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，マスを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 [2] により証明されている．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善したものであり，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．また，生成したヒント数 17 の問題の中に異なる問題がどの程度含まれているか調べた結果，200 問中 15 ～ 34 問が含まれていた．

2. 本研究で用いる手法

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

SA は，ある状態からランダムな近傍状態に遷移させながら，最適解を得るアルゴリズムである．内部温度というパラメータを設定し，徐々に内部温度を低下させていき，内部温度に依存した遷移確率で状態の遷移を行う．本研究では，メトロポリス法を用いてマルコフ連鎖を構成し，マルコフ連鎖上で SA を行っている．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

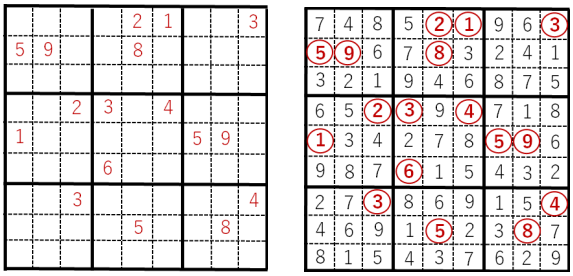

図 1: ヒント数 17 の数独パズル（左）とその解（右）．

[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.

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ため，慎重に設定する必要がある．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，効率よく枝狩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，シミュレイテッド・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善

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善したものである．改善した点は主に 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善で，残りの 2 点はヒント生成の高速化のために行った改善である．ヒント数 17 の問題生成割合を高めるための改善の 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，生成してきた $H^{(14)}$ 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善を加えた場合のヒント数別の問題生成割合は約 5%から約 10%に上昇していた．2 点目は，$H^{(i)}$ の解集合の大きさ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．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 $H^{(i)}$ から構築する $H^{(i+1)}$ 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ずらい $H^{(i)}$ から構築された $H^{(i+1)}$ のみを BS で保持することを避ける目的がある．図 2 より，実験した w と ρ $(2 \leq \rho \leq w \leq 10)$ の範囲では，

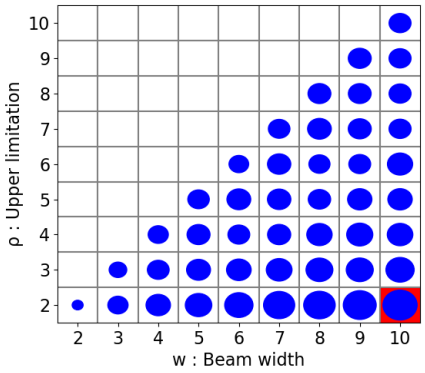

図 2: ビーム幅 w と上限 ρ $(2 \leq \rho \leq w \leq 10)$ の組み合わせ毎に，生成した w 個の $H^{(14)}$ の中で最小の解集合の大きさが 160,000 以下になる確率を円の大きさで表した図．最も生成割合が高かった w と ρ の組を赤色で表している．

最も柔軟に解集合が減少するパラメータは $(w, \rho) = (10, 2)$ であった．w を大きくすると，さらに柔軟に解集合が減少すると予想する．ヒント生成の高速化のための改善 1 点目は，数独パズルが厳密被覆問題と見なすことができることから，$H^{(i)}$ $(i \geq 14)$ から解集合を得るために BT ではなく，より効率の良い AX を用いるようにした点である．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，近傍状態を変更した点である．先行研究の SA は，ランダムに選択した行の中で，ヒント以外の 2 マスの数字を交換してできる盤面を近傍状態とする．一方で，本研究の SA は，ランダムに選択した 3×3 ブロックの中で，ヒント以外の 2 マスの数字を交換してできる盤面を近傍状態とする．ただし，初めに H を考慮して，先行研究は全行に，本研究は全ブロックに 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．

5. 改善によるヒント数 17 の問題生成の割合と効率の変化

前節で述べたような，ヒント数 17 の問題生成割合を高めるための改善の 2 つの組み合わせで，どの程度ヒント数 17 の問題生成の割合と効率が変化するか調べた．その結果を図 3 に示し，1 つのヒント数 17 の問題を生成する平均秒数を図 4 に示す． 図 3 より，w が大きいほどかつ，$H^{(14)}$ に 1 つずつではなく 3 つのヒントをまとめて添加する方が，ヒント数 17 の問題生成割合は高くなることが分かった．一方で，図 4 からは $H^{(14)}$ 以降のヒント生成方法の差異によってはヒント数 17 の問題生成効率に目立った違いがないことが分かった．つまり，$H^{(14)}$ に 1 つずつではなく 3 つのヒントをまとめて添加する方が計算量が大きいことが分かる．図 4 の中で最もヒント数 17 の問題生成効率が良かった場合は，$(w, \rho) = (10, 2)$ かつ $H^{(14)}$ に 3 つのヒントをまとめて添加する場合であり，1 つのヒント数 17 の問題を生成する平均時間は 736 秒であった．また，生成したヒント数 17 の問題の中に異なる問題がどの程度含まれているか調べた結果，200 問中 15 ∼ 34 問が含まれていた．

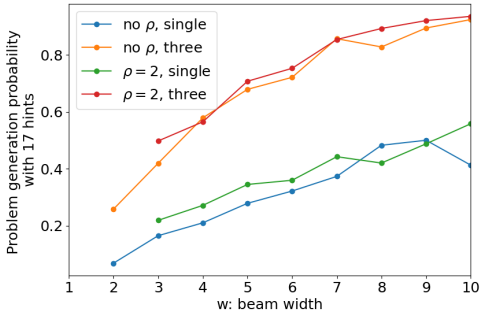

図 3: w, ρ $(2 \leq w \leq 10, \rho : 2, w)$ と $H^{(14)}$ にヒントを 3 つ (three) または 1 つずつ (single) ヒントを添加する場合のヒント数 17 の問題生成割合を示した図.

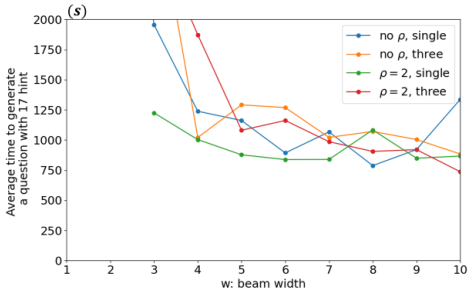

図 4: 図 3 の結果を用いて 1 つのヒント数 17 の問題を生成する平均秒数を求めて示した図.

6. まとめと今後の課題

先行研究のヒント生成アルゴリズムにビームサーチや Algorithm X を用いることと，$H^{(14)}$ に 3 つのヒントをまとめて添加する変更を行い，ヒント数 17 の問題生成効率を改善した．1 つのヒント数 17 の問題を生成する平均時間は先行研究が 126,000 秒であったのに対し，本研究は $(w, \rho) = (10, 2)$ かつ $H^{(14)}$ に 3 つのヒントをまとめて添加する場合の 736 秒であった．また，生成した多くのヒント数 17 の問題の中に異なる問題がどの程度含まれているか調べ，200 問中 166 ∼ 185 問が含まれていた．今後の課題は，w や ρ をより広い範囲で実験して，より適切な w や ρ の組を見つけることや，より等確率にヒント数 17 の問題を生成するようにアルゴリズムを改善することである．

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] 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).

[3] Bruce E. Rosen, 中野 良平：シミュレーテッドアニーリング：ー基礎と最新技術ー，人工知能学会誌，Vol.9, No.3，pp.365–372（1994）．

[4] Y. Inoue and S. Minato: Acceleration of ZDD Construction for Subgraph Enumeration via Pathwidth Optimization. TCS Technical Report, TSC-TR-A-16-80, (2016).

[5] D. E. Knuth: Dancing links. *Millennial Perspectives in Computer Science*, pp. 187–214 (2000).