画像構成パズル生成のためのタイル彩色問題

学生番号　2015130
氏名　後藤大輝
提出年度　平成 30 年度

目次

1. はじめに

本研究では，**画像構成パズル**を取り上げる．画像構成パズルとは，m 枚の画像と n 枚の単位正方形（タイル）が与えられ，この n 枚のタイルを用いて m 枚の画像それぞれを構成することを問うパズルである．なお各画像はタイルの枚数と等しい n 個の**ピクセル**（画素）から成るものとし，タイルを過不足なく使用しなければならない．また各ピクセルには与えられた d 色のうちいずれか 1 色が割り当てられているものとする．さらに各タイルは表と裏のそれぞれに，d 色のうちいずれか 1 色が割り当てられているものとする．実際の画像構成パズルの例を以下の図 1 に示す[1]．この例は，$m = 2, n = 64, d = 7$ の問題例である．

本研究では，この画像構成パズルを生成する問題を考える．すなわち，与えられた画像集合に対し，そのすべての画像を構成できるように，各タイルの表と裏に色を割り当てる問題である．この問題を**タイル彩色問題**という．山本ら [1] は，タイル彩色問題における解の存在性に関する必要十分条件を示し，$m = 3$ の場合の多項式時間アルゴリズムを与えた．なお $m \geqq 4$ の場合は多項式時間で解けるかどうか未解決である．山本らは $m \geqq 4$ の場合について $O(2^{2^m n})$時間の全探索アルゴリズムの存在を示唆したが，これは指数の肩にさらに指数が乗っているため，現実的な計算時間を達成するとは言い難い．一方整数最適化は，近年のハードウェアとアルゴリズムの発展に伴い，それを解くための**ソルバ**の性能は 20 年間で 1000 万倍向上したとする意見もある [3]．整数最適化ソルバは指数時間を要する厳密解法であるため，決して多項式時間アルゴリズムを与えるものではないが，現状では問題を解くための現実的な解決策の 1 つとして考えられるだろう．

本研究ではタイル彩色問題を**整数最適化問題**として定式化する．さらに提案する定式化では，部分問題として**最大フロー問題**が現れることに着目し，一部の整数制約を緩和できることを示す．制約緩和の結果，計算時間を大幅に改善することができた．

以下，2 章ではタイル彩色問題とその応用例について紹介し，3 章ではタイル彩色問題の定式化，最大フロー問題，そしてタイル彩色問題における整数制約の緩和について解説する．4 章では計算実験の結果を報告し，制約緩和の有用性を示す．また現実の画像を元にタイル彩色問題を生成し，それを解いた結果も報告する．5 章でまとめを行う．

図 1　画像構成パズルに用いられるタイルと画像の例

2.　タイル彩色問題

2.1.　定義

タイル彩色問題とは，任意の m 枚の画像と n 枚のタイルが与えられたとき，その n 枚のタイルで m 枚の画像を全て構成できるようにするために，各タイルの表面と裏面にどのように色を塗ればよいかを問う問題である．なお使用可能な色の数は d 色で，与えられているものとする．

山本ら[1]は，画像枚数が 3 枚（m = 3）のとき，計算時間がオーダ O(n)となる多項式時間アルゴリズムを与えた．なお m = 1,2 のときはこの問題が解を持つことは自明である（タイルの表面と裏面にそれぞれの画像の色を塗ればよいため）．また m≧4 の場合は未解決となっている．

本研究で提案するのは，整数最適化によるタイル彩色問題の定式化である．整数最適化問題は NP 困難と呼ばれる複雑さのクラスに属しており[2]，やはり多項式時間でこれを解くのは絶望的である．ところが，整数最適化問題を解くためのソルバ（ソフトウェア）の進化は目覚ましい．これらのソルバは，理論的には指数時間を要する厳密解法であるが，実際の計算時間は，問題例がよほど大きくない限り，実用的な範囲内で収まることが多い．本研究ではタイル彩色問題を整数最適化問題として定式化し，そのソルバを用いて解を得ることを考える。

2.2.　応用例

タイル彩色問題は，次のような**雇用計画問題**に応用可能と考えられる．近年の社会では，正社員として働くのではなくフリーターや派遣などの非正規雇用で生計を立てている人々が存在する．そして経営者は，こういった人々を効率よく雇用するということが求められる．これらのことから現代社会では，雇用計画が重要なことだということが言える．

雇用計画問題とは，ある企業における雇用計画を想定しており，その企業では複数の異なる**業務**を日ごと代わる代わる行っているとする．また各業務における必要な人数の合計数は変わらないが，各**ジョブ**の人数の内訳は日によって異なる．そして雇用される人々が得意とするジョブは高々2 つとする．次の図 2 は，企業が行う業務のイメージ図である．

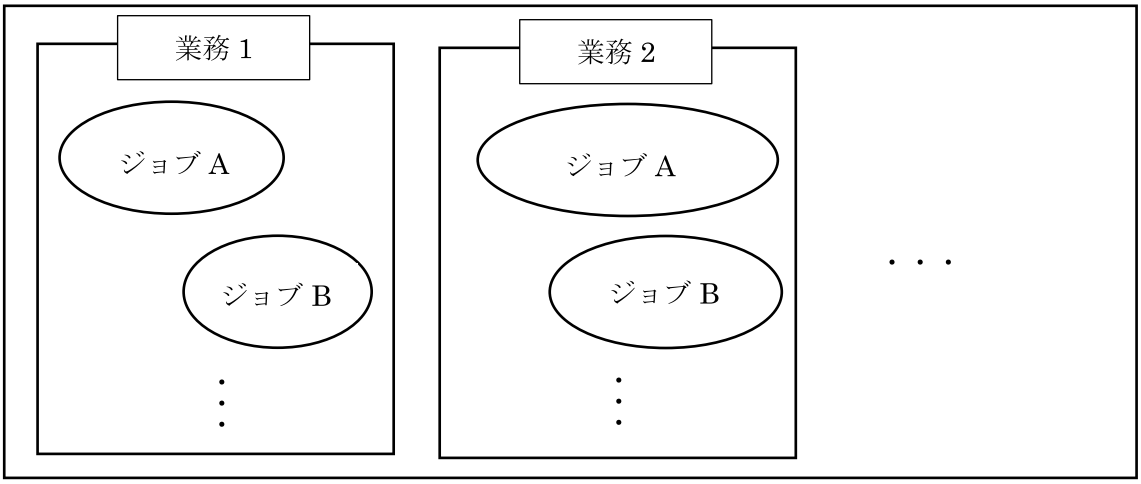


図 2　企業が行う業務とジョブの概念図

これをタイル彩色問題に当てはめると，業務の種類数が画像枚数 m，雇用される人の数がタイルの枚数 n，複数の異なるジョブの数が色の数 d ということになる．すなわち，タイル彩色問題と雇用計画問題は次のように対応している．

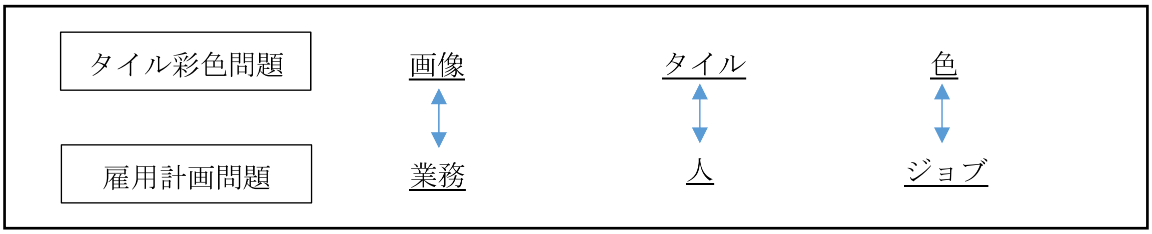


図 3　タイル彩色問題と雇用計画問題の対応関係

例を挙げて解説すると，色 A～D で構成された 3 枚の画像を 100 枚のタイルで構成することを問うタイル彩色問題は，業務が 3 種類ある企業で A～D の 4 つのジョブを行う場合，どのような技能を持つ 100 人を雇えばよいかを問う雇用計画問題と同じになる．この問題例を表で表すと次のようになる．

表 1　タイル彩色問題の問題例

	色 A	色 B	色 C	色 D
画像①	30	20	15	35
画像②	40	18	23	19
画像③	5	40	30	25

表の数値は各画像（業務）が必要とする各色（ジョブ）の数値で，タイル彩色問題ならばタイルの枚数を，雇用計画問題ならば人数を表す．

上記のことから，タイル彩色問題の考え方は実際の社会にも生かすことができると考えられる．

3.　整数最適化による定式化

本章では，タイル彩色問題を整数最適化問題として定式化する．また，その整数制約の一部を緩和できることを示す．

3.1.　整数最適化

整数最適化とは，以下のような問題を指す．

<u>整数最適化問題</u>
maximize　Cx
subject to　$Ax \le b$
　　　　　　x は整数列ベクトル

ここで C は行ベクトル，A は行列，b は列ベクトルである．この問題は，$Ax \le b$ という条件の下，Cx を最大化する問題である．また，Cx は**目的関数**，$Ax \le b$ は**制約条件**という．このような問題で，変数 x に整数制約がついている問題を，整数最適化問題という．

3.2.　定式化

定式化するにあたって，画像の枚数を m，各画像のピクセル数（画素数）を n とする．色を整数 $1,\ldots,d$ で表す．また，相異なる色の対の集合を P とする（$P=\{\{1,2\},\{1,3\},\ldots,\{d-1,d\}\}$）．すなわち $|P|= d(d+1)/2$ である．一方の面を色 j，もう一方の面を色 k で彩色するタイル（$\{j,k\}$-タイル）の枚数を $y_{j,k}$ とする（$\{j,k\} \in P$）．また各画像 i について，$\{j,k\}$-タイルのうち，色 j として用いるタイルの枚数を $x_{i,j,k,j}$，色 k として用いる枚数を $x_{i,j,k,k}$ とする．

タイル彩色問題は以下のように定式化される．

<u>タイル彩色問題</u>
目的関数：割当枚数 $x_{i,j,k,j}$ の総和
制約条件：(1)タイルの合計枚数は n 枚である
　　　　　(2) $\{j,k\}$-タイルは最大 $y_{j,k}$ 枚しか使えない
　　　　　(3)各画像は構成可能でなければならない
　　　　　(4) $y_{j,k}$ は整数である
　　　　　(5) $x_{i,j,k,j}$ は整数である

次に制約条件(1)~(3)を表すための制約式について解説する．

3.2.1.　制約条件(1)について

「タイルの合計枚数は n 枚である」という条件を表す式は,

$$y_{1,2} + y_{1,3} + \cdots + y_{d-1,d} = n \qquad (1)$$

で表すことができる．なお $y_{j,k}$ は，一方の面に色 j，もう一方の面に色 k が塗られたタイルの枚数を表す.

3.2.2　制約条件(2),(3)について

画像の構成可能性を定義する．画像 i における色 c のピクセル数を $p_{i,c}$ とする（$i = 1,...,m$, $c = 1,...,d$)．与えられたタイル彩色（$y_{1,2},..., y_{d-1,d}$）について，各タイルの表裏の向き付けをうまく定めることで，すべての色 c についてちょうど $p_{i,c}$ 枚のタイルが見えるようにできるとき，**画像 i は構成可能**ということにする.

タイル彩色問題を，各画像 i について構成可能性に寄与するタイルの枚数を最大化する問題として捉える．すると,

$$x_{i,j,k,j} + x_{i,j,k,k} \leq y_{j,k} \qquad (2)$$

が満たされなければならない．これは,「{j,k}-タイルは最大 $y_{j,k}$ 枚しか使えない」という条件を表す制約式である．一方，色 c のピクセルは $p_{i,c}$ 個存在するので,

$$x_{i,c,1,c} + \cdots + x_{i,c,d,c} \leq p_{i,c} \qquad (3)$$

が満たされなければならない．これは,「各画像は構成可能でなくてはならない」という条件を表す制約式である．タイル彩色問題は(1), (2), (3)の条件の下，整数変数 $x_{i,j,k,j}$ の総和を最大化する問題に帰着される．もし最適値が mn のとき，かつそのときに限り，すべての画像が構成可能なタイル彩色が存在すると結論づけることができる.

なお画像 i に対する制約式(2), (3)の系は，タイルの枚数 $y_{j,k}$ を整数定数とみなすと最大フロー問題の制約とみなすことができる．したがって変数 $x_{i,j,k,j}$ は実数変数として取り扱っても差し支えない[2]．すなわち(5)の制約条件を**緩和**する（取り外す）ことができる.

次の節では最大フロー問題の制約とみなすことで変数 $x_{i,j,k,j}$ の整数制約を緩和することのできる理由を解説する.

3.3.　最大フロー問題を用いた制約緩和

グラフとは，**頂点**の集合と，頂点と頂点を結ぶ**辺**の集合から成る．辺が向きを持たないグラフを**無向グラフ**，向きを持つグラフを**有向グラフ**と言い，本研究で取り扱うグラフは全て有向グラフとする.

最大フロー問題とは，以下のような問題である.

最大フロー問題
目的：グラフの始点（ソース）から終点（シンク）にできるだけ多くのフローを流すこと
制約：①各辺についてその辺を流れるフローの量が，その辺に設定されている容量を超えてはいけない
②始点と終点以外の全ての点においてフローの流入量と流出量は等しい**（フロー保存則）**

以下の図 4 に最大フロー問題の例題を示す．なおアルファベットは辺の名前を，カッコ内の数値は辺の容量を表す．

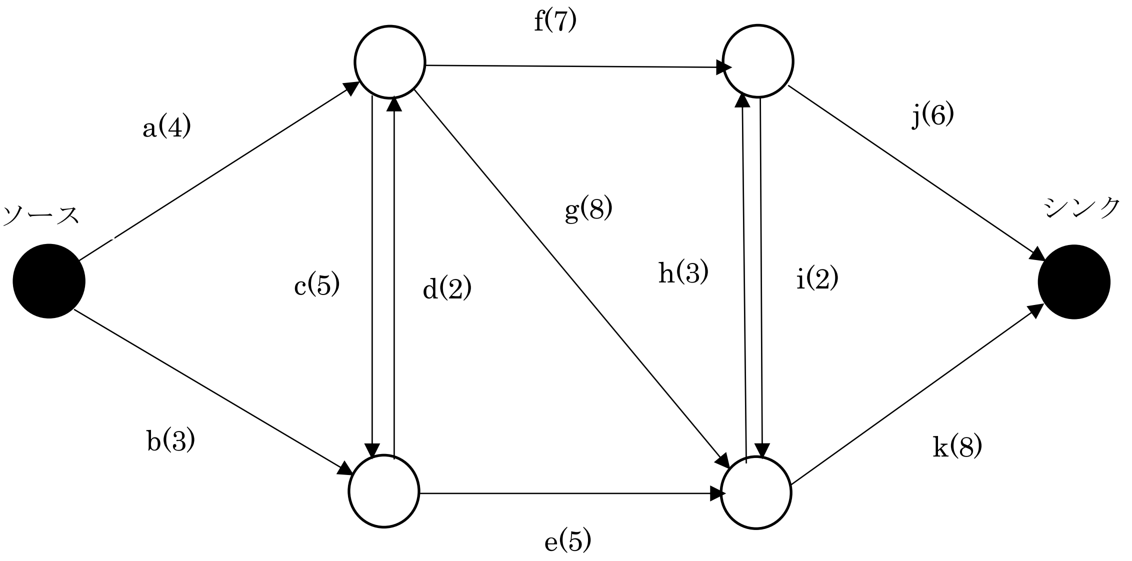


図 4　最大フロー問題　例題

辺 a に流すフローの量を x_a，フローの総量を z とするとき，この問題は以下のように式で表すことができる．

最大フロー問題　例題

maximize z

subject to

$$
\begin{aligned}
x_a &\leq 4 \\
x_b &\leq 3 \\
&\cdots \\
x_k &\leq 8
\end{aligned}
$$

各辺の容量についての制約式

$$
\begin{aligned}
x_a + x_b &= z \\
x_j + x_k &= z \\
x_a + x_d &= x_c + x_f + x_g \\
&\cdots
\end{aligned}
$$

各点のフロー保存則についての制約式

この問題において，各辺の容量が全て整数ならば流すフロー変数を実数として取り扱っても整数の最適解が得られることが知られている[2]．なぜなら，最大フロー問題ではフローをできるだけ多く流そうとするため，各辺の容量いっぱいまで流すことが望ましいからである．このことから，前節の制約式(2),(3)から成る最大フロー問題について，流すフロー変数が $x_{i,j,k,j}$ であるため $x_{i,j,k,j}$ の整数制約を外し，実数変数として取り扱うことができるということになる．よって，提案手法によるタイル彩色問題の定式化は以下のようになる．

タイル彩色問題

目的関数：割当枚数 $x_{i,j,k,j}$ の総和

制約条件：(1)タイルの合計枚数は n 枚である

(2)各タイルはあるだけしか使えない

(3)各画像は構成可能でなければならない

(4) $y_{j,k}$ は整数である

~~(5) $x_{i,j,k,j}$ は整数である~~

以下の図 5 は，1 つの画像 i に関する制約式(2),(3)に対応する最大フロー問題を表したグラフである．

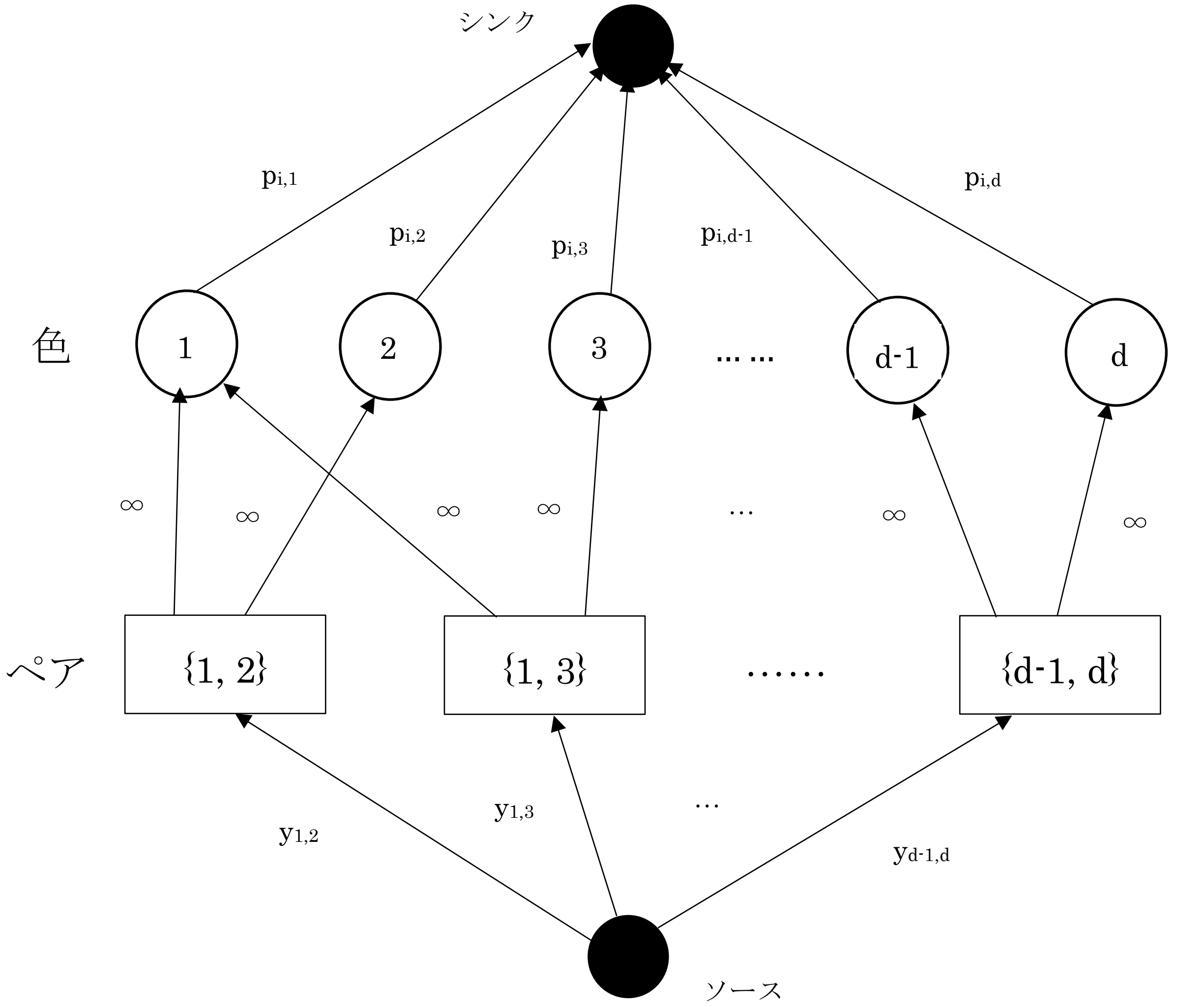


図 5　画像 i に関する制約式(2),(3)に対応する最大フロー問題のグラフ

各辺に付された値はその辺の容量を表し，各頂点に付された表現はその頂点の名前を表す．ソースから{j,k}-ペアに出ている辺の容量は{j,k}-タイルの枚数分（すなわち $y_{j,k}$ 枚）である．ペア頂点と色頂点を結ぶ辺の容量は，制約が設定されていないので無限大としている．色頂点からシンクに向かう辺の容量は，画像 i に用いられている色 c のタイル枚数 $p_{i,c}$ である．この最大フロー問題のグラフにある，制約のある辺の容量はすべて整数であるため，流すフローは整数制約を緩和し，実数として扱うことができる．

4. 実験

提案手法の有用性を示すために，二種類の実験を行った．一つ目の実験では，割当枚数 $x_{i,j,k,j}$ の整数制約を外した場合（**緩和した場合**）と外さなかった場合（**緩和しなかった場合**）で計算時間に変化が現れるかどうかとタイルが彩色可能かを検証した．二つ目の実験では，実際の画像を用いて画像構成パズルを生成できるかとその計算時間を検証した．なお，どちらの実験も使用したコンピュータはメモリ 2GB，プロセッサが Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz，OS は Windows10 Pro である．

4.1. 実験①：整数制約の有無による変化の検証

色の数 d，画像の枚数 m，ピクセル数（タイルの枚数）n，色分布の決定法（完全ランダム or 偏りあり）を入力することで問題例を生成し，そのタイル彩色可能性を判定し，計算時間を表示するプログラムを Python[4]で作成し，実験を行った．色分布決定法の「偏りあり」とは，一つの色が多く使われるなど，実際の画像をイメージして導入したものである．具体的には，色 1 が暗い色，色 d が明るい色であることを想定し，各画像が暗めの画像か明るめの画像かを，確率 1/2 で決定する．そして暗めの画像の場合は，各ピクセルが色 k を確率 $1/2^k$ で取るように生成し，明るめの画像の場合は，各ピクセルが色 k を確率 $1/2^{d+1-k}$ で取るように生成する．数理最適化ソルバには pulp ライブラリ[5]に標準搭載されている COC を用いた．

この実験では，d, m∈{3,4, ..., 10, 12, ..., 20}，n∈{100, 200, 300}とし，2 通りの分布，5 通りの擬似乱数について，計 5070 通りの問題例を割当枚数 $x_{i,j,k,j}$ の整数制約を緩和した場合と緩和しなかった場合の 2 回解いた．次の表 2 は 5070 通りの計算時間の平均値と最大値と最小値を表したものである．

表 2　計算時間（秒）

	整数制約緩和なし	整数制約緩和あり
平均値	13.70	1.27
最大値	520.84	21.14
最小値	0.01	0.03

この表から整数制約を緩和することで計算時間を 1/12 ほどに短縮できたことが分かった．

ここからは，画像枚数，色の数，ピクセル数，色分布決定法ごとに計算時間や彩色可能性を見ていく．5070 通りの問題例のうち，彩色不可能なものは 191 個であった．次の図 6,図 7 はそれぞれ画像枚数の変化に伴うタイル彩色可能性と計算時間の変化である．

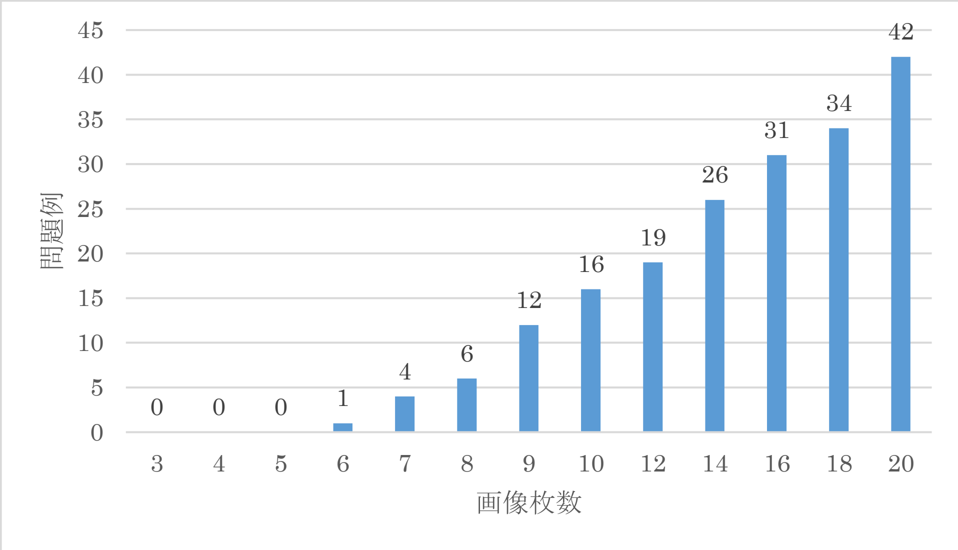


図 6　彩色不可能となる問題数の例

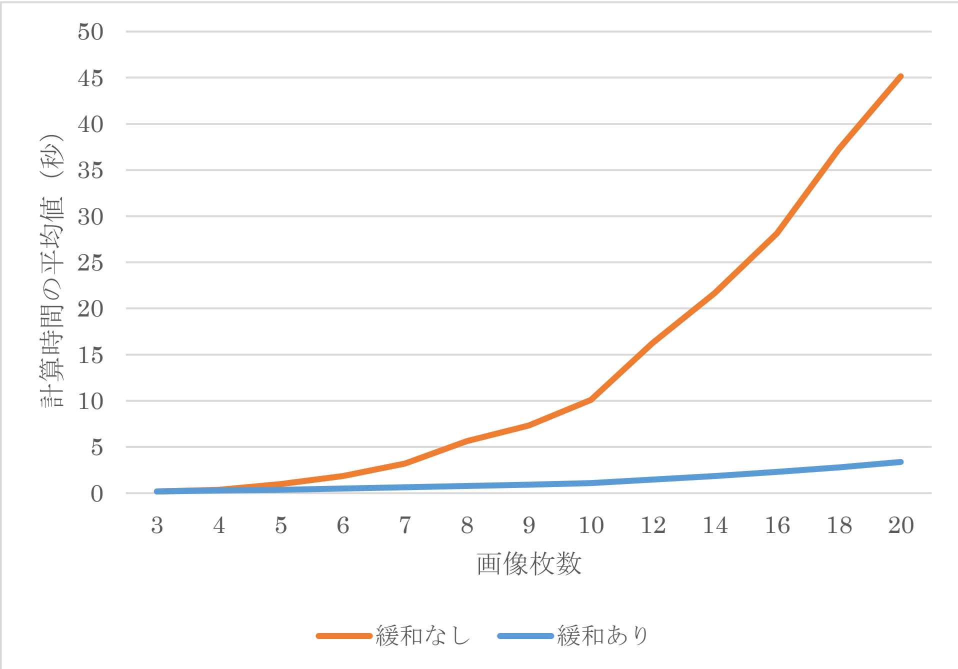


図 7　計算時間の比較

図 6 のグラフからわかることとして，画像枚数が増え，問題が複雑になるにつれて彩色不可能となる問題例が多くあらわれるようになった．また，図 7 からは整数制約を緩和する場合と緩和しない場合のどちらも，画像枚数が増えるにつれて計算時間が増えていることがわかるが，緩和ありの計算時間の伸びが明らかに緩やかであると読み取ることができる．

次に色の数の変化に伴うタイル彩色可能性と計算時間の変化をそれぞれ以下の図 8,図 9 に示す.

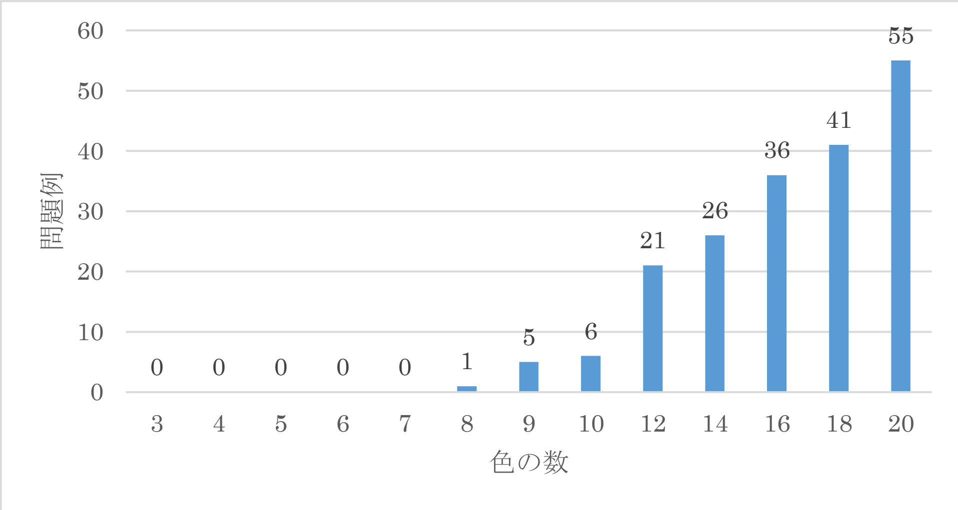


図 8　彩色不可能となる問題数の例

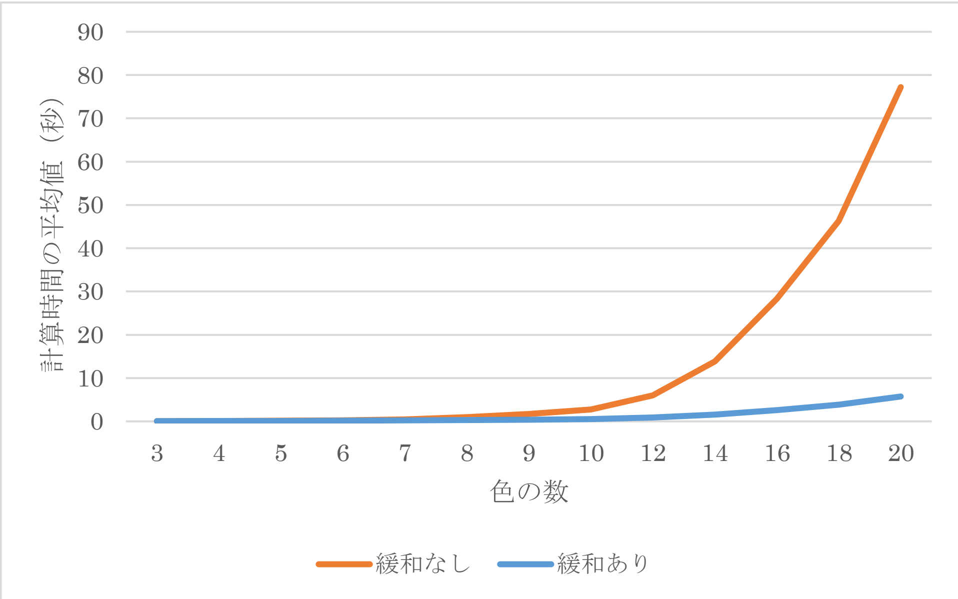


図 9　計算時間の比較

図 8,図 9 から，色の数の変化によるタイル彩色可能性と計算時間の変化はともに画像枚数の変化の場合と同じく，色の数の増加に伴い彩色不可能となる問題例の数と計算時間が増加する傾向がみられた．また画像枚数の変化の場合と比べて，3 から 8 では変化はあまり見られないが，9 以上の数値では急激に値が伸び，色の数が 20 のときは彩色不可能となる問題例，緩和なしの計算時間，緩和ありの計算時間のすべての項目で画像枚数が 20 のときのそれぞれの値よりも高い数値が表れていることが分かった.

次にピクセル数の変化に伴うタイル彩色可能性と計算時間の変化をそれぞれ以下の図 10,図 11 に示す.

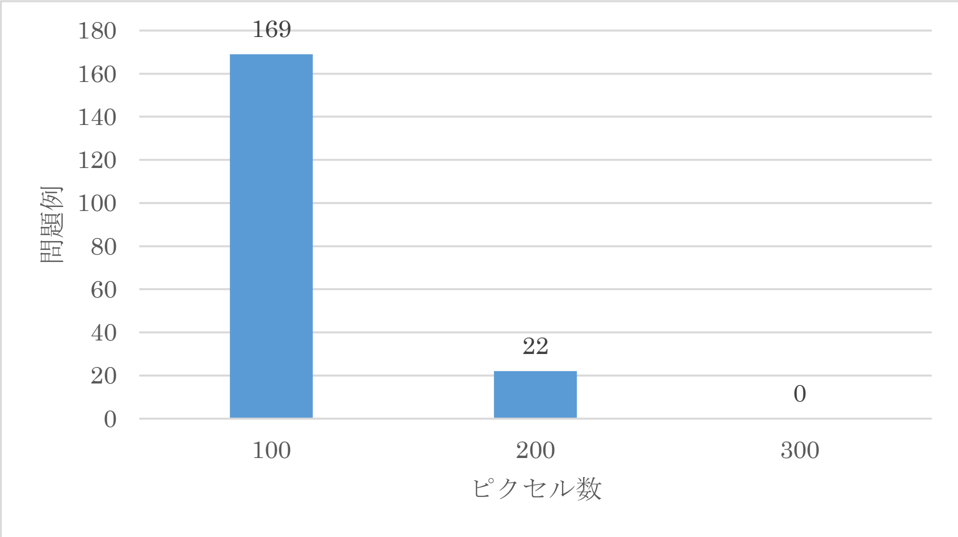


図 10　彩色不可能となる問題数の例

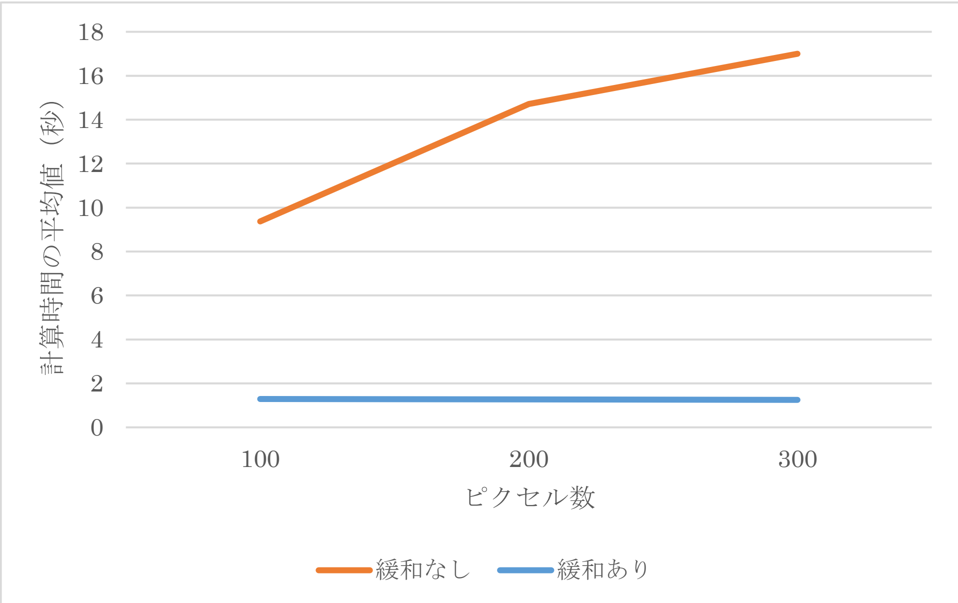


図 11　計算時間の比較

図 10 から，ピクセル数が増えると彩色不可能となる問題例の数は減少するということを読み取ることができる．また図 11 から，計算時間は緩和なしの場合はピクセル数の増加とともに計算時間も増加するという傾向を読み取ることができるが，緩和ありの場合は横ばいでほとんど変化しなかった．このことからピクセル数の増加は，画像枚数や色の数の増加と異なる傾向があることが分かった．

最後に生成法の変化に伴うタイル彩色可能性と計算時間の変化をそれぞれ以下の図 12, 図 13 に示す．

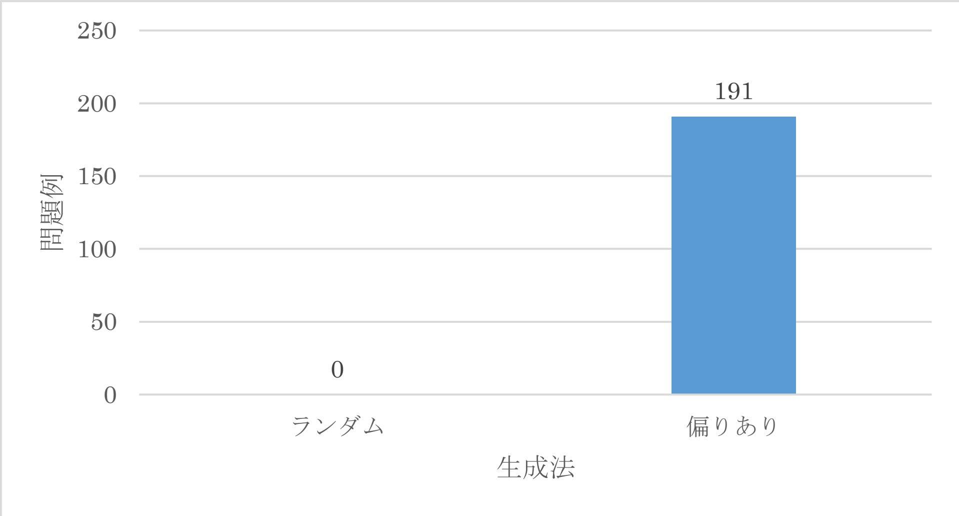


図 12　彩色不可能となる問題例の数

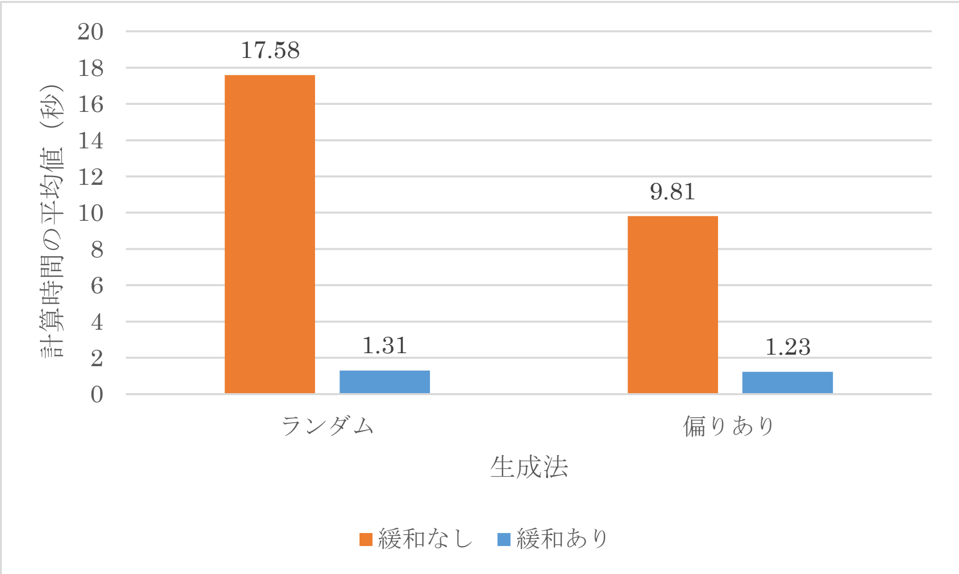


図 13　計算時間の比較

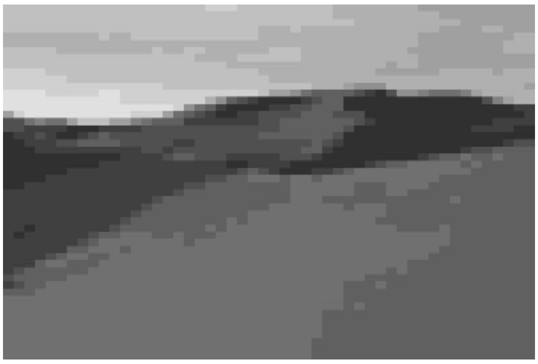
図 12 から彩色不可能となった問題例は，全て偏りありで生成した問題例であることが分かった．また，計算時間については緩和ありではほぼ変化しなかったのに対し，緩和なしでは生成法の違いで大きく計算時間が変わっていることが読み取れる．ここで図 10,図 11 の結果と図 12,図 13 の結果を合わせてみると，彩色不可能になる問題例が多いほど，計算時間は短くなっているといえることが分かった．

以上のことから，提案手法である整数制約の緩和が有用だといえる．次は実際の画像を用いた実験の結果について解説する．

4.2.　実験②：実際の画像を用いた検証

この実験では実際の画像を用いてタイル彩色の可能性を判定し，画像構成パズルを生成できるか検証する．実際の画像を用いるにあたり，実際の画像はピクセル数や色の数が非常に多く問題が複雑になることが予想された．取り扱いやすくするために画像を n×n に分割し，指定した色の数でグレースケール化するプログラムを Python で作成した．画像処理には Pillow ライブラリ[6]を用いた．以下の図 14 は画像をグレースケール化したものの例である．

原画像[7]



50×50　16 色

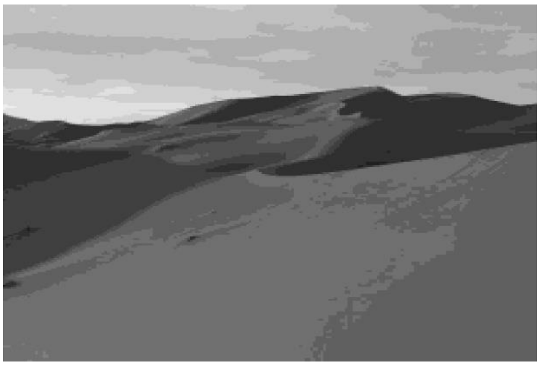
50×50　32 色

50×50　64 色

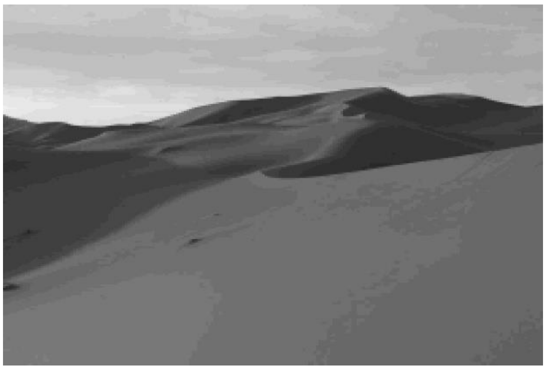
100×100　16 色

100×100　32 色

100×100　64 色



200×200　16 色



200×200　32 色

200×200　64 色

図 14　グレースケール化した画像の例

この実験では，m∈{5,10}，d∈{16,32,64}，n∈{2500, 10000, 40000}とし，実験①で使用したプログラムで，彩色可能性と計算時間を検証した．使用した画像について，10 枚の画像を用意し，それらを 3 通りの d，および 3 通りの n について，合わせて 3×3 = 9 通りにグレースケール化した．また，m=5 のときは 10 枚の中からランダムに選択した 5 枚で実験を行った．ただし，5 枚の実験は全てその 5 枚で実験を行うものとし，途中で使用する画像を変更していない．

次の表 3 は実験②の結果を表す表である．

表 3　実験②の結果

画像枚数	色の数	ピクセル数	彩色可能性	計算時間（秒）
5	16	2500	TRUE	1.86
5	16	10000	TRUE	0.61
5	16	40000	TRUE	0.66
5	32	2500	TRUE	6.86
5	32	10000	TRUE	6.34
5	32	40000	TRUE	6.58
5	64	2500	TRUE	170.73
5	64	10000	TRUE	1146.62
5	64	40000	TRUE	1065.84
10	16	2500	FALSE	1.63
10	16	10000	FALSE	1.58
10	16	40000	FALSE	1.64
10	32	2500	FALSE	52.39
10	32	10000	FALSE	25.34
10	32	40000	FALSE	26.42
10	64	2500	FALSE	1452.68
10	64	10000	FALSE	1269.87
10	64	40000	FALSE	1276.95

表の結果から，画像枚数 5 枚であれば色の数やピクセル数がいくつであっても彩色可能となったことに対し，画像枚数 10 枚では色の数やピクセル数がいくつであっても彩色不可能となった．計算時間については，色の数の値によって計算時間が大きく変わることが読み取ることができる．画像枚数によって計算時間は多少変化するが，色の数の違いによるほどの変化は見られなかった．

この結果から，実際の画像でもタイル彩色は可能であるが，画像の枚数を増やすと彩色不可能となる場合が増えるということがわかる．また，計算時間はピクセル数ではなく画像枚数や色の数に依存し，特に色の数に大きく依存していることが読み取ることができた．

5. まとめ

本論文では，画像構成パズルのためのタイル彩色問題について整数最適化により定式化し，その一部を最大フロー問題とみなすことで整数制約を緩和する手法を提案した．計算実験の結果，この緩和によって大幅な高速化を達成できることを確認した．また，山本ら[1]の先行研究では現実的とは言い難かった，m≧4 の場合でも提案手法では現実的な時間で解くことが可能であると 2 つの実験を通して明らかにした．

今後の課題を 3 つ挙げる．1 つ目は m≧4 の場合の多項式時間アルゴリズムを与えることである．今回筆者が実装したプログラムによる実験では，彩色が可能か不可能かを判定するまでにかかった時間をそのまま表示するものだった．そのため，先行研究で未解決とされていた m≧4 の場合の多項式時間アルゴリズムは本論文でも未解決のままとなった．ただ，先行研究で示唆されていたオーダ $O(2^{2^m n})$よりは早くできるだろうということが今回の実験を通して示すことができた点において有益だったであろう．

2 つ目の課題は彩色可能となる問題例を増やすことである．今回の実験では，画像枚数や色の数が増え問題が複雑になればなるほど彩色不可能となる問題例が増えていった．現実のパズルでも，簡単に解けてしまうパズルよりも，複雑で難しいパズルの方が解きがいのある面白いパズルであると考える．そのため，今回のタイル彩色問題でも複雑な問題を生成できるようになるということが目標の 1 つになるといえるだろう．そのためにはどういったときに彩色不可能になるのかという法則性をつかむことが必要であり，法則性をつかむにはさらに実験を重ねていく必要があるだろう．

3 つ目の課題はアプリの開発である．スマートフォンなどで撮影した画像を読み取り，それを自動で分割し画像構成パズルを生成するアプリの開発を思案している．そのためにはクリアしなければならない問題が数多くあるため，まずはより深く実験を重ねていく必要があると考えている．

本論文の末尾に，付録として実験①の結果の詳しい数値を示す表を掲載する．

謝辞

本研究において指導をしていただいた先生をはじめ，色々な面で支援してくださった同ゼミの方々に心より感謝を申し上げます．

参考文献

[1]山本陽平，金森由博，三谷純：複数の正答を持つ単位正方形の組合せパズルに関する研究．情報処理学会論文誌，Vol.56，No.6，2015, pp.1507--1516.
[2] B. コルテ，J. フィーゲン：組合せ最適化〜理論とアルゴリズム〜．シュプリンガー・フェアラーク東京，2005.
[3]梅谷俊治：組合せ最適化入門：線形計画から整数計画まで．言語処理学会第 19 回年次回（NLP2013），スライド，https://www.slideshare.net/shunjiumetani/ss-17197023（2018 年 12 月 14 日閲覧）
[4]Python：https://www.python.org/ （2018 年 12 月 22 日閲覧）
[5]Pulp ライブラリ：https://pypi.org/search/?q=pulp（2018 年 12 月 22 日閲覧）
[6]Pillow ライブラリ：https://pypi.org/search/?q=pillow（2018 年 12 月 22 日閲覧）
[7]実験用画像：https://weathernews.jp/s/topics/201702/090065/ （2018 年 12 月 18 日閲覧）

付録

表 4　画像枚数についての実験結果

	問題例の個数（個）		計算時間の平均値（秒）	
画像枚数	彩色可能	彩色不可能	緩和なし	緩和あり
3	390	0	0.18	0.18
4	390	0	0.35	0.27
5	390	0	0.98	0.37
6	389	1	1.85	0.51
7	386	4	3.20	0.62
8	384	6	5.61	0.76
9	378	12	7.31	0.91
10	374	16	10.11	1.09
12	371	19	16.26	1.47
14	364	26	21.69	1.86
16	359	31	28.12	2.31
18	356	34	37.26	2.79
20	348	42	45.15	3.37

表 5　色の数についての実験結果

	問題例の個数（個）		計算時間の平均値（秒）	
色の数	彩色可能	彩色不可能	緩和なし	緩和あり
3	390	0	0.05	0.05
4	390	0	0.09	0.09
5	390	0	0.12	0.11
6	390	0	0.22	0.15
7	390	0	0.45	0.20
8	389	1	0.99	0.27
9	385	5	1.71	0.38
10	384	6	2.77	0.52
12	369	21	5.99	0.93
14	364	26	13.82	1.59
16	354	36	28.31	2.63
18	349	41	46.33	3.85
20	335	55	77.21	5.74

表 6　ピクセル数についての実験結果

	問題例の個数（個）		計算時間の平均値（秒）	
ピクセル数	彩色可能	彩色不可能	緩和なし	緩和あり
100	1521	169	9.37	1.29
200	1668	22	14.72	1.27
300	1690	0	17.00	1.25

表 7　生成法についての実験結果

	問題例の個数（個）		計算時間の平均値（秒）	
生成法	彩色可能	彩色不可能	緩和なし	緩和あり
ランダム	2535	0	17.58	1.31
偏りあり	2344	191	9.81	1.23