

# ペイントロジックの問題作成支援 Supporting the Creation of Nonogram Puzzles

長谷川 真子

Mako Hasegawa

法政大学情報科学部コンピュータ科学科

E-mail: mako.hasegawa.9n@stu.hosei.ac.jp

## Abstract

“Pencil puzzles” are known to be puzzles that people solve by writing answers to illustrated questions such as mazes, arithmetical restorations, and crossword puzzles. Nonogram puzzles are pencil puzzles consisting of grids of squares whose answers show pictures or letters. Although there has been research on the automatic solving of nonogram puzzles, there is not much research on the creation of nonogram puzzles. One of a few such studies is by Henstra; it makes puzzles based on images such as a human’s face. However, it is still difficult to make complicated and interesting puzzles automatically. This paper proposes a method that uses pictures drawn by hands instead of automatically making puzzles based on images. The method also supports users in creating interesting puzzles. For this purpose, the method examines the difficulty of a given puzzle by using the number of solutions, the necessary computation time, and the number of backtracks. The user tries to improve the puzzle by reducing the number of solutions, reducing the computation time, and reducing the number of backtracks. This paper also presents a tool based on the proposed method, and provides the results of experiments on the creation of nonogram puzzles by using the proposed method.

## 1. はじめに

ペンシルロジックは、クロスワードや数独など、様々なパズルが存在し、多くの人々が楽しんでいる。このペンシルパズルとは、図示された問題に対して答えを書き込むことによって、最終的な解答を行う形式のパズルのことである。ペンシルパズルは雑誌やゲーム機でも多く販売されており、例えばニンテンドーDSの「SUDOKU数独」の売り上げ本数は約16万3000本である。

愛好者の多いペンシルパズルの1つにペイントロジックがある。ペイントロジックとは、縦と横の数字をヒントに、塗り潰すべきマス目を割り出し、そのとおりに塗り潰していくと、最終的に絵や文字が浮かび上がるというペンシルパズルである。ペイントロジック以外にも、お絵かきロジック、ののぐらむ、イラストロジック、ピクロスなどとも呼ばれ、1988年に西尾徹也といしだのん

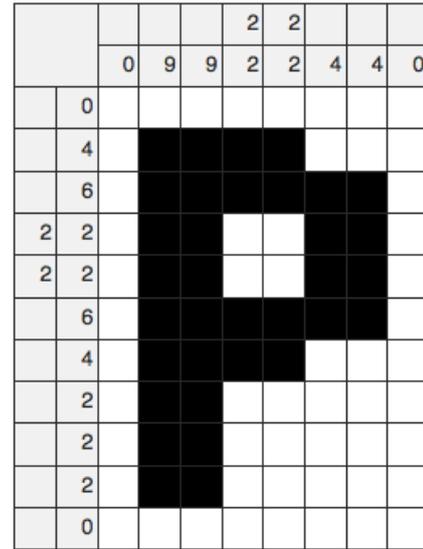


図1 ペイントロジックの問題例。

の二者が独自に発案したとされる[1]。

ペイントロジックの基本的なルールは以下のとおりである。最初の時点ではいずれのマスも塗り潰されておらず、連続して塗り潰すべきマスの個数が縦と横の各列に対して示されている。解答者はそれらの数字に基づきマスを塗り潰していく。ただし、縦と横で同じ列の隣り合った数字の間は全て1マス以上空ける必要がある。

ペイントロジックに関しては、問題を自動で解くアルゴリズムの研究は多く存在するが、問題を作成する研究は少ない。その中でも、図からの問題作成の研究は存在する[2]が、問題作成支援についての研究は著者の知る限り存在しない。

本研究では、ペイントロジックの問題の作成を支援する手法を提案する。特に問題の難易度を自動的に判定することで、適切な問題を作成できるようにする。ペイントロジックの簡単な問題と難しい問題の判定基準として、一般的には「塗るための大きい数があり見られない」、「全体のマス目が大きい」、「塗るマスの数が細かい」などがあげられるが、具体的な計算方法を見つけることは困難である。そこで本研究では、作成した問題をコンピュータに解かせ、解の個数と解答時間を表示して簡単な問題と難しい問題の判定を行う。

## 2. 関連研究

ペイントロジックについては、解答方法についていくつかの研究が存在する[3][4][5]。問題作成については、エディタで絵を描いて問題を作るツールは多数存在している。また、図からの問題作成の研究も存在する[2]。これは、人などの画像を基に問題を作成するものであるが、自動作成だけでは複雑な問題を作るのは困難であった。一方、問題作成支援についての研究は著者の知る限りで存在しない。

他のパズルでは自動作成の研究がある。例えば倉庫番の問題の自動作成が挙げられる[10]。これは、倉庫番というパズルゲームをコンピュータによっていかに面白く自動作成するかという研究である。倉庫番は奥が深く、問題によっては難しいパズルとなる。この研究の問題作成は、作成した問題を解き、解のデータを作り、得られたデータを基に問題の面白さを評価するという手順からなる。

## 3. 提案手法

本研究で提案する問題作成支援手法について説明する。本研究における問題の作成支援とは、ユーザが作成した問題をツールが分析して、その結果をユーザに提示することで、ユーザがより良い問題を作成できるようにすることである。

本研究ではバックトラック法でペイントロジックを解くソルバーを採用しており、問題を分析して得られる以下の情報をユーザに提示する。

- 全ての別解の計算
- バックトラック法で解く時間
- バックトラックの回数

ユーザはこれらの数値を減らせるように問題の改善を行う。

## 4. 実装

本論文の提案手法に基づき、ペイントロジックの問題作成支援ツールを実装した。実装には Java 言語を用い、グラフィカルユーザインタフェースの作成には Swing を使用した。

本ツールを用いることで、図 2 のような問題を作成することができる。この画面では、マウスの左クリックで操作することにより問題を作成できる。一度クリックするとそのマスは黒に変わり、もう一度塗り潰したマスをクリックすると白に戻る。この画面では、クリアボタンを押すと白紙の状態に戻すことができる。また、保存ボタンを押すことで作成した問題を保存でき、一覧ボタンを押すと、今までに作成した問題を見ることができる。その一覧画面は図 3 のように表示される。ここでは保存したデータが表示されている。編集ボタンを押すと、問題作成画面に戻り、再度問題を作成し直すことができる。

難易度チェックボタンを押すと、解の個数と自動解答時間、バックトラックの回数が図 4 のように表示される。本研究ではこの難易度表示の結果を用いて実験を行う。

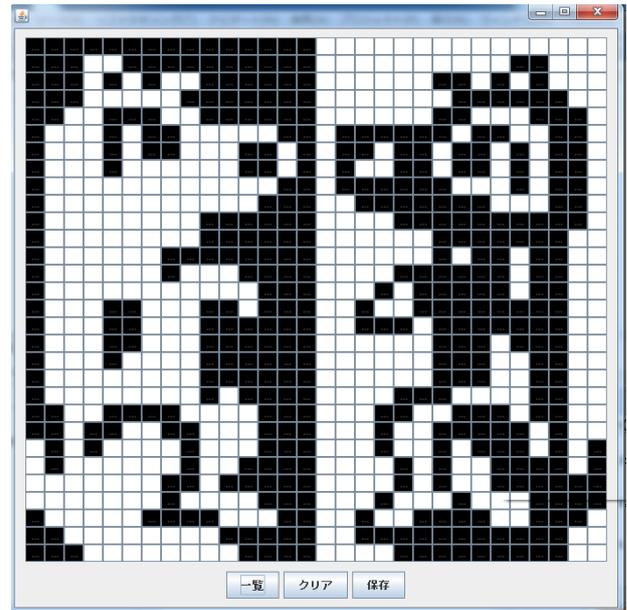


図 2 問題(dog)の作成例.

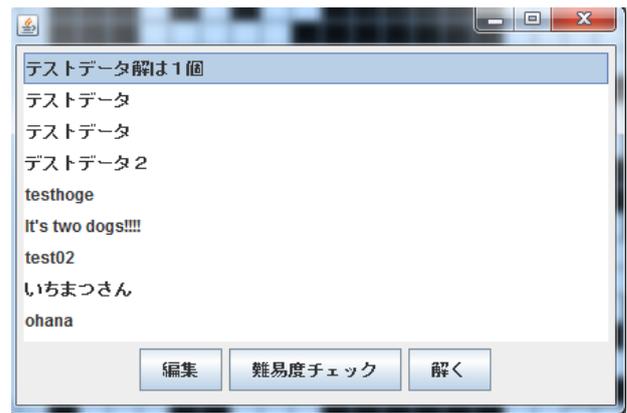


図 3 作成した問題の一覧画面.



図 4 難易度チェック表示.

### 5. 実験

10 個の問題を作成し、それぞれの解の個数と解答時間、バックトラック数、黒の個数、行と列のそれぞれの幅、最小値と最大値を調べる実験を行った。これらの問題の中で「dog」とは図 2 を示す。塗るマスの小さな数が多く、大きい数が少ないと時間がかかり、計測できなくなってしまうため、今回はできるだけ大きい数を用いた問題で実験している。

この実験結果を表 1 に示す。この表は、黒を塗り潰した個数、解の個数、自動解答時間、行列の幅（行または列で塗り潰すマス目の個数）、行列それぞれの最小のマス数と最大のマス数をまとめている。この表から規則性を見ていく。この実験結果から推測できることについて、いくつかの特性が認められる。

表 1 実験結果(1).

	黒の個数	解の個数	時間	バックトラック数	行幅	行最小値	行最大値	列幅	列最小値	列最大値
dog	438	1	87	169437	10	1	15	7	1	30
市松模様	450	2	1986	8144179	15	1	1	15	1	1
問題1	300	2	565	2526869	1	1	10	2	1	10
問題2	150	34	593	2499936	2	1	5	2	1	5
問題3	240	4	240	1089468	2	1	8	2	1	8
問題4	270	4	355	1598309	2	1	9	2	1	9
問題5	210	10	192	880008	2	1	7	2	1	7
問題6	288	1	2217	9755027	3	1	6	2	2	10
問題7	218	4	13917	58247176	3	1	4	2	2	8
問題8	332	1	2884	13109671	5	2	3	2	1	14

まず、解が 1 個、2 個、4 個、10 個などの複数解の問題があるが、解が 3 個や 15 個など、1 を除く奇数個存在する問題は見つからなかった。手動で行ったため探すことが困難であるので、手動で問題を作成しない手法を利用してコンピュータに解かせることで 1 を除く奇数の解を見つけることができるのではないかと考えられる。

また、黒の個数は関係なく、行と列の最大値が小さいほど、または行または列の塗るマスの個数が多いほどかかる時間やバックトラック数が大きい。特に 10 より小さい値になるとその大きさが飛躍的に上がる。しかし、塗るマスの最大値が 4 以下になると計測できなくなってしまう。

この中から最も解の個数が多い問題 2 (図 5) を選択し、解の個数、時間、バックトラック数を少なくなるように、少しずつ手動で修正した。その結果が表 2、表 3、表 4 である。これらの表は、1 個または複数個黒を増やし、その塗り潰した場所によって解の個数、時間、バックトラック数がどう減少できるか調べたものである。表 2 では、塗り潰したマス目が最も多い部分の最短距離を中心に調べて比較し、表 3 では、塗り潰したマス目が最も少ない部分からの最短距離を中心に調べたものである。表 4 は、塗り潰したマス目が最も多い部分と最も少ない部分の最短距離も共に照らし合わせて調べたものである。各々の表に表示されている問題 2 の修正版は、表 1 の問題 2 をもとに少しずつ修正したものである。

これらの表から、いくつかの特性が見られた。一つ目は、塗り潰したマス目が最も少ない部分の最短距離に関係なく、最も多い部分の最短距離が短くなるにつれて、解の個数、時間、バックトラック数が減少していること

がわかる。また、最も少ない部分からの最短距離を中心に調べた表では、全体で行または列で塗り潰すマス目の最大値が増えない限りほとんど変化しないことがわかった。さらに、やはり塗るマスの個数を増やす際は、塗るマス目を長く作成した方が解の個数、時間、バックトラック数が減少することがわかった。解が 1 個のみである修正 15 と修正 19 の問題を図 6 と図 7 に示す。

今回の結果から、黒の個数を増やせば全ての値が減少するとは限らず、場所によって解の個数や時間、バックトラック数を十分に減少させることができることがわかった。

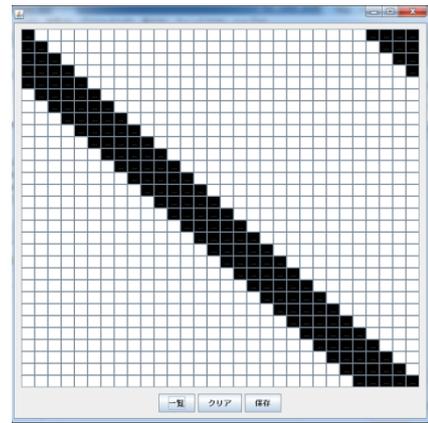


図 5 問題 2.

表 2 実験結果(2).

	黒の個数	マス目最大値からの位置	解の個数	時間	バックトラック数
問題2	150		34	593	2499936
修正1	151	左13、下12	24	1324	2510467
修正2	151	左12、下12	24	940	2532949
修正3	151	左9、下15	24	1100	2524459
修正4	151	左11、下12	24	1100	2530199
修正5	151	左11、下11	24	1037	2531941
修正6	151	左7、下7	11	1198	2629709

表 3 実験結果(3).

	黒の個数	マス目最小値からの位置	解の個数	時間	バックトラック数	列の幅	行と列の最大値
問題2	150		34	593	2499936	2	5, 5
修正7	151	左3、下4	44	1052	2522938	2	5, 5
修正8	151	左1、下1	34	756	1547187	3	5, 5
修正9	152	左1、下0	34	841	1748153	2	5, 5
修正10	152	左1、下0	34	863	1748556	2	5, 5
修正11	158	左0、下1	10	669	1527277	2	6, 5
修正12	151	左2、下1	4	933	1938351	2	5, 5
修正13	151	左0、下1	6	593	685043	2	5, 5
修正14	152	左0、下1	2	282	608621	2	6, 6
修正15	153	左1、下0	1	386	545780	2	6, 6

表 4 実験結果(4).

	黒の個数	マス目最大値との距離	マス目最小値との距離	解の個数	時間	バックトラック数	行と列の幅
問題2	150			34	593	2499936	2, 2
修正16	151	左10、下10	左5、下0	10	1176	2184353	3, 2
修正17	151	左11、下11	左3、下3	34	406	904732	3, 2
修正18	151	左4、下4	左6、下12	8	1758	3618098	2, 2
修正19	151	左2、下2	左1、下21	1	265	390956	3, 2
修正20	151	左11、下12	左2、下7	34	628	1236653	2, 3

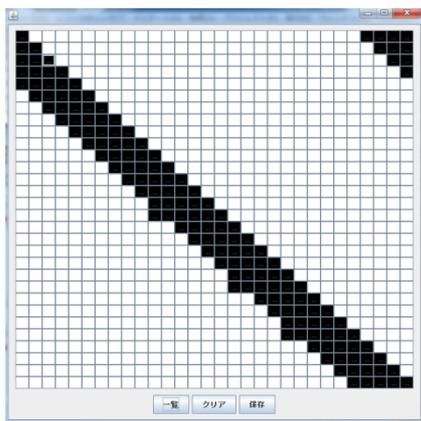


図 6 修正 15.

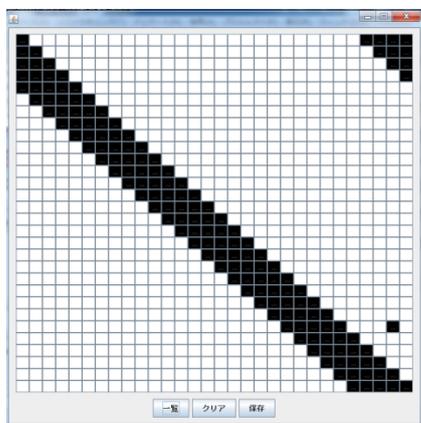


図 7 修正 19.

## 6. おわりに

本研究ではペイントロジックの問題作成支援手法を提案した。本手法では、バックトラック法を用いたソルバーで問題を分析し、その結果を利用して問題を改善できるようにした。さらに本研究ではこの問題作成支援手法に基づき、問題作成支援ツールを実装した。このツールでは、問題をユーザがいくつか作成し、Java を用いたツールで各々の問題に対する解の個数、自動解答時間、バックトラックの回数を調べることができる。

この研究を通して、いくつか今後の課題が明確になった。まず、人がどのような問題に面白さを感じるか評価を行うことである。コンピュータで問題の面白さを判断するのは容易でない。そのため、人に問題を解かせ、どの問題がより面白くなるのか判定してもらうことで、より良い問題の作成支援に活かすことが必要である。

さらに、本論文の問題作成支援ツールでは、塗るマス数が少ない列が存在すると、バックトラックの分岐数が膨大になってしまい、計算を続行できなくなってしまう。そのため、より優れたアルゴリズムでソルバーを作成しなければならない。塗るマスが本研究よりもさらに少ない問題では、解の個数や自動解答時間、バックトラ

ックの回数の値はどう変化するか、その修正によってどのように値が変化するか実験する必要がある。

その他にも、解が奇数存在する問題を探すことができなかったため、奇数個の問題を作成することは不可能なのか否かを調べる必要がある。

他にも、バックトラック法以外の方法ではどのような結果が出るのか調べようと考えている。関連研究で紹介した **fully probing** と呼ばれる手法[4]を参考にして研究を行い、より効果的なアルゴリズムを発見することを目標にする。

最後に今後の課題の方向性として、問題の難易度をどのように判定していくのが挙げられる。ペイントロジックに関する雑誌やゲーム、アプリケーションでは 3 段階や 5 段階に分けて難易度を判定している。その判断基準は具体的にどう分けているのか調べる必要がある。おそらく何らかの自動判定手法を利用して分類しているのではないかと考えられる。

これらの課題から、さらなる研究の進展を行い、多くの人に面白いと判断される問題はどのような問題なのかを見つけていく予定である。

## 文 献

- [1] Wikipedia, "Nonogram." <https://en.wikipedia.org/wiki/Nonogram>
- [2] Sjoerd Jan Henstra, "From Image to Nonogram: Construction Quality and Switching Graphs," Master's Thesis, Leiden University, 2011.
- [3] K.J.Batenburg and W.A.Kosters, "A Discrete Tomography Approach to Japanese Puzzles," Leiden University, 2004.
- [4] I-Chen Wu, Der-Johng Sun, Lung-Pin Chen, Kan-Yueh Chen, Ching-Hua Kno, Hao-Hua Kang and Hung-Hsuan Liu, "An Efficient Approach to Solving Nonograms," *Proc. IEEE*, vol.5, no.3, pp.251-264, September 2013.
- [5] Shu-Jim Yen, Tsan-Cheng Su, Shih-Yuan Chiu and Jr-Chang Chen, "Optimization of Nonogram's Solver by Using an Efficient Algorithm," *Proc. TAAI*, pp.444-449, November 2010.
- [6] Jessica Benton, Rion Snow and Nolan Wallach, "A Combinatorial Problem Associated with Nonograms," *Linear Algebra and its Applications*, vol.412, no.1, pp.30-38, 2005.
- [7] Ryan Mullen, "On Deter 最小値 ing Paint by Numbers Puzzles with Nonunique Solutions," *Journal of Integer Sequences*, vol.12, article 09.6.5, 2009.
- [8] 鈴木浩, 佐藤尚, 速水治夫, "子供を意欲的にペーパークラフト作りへと導く 3 次元ゲームシステムの開発," *情報処理学会論文誌*, vol.3, no.1, pp.10-19, 2015.
- [9] 八登崇之, "スリザーリンクの NP 完全性について," *情報処理学会研究報告(アルゴリズム)*, vol.2000, no.84 (2000-AL-074), pp.25-31, 2000.
- [10] 村瀬芳生, 松原仁, 平賀謙, "「倉庫番」の問題の自動作成," *情報処理学会論文誌*, vol.39, no.3, pp.567-574, 1998.