*JSSAC Bulletin of JSSAC(2018)*
*Vol. 24, No. 2, pp. 77 - 90*

論文

# On the Inoue invariants of the puzzles of Sudoku type II

Tetsuo NAKANO* — Tokyo Denki University
Sayaka MINAMI — Tokyo Denki University
Satoshi HARIKAE — Tokyo Denki University
Kenji ARAI — Tokyo Denki University
Hiromasa WATANABE — Tokyo Denki University
Yoshimune TONEGAWA — Tokyo Denki University

(Received 2015/3/5 Revised 2016/2/8 Accepted 2017/2/20)

**Abstract**

The Inoue algorithm is a superb method for solving a system of Boolean polynomial equations and it has been successfully applied to solving Sudoku puzzles. Recently several works have been done on the evaluation of the level of Sudoku puzzles by this method.

In our preceding paper, we defined the Inoue invariant of a system of Boolean polynomial equations, which measures the difficulty of solving such a system by the Inoue algorithm. But it does depend on the given linear order on the set of variables so that it is desirable to have an invariant of such a system which does not.

The purpose of this paper is firstly to define two kinds of new invariants of such a system, *the virtual and the computable Inoue invariant*, determined only by the system, and secondly to show by experiments that, when applied to the system arising from a Sudoku puzzle, these invariants are good indicator of the difficulty of the Sudoku puzzles.

## 1 Introduction

The Inoue algorithm [5] is an excellent method for solving a system of Boolean polynomial equations and it has been applied for solving Sudoku puzzles ([4, 7]). Recently several research works have been done on the evaluation of the difficulty of (or giving hierarchy to) Sudoku puzzles by this method ([6, 8, 14]). The purpose of this note is firstly to define two new invariants of a Boolean polynomial ideal by modifying the Inoue invariant defined in our preceding paper [10]. Secondly, we show by experiments that our new invariant is a fairly good indicator of the difficulty of the

*tnakano@mail.dendai.ac.jp

puzzles for humans. We also compare our new invariants with the known evaluation method for Sudoku puzzles.

More precisely, in our preceding paper [10], we defined the Inoue invariant of a system of Boolean polynomial equations, which measures the difficulty of solving such a system by the Inoue algorithm. We then classified the Inoue invariant of the puzzles of Sudoku type of smaller size, namely the 4-doku puzzles and the diagonal 5-doku puzzles. The Inoue invariant in [10] is a good indicator of the difficulty of solving a given system of Boolean polynomial equations, but it has a defect that it does depend on a given linear order on the set of variables.

In this paper, we will define two kinds of new invariants, *the virtual* and *the computable Inoue invariant*, determined only by a given system by modifying the definition of the Inoue invariant. The virtual Inoue invariant is defined as the minimal Inoue number of all the Inoue numbers obtained by applying the possible generalized Inoue algorithms to the system. The virtual Inoue invariant is theoretically satisfactory but hard to compute so that we also define the computable Inoue invariant, which approximates the virtual Inoue invariant well and is actually computable.

We have implemented the program for computing the computable Inoue invariant on the computer algebra system Magma [1] (we have put it in our website `http://www.r.dendai.ac.jp/~nakano/research.html`). We then show by experiments that these new invariants indicate the difficulty of Sudoku puzzles effectively when applied to the systems arising from the Sudoku puzzles.

The contents of this paper are as follows. In Section 2, we review briefly the Boolean Groebner bases, the Inoue algorithm and the Inoue invariant after [5, 8, 10]. In Section 3, we define the virtual and the computable Inoue invariant of a given Boolean polynomial ideal. In Section 4, we show by experiments that the computable Inoue invariant is a good indicator of the difficulty of Sudoku puzzles. In section 5, we compare our computable Inoue invariants with the known evaluation methods for Sudoku puzzles ([6, 8, 14]) , especially the b-rank and s-rank defined in [8]. Finally in Section 6, the results obtained are summarized.

We finally note that part of the results in this paper are surveyed in [11] and this work was supported by JSPS KAKENHI (23540057).

Acknowledgment: we thank the referees for teaching us the preceding works on this topic such as [4, 6, 7, 8, 14]. We are also grateful to Prof. Y. Sato for communicating some useful information to us on this work.

## 2 Preliminaries

In this section, we first summarize some basic facts on Boolean Groebner bases of ideals in a Boolean polynomial ring. For more details, see [12, 13, 15]. We then recall the Inoue algorithm [5] and the Inoue invariant [10] briefly.

### 2.1 Boolean Groebner bases

*A Boolean ring* $\mathbf{B}$ is a commutative ring with an identity such that any element $a \in \mathbf{B}$ satisfies $a^2 = a$. For a Boolean ring $\mathbf{B}$, if we define

$$a \vee b := a + b + ab, a \wedge b := ab, \neg a := 1 + a,$$

then $(\mathbf{B}, \vee, \wedge, \neg)$ is a Boolean algebra. Conversely, if $(\mathbf{B}, \vee, \wedge, \neg)$ is a Boolean algebra, then by defining

$$a + b := (a \wedge \neg b) \vee (\neg a \wedge b),\ \ a \cdot b := a \wedge b,$$

$(\mathbf{B}, +, \cdot)$ becomes a Boolean ring. Thus Boolean rings and Boolean algebras are the same concept.

For example, let $X_m := \{1, 2, \ldots, m\}$ be the set with $m$ elements and $\mathbf{B}_m$ the set of all subsets of $X_m$. Then $\mathbf{B}_m$ is a Boolean algebra with the "join, meet, complement" operations of subsets. Thus $\mathbf{B}_m$ is a Boolean ring with the addition (symmetric difference) and multiplication (meet) as defined above. Conversely, any finite Boolean ring is isomorphic to $\mathbf{B}_m$ for some natural number $m$ by the Stone representation theorem. There is a natural ring isomorphism $\mathbf{B}_m \cong (\mathbb{F}_2)^m$, where $\mathbb{F}_2 := \{0, 1\}$ is the field with 2 elements (the addition and multiplication in $(\mathbb{F}_2)^m$ are defined componentwise).

Let $\mathbf{B}$ be a Boolean ring and $\mathbf{B}[x] := \mathbf{B}[x_1, \ldots, x_n]$ the polynomial ring over $\mathbf{B}$ with $n$ variables. Given a monomial order on $\mathbf{B}[x]$, for a polynomial $f \in \mathbf{B}[x]$, we denote by $\mathrm{LT}(f)$, $\mathrm{LM}(f)$ and $\mathrm{LC}(f)$ the leading term, the leading monomial and the leading coefficient respectively. Thus $\mathrm{LT}(f) = \mathrm{LC}(f)\mathrm{LM}(f)$ holds.

For an ideal $I \subset \mathbf{B}[x]$, we denote by $\mathrm{LT}(I)$ the set of leading terms of the elements (except 0) in $I$. We now define a Groebner basis of an ideal in $\mathbf{B}[x]$.

**Definition 1 (Groebner Bases)**
*Given a monomial order on* $\mathbf{B}[x]$, *let* $I \subset \mathbf{B}[x]$ *be an ideal and* $G := \{g_1, \ldots, g_s\} \subset I$ *a finite subset of* $I$. *We say* $G$ *is a Groebner basis of* $I$ *if* $\langle \mathrm{LT}(I) \rangle = \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) \rangle$ *holds.*

We note that a division algorithm similar to (but slightly different from) the one in the case where the coefficient ring is a field holds in $\mathbf{B}[x]$. Based on this division algorithm, the Buchberger criterion and algorithm hold with slight modifications, and most of the results in Section 2 of [2] hold. We next define the reduced Groebner bases. We denote by $\overline{f}^G$ the remainder of division of $f$ by $G$.

**Definition 2 (Reduced Groebner Bases)**
*Let* $G$ *be a Groebner basis of an ideal* $I \subset \mathbf{B}[x]$. $G$ *is called reduced if for any* $g \in G$, $\overline{g}^{G \setminus \{g\}} = g$ *holds.*

Since reduced Groebner bases are not unique for a given ideal, we now define the stratified Groebner bases.

**Definition 3 (Stratified Groebner Bases)**
*Let* $G$ *be a reduced Groebner basis of an ideal* $I$. $G$ *is called a stratified Groebner basis if* $\mathrm{LM}(f) \neq \mathrm{LM}(g)$ *holds for any* $f, g \in G, f \neq g$.

The following theorem shows that a stratified Groebner basis is the canonical basis of a given ideal.

**Theorem 4**
*Fix a monomial order on* $\mathbf{B}[x]$. *For a given finitely generated ideal* $I \subset \mathbf{B}[x]$, *a stratified Groebner basis of* $I$ *exists and it is determined by* $I$ *uniquely.*

For the existence, we can construct the stratified Groebner basis of a given finitely generated ideal by the Buchberger algorithm followed by the reduction and stratification process. For the actual computation of the stratified Groebner bases in the case of $\mathbf{B} = (\mathbb{F}_2)^m$, we use *the componentwise method* (see [13, Theorem 22]).

We now turn to *the Boolean Groebner bases*. Since $\mathbf{B}[x]$ itself is not a Boolean ring, we set

$$\mathbf{B}(x) = \mathbf{B}(x_1, \ldots, x_n) := \mathbf{B}[x]/\langle x_1^2 - x_1, \ldots, x_n^2 - x_n \rangle$$

and call $\mathbf{B}(x)$ *a Boolean polynomial ring over* $\mathbf{B}$. We note that $\mathbf{B}(x)$ is a Boolean ring and any element $f(x) \in \mathbf{B}(x)$ can be written uniquely as $f(x) = \sum_\alpha c_\alpha x^\alpha$ where $c_\alpha \in \mathbf{B}$ and $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\alpha_i \in$

$\{0,1\}$. We call this expression *the canonical representation of $f \in \mathbf{B}(x)$*. Given a monomial order on $\mathbf{B}[x]$, we can define $\mathrm{LT}(f), \mathrm{LM}(f)$ and $\mathrm{LC}(f)$ of $f \in \mathbf{B}(x)$ by using the canonical representation of $f$.

**Definition 5 (Boolean Groebner Bases)**
*Fix a monomial order on $\mathbf{B}[x]$. For a given ideal $I \subset \mathbf{B}(x)$, a finite subset $G = \{g_1, \ldots, g_s\} \subset I$ is called a Boolean Groebner basis of $I$ if $\langle \mathrm{LT}(I) \rangle = \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) \rangle$ holds.*

The division algorithm works also in $\mathbf{B}(x)$ and we can define the reduced and stratified Boolean Groebner bases as in Definition 2 and 3. Then the existence and uniqueness of the stratified Boolean Groebner bases hold too.

We can compute the Boolean Groebner bases as follows. Let $I = \langle F \rangle \subset \mathbf{B}(x)$ be a given ideal with a set of generators $F$. Compute a Groebner basis $G$ of $\langle F \cup \{x_1^2 - x_1, \ldots, x_n^2 - x_n\} \rangle$ in $\mathbf{B}[x]$. Then $G' := G \setminus \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$ is a Boolean Groebner basis of $I$. If $G$ is stratified, then so is $G'$.

We finally refer to the Boolean Hilbert Nullstellensatz. For an ideal $I \subset \mathbf{B}(x)$, let $\mathbb{V}(I) := \{a \in \mathbf{B}^n \mid f(a) = 0 \text{ for any } f \in I\}$ be the affine variety defined by $I$.

**Theorem 6 (Boolean Hilbert Nullstellensatz)**
*Let $I \subset \mathbf{B}(x) = \mathbf{B}(x_1, \ldots, x_n)$ be a finitely generated ideal.*
*(i) $\mathbb{V}(I) = \phi \Longleftrightarrow I$ contains a non-zero constant (weak form).*
*(ii) Assume $\mathbb{V}(I) \neq \phi$. Then $f(x) \in I \Longleftrightarrow$ for any $a \in \mathbb{V}(I)$ $f(a) = 0$ holds (strong form).*

## 2.2 The Inoue algorithm and the Inoue invariant

We will first explain the Inoue algorithm for computing the singleton set solutions of a system of Boolean polynomial equations ([5]). We follow the method based on minimal polynomials and semi-solution polynomials ([7, 8]) rather than the original one based on almost solution polynomials. We will then review the Inoue invariant of a Boolean polynomial ideal after [10].

We work in the Boolean polynomial ring $(\mathbb{F}_2)^m(x) = (\mathbb{F}_2)^m(x_1, \ldots, x_n)$. For an ideal $I \subset (\mathbb{F}_2)^m(x)$, we set

$$\mathbb{VS}(I) := \{(a_1, \ldots, a_n) \mid a_i \in \{e_1, \ldots, e_m\},\ f(a_1, \ldots, a_n) = 0 \text{ for any } f \in I\},$$

where $e_1 := (1, 0, \ldots, 0), \ldots, e_m = (0, 0, \ldots, 0, 1) \in (\mathbb{F}_2)^m$. Thus $\mathbb{VS}(I)$ is the set of singleton set solutions and the Inoue algorithm computes it effectively.

The Inoue algorithm is based on the concept "*solution polynomials*" and "*semi-solution polynomials*" contained in the ideal. In the following, for an element $a = e_{i_1} + \ldots + e_{i_l}$ $(1 \leq i_1 < \ldots < i_l \leq m)$ in $(\mathbb{F}_2)^m$, we say the cardinality of $a$ is $l$ and denote it by $\sharp a = l$.

**Definition 7 (Solution Polynomial and Semi-solution Polynomial)**
*Let $f \in (\mathbb{F}_2)^m(x)$ be a univariate Boolean polynomial.*
*(i) If $f$ is of the form $f = x_j + e_k$ for some $j, k$, $f$ is called a solution polynomial.*
*(ii) If $f$ is of the form $f = ax_j + e_k$ $(a \in (\mathbb{F}_2)^m)$ with $a \cdot e_k \neq 0$, then $f$ is called a semi-solution polynomial of type-a. A solution polynomial $g := x_j + e_k$ is called the one associated with the semi-solution polynomial $f$ and denoted by $\mathrm{Asp}(f)$.*
*(iii) If $f$ is of the form $f = ax_j$ for some $a \in (\mathbb{F}_2)^m$ with $\sharp a = m - 1$, then $f$ is called a semi-solution polynomial of type-b. Let $e_k$ be the unique singleton element such that $a \cdot e_k = 0$. Then the solution polynomial $x_j + e_k$ is called the one associated with the semi-solution polynomial $f$ and denoted by $\mathrm{Asp}(f)$.*

*For a subset $S \subset (\mathbb{F}_2)^m(x)$, we denote by* SemiSol($S$) *the set of semi-solution polynomials contained in $S$.*

If a solution polynomial $f = x_j + e_k$ is contained in an ideal $I$, then the the value of the variable $x_j$ is determined to be $e_k$. We also note that if a semi-solution polynomial $f$ with the $\mathrm{Asp}(f) = x_j + e_k$ is contained in the ideal, then the value of $x_j$ must be $e_k$ since we are computing $\mathbb{VS}(I)$ (more precisely $\mathbb{VS}(I) = \mathbb{VS}(I + \langle x_j + e_k \rangle)$ holds). We next define a contradiction polynomial.

**Definition 8 (Contradiction Polynomial)**
*Let $f \in (\mathbb{F}_2)^m(x)$ be a univariate Boolean polynomial. If $f$ is of the form $f = a$ ($a$ is a non-zero constant), $f = x_j$* or *$f = ax_j + b$ ($a, b \in (\mathbb{F}_2)^m$ and $\sharp b \geq 2$), then $f$ is called a contradiction polynomial.*

*For a subset $S \subset (\mathbb{F}_2)^m(x)$, we denote by* Contra($S$) *the set of contradiction polynomials contained in $S$.*

If a contradiction polynomial $f$ is contained in the ideal $I$, then it is clear that $\mathbb{VS}(I)$ is empty. We now define *the basic closure* of an ideal, which is the main ingredient of the Inoue algorithm.

**Definition 9 (Basic Closure)**
*Let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal. We define the operation $\Psi_0$ by $\Psi_0(I) := I + \langle \mathrm{Asp}(\mathrm{SemiSol}(I)) \rangle$, where $\mathrm{Asp}(\mathrm{SemiSol}(I)) = \{\mathrm{Asp}(f) \mid f \in \mathrm{SemiSol}(I)\}$. Namely, $\Psi_0(I)$ is the ideal obtained from $I$ by adding all the solution polynomials associated with the semi-solution polynomials in $I$. Then we have an ascending ideal chain*

$$I \subset \Psi_0(I) \subset (\Psi_0)^2(I) \subset \ldots \subset \ldots.$$

*Since $(\mathbb{F}_2)^m(x)$ is finite, there exists $m$ such that $(\Psi_0)^m(I) = (\Psi_0)^{m+1}(I) = \ldots$. We set* $\mathrm{BasicClosure}(I) := (\Psi_0)^m(I)$ *for such $m$ and call it the basic closure of $I$.*

For the computation of the basic closure of an ideal, we need to know the semi-solution polynomials contained in the ideal. For that, we introduce the minimal polynomial of each variable.

**Definition 10 (Minimal Polynomial)**
*Let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal such that $\mathbb{V}(I) \neq \phi$ and $x_i$ an variable. The ideal $I \cap (\mathbb{F}_2)^m(x_i)$ is a principal ideal of $(\mathbb{F}_2)^m(x_i)$ and can be written as $I \cap (\mathbb{F}_2)^m(x) = \langle h(x_i) \rangle$ for some $h(x_i) \in (\mathbb{F}_2)^m(x_i)$. The univariate polynomial $h(x_i)$ is determined by $I$ and $x_i$ uniquely and called the minimal polynomial of $x_i$ with respect to $I$.*

Minimal polynomials are easily computed from the stratified Boolean Groebner basis of $I$ as follows:

**Proposition 11**
*Let $I$ be an ideal of $(\mathbb{F}_2)^m(x)$ with $\mathbb{V}(I) \neq \phi$. Fix a monomial order on $(\mathbb{F}_2)^m(x)$ and let $G$ be the stratified Boolean Groebner basis of $I$.*

*Then, for each variable $x_i$, there exists a non-zero minimal polynomial $h(x_i)$ of $x_i$ with respect to $I$ if and only if $G$ contains a polynomial $g = ax_i + b_1t_1 + \ldots + b_lt_l + c$ where* $\mathrm{LM}(g) = x_i$*, $t_1, \ldots, t_l$ are distinct terms ($\neq 1$) and $a, b_1, \ldots, b_l, c \in (\mathbb{F}_2)^m$ such that $a \neq b_1 \vee \ldots \vee b_l$.*

*Moreover, if such $g$ exists, the minimal polynomial $h(x_i)$ has the following form:*

$$h(x_i) = a(1 + b_1 \vee \ldots \vee b_l)x_i + c(1 + b_1 \vee \ldots \vee b_l).$$

The following proposition enables us to compute the basic closure of an ideal by means of the stratified Boolean Groebner basis and the minimal polynomials of the ideal.

**Proposition 12**
*Let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal and $G$ the stratified Boolean Groebner basis of $I$ with respect to a monomial order. Then $I$ contains a non-zero constant if and only if $G$ does.*

*Further suppose $\mathbb{V}(I) \neq \phi$ and let $M$ be the set of $n$ minimal polynomials of each variable $x_i$ $(1 \leq i \leq n)$ with respect to $I$. Then the following assertions hold:*
*(i)* $\mathrm{Contra}(I) \neq \phi$ *if and only if* $\mathrm{Contra}(M) \neq \phi$.
*(ii) If* $\mathrm{Contra}(I) = \phi$, *then* $I + \langle \mathrm{Asp}(\mathrm{SemiSol}(I)) \rangle = I + \langle \mathrm{Asp}(\mathrm{SemiSol}(M)) \rangle$ *holds.*

The proof of this proposition is easy and we omit it.

Now we will explain the idea of the Inoue algorithm. We call an ideal $J$ *a maximal ideal* if $J$ has a set of generators consisting of $n$ solution polynomials of each variable, namely, $J = \langle x_1 - e_{i_1}, \dots, x_n - e_{i_n} \rangle$.

Let $I$ be an ideal. Enlarge the ideal by setting $J := \mathrm{BasicClosure}(I)$. At this point, $J$ contains no semi-solution polynomials and thus we cannot go further. So we take the "try and error" method. Namely, choose one variable, say $x_i$, among the ones whose values are not determined yet, and add to $J$ the solution polynomial of the form $x_i + e_j$, where $e_j$ is a possible value of $x_i$ (see Algorithm 13 below for the precise definition of this). Then the new ideal $J' := J + \langle x_i + e_j \rangle$ may contain several semi-solution polynomials. Thus we can take the basic closure of $J'$ again and go further until we reach a maximal ideal.

We state the procedure of the Inoue algorithm as follows.

**Algorithm 13 (Inoue Algorithm)**
*Fix a linear order on the set of variables $\{x_1, \dots, x_n\}$ and let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal. Set $L := \{\}$ (an empty set). We will put a singleton set solution in $L$ in order.*
*(i) If* $\mathrm{Contra}(I) \neq \phi$, *then set $L := L \cup \{\}$ (no solution is added to $L$).*
*(ii) If* $\mathrm{Contra}(I) = \phi$, *then redefine $I :=$* BasicClosure*(I).*
*(iii) If* $\mathrm{Contra}(I) \neq \phi$, *then $L := L \cup \{\}$. If* $\mathrm{Contra}(I) = \phi$ *, we have 2 cases.*
*(a) If $I$ is a maximal ideal, then $L := L \cup \{I\}$ (we get a singleton set solution $I$).*
*(b) Else let $x_j$ be the least variable among the undetermined ones (the linear order on the set of variables is used only for this), and $\{e_{k_1}, \dots, e_{k_p}\}$ the possible values of $x_j$. Here we say $\{e_{k_1}, \dots, e_{k_p}\}$ are the possible values of $x_j$ if the minimal polynomial $h(x_j)$ of $x_j$ with respect to $I$ is of the form $h(x_j) = (1 + e_{k_1} + \dots + e_{k_p})x_j$. For each $l$ $(1 \leq l \leq p)$, redefine $I := I + \langle x_j + e_{k_l} \rangle$ and go to (ii).*
*(iv) The final output* $\mathrm{Inoue}(I) = L$.

For the implementation of this algorithm, we need Proposition 12. We next describe the performance of the Inoue algorithm by a tree diagram defined as below.

**Definition 14 (Tree Diagram)**
*Fix a linear order on the set of variables. Let $I$ be an ideal. Perform the Inoue algorithm starting from $I$. We will construct a tree diagram* Tree*(I) from $I$ as follows:*
*(i) The input $I$ is the first node at level 0 (root).*
*(ii) Take the basic closure of $I$ and set $J :=$* $\mathrm{BasicClosre}(I)$. *We have a second node $J$ at level 1.*
*(iii) There are three cases at this node. (a) If* $\mathrm{Contra}(J) \neq \phi$, *we have reached a terminal node called "a non-solution leaf". (b) If* $\mathrm{Contra}(J) = \phi$ *and $J$ is a maximal ideal, then we have reached a terminal node called "a solution leaf". (c) If* $\mathrm{Contra}(J) = \phi$ *and there remains a variable whose value is not determined yet, we call this node "a branch point". At a branch point, select the least variable $x_j$ among the undetermined ones. If there are $p$ possible values $\{e_{k_1}, \dots, e_{k_p}\}$ for $x_j$, then*

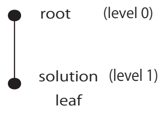
*this tree branches in $p$ directions at this node. Namely apply* BasicClosure *to the ideal $J + \langle x_j + e_{k_l} \rangle$ for each $l$ $(1 \leq l \leq p)$.*
*(iv) Repeat this process until all the branches reach a solution leaf or a non-solution one so that we get a tree diagram* Tree*(I).*

*We set $i_1 := \sharp\{\text{nodes}\}$, $i_2 := \sharp\{\text{leaves}\}$ and $i_3 :=$ the depth of Tree(I) and call the triple $\mathrm{II}(I) := (i_1, i_2, i_3)$ the Inoue invariant of the ideal $I$.*

For the comparison of two Inoue invariants, we use the lexicographic order with $i_1 > i_2 > i_3$.

**Example 15**
*Suppose $\sharp(\mathbb{VS}(I)) = 1$. If* BasicClosure*(I) is the maximal ideal, then* Tree*(I) is the simplest tree as shown below and* $\mathrm{II}(I) = (2, 1, 1)$*. In this case, we say $I$ has a trivial Inoue invariant.*

Fig. 1: The simplest tree with the trivial Inoue invariant (2,1,1)



## 3 The virtual and the computable Inoue invariants

We will define two modified versions of the Inoue invariant which evaluate the difficulty of solving a system of Boolean polynomial equations fairly well. These new Inoue invariants do not depend on the linear order on the set of variables and are determined only by the the given ideal.

In Algorithm 13 (iii)-b or Definition 14 (iii)-c, we will change the way of choosing a variable at a branch point and *allow selecting any undetermined variable* there. The algorithm works in this case too and we obtain a tree diagram. We call this modified algorithm *a generalized Inoue algorithm* and the triple numbers of this tree *an Inoue number of $I$.* An Inoue number of $I$ depends on the choice of the undetermined variable at each branch point and thus $I$ has many Inoue numbers.

**Definition 16 (Virtual Inoue Invariant)**
*Let $I$ be an ideal. We define the virtual Inoue invariant (VII for short) of the ideal $I$ as the minimal Inoue number of the set of trees for the possible performance of the generalized Inoue algorithm. We denote by* VII*(I) the VII of the ideal $I$.*

The VII shows how difficult it is to solve the given system in the best possible way by the generalized Inoue algorithm. It is determined by the ideal only and is theoretically satisfactory. Its defect is that, except it is relatively small, it is very hard to compute since we have to know all the Inoue numbers for the possible choice of a variable at any branch point.

So we next define *a computable Inoue invariant* according to the following strategy for the choice of variables at each branch point. Let $I$ be an ideal such that $\mathbb{V}(I) \neq \phi$ and $M$ the set of $n$ minimal polynomials of each variable with respect to $I$. If $I$ (and so $M$) contains no contradiction polynomials, then $M$ consists of 3 types of polynomials: a solution polynomial, a semi-solution polynomial and a polynomial of the form $h(x_j) = ax_j$ $(a \in (\mathbb{F}_2)^m)$ with $\sharp a \leq m - 2$, which we call *a strategy polynomial* in this note. Suppose $a = e_{i_1} + \ldots + e_{i_l}$ $(l = \sharp a \leq m - 2)$ and consider a strategy polynomial $h(x_j) = ax_j \in M$. Then we know $x_j \neq e_{i_k}$ $(1 \leq k \leq l)$. We call the number $l$ the length of the strategy polynomial $ax_j$.

**Definition 17 (The standard strategy for the selection of variables)**
*Let $I$ be an ideal which is a branch point in the tree diagram.*
*(i) We first choose the set of variables $S := \{x_{f_1}, \ldots, x_{f_s}\}$ where each $x_{f_j}$ has the minimal branch number $p$ among the undetermined variables.*
*(ii) For a variable $x_{f_j}$ in $S$ and $p$ possible values $e^j_k$ $(1 \leq k \leq p)$ of $x_{f_j}$, set $I'_{j,k} := I + \langle x_{f_j} + e^j_k \rangle$. Let $M'_{j,k}$ be the set of the $n$ minimal polynomials of each variable with respect to the ideal $I'_{j,k}$. We set $n(x_{f_j}, e^j_k) :=$ the total number of the solution polynomials and the semi-solution ones contained in $M'_{j,k}$, and define $n(x_{f_j}) := \sum_{k=1}^{p} n(x_{f_j}, e^j_k)$. Set $S' := \{x_{f_j} \in S \mid n(x_{f_j}) \text{ is maximal}\}$.*
*(iii) For a variable $x_{f_j}$ in $S'$ and $p$ possible values $e^j_k$ $(1 \leq k \leq p)$ of $x_{f_j}$, let $a(x_{f_j}, e^j_k)$ be the total sum of the length of the strategy polynomials contained in $M'_{j,k}$, and set $a(x_j) := \sum_{k=1}^{p} a(x_{f_j}, e^j_k)$. We finally set $S'' := \{x_{f_j} \in S' \mid a(x_j) \text{ is maximal}\}$ and we allow choosing the variable in $S''$ at this branch point.*

In Definition 17 (i), we choose the variables which give the minimal number of branches so that we expect to have smaller trees. In (ii), $n(x_{f_j}, e^j_k)$ is the number of the variables determined uniquely by setting $x_{f_j} = e^j_k$. So if $n(x_{f_j})$ is bigger, it means that the values of more variables are determined by choosing the variable $x_{f_j}$. In (iii), we choose the better variables from $S'$ by means of the strategy polynomials.

We call this strategy *the standard strategy for the selection of variables* at the branch points. It turns out that this strategy drastically lessens the number of variables chosen at each branch point (sometimes to 1).

**Definition 18 (Computable Inoue Invariant)**
*For an ideal $I$, we define the computable Inoue invariant (CII for short) of $I$ as the minimal Inoue number of the set of all trees where the choice of variables is done by the standard strategy at each branch point. We denote by $\mathrm{CII}(I)$ the computable Inoue invariant of $I$.*

CII($I$) is also determined only by the ideal $I$, and approximates VII($I$) fairly well. Actually, CII($I$) $\geq$ VII($I$) holds by definition and, when CII($I$) is relatively small, we can determine VII ($I$) (in most cases CII($I$) = VII($I$)).

**Proposition 19**
*(i)* $\mathrm{CII}(I) = (2, 1, 1) \Longleftrightarrow \mathrm{VII}(I) = (2, 1, 1)$.
*(ii) If* $\mathrm{CII}(I) = (4, 2, 2)$, *then* $\mathrm{VII}(I) = (4, 2, 2)$.

**Proof** (i) This is clear from the definition since there are no branch points.
(ii) If CII($I$) = (4,2,2), then the minimal branch number at the first branch point at level 1 is 2, and (4,2,2) is clearly the minimal Inoue number for any choice of the variable among the undetermined ones. We note the converse assertion is false. ∎

**Remark 20**
*We may take some different strategy at the branch points to define CII's. If we weaken (resp. strengthen) the conditions of the strategy, namely allow more (resp. less) variables at the branch points, then the resulting CII approximates the VII better (resp. worse), but the amount of computation increases (resp. decreases).*

## 4 Evaluation of the difficulty of Sudoku puzzles

We first formulate the rules of Sudoku puzzles in terms of a system of Boolean polynomial equations after [4, 13]. We assign 81 variables $a_{11}, \ldots, a_{99}$ to each cell as shown in Table 1 below.

Table 1: Assignment of 81 variables

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

Consider the Boolean polynomial ring $(\mathbb{F}_2)^9(a_{11}, a_{12} \ldots, a_{99}) = (\mathbb{F}_2)^9(a_{ij})$. Set $0 = (0, \ldots, 0), 1 = (1, \ldots, 1)$ and $e_1 := (1, 0, \ldots, 0), \ldots, e_9 := (0, \ldots, 0, 1) \in (\mathbb{F}_2)^9$. Let us take the first row. Then the 37 equations below express the rules of Sudoku for the first row:

$$\begin{cases} \sum_{j=1}^{9} a_{1j} = 1 \\ a_{1j} \cdot a_{1k} = 0 \ \ (1 \leq j < k \leq 9) \end{cases}$$

For example, $a_{1j} = e_j$ $(1 \leq j \leq 9)$ satisfies these equations.

There are 9 rows, 9 columns and 9 $3 \times 3$ blocks. So there are $37 \times 27 = 999$ equations (or generators of an ideal) in all. Adding the clues (initial values) to these generators, we can represent the rules of Sudoku puzzles by a system of Boolean polynomial equations. We call the ideal $I$ generated by these 999 polynomials together with the initial values *the ideal of the given Sudoku puzzle*. Then $\mathbb{VS}(I)$ is the set of solutions of this Sudoku puzzle.

We now turn to the evaluation of the difficulty of Sudoku puzzles. We have implemented the program for computing CII on Magma [1]. Then we have computed the CII's of the puzzles contained in a series of 6 Sudoku puzzle books [3], where each book contains 105 puzzles so that there are 630 puzzles in all. The result is summarized in Table 2 below.

In Table 2, the level shows the difficulty of the puzzle for humans assigned by the author of the books. The second and third columns show the average and maximum of the CII's of the puzzles of each level respectively. The column $\sharp$(puzzles) (resp. $\sharp$(trivial)) shows the number (= $a$) of puzzles of each level (resp. the number (= $b$) of puzzles with trivial CII of this level). The last column shows the proportion $\frac{b}{a} \times 100(\%)$.

From the average column of this table, we conclude that CII has a fairly strong correlation with the level of the puzzles, taking the ambiguity of the level assigned by humans into account. We should remark that the puzzles with trivial CII are included even in the most difficult puzzles such as level 16–18, though the proportion of them is small if the level is large. One reason for this is that humans seem to feel that some kind of the puzzles with trivial CII is difficult. More precisely, if the b-rank of the puzzle with trivial CII is large, humans tend to feel it difficult (see the next section for this).

We should also refer to the two puzzles $\infty_1$ and $\infty_2$ in Table 3 with extremely large CII. The puzzle $\infty_1$ (resp. $\infty_2$) is the famous "the world's hardest Sudoku puzzle" in 2010 (resp. in 2012) designed by Dr. A. Inkala. The puzzle $\infty_1$ (resp. $\infty_2$) has (40,20,7) (resp. (80,40,12)) as CII, and (80,40,12) is the largest CII as far as we know. These two puzzles are extremely rare also in the sense that both have s-rank $\infty$ (for the definition of s-rank, see the next section).

Table 2: The average of CII's of Sudoku puzzles of various level

| Level | average of CII | max of CII | ♯(puzzles) | ♯(trivial) | proportion (%) |
|---|---|---|---|---|---|
| 1 | (2,1,1) | (2,1,1) | 12 | 12 | 100 |
| 2 | (2,1,1) | (2,1,1) | 40 | 40 | 100 |
| 3 | (2,1,1) | (2,1,1) | 22 | 22 | 100 |
| 4 | (2,1,1) | (2,1,1) | 51 | 51 | 100 |
| 5 | (2,1,1) | (2,1,1) | 34 | 34 | 100 |
| 6 | (2,1,1) | (2,1,1) | 31 | 31 | 100 |
| 7 | (2,1,1) | (2,1,1) | 38 | 38 | 100 |
| 8 | (2.3,1.1,1.1) | (4,2,2) | 55 | 47 | 85 |
| 9 | (2.6,1.3,1.3) | (6,3,3) | 52 | 38 | 73 |
| 10 | (3.8,1.9,1.9) | (10,5,5) | 62 | 21 | 34 |
| 11 | (4.5,2.3,2.3) | (12,6,6) | 47 | 8 | 17 |
| 12 | (5.4,2.7,2.6) | (12,6,6) | 65 | 5 | 8 |
| 13 | (5.5,2.8,2.6) | (14,7,6) | 32 | 4 | 13 |
| 14 | (5.0,2.5,2.4) | (10,5,4) | 24 | 3 | 13 |
| 15 | (5.6,2.8,2.6) | (14,7,6) | 33 | 5 | 15 |
| 16 | (7.7,3.8,3.7) | (14,7,7) | 12 | 1 | 8 |
| 17 | (5.0,2.5,2.5) | (10,5,5) | 10 | 3 | 30 |
| 18 | (7.2,3.6,3.3) | (14,7,6) | 10 | 1 | 10 |

Table 3: The world's hardest Sudoku puzzles in 2010 and 2012

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

$\infty_1$: CII = (40,20,7)

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

$\infty_2$: CII = (80,40,12)

We finally note that the main results of [10] can be rephrased in terms of VII as follows.

**Theorem 21 ([10] Theorem 24 and Theorem 31)**
*(i) All the 4-doku puzzles* ($4 \times 4$) *with a unique solution have the trivial* VII $(2,1,1)$.
*(ii) All the diagonal 5-doku puzzles* ($5 \times 5$ *with no block conditions but with 2 diagonal conditions) with a unique solution have the trivial* VII $(2,1,1)$ *except the 2 puzzles W1 and W2 in Table 4 below (modulo rotation and reflection). These 2 puzzles have (4,2,2) as the* VII.

Table 4: The diagonal 5-doku puzzles with the non-trivial VII $(4, 2, 2)$

W1

| | | | | |
|---|---|---|---|---|
| 1 | | | 4 | 5 |
| | | | | |
| | | | | |
| 3 | 1 | | 5 | |
| | | | | |

W2

| | | | | |
|---|---|---|---|---|
| 1 | | | | |
| | 4 | | | |
| | | | | |
| 3 | 1 | | | |
| 4 | 5 | | | |

## 5 Comparison with the known methods for evaluating the Sudoku puzzles

In this section, we review the 3 known methods for evaluating the difficulty of Sudoku puzzles ([6, 8, 14]). Especially, we will compare the b-rank and s-rank defined in [8] with our CII.

In [6], the authors tried to evaluate the difficulty of the puzzles basically by the quantity of computations needed for the performance of the generalized Inoue algorithm. They chose 4 kinds of data to evaluate the puzzles, one of which is the number of times of computations of the Boolean Groebner bases. The experimental data shows that this method gives a good indicator of the difficulty of the puzzles.

One of the defects of this method is that the 4 kinds of data used for the evaluation are not an invariant of the given ideal. Indeed, in the generalized Inoue algorithm used in [6], how to choose a variable at the branch points is not shown ( it is only claimed that "choose a suitable variable and continue"). Anyway, the 4 kinds of data used for the evaluation depend on how to choose a variable at the branch points, and hence are not an invariant of the given ideal.

We also note that this method depends on the kinds of data chosen for the evaluation. Namely, the more kinds of data are chosen for evaluation, the more precisely we can evaluate the level of the puzzles but the amount of computation gets larger. Thus how to choose a suitable set of data is to be studied in this direction.

In [14], the authors evaluated the puzzle with the ideal $I$ by the average LP($I$) of the length of all the paths in the tree diagrams arising from the generalized Inoue algorithm. This LP($I$) is an invariant of the ideal $I$, and turns out to be a good indicator of the difficulty of the puzzles.

For the computation of LP($I$), they had to compute the length of all the paths in the tree diagrams arising from the generalized Inoue algorithm, which is a huge computation. So they needed to develop parallel computing for this.

Thus LP($I$) is close to CII($I$) in idea, whereas our CII (approximately) measures the minimum size of the tree diagrams arising from the generalized Inoue algorithm. We note that the amount of computations for CII is so small that we can compute CII with a PC mostly in several minutes without parallel computing.

Finally in [8], a hierarchy of Boolean polynomial ideals was defined by two invariants, *the basic rank* and *the strategy rank*. We now review their definitions and compare them with our CII in detail. The definition of basic rank is easy:

**Definition 22 (Basic Rank)**
*Let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal with $\sharp\mathbb{V}\mathbb{S}(I) = 1$. If* BasicClosure*($I$) is a maximal ideal, then we call $I$ a basic solvable ideal.*

*Suppose $I$ is basic solvable. Recall that the operation $\Psi_0$ is defined by $\Psi_0(I) := I + \langle \mathrm{Asp}(\mathrm{SemiSol}(I))\rangle$. Then $J := \mathrm{BasicCosure}(I)$ is obtained from $I$ by applying $\Psi_0$ several times. We define the basic rank (or b-rank for short) of $I$ as the least $m$ such that $(\Psi_0)^m(I) = J$.*

Thus by definition, $I$ has trivial CII (2,1,1) if and only if $I$ is basic solvable. We have computed the b-rank of the 364 basic solvable ideals contained in the 6 books [3]. Table 5 below shows the average of b-rank of each level. From this we know that the b-rank of an ideal $I$ gives a good hierarchy to Sudoku puzzles with the trivial CII, in the sense that it has a strong correlation with the level of the puzzle for humans.

Table 5: The average of b-ranks of the puzzles with trivial CII

| Level | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ave. | 1.6 | 2.5 | 2.6 | 2.8 | 4.0 | 4.6 | 4.7 | 5.1 | 6.7 | 6.8 | 7.9 | 7.2 | 7.3 | 8.7 | 8.4 | 7.0 | 7.7 | 10.0 |

For the definition of the strategy rank, we need to define *a refutable polynomial.*

**Definition 23 (Refutable Polynomial)**
*Let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal and $f(x_j) = x_j + e_k$ a solution polynomial. If $\mathrm{BasicClosure}(I + \langle x_j + e_k\rangle)$ contains a contradiction polynomial, then $f(x_j)$ is called a basic refutable polynomial with respect to $I$.*

*Suppose $f(x_j) = x_j + e_k$ is basic refutable. Then the minimal number $m$ such that $(\Psi_0)^m(I)$ contains a contradiction polynomial is called the basic refutable rank of $f$ with respect to $I$.*

**Definition 24 (Strategy Closure)**
*Let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal. For a natural number $k$, we set $BR_k(I) := \{e_k x_j \mid x_j + e_k$ is basic refutable with respect to $I$ of rank $\leq k\}$. We define $\Phi_k(I) := \mathrm{BasicClosure}(I + \langle BR_k(I)\rangle)$. Then we have an ascending ideal chain:*

$$I \subset \Phi_k(I) \subset (\Phi_k)^2(I) \subset \ldots \subset \ldots.$$

*Since $(\mathbb{F}_2)^m(x)$ is a finite set, there exists an $m$ such that $\Phi_k^m(I) = \Phi_k^{m+1}(I) = \ldots$. For such an $m$, we set $\Phi_k^*(I) := \Phi_k^m(I)$ and call this the strategy closure of $I$ of rank $k$.*

*Then we move $k$. Consider the ascending ideal chain:*

$$\Phi_1^*(I) \subset \Phi_2^*(I) \subset \ldots \subset \Phi_k^*(I) \subset \ldots \subset \ldots.$$

*Then there exists an $m$ such that $\Phi_m^*(I) = \Phi_{m+1}^*(I) = \ldots$. For such an $m$, we set $\Phi_\infty^*(I) := \Phi_\infty^m(I)$ and call it the strategy closure of $I$.*

**Definition 25 (Strategy Rank)**
*Let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal with $\sharp\mathbb{VS}(I) = 1$. Suppose $I$ is not basic solvable. If the strategy closure $\Phi_\infty^*(I)$ of $I$ is a maximal ideal, then we call $I$ a strategy solvable ideal.*

*For a strategy solvable ideal $I$, let $m$ be the least integer such that $\Phi_m^*(I) = \Phi_\infty^*(I)$. We call such $m$ the strategy rank (or s-rank for short) of $I$. We say a basic solvable ideal has s-rank 0. If the strategy closure $\Phi_\infty^*(I)$ is not a maximal ideal, we say $I$ has s-rank $\infty$.*

In [8, 9], the s-ranks of all Sudoku puzzles in [3] were computed (the computational data in [8] was incorrect, and is corrected in [9]). It turned out that all the values of the s-rank of these puzzles are finite, and actually are in $\{0, 1, 2\}$. So we know that, from the viewpoint of solving the Sudoku puzzles, the method of taking strategy closure is so powerful that all the puzzles in [3] can be solved by it. The puzzles with s-rank $\infty$ are rare, and the 2 puzzles $\infty_1$ and $\infty_2$ in Table 3 are the only ones with s-rank $\infty$ as far as we know.

From the viewpoint of giving hierarchy to Sudoku puzzles, the s-rank gives only 3 categories as far as the puzzles in [3] are concerned. Thus s-rank is too coarse for evaluating Sudoku puzzles by itself. In [9], giving a finer hierarchy to the Sudoku puzzles was tried by means of two numbers $A, B$. We here propose a very simple and natural invariant in order to give a finer hierarchy to Sudoku puzzles with finite positive s-rank.

**Definition 26** (**$(s, t)$-rank**)
*Let $I$ be a strategy solvable ideal of s-rank $s$ and $\Phi^*_\infty(I)$ its strategy closure which is maximal. Let $t$ be the least number such that $(\Phi_s)^t(I) = \Phi^*_\infty(I)$. We call the pair $(s, t)$ the $(s, t)$-rank of $I$.*

We have computed the $(s, t)$-rank of the 266 puzzles which are not basic solvable in [3]. Then we have found that the $(s, t)$-rank ranges from $(1, 1)$ to $(1, 7)$ and from $(2, 1)$ to $(2, 3)$. In order to show the result clearly, we give a point to each $(s, t)$-rank as follows:

Table 6: The 10 point hierarchy of strategy solvable puzzles by $(s, t)$-rank

| $(s, t)$ | (1,1) | (1,2) | (1,3) | (1,4) | (1,5) | (1,6) | (1,7) | (2,1) | (2,2) | (2,3) |
|---|---|---|---|---|---|---|---|---|---|---|
| point | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

The average of 10-point-rated $(s, t)$-rank of each level is shown in Table 7 below. For the comparison, we also show the average of CII's with non-trivial CII. We note that all the puzzles of level 1 – 7 are basic solvable (namely, s-rank 0) so that they are excluded from this table.

From Table 7, we conclude that both $(s, t)$-rank and CII give a good hierarchy to Sudoku puzzles of finite positive s-rank in the sense that they have a correlation with the level of the puzzles for humans.

Table 7: The average of $(s, t)$-rank and CII of the puzzles with non-trivial CII

| Level | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| (s,t)-rank | 1.13 | 1.36 | 1.63 | 1.87 | 1.67 | 2.21 | 2.86 | 2.43 | 2.09 | 2.86 | 3.78 |
| CII | (4.0,2.0,2.0) | (4.1,2.1,2.1) | (4.7,2.4,2.4) | (5.0,2.5,2.5) | (5.7,2.9,2.8) | (6.0,3.0,2.9) | (5.4,2.7,2.6) | (6.2,3.1,2.9) | (8.2,4.1,3.9) | (6.3,3.1,3.1) | (7.8,3.9,3.6) |

**Remark 27**
*For an ideal $I \subset (\mathbb{F}_2)^m(x)$, set $BR_\infty(I) := \{e_k x_j \mid x_j + e_k$ is basic refutable with respect to $I\}$. We set $\Phi_\infty(I) :=$ BasicClosure$(I + \langle BR_\infty(I)\rangle)$. Consider the ascending ideal chain*

$$I \subset \Phi_\infty(I) \subset (\Phi_\infty)^2(I) \subset \ldots \subset \ldots.$$

*Then there exists an natural number $m$ such that $(\Phi_\infty)^m(I) = (\Phi_\infty)^{m+1}(I) = \ldots$. It is easy to see that the strategy closure $\Phi^*_\infty(I)$ is equal to $(\Phi_\infty)^m(I)$ for such an $m$. We note that this gives a convenient way of computing the strategy closure. For a strategy solvable ideal $I$, we call the least $m$ such that $\Phi^*_\infty(I) = (\Phi_\infty)^m(I)$ the $s_\infty$- rank of $I$.*

*We have computed the $s_\infty$-rank of all the puzzles in [3], and found that all the puzzles have $s_\infty$-rank 1 except one puzzle. This special puzzle is Q101 in UltraHard of [3], which has $s_\infty$-rank 2. Thus $s_\infty$-rank turns out not to give enough hierarchy to Sudoku puzzles.*

## 6 Conclusion

We have defined two kinds of new invariants of a Boolean polynomial ideal, namely CII and VII, which measure the difficulty for solving it by the generalized Inoue algorithm in the best possible way. We have then confirmed by experiments that, in the case of those ideals arising from Sudoku puzzles, CII is a fairly good indicator of the difficulty of puzzles for humans.

We have also compared them with the b-rank and s-rank defined in [8]. The puzzles from the elementary level to the middle have trivial CII (2,1,1). To evaluate these puzzles with trivial CII, the b-rank turns out to be an excellent indicator of the level of the puzzles for humans.

We emphasize that the s-rank and the finer ranks derived from it cannot evaluate the puzzles with s-rank $\infty$. On the other hand, CII can evaluate the puzzles with s-rank $\infty$ fairly well with a small amount of computation. For example, the puzzles $\infty_1$ and $\infty_2$ in Table 3 have both s-rank $\infty$, but we can say that $\infty_2$ is much harder that $\infty_1$ by comparing their CII's.

## References

[1] Bosma, W., Cannon, J.J., Fieker, C. and Steel, A. (eds.): *Handbook of Magma functions*, http://magma.maths.usyd.edu.au/magma/, accessed 10 April 2015.

[2] Cox, D., Little, J. and O'Shea, D.: *Ideals, Varieties, and Algorithms* (third ed.), Springer, New York, 2007.

[3] Gohnai, K.: *Number Placement Puzzles (Basic, Middle, High, Hard, SuperHard, UltraHard)* (in Japanese), Kosaido Publishing Co., Tokyo, 2008.

[4] Inoue, S., Sato, Y., Suzuki, A. and Nabeshima, K.: Solving Sudoku puzzles by means of Boolean Groebner bases (in Japanese), *RIMS Kôkyûroku* **1666**, 2009, 1–5.

[5] Inoue, S.: Efficient Singleton Set Constraint Solving by Boolean Gröbner Bases, *Communications of JSSAC*, **1**, 2012, 27–37.

[6] Inoue, S. and Sato, Y.: The evaluation of the difficulty of Sudoku puzzles and the puzzle making by means of Groebner bases (in Japanese), *RIMS Kôkyûroku* **1785**, 2012, 51–56.

[7] Inoue, S.: The improvement of set constraint solving by means of Boolean Groebner bases (in Japanese), *RIMS Kôkyûroku* **1907**, 2014, 110–115.

[8] Inoue, S. and Sato, Y.: A Mathematical Hierarchy of Sudoku Puzzles and Its Computation by Boolean Groebner Bases, *LNCS* **8884**, Springer, 2014, 88–98.

[9] Nagai, A. and Sato, Y.: An efficient implementation of Boolean Grobner Bases of a power set algebra, *Proceedings of the 20th Asian Technology Conference in Mathematics*, Leshan, China, 2015, 326–335.

[10] Nakano, T., Arai, K. and Watanabe, H.: On the Inoue invariants of the puzzles of Sudoku type, to appear in *Communications of JSSAC*.

[11] Nakano, T. and Tonegawa, Y.: Introduction to Boolean Groebner bases and their applications to puzzles of Sudoku type, *J. of Algebra and Applied Math.*, **12**, 2014, 1–31.

[12] Sato, Y.: A New Type of Canonical Gröbner Bases in Polynomial Rings over Von Neumann Regular Rings, *Proceedings of ISSAC*, ACM press, 1998, 317–321.

[13] Sato, Y., Inoue, S., Suzuki, A., Nabeshima, K. and Sakai, K.: Boolean Gröbner bases, *J. of Symbolic Computation*, **46**, 2011, 622–632.

[14] Sato, Y, Inoue S. and Sato, Y.: On the parallel computing of Boolean Groebner bases for the determination of the difficulty of Sudoku puzzles (in Japanese), *RIMS Kôkyûroku* **1843**, 2013, 28–37.

[15] Weispfenning, V.: Gröbner Bases in Polynomial Ideals over Commutative Regular Rings, *EUROCAL'87, Springer LNCS* **378** (Davenport, E. ed.), Springer, 1989, 336–347.