

数独問題のイデアル構造による階層付けとブーリアン
グレブナ基底による並列計算

A Mathematical Hierarchy of Sudoku Puzzles and its
Parallel Computation by Boolean Gröbner Bases

井上 秀太郎

SHUTARO INOUE

東京理科大学理学部数理情報科学科

DEPARTMENT OF MATHEMATICAL INFORMATION SCIENCE, TOKYO UNIVERSITY OF SCIENCE *

野口 慎司

SHINJI NOGUCHI

東京理科大学大学院理学研究科数理情報科学専攻

DEPARTMENT OF MATHEMATICAL SCIENCE FOR INFORMATION SCIENCES,
GRADUATE SCHOOL OF SCIENCE, TOKYO UNIVERSITY OF SCIENCE †

和氣 宏明

HIROAKI WAKI

東京理科大学大学院理学研究科数理情報科学専攻

DEPARTMENT OF MATHEMATICAL SCIENCE FOR INFORMATION SCIENCES,
GRADUATE SCHOOL OF SCIENCE, TOKYO UNIVERSITY OF SCIENCE ‡

佐藤 洋祐

YOSUKE SATO

東京理科大学理学部数理情報科学科

DEPARTMENT OF MATHEMATICAL INFORMATION SCIENCE, TOKYO UNIVERSITY OF SCIENCE §

*sinoue@rs.kagu.tus.ac.jp

†j1413612@ed.tus.ac.jp

‡j1413625@ed.tus.ac.jp

§ysato@rs.kagu.tus.ac.jp

1 はじめに

我々が行ってきた数独の難易度判定の研究では, naked single, hidden single 等の標準的なストラテジのみで解ける数独のみを扱っていた. そのために, 高度なストラテジを必要とする数独の難易度判定はできないという問題を抱えていた. その中で [1] は高度なストラテジを必要とする数独の難易度判定方法を提案している. 本稿では, 高度なストラテジを必要とする数独の難易度判定の詳細な検証結果を報告する. また, 難易度判定アルゴリズムの並列計算プログラムについての報告も記載する.

2 数独とは

数独とは「 9×9 の 81 枠の各行, 列と 3×3 の 9 つのブロックそれぞれに 1~9 の数字が一つずつ入る」というルールで, 空白の枠に数字を入れるゲームである.

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

上記の数独の 5 行 1 列目に着目してみると, 既に同じ行, 列, ブロックに 2 以外の数字が入っている. この枠に 2 以外の数字は当てはまらず, 2 が入ることが分かる. 他の空枠にも同様に当てはめていき, 空枠が全て埋まれば終了となる.

3 標準的なストラテジによる数独の解法

3.1 ブーリアングレブナ基底

定義 1

すべての要素が冪等であるような, 単位元をもつ可換環 \mathbb{B} をブール環と呼ぶ.

このとき重要な性質として, $X \in \mathbb{B}$ に対して $X + X = 0 \iff X = -X$ がある.

定義 2

ブール環 \mathbb{B} を係数とする多項式環 $\mathbb{B}[\bar{X}]$ のイデアル $\langle X_1^2 + X_1, \dots, X_n^2 + X_n \rangle$ による剰余環をブール多項式環とよび, $\mathbb{B}(\bar{X})$ で表す ($\bar{X} = X_1, \dots, X_n$).

定義 3

単項式順序を固定する. イデアル $I \subset \mathbb{B}(\bar{X})$ の有限部分集合 $G = \{g_1, \dots, g_t\}$ がブーリアングレブナ基底であるとは $\langle LM(G) \rangle = \langle LM(I) \rangle$ を満たすことと定義する ($LM(G) : G$ の先頭単項式).

ブール環上の計算について, 集合 S に対し, $\mathcal{P}(S) = \{A | A \subseteq S\}$ は以下の演算でブール環になる.

定義 4

$X, Y \in \mathcal{P}(S)$ とする. 乗法を $X \cdot Y = X \cap Y$, 加法を $X + Y = (X \cap \neg Y) \cup (Y \cap \neg X)$ と定義する. ただし $\neg X$ は X の補集合とする.

3.2 ブーリアングレブナ基底による数独の解法

まず, 各枠に変数 X_{ij} ($1 \leq i, j \leq 9$) を以下のように割り当てる.

X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}
X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}	X_{27}	X_{28}	X_{29}
X_{31}	X_{32}	X_{33}	X_{34}	X_{35}	X_{36}	X_{37}	X_{38}	X_{39}
X_{41}	X_{42}	X_{43}	X_{44}	X_{45}	X_{46}	X_{47}	X_{48}	X_{49}
X_{51}	X_{52}	X_{53}	X_{54}	X_{55}	X_{56}	X_{57}	X_{58}	X_{59}
X_{61}	X_{62}	X_{63}	X_{64}	X_{65}	X_{66}	X_{67}	X_{68}	X_{69}
X_{71}	X_{72}	X_{73}	X_{74}	X_{75}	X_{76}	X_{77}	X_{78}	X_{79}
X_{81}	X_{82}	X_{83}	X_{84}	X_{85}	X_{86}	X_{87}	X_{88}	X_{89}
X_{91}	X_{92}	X_{93}	X_{94}	X_{95}	X_{96}	X_{97}	X_{98}	X_{99}

次に $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $\mathbb{B} = \mathcal{P}(S) = \{A | A \subseteq S\}$ とし, 数独のルールを以下のように数式化を行う.

- 1) 同じ行, 列, ブロック内の枠に異なる数字が入る

$$X_{ij}X_{i'j'} = 0 \quad (= \emptyset) \quad \text{for } (i, j) \neq (i', j')$$

例: 1 行目: $X_{11}X_{12} = 0, X_{11}X_{13} = 0, \dots, X_{12}X_{13} = 0, \dots, X_{18}X_{19} = 0$

- 2) 同じ行, 列, ブロック内の枠に 1~9 の数字が入る

$$\sum_{i,j} X_{ij} + 1 = 0 \quad (1 = \{1, 2, \dots, 9\})$$

例: 1 番左上ブロック: $X_{11} + X_{12} + X_{13} + X_{21} + X_{22} + X_{23} + X_{31} + X_{32} + X_{33} + 1 = 0$

- 3) 初期条件 (最初に与えられている数字)

$$X + \{s\} = 0$$

例: $X_{13} + \{4\} = 0, X_{14} + \{2\} = 0, \dots, X_{97} + \{6\} = 0$

という方程式に対する単集合 (要素数 1) の解が, もとの数独の解と一致する.

3.3 標準的なストラテジ (naked single, hidden single)

表 1 は naked single の例である. この例では X_{28} の枠に 1 が当てはまることが分かる. 表 1 に対応するイデアルは $\{1, 2, 3, 4\}X_{28} + \{1\}$ を含む. $\{1, 2, 3, 4\}X_{28} + \{1\} = 0$ の解は,

$$X_{28} = \{1\} \cup p, \quad p \subseteq \{5, 6, 7, 8, 9\}$$

表 1: naked single

1		2						
			2		3		X_{28}	4
			3		2			
						1		
			4		5			

表 2: hidden single

		9						
		1						
	5	4						
3		X_{93}		6		2	8	

となるが, 単集合の解は,

$$X_{28} = \{1\}$$

のみである.

表 2 は hidden single の例である. この例では X_{93} の枠に 7 が当てはまることが分かる. 表 2 に対応するイデアルは $(1 + \{7\})X_{93}$ を含む. $(1 + \{7\})X_{93} = 0$ の解は,

$$X_{93} = \{7\}, \{\} = \emptyset$$

となるが, 単集合の解は,

$$X_{93} = \{7\}$$

のみである.

3.4 数独の解に関わる多項式について

定義 5

以下のブール多項式は,

(1) *solution polynomial* $f(X) = X + \{s\}$

(2) *semi-solution polynomial (naked single)* $f(X) = \{s_1, s_2, \dots, s_l\}X + \{s_i\} \ (i = 1, \dots, l)$

(3) *semi-solution polynomial (hidden single)* $f(X) = (1 + \{s\})X$

(4) *contradiction polynomial* $f(X) = a, X, aX + b \ (a, b \neq 0, b \text{ の要素数} > 1)$

と呼ばれ, また $X + \{s\}$, $X + \{s_i\}$ は (2), (3) の *associated solution polynomial* と呼ばれる.

(2) の semi-solution polynomial は naked single に, (3) は hidden single に対応している.

数独を解くには数独のルールと初期条件によるイデアルに対してブーリアングレブナ基底を計算する. 基底の中に semi-solution polynomial を含んでいたら associated solution polynomial に置き換えて, 再度ブーリアングレブナ基底を計算する. これを semi-solution polynomial が出なくなるまで繰り返すことで数独を解いていく. 例えば以下のように多項式の置き換えを行う.

$$(1 + \{2\})X_{11} \rightarrow X_{11} + \{2\}$$

定義 6

数独のルールと初期条件によるイデアルに含まれる *semi-solution polynomial* を *associated solution polynomial* に置き換える操作の繰り返しのみによって得られるイデアルを I とする. このとき I が *maximal ideal* ならば, I を *basic solvable ideal* と呼ぶ.

本稿において, *maximal ideal* は $\langle X_{11} + \{s_{11}\}, \dots, X_{99} + \{s_{99}\} \rangle$ のことである.

I が *basic solvable ideal* ならば, 与えられた問題が *naked single*, *hidden single* 等の標準的なストラテジのみで解ける簡単な問題であることを意味している. I が *basic solvable ideal* とならない場合, 我々はさらに高度なストラテジを必要とする. このような数独に対しては [1] で提案されたアルゴリズムを使用する.

4 高度なストラテジによる数独の解法

4.1 高度なストラテジ (XY-wing, XY-chain)

数独を解くために有効な, より高度なストラテジが存在する. その多くは, ある枠にある数字が当てはまらないことが分かるものである. 例として XY-wing, XY-chain を紹介する.

表 3: XY-wing

	¹ ₂			¹ ₃				
	² ₃			<i>a</i>				

表 4: XY-chain

		<i>b</i>			<i>d</i>			
<i>a</i>	¹ ₂	<i>a</i>	<i>a</i>	¹ ₄	<i>a</i>	<i>a</i>	<i>a</i>	<i>a</i>
		<i>b</i>			<i>d</i>			
		<i>b</i>			<i>d</i>			
<i>c</i>	² ₃	<i>c</i>	<i>c</i>	³ ₄	<i>c</i>	<i>c</i>	<i>c</i>	<i>c</i>
		<i>b</i>			<i>d</i>			
		<i>b</i>			<i>d</i>			
		<i>b</i>			<i>d</i>			
		<i>b</i>			<i>d</i>			

表 3 が XY-wing を使用できる場合の例である. まず X_{22} の枠には 1 か 2 のどちらかが当てはまる. X_{25} , X_{52} についても同様である. これら 3 つの枠を式にすると以下ようになる.

$$(1 + \{1, 2\})X_{22} = 0, (1 + \{2, 3\})X_{52} = 0, (1 + \{1, 3\})X_{25} = 0$$

XY-wing とは, この例では $a \neq 3$ つまり $X_{55} \neq \{3\}$ が分かるものである. もし $X_{55} = \{3\}$ であるならば, $X_{25} = \{1\}$, $X_{52} = \{2\}$ である. よって X_{22} の枠に当てはまる数字が無くなってしまい, $X_{55} \neq \{3\}$ が分かる.

次に $(1 + \{1, 2\})X_{22}, (1 + \{2, 3\})X_{52}, (1 + \{1, 3\})X_{25}$ を数独のルールの多項式から生成されるイデアル I_0 に加えたイデアル,

$$I = I_0 + \langle (1 + \{1, 2\})X_{22}, (1 + \{2, 3\})X_{52}, (1 + \{1, 3\})X_{25} \rangle$$

を考える.

$\{3\}X_{55} \notin I$ であるため, $\{3\}X_{55}$ を得るために以下の順に操作を行う.

1. $I + \langle X_{55} + \{3\} \rangle \ni (1 + \{1\})X_{25}, (1 + \{2\})X_{52}$
2. $I + \langle X_{55} + \{3\}, X_{25} + \{1\}, X_{52} + \{2\} \rangle \ni X_{22}$

I に $X_{55} + \{3\}$ を加えた結果 contradiction polynomial X_{22} を含み, $X_{55} + \{3\}$ は間違っていることが分かり, 以上のようにして $\{3\}X_{55}$ を得ることができた.

表4がXY-chainを使用できる場合の例である. X_{22} の枠には1か2のどちらかが当てはまる. X_{25} , X_{52} , X_{55} についても同様である. これら4つの枠を式にすると以下のようになる.

$$(1 + \{1, 2\})X_{22} = 0, (1 + \{1, 4\})X_{25} = 0, (1 + \{2, 3\})X_{52} = 0, (1 + \{3, 4\})X_{55} = 0$$

XY-chainとは, この例では $a \neq 1, b \neq 2, c \neq 3, d \neq 4$ が分かるものである. 例えば X_{24} の枠について考えてみる. もし $X_{24} = \{1\}$ であるならば, $X_{22} = \{2\}, X_{25} = \{4\}$ である. そして $X_{52} = \{3\}, X_{55} = \{3\}$ となり同じ行に同じ数字が当てはまることになってしまう. よって $X_{24} \neq \{1\}$ であることが分かる.

次に $(1 + \{1, 2\})X_{22}, (1 + \{1, 4\})X_{25}, (1 + \{2, 3\})X_{52}, (1 + \{3, 4\})X_{55}$ を数独のルールの多項式から生成されるイデアル I_0 に加えたイデアル,

$$I = I_0 + \langle (1 + \{1, 2\})X_{22}, (1 + \{1, 4\})X_{25}, (1 + \{2, 3\})X_{52}, (1 + \{3, 4\})X_{55} \rangle$$

を考える. X_{24} を例にXY-wingと同様, 以下の順に操作を行う.

1. $I + \langle X_{24} + \{1\} \rangle \ni (1 + \{2\})X_{22}, (1 + \{4\})X_{25}$
2. $I + \langle X_{24} + \{1\}, X_{22} + \{2\}, X_{25} + \{4\} \rangle \ni (1 + \{3\})X_{52}, (1 + \{3\})X_{55}$
3. $I + \langle X_{24} + \{1\}, X_{22} + \{2\}, X_{25} + \{4\}, X_{52} + \{3\}, X_{55} + \{3\} \rangle \ni \{3\}$

I に $X_{24} + \{1\}$ を加えた結果 contradiction polynomial $\{3\}$ を含み, $X_{24} + \{1\}$ は間違っていることが分かった.

4.2 br-rank

定義 7

ある枠 X に入りうる数字 $\{s\}$ に対して, 数独のルールによるイデアル I_0 に $X + \{s\}$ を加えたイデアルを $I = I_0 + \langle X + \{s\} \rangle$ とする. I に含まれる semi-solution polynomial を associated solution polynomial に置き換える操作を繰り返すことで contradiction polynomial を含むまでの操作の回数を, I_0 に対する $X + \{s\}$ の br-rank (basic refutable-rank) と呼ぶ.

前述のXY-wingの例 $X_{55} + \{3\}$ では, contradiction polynomial を含むまでに行った操作の回数は1回であった. よって $X_{55} + \{3\}$ の br-rank は1である. XY-chainの例 $X_{24} + \{1\}$ では, contradiction polynomial を含むまでに行った操作の回数は2回であったため, $X_{24} + \{1\}$ の br-rank は2である.

一般的にXY-wingよりXY-chainの方が難しいとされている. br-rankにおいてもXY-wingよりXY-chainの方がランクが高くなった.

4.3 s-rank

定義 8

semi-solution polynomial を associated solution polynomial に置き換える操作によって得られるイデアル I に対して, br-rank 1 の情報を加える操作を行う. br-rank 1 の多項式による情報では解けない場合は, 再び I に対して br-rank 1, 2 の多項式の情報加える操作を行う. これを数独が解けるまで繰り返し, 必要になった最大の br-rank を, その問題の難易度 s-rank (strategy-rank) と呼ぶ. basic solvable ideal は s-rank 0 とする.

数独の答えが出るまでに必要になった多項式の br-rank が低ければ低いほど, その問題の難易度は低く, 逆に高ければ高いほど難易度が高い. 低い br-rank による情報で解けるかどうかを順番に試し, どれだけ高い br-rank による情報を必要としたかで s-rank を求める.

また本研究では s-rank による階層付けに加えて, s-rank に対する更に詳細な階層付けを提案する.

定義 9

s-rank i ($1 \leq i$) の問題に対して br-rank k ($1 \leq k \leq i$) による情報を加える操作を行った回数 j によって更に階層付けを行い, s-rank $i-j$ と表す.

s-rank とこの定義による計算アルゴリズムを以下のように提案する.

[アルゴリズム]

step0 $i = 0, j = 0$ とし, 数独のルールと初期条件によるイデアルを I とする.

step1 I に含まれる semi-solution polynomial を associated solution polynomial で置き換える操作のみによって得られるイデアル I' を計算する.

- I' が maximal ideal なら終了し s-rank $i-j$ (s-rank $0 - 0$ は s-rank 0 とする.)
- そうでない場合は step2 へ.

step2 br-rank k ($1 \leq k \leq i$) のすべての多項式 $X + \{s\}$ に対して $\{s\}X$ をイデアル I' に加える操作をし, I'' とする.

- br-rank k による多項式を得られた場合
 $j + 1 \rightarrow j, I'' \rightarrow I$ とし step1 に戻る.
- br-rank k による多項式が得られなかった場合
 $i + 1 \rightarrow i, j = 0, I'' \rightarrow I$ とし step1 に戻る.

このアルゴリズムを用いて実際に s-rank $i-j$ の計算を行った.

5 難易度判定プログラム

5.1 並列計算プログラムについて

難易度判定には非常に時間のかかる問題がいくつかある. 難易度判定を行う上で必要な br-rank の計算は, 入りうる候補 (100 程度) が一つずつ入ると仮定したイデアルに対して計算を行う. そのため 100 程度の独立した計算となり, この計算を並列計算で行い, 計算量の多い問題に対処した.

実験環境は以下の通り.

- ubuntu 14.04 LTS 64bit
- Risa/Asir OpenXM
- Intel(R) Core(TM) i7-3970X CPU @ 3.50GHz
- memory 64GB

5.2 実行結果例

次の例題は s-rank 1-2 に該当する問題で、並列計算を行わずに難易度判定を行った場合 91.57 秒の計算時間がかかった。この問題を並列計算させることで、計算時間は 20.85 秒となり、約 4.4 倍の速度で難易度を判定することができた。

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

5.3 実行結果

使用した数独の問題集には難易度が設定されており、難易度毎に 105 問の問題がある。並列計算プログラムで計算した s-rank と問題集の難易度との関係は次の表 5 のようになった。

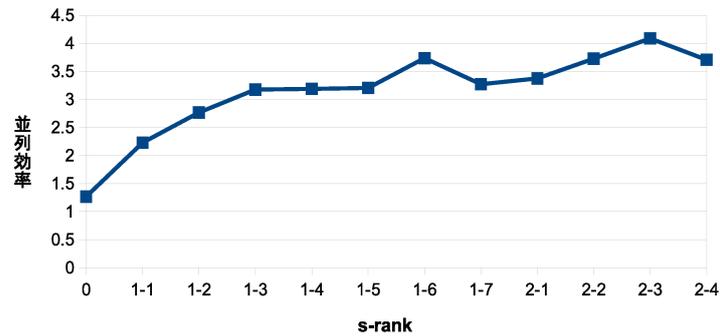
表 5: s-rank と問題集の難易度の比較表

	0	1-1	1-2	1-3	1-4	1-5	1-6	1-7	2-1	2-2	2-3	2-4
初級	105	0	0	0	0	0	0	0	0	0	0	0
中級	105	0	0	0	0	0	0	0	0	0	0	0
上級	84	16	3	1	1	0	0	0	0	0	0	0
超上級	58	39	6	0	1	0	0	0	1	0	0	0
難問	40	44	12	4	2	0	0	0	1	2	0	0
超難問	17	49	17	7	4	4	0	2	2	1	2	0
限界	13	43	15	12	3	1	1	2	6	7	1	1

表より、問題集の難易度と s-rank との相関関係を確認することができる。

また、難易度 (s-rank) と並列効率 (並列計算なしでの全体の計算時間 ÷ 並列計算を行った場合の全体の計算時間) の関係は以下ようになった。

s-rank と並列効率



グラフから、s-rank が高い問題では台数効果を得られたことが分かり、3 倍から 4 倍の高速化を実現した。ここで並列効率の最も良かった問題は s-rank 2-2 であり、また最も計算時間の長い問題であった。その計算時間は並列計算なしで 2324 秒、並列計算ありで 460 秒であり、約 5 倍の速度で難易度を判定できた。

5.4 s-rank ∞ について

以下の問題はフィンランドの数学者 Arto Inkala が作成した世界一難しいとされる問題で、s-rank ∞ に該当する問題である。s-rank ∞ とは、すべての br-rank の多項式を加えて計算しても解くことが出来ない問題であることを指す。この問題は我々のアルゴリズムを使用しても新たに枠を埋めることは出来なかった。

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

6 まとめ

我々は高度なストラテジを必要とする数独に対し、プーリアングレブナ基底を使用した難易度判定アルゴリズムを検証し、市販されている数独の問題集の難易度との相関性を確認した。さらに、難易度判定の並列計算プログラムを実装し、非常によい実験結果が得ることができた。

参 考 文 献

- [1] Shutaro Inoue, Yosuke Sato, "A Mathematical Hierarchy of Sudoku Pazzles and Its Computation by Boolean Gröbner Bases", Artificial Intelligence and Symbolic Computation, Proceedings of 12th International Conference, AISC2014, Lecture Notes in Artificial Intelligence, Springer, (2014), pp.88-98
- [2] Yosuke Sato, Shutaro Inoue, Akira Suzuki, Katsusuke Nabeshima, "Boolean Gröbner Basis", Journal of Symbolic Computation, 46(2011), pp.622-632
- [3] Shutaro Inoue, Akira Nagai, "On the Implementation Boolean Gröbner Basis", The Joint Conference of ASCM 2009 and MACIS 2009, (2009), pp.58-62
- [4] Yusuke Sato, Yosuke Sato, Shutaro Inoue, "数独の難易度判定のためのブーリアングレブナー基底の並列計算について" 数理解析研究所講究録 1843, (2013), pp.28-37
- [5] 郷内邦義著 クロスワード編集部編 「脳を鍛える数字パズル ナンプレ 初級編」(2006) 廣濟堂出版
- [6] 郷内邦義著 クロスワード編集部編 「脳を鍛える数字パズル ナンプレ 中級編」(2006) 廣濟堂出版
- [7] 郷内邦義著 クロスワード編集部編 「脳を鍛える数字パズル ナンプレ 上級編」(2006) 廣濟堂出版
- [8] 郷内邦義著 クロスワード編集部編 「脳を鍛える数字パズル ナンプレ 超上級編」(2007) 廣濟堂出版
- [9] 郷内邦義著 クロスワード編集部編 「脳を鍛える数字パズル ナンプレ 難問編」(2007) 廣濟堂出版
- [10] 郷内邦義著 クロスワード編集部編 「脳を鍛える数字パズル ナンプレ 超難問編」(2008) 廣濟堂出版
- [11] 郷内邦義著 クロスワード編集部編 「脳を鍛える数字パズル ナンプレ 限界編」(2008) 廣濟堂出版