

Boolean Groebner 基底を用いた数独パズルの数学的難易度指標の相関について

On the correlation of some mathematical indicators of difficulty level of Sudoku puzzles in terms of Boolean Groebner bases

東京電機大学工学部 中野哲夫 *1

TETSUO NAKANO

FACULTY OF SCIENCE AND ENGINEERING, TOKYO DENKI UNIVERSITY

東京電機大学大学院理工学研究科 進藤未来 *2

MIKU SHINDOU

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING,

GRADUATE SCHOOL OF TOKYO DENKI UNIVERSITY

東京電機大学大学院理工学研究科 吉原元 *3

TSUKASA YOSHIHARA

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING,

GRADUATE SCHOOL OF TOKYO DENKI UNIVERSITY

Abstract

The Inoue algorithm is a fundamental mathematical method for solving Sudoku puzzles by Boolean Groebner bases. We have been investigating the CII algorithm, which is a refined form of Inoue algorithm. Both of these algorithms are the mathematical version of "Try and Error method" for solving the puzzles by humans, and have been applied successfully to the evaluation of difficulty level of the puzzles.

In this note, we study the correlation of several mathematical indicators of difficulty level such as SMYI, MDSL, s_∞ -rank and LAC by experiments. Especially, we have confirmed that the puzzles with infinite s_∞ -rank can be well hierarchically classified according to the difficulty level by LAC.

1 概要

ブーリアングレブナ基底を用いて数独パズルを解く数学的解法として、井上アルゴリズム ([4]) がある。井上アルゴリズムは人間のトライアンドエラー方式の解法と似ているため、数独パズルの難易度判定に用いられて成果を上げている。

*1 〒 350-0394 埼玉県比企郡鳩山町石坂 E-mail: tnakano@mail.dendai.ac.jp

*2 〒 350-0394 埼玉県比企郡鳩山町石坂 E-mail: 21rmu08@ms.dendai.ac.jp

*3 〒 350-0394 埼玉県比企郡鳩山町石坂 E-mail: 21rmu15@ms.dendai.ac.jp

井上アルゴリズムに基づく数独パズルの数学的な難易度指標には、大きく分けて2つある。すなわち、
(1) 佐藤先生、井上先生のグループによる **b-rank**, **s-rank** ([5]) およびそれらから派生する rank (s_∞ -rank など)

(2) 著者達のグループが提唱した CII アルゴリズムを用いる指標 **CII** (Computable Inoue Invariant, 計算可能井上不変量, [6]-[8], [14]) と **SMY 不変量** (Shindou-Mikoshiba-Yoshihara 不変量, SMYI, [10], [11]) の2種類である。

本稿では、超難問について、CII アルゴリズムに基づいて定義される**不変量 SMYI** と、**MDSL** ([10], [11]), s_∞ -rank (s_∞ -rank 有限の場合), **LAC** (s_∞ -rank 無限の場合) のそれぞれとの相関について実験的に検証した結果を報告する。特に、 s_∞ -rank 無限の場合、**SMYI** と **LAC** は強い負の相関があるという興味深い結果を得た。

最も難しい s_∞ -rank 無限のパズルの例は従来3つしか知られていなかったため、このクラスのパズルの階層付けはほとんど研究されていなかった。しかし、今回、約150問の s_∞ -rank 無限のパズルの例が新たに見つかり、実験により **LAC** がこのクラスのパズルの難易度による階層付けの良い指標になることが検証できた。

本稿の構成は次の通りである。まず、2節で Boolean Groebner 基底等の数学的準備をしてから、3節で基本的な井上アルゴリズムと CII アルゴリズムの復習をする。次に4節で戦術閉包と s_∞ -rank について確認してから、最後に5節で数学的難易度指標の相関に関する主要結果を報告する。

本研究では、数独の実験でよく用いられる郷内氏の7冊の標準的問題集735問 ([2]) および、インカラ博士の作成した超難問12問 ([3])、ターレック氏の超難問634問 ([16]) を実験サンプルとして用いた。ターレック氏の634問は今回新たに Magma 形式の入力で採録したものである。計算に用いるプログラムは、数式処理システム Magma ([1]) 上で開発・実装した。

2 準備

本節では、3節以後で必要となる基礎事項の準備をする。詳しくは、[12], [13], [15] 等を参照せよ。

2.1 ブーリアングレブナ基底の定義と基本性質

定義 1 (ブール環)

A を単位元をもつ可換環とする。任意の $a \in A$ について $a^2 = a$ が成立するとき、 A を**ブール環**という。

$\mathbb{F}_2 = \{0, 1\}$ を2元体、 \mathbb{F}_2^m を成分ごとの加法、乗法について可換環とみなすと、 \mathbb{F}_2^m はブール環となる。次に、 \mathbb{F}_2^m に係数をもつブール多項式環を定義する。

定義 2 (ブール多項式環)

$\mathbb{F}_2^m[x] = \mathbb{F}_2^m[x_1, \dots, x_n]$ を \mathbb{F}_2^m に係数にもつ n 変数多項式環とする。 $\mathbb{F}_2^m[x]$ 自体はブール環ではない。そこで

$$\mathbb{F}_2^m(x) := \mathbb{F}_2^m[x_1, \dots, x_n] / \langle x_1^2 - x_1, \dots, x_n^2 - x_n \rangle$$

とおくと、 $\mathbb{F}_2^m(x)$ はブール環となる。これを**ブール多項式環**と呼ぶ。

次に、 $\mathbb{F}_2^m[x] = \mathbb{F}_2^m[x_1, \dots, x_n]$ のグレブナ基底を説明する。 $\mathbb{F}_2^m[x]$ でも、通常の体を係数とする多項式と同様に、余りのある割り算アルゴリズムが(少しの修正をした上で)成立する(余りの条件が少し異なる)。

定義 3 (グレブナ基底)

$\mathbb{F}_2^m[x]$ 上に単項式順序を 1 つ与え、 $I \subset \mathbb{F}_2^m[x]$ をイデアルとする。 $G := \{g_1, \dots, g_s\}$ を I の有限部分集合とし、 $LT(I) = \{LT(f) \mid f \in I, f \neq 0\}$ を I の元の先頭項の集合とする。このとき、

$$\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_s) \rangle$$

が成り立つならば、 G は I のグレブナ基底という。

定義 4 (極小グレブナ基底, 簡約グレブナ基底)

(1) G をイデアル I のグレブナ基底とする。任意の $g \in G$ について、 $LT(g) \notin \langle LT(G \setminus \{g\}) \rangle$ が成立するとき、 G を極小グレブナ基底という。

(2) G を極小グレブナ基底とする。任意の $g \in G$ に対して、 $\bar{g}^{G \setminus \{g\}} = g$ が成り立つとき、 G を簡約グレブナ基底という。ここで、 $\bar{g}^{G \setminus \{g\}}$ は、 g を $G \setminus \{g\}$ で割った余りである。

体上の多項式環の場合と違って、簡約グレブナ基底は一意に定まらないので、次の階層的グレブナ基底が必要である。

定義 5 (階層的グレブナ基底)

G をイデアル $I \subset \mathbb{F}_2^m[x]$ の簡約グレブナ基底とする。もし任意の $f, g \in G (f \neq g)$ について $LM(f) \neq LM(g)$ ならば、 G は階層的グレブナ基底と呼ばれる。

任意のイデアル $I \subset \mathbb{F}_2^m[x]$ は、唯一の階層的グレブナ基底をもつ。これは体上の係数の場合と似た（しかし、少し異なる）Buchberger アルゴリズムによって証明できる。

以上の $\mathbb{F}_2^m[x]$ のイデアルのグレブナ基底の定義および性質は、ブール多項式環 $\mathbb{F}_2^m(x)$ のイデアルについても全く同様に成立する。例えば、

定義 6 (ブーリアングレブナ基底)

$\mathbb{F}_2^m[x]$ に単項式順序を 1 つ与えると、自然に、 $\mathbb{F}_2^m(x)$ の（標準単項式の集合に）単項式順序が誘導される。 I を $\mathbb{F}_2^m(x)$ のイデアルとし、 $G := \{g_1, \dots, g_s\}$ を I の有限部分集合とする。このとき、 $\langle LT(I) \rangle = \langle LT(g_1), \dots, LT(g_s) \rangle$ が成り立つならば、 G を I のブーリアングレブナ基底という。

階層的グレブナ基底の存在と一意性も、ブール多項式環で成立する。ブール多項式環のイデアルの具体的な計算は、すべてブーリアングレブナ基底を用いて遂行される。

2.2 数独パズルの数学的定式化

次に数独パズルのルールをブール多項式環を用いて定式化する。 x_{ij} を i 行 j 列のマスに対応する変数とし、 \mathbb{F}_2^9 上の 81 変数 x_{11}, \dots, x_{99} のブール多項式環

$$\mathbb{F}_2^9(x) = \mathbb{F}_2^9(x_{11}, \dots, x_{99})$$

を考える。 $e_1 := (1, 0, \dots, 0), \dots, e_9 := (0, \dots, 0, 1), 0 = (0, \dots, 0), 1 = (1, \dots, 1) \in \mathbb{F}_2^9$ とおく。各 e_i は、数独のマスを埋める $\{1, 2, \dots, 9\}$ の値 i に相当する。マス x_{ij} の初期値（ヒント）が k ($1 \leq k \leq 9$) の場合は、方程式 $x_{ij} - e_k = 0$ でこの条件が表される。

数独パズルの一行目に注目すると、パズルのルールは次の 37 本のブール方程式で表せる。

$$\sum_{i=1}^9 x_{1i} = 1, x_{1i} \cdot x_{1j} = 0 (i \neq j).$$

数独パズルには9行, 9列, 9個の 3×3 のブロックがあるため, 合計で $37 \times 27=999$ 本のブール方程式でパズルのルールは表せる. $I \subset \mathbb{F}_2^9(x)$ を初期値と999本のブール多項式によって生成されるイデアルとする. このとき, I を与えられた数独パズルの**初期イデアル**という.

次に, イデアル I の2種類の解集合を定義する.

定義 7 (2種類の解集合)

イデアル $I \subset (\mathbb{F}_2)^9(x)$ に対して

$$\mathbb{V}(I) := \{(a_{11}, \dots, a_{99}) \mid a_{ij} \in \mathbb{F}_2^9, \text{すべての } f \in I \text{ について}$$

$$f(a_{11}, \dots, a_{99}) = 0\} \subset (\mathbb{F}_2^9)^{81}$$

$$\mathbb{VS}(I) := \{(a_{11}, \dots, a_{99}) \mid a_{ij} \in \{e_1, \dots, e_9\}, \text{すべての } f \in I \text{ について}$$

$$f(a_{11}, \dots, a_{99}) = 0\}$$

と定義する. $\mathbb{V}(I)$ はイデアル I の定めるアフィン多様体であり, 一方, $\mathbb{VS}(I)$ は各変数が e_1, \dots, e_9 のみを値として取ることができる解集合である. したがって, I が数独パズルのイデアルの場合, $\mathbb{VS}(I)$ のほうを用いる.

以下では, 簡単のため変数の表記を $x = (x_{11}, \dots, x_{99})$ から $x = (x_1, \dots, x_{81})$ に変更し, ブール多項式環 $\mathbb{F}_2^9(x) = \mathbb{F}_2^9(x_1, \dots, x_{81})$ を用いることにする.

次に極小多項式を定義する.

定義 8 (極小多項式)

$I \subset \mathbb{F}_2^9(x)$ を $\mathbb{V}(I) \neq \emptyset$ を満たすようなイデアルとする. 1つの変数 x_i に対して, イデアル $I \cap \mathbb{F}_2^9(x_i)$ は $\mathbb{F}_2^9(x_i)$ の単項イデアルであり, ある $h(x_i) \in (\mathbb{F}_2^9(x_i))$ を用いて,

$$I \cap (\mathbb{F}_2^9(x_i)) = \langle h(x_i) \rangle.$$

と表せる. この1変数多項式 $h(x_i) = ax_i + b$ ($a, b \in \mathbb{F}_2^9$)は I と x_i によって一意的に決まり, 変数 x_i に対する I の**極小多項式**という.

極小多項式は, 変数 x_i の値に関する重要な情報を持っている (下記の命題 10 参照).

次に, $\mathbb{F}_2^9(x)$ 上の1変数多項式の分類をする.

定義 9 (1変数多項式の分類)

$f(x_j) = ax_j + b \in \mathbb{F}_2^9$ ($a, b \in \mathbb{F}_2^9$)を変数 x_j の0でない1変数ブール多項式とする.

(1) 解多項式 (solution-polynomial)

ある k に対して, $f = x_j + e_k$ の形をしている場合, 解多項式と呼ぶ.

(2) 準解多項式 (semisolution-polynomial)

$a \neq 1$, $a \cdot e_k \neq 0$ である $a \in \mathbb{F}_2^9$ に対して, $f = a \cdot x_j + e_k$ の形をしている場合, a型の準解多項式と呼ぶ.

$\#a = 8$ の $a \in \mathbb{F}_2^9$ に対して, $f = a \cdot x_j$ の形をしている場合, b型の準解多項式と呼ぶ. ここで, $\#a$ は a に含まれる1の数のことである. ある k に対して, $a = 1 + e_k$ と表されることに注意する.

a型, b型ともに, 解多項式 $g = x_j + e_k$ は, この準解多項式 f に対応する解多項式と呼ばれ, $\text{Asp}(f)$ で表す.

(3) 矛盾多項式 (contradiction-polynomial)

f が次のいずれかの形をしている場合, f を矛盾多項式と呼ぶ.

(a) $f = b$ ($b \in \mathbb{F}_2^9, b \neq 0$).

- (b) $f = x_j$.
(c) $f = ax_j + b$ ($a \neq 0, \#b \geq 2$).
(d) $f = ax_j + e_k$ ($a \neq 0, 1, a \cdot e_k = 0$).

(4) **分岐多項式** (branch-polynomial)

$f = a \cdot x_j$ ($1 \leq \#a \leq 7$) の形をしている場合, f を分岐多項式と呼ぶ. また, $\#a$ をこの分岐多項式の長さと呼ぶ.

次の命題は, 極小多項式がもつ変数の情報を示す.

命題 10

f を変数 x_j に関するイデアル I の極小多項式とする. このとき, 次が成り立つ:

- (1) もし f が解多項式 $f(x_j) = x_j + e_k$ ならば $x_j + e_k = 0$, つまり $x_j = e_k$ となる. したがって, 変数 x_j の値は k と決まる.
(2) f が $\text{Asp}(f) = x_j + e_k$ に対応する準解多項式であるとき, 変数 x_j の値は $\text{VS}(I)$ 内で e_k と決まる. したがって, 準解多項式は解多項式ではないが, 変数 x_j の値を一意に決定できる.
(3) $f = ax_j$ ($1 \leq \#a \leq 7$) を分岐多項式とする. このとき, $a = 1 + e_{i_1} + \dots + e_{i_p}$ ($1 \leq i_1 < \dots < i_p \leq 9, 2 \leq p = 9 - \#a \leq 8$) と書くと, 方程式 $f = 0$ は $\text{VS}(I)$ 内で p 個の解 e_{i_1}, \dots, e_{i_p} を持つ. したがって, 変数 x_j の取り得る値は $\{e_{i_1}, \dots, e_{i_p}\}$ と分かる.

次に, 基本閉包を定義する. 基本閉包は, 井上アルゴリズムと CII アルゴリズムを実行する際にとても重要である.

定義 11 (基本閉包)

(1) $I \subset \mathbb{F}_2^9(x)$ をイデアルとする. $\text{SemiSol}(I)$ を I の準解多項式の集合, $\text{Asp}(\text{SemiSol}(I))$ を $\text{SemiSol}(I)$ の準解多項式に対応する解多項式の集合とする. ここで,

$$\Psi_0(I) := I + \langle \text{Asp}(\text{SemiSol}(I)) \rangle$$

と定義する. $\Psi_0(I)$ は I に含まれる準解多項式に対応する解多項式を I にすべて付け加えて拡張されたイデアルになる.

(2) 次のようなイデアルの昇鎖列を考える.

$$I \subset \Psi_0(I) \subset (\Psi_0)^2(I) \subset \dots \subset \mathbb{F}_2^9(x).$$

$\mathbb{F}_2^9(x)$ は有限集合なので, $(\Psi_0)^m(I) = (\Psi_0)^{m+1}(I) = \dots$ を満たすような m が存在する. そのような m に対して $\text{BC}(I) := (\Psi_0)^m(I)$ と定義し, I の**基本閉包**と呼ぶ.

定義 12 (極大イデアル)

I を各変数の 81 個の解多項式によって生成されるイデアル $I := \langle x_1 + e_{i_1}, \dots, x_{81} + e_{i_{81}} \rangle$ とする. このとき, I を**極大イデアル**と呼ぶ. もし, I が極大イデアルならば, $\text{VS}(I) = \{e_{i_1}, \dots, e_{i_{81}}\}$ となる. したがって, 極大イデアルはすべてのマス (変数) の値が決まった, 解けた形のイデアルである.

定義 13 (基本可解と基本ランク)

$I \subset \mathbb{F}_2^9(x)$ を $\text{BC}(I)$ が極大イデアルであるようなイデアルとする. このとき, I を**基本可解なイデアル**と呼ぶ. もし I が数独パズルのイデアルならば, I は**基本可解なパズル**という. また, I を基本可解とするとき, $(\Psi_0)^m(I)$ が極大イデアルとなるような最小の m をイデアル I の**基本ランク**という. I が基本可解ではない場合, I の基本ランクは ∞ と定める.

初級から中級レベルのパズルは基本可解であることが知られている. また, 基本ランクは基本可解なパズルに対する難易度判定の優れた指標であることがわかっている ([8]).

3 井上アルゴリズムとCIIアルゴリズム

3.1 井上アルゴリズム

まず、人間が数独パズルを解くときの過程を考える。

(1) 初期値 (ヒント) から始めて、一意的に値が決まるマスを手できるだけ多く埋めていく。すると、一意的にはマスを埋められない状態に到達する。この状態では、空きマスはすべて複数の値を取り得る。この状態を**分岐点**と呼ぶが、数学的には初期イデアルの基本閉包に相当する。

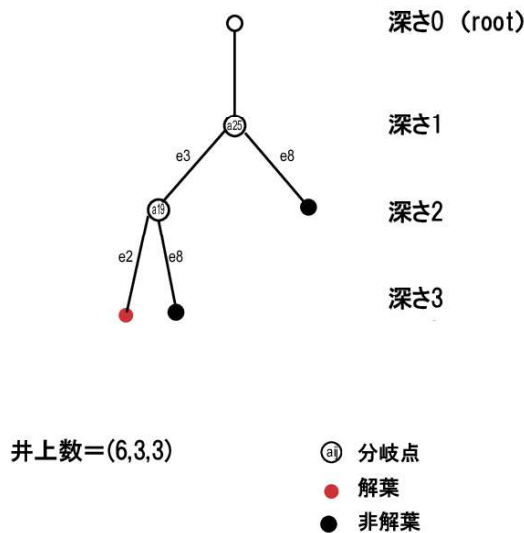
(2) 分岐点では、任意の空きマスを1つ選ぶ。例えば、その空きマスで取り得る値が3通りで2,4,5の場合、まず2を入れて新たに決まるマスをすべて埋めていく。このとき、入れた値が解でない場合、矛盾が起きる(1-9のどの値も入らないマスが出てきてしまうなど)ため、次の値4を試す。入れた値が解である場合、再び分岐点に到達し、そこで任意の空きマスを選ぶ。

(3) この操作をすべてのマスがただ1つの値で埋まるまで繰り返すと、必ず解に到達する。

このような解法を**トライアンドエラー方式**と呼び、この解法をブール多項式環のイデアルを用いて定式化したアルゴリズムを**井上アルゴリズム** ([4], [8], [15]) という。井上アルゴリズムは、解に到達したら終了するのではなく、矛盾する場合もすべて含めて計算してから終了することにする。

井上アルゴリズムの実行結果は、**井上樹形図**で簡明に表すことができ、終点(解葉または非解葉)を葉、葉以外の節を分岐点と呼ぶ。分岐点では、任意の空きマスを選んで良いため、井上アルゴリズムの実行経路には多数の異なる経路があり得る。したがって、1つのパズルは多数の井上樹形図を持ち得ることに注意する。数独は唯一の解を持つので各井上樹形図は唯一の解葉を持ち、その他の終点は全て非解葉(矛盾多項式を含むイデアルと対応)である。また、井上樹形図の大きさは(節数, 葉数, 深さの最大値)の3つ組で表すことができ、これを**井上数**と呼ぶ。

図 1: 井上樹形図 (L15) の例



上記の図1は、数独パズルL15の井上樹形図の例である。深さ0にあるルートが初期イデアルで、1回基本閉包をとると深さ1で約50個程度の空きマスが残る。ここでは、a25という任意の空きマスを選んでおり、取り得る値が3と8である。値3を入れるとまた分岐点に到達する。一方、値8を入れると矛盾し非解葉に到達する。ここが深さ2である。深さ2で到達した分岐点で空きマスa19を選び、取り得る値が2と8となっている。値2を入れると解葉に到達し、値8を入れると矛盾し非解葉となる。ここが深さ3となる。

井上アルゴリズムを数学的に記述すると次のようになる。

アルゴリズム 1 (井上アルゴリズム)

入力：与えられた数独パズルの初期イデアル I_0 . I_0 は井上樹形図のルートである。

出力：解のデータ L と井上樹形図のデータ. L の初期値は空配列とし、解が見つかるたびに解のデータを L に格納していく。

- (1) 初期イデアル I_0 の基本閉包をとり、このイデアルを $J := BC(I)$ とする. J は、井上樹形図における深さ 1 の節である。
- (2) ここで 3 つの場合がある。
 - (a) J が矛盾多項式を含む場合、 L には何も格納しない. このとき、井上樹形図において、 J は非解葉と呼ばれる終点となる。
 - (b) J が矛盾多項式を含まずに、極大イデアルとなる (J が 81 個の解多項式で生成される) 場合、 L には解のデータ (81 個のマス値) を格納する. このとき、井上樹形図において、 J は解葉と呼ばれる終点となる。
 - (c) J が非解葉でも解葉でもない場合、 J は分岐点と呼ばれる. このとき、任意の空きマス x_j を選び、その取り得る値が e_{k_1}, \dots, e_{k_p} の p 個あるとする. 各 l ($1 \leq l \leq p$) に対して、 $I := I + \langle x_j + e_{k_l} \rangle$ と再定義し、(1) に戻る。
- (3) この操作をすべての節が終点になるまで繰り返す. このアルゴリズムは有限回で終了し、その結果解のデータ L と井上樹形図が得られる。

この井上アルゴリズムを実行すると、分岐点で任意の空きマスを選んで良いため、多くの井上樹形図があり得る. それら全ての井上樹形図を考え、各井上樹形図の解葉の深さの最小値 (the Minimum Depth of Solution Leaves) を **MDSL** という。

郷内氏の 735 問、インカラ博士の 12 問、ターレック氏の 634 問合計 1381 問の MDSL を計算した結果、すべての問題で **MDSL=3** 以下となった. また、この実験の結果から、**MDSL=3** であるパズルは超難問に限られることがわかっている (後の関連の部分参照). さらに、**現在のところ MDSL が 4 以上のパズルは見つかっていない**. そこで著者達は次の予想を提唱している。

MDSL 予想 ([10], [11])

井上アルゴリズム (トライアンドエラー方式) で数独を解くとき、**唯一解をもつ数独パズルの解葉の最小の深さは必ず 3 以下**であろう。

これは、どんなに難しいパズルでも分岐点で 2 回適当な空きマスと値を選べば、必ず解に到達するというとても強い予想である。

3.2 CII アルゴリズム

まず、**VII** (Virtual Inoue Invariant, **理想的井上不変量**) を定義する。

定義 14 (理想的井上不変量)

$I \subset \mathbb{F}_2^0(x)$ を数独パズルの初期イデアルとする. 井上アルゴリズムから生じる全ての井上樹形図を考慮し、それらの井上数の最小値をイデアル I (または数独パズル) の**理想的井上不変量** (Virtual Inoue Invariant, **VII**) と呼ぶ. ただし、2 つの井上数は、左から比較する辞書式順序で大小を比較することにする。

VII は数独（イデア）の不変量ではあるが、実際は井上樹形図の数が多すぎて計算不可能である。そこで、CII（Computable Inoue Invariant, 計算可能井上不変量）というものを考える。CII は VII を近似する量で計算可能な量である。

定義 15 (CII アルゴリズム, 計算可能井上不変量 (CII))

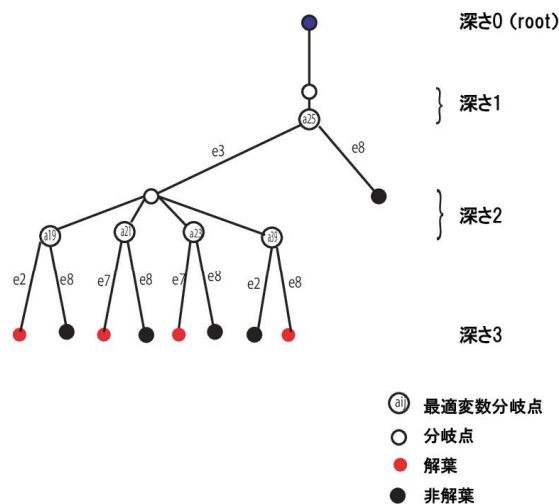
$I \subset \mathbb{F}_2^9(x)$ を数独パズルの初期イデアとする。CII アルゴリズムでは、分岐点で任意の空きマスを選ぶのではなく、次のようにできるだけ井上樹形図が小さくなりそうな変数に絞り、それらの変数のみを選択できるようにする。より詳しく、下記の定義 17 の最適変数絞り (A)-(D) または (0) を 1 つ選んで固定し、各分岐点で指定した絞り方によって変数を選ぶことにする。このとき、生成されるすべての井上樹形図における井上数の最小値を I の計算可能井上不変量 (Computable Inoue Invariant, CII) と呼ぶ。CII は、分岐点での最適変数の絞り方に依存するが、絞り方を 1 つ決めてしまえばイデアの数学的不変量となる。選んだ最適変数絞りを明確にするために、CIL.D, CIL.C などと表す。ここで、CIL.0 は VII と同じである。CII アルゴリズムの実行結果は CII 樹形図で簡明に表すことができる。

次の定義は、変数選択（最適変数絞り）のための基本戦術である。

定義 16 (変数選択のための 5 つの基本戦術)

- (A) この分岐点で分岐数最小の変数に絞る。
- (B) まず A で空きマスを絞ったあとに、それらの取り得る値を入れてできる次の節たちでの分岐数（の和）が最小のマスに絞る。ただし、葉の分岐数は 0 と約束する。
- (C) B まで絞ったあとに、次の節たちでの空きマスの個数（の和）が最小の空きマスに絞る。
- (D) C で絞ったあとに、次の節たちでの分岐数最小の空きマスの個数（の和）が最小の空きマスに絞る。
- (0) 便宜上、分岐点で全く変数を絞らずに、全ての空きマスを選んで良い場合も、CII アルゴリズムの 1 つの場合に含め、この場合は (0)（絞りが無いという意味）と表すことにする。この場合は、本質的には井上アルゴリズムと同じである。

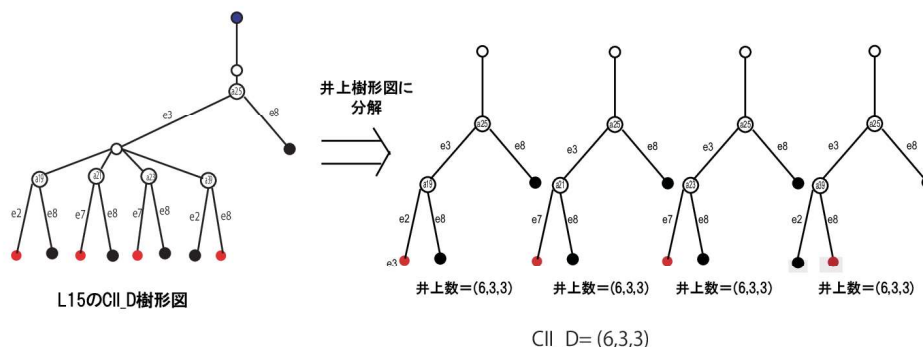
図 2: CIL.D 樹形図 (L15) の例



上記の図 2 は、数独パズル L15 の CIL.D 樹形図の例である。初期イデアで基本閉包をとり、深さ 1 の分岐点で、変数絞り (D) によって最適変数を 1 つ (a25) に、さらに、深さ 2 の分岐点でまた変数絞り (D)

によって最適変数を4つまで絞ることができている。下記の図3は、CII 樹形図を井上樹形図に分解し、それらの井上数の最小値を求めることにより CII を求めている図である。

図 3: CILD 樹形図 (L15) を井上樹形図に分解して CII_D を求める。



CII は数独パズルの良い難易度指標であることが分かっているが、一方で、超難問については、まれに、計算量が多く実行に時間がかかりすぎて計算できないことがある。そこで、CII 樹形図の大きさを直接測る不変量 SMYI を次のように定義する。

定義 17 (SMYI, Shindou-Mikoshiha-Yoshihara Invariant)

T を数独パズルの CII 樹形図とする。 $a := T$ のすべての節の数 (2 種類の分岐点については、最適変数分岐点を数え、通常分岐点を数えない), $b := T$ の葉の数, $c := T$ の深さの最大値とし, (a, b, c) の3つ組を **SMY 不変量** (略して **SMYI**) と定義する。

SMYI は分岐点での変数の絞り方に依存するが、一度変数の絞り方を決めてしまえば、数独 (イデアル) の不変量である。そこで、最適変数絞り (D) を採用し、これを SMYI として郷内氏の問題集 (全部で 735 問) を使用して、実験で SMYI とパズルの難易度の相関を調べた。ここで、SMYI の欄の値は、節の個数の平均値を表している。また、郷内レベルとは郷内氏の割り振った数独の難易度レベルのことである。

表 1: SMYI と郷内レベルの相関

郷内レベル	1	2	3	4	5	6	7	8	9	10
SMYI	2.00	2.00	2.00	2.00	2.00	2.00	2.00	14.11	14.28	23.03
郷内レベル	11	12	13	14	15	16	17	18		相関係数
SMYI	26.79	29.01	20.38	43.79	47.45	29.17	37.40	57.60		0.91618

この結果より、SMYI は、郷内レベルと約 0.92 という高い正の相関関係をもつことが判明した。これにより、SMYI は数独パズルの難易度の良い数学的指標として使用できることが検証できた。

4 戦術閉包と s_∞ -rank

4.1 戦術閉包の定義

初期イデアルにおいて、任意の空きマス x_j を選び 1-9 までの値 e_k を順に試し、一意的に決まるマスすべて埋めていき途中で矛盾が起きた場合、 x_j には e_k が入らないという情報が得られる。この情報を全部初期イデアルに付け加えてから、基本閉包をとる操作を考える。この操作を何度も行うと初期イデアルが拡張されていき、イデアルの拡張ができなくなったときのイデアルを**戦術閉包**と呼ぶ。

戦術閉包をとる操作は、人間の解法テクニックのいくつかを含む (XYwing 法, XY-chain 法など, [5]), 非常に強力な数独の数学的解法であり、戦術閉包をとっても解けないパズル (戦術閉包が極大イデアルにならないパズル) は (私たちの知る限り) これまで 3 例しか知られていなかった。戦術閉包を数学的に定義すると次のようになる ([5]).

定義 18 (戦術閉包)

• $I \subset \mathbb{F}_2^9(x)$ をイデアルとする。解多項式 $x_j + e_i$ が rank k の**反駁解多項式**とは、イデアル $\Phi_0^k(I + \langle x_j + e_i \rangle)$ が矛盾多項式を含むことをいう。ただし、 k はそのような最小値を意味する。

• 自然数 k に対して、 $BR_k(I) := \{e_i x_j \mid x_j + e_i \text{ が } I \text{ の rank } k \text{ 以下の反駁解多項式}\}$ とする。また、 $\Phi_k(I) := BC(I + \langle BR_k(I) \rangle)$ と定義すると、イデアルの昇鎖列

$$I \subset \Phi_k(I) \subset (\Phi_k)^2(I) \subset \dots$$

ができる。 $\mathbb{F}_2^9(x)$ は有限集合であるから、ある自然数 n が存在して $(\Phi_k)^n(I) = (\Phi_k)^{n+1}(I) = (\Phi_k)^{n+2}(I) = \dots$ となる。この n について、 $\Phi_k^*(I) := (\Phi_k)^n(I)$ と定義して、これを rank k の I の**戦術閉包**という。

次に k を動かすと次のイデアルの昇鎖列

$$\Phi_1^*(I) \subset \Phi_2^*(I) \subset \dots \subset \Phi_k^*(I) \subset \Phi_{k+1}^*(I) \subset \dots$$

ができるが、 $\mathbb{F}_2^9(x)$ は有限集合であるから、ある自然数 l が存在して $\Phi_l^*(I) = \Phi_{l+1}^*(I) = \Phi_{l+2}^*(I) = \dots$ となる。この l について、 $\Phi_\infty^*(I) := \Phi_l^*(I)$ と定義して、これを I の**戦術閉包**という。

次に s-rank を定義する。

定義 19 (s-rank)

$I \subset \mathbb{F}_2^9(x)$ を戦術閉包が極大イデアルになる (戦術可解イデアルという) ようなイデアルとして、 $\Phi_k^*(I) = \Phi_\infty^*(I)$ となる最小の k をとる。この k を I の **s-rank** という。ただし、基本可解イデアルの s-rank は 0 とする。また、 $\Phi_\infty^*(I)$ が極大イデアルでないとき、s-rank は ∞ とする。

戦術閉包 (と s-rank) を実際に計算するには、上記の定義は、昇鎖を 2 回取る必要があり、また、昇鎖が一旦止まってもその後も止まり続けるとは限らないため、定義通りの計算はやや面倒になる。そこで、本稿では次の戦術閉包の定義および s_∞ -rank を用いることにする。

定義 20 (戦術閉包の定義 2 と s_∞ -rank)

$I \subset \mathbb{F}_2^9(x)$ を基本可解でないイデアルとする。解多項式 $x_j + e_k$ が**反駁解多項式**とは、イデアル $BC(I + \langle x_j + e_k \rangle)$ が矛盾多項式を含むことをいう。

$$BR_\infty(I) := \{e_k x_j \mid x_j + e_k \text{ は } I \text{ の反駁解多項式}\}$$

とする。さらに、 $\Phi_\infty(I) := BC(I + \langle BR_\infty(I) \rangle)$ とすると、次の昇鎖列

$$I \subset \Phi_\infty(I) \subset (\Phi_\infty)^2(I) \subset \dots$$

ができる。この昇鎖はある自然数 n 以後は安定し、 $(\Phi_\infty)^n(I) = (\Phi_\infty)^{n+1}(I) = \dots$ となる。このとき、 $\Phi_\infty^*(I) = (\Phi_\infty)^n(I)$ (戦術閉包) となる。

また、これを満たす最小の n を I の s_∞ -rank という。ただし、基本可解イデアルの s_∞ -rank は 0 とする。また、戦術閉包 $\Phi_\infty^*(I)$ が極大イデアルでないとき、 s_∞ -rank は ∞ とする。

s_∞ -rank が ∞ のイデアルについては、上の最小の n を昇鎖の長さ (LAC, the Length of Ascending Chain) と呼ぶことにする。

ここで、s-rank が ∞ であることと、 s_∞ -rank が ∞ であることは同値であり、また、s-rank が有限値のパズルについては、 s_∞ -rank は s-rank より少し値が小さくなることが分かっている。

表 2: ターレック氏の s_∞ -rank ∞ のパズルの LAC

昇鎖の長さ (LAC)	0	1	2	3	4
B (ターレック氏)	5	9	0	0	0
HD (ターレック氏)	67	60	8	5	3
SM (ターレック氏)	0	0	1	0	0
問題数	72	69	9	5	3

上記の表 2 は、ターレック氏の 634 問に含まれる s_∞ -rank が ∞ のパズル (158 問) についての、LAC の計算結果である。なお、これまでに著者達の知る限りでは s_∞ -rank= ∞ であるパズルは 3 例しか知られていなかった。

5 主要結果

郷内氏の標準的な 735 問については、人間の難易度として「郷内ランク」がついていたので、これが人間の難易度の目安となった。さらに、この 735 問では CII を計算するのにそれほど時間がかからず、CII, SMYI とともに郷内ランクと高い相関があった。一方、ターレック氏の超難問 634 問には、人間の難易度が付与されていないため、難易度の目安となる指標がない。そこで、超難問の場合、ここでは **SMYI を難易度の目安**として採用することにする。

SMYI を難易度の目安として採用する根拠は

(a) CII アルゴリズムは、人間のトライアンドエラー方式の解法を模しているため、CII アルゴリズムの実行結果の大きさを直接はかる SMYI は**人間の難易度に近い**と考えられること。

(b) 郷内氏の 735 問の問題では、郷内ランクと SMYI は**高い相関**があったこと。

(c) CII, SMYI とともに郷内ランクと高い正の相関があり、どちらも難易度の目安として使えるが、CII は超難問については計算に時間がかかりすぎるため、全部計算するには無理がある。一方、SMYI は超難問でも**1 問数分程度で計算**できること。

の 3 点である。

よって、超難問については SMYI を難易度の指標と考え、実験で

(1) SMYI と MDSL

(2) SMYI と s_∞ -rank (s_∞ -rank 有限の場合)

(3) SMYI と昇鎖の長さ (LAC) (s_∞ -rank 無限の場合)

の比較を行い、傾向が判明したので報告する。次の表の SMYI の欄の値は、節の個数の平均値を表している。

(1) SMYI と MDSL は約 **0.98** という高い正の相関関係をもつことが判明した (下記表 3)。これは、**MDSL が大きいほど SMYI の値は大きくなり、パズルが難しくなる**ことを表している。

表 3: (1) SMYI と MDSL

MDSL	1	2	3	
SMYI	2.00	213.37	603.02	相関係数
問題数	11	270	365	0.98565

(2)(a) s_{∞} -rank が 1-5 までの場合は、SMYI と s_{∞} -rank は約 **0.95** という高い正の相関関係をもつことが判明した (下記表 4)。これは、 **s_{∞} -rank が大きいほど SMYI の値は大きくなり、パズルが難しくなる**ことを表している。

(b) s_{∞} -rank が 1-6 までの場合は、 s_{∞} -rank=6 である問題数が 2 問しかなくどちらも SMYI が小さいため、相関係数が低くなってしまふ。

表 4: (2) SMYI と s_{∞} rank (s_{∞} rank 有限の場合)

s_{∞} -rank	1	2	3	4	5	6		
SMYI	183.01	288.69	359.88	661.87	1029.75	254.00	相関係数	
問題数	168	247	49	15	4	2	0.9517641	0.4756188

(3) SMYI と LAC は約 **-0.96** という高い負の相関関係をもつことが判明した (下記表 5)。これは、**LAC が大きいほど SMYI の値は小さくなり、パズルが簡単になる**、また、**LAC が小さいほど SMYI の値は大きくなり、パズルが難しくなる**ことを表している。

著者達は、当初、 s_{∞} -rank が有限値の場合と同様に、LAC が大きいほどパズルは難しいのではないかと考えていたので、これはとても意外な結果であった。このようになる直観的な理由としては、例えば LAC=0 の場合は、戦術閉包をとっても全く空きマスが埋まらない (イデアルが拡張されない) ので、戦術閉包という強力な解法に対して抵抗力が最大であると考えられる。よって、LAC が小さいほど難しい、という結果になったと解釈できる。

表 5: (3) SMYI と昇鎖の長さ (LAC) (s_{∞} rank ∞ の場合)

昇鎖の長さ	0	1	2	3	4	
SMYI	917.26	931.31	771.22	610.67	425.75	相関係数
問題数	72	70	9	6	4	-0.96083

(3) の結果により、**LAC が s_{∞} -rank ∞ のパズルの難易度による階層付けの良い指標になる**ことが実験的に検証できた。特に、 **s_{∞} -rank ∞ のパズルで LAC=0 のパズルが数学的には最も難しい**と考えられる。このクラスのパズルは**初期イデアル=戦術閉包**という際立った特徴を持ち、**戦術閉なパズル**と呼ぶべきパズルであって、今後、数学的構造の研究が望まれる。

謝 辞

本研究は、科学研究費補助金（22K03275、代表：中野哲夫）の助成を受けたものである。また、本研究の発表に際しては、京都大学数理科学研究所の支援を受けた。

参 考 文 献

- [1] J. Cannon, W. Bosma, C. Fieker and A. Steel (eds.), HANDBOOK OF MAGMA FUNCTIONS Version 2.20, January 30, 2014.
- [2] 郷内邦義, 脳を鍛える数字パズルナンプレ (初級篇, 中級篇, 上級篇, 超上級篇, 難篇, 超難問篇, 限界篇), 廣濟堂出版, 2008.
- [3] A. Inkala, AI Sudoku Top 10 set, [http : //www.aisudoku.com/en/AIsudoku Top10s1 en.pdf](http://www.aisudoku.com/en/AIsudoku%20Top10s1%20en.pdf), 2006, 最終アクセス 2020 年 12 月 23 日.
- [4] S. Inoue, Efficient Singleton set constraint solving by Boolean Groebner bases, *Communications of JSSAC* **1** (2012), 27-37.
- [5] S. Inoue and Y. Sato, A Mathematical Hierarchy of Sudoku Puzzles and Its Computation by Boolean Groebner Bases, LNCS 8884, Springer, 2014, 88-98.
- [6] 丸山陽太郎, 井上不変量を用いた数独型パズルの難易度判定, 東京電機大学大学院理工学研究科修士論文, 2019.
- [7] T. Nakano, K. Arai and H. Watanabe, On the Inoue invariants of the puzzles of Sudoku type, *Communications of JSSAC* **2** (2016), 1-14.
- [8] T. Nakano, Y. Maruyama and S. Ohki, On the Mathematical Evaluation of Difficulty Level of Sudoku Puzzles by Boolean Groebner Bases, *Far East Journal of Applied Mathematics* **106** (2020), 43-70.
- [9] T. Nakano, S. Minami, S. Harikae, K. Arai, H. Watanabe and Y. Tonegawa, On the Inoue invariants of the puzzles of sudoku type II, *Bulletin of JSSAC* **24** (2018), 77-90.
- [10] T. Nakano, M. Shindou, N. Mikoshiba and T. Yoshihara, The SMY Invariant and the MDSL Conjecture in the CII Algorithm for Solving Sudoku Puzzles, *Far East Journal of Applied Mathematics* **114** (2022), 25-48.
- [11] 中野哲夫, 進藤未来, 御子柴尚希, 吉原元, 数独解を求める CII アルゴリズムの最近の発展と応用-不変量 SMYI と MDSL 予想-, 日本数式処理学会発表, 2022.6.18.
- [12] T. Nakano and Y. Tonegawa, Introduction to Boolean Groebner bases and their applications to puzzles of Sudoku type, *Journal of Algebra and Applied Mathematics* **12**, 2014, 1-31.
- [13] 落合啓之, 示野信一他, グレブナ基底と代数多様体入門 (上), 丸善出版, 2000.
- [14] 大木茂樹, Boolean Groebner 基底を用いた数独型パズルの 2 つの数学的難易度判定指標について, 東京電機大学大学院理工学研究科修士論文, 2020.
- [15] Y. Sato, S. Inoue, A. Suzuki, K. Nabeshima and K. Sakai, Boolean Groebner bases, *J. Symbolic Comput.* **46** (2011), 622-632
- [16] D. Tarek, <http://forum.enjoysudoku.com/the-hardest-sudokus-new-thread-t6539.html>