Vol.2016-GI-35 No.1
2016/3/8

初期配置が指定された場合に適した数独問題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、初期配置が図や模様を描くように配置されているものがある。そのような視覚的なデザインを考慮した初期配置を持つ問題を生成する際には、既存の推論規則を用いた素朴な問題生成手法は必ずしも有効ではない。そこで本研究では、初期配置を問題製作者に指定させ、それに従って問題生成を行う生成手法を提案し、実装する。この手法では、指定されたマスに数字を置く際に、解探索によって初期配置以外の各マスで絞り込まれる数字の候補が少なくなるように数字を置くことで、唯一解を持つ問題生成の成功率を高める。また、生成した問題が人間が解くのに適した問題になっているかどうかの判定を行うために、人間が通常用いる解法を反映した解探索を行う。解探索には Basic Fish や浜田ロジックといった高度な解法も使用し、難易度が高い問題生成も可能にする。また、解探索時にバックトラックを 1 段行う処理を導入した。

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。また、近年では携帯電話向けのアプリとしても人気である。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。そのため、難易度設定とデザインを両立した良質な問題の生成は問題作家の手に委ねられている。

			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

先行研究 [1] では、問題生成者の支援を目的に、あらかじめ初期配置の位置を指定しつつ問題を自動生成するプログラムの開発に着手した。このプログラムにより、問題製作者は問題生成を自動化しながら、初期配置の位置の指定によって視覚的なデザインを考慮する事が可能となる。しかし、現状では生成できる問題のパターンが少なく、特に初期配置の数が少ない少数ヒント問題の生成が困難である。また、問題生成にかかる時間も長い。そこで本研究では、問題制作者へのさらなる支援を行うため、より初期配置の少ない問題の生成および問題生成の高速化を目標として、新たな問題生成方式を考案し、実装する。

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、数字配置と解探索の 2 段階によって問題を生成する。まず、数字配置によって、盤面の初期配置となる数字を決定する。そして、得られた初期配置が数独の問題として適切かどうかを解探索によって判別する。

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

2.1 問題生成手法

本研究で提案する手法では、まず利用者に盤面から初期配置の位置を指定してもらう。そして、初期配置として利用者が指定したマスに対して、数独の制約に従いながらランダムに数字配置を行う。指定されたマスの全てに数字を配置し終えると、その配置が唯一解を持ち、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、得られた配置が唯一解を持ち、数独の問題として成立していたならば、問題生成を成功とし、得られた配置を問題として出力する。そして、もし得られた配置が唯一解を持たず、数独の問題として不成立であったならば、再び指定されたマスへの数字配置をやり直し、問題として成立している配置が得られるまでこれを繰り返す、という手順によって問題生成を行う。得られた配置が問題として成立しているかどうかを確かめる解探索には、松原氏が提示した 8 個の推論規則 [2]、および Basic Fish[3]、浜田ロジック [4] という高度な推論規則を用いる。これらの解法は人間が数独の問題を解く際に実際に用いる解法であるので、この解探索手法により、人間が解を求めることができる問題が生成されることが保証される。また、これらの解法で解を求められなかった場合には、数字の仮置きを行って解探索を進めるようにした。

2.2 数字配置手法

数字配置を行う際に、先行研究 [1] では盤面の全てのマスのそれぞれに対し、その時点で数独の制約を満たしている数字のいずれかをランダムで配置するという手法を採用していた。この手法では数字配置を終えた後、初期配置として指定されたマスを除いた全てのマスを空白にし、得られた盤面に対して解探索を行っていた。しかし、この手法は数字配置の成功率が低く、一度の問題生成に対して数字配置のやり直しを頻繁に行うため、問題生成に長い時間がかかる原因となっていた。よって、本研究では数字配置を行うマスを初期配置として指定されたマスのみに限定し、新たに候補数字の合計数に着目したアルゴリズムを導入した。

候補数字とは、空白のマスに入る可能性のある数字である。ある数独の盤面の一部を切り出した**図 2** を例に取ると、X のマスは同じ領域内に 1 から 5 までの数字が配置されている。数独の制約により、同じ領域内には同じ数字が 2 つ以上入る事は許されないので、この時点で X のマスには 6 から 9 までの数字のいずれかのみが入る。この時、X のマスの候補数字は 6、7、8、9 の 4 つとなる。

本研究で導入する数字配置手法をアルゴリズム 1 に示す。このアルゴリズムでは、空白の初期配置マスに数字を仮置きし、節 2.3.1 に示す候補数字の絞り込み処理を行ったあと、盤面の候補数字の合計数を計算する。これを空白の初期配置マス全てに対して行い、最終的に合計数が最も小さな値となった配置を実際に行う。配置の候補が複数ある場合は、それらの中からランダムに 1 つ選び、配置を行う。これを全ての初期配置マスに数字を配置するまで続け、問題を生成する。なお、候補数字が 0 個のマスが生じる配置しかできなくなった場合は、配置を初めからやり直すこととする。

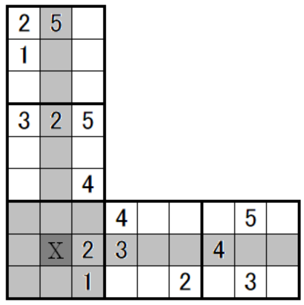

図 2 数独の問題の一部

Algorithm 1 数字配置アルゴリズム

```
while 空白の初期配置マスが存在する do
    min ← 現在の候補数字の合計個数
    マスと置く数字の組の記録を空にする
    盤面の状況を一時保存

    while 候補数字の計算をしていない初期配置マスがある do
        初期配置マスの 1 つに候補数字を仮置きして絞り込み処理
        if 候補数字 0 個になったマスが無い and 現在の候補数字の
        合計個数 <= min then
            if 仮置き後の候補数字の合計個数 < min then
                マスと置く数字の組の記録を空にする
                min ← 仮置き後の候補数字の合計個数
            end if
            マスと置く数字の組を記録に追加
        end if
        盤面の状況を一時保存したものに復帰
        次の空白の初期配置マスへ
    end while
    if マスと置く数字の組の記録が空 then
        配置失敗として break;
    end if

    マスと置く数字の組の記録からランダムで 1 つを盤面に反映
    候補数字の絞り込み処理
end while
```

このアルゴリズムが持つ特徴としては、まず、可能な限り候補数字の合計が少なくなるように数字配置を行うため、解探索の際に絞り込むべき候補数字が少ない配置が得られる点である。これにより、完全にランダムな数字配置を行うのに比べ、得られた配置が唯一解を持つ可能性が高いのではないかと推測される。また、数字配置を行うマスの数が少なくなるため、数字配置にかかる時間を短縮する事ができるのではないかと考えられる。

2.3 解探索手法

数字配置アルゴリズムによって得られた初期配置が問題

として成立する為には、その初期配置が唯一解を持っている必要がある。本研究の手法では、人間が問題を解く際に用いる解法を導入し、唯一解の探索を行う。以下において、解探索に用いた、10 個の推論規則、絞り込み処理、および数字の仮置きによる推論ついて述べる。

2.3.1 推論規則

人間が数独の問題を解く際には、コンピュータが探索を行うように、あらゆる可能性を試行する事は通常不可能である。よって、マスに入る数字を推論を用いて絞り込み、確定させる事で問題を解いていく。本研究で導入した推論規則は以下の 10 通りである。

推論規則 1 候補数字が 1 つしかないマスをその候補数字で確定させる。

推論規則 2 行、列、ブロック内にある数字を候補数字に持つマスが 1 つしかないならば、そのマスをその候補数字で確定させる。

推論規則 3 ブロック内で、ある候補数字が特定の行内にあるマスにのみ存在するならば、その行内でこのブロック外であるマスからその候補数字を取り除く。

推論規則 4 ブロック内で、ある候補数字が特定の列内にあるマスにのみ存在するならば、その列内でこのブロック外であるマスからその候補数字を取り除く。

推論規則 5 行内で、ある候補数字が特定のブロック内にあるマスにのみ存在するならば、そのブロック内でこの行外にあるマスからその候補数字を取り除く。

推論規則 6 列内で、ある候補数字が特定のブロック内にあるマスにのみ存在するならば、そのブロック内でこの列外にあるマスからその候補数字を取り除く。

推論規則 7 行、列、ブロック内で、ある n 個のマスの候補数字に特定の n 種の数字のみが存在するならば、それ以外のマスからそれら n 種類の候補数字を取り除く。

推論規則 8 行、列、ブロック内で、ある n 種の候補数字が特定の n 個のマスにのみ存在するならば、それらのマスからその n 種以外の候補数字を取り除く。

推論規則 9 ある n 個の行群 R 内で、候補数字 X を持つマスが n 個の列内にのみ存在するならば、それらの列内で R の行外にあるマスから候補数字 X を取り除く。行と列が逆でも同様に成り立つ。

推論規則 10 ある 2 つの候補数字のみを持つマス群が複数の領域に連なって存在している時、それらと共通した行にのみ 2 つの候補数字のいずれかを持つ列があるならば、マス群の候補数字の一方を取り除く。

推論規則 1 から推論規則 8 までは松原氏が提案した推論規則である [2]。また、推論規則 9 は Basic Fish[3]、推論規則 10 は浜田ロジック [4] と呼ばれている。そして、推論規則 1 および 2 の適用によって数字が確定した際には、次のような絞り込み処理を行う。

絞り込み処理 数字が確定したマスと同じ行、列、ブロック内のマスの候補数字からその数字を取り除く。

本研究ではこれらの推論規則および絞り込み処理に加え、数字の仮置きによる推論を導入した。

2.3.2 数字の仮置きによる推論

数字が確定していないマスに対して、そのマスが持つ候補数字の 1 つを仮に置き、その仮定の下で盤面に矛盾が生じた場合、仮置きした数字はそのマスに入ることはないと分かる。よって、候補数字からその数字を取り除くことができる。

一般に、人間が仮置きの後に推論を長く続けていくことは困難である。よって、本研究では解探索に行き詰った際に数字の仮置きを行うかどうかは問題製作者が事前に指定できるようにした。さらに、数字の仮置きを行う対象となるマスは候補数字が 2 つまでに絞り込まれているマスのみとし、仮置き後に推論規則を適用するのは 1 回だけとした。

本研究での数字の仮置きによる推論の手順は以下の通りである。まず、盤面から候補数字が 2 個のマスを見つけ、数字の仮置きの対象とする。この時点で候補数字が 2 個のマスが見つからなかった場合には、仮置きは行わずに解探索を終了する。次に、盤面の状況を一時保存してから、対象のマスに 2 つの候補数字の一方を仮置きし、絞り込みを行ったあと、その仮定の下で推論規則を 1 回適用する。その結果、候補数字が 0 個のマスが発生した場合には、盤面を一時保存した状況に復帰させ、仮置きした候補数字を取り除く。そのようなマスが発生しなければ、盤面を一時保存した状況に復帰させ、もう一方の候補数字に対して同様に仮置きを行う。いずれの候補数字でも矛盾が生じなければ、候補数字が 2 個の他のマス全てに対し、順に同様の処理を行う。

2.3.3 解探索手順

本研究における解探索の手順をアルゴリズム 2 に示す。まず、推論規則を順次適用していき、候補数字の絞り込みや数字の確定を行う。推論規則の適用は、1 から 10 の番号順で行い、いずれかの規則が適用できた場合には再び 1 から番号順に適用していく。この時、推論規則 1 か 2 の適用により数字の確定が発生した場合には、絞り込み処理を行ってから推論規則の適用に移る。

解探索を進めた結果、どの推論規則を用いても盤面に変化が生じなくなった時に、全てのマスの数字が確定されていれば、その盤面が問題の唯一解であり、解探索を成功とする。もし空白のままであるマスが存在していた場合は、作製者が選択している場合には数字の仮置きを行い、解探索を再開する。仮置きが選択されていない、もしくは仮置きではどのマスからも候補数字を取り除くことができなかった場合、その盤面は唯一解が無い、もしくは本研究の手法では唯一解を求める事ができないと判定し、解探索を失敗とする。

Algorithm 2 解探索アルゴリズム

```
while true do
  推論規則 1 から 10 適用
  if 盤面に変化 then
    if 適用した解法が 1 textbfor 2 then
      絞り込み処理
    end if
    continue;
  end if

  if 仮置き未使用 and 仮置き許可あり then
    while true do
      候補数字が 2 個のマスの捜索
      if これ以上候補数字が 2 個のマスが見つからない then
        break;
      end if
      盤面の状況を一時保存
      2 つの候補数字を仮置きして絞り込み処理
      推論規則 1 から 10 適用
      if 矛盾が発生 then
        盤面の状況を一時保存したものに復帰
        仮置きした候補数字を削除
        break;
      else
        盤面の状況を一時保存したものに復帰
      end if
    end while
    if 盤面に変化 then
      continue;
    end if
  end if

  break;
end while
```

3. 実験

本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。なお、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/2016-GI-35/`

3.1 先行研究との比較実験

本研究で提案した手法と、先行研究 [1] の手法の問題生成能力と生成の所要時間に関して比較実験を実施した。以下に、実験の手順を記す。

(1) 問題の初期配置数を指定する。
(2) 指定された初期配置数に従って、100 通りの初期配置の位置のパターンをランダムに生成する。
(3) 各パターンに対して、本研究の手法と先行研究 [1] の手法とでそれぞれ 100 回ずつ問題生成を行う。

以上の手順によって問題を生成することができたパターンの数を問題生成能力の基準として比較した。また、問題生成に成功したパターンにおいて、生成にかかった時間を測定し、初期配置ごとに所要時間の平均値を算出した。今回は、比較を行う問題の初期配置数の範囲を 32 個から 18 個として実験を実施した。実験の結果を**図 3** と**表 1** に示す。

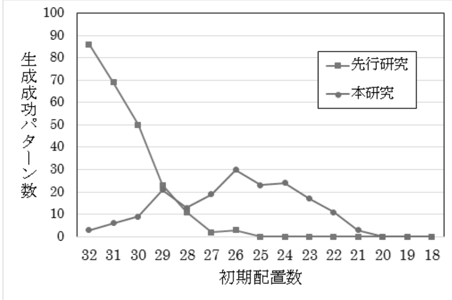

図 3 先行研究 [1] と本研究での問題生成成功パターン数の比較

表 1 先行研究 [1] と本研究での所要時間の平均値の比較

初期配置数	本研究（ミリ秒）	先行研究 [1]（ミリ秒）
32	695	56659
31	447	72762
30	638	77346
29	303	110972
28	380	102137
27	179	145207
26	158	119909
25	123	0
24	133	0
23	112	0
22	140	0
21	181	0
20	0	0
19	0	0
18	0	0

図 3 から、初期配置数が 25 個から 21 個の問題に関しては、先行研究 [1] の手法では問題を生成できたパターンが存在しなかったのに対し、本研究の手法では問題の生成が可能だった。この事から、初期配置数の少ない問題においては、本研究の手法は先行研究 [1] の手法よりも高い問題生成能力を有していると分かった。しかし、初期配置数が 29 個以上の問題に関しては、本研究の手法の方が成功パターン数が少なくなってしまっている。

表 1 から、本研究のプログラムによる問題生成の所要時間はいずれの初期配置数においても先行研究 [1] より短く、問題生成の所要時間の大幅な短縮に成功している事が分かった。

3.2 問題生成能力の測定実験

本研究で提案した手法による問題生成が現実的となる初期配置数の限度を調査する為に、問題生成能力の測定実験を実施した。実験の手順は節 3.1 の比較実験と同様である

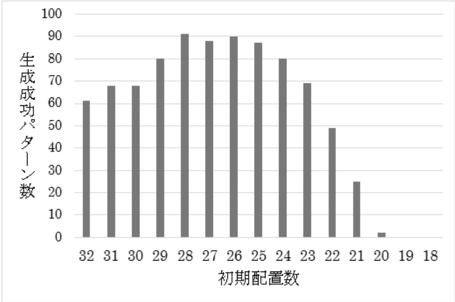

図 4 初期配置数ごとの問題生成成功パターン数

が、問題生成の試行回数は 10000 回とした。実験の結果を**図 4** に示す。

図 4 から、初期配置数が 21 個以上の問題に関して、100 通り中 20 通り以上のパターンにおいて問題の生成に成功しており、本研究で提案した手法による問題生成は現実的であると言える。しかし、初期配置数が 20 個の問題に関してはわずか 1 通りのパターンでしか問題の生成に成功しておらず、初期配置数が 19 個以下の場合は問題を全く生成できなかった。したがって、本研究のプログラムによって問題を生成するならば、21 個以上の初期配置を持つ問題である事が望ましいと分かった。

4. 考察

前章における 2 つの実験により、提案手法が先行研究 [1] の手法に比べ、問題生成能力が向上していることと所要時間を短縮していることが確認できた。これらは、新たな数字配置アルゴリズムの導入や解探索手法に施した改良によって実現したものと考えられる。本章では、これらの改善に対する、数字配置アルゴリズムと解探索手法の寄与について考察する。

4.1 問題生成能力

問題を生成するには、複数解を持つ初期配置になることを避けなければならない。那須氏の研究 [5] で、候補数字の合計数が少なくなるよう数字の配置を行ったことで、ランダムな配置に比べて解無しの可能性が高まったものの、唯一解を持つ配置を行う可能性も高まったことが報告されている。

前章で示したように本研究の手法がランダムな配置を行う先行研究 [1] より生成能力が高まっていることから、初期配置の位置が指定されている場合でも、候補数字の合計数を減らすことは、生成の成功率を高めるために有効な方針であると考えられる。

また、那須氏の手法 [5] では初期配置数が多くなると解無しの可能性が高くなっていたが、本研究では初期配置 22 個以上の問題においても半分以上の初期配置パターンで問題生成に成功している。これは、本研究では候補数字が 0 個のマスが発生しないものに限った配置をしたことが有効に働いたものと考えられる。ただし、初期配置数が 30 個程度を境に生成能力が落ちており、初期配置数が多すぎる場合それだけでは解無しを防ぎにくくなっていると推測される。解決策としては、数字配置の際に漠然と候補数字の合計数を少なくしようとするのではなく、初期配置数ごとに問題生成に成功しやすいような基準となる値を考案し、その値を目指すようにすることなどが考えられる。

4.2 所要時間

前章で示したように、本研究の手法は先行研究 [1] の手法よりも問題生成の所要時間を大きく短縮した。その要因として数字配置の方法を変更した事が挙げられる。先行研究 [1] では、盤面の左上端のマスから一マスずつ、そのマスが候補数字として持つ数字のいずれかをランダムに配置していくという手順で数字配置を行っていた。この手順は、数字配置を進めていくと候補数字を持たないマスが生じる可能性が急激に高まっていき、数字配置を何回もやり直さなければならなくなるという欠点があった。

しかし、本研究で導入したアルゴリズムは数字配置を行う対象となるマスは初期配置として指定されたマスに限られており、数字配置の回数が少なく、数字配置が最後まで行われる割合も先行研究 [1] より高い。これらの事から、初期配置として指定されたマスにのみ数字配置を行うようにしたことは、問題生成の所要時間の短縮に貢献したと考えられる。

5. 関連研究

本章では、推論規則と難易度判定に関する研究 [2]、問題生成の支援システムに関する研究 [6]、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 [7]、解盤面の列挙と番号付けに関する研究 [8] の 5 つについて本研究との関連について議論する。

5.1 数独の推論規則と難易度判定に関する研究

数独問題の難易度を数量的に示すための研究が数多く行われている。松原氏 [2] 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の算出に用いることを提案した。そして、推論の方法を盤面の状況毎に規則化した 8 個の推論規則を提案し、人間が数独を解く際に用いる解法をコンピュータで実現した。推論規則とそれを用いた難易度の算出は、コンピュータによる難易度別の問題生成につながると松原氏は推測している。

本研究の手法では、節 2.3.1 で述べたように、松原氏の提案した推論規則を解探索に導入した。また、四角の対角線の理論を一般化した解法である Basic Fish[3] と、浜田ロジック [4] を解探索に導入した。

5.2 数独の問題作成支援システムの設計と開発

前田氏は、数独の難易度と問題構造との関連を解明する事を課題として、作成中の問題についての様々な情報を提供して作成者の支援を行う問題作成支援システムを開発した [6]。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、解答と初期配置の依存関係、推定される問題の難易度などであり、利用者はこれらの情報を確認しながら、盤面に 1 つずつ数字を配置していく。

この研究 [6] では、問題作成の難点として、唯一解の保証、初期配置の位置のデザイン性、難易度の制御の 3 つを挙げて、システム設計の課題としている。また、候補数字を、その数字を入れる事によって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 候補に分類し、全ての空白マスの active 候補をただ 1 つに定める事が問題作成時の目標であるとしている。これらを踏まえて、各空白マスの dead 候補と active 候補を表示する機能と、あるマスに active 候補を配置した時のその他のマスの dead 候補と active 候補の変化を表示する機能を問題作成支援システムに実装している。

さらに、問題作成支援システムに導入した dead 候補と active 候補の判定を応用し、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 候補ができ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。ただし高速化の為に、数字配置の最初の半分程度に関してはランダムに配置を行い、残りの半分で active 候補を一気に絞り込むという手順に従う。この手法には、初期配置数 24 個未満の問題の生成成功率が著しく低いという結果が出ている [6]。

前田氏の研究 [6] で考案された手法では解の絞り込みに対して有効な active 候補のみを絞り込んでいるのに対し、本研究の手法では各空白マスの全ての候補数字の合計数ができる限り少なくなるように配置を決定しており、注目する候補数字を限定していない。

5.3 数独の少数ヒント問題の生成に関する研究

那須氏の研究 [5] では、初期配置が少ない問題の自動生成を目標として問題生成プログラムを開発し、実験を行っている。那須氏は事前に初期配置の数を指定し、候補数字の合計数ができる限り少なくなるように配置を行う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。このプログラムで 10000 通りの数字配置を生成し、解探索する実験を行った結果、初期配置が 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。

那須氏のプログラムと本研究の手法の違いとしては、本研究では初期配置の位置を指定できるという点が挙げられる。

5.4 ゲームにおける問題の評価および作成に関する研究

石田氏の研究 [7] では、数独の問題を解く過程に現れる面白さの要因として、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。石田氏は数独問題の自動生成について、人間が作製した問題に比べて面白さが劣る傾向にあると指摘し、面白さを測る尺度によって面白い問題と面白くない問題とを分別することで、自動生成でもある程度の面白さを持つ問題を得ることができるのではないかと提案している。

本研究では問題生成に際して面白さを考慮してはいないが、石田氏が着目した解法の複雑さと絞り込み回数という 2 点は、次章で挙げる問題の難易度の指定という課題を解決する上で重要な要素となるのではないかと考えられる。

5.5 本質的に異なる数独解盤面の列挙と番号付け

井上氏 [8] 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。井上氏の研究では、存在し得る全ての解盤面の中から本質的に異なる解盤面を高速に列挙する手法を提案し、さらに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。

本研究では数字配置の際に候補数字が 0 個のマスの発生を避けることで解無しの配置をしにくくしているが、数字配置途中の盤面と井上氏が本質的に異なる解盤面に対して付けた番号との対応がつけば、解無しの配置を数字配置の途中で判定することができると考えられる。

6. まとめと今後の課題

本研究では、初期配置の少ない問題を高速に生成できる問題生成手法の実装を目的として、高度な解法を用いる解探索手法と、初期配置を決定する際に、盤面の空白マスが持つ候補数字の合計数に着目し、これを最小化する数字配置アルゴリズムを導入した。これにより、唯一解を持つ初期配置を得られる可能性が向上し、解探索能力が向上した。その結果として、先行研究 [1] の手法に比べ、より少ない初期配置数の問題の生成に成功した。さらに、問題生成の高速化にも成功した。今後の課題として、以下のことが挙げられる。

- 初期配置数の多い問題の生成
  3.1 節で示したように、初期配置数が一定以上の問題に関しては先行研究 [1] のプログラムよりも問題生成に成功する初期配置が少ない。これを改善するため、候補数字の合計数に対し何らかの基準を考案する必要がある。
- 初期配置数のより少ない問題の生成

9 × 9 サイズの問題における最小の初期配置数は McGuire により 17 個であると証明されている [9]。また、5.3 節で述べたように、那須氏の研究 [5] において、初期配置数 18 個の問題の自動生成が確認されている。しかし、本研究では初期配置数 20 個までの問題しか生成できなかった。

- 難易度の指定

本研究で作成した問題生成プログラムでは、初期配置の位置を指定する事が可能だが、難易度を指定する事はできない。よって、問題作成者へのさらなる支援を目的として、指定された難易度の問題を生成する機能の導入を目指したい。

参考文献

[1] 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).
[2] 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).
[3] HoDoKu: Solving Techniques - Basic Fish (X-Wing, Swordfish, Jellyfish), http://hodoku.sourceforge.net/en/tech_fishb.php.
[4] 西尾徹也：ナンプレ超上級編 21，世界文化社 (2009).
[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] 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).
[7] 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).
[8] 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，pp. 741–742（オンライン），入手先 ⟨http://ci.nii.ac.jp/naid/110007505881/⟩ (2009).
[9] McGuire, G., Tugemann, B. and Civario, G.: There is no 16-Clue Sudoku: Solving the Sudoku Minimum Number of Clues Problem, *CoRR*, Vol. abs/1201.0749 (online), available from ⟨http://arxiv.org/abs/1201.0749⟩ (2012).