 個であると証明されている [14]。よって、将来的には初期配置数が 17 個の問題を大量に生成できるような手法を考案することが考えられる。

取り扱う数独の種類の拡張

本研究で扱う数独の問題は 9 × 9 サイズのものに限定していたが、その他にも扱う数字を 1 から 16 までに増やした 16 × 16 サイズの問題などが存在する。16 × 16 サイズの数独問題については、2011 年の GPCC で少数ヒント盤面を求める問題が出題される [15] など、今後さらなる研究が行われることが期待されており、本研究の提案手法においても生成可能な問題の条件を広げていくことが必要であると考えられる。

高難易度問題の自動生成手法の改善

本研究では高難易度問題を自動生成する手法を提案したが、その際には解探索に高度な解法を要する問題が難易度の高い問題であると想定していた。しかし、簡単な解法のみで解くことが可能な高難易度問題も存在し、現状の手法ではそのような問題が生成される可能性は低く、解法の難易度のみを問題の難易度の判断基準とするのは不十分であると言える。よって、解法の難易度以外に問題の難易度を判断する基準を設けるなどして、生成できる高難易度問題の幅を広げていくことが望まれる。

難易度を指定可能な問題生成手法

本研究で提案した高難易度問題の自動生成手法を用いることでより難しいと思われる問題の生成は可能だが、細かい難易度の制御をすることはできない。特定の難易度の問題を生成したい場合、現状では、NumberPlaceGenerator[6] のように、難易度でフィルターをかけ、問題を生成してから選別する手法を用いる他ない。よって、事前に指定された難易度の問題を自動生成する手法の考案は今後の課題として考えられる。

参考文献

[1] 井上秀太郎，佐藤洋祐：グレブナー基底を使った数独の難易度判定と問題作成 (数式処理：その研究と目指すもの)，数理解析研究所講究録， Vol. 1785, pp. 51–56（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110009322360/⟩ (2012).

[2] 石田伸輔：ゲームにおける問題の評価および作成に関する研究，修士論文，三重大学大学院工学研究科博士前期課程電気電子工学専攻 (2007).

[3] 井上真大， 奥乃博：本質的に異なる数独解盤面の列挙と番号付け，第 71 回情報処理学会全国大会講演論文集，pp. 741–742（オンライン），入手先

⟨http://ci.nii.ac.jp/naid/110007505881/⟩ (2009).

[4] 前田一貴，　奥乃博：数独の問題作成支援システムの設計と開発，第 70 回情報処理学会全国大会講演論文集，pp. 799–800（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110006867930/⟩ (2008).

[5] 那須律政：数独の少数ヒント問題の生成に関する研究，高知工科大学情報システム工学科 2012 年度卒業論文.

[6] 株式会社タイムインターメディア：Puzzle Generater JaPan, https://web.archive.org/web/20150505004416/http://www.puzzle.gr.jp/show.

[7] とん：ナンプレの自動生成，岩波データサイエンス Vol.2，岩波書店，pp. 101–109 (2016).

[8] 奥原克彦：初期配置を考慮した数独の問題生成手法に関する研究，卒業論文概要集第 35 号，芝浦工業大学 情報工学科，pp. 91–92 (2014).

[9] Hobiger, B.: HoDoKu: Solving Techniques, http://hodoku.sourceforge.net/en/techniques.php.

[10] 西尾徹也：ナンプレ超上級編 21，世界文化社 (2009).

[11] 川崎光徳：ナンプレ DX200 初級→上級 1，成美堂出版 (2012).

[12] 土出智也，真貝寿明：数独パズルの難易度判定：解法ロジックを用いた数値化の提案，大阪工業大学紀要. 理工篇， Vol. 56, No. 1, pp. 1–18（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/40019056200/⟩ (2011).

[13] 松原康夫：数独の推論規則と難易度に関する考察，情報処理学会研究報告エンタテインメントコンピューティング（EC），Vol. 2006, No. 134, pp. 1–6（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110006164288/⟩ (2006).

[14] McGuire, G., Tugemann, B. and Civario, G.: There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem via Hitting Set Enumeration, *Experimental Mathematics*, Vol. 23, No. 2, pp. 190–217 (online), DOI: 10.1080/10586458.2013.870056 (2014).

[15] 藤波順久，酒井香代子：GPCC 報告 (2011 年) Games and Puzzles Competitions on Computers，第 53 回情報処理学会プログラミング・シンポジウム予稿集，pp. 53–56 (2012).