数式処理 *Bulletin of JSSAC(2014)*
Vol. 20, No. 2, pp. 3 - 22

論文

グレブナー基底計算プログラムは どの程度のパズルを解くことができるか？

杉原 彩　　田辺 康範

徳島大学大学院総合科学教育部　　広島県立大門高等学校

鍋島 克輔*

徳島大学大学院ソシオ・アーツ・アンド・サイエンス研究部

(Received 2013/11/27　Revised 2014/3/27　Accepted 2014/5/26)

概　要

It is known that some of number puzzles can be solved by using Gröbner bases. In this paper, we discuss methods for solving four kinds of number puzzles (Sudoku, Kakkuro, Hashiwokakero and Tile Paint) by using Gröbner bases. Especially, we study the following question. Of what kind of difficulty can Gröbner bases computation programs solve number puzzles? In order to give some answers of this question, we analyze data of the computation experiments. Moreover, in this paper, we introduce efficient techniques for solving number puzzles. Number puzzles have a lot of hidden mathematics rules and useful properties. We introduce these rules and properties, and describe how to translate these ones into algebraic systems of polynomial equations. In general, the Gröbner bases technique for solving number puzzles, is not considered as a good solver for number puzzles, but we show that the Gröbner bases technique can be a good solver for one of the four puzzles (Tile Paint).

1　はじめに

先行研究により，いくつかのパズルはグレブナー基底を用いることによって，解くことが可能であることが知られている [1, 2, 3, 4, 7, 9]．しかしながら，それら先行研究では，パズルを数式にモデル化する議論に終始し，グレブナー基底計算の解法によって“どの程度のパズルを解くことが可能であるか？”が議論されていないと共に，実験データが与えられていない．そこで，本稿の 1 つ目の目的は，「グレブナー基底計算プログラムはどの程度の難易度のパズルを解くことができるか？」を調べることである．実際，本研究では，計算機代数システム Risa/Asir[18] を用いて，多くの計算実験を行ったので，そのデータを使い，現在の計算機ではどの程度の問題ま

*nabeshima@tokushima-u.ac.jp

で解くことが可能であるかを示す．本稿でのすべての計算実験は，Risa/Asir のグレブナー基底計算プログラムを使用した．本稿で紹介する解法は，グレブナー基底計算プログラムを持つ他の計算機代数システムにおいても可能である．本稿で提示されるデータは，問題の程度を計るための 1 つの目安である．

2 つ目の目的は，「先行研究で紹介されていないパズルを紹介する」ことである．先行研究で議論されたパズル以外にも，グレブナー基底を用いることによって解くことができるパズルは存在する．そのようなパズルについてモデル化とその解法を示し，実験データを提示する．

3 つ目の目的は，パズルをグレブナー基底を用いて解くための「効率的テクニックを紹介する」ことである．パズルのルールのみでなく，パズル自体が持つ良い性質をモデル化することにより，グレブナー基底計算を効率的に行うテクニックについて議論する．本稿の目的をまとめると，次の 3 つである．

(1) グレブナー基底を用いることで，どの程度の難易度の問題を解くことが可能であるかを調べる．
(2) 論文 [1, 3, 4] などで紹介されているパズル以外で，グレブナー基底を用いて解くことが可能である具体的なパズルを紹介する．
(3) パズルをグレブナー基底を用いて解くための効率的なテクニックを紹介する．

グレブナー基底をパズルの解法として用いることは，漠然と計算量の問題から現実的ではないと考えられている [3, 7]．しかしながら，本稿で初めて紹介するパズル『タイルペイント』は効率的なテクニックを用いることにより，ある程度は，解くことが可能であることが，本稿で述べられる．すべてのパズルが駄目というわけではない．

本稿では，著者たちが各問題に対して分担して計算実験を行ったため，問題によって計算機が違う．本稿で使用した計算機は以下の 2 台である．

PC1: PC ［OS:Windows 7 Professional (64bit)，CPU: Intel(R) Xeon (R) X560@ 2.67GHz 2.66GHz，RAM:48GB］
PC2: PC ［OS:Windows 7 Home Premium(64bit)，CPU: Intel(R) Core (TM) i7-2600 CPU@ 3.40GHz 3.40GHz，RAM:4GB］

本稿での time（計算時間）は CPU 時間であり単位は秒（seconds）である．また，$> 2h$，$> 3h$，$> 5h$ は 2 時間，3 時間，5 時間計算しても答えが返ってこなかったことを意味する．本稿で使用した Risa/Asir は version 20121217 （Kobe Distribution）である．

パズルをグレブナー基底を用いて解くためには，パズルを多項式にモデル化する必要がある．まず，解きたいパズルのマスなどに変数 x_i を設定する．その後，パズルの問題が各パズルのルールのもと，完璧に多項式へモデル化でき，かつ，唯一の解を持つとき，論文 [3] の Proposition 3 が同様に成り立つ．論文 [3] の Proposition 3 を本稿では，唯一の解を持つときの判定基準とする．

判定基準

モデル化して得られるいくつかの多項式から生成されるイデアルを I，設定された変数を $x_1, \ldots, x_n$ とする．このとき，任意の項順序に関して I の簡約グレブナー基底の形はいつも $\{x_i - a_i | i = 1, \ldots, n\}$ である．ただし， $a_i \in \{$ 各パズルで使う要素 $\}$ である．

グレブナー基底の計算スピードは，項順序に左右されることが知られている．本稿では， 断らない限り， 計算スピードが一般的に速いと言われる全次数逆辞書式項順序を使用する．

2 ラテン方格系パズル

ラテン方格とは n 行 n 列の表に n 個の異なる “記号” を，各記号が各行および各列に 1 回だけ現れるように並べるパズルである．ラテン方格から派生したパズルとして『数独』,『カックロ』,『因子の部屋』,『賢くなるパズル』などがある．ここではこのようなパズルをラテン方格系パズルと呼び， ラテン方格系パズルについて考える．

グレブナー基底を用いてパズルを解くとは，パズルを数式に翻訳（モデル化）し，その数式からなる連立方程式を解くことである．本章で扱うパズル『数独』,『カックロ』でのラテン方格ルールは次のように数式にモデル化することができる．本章では， パズルに存在するすべてのマスに変数 x_i を対応させる（添え字 i は 1 からマスの総数までの各数字からなる）．また， 1 から n までの異なる数字が各行と各列に 1 回だけ現れるとする．

ラテン方格ルールのモデル化

1. 変数 x_i において， 多項式 $F_i := \prod_{j=1}^{n} (x_i - j)$ を定義する．
2. 各行および各列（または指定された範囲）に 1 から n までの異なる数字が現れるため， その行や列にある 2 変数 x_i，x_j から，多項式 $G_{ij} := \dfrac{F_i - F_j}{x_i - x_j}$ を定義する．

ラテン方格ルールのモデル化は論文 [3] で紹介されている．

2.1 数独

ここで扱う数独（すうどく）とは， 3×3 のブロックに区切られた 9×9 の正方形の枠内に 1 から 9 までの数字を入れるパズルである．数独をグレブナー基底を用いて解く方法は論文 [1, 2, 3, 5, 6, 7, 16] などで紹介されている．論文 [5, 6, 7] では， ここで用いるグレブナー基底（Risa/Asir のコマンド `dp_gr_main` ）とは異なるブーリアン・グレブナー基底という特殊なグレブナー基底を用いる方法であるので， ここではブーリアン・グレブナー基底を用いた解法は採用せず， 論文 [1, 3, 16] の解法を採用し計算実験として扱う．数独は世界的に有名なパズルであり，前述の論文を含め，多くの議論がなされているので，ここでは深入りせず Risa/Asir のコマンド `dp_gr_main` の実験データのみを与える．

数独を方程式にモデル化する方法は前述した**「ラテン方格ルールのモデル化」**の後， 変数 x として割り当てた正方形の枠内に，すでに数字 j が決まっているなら $x - j$ とする．このように，モデル化して得られた多項式からなるイデアルのグレブナー基底を計算することで数独を解くことが可能である．（詳しくは論文 [3] を参照.）

すでにグレブナー基底を使った解法を試みた研究者からの意見として， この解法は “実用的

でない”，“グレブナー基底計算がいつまで経っても終わらない” ということを聞く．しかしながら，明確な実験データがない．ここでは，我々が実際に計算実験した結果を与える．実験に使用した問題は，インターネット上にある**「数独無料ゲーム – 数独問題集」**[1)]を使用した．ここには初級，中級，上級と問題が難易度別に分けられており，各難易度ごとに 400 問の問題が存在する．すべての難易度に対して 30 問グレブナー基底を用いた解法を試みた．次の表 1 では，各難易度から「数独無料ゲーム – 数独問題集」の問題番号 1 から 10 の結果を表す．（問題番号 11 から 30 の結果も同じような結果となっている．）これによりこの解法（`dp_gr_main`）がどの程度の問題を解くことが可能であるか予想できるであろう．表 1 にある「配置数」とは数独の問題に数字が何個配置されているかを表している．ここで使用した計算機は PC2 である．我々はグレブナー基底を用いた数独ソルバーを Risa/Asir 上に実装し計算実験を行った．解が得られるまでの時間を表したものが表 1 である．

初級		
問	配置数	time
1	45	14.04
2	45	15.24
3	43	15.72
4	42	17.42
5	42	15.99
6	46	15.88
7	44	16.44
8	43	15.9
9	42	16.5
10	44	15.49

中級		
問	配置数	time
1	17	$> 2h$
2	17	$> 2h$
3	17	$> 2h$
4	25	3971
5	17	$> 2h$
6	26	$> 2h$
7	24	$> 2h$
8	17	$> 2h$
9	17	$> 2h$
10	17	$> 2h$

上級		
問	配置数	time
1	17	$> 2h$
2	17	$> 2h$
3	17	$> 2h$
4	18	$> 2h$
5	18	$> 2h$
6	25	$> 2h$
7	17	$> 2h$
8	17	$> 2h$
9	17	$> 2h$
10	26	7.644

表 1: 数独の計算時間

実験の結果から，初級レベルの問題であれば，現実的な時間で解くことが可能であるが，それ以外では，この解法は適さないことが分かる．

2.2 カックロ

カックロとは足し算を利用したラテン方格系パズルである．カックロのルールは次である．

カックロのルール

(1) すべての白マスに 1 から 9 までの数字のどれかを 1 つずつ入れる．

(2) 斜めの線の右上の数字は，その右に連続した白マスに入る数字の合計を表し，左下の数字は，その下に連続した白マスに入る数字の合計を表す．

(3) 縦・横への 1 つの白マスのつながりには，同じ数字を入れない．

[1)] `http://www.sudokugame.org/`

次の図 1 はカックロのサンプル問題としてニコリ [10] や多くの雑誌，Web サイトで広く公開されているものである．この図 1 をグレブナー基底を用いて解くためには，まず図 2 のように，各白マスに変数を設定する．次に，各変数について，ラテン方格ルールのモデル化 1 を行い，斜めの線の右上の数字から右に連続した変数を 1 つの範囲とみなし，ラテン方格ルールのモデル化 2 を行う．同様に，斜めの線の左下の数字から下に連続した変数を 1 つの範囲とみなし，ラテン方格ルールのモデル化 2 を行う．このようなモデル化は，カックロのルール (1)，(3) を意味している．カックロのルール (2) を満足するために，斜めの線の右上の数字は，その右に連続した変数の和とし，左下の数字は，その下に連続した変数の和とする．これらのモデル化によって得られた多項式により生成されるイデアルのグレブナー基底を任意の項順序で計算することによって，カックロのパズルは解かれる．このカックロをグレブナー基底を用いて解く方法は論文 [3] で紹介されている．

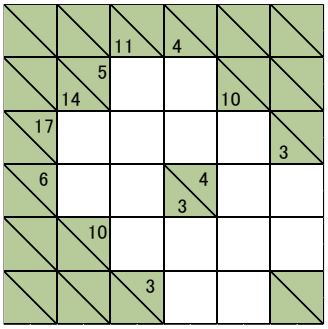


図 1: カックロサンプル問題

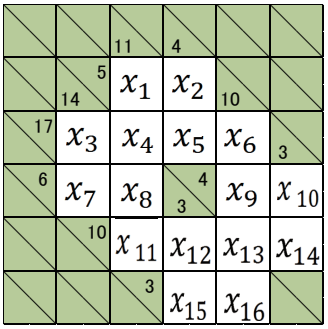


図 2: 変数の設定

我々は，カックロを解くための簡単なプログラムを Risa/Asir 上に実装した．次は，実際，図 2 の情報を，我々のプログラムの関数に入力したものと，その出力である．このときに使用した項順序は全次数逆辞書式項順序である．入力はリストでその中身の成分は，またリストであり［斜めの線の右上もしくは左下の数字，［数字から連続した変数］］ を表している．

```
kakuro([[5,[x1,x2]],[17,[x3,x4,x5,x6]],[6,[x7,x8]],[4,[x9,x10]],[10,[x11,x
12,x13,x14]],[3,[x15,x16]],[14,[x3,x7]],[11,[x1,x4,x8,x11]],[4,[x2,x5]],[3
,[x12,x15]],[10,[x6,x9,x13,x16]],[3,[x10,x14]]]);
[x5-1,-x16+1,x15-2,-x14+2,x13-4,-x12+1,-x11+3,x10-1,-x9+3,-x8+1,-x6+2,-x4+
5,-x2+3,x1-2,x7-5,-x3+9]
```

これより，図 2 のパズルの答えは $x_1 = 2$，$x_2 = 3$，$x_3 = 9$，$x_4 = 5$，$x_5 = 1$，$x_6 = 2$，$x_7 = 5$，$x_8 = 1$，$x_9 = 3$，$x_{10} = 1$，$x_{11} = 3$，$x_{12} = 1$，$x_{13} = 4$，$x_{14} = 2$，$x_{15} = 2$，$x_{16} = 1$ となる．

論文 [3] では，グレブナー基底を用いることにより，カックロを解くことが可能であることが述べられているが，しかし，論文 [3] に 1 問だけ掲載されているカックロの問題さえ計算されていない．また，どの程度の問題を解くことが可能であるのかも述べられていない．そこで，カックロの問題をグレブナー基底を用いた解法で計算実験した結果が表 2 である．

実験で使用した問題は，論文 [3] に掲載されたパズル 1 問，Wikipedia のカックロのページで紹介されているパズル 1 問[2)]，パズル作家 “E 坂もるか” 氏のホームページ EPSILON DELTA[3)]で公開しているカックロ初級問題 2 問，（株）ニコリのホームページで公開されている，カックロお試し問題[4)]“らくらく” レベルの 4 問である．表 2 にある「大きさ」とは，問題の大きさを意味し，数字の入ったマスも含めて（縦のマス数）×（横のマス数）を表している．例えば，図 1 の問題の大きさは 6×6 である．ここで使用した計算機は PC1 である．

問題	大きさ	変数の数	time
図 1	6×6	16	31.64
論文 [3] にあるパズル	8×8	36	4098
Wikipedia カックロ (2007/10/25)	8×8	36	$> 5h$
"E 坂もるか" 初級 1	9×9	44	1668
"E 坂もるか" 初級 2	9×9	48	$> 5h$
ニコリおためし問題 1（らくらく）	9×11	73	$> 5h$
ニコリおためし問題 2（らくらく）	9×11	73	$> 5h$
ニコリおためし問題 3（らくらく）	9×11	73	$> 5h$
ニコリおためし問題 4（らくらく）	9×11	70	$> 5h$

表 2: カックロの計算時間

計算実験の結果から，グレブナー基底を用いた解法は，カックロには適さないことが分かる．理論的には解を求めることが可能であるが，現実問題としてカックロには，グレブナー基底の解法は不向きである．また，計算実験で使用した問題はすべて初級レベルの問題であり，カックロの中でも小さなサイズの問題である．このような易しく，かつ，小さな問題に対してすら解が得られないことより，カックロの解法として，この方法はのぞましくない．この実験より，グレブナー基底を用いて解くことができるカックロの現実的な大きさは 6×6 である．

カックロは数独と比べ制約条件の数が少ない．数独での 1 つのマスは縦・横・ブロックの 3 つで制約され，それぞれに対して同じ数字が重ならないようにしなければならない．1 から 9 の 9 個の数字が常に縦・横・ブロックに存在するのだからその分，制約条件が多いと考えられる．しかしながら，カックロの場合，1 つのマスには，まず，縦・横の 2 つの制約に対してのみ同じ数が重ならないようにしなければならない．次に，斜めの線に右上もしくは左下の数字の制約として縦と横の 2 つの和の制約が付加されるだけである．解となる数字は 1 から 9 の数字であるが，これらすべての数字が縦・横に存在するわけではない．このことを考えると，カックロの制約条件数は数独より少なく，解の決定には数独より時間が掛かると予想される．また，数独と違い問題の大きさが固定されておらず，中級レベルの問題になると（大きさが 19×11），マス

[2)] http://upload.wikimedia.org/wikipedia/commons/c/c8/Kakuro_black_box.svg
Wikipedia のカックロのページ，2007 年 10 月 25 日（木）06:50

[3)] http://www20.big.or.jp/~morm-e/

[4)] http://www.nikoli.com/ja/puzzles/kakuro/

の数も 100 を超えることより，なおさらグレブナー基底計算には適さないパズルである．

2.3 ラテン方格系パズルの注意とまとめ

計算実験の結果から，ラテン方格系のパズルはグレブナー基底を用いた解法には適さないことが分かる．これは，設定する変数の数が多いことと，各変数のみからなる多項式の次数が 9 と大きいことから，グレブナー基底計算が困難になるためであると考える．唯一，グレブナー基底を用いる解法の良い点は，グレブナー基底計算の組み込み関数さえ計算機代数システムに存在すれば，パズルを解くためのプログラムを容易に作ることができることである．難しいパズルを解くための複雑なアルゴリズムを考える必要は無く，組み込みのグレブナー基底計算プログラムに任せるだけでよい．変数が少ない問題であればこのプログラムで解くことができるかもしれない．

ここでは，9×9 の数独を扱ったが，4×4 の数独の場合，ここで使用した解法で十分対応可能である．この場合については，論文 [16] や [17] に書かれている．また，数独に関しては「要素」としての数字のみを扱っていることから，集合制約問題と考えることができる．このような集合制約問題を解く 1 つの方法として，ブーリアン・グレブナー基底 [20] を使う方法がある．このブーリアン・グレブナー基底を用いる解法では 9×9 の問題を現実的な時間で（1 分もあれば）解くことが可能である．これは論文 [5, 6, 7, 16] に述べられている．カックロは「要素としての数字」と「大きさ（量）としての数字」の異なる 2 つの性質を利用したパズルであることから，数独より更に難しいと考えられる．

他のラテン方格系のパズルとして，『因子の部屋』，『賢くなるパズル』などが存在する．論文 [4] では，『賢くなるパズル』が扱われているが，他の論文同様に，計算データが示されていない．これら他のパズルについて，本稿では計算実験を行っていないが，カックロ同様に「要素としての数字」と「大きさ（量）としての数字」の異なる 2 つの性質を利用したパズルであることから，カックロと同様な結果になると予想される．他に知られているものとして，本来のラテン方格を解く方法として，トーリックイデアルのグレブナー基底を計算する方法がある．これは，[8, 9] で紹介されているが，莫大な計算時間を要する．

3 橋をかけろ

本章では，パズル『橋をかけろ』[5)]について考える．パズル『橋をかけろ』は数字を島に見立て，橋に見立てた線を引くパズルである．

橋をかけろのルール

(1) ○同士を線で結び，すべての数字が線でつながっているようにする．
(2) 線は水平方向か垂直方向に引く．
(3) どの○の間にも 2 本までしか線は引けない．
(4) ○の中の数字はその○から引かれる線の数を表す．
(5) 線は他の線と交差したり数字を横切ったりしない．

[5)]『橋をかけろ』は，1990 年にパズル通信ニコリにてパズル作家 “れーにん” 氏によって発表されたパズルである．発表後，人気を博し，パズル通信ニコリに掲載されつづけると共に，単行本も出版されている．

図 3 はパズル『橋をかけろ』の簡単な例であり，図 4 はその解答である．

論文 [3] において，グレブナー基底を用いてパズル『橋をかけろ』を解く方法が紹介されている．ここでは，モデル化の “問題点を指摘する” と共に，より “効率的に解く方法を紹介する”．また，論文 [3] で成されていない計算実験のデータを与える．これにより，現在の計算機ではどの程度のレベルまでグレブナー基底を用いて解くことが可能であるか分かる．

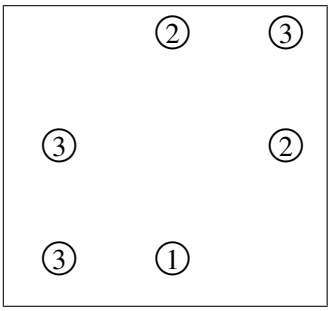


図 3: サンプル問題

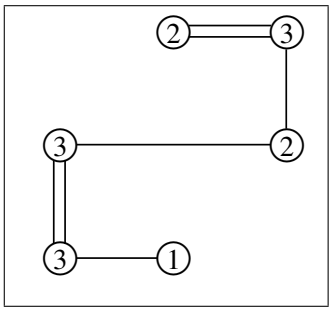


図 4: 図 3 の解答

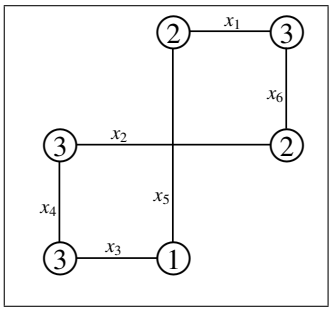


図 5: 変数の設定

3.1 モデル化の手法

パズル『橋をかけろ』を数式にモデル化するため，論文 [3] でのモデル化の方法を採用する．パズル『橋をかけろ』は次のようにすれば，数式に変換される．

パズルのモデル化

1. ルール (2) より線（橋）を引く可能性のある場所すべてに変数を設定する．（水平方向，垂直方向のみ）
2. ルール (3) から各変数のとる値は 0，1，2（本）のいずれかである．x をモデル化 **1** で設定された変数とすると，式 $x(x-1)(x-2)=0$ を作る．
3. ルール (4) より○から出ている線の変数をすべて足した値が○の中の値とする．
4. ルール (5) より線は交差してはいけないので，交差する線に対応する変数同士の積が 0 となるようにする．

例として図 3 のパズルを考える．線を引く可能性のあるところすべてに，図 5 のように変数を設定する．ルール (3) より各変数 x_i の値が 0，1，2 であればよいので，数式 $x_i(x_i-1)(x_i-2)=0$ を作る．ただし，$i=1,\ldots,6$ である．次に，ルール (4) より○の中の数字が引かれている線の本数にならなければならないので，その○から出ている変数の合計が○の値になるようにする．すなわち，多項式 $x_1+x_5-2=0$，$x_1+x_6-3=0$，$x_2+x_6-2=0$，$x_2+x_4-3=0$，$x_3+x_4-3=0$，$x_3+x_5-1=0$ を作る．ルール (5) より線は交差してはいけない．したがって，交差している x_2 と x_5 はどちらかが 0 になるので，その積を考え $x_2x_5=0$ という数式を作る．このモデル化により 6 変数の連立方程式が得られる．この連立方程式の解をグレブナー基底を用いて解くと，解は唯一となり図 4 となる．すなわち，図 3 の答えは，$x_1=2$，$x_2=1$，$x_3=1$，$x_4=2$，$x_5=0$，

$x_6 = 1$ となる．

ここで， パズル『橋をかけろ』のルールを再確認する．数式へのモデル化の手法において，ルール (2)，(3)，(4)，(5) が用いられていることが確認できる．しかしながら， ルール (1) が上で紹介したモデル化の手法においては確認できない．したがって， このモデル化はルール (1) を無視した “不完全なモデル化” である．本来のパズルの解が唯一であるとしても，ルール (1) を満たさず，グラフが孤立成分を持つような不適切な解が出てくる可能性がある．図 6 は不適切な解となる場合の簡単な例であり，その真の解は図 7 である．

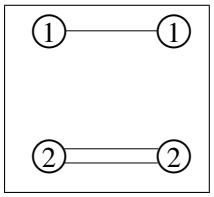


図 6: 不適切な解

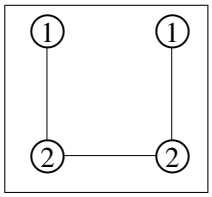


図 7: 真の解

したがって，この手法では唯一の解を決めることができない．

問題点：ルール (1) がモデル化できず， この方法では解けない．

モデル化が不完全なため，全次数の項順序では，直接にグレブナー基底それ自体からは，連立方程式の解を得ることができない．直接グレブナー基底から連立方程式の解をすべて求めるには，辞書式項順序を用いる必要がある．（直接的でなければ連立方程式を解く方法として，最小多項式を計算する方法などがある．しかし，ここでは，論文 [3] に従い計算されたグレブナー基底そのものから直接，解を得る方法を考える．）

我々は， この問題点を解決すべく，ルール (1) のモデル化を試みたが残念ながら良いモデル化の手法が見つからなかったので，ルールを緩和したものを考える．

3.2 ルールの緩和と効率的テクニック

サブセクション 3.1 で述べたモデル化の手法では，複数個の解が存在する可能性があり，パズルを解いたことにはならない．しかしながら，その複数個の解の中には真の解が必ず存在する．複数個の解を得た後で人間が真の解をその中から探すことは可能である．そこで， ここではルール (1) を除外したものを “基本的には” 考えるが， 次の図 8 の特別な場合を含む解は除くものとする．もし， 1 つの問題に○（島）が 3 個以上あれば，図 8 は，明らかに他とは連結しないので，真の解とはならない．

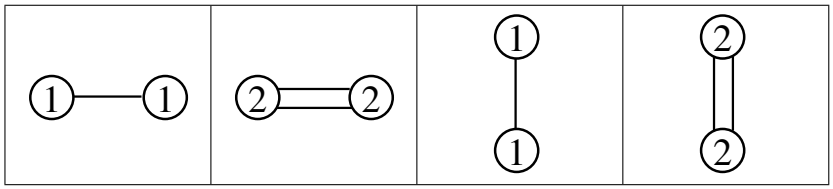


図 8: 除外ルール

図 8 の状態はモデル化の段階で，表 3 にある除外ルールのモデル化のテクニックを使用すれば，すぐに除外可能である．除外可能な図 8 は解から特別に除くこととし，ルール (1) とこの除外ルールを交換し，改めてこの除外ルールを (1)’ とする．

設定	②……x……②	①と①の間の橋
制約	数字の 2 があり，ある方向にまた 2 がある場合はそこには 2 本橋をかけない．	数字の 1 があり，ある方向にまた 1 がある場合はそこには橋をかけない．
数式	$x(x-1)=0$	変数を設定しない．

表 3: 除外ルールのモデル化のテクニック

パズル『橋をかけろ』のルールを次のように緩和したものを次から考える．

橋をかけろのルール

(1)’ 図 8 の部分は含まない．
(2) 線は水平方向か垂直方向に引く．
(3) どの数の間にも 2 本までしか線は引けない．
(4) ○の中の数字はその○から引かれる線の数を表す．
(5) 線は他の線と交差したり数字を横切ったりしない．

基本的に，本来のルール (1) を除外したものを考えているので，パズルの問題によっては複数の解が存在する．すべての解は連立方程式から求められる．

サブセクション 3.1 のモデル化では問題が大きくなるにつれ，グレブナー基底の計算量が多くなる．効率よく計算するためには，パズルの問題から更なる数学的情報を引き出す必要がある．表 4 と表 5 は，人間がパズル『橋をかけろ』を解くときに使うテクニックである．これをモデル化することによって計算の効率化を計る．ここでは，このテクニックを人間的テクニックと呼ぶことにする．（$i=1,2,3,4$，$j=1,2,3$，$k=1,2$ とする．）

問題を数式にモデル化し，その連立方程式のすべての解を辞書式項順序を用いたグレブナー基底を用いて解く方法が，本稿の『橋をかけろ』の解法である．我々は，紹介した人間的テクニックが無い場合と， テクニックが有る場合の 2 つの解法を数式処理ソフト Risa/Asir に実装した．表 6 は，この 2 つの解法の比較であると共に， どの程度の問題をこの解法を用いて解くことができるかの目安である．ここで扱う問題は株式会社ニコリから出版されている『橋をかけろ』専門のパズル本「橋をかけろ 1」[11] からである．表にある問題の番号は，パズル本 [11] の問題の番号を意味し， 大きさは問題を縦横 1 マスずつの格子状のマス目（または行列）で表わした時の（縦マスの個数）×（横マスの個数）を意味する．

我々は，問題 1 から 52 までを計算実験した．問題 1 から 19 は初級（easy）レベルであり易しく大きさが小さな問題である．これら初級（easy）レベルの問題は，すべて 3 秒以内で解くこ

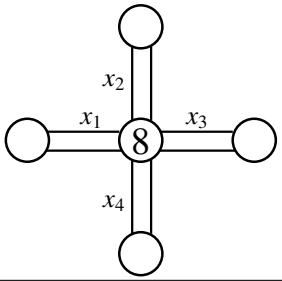
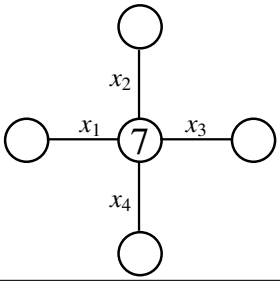
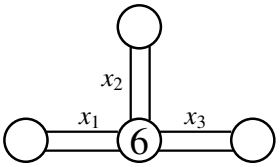
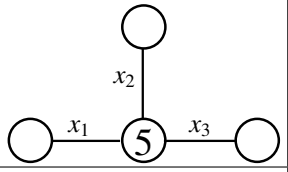
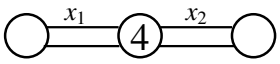
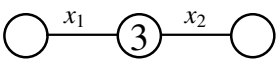
設定 (数値)			
制約	数字の 8 があると必ず 4 方向に 2 本ずつ橋をかけなければならない.	数字の 7 があると必ず 4 方向に少なくとも 1 本ずつ橋をかけなければならない.	数字の 6 があり 3 方向にのみ橋をかける可能性がある場合は 3 方向に 2 本ずつ橋をかけなければならない.
数式	$x_i - 2 = 0$	$(x_i - 2)(x_i - 1) = 0$	$x_j - 2 = 0$
設定 (数値)			
制約	数字の 5 があり 3 方向にのみ橋をかける可能性がある場合は 3 方向に少なくとも 1 本ずつ橋をかけなければならない.	数字の 4 があり 2 方向にのみ橋をかける可能性がある場合は 2 方向に 2 本ずつ橋をかけなければならない.	数字の 3 があり 2 方向にのみ橋をかける可能性がある場合は 2 方向に少なくとも 1 本ずつ橋をかけなければならない.
数式	$(x_j - 2)(x_j - 1) = 0$	$x_k - 2 = 0$	$(x_k - 2)(x_k - 1) = 0$

表 4: 人間的テクニック I

とができ，似たような結果となったので初級レベル 1，17，18 の 3 つの結果のみを表 6 に記す．ここで使用した計算機は PC2 である．

人間的テクニックが無い場合， 問題の大きさが 9×9 までであれば “現実的な時間（3 時間以下）” で解を得ることが可能であるが，問題の大きさが 1 つ大きくなると，解を得ることが難しくなる．それに比べ人間的テクニックが有る解法では，9×16 までであれば多くの問題で解を得ることができる．これは， テクニックにより必要となる変数の個数が減少したことと， 制約条件（多項式の個数）が多くなったことからグレブナー基底計算時間が短縮されたと考えられる．

3.3 『橋をかけろ』のまとめ

簡単な人間的テクニックを使うことで， グレブナー基底を用いた解法の効率は格段に良くなることが計算実験より分かる．パズルのルールのみに着目しモデル化するのではなく， パズルの持つ性質を考慮に入れモデル化することが，グレブナー基底を用いた解法では大切である．大きさ 9×9 の『橋をかけろ』は難易度レベル初級（easy）の問題である．また， 大きさ 9×16 は

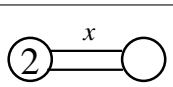
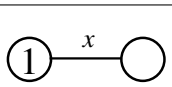
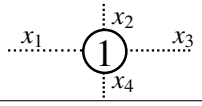
設定 (数値)			
制約	数字の 2 があり 1 方向にのみ橋をかける可能性がある場合はそこに 2 本橋をかけなければならない.	数字の 1 があり 1 方向にのみ橋をかける可能性がある場合はそこに 1 本橋をかけなければならない.	数字の 1 からある方向 (1 から 4 方向) に変数を設定する可能性がある場合はその変数は 2 をとらない.
数式	$x-2=0$	$x-1=0$	$x_i(x_i-1)=0$

表 5: 人間的テクニック II

難易度レベル初級（easy）から中級（medium）の問題である．人間的テクニックを使うことによって，この解法では，中級レベルの問題まで解くことが可能である．しかし， 問題が大きくなる上級レベル（22×13）では，この解法は適さないことが分かる．問題が大きくなることによって， 変数の数が多くなりグレブナー基底の計算量が莫大に増えるからである．

パズル『橋をかけろ』がラテン方格系パズルより，広く解くことが可能である原因は， 各変数のみからなる多項式の次数が 3 と小さいからだと考えられる．このことから， グレブナー基底を用いた解法が適するパズルは， 「多項式の次数を小さくモデル化できる」,「制約条件が多い」,「パズルの性質から簡単な解法のためのテクニックが導き出せる」ものだと考えられる．これらの条件が当てはまるパズルが次のパズル『タイルペイント』である．

4 タイルペイント

本章では， パズル『タイルペイント』について考える．

タイルペイントのルール

(1) 盤面上にある， 太線で区切られた部分（タイルと呼ぶ）のいくつかを黒く塗る．
(2) 盤面の数字は， その右あるいは下の， 外周か次の斜線のマスまでの区切られた 1 列（または行）のうちで黒く塗られるマスの数を表す．
(3) どのタイルも， すべてのマスを塗るか， もしくはすべてのマスを塗らずにおくかのどちらかとする．タイルの一部のマスだけを塗ることはできない．

パズル『タイルペイント』[6]は，各グループ化されたマス（□）の組（タイルと呼ぶ）をすべて塗る， もしくはすべて塗らないというルールのもと， 最終的に絵が出てくるパズルである．図 9 は， ニコリの web ページ [10] やニコリ本誌 [12, 13, 14, 15] などで公開されているパズル『タイルペイント』のサンプル問題であり， 図 10 はその解答を表している．

パズル『タイルペイント』も，グレブナー基底を用いて解くことが可能である．本章では， グレブナー基底を用いた『タイルペイント』の解法を紹介し， その実験データを与える．この計算により， 現在の計算機では，どの程度の問題までグレブナー基底を用い解くことが可能であるか考察する．

[6]パズル作家 “ヤンマー” 氏により発案されニコリ本誌 53 号から掲載されている人気のパズルである．

		テクニック無し			テクニック有り		
問題	size	変数	解	time	変数	解	time
1	9×9	40	1	0.078	30	1	0.016
17	9×9	40	1	0.14	37	1	0.047
18	9×9	46	1	0.437	39	1	0.031
19	9×9	48	5	2.106	46	2	0.109
20	16×9	69	1	89.34	61	1	0.047
21	16×9	61	3	55.96	57	3	0.109
22	16×9	73	–	$>3h$	67	1	32.12
23	16×9	76	–	$>3h$	68	1	0.343
24	16×9	77	–	$>3h$	70	1	0.484
25	16×9	73	–	$>3h$	67	2	2.761
26	16×9	87	–	$>3h$	83	2	158
27	16×9	71	1	$>3h$	66	1	0.125
28	16×9	73	1	$>3h$	66	1	0.14
29	16×9	79	–	$>3h$	67	2	0.109
30	16×9	71	–	$>3h$	69	1	88.14
31	16×9	91	–	$>3h$	87	1	129.8
32	16×9	87	–	$>3h$	85	–	$>3h$
33	16×9	59	1	$>3h$	52	4	0.047
34	16×9	91	–	$>3h$	89	1	$>3h$
35	16×9	83	–	$>3h$	77	3	2.2
36	16×9	63	1	$>3h$	63	1	1.825
37	16×9	67	1	$>3h$	66	1	0.265
38	16×9	71	–	$>3h$	66	2	2.964
39	16×9	63	9	$>3h$	61	9	0.14
40	16×9	67	2	$>3h$	65	2	0.39
41	16×9	75	–	$>3h$	71	1	0.14
42	16×9	77	–	$>3h$	77	–	$>3h$
43	16×9	101	–	$>3h$	101	–	$>3h$
44	16×9	107	–	$>3h$	105	–	$>3h$
45	16×9	81	–	$>3h$	80	1	1354
46	16×9	85	–	$>3h$	78	3	6.443
47	16×9	87	–	$>3h$	84	–	$>3h$
48	16×9	75	–	$>3h$	74	1	$>3h$
49	22×13	125	–	$>3h$	113	1	163.4
50	22×13	141	–	$>3h$	132	–	$>3h$
51	22×13	134	–	$>3h$	129	–	$>3h$
52	22×13	145	–	$>3h$	134	–	$>3h$

表 6: 『橋をかけろ』の計算結果

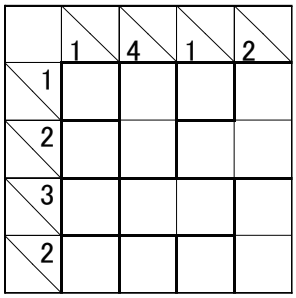


図 9: タイルペイントの問題

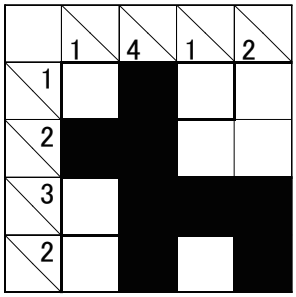


図 10: 図 9 の解答

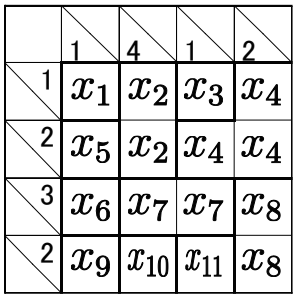


図 11: 変数設定

4.1 モデル化の手法と解法

パズル『タイルペイント』を数式にモデル化する．盤面の数字は，その右あるいは下の，外周か次の斜線のマスまでの区切られた 1 列（または行）のうちで黒く塗られるマスの数を表しているので，本稿では，1 つのマスを “黒く塗る” ことをマスに “1 を入れる” こととし，1 つのマスを “黒く塗らない” ことをマスに “0 を入れる” こととする．この考えのもと，パズル『タイルペイント』の問題は次のようにモデル化することが可能である．

パズルのモデル化

1. 各タイル（太線で区切られた部分）に変数を割り当てる．同じタイル内の各マスには割り当てられた同じ変数を置く．
2. マスには 0 か 1 が入る．すなわち, x が上のモデル化 **1** で設定された変数とすると $x(x-1) = 0$ とする．
3. 盤上の数字は， その右あるいは下の，外周か次の斜線のマスまでの区切られた 1 列（または行）のマスの和とする．

例として図 9 の問題を考える．タイルに変数を設定し，各マスにその変数を振り分けたものが図 11 である．モデル化の **2** より数式 $x_i(x_i - 1) = 0$ が必要となる．ただし， $i = 1, \ldots, 11$ である．次に盤上の数字は右あるいは下の，外周までの区切られた 1 列（または行）の和より， 縦方向の和として数式 $x_1 + x_5 + x_6 + x_9 = 1$，$2x_2 + x_7 + x_{10} = 4$，$x_3 + x_4 + x_7 + x_{11} = 1$，$2x_4 + 2x_8 = 2$ を得る．また，横方向の和として数式 $x_1 + x_2 + x_3 + x_4 = 1$，$x_5 + x_2 + 2x_4 = 2$，$x_6 + 2x_7 + x_8 = 3$，$x_9 + x_{10} + x_{11} + x_8 = 2$ を得る．得られた連立方程式の解が，問題の答えである．実際， 得られた多項式から生成されるイデアルのグレブナー基底を Risa/Asir で計算すると次となる．

```
[113] gr([x1+x5+x6+x9-1,2*x2+x7+x10-4,x3+x4+x7+x11-1,2*x4+2*x8-2,x1+x2+x3+
x4-1, x5+x2+2*x4-2,x6+2*x7+x8-3,x9+x10+x11+x8-2,x11^2-x11,x10^2-x10,x9^2-x
9,x8^2-x8,x7^2-x7,x6^2-x6,x5^2-x5,x4^2-x4,x3^2-x3,x2^2-x2,x1^2-x1],[x1,x2,
x3,x4,x5,x6,x7,x8,x9,x10,x11],2);
[x11,x10-1,-x9,-x8+1,-x7+1,-x6,-x5+1,x4,-x3,-x2+1,x1]
```

これにより， $x_1 = x_3 = x_4 = x_6 = x_9 = x_{11} = 0$， $x_2 = x_5 = x_7 = x_8 = x_{10} = 1$ が解となる．

ここでは，グレブナー基底の計算において辞書式項順序を利用したが，真のパズル『タイルペイント』の問題の解は唯一であり，モデル化はルールを完璧におさえていることから判定基準がなりたつので辞書式項順序の必要はない．パズル『タイルペイント』を考えるときは，全次数逆辞書式項順序を使う．

タイルペイントの解法（**Step 1** から **Step 2**）は

Step 1. 問題を数式にモデル化する．
Step 2. 簡約グレブナー基底を計算する．

となる．

4.2 効率的テクニック

人間がパズル『タイルペイント』を解くとき，まず次のテクニックを使う．今，盤上の 1 つの数字を k とし，その右あるいは下の外周か次の斜線のマスまでの区切られた 1 列（または行）のマスの総数を n とする．このとき，パズルの一部が次のような状態のとき，一部のタイルはすぐに決定される．次の **1**，**2** をパズル『タイルペイント』の人間的テクニックと呼ぶようにする．

1. $n-k=s$ のとき，指定された 1 列（または行）内のタイルの一部のマスの数が s より大きいものがあるとき，そのタイルのすべてのマスを 1 とする．
2. 指定された 1 列（または行）内のタイルの一部のマスの数が k より大きいものがあるとき，そのタイルのすべてのマスを 0 とする．

ここで，**1**，**2** において $k=0$，$n=k$ のときは，明らかにすべてが 0 または 1 となるときである．テクニック **1** での $n-k=s$ のときでは，タイルの一部のマスの数が s より大きいものがあるとき，そのタイルがもし 0 ならば，ルールに矛盾するので 1 しかありえない．また，テクニック **2** も同様に，そのときは 0 しかありえない．

図 12 の上図は問題の一部分を表しており，1 行には 10 マス存在する．数字の 7 の行を見る．$10-7=3$ であり，この行には 4 マスを持つタイルが存在する．$3<4$ なのでこのタイルが 0 ならルールに矛盾するので，このタイルは 1 となる．また，数字の 9 の行を見ると $10-9=1$ となり，同様に 1 より大きいマスを持つタイルは 1 となる．最後に，1 の行を見ると 1 より大きいマスを持つタイルは 0 となる．

図 12 の下図はテクニックを使った結果を表している．図 12 にある ①はタイルの性質から 1 にならなければならないことが分かる．パズル『タイルペイント』を数式へモデル化する際，このテクニックを使いモデル化することは容易なことである．人間的テクニックを有するタイルペイントの解法（**Step 1** から **Step 3**）は

Step 1. 人間的テクニックにより決定されるところに 0，1 を入れる．（モデル化 1）
Step 2. 問題を数式にモデル化する．（モデル化 2）
Step 3. 全次数逆辞書式項順序で簡約グレブナー基底を計算する．

となる．

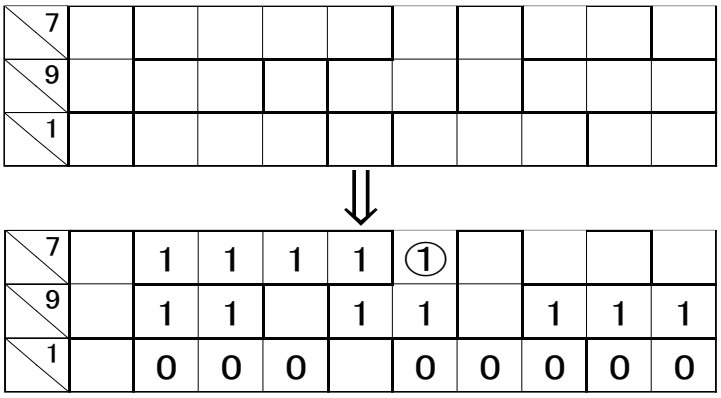


図 12: 人間的テクニックの例

表 7 と表 8 は，紹介した 2 つの解法（人間的テクニックが無い解法と，人間的テクニックを有する解法）で計算実験した結果である．問題は参考文献に上げてあるニコリ本誌と web 上で公開されてあるものを計算実験として使用した．（問題にある［[12] p.49，1，（10 × 10)］は［参考文献　ページ番号，問題番号，縦マスの数 × 横マスの数］を意味する．）表にある「変数の数」とは，モデル化した際に使用した変数の個数を意味し，time は有理数上でのグレブナー基底の計算時間を意味している．ここで使用した計算機は PC1 である．

番号	問題 (大きさ)	変数の数	time
1	図 11 (4 × 4)	11	0.016
2	[19] No.00 (6 × 6)	20	0.032
3	[19] No.01(10 × 10)	49	47.66
4	[19] No.04 (10 × 10)	43	32.78
5	[12] p.49, 1 (10 × 10)	27	0.063
6	[13] p.39, 1 (10 × 10)	42	> 5*h*
7	[14] p.41, 1 (10 × 10)	38	> 5*h*
8	[15] p.71, 1 (10 × 10)	41	401.3
9	[15] p.71, 2 (10 × 10)	39	> 5*h*
10	[19] No.02 (10 × 10)	51	> 5*h*
11	[19] No.03 (10 × 15)	64	> 5*h*
12	[13] p.39, 2 (25 × 25)	199	> 5*h*

表 7: 人間的テクニック無

番号	問題 (大きさ)	変数の数	time
1	図 11 (4 × 4)	8	0.016
2	[19] No.00 (6 × 6)	18	0.016
3	[19] No.01(10 × 10)	33	0.062
4	[19] No.04 (10 × 10)	26	0.031
5	[12] p.49, 1 (10 × 10)	26	0.031
6	[13] p.39, 1 (10 × 10)	34	1.529
7	[14] p.41, 1 (10 × 10)	26	0.062
8	[15] p.71, 1 (10 × 10)	26	0.081
9	[15] p.71, 2 (10 × 10)	31	2.059
10	[19] No.02 (10 × 10)	41	7.925
11	[19] No.03 (10 × 15)	52	9.216
12	[13] p.39, 2 (25 × 25)	144	> 5*h*

表 8: 人間的テクニック有

この実験結果から，人間的テクニックを有する方法は，明らかにテクニックが無い方法より優れていることが分かる．人間的テクニックを使用することで変数の数が減少し，グレブナー基底をより速く求められることが原因であると考える．

実験において，大きさ 10 × 10 程度なら人間的テクニックを有する方法で，解くことが可能であることが分かる．大きさ 10 × 10 の問題は『タイルペイント』では大きくも小さくもない比較的多く見られる一般的な問題である．しかし，番号 12 にある大きさ 25 × 25 は大きい部類の

問題であり，人間的テクニックを有する方法でも 5 時間以内では解くことができない．これは，パズルのサイズが大きくなるにつれ変数の数が増えることが問題となるためである．テクニックを有する方法においても，ニコリ本誌 [12, 13, 14, 15] に掲載されているような 25×25 の問題をグレブナー基底を使って解くことは不可能であると思われる．人間的テクニックを有する方法では， 大きさ 10×10 から 10×15 程度なら解くことができると考えられる．

次に， 再び人間的テクニックを再考する．先に紹介した解法では，人間的テクニックを，各数字に対応する箇所（列， 行）に 1 度のみ使用した．しかしながら， すでに決定したマスの個数を l， 対応する列（または行）全体のマスの個数を m， 1 が決定されているマスの個数を n とすると， まだ決定されていない 1 の個数 $m-n$ と，決定されていない空白マスの個数 $m-l$ に対して， 再度人間的テクニックを使用することは可能である．図 13 がその例である．図 13 は 10 マスのうち 1 が 2 マス入らなければならない問題である．図 13 の上段は人間的テクニックを 1 度使用した状態を表している．ここで， すでに決まったマスは 7 マス存在する．全体の 10 マスから 7 マスを引くと 3 マスがまだ決定されていない．決定したマスを黒く塗りつぶしたものが図 13 の下段である．今， 全体 3 マスのうち 1 が 2 マス入らなければならない問題となった．ここで， もう一度， 人間的テクニックを使用すると図 13 の下段のように全体が決定される．こ

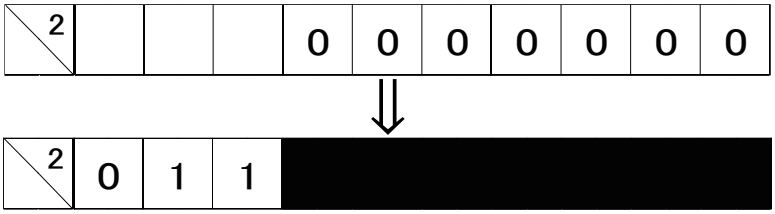


図 13: 人間的テクニックの再帰的使用の例

の性質を考慮すると， テクニック有りの解法において， モデル化 **1** の人間的テクニックを，テクニックが使用不可能になるまで “再帰的” に使用し続けることにより，効率的な解法となると考えられる．この改良を加えたものが次の解法である．これを，**再帰的テクニックを有する解法**（**Step 1** から **Step 3**）と呼ぶ.

Step 1. 人間的テクニックを “再帰的に使用” することで決定されるタイルはすべて決定する．（モデル化 1）

Step 2. 問題を数式にモデル化する．（モデル化 2）

Step 3. 全次数逆辞書式項順序で簡約グレブナー基底を計算する．

表 9 は人間的テクニックを使用しない解法での計算結果を表し，表 10 は再帰的テクニックを有する解法での計算結果である．使用した計算機は前述の計算実験と同じ PC1 であり，time は有理数上でのグレブナー基底計算時間を意味する．問題の大きさは 25×25 のみを扱っている．

表 10 より，再帰的テクニックを有する解法の結果をみると，すべての番号において変数の数は 0 個，time は 0 秒となっている．これは，グレブナー基底計算無しで解が得られていること

番号	問題 (大きさ)	変数の数	time
12	[13] p.39, 2 (25 × 25)	199	> 5*h*
13	[12] p.49, 2 (25 × 25)	129	> 5*h*
14	[14] p.41, 2 (25 × 25)	137	> 5*h*
15	[15] p.71, 3 (25 × 25)	173	> 5*h*

表 9: 人間的テクニックを使用しない解法

番号	問題 (大きさ)	変数の数	time
12	[13] p.39, 2 (25 × 25)	0	0
13	[12] p.49, 2 (25 × 25)	0	0
14	[14] p.41, 2 (25 × 25)	0	0
15	[15] p.71, 3 (25 × 25)	0	0

表 10: 再帰的テクニックを有する解法

を意味する．すなわち， 再帰的に人間的テクニックを使用することのみで解が得られる．人間的テクニックとは数字の大小比較のみで決定される簡単なテクニックでありモデル化も容易である．25 × 25 は大きく難しい部類の問題であるが，使用した問題は大小比較のみで決定される問題であることが分かる．ここで， 次の疑問が出る．

“『タイルペイント』は人間的テクニックを再帰的に使うのみで解けるのではないか？”

この問題の答えとして表 11 がある．表 11 は番号 1 から 11 の問題を再帰的テクニックを有する解法で再度解いたものである．

番号	問題 (大きさ)	変数の数	time
1	図 11 (4 × 4)	0	0
2	[19] No.00 (6 × 6)	0	0
3	[19] No.01(10 × 10)	0	0
4	[19] No.04 (10 × 10)	0	0
5	[12] p.49, 1 (10 × 10)	26	0.031
6	[13] p.39, 1 (10 × 10)	0	0
7	[14] p.41, 1 (10 × 10)	0	0
8	[15] p.71, 1 (10 × 10)	0	0
9	[15] p.71, 2 (10 × 10)	0	0
10	[19] No.02 (10 × 10)	22	0.015
11	[19] No.03 (10 × 15)	0	0

表 11: 再計算の結果

表 11 の番号 5 と 10 を見ると上の疑問は否定される．人間的テクニックを再帰的に使うのみでは，パズル『タイルペイント』は解くことができない．すなわち，真に連立方程式を解く必要がある場合が存在する．しかしながら， 実験結果を見ると人間的テクニックを再帰的に使うことは大変有効であることが分かる．また， 番号 5 と 10 の問題は，パズルの大きさは小さいが，簡単な数字の大小関係を比較するのみでは解くことができない問題である．番号 12，13，14，15 のような大きな問題より本質的には難しい問題ではないかと推察される．逆に番号 12，13，14，15 はサイズが大きいだけで，本質的には簡単な問題ではないかと推察される．

4.3 『タイルペイント』のまとめ

パズル『タイルペイント』は，ラテン方格系のパズルと違いモデル化したときの多項式の次数が小さい．『タイルペイント』での 1 つの変数に対する次数は 2 なのに対し，ラテン方格系のパズルの次数は 9 である．次数が小さいことは，グレブナー基底計算を高速に行うためには良いことである．しかしながら，何もテクニックを使わずにモデル化し計算すると，さすがに Risa/Asir が高速にグレブナー基底計算ができるからといって，問題が大きくなると計算結果が得られない．そこで，パズルのモデル化のテクニックとして人間的テクニック，再帰的テクニックを導入した結果，実験で使用した問題はすべて高速に解くことができた．パズル『タイルペイント』であれば，テクニックを使用したグレブナー基底を用いることによって，大抵の問題は解くことが可能であると実験結果から推察される．

5 おわりに

本稿では，Risa/Asir の組み込み関数 `dp_gr_main` を用いてどの程度のパズルを解くことが可能であるか計算実験をし，データを与えた．また，パズルを解く際に，人間的なテクニックを情報として導入することで，計算の効率化が計れることもデータとして与えた．本研究を始める前の我々の予想は，グレブナー基底を使って “現実的な時間” で解くことができるパズルは存在しないだろう，と予想をしていたが，パズル『タイルペイント』は簡単な人間的なテクニックを使うことにより “現実的な時間” で解くことが可能であることが分かった．これは，人間的なテクニックが大変有用に働くと共に，各変数からのみなる多項式の次数が 2 と小さいからである．ここで紹介したパズル以外にもグレブナー基底を用いて解くことができるパズルが存在するであろう．多項式の次数が小さく人間的なテクニックを多項式としてモデル化することができる，もしくは，簡単に解の一部が判定可能であれば，グレブナー基底を用いて “現実的な時間” で解くことが可能であろう．グレブナー基底の理論を学んだ学習者へグレブナー基底の応用問題として『グレブナー基底を使って解くことができる他のパズルを探す』ことを，チャレンジ問題として残し本稿を終える．

参 考 文 献

[1] Elizabeth Arnold, Stephen Lucas and Laura Taalman, Gröbner basis representations of Sudoku, *The College Mathematics Journal*, Vol. 41(2), pp. 101-112, 2010

[2] R. M. Falcón and J. Martín-Morales, Gröbner bases and the number of Latin squares related to autotopisms of order ≤ 7, *Journal of Symbolic Computation*, Vol.43(11-12), pp.1142–1154, 2007

[3] Jesús Gago-Vargas, Isabel Hartillo-Hermoso, Jorge Martín-Morales and José María Ucha-Enríquez, Sudokus and Gröbner bases: not only a divertimento, *Proc. Computer Algera in Scientific Computing, LNCS*, Vol. 4194, pp.155–165, Springer, 2006

[4] Alex Griffith and Adam Parker, A Gröbner basis approach to number puzzles, *Proc. Midstates Conference for Undergraduate Research in Computer Science and Mathematics*, pp.57–64, Oberlin College, 2009

[5] Shutaro Inoue, Efficient singleton set constraint solving by Boolean Gröbner bases, *Communication of JSSAC*, Vol. 1, pp. 27–38, 2012
[6] 井上秀太郎，佐藤洋祐，グレブナ基底を使った数独の難易度判定と問題作成，数理解析研究所講究録，第 1785 巻，pp. 51–56，2012
[7] 井上秀太郎，佐藤洋祐，鈴木晃，鍋島克輔，ブーリアングレブナ基底を使った数独の解法，数理解析研究所講究録，第 1666 巻，pp. 1–5，2009
[8] JST CREST 日比チーム (編)，グレブナー道場，共立出版，2011
[9] 中山洋将，グレブナー基底・マルコフ基底によるパズルの解法，グレブナー 若手集会 (静岡大学)，講演発表& プレプリント，2012
[10] ニコリ http://www.nikoli.com/ja/
[11] ニコリ編，ペンシルパズル本 73，橋をかけろ 1，株式会社ニコリ，2001
[12] ニコリ編，パズル通信ニコリ Vol. 120，株式会社ニコリ，2007
[13] ニコリ編，パズル通信ニコリ Vol. 122，株式会社ニコリ，2008
[14] ニコリ編，パズル通信ニコリ Vol. 124，株式会社ニコリ，2008
[15] ニコリ編，パズル通信ニコリ Vol. 135，株式会社ニコリ，2011
[16] 南沙也加，グレブナ基底を用いた四独・数独の研究，東京電機大学大学院理工学研究科 修士論文，2012
[17] 南沙也加, 張替賢, 中野哲夫，唯一の解をもつ四独の初期配置の解析，数式処理，Vol.18-2, pp. 21 – 24，2012
[18] Masayuki Noro and Taku Takeshima, Risa/Asir- A Computer Algebra System, *Proc. ISSAC 1992*, pp. 387 – 396, ACM-Press, 1992
http://www.math.kobe-u.ac.jp/Asir/asir-ja.html
[19] 連続発破 (はっぱ)，連続発破 保管庫『タイルペイント』 http://indi.s58.xrea.com/
[20] Yosuke Sato, Shutaro Inoue, Akira Suzuki, Katsusuke Nabeshima and Ko Sakai, Boolean Gröbner bases, *Journal of Symbolic Computation*, Vol.46(5), pp.622–632, 2011