2P-02

確率的逐次添加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，9 × 9 マスから成るペンシルパズルの一種である．現在の数独は 1979 年ごろアメリカの建築家ハワード・ガーンズが発明したとされ [1]，日本では 1984 年にパズル製作会社ニコリが数独のパズル本を出版したことをきっかけに流行した．数独という名称は「数字は独身に限る」の略であり，ナンバープレース (Number Place) とも呼ばれる．ルールの簡単さから新聞やパズル雑誌など世界中で幅広く遊ばれている．

数独の難易度はヒントが多いほど簡単になり，少ないほど難しい傾向にある．数独の問題をコンピュータに解かせる場合，バックトラックを用いることで難易度を問わず短時間で解くことができる．一方，問題を生成する場合は，すでに解いた 9 × 9 マスの問題を用意し，数字を消していくという方法があるが，多くの場合，ヒント数が多い問題になってしまう．本研究ではシミュレイテッドアニーリングと最急降下法を用いて，ヒントなしの状態からヒントを順次添加していくことでヒント数の少ない数独問題を生成することを試みた．結果，図 1 左に示すようなヒント数 17 の問題を生成することができた．以下に数独のルールと基本的な特徴を説明する．それぞれのマスには 1 から 9 のいずれかの数字が入り，すべての行と列，9 つに区分された 3 × 3 ブロック中に数の重複がない．このような並びをこのパズルの**解**と定義する．あらかじめ**ヒント**と呼ばれるいくつかの数が特定のマスに与えられている．このとき，その他のマスの中に数を入れて解を見つけるのがこのパズルの目的である．すでにあるヒントを変更することはできない．数独では通常与えられたヒントから得られる解は唯一でなければならない．本稿では与えられたヒントから得られる解が唯一であるとき，このヒントの集合を**適正な問題**と呼ぶ．適正でない問題の一例を図 1 右に示す．この問題は数独の問題に見えるが解を 149 個有しているため，適正な問題ではない．ヒントは解を唯一にする縛りと見ることができ，数独はヒントが少ないほど適正でない問題になりやすくなる．

7						3		2
		4	6		5			
3				2		1		
				4				
							9	
				3				
							6	
	9	6			8		5	

		7				3		2
		4	6		5			
3				2		1		
				4				
							9	
				3				
							6	
	9	6			8		5	

図 1: A problem with seventeen hints obtained by our method(left). An improper problem which has 149 different solutions(right). The only difference of those figures is the position of 7.

*Generation of Sudoku problems with small number of hints by a probabilistic sequential addition method

†Yu Furukawa Meijo University

‡Osami Yamamoto Meijo University

適正な問題のヒント数は 17 以上であることが知られている [2]．本稿ではヒントの数が 17 の適正な数独問題をシミュレーテッドアニーリング (SA) [3] による確率的な方法を用いて効率的に作り出す方法について述べる．

2 ヒントの生成方法について

ここではヒント数 0 の非適正問題からスタートして 1 ずつヒント数を増やしてヒント数 17 の適正問題を生成することを試みる．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．ある問題のヒントが $H_m = \{(p_1, h_1), (p_2, h_2), \ldots, (p_m, h_m)\}$ であるとする．ただし p_i はパズル上の位置（座標）であり，h_i はそこに置かれる数字であるとする．これに対応する解を S_m とする．$H_m = \phi$ のとき解集合は約 $S_m = 6.67 \times 10^{21}$ である [4]．このとき，H_m にヒント (p_{m+1}, h_{m+1}) を添加することでできる限り S_m を小さくすることを考える．あるヒントの集合 H_m を適正でない問題とする．これから解 S_m を求め，全てのマスに出現した数字の回数を求める．このとき最も出現回数の少なかった数字とマスの組み合わせを新たなヒントとして追加した集合を H_{m+1} とする．このような**最急降下法**を用いる．

3 バックトラックによる効率的な解の探索

前節のように解から新たなヒントを生成するには H_m に対する S_m を全て求めることが望ましい．数独の解き方の一つとして，入る可能性のある数字の個数 (候補数字) が少ないマスから探す方法がある．図 1 左をこの方法で解くと探索ノード数は 468，探索時間 0.114 秒となる一方でマスを順に探索する方法では探索ノード数は 829020，探索時間 0.006 秒となる．これは探索ノード数をおよそ 1/1800，探索時間を 1/19 ほどにしている．

ヒントがある程度多い場合には解集合は小さくなり，その個数は本節のバックトラックで求めることができ，その効率はそれほど悪くない．一方，ヒント数が少ない場合には，解集合の個数は膨大となり，バックトラックですべてを数え上げることが困難となる．本研究では次節で示すように解集合の要素のサンプリングにシミュレイテッドアニーリング（SA）を利用する．

4 SA による解の個数の見積もり

4.1 SA とは

シミュレイテッドアニーリングとは，ある状態からその近傍状態へと遷移させ続け，より良い状態に遷移するように内部の温度変数を変化させる，最適化問題を解くための方法である．SA では一般に遷移した状態を評価関数で評価し，評価が良くなった場合は遷移させ，悪くなった場合はある確率で遷移させない．温度変数が高い場合はこの悪くなった場合に遷移させる確率が高く設定される．具体的にこの確率を定めるためにメトロポリスのアルゴリズムを用いて，ボルツマン因子を定める．

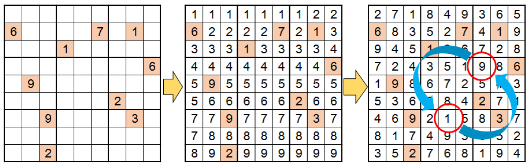

図 2: Finding a solution of a Sudoku problem by SA.

4.2 メトロポリスのアルゴリズムとボルツマン因子

メトロポリスのアルゴリズムを用いることで評価の高い状態の出現割合を高くすることができる．状態空間 C から評価値への関数 f が与えられているとき，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 を全パターンの集合とする．ある確率分布 $C(j|i)$ が与えられる．これに基づいて状態 C_i からその近傍状態 C_j へ遷移する確率と遷移せず留まる確率をそれぞれ

$$\begin{cases} P(C_{i \to j}) = f(C_i)/f(C_j) & (f(C_i) < f(C_j)) \\ P(C_{i \to i}) = 1 - P(C_{i \to j}) & (f(C_i) < f(C_j)) \end{cases} \tag{2}$$

とすれば良い．ボルツマン因子は評価値から評価関数 f をとるための一つの方法である．状態 C に対する評価を $E(C)$，逆温度関数を β とする．このとき，

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

とする．β の値は SA の温度変数に相当しており，この値によって $f(C)$ の値が高い割合と低い割合を調整する．ここでは経験的に最も計算時間短かった $\beta = 3.1$ に設定している．$E(C)$ はエネルギー関数であり，状態 C がどれほど解から遠いかを示す．$E(C) = 0$ になれば，数の配置は解となる．

4.3 SA による解の生成

数独の解を求めるためにはまず，図 2 左のような適正でない問題を用意し，同図中央のように数字を入れる．次に SA を用いて状態エネルギー関数が低くなるように数字を入れ替える．図 2 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．SA で求める解の個数は最小の組み合わせ (p, h) が経験的に全体の 2〜8 パーセント程であることから $1,000$ 個の解をサンプリングする．このような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．m がある程度大きくなり解集合が小さくなると，すべての解をバックトラックで求めることができるようになる．この場合にはサンプリングをバックトラック法に切り替えて同様の処理を行う．ここでは $m \geq 14$ でバックトラック法を用いた．

解を求めるとき，ヒントが多い場合はバックトラックによって解の列挙が可能である．ヒントが少なく，解が膨大である場合は SA によって解を 1,000 個サンプリングし，次のヒントを決める．

5 計算時間と生成した問題

ヒントなしの状態から探索を始め，約 7 分で m が 1 ずつ増えて，適当な回数繰り返すことで適正問題が一つ得られた [1]．適正

[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)．

	3							
							6	
8				9	6		5	
			7			2		3
5		6			4			
	2		3	4				1
	4							
							9	

図 3: A point-wise symmetric problem with 18 hints obtained by our method.

問題を一つ生成するのに単一スレッドで平均 90 分ほど要した．この操作を繰り返し 40 日間続けてヒント数 17 の適正問題が 1,325 個得られた．このうち，マスと数字が完全に一致している同一の問題を除くと，生成できた問題は 1,192 個となった．この場合，行列やブロックの入れ替えおよび数字の入れ替えによって同一になる問題は区別していない．また，ヒント数 18 以上 21 以下の適正問題は 23,943 個得られた．ヒント数 17 の適正問題は約 19 個に 1 個の割合で生成された．図 1 左に生成されたヒント数 17 の適正問題の一つを示す．また，SA を用いて行，列，ブロックの交換を行いヒントの配置が対象的になるよう試みた結果，図 3 のような点対象な問題を生成することができた．

6 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たシミュレイテッドアニーリングとバックトラックを用いて数独の解を列挙し，適切なヒントを追加する最急降下法を用いて約 90 分かけて最小ヒント数であるヒント 17 の問題を生成した．さらにこの問題を SA を用いて行，列，ブロックの入れ替えとヒントを追加することによりヒント 18 かつ配置が点対象となっている問題を生成した．今後の課題として，メトロポリスのアルゴリズム以外のアルゴリズムを使った解の探索や，より解の収束が早いヒントを探索する手法の模索する必要がある．また，16×16 の問題で同じようにヒントの少ない問題が生成できるか試みる．今回生成した問題は中級者程度の問題だったが，人間が解くのが難しい問題の生成も課題である．

参考文献

[1] Jean-Paul Delahaye: The Science behind SUDOKU. *Scientific American*, June, 2006, pp. 80-87 (2006)

[2] G. McGuire, B. Tugemann, 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] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi: Optimization by simulated annealing. *Science*, New Series, Vol. 220, No. 4598, pp. 671-680 (1983)

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