6K-06

MCMC法によるヒントの少ない数独問題の生成*

古川　湧†　　山本　修身‡

名城大学大学院 理工学研究科†　　名城大学 理工学部‡

1　はじめに

数独は「数字は独身に限る」の略称である．ナンバープレースとも呼ばれ，新聞やパズル雑誌，スマホアプリなどに普及しており世界中で遊ばれている．数独は初期状態におけるヒントの個数が少ないほどパズルの難易度は高くなり，多いほど簡単になる傾向にある．本研究で扱う数独のルールは以下の通りである: 数独は 9×9 マスで構成されている．この場合，マスには 1 から 9 のいずれかの数字がはいり，同じ行と列，9 つに分けられた 3×3 のブロックの中に同じ数字が出現しないように数を入れる．また，初期にあらかじめ与えられているヒントの数字を変えてはいけない．

本稿ではメトロポリス法によって生成した問題の中から，ヒント数が多かった問題についてマルコフ連鎖モンテカルロ法（MCMC法）によって冗長なヒントの組を見つけ出し，それらのヒントを削減して新たにヒントを添加することでヒントの数を減らすことを試みた．その結果，図 4 に示すようにヒント数 21 の数独問題をヒント数 18 の数独問題にすることができた．

2　メトロポリス法による問題の生成

2.1　メトロポリスのアルゴリズム

メトロポリスのアルゴリズムを用いることで，与えた状態空間 C から評価値をとる関数 $f(C)$ の値に比例した出現確率を持つようなマルコフ連鎖を作ることができる．定常確率分布が

$$P(C_i)=\frac{f(C_i)}{\sum_{c\in S}f(c)} \tag{1}$$

となるマルコフ連鎖を構成する．ただし，S は全パターンの集合とする．今，この状態間を遷移するマルコフ過程提案分布があるとし，状態 C_i から状態 C_j への遷移確率が $C(j|i)$ であるとする．このとき，$C(j|i)=C(i|j)$ であると仮定する．メトロポリスのアルゴリズムでは，このマルコフ過程から目的のマルコフ過程を以下のようにして作る: 確率 $p_{ij}=\min\left(1,\frac{f(C_j)}{f(C_i)}\right)$ で C_i から C_j に遷移させ，確率 $1-p_{ij}$ で遷移させずに，C_i に留まらせる．

2.2　メトロポリス法による解のサンプリング

まず，マスの状態 C が数独の解からどの程度解離しているかを表す関数を $E(C)$ として以下のように定義する．この値が 0 の場合は数独の解となり，0 よりも大きい場合には数独の解ではない:

$$E(C)=\sum_{j=1}^{9}E_{r,j}(C)+\sum_{k=1}^{9}E_{b,k}(C) \tag{2}$$

ただし，$E_{r,j}(C)$，$E_{b,k}(C)$ はそれぞれ j 列目，k 番目のブロックにおいて 1 から 9 までの数字のうち使われていない数字の個数を表している．行はあらかじめ 1 から 9 の数字をそろえておく．それぞれ，1 から 9 までの数字を重複なく配置している場合は $E(C)=0$ となり，C は解であるといえる．次に状態 C を評価する関数 $f(C)$ をボルツマン因子を用いて次のように決める．このとき，

$$f(C)=\exp(-\beta E(C)) \tag{3}$$

とする．ただし，β は適当な正の実数である．

数独の解を求めるためにはまず，図 1 左のような適正でない問題を用意する．赤いマスはすでに確定しているヒントであり，これを動かしてはいけない．ここに図 1 中央のように適当に数字を入れる．次に適当な二つのマスを選び，そのマスの数字を入れ替えるというを何度も繰り返し行う．図 1 右のように行，列，3×3 ブロックに数字の重複がなくなった場合，数独の解がひとつ求まったといえる．メトロポリスのアルゴリズムを用いて状態エネルギー関数が $E(C)=0$ となるように数字を入れ替え，解を求める．このようにして解はバックトラックで得られる解と異なり，解空間からの一様なサンプリングになっていると考えられる．$E(C)=0$ となる完全な解を複数求めることでヒント数 17 の問題が生成できる [1] が，本稿では解もしくは解に近い状態 S_m から，出現したマスと数字の組み合わせを求める．図 1 中央のような状態から数字の入れ替えを $1,000,000$ 回行う．このとき入れ替えるごとに数字とマスの出現回数を

$$f_2(C)=\sum_{i=1}^{N}\exp-\beta E(C) \tag{4}$$

と評価してカウントする．ただし，N は数字を入れ替える回数である．温度変数は $\beta=2.0$ と設定している．完全な解が出現した場合は，一様な解を見つけるために $\beta=0$ に設定し，15 回の数字の入れ替えを行っている．

2.3　ヒントの逐次添加法

ヒント数 0 の適正でない問題から 1 ずつヒント数を増やすことで，ある程度ヒントの少ない適正問題が生成できる [2]．ここで注目するのは与えられたヒントから得られる解集合の個数である．ヒント数 0 の適正でない問題の解集合の大きさは約 $M=6.67\times10^{21}$ である [3]．ヒントを順次増やすことでなるべく急速に解集合の大きさが小さくなるようにする．バックトラッキングによる全探索またはメトロポリス法によるサンプリングによって得られた解から，最も出現回数の少なかったヒントを新たなヒントとして添加する．これを適正な問題になるまで繰り返す．この方法を実行し約 60 分で適正な問題を 1 個生成した [1]．これを繰り返し，ヒン

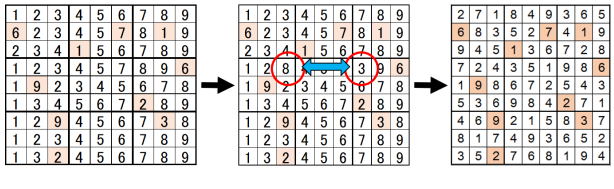


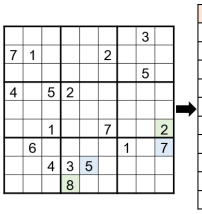
図 1: Finding a solution of a Sudoku problem by SA.

*Generation of Sudoku problems with small number of hints by MCMC

†Yu Furukawa Meijo University

‡Osami Yamamoto Meijo University

[1]実行環境は OS: Linux，コンパイラ: gcc 5.4.0，CPU: Intel(R) Xeon(R) E5-2640 v4 @ 2.40GHz×2(40 スレッド)．



Number of solutions	Deleted hints
1.34E+20	(11,16)
1.32E+20	(13,16)
1.28E+20	(10,17)
1.24E+20	(14,15)
1.03E+20	(4,13)
1.02E+20	(0,6)
1.01E+20	(2,11)
9.96E+19	(1,7)
9.90E+19	(10,12)
9.90E+19	(4,7)

図 2: Number of solutions from problem with 18 hints

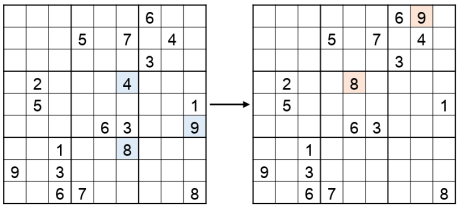


図 3: Reducing the number of hints from 19 to 18 by our method. The method took away three digits(blue) and added two other digits(red).

ト数 18 から 21 の問題を 100 個ほど生成することができた．

3 MCMC による解の見積もり

メトロポリス法によるマルコフ連鎖を用いることによって，任意のヒントによる解の個数を見積もることが可能である．本節で用いる方法は [4] において魔法陣の個数を見積もる方法とほぼ同じ方法になっている．以下の関数を定義する．

$$Z(\beta) = \sum_{C \in B} \exp(-\beta E(C)) \tag{5}$$

ただし，β を正の実数とし，B をヒントが与えられた時の全状態数とする．$Z(0) = |B|$ は自明である．ここで

$$\lim_{\beta \to \infty} Z(\beta) = |\{C : E(C) = 0\}| \tag{6}$$

を求めることで解の見積もりができる．本稿では β を 0 から始め，$0 < \beta_1 < \beta_2 < ... < \beta_k < \beta_{k+1} < ...\beta_{n-1} < \beta_n = \bar{\beta}$ と徐々に大きくすることで，近似値の精度を上げている．本稿では 0 から 4.3 まで 32 区分している．このとき解の個数の近似値 $\bar{N}$ は

$$\bar{N} = |B| \prod_{i=0}^{31} \frac{Z(\beta_{i+1})}{Z(\beta_i)} \tag{7}$$

と示すことができる．

4 冗長なヒントの組の削減について

これまで我々はそれぞれの時点での最適なヒントをヒント集合に逐次添加して解集合を小さくしてきた．しかし，逐次添加だけでは一度悪い組み合わせのヒントが集合の中に出来てしまうと，効率的に解集合の大きさを小さくしていくことが困難であるとことを経験している．そこで，すでに出来上がったヒント集合から冗長なヒントの組を見つけ出すことを試みた．本節ではそのための方法を示す．ある問題からヒントを二つ選び，解の個数を MCMC 法により見積もる．これをヒント数 18 個の問題であれば ${}_{18}C_2 = 153$

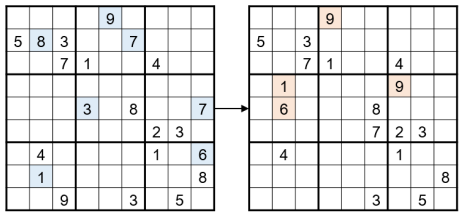


図 4: Reducing the number of hints from 21 to 18 by our method. The method took away seven digits(blue) and added four other digits(red).

通り全てに行う．次に，この中から解の個数の減少量が少なかったヒント 2 個の組み合わせを見つける．このような組み合わせを非効率と思われるヒントと定義する．さらにこれらを組み合わせて，4 個以下の非効率と思われるヒントの組を見つける．非効率と思われるヒントを削除し，逐次添加法を再度行うことによってヒントを削減する方法を試みた．ヒント数 18, 19, 21 の問題において解の個数を見積もり昇順に並べた．ヒント数 18 の問題について見積もった結果の表の一部を図 2 に示す．ただし，ヒントは左上から数えた通し番号を表しており，数字は 0 から 17 で示している．それぞれ降順から数えて 10 組についてヒントの削除を行い，再び問題の生成を試みた．結果，ヒント数 18 の問題はヒントの削減ができなかったが，図 3 に示すようにヒント数 19 の問題はヒント数 18 の問題に，ヒント数 21 の問題はヒント数 20 の問題に削減できた．ただし，青いマスは削除したヒントのマスを示しており，赤いマスは新たに追加されたヒントのマスを示している．さらにこの方法を繰り返すことで，ヒント数 21 の問題は図 4 に示すようなヒント数 18 の問題まで削減することができた．

5 まとめと今後の課題

メトロポリスのアルゴリズムを用いて解のサンプリングを行うことでヒント数 18 から 21 の問題を生成することができている．適正な問題 1 個の生成時間は 60 分程である [1]．生成するヒントの数に幅があるのは不適切なヒントを追加している可能性があると考え，MCMC を用いて解の個数の見積もりを行った．ヒント 2 組毎の解の個数を全て見積もり，その組み合わせにより 4 個以下の非効率と思われるヒントの組を削除し再生成を行った．結果，一部の問題はヒント数を 19 から 18 に，ヒント数を 21 から 18 に削減することができた．今後の課題として 5 個以上の非効率と思われるヒントの集合を削除し，再生成することで大幅にヒントの削減を行える可能性がある．

参考文献

[1] 古川湧，山本修身: シミュレイテッドアニーリングを用いたヒント数 17 の数独問題の生成．WiNF2019，A3 (2019)

[2] 古川湧，山本修身: 確率的逐次添加によるヒントの少ない数独問題の生成．情報処理学会 第 82 回全国大会，2P-02 (2020)

[3] B. Felgenhauer and F. Jarvis: Enumerating possible Sudoku grids. Technical Report pm1afj/sudoku/ (2005)

[4] Pinn, K. and Wieczerkowski, C: Number of magic squares from parallel tempering monte carlo. Int. J. Mod. Phys. C 9, 541-547, (1998)