

ヒント数17の数独パズルの効率的な生成に関する研究

223426015 長尾 卓

山本研究室

1. はじめに

数独パズルは、ペンシルパズルの一種である。ペンシルパズルとは、問題に対して答えを徐々に鉛筆で書き込んでいき、答えを導くようなパズルのことである。ペンシルパズルには、数独パズルのほかにスリザーリンクや虫食い算などが知られている。数独パズルは、与えられたヒント（例：図1左）から、1から9の数字を用いて縦、横、 3×3 ブロックのどの数字にも重複させないように、マスに埋めていくパズルである。図1左で与えられる問題の答えは図1右である。また、1つの問題から得られる最終盤面はただ1通りである必要がある。以降、数独パズルの最終盤面を「解」と表す。本研究には先行研究[1]がある。先行研究の目的は、ヒント数が少ない数独パズルの問題を確率的に効率よく生成することである。先行研究が提案するヒント生成アルゴリズムのヒント数17の問題生成割合は約5%であり、1つの問題を生成する平均時間¹は約105分であった。なお、数独パズルにおけるヒント数の下限は17個であることが[3]により証明されている。先行研究により、確率的にヒント数17の問題を生成できることが分かったため、本研究ではヒント数17の問題を効率よく生成することを目的とした。本研究が提案するヒント生成アルゴリズムは、先行研究のヒント生成アルゴリズムを改良したものである。提案するヒント生成アルゴリズムはヒント数17の問題を約93%の確率で生成し、先行研究と同環境で1つの問題を生成する平均時間は約11分であった。

2. 本研究で用いる手法

本研究ではシミュレйтеッド・アニーリング (SA) [4] と、ビームサーチ (BS) [5], Algorithm X [6] の3つの手法を用いる。以下にそれらの説明をする。

SAは目的の定常分布をもつマルコフ連鎖[7]を安定して構成するヒューリスティックなアルゴリズムである。マルコフ連鎖上のある状態からランダムな近傍状態に遷移させることを繰り返し、マルコフ連鎖を定常状態にした後に、SAのパラメータである内部温度 T を微減させて定常分布を目的の定常分布へ緩く変化させる。以上を繰り返すことで、目的の定常分布をもつマルコフ連鎖を構成していく。そ

			2	1			3
5	9			8			
		2	3		4		
1						5	9
			6				
		3					4
				5			8

7	4	8	5	2	1	9	6	3
5	9	6	7	8	3	2	4	1
3	2	1	9	4	6	8	7	5
6	5	2	3	9	4	7	1	8
1	3	4	2	7	8	5	9	6
9	8	7	6	1	5	4	3	2
2	7	3	8	6	9	1	5	4
4	6	9	1	5	2	3	8	7
8	1	5	4	3	7	6	2	9

図1: 17個のヒントによる数独パズルの問題（左）とその解（右）。

して、目的の定常分布を得るまでに、得たい状態をサンプリングすることが可能である。一般的なSAは、ある状態の出現確率をボルツマン因子により決定し、近傍状態への遷移にはメトロポリス法[4]を用いる。本研究もボルツマン因子やメトロポリス法を用いる。

BSは、木構造上の根からスタートする w 本のパスを考えて、それらのパスを同時に葉の方向に伸ばしながら評価値がより良いノードを探索するヒューリスティックな探索アルゴリズムである。BSの性質は w によって大きく変化する。 w が大きい場合は、空間的計算量と時間的計算量の両方が大きくなる。一方で、 w が小さい場合は、最適解や最適解に近いノードへのパスが途中で絶たれる可能性が大きくなる。

AXは、集合 S と複数の部分集合 s をもつ厳密被覆問題をバックトラックで効率よく解くアルゴリズムである。ある時点までに選択してきた全ての s と互いに素の s のみを保持し、保持している部分集合から適切に s を選択することを再帰的に繰り返す。保持している部分集合のみでは S の要素をカバーできないと判断した場合に、効率よく枝切りを行うため効率がよい。

3. 先行研究のヒント生成アルゴリズム

本研究が提案するヒント生成アルゴリズムは[1]の方法を改善したものである。先行研究のヒント生成アルゴリズムは、まず、ヒント集合 $H^{(0)} = \emptyset$ を用意する。 $H^{(0)}$ の右肩の数字が表記されている場合は、ヒント集合 H のサイズを表す。次に、適切なヒント h を H に添加していき、 H から得られる解の集合の大きさ $|S(H)|$ を徐々に減少させていく。 h はマスの位置 p と数字 n の組である。適切な h とは、 $H^{(i)}$ に添加して得られる $H^{(i+1)}$ の $|S(H)|$ が最小となる h のことである。最終的には、 $|S(H)| = 1$ となるまで h を H に添加して問題を生成する。 h の選択方法は、 H から $S(H)$ を得て、得た解に出現した場所と数字の組である複数の要素のうち最も出現回数が少ない要素を h とする。なお、 H から $S(H)$ を得るためには、バックトラック(BT)を用いる。一方で、 $H^{(i)}$ ($i < 14$)は十分に解集合の大きさが減少しておらず、BTで解を列挙することは効率が悪い。そのため、シミュレйтеッド・アニーリング(SA)を用いることで $H^{(i)}$ から得られる解を偏りなく等確率に多く生成することを行い、確率的に h を選択する。

4. 先行研究のヒント生成アルゴリズムに加えた改良

前述したように、本研究が提案するヒント生成アルゴリズムは、先行研究のヒント生成アルゴリズムを改良したものである。改良した点は主に4点であり、2点はヒント数17の問題生成割合を高めるための改良で、残りの2点はヒント生成の高速化のために行った改良である。ヒント数17の問題生成割合の向上のための改良1点目は、生成した $H^{(14)}$ に3つの h をまとめて添加するようにした点である。これにより、生成した $H^{(14)}$ があるヒント数17の問題の部分集合である場合は、必ずそのヒント数17の問題を生成できるようになる。先行研究のヒント生成アルゴリズムにこの改良を加えた場合のヒント数別の問題生成割合は図2

¹プログラムの実行環境は以下の通りである: OS: Linux Ubuntu 16.04.7, CPU: Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz, メモリ: 64G, コンパイラ: gcc5.4.0.

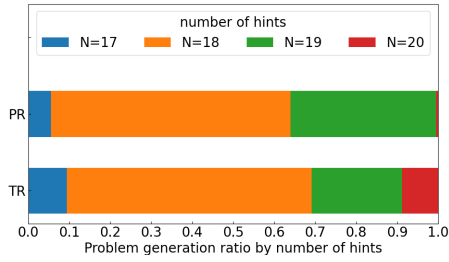


図 2: 先行研究のヒント生成アルゴリズムに $H^{(14)}$ に 3 つのヒントをまとめて添加する変更を加えたアルゴリズムの問題生成割合. PR が先行研究であり, 3R が本変更を加えたアルゴリズムを表している.

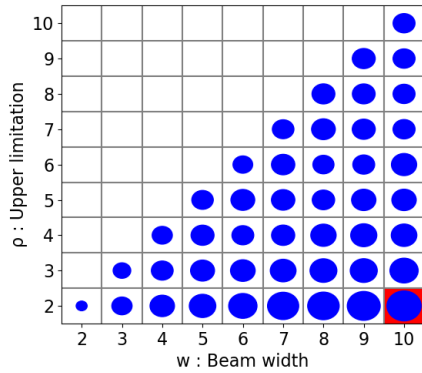


図 3: ビーム幅 w ($2 \leq w \leq 10$) と上限 ρ ($2 \leq \rho \leq w$) の組み合わせにより, $B^{(14)}$ に含まれる最も解集合が小さい $H_j^{(14)}$ の解集合の大きさが $U = 160,000$ 以下になる確率を円の大きさで表した図. w は横軸で表し, ρ は縦軸で表す. 最も生成割合が高かった w と ρ の組を赤色で表している.

のように変化し, ヒント数 17 の問題生成割合は約 5% から約 10% に上昇した. 2 点目は, 生成していく $H^{(i)}$ の解集合を柔軟に減少させていくために, 最も解集合が小さい w 個の $H^{(i)}$ を保持していく BS を用いた点である [2]. 生成した $H^{(14)}$ の解集合が小さいときほど, $H^{(14)}$ に 3 つのヒントを添加する場合に, ヒント数 17 の問題を生成しやすくなることが実験により分かっている. 本研究で行う BS には一般的な BS にはないパラメータがあり, 1 つの $H^{(i)}$ から生成する $H^{(i+1)}$ の個数の上限 ρ を設けている. パラメータ ρ は, 解集合が減少していきづらい $H^{(i)}$ から生成された $H^{(i+1)}$ のみを BS で保持することを避ける目的がある. ヒント生成の高速化のための改良 1 点目は, $H^{(i)}$ ($i \geq 14$) から解集合を得るために BT ではなく, Algorithm X を用いるようにした点である. Algorithm X は厳密被覆問題を効率よく解くことができるアルゴリズムであり, 数独パズルは厳密被覆問題として定義することができることから, 本研究は BT ではなく Algorithm X を用いることとした. この変更により, $H^{(14)}$ にまとめて添加する 3 つのヒントの選択時間は約 66 ~ 80% ほど短縮した. 2 点目は, H から解をサンプリングする SA における, 近傍状態の決定方法を変更した点である. 先行研究の SA は, ランダムに行を

選択し, 選択した行内のヒント以外の 2 マスの数字を交換するように遷移先の近傍状態を決定する. ただし, 初めに H を考慮して全行に 1 から 9 が無作為に書き込まれた初期盤面から遷移を行うようにする. 一方で, 本研究の SA は, ランダムに 3×3 ブロックを選択し, 選択したブロック内のヒント以外の 2 マスの数字を交換するように遷移先の近傍状態を決定する. ただし, 初めに H を考慮して全ブロックに 1 から 9 が無作為に書き込まれた初期盤面から遷移を行う. この変更により, SA の解のサンプリング効率が約 5 倍向上した.

参考文献

- [1] 古川 湧: ヒントの少ない数独パズルの生成に関する研究. 2020 年度名城大学大学院理工学研究科修士論文 (2021).
- [2] 長尾 卓: ビームサーチを用いたヒント数 17 の数独パズルの効率的な生成について. ゲームプログラミングワークショップ 2022 論文集, pp. 96–103 (2022).
- [3] G. McGuire, B. Tugemann, and G. Civario: There is no 16-clue Sudoku: Solving the Sudoku minimum number of clues problem via hitting set enumeration. *Experimental Mathematics*, 23:2, pp. 190–217 (2014).
- [4] Bruce E. Rosen, 中野 良平: シミュレーテッドアニーリング: 基礎と最新技術, 人工知能学会誌, Vol.9, No.3, pp.365–372 (1994).
- [5] Y. Inoue and S. Minato: Acceleration of ZDD Construction for Subgraph Enumeration via Pathwidth Optimization. TCS Technical Report, TSC-TR-A-16-80, (2016).
- [6] D. E. Knuth: Dancing links. *Millennial Perspectives in Computer Science*, pp. 187–214 (2000).
- [7] Berg, B.: Markov Chain Monte Carlo Simulations and Their Statistical Analysis. World Scientific Publishing Co. Pte. Ltd. (2004).