本質的に異なる数独解盤面の列挙と番号付け

井上真大†　　奥乃博‡

京都大学 工学部 情報学科†　京都大学 大学院情報学研究科 知能情報学専攻‡

1. はじめに

「数独（ナンバープレイス）」は，かつてより日本において親しまれてきたペンシルパズルの一種で，昨今ではその世界的な人気も非常に高く「Sudoku」の名で親しまれており，2006 年には世界大会が行われるまでになった．

これほどにまで高い人気を誇るようになった理由としては，紙と鉛筆さえあればでき，それでいて初心者から上級者まで楽しめる幅広い難易度を持つという数独の性質にあるだろう．

また完成された数独の盤面はそれ自体一定の規則を持つものなので美しい．

この完成された盤面の事を解盤面と呼び，数独の解盤面は全部で 6670903752021072936960≒6.7 $\times 10^{21}$個ある事が分かっている．[1]

この量は計算機から見ても膨大で，全解探索以外有効な手法が見つかっていない問題において探索空間の削減が求められてきた．例えば，数独の問題として成立する最小ヒント数はいくつかという最小ヒント問題などがそれにあたる．

そこで性質の変わらない解盤面を一くくりにして性質の異なる解盤面だけを寄せ集めた本質的に異なる解盤面というものが考え出された．本質的に異なる解盤面は 5472730538≒5.4$\times 10^{9}$ 個存在し[2]，探索空間は約 10^{-12} 倍にまで縮小される．

今回我々は，この本質的に異なる全解盤面の列挙とそれに対して一意に番号付けを行った．

また，これは，数独全解盤面を一意に番号付けした戸神氏の研究において今後の課題として残されたものでもある．[3]

2. 数独のルール

数独とは図 1 にある様な 9x9 の 81 マスに以下の様な規則に則って 1-9 までの数字を入れていくパズルである．

- 一つのマスには一つの数字が入る
- どの行も列も 1 から 9 の数字が一回ずつ現れる
- 九つあるどの 3x3 のブロックも 1 から 9 の数字が一回ずつ現れる

これらの規則に従って全てのマスに数字が入った盤面を解盤面と呼び，任意個のマスに最初から数字が入っており規則に従って数字を入れるとただ一つの解盤面に帰着できる様な盤面を問題と呼ぶ．図 1 にその例を示す．

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

問題

一意に帰着 ⇒

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

解盤面

図 1

3. 本質的に異なる解盤面

解盤面に対して以下の操作およびこれらを組み合わせた操作を行ってできる解盤面を本質的に同じ解盤面と呼ぶ．

- 数字の入れ替え
- 盤面の転置
- 1-3 行,4-6 行,7-9 行の行ブロックの入れ替え
- 1-3 列,4-6 列,7-9 列の列ブロックの入れ替え
- 1,2,3 行の入れ替え
- 4,5,6 行の入れ替え
- 7,8,9 行の入れ替え
- 1,2,3 列の入れ替え
- 4,5,6 列の入れ替え
- 7,8,9 列の入れ替え

本質的に同じ解盤面同士は全く同じ問題集合を持つ．つまりその解盤面が答えとなる問題の総数は等しく，また最小の初期配置の数も等しい．ゆえに，前述した最小ヒント問題を考える際には本質的に同じ解盤面のうち一つだけを調べれば十分なのである．この本質的に同じという関係は同値関係であり，完全代表系つまり本質的に同じ解盤面の集合からそれぞれ一つずつ代表となる盤面を選んでできた集合が本質的に異なる解盤面の集合である．

4. 本質的に異なる解盤面の列挙

まず上で述べた代表となる解盤面の選び方であるが，解盤面を 81 桁の整数として見た時に最小となるものを選んでいる．盤面を 81 桁の整数として見ることは計算機で数独を扱う際の基本

Enumeration and Numbering of Essentially Different Sudoku Grids: *Masahiro Inoue and Hiroshi G. Okuno (Kyoto University)*

で図２で数字の小さいマスほど高い位として扱う．

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81

図 2

さて列挙の方法であるが，まず全解盤面を列挙しそれぞれに対して上述した操作を施し，より小さな解盤面が存在すれば破棄していく．そうすると残った盤面は代表解盤面の集合となっているのでこれで本質的に異なる解盤面の列挙となるのである．
しかしながら，前述したとおり全解盤面は約 6.7 $\times 10^{21}$ 個存在しそれら全てに対して走査するのは現実的ではない．以降で走査する必要のない解盤面を求めて列挙の高速化を図る事について述べる．

4.1. 高速化 1

代表盤面は必ず"123456789"で始まっているはずである．なぜなら全ての解盤面は数字の入れ替えによって"123456789"で始まる様にでき，それより小さくはできないからである．従って全解盤面の内，"123456789"で始まっている盤面のみ列挙すればよいことが分かり，探索区間は 1/9!に縮小できる．

4.2. 高速化 2

上の高速化 1 を盤面の最初の 27 マスにまで発展する事を考える．つまり解盤面に含まれ得る最初の 27 マスに対して，上述した操作のうち残り 54 マスをドントケアとしても行える操作，例えば数字の入れ替えや 1,2,3 列の入れ替えなど，を行ってより小さい 27 マスがでてきたならば，元の 27 マスを含む解盤面は全て代表盤面となり得ないので列挙する必要がない．最終的には代表盤面に含まれ得る 27 マスは 401 個にまで絞る事ができ，探索空間はおよそ 1/20000 となる．

4.3. 高速化 3

図２の 28,37,46,55,64,73 の 6 マスに着目すると，4,5,6 行の入れ替え，7,8,9 行の入れ替え，4-6 行,7-9 行の入れ替えより，代表盤面では (28)<(37)<(46)かつ(55)<(64)<(73)かつ(28)<(55)が成り立つ事が分かる．ここで，()はそのマスに入る数字を表す．同様にこれが成り立っていない解盤面は列挙しなくてよく，探索空間は約 10/6!になる．

以上を総合し，実際の列挙ではまず最初の 27 マスで解盤面をグループ分けし，その後それぞれについて高速化３で述べた条件を満たす (28),(37),(46),(55),(64),(73)の組み合わせで更にグループ分けして（つまり計 33 マスでグループ分けしている），そしてそれぞれについて最初に述べた様にそれを含む全解盤面を列挙して，代表解盤面以外を取り除いていく．実際に列挙した解盤面は一つ当たり 24byte（計 130Gbyte）に圧縮して保存した．なおここで解盤面の列挙には，最も高速である，数独の問題を厳密カバー問題として見て解く方法を用いている．

5. 番号付け

前項で述べたグループ分けに従って番号付けを行う．このグループは全部で 3541 グループ存在し，各グループをその 33 マスの昇順で並べ，更にそのグループ内の解盤面を昇順に並べ先頭から順に番号付けを行う．結果，1 番の解盤面は 123456789456789123789123456214835697367291845598647312632918574845372961971564238，最後である 5472730538 番の解盤面は 123456789457893612986217354274538196531964827698721435342685971715349268869172543 となる．

6. 終わりに

本稿で述べた数独の全解探索空間の削減により，最小ヒント問題の解決に兆しが見えた様に思えるが，実際には依然探索空間は大きく現実時間では解決できそうにないのが現状である．
今後の課題として，更なる探索空間の削減が求められる．

参考文献

[1] Felgenhauer and Jarvis, “Enumerating possible Sudoku grids”, http://www.afjarvis.staff.shef.ac.uk/sudoku/sudoku.pdf
[2] Russell and Jarvis, “Sudoku enumeration”, http://www.afjarvis.staff.shef.ac.uk/sudoku/sudgroup.html
[3] 戸神星也(2006), “数独の解生成と解に対する番号付け”, 東京工業大学理学部情報科学科卒業論文, http://doorgod.org/sudoku/doc/SUDOKU_Index_abst.pdf