Communications of JSSAC (2016)
Vol. 2, pp. 1 – 14

On the Inoue invariants of the puzzles of Sudoku type

Tetsuo Nakano*

Graduate School of Science and Engineering, Tokyo Denki University

Kenji Arai

Graduate School of Science and Engineering, Tokyo Denki University

Hiromasa Watanabe

Graduate School of Science and Engineering, Tokyo Denki University

(Received 29/Mar/2013 Accepted 13/Sept/2014)

Abstract

A Sudoku puzzle is a worldwide popular game, and is also an interesting object in combinatorics and computer algebra. Recently, Inoue applied his excellent algorithm on finding the singleton set solutions of a system of Boolean polynomial equations to the solution of the puzzles of Sudoku type. Further, by means of his algorithm, we have defined the Inoue invariant of puzzles of Sudoku type, which measures the mathematical difficulty of them.

The purpose of this note is study the Inoue invariants of the easier puzzles of Sudoku type, namely, 4-doku, diagonal 5-doku and diagonal 6-doku puzzles. Our main results show that all the 4-doku and diagonal 5-doku puzzles (with a unique solution) have the trivial Inoue invariant $(2, 1, 1)$ except 2 puzzles, whereas there exist many diagonal 6-doku puzzles with a non-trivial, big Inoue invariant.

1 Introduction

A Sudoku puzzle is a very popular game played by everybody in the world. Recently, numerous researches have been done on the mathematical (combinatorial) structure of Sudoku (see, for instance, the book [7] and references in it). Among them, Sato, Inoue and others [9, 10] studied it by means of Boolean Groebner bases.

Quite recently, Inoue [3] obtained an excellent method for finding the singleton set solutions of a system of Boolean polynomial equations. He also applied his algorithm to Sudoku and observed that relatively easy Sudoku puzzles can be solved without branches (namely without "else" procedure in Algorithm 34 of [3]).

*tnakano@mail.dendai.ac.jp

This work was supported by JSPS KAKENHI (23540057).

Stimulated by his observation, we went one step further and defined *the Inoue invariant* of puzzles of Sudoku type as follows ([5]). The performance of Inoue's algorithm for a Boolean polynomial ideal is well described by a tree diagram and we have defined the Inoue invariant of such an ideal as the triple of the basic numbers of this tree. We discovered that, in the case of those ideals arising from the puzzles of Sudoku type, this invariant is an excellent indicator of the difficulty of the puzzles by experiments. Thus we have defined the mathematical difficulty of the puzzles of Sudoku type as the Inoue invariant of their ideals. For example, in the case of Sudoku, the easier puzzles up to the middle level have the trivial Inoue invariant $(2, 1, 1)$, whereas the difficult ones have a non-trivial Inoue invariant. As far as we know, the biggest Inoue invariant so far is $(964, 558, 13)$, which is achieved by a 20-clues puzzle.

In this note, we study the Inoue invariants of the simpler puzzles of Sudoku type, namely 4-doku and the diagonal 5-doku. We computed many examples of them and got a conjecture that all the 4-doku and diagonal 5-doku puzzles with a unique solution have the the trivial Inoue invariant $(2, 1, 1)$. The purpose of this note is to give an answer to this conjecture. Our main results are summarized as follows.

Theorem 1 (Inoue invariants of 4-doku)
All the 4-doku puzzles with a unique solution have the trivial Inoue invariant $(2, 1, 1)$.

Theorem 2 (Inoue invariants of diagonal 5-doku)
(i) There exist exactly 30964554720 diagonal 5-doku puzzles with a unique solution.
(ii) They all have the trivial Inoue invariant $(2, 1, 1)$ *except the 2 puzzles* W_i $(i = 1, 2)$*, both of which have the Inoue invariant* $(4, 2, 2)$ *(see Table 5 in Section 5 for* W_i*,* $i = 1, 2$*).*

Thus our conjecture is false in the case of the diagonal 5-doku puzzles, but we discovered 2 special puzzles W_i $(i = 1, 2)$ with a non-trivial Inoue invariant. We also report a partial result on the Inoue invariants of the diagonal 6-doku puzzles, which shows that the there exist many diagonal 6-doku puzzles with a non-trivial, big Inoue invariant.

The contents of this note are as follows. In Section 2, we review Boolean Groebner bases, especially the stratified Boolean Groebner bases. In section 3, we summarize Inoue's algorithm and the Inoue invariants after [3, 5]. In section 4, we formulate the rules of puzzles of Sudoku type by a system of Boolean polynomial equations following [9, 10], and we report our main results in Section 5. In Appendix [6], which is separated from the main body of this note and put in our website, we summarize the detailed data and the programs used in the proof of our main results.

For the implementation of Inoue's algorithm, we have used the computer algebra system Magma [1].

Acknowledgment: we thank the referee for pointing out and correcting a critical mistake in the first version of this note.

2 Boolean Groebner Bases

In this section, we will briefly review the Groebner bases of ideals in the polynomial ring over a Boolean ring and the Boolean Groebner bases of ideals in a Boolean polynomial ring. For more details on Boolean Groebner bases, see [8, 9, 10, 11].

Let $\mathbf{B}$ be a Boolean ring. Namely, $\mathbf{B}$ is a commutative ring with an identity such that any element $a \in \mathbf{B}$ satisfies $a^2 = a$. For example, for a natural number m, $(\mathbb{F}_2)^m$ is a finite Boolean ring, where $\mathbb{F}_2 := \mathbb{Z}/2\mathbb{Z}$ is the field with 2 elements, and the addition and multiplication in $(\mathbb{F}_2)^m$ are defined componentwise. Conversely, any finite Boolean ring is isomorphic to $(\mathbb{F}_2)^m$ for some m by the Stone representation theorem.

Let $\mathbf{B}[x] = \mathbf{B}[x_1, \ldots, x_n]$ be the polynomial ring over $\mathbf{B}$ with n indeterminates with a given monomial order. For the notation on polynomials, we follow [2] as below.

Notation 3
(i) $\mathrm{LM}(f)$ *(resp.* $\mathrm{LT}(f)$*,* $\mathrm{LC}(f)$*,* $\mathrm{mdeg}(f)$*) is the leading monomial (resp. the leading term, the leading coefficient, the multidegree) of a polynomial* f *so that* $\mathrm{LT}(f) = \mathrm{LC}(f) \cdot \mathrm{LM}(f)$ *and* $\mathrm{LM}(f) = x^{\mathrm{mdeg}(f)}$ *hold.*
(ii) For monomials x^α *and* x^β*,* $x^\alpha \mid x^\beta$ *means that* x^α *divides* x^β*.*

We first show the division algorithm in $\mathbf{B}[x]$.

Theorem 4 (Division algorithm)
Given a polynomial f *and an ordered set of* s *polynomials* $F := (f_1, \ldots, f_s)$ *in* $\mathbf{B}[x]$*, we get an expression of the form* $f = a_1 f_1 + \cdots + a_s f_s + r$*, where* $(a_1, \ldots, a_s)$ *is the quotient and* r *the remainder, by the following algorithm:*
Algorithm variables: p *(intermediate dividend),* $B = (b_1, \ldots, b_s)$ *(intermediate quotient),* r *(intermediate remainder).*
Initial values: $p := f$*,* $B := (0, \ldots, 0)$*,* $r := 0$*.*
(i) If there exists i *such that* $\mathrm{LM}(f_i) \mid \mathrm{LM}(p)$ *and* $\mathrm{LC}(p) \cdot \mathrm{LC}(f_i) \neq 0$*, then take the least such* i *and redefine* $p := p - \mathrm{LC}(p) \cdot \frac{\mathrm{LM}(p)}{\mathrm{LM}(f_i)} \cdot f_i$ *and* $b_i := b_i + \mathrm{LC}(p) \cdot \frac{\mathrm{LM}(p)}{\mathrm{LM}(f_i)}$ *(***division step***).*
(ii) If there exists no i *such that* $\mathrm{LM}(f_i) \mid \mathrm{LM}(p)$ *and* $\mathrm{LC}(p) \cdot \mathrm{LC}(f_i) \neq 0$*, then redefine* $p := p - \mathrm{LT}(p)$ *and* $r := r + \mathrm{LT}(p)$ *(***remainder step***).*
This algorithm terminates (namely $p = 0$*) in a finite number of steps and yields an expression of division*

$$f = a_1 f_1 + \cdots + a_s f_s + r,$$

where r *satisfies the condition of the remainder:* $r = 0$ *or in case* $r \neq 0$*, any term* t *of* r *satisfies either* $\mathrm{LM}(f_i) \nmid \mathrm{LM}(t)$ *or* $\mathrm{LC}(t) \cdot \mathrm{LC}(f_i) = 0$ *in case* $\mathrm{LM}(f_i) \mid \mathrm{LM}(t)$ *for any* i*. Furthermore, if* $a_i f_i \neq 0$ *then* $\mathrm{mdeg}(a_i f_i) \leq \mathrm{mdeg}(f)$ *holds.*

This division algorithm in $\mathbf{B}[x]$ is quite similar to that in the polynomial ring over a field, except that one additional condition (the product of coefficients is not equal to 0) is necessary for the division step to occur.

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

Definition 5 (Groebner bases)
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* $\langle \mathrm{LT}(I) \rangle = \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) \rangle$*.*

Based on the division algorithm, most of the results in [2, Chapter 2] hold with suitable modifications. Especially, the Buchberger criterion and algorithm hold (with slight modifications) so that we can obtain a Groebner basis of a finitely generated ideal by the Buchberger algorithm.

We next define reduced and stratified Groebner bases respectively. We denote by $\overline{f}^F$ the remainder of the division of f by F.

Definition 6 (Reduced Groebner bases)
Let G *be a Groebner basis of an ideal* I*.* G *is called reduced if* $\overline{g}^{G \setminus \{g\}} = g$ *holds for any* $g \in G$*.*

Reduced Groebner bases are not unique as shown in the following example.

Example 7
In the polynomial ring $(\mathbb{F}_2)^2[x]$ *of one variable,* $\{(1,0)x, (0,1)x\}$ *and* $\{(1,1)x\}$ *are both reduced Groebner bases of the same ideal* $I = \langle x \rangle$.

Definition 8 (Stratified Groebner bases)
Let $G \subset I$ *be a reduced Groebner basis.* G *is called a stratified Groebner basis if* $\mathrm{LM}(f) \neq \mathrm{LM}(g)$ *for any* $f, g \in G$, $f \neq g$.

Proposition 9 (Stratification algorithm)
Let $G \subset I$ *be a reduced Groebner basis. Divide* G *into several groups* $G_1, \ldots, G_t$ *according to leading monomials, where each member of a group has the same leading monomial and different groups have different leading monomials:* $G = G_1 \cup \cdots \cup G_t$ *(disjoint union). For each group* G_i, *set* $h_i := \sum_{g \in G_i} g$. *Then* $G' := \{h_1, \ldots, h_t\}$ *is a stratified Groebner basis of* I.

The following is the main theorem of the Groebner bases.

Theorem 10 (Existence and uniqueness of the stratified Groebner bases)
Fix a monomial order on $\mathbf{B}[x]$. *For a given finitely generated ideal* $I \subset \mathbf{B}[x]$, *a stratified Groebner basis exists and it is determined by* I *uniquely.*

For the actual computation of the stratified Groebner bases in the case of $\mathbf{B} = (\mathbb{F}_2)^m$, we use the "componentwise" method explained below. We first prepare some notations.

Consider the natural isomorphism $(\mathbb{F}_2)^m[x] \cong (\mathbb{F}_2[x])^m$ and let $\pi_i : (\mathbb{F}_2[x])^m \to \mathbb{F}_2[x]$ be the projection to the i-th component. For any $f \in (\mathbb{F}_2)^m[x]$, we set $f_i := \pi_i(f) \in \mathbb{F}_2[x]$ and call it the i-th component of f. Then the isomorphism $(\mathbb{F}_2)^m[x] \cong (\mathbb{F}_2[x])^m$ is given as $(\mathbb{F}_2)^m[x] \ni f \longleftrightarrow (f_1, \ldots, f_m) \in (\mathbb{F}_2[x])^m$. For an ideal $I \subset (\mathbb{F}_2)^m[x]$, we set $I_i := \{f_i \mid f \in I\} \subset \mathbb{F}_2[x]$ and call this the i-th component ideal of I.

The algorithm is based on the following theorem:

Theorem 11
Fix a monomial order on $(\mathbb{F}_2)^m[x]$ *and let* $I \subset (\mathbb{F}_2)^m[x]$ *be an ideal. For any* i $(1 \leq i \leq m)$, *let* $I_i \subset \mathbb{F}_2[x]$ *be the* i*-th component ideal of* I *and* G_i *the reduced Groebner basis of* I_i. *Then* $G := (G_1, 0 \ldots, 0) \cup (0, G_2, 0, \ldots, 0) \cup \cdots \cup (0, \ldots, 0, G_m)$ *is a reduced Groebner basis of* I, *where* $(G_1, 0, \ldots, 0) = \{(g, 0, \ldots, 0) \mid g \in G_1\}$ *etc.*.

Thus we can compute the stratified Groebner basis of I by Theorem 11 followed by the stratification process (Proposition 9).

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

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

$\mathbf{B}(x)$ is a Boolean ring and we call it *the Boolean polynomial ring* over $\mathbf{B}$ with n indeterminates. A monomial $x_1^{\alpha_1} \ldots x_n^{\alpha_n}$ is called *a Boolean monomial* if $\alpha_i \in \{0, 1\}$ for any i. We note any $f \in \mathbf{B}(x)$ can be written uniquely as $\sum_k c_k x^{\beta_k}$ where $c_k \in \mathbf{B}$ and x^{β_k} is a distinct Boolean monomial, which we call *the canonical representation of* f. Given a monomial order on $\mathbf{B}[x]$ and $f \in \mathbf{B}(x)$, we can define $\mathrm{LT}(f), \mathrm{LM}(f)$ and $\mathrm{LC}(f)$ using the canonical representation of f.

Definition 12 (Boolean Groebner bases)
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 Boolean Groebner basis (a BG basis for short) of* I *if* $\langle \mathrm{LT}(I) \rangle = \langle \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) \rangle$.

The division algorithm (Theorem 4) works also in $\mathbf{B}(x)$ and we can define reduced and stratified BG bases as in Definition 6 and 8. Then the existence and uniqueness of the stratified BG bases hold too. We abbreviate the stratified BG bases as *the SBG bases* in the following sections.

We can compute a BG basis of the ideal $I = \langle F \rangle \subset \mathbf{B}(x)$ as follows. 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 BG basis of I. Furthermore, if G is stratified, then G' is also stratified. We note that the componentwise method (Theorem 11) also works for the BG bases.

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 13 (Boolean Hilbert Nullstellensatz)
Let $I \subset \mathbf{B}(x)$ be a finitely generated ideal. Then the following assertions hold.
(i) $\mathbb{V}(I) = \phi$ if and only if I contains a non-zero constant.
(ii) Assume $\mathbb{V}(I) \neq \phi$. Then $f(x) \in I$ if and only if $f(a) = 0$ for any $a \in \mathbb{V}(I)$.

3 The Inoue algorithm and the Inoue invariants

In this section, we will briefly review the Inoue algorithm and the Inoue invariants ([3, 5]). The Inoue algorithm is an excellent and almost canonical method for computing the singleton set solutions of a system of Boolean polynomial equations.

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 $e_i := (\delta_{ij})_{j=1,\ldots,m}$. $\mathbb{VS}(I)$ is the set of singleton set solutions and we would like to compute this set.

The Inoue algorithm is based on the concept *almost solution polynomials* contained in the ideal. In the following, we set $E := \sum_{i=1}^{m} e_i = 1_{(\mathbb{F}_2)^m}$.

Definition 14 (Solution polynomial)
We call $f \in (\mathbb{F}_2)^m(x)$ of the form $f := E \cdot x_j + e_k = x_j + e_k$ for some j, k a solution polynomial.

We note that for a solution polynomial $f := x_j + e_k$, $f = 0$ is equivalent to $x_j = e_k$. We next define an almost solution polynomial.

Definition 15 (Almost solution polynomial)
(i) A polynomial $f(x) \in (\mathbb{F}_2)^m(x)$ is called an almost solution polynomial of type 1 (ASP of type 1 for short) if there exist j, k such that $e_k \cdot f(x) = e_k \cdot x_j + e_k$ (namely, $f_k(x) = x_j + 1$ where f_k is the k-th component of f). We call $\mathrm{Sol}(f) := x_j + e_k$ the solution polynomial associated to the ASP f. We require a solution polynomial to be excluded from the ASP's of type 1.
(ii) A polynomial $g(x) \in (\mathbb{F}_2)^m(x)$ is called an ASP of type 2 if there exist j, k such that $e_t \cdot g = e_t \cdot x_j$ for any t except k (namely, $g_t(x) = x_j$ for any t except k). We call $\mathrm{Sol}(g) := x_j + e_k$ the solution polynomial associated to g. We require a solution polynomial to be excluded from the ASP's of type 2.

Suppose f is an ASP of type 1 with its solution polynomial $\mathrm{Sol}(f) = x_j + e_k$. Then $f = 0$ implies that the k-th component of the variable x_j is 1. Thus x_j must be equal to e_k since we are computing $\mathbb{VS}(I)$. But note that $f = 0$ is not equivalent to $x_j = e_k$. A similar reasoning holds for ASP of type 2.

We prepare some notations for the Inoue algorithm.

Notation 16
Let $I \subset (\mathbb{F}_2)^m(x)$ be an ideal.
(i) $\mathrm{CONST}(I)$:= *the set of non-zero constants contained in* I.
(ii) $\mathrm{SP}(I)$:= *the set of solution polynomials contained in* I. *For a variable* x_j, *if a solution polynomial* $f = x_j + e_k$ *is contained in* $\mathrm{SP}(I)$, *then we say the variable* x_j *is determined (with the value* e_k*).*
(iii) $\mathrm{ASP}(I)$:= *the set of ASP's (of type 1 or 2) in* I.
(iv) $\mathrm{Sol}(\mathrm{ASP}(I)) := \{\mathrm{Sol}(f) \mid f \in \mathrm{AS}(I)\}$ = *the set of solution polynomials associated to ASP's contained in* I.

The following algorithm ASPTransform is the main part of the Inoue algorithm.

Algorithm 17 (ASPTransform)
Let I *be an ideal (Input).*
(i) If $\mathrm{CONST}(I) \neq \phi$, *then the output ASPTransform(I) is* I.
(ii) If $\mathrm{CONST}(I) = \phi$, *then redefine* $I := I + \langle \mathrm{Sol}(\mathrm{ASP}(I)) \rangle$. *Namely, add to* I *all the solution polynomials associated to the ASP's in* I. *Then go to (i) again.*
(iii) Repeat this process until $\mathrm{CONST}(I) \neq \phi$ *or* $\mathrm{ASP}(I) = \phi$, *and the output ASPTransform(I) is* I.

Now we can state the Inoue algorithm.

Algorithm 18 (The Inoue algorithm)
Fix a linear order on the set of variables $\{x_1, \ldots, x_n\}$ *(not a monomial order). Let* $I \subset (\mathbb{F}_2)^m(x)$ *be an ideal (input). Set* $L := \{\}$ *(empty set). We will put a singleton set solution in* L *in order.*
(i) If $\mathrm{CONST}(I) \neq \phi$, *then set* $L := L \cup \{\}$.
(ii) If $\mathrm{CONST}(I) = \phi$, *then redefine* $I :=$ *ASPTransform(I).*
(iii) If $\mathrm{CONST}(I) \neq \phi$, *then* $L := L \cup \{\}$. *If* $\mathrm{CONST}(I) = \phi$ *, we have 2 cases. (a) If* $\mathrm{SP}(I)$ *consists of* n *solution polynomials (namely* $I = \langle x_j + e_{j_k} \mid j = 1, \ldots, n \rangle$ *) so that all the variables are determined, then* $L := L \cup \{\mathrm{SP}(I)\}$ *(this is a solution). (b) Else let* x_j *be the least variable among the undetermined ones and* $\{e_{k_1}, \ldots, e_{k_p}\}$ *the possible values of* x_j. *Here* e_{k_l} *is a possible value of* x_j *if and only if* $\mathrm{CONST}(I + \langle x_j + e_{k_l} \rangle) = \phi$. *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 $\mathrm{Inoue}(I) = L$.

The following theorem makes it possible to rephrase the Inoue algorithm in terms of SBG bases instead of ideals.

Theorem 19
Let I *be an ideal in* $(\mathbb{F}_2)^m(x)$ *and* G *its SBG basis for a given monomial order. In the assertions (ii), (iii) below, we assume* I *does not contain non-zero constants.*
(i) For a non-zero constant $c \in (\mathbb{F}_2)^m$, $c \in I$ *if and only if* $c \in G$.
(ii) For a solution polynomial f, $f \in I$ *if and only if* $f \in G$.
(iii) If an ASP g *is in* I, *then there exists an ASP* $g' \in G$ *such that* $\mathrm{Sol}(g) = \mathrm{Sol}(g')$.

The assertion (iii) of Theorem 19 above is the main result (Theorem 31) of [3]. By Theorem 19, we can rephrase the Inoue Algorithm in terms of SBG bases instead of ideals. Namely, just replace the ideal I by its SBG basis G in Algorithms 17 and 18.

For the actual implementation of this algorithm, we also need the explicit classification of ASP's contained in an SBG basis (see [5, Corollary 3.9]). Inoue has implemented his algorithm on the computer algebra system Risa/Asir, whereas we have implemented it on the computer algebra system Magma [1].

We next define the Inoue invariant of an ideal $I \subset (\mathbb{F}_2)^m(x)$. The performance of the Inoue algorithm is well described by a tree diagram defined as below.

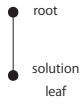
Definition 20 (Inoue Invariant)
Let I be an ideal and perform the Inoue algorithm starting from I. We will construct a tree Tree(I) of I as follows:
(i) I is the first node (root).
(ii) When the algorithm ASPTransform stops, we have a second node.
(iii) There are three cases. (a) If at this node $\mathrm{CONST}(I) \neq \phi$*, we have reached a terminal node (non-solution leaf). (b) If* $\mathrm{CONST}(I) = \phi$ *and all the n variables are determined, then we have reached a terminal node (a solution leaf). (c) If* $\mathrm{CONST}(I) = \phi$ *and there are undetermined variables, then select the least undetermined variable x_j. If there are p possible values $\{e_{k_1}, \ldots, e_{k_p}\}$ for x_j, then this tree branches in p directions at this node.*
(iv) Repeat this process until all the branches reach a (solution or non-solution) leaf.

By the process above, we get a tree 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* $\mathrm{Ino}(I) := (I_1, I_2, I_3)$ *the Inoue invariant of the ideal I.*

For the comparison of two Inoue invariants, we use lex order temporarily. The Inoue invariant measures the complexity of computation of the singleton set solutions $\mathbb{VS}(I)$ by the Inoue algorithm and is a very subtle invariant of I.

Example 21
Suppose $\sharp(\mathbb{VS}(I)) = 1$*. In case the Inoue algorithm calls ASPTransform only once and we reach the unique solution at once, then Tree(I) is the simplest tree with 2 nodes, 1 leaf and depth 1 (see Figure 1 below). In this case, we say I has a trivial Inoue invariant* $(2, 1, 1)$*.*

Fig. 1: The simplest tree with $\mathrm{Ino}(I) = (2, 1, 1)$



4 Formulation of puzzles of Sudoku type by a system of Boolean polynomial equations

In this section, we formulate the rules of the puzzles of Sudoku type in terms of Boolean polynomial equations after [9, 10].

A Sudoku puzzle is a partially-filled 9×9 square board with the integers $1, 2, \ldots, 9$, which should be completed in such a way that every row, column and the designated 3×3 block (see Table 1 below) is filled with no repeated entries.

We study the simpler versions of Sudoku, namely 4-doku, diagonal 5-doku and diagonal 6-doku puzzles. *A 4-doku puzzle* is a partially-filled 4×4 square board with integers $1, 2, 3, 4$. Every row, column and 2×2 block of the board should be filled with no repeated entries.

A diagonal 5-doku puzzle is a partially filled 5×5 table, where each row, column and diagonal (there are two diagonals) should be filed with numbers $1, \ldots, 5$ (no repeated entries). Since there are no blocks in 5-doku, it is natural to impose the diagonal conditions.

A diagonal 6-doku puzzle is a partially filled 6×6 table, where each row, column, 2×3 block and diagonal should be filled with numbers $1, \ldots, 6$ (no repeated entries). Note that there are six 2×3 rectangular (not square) blocks (see Table 2 below).

Table 1: An example of Sudoku puzzles

A Sudoku puzzle (clues)

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

$\Longrightarrow$

The solution board

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

Table 2: An example of a diagonal 6-doku puzzle

A puzzle (clues)

					1
5					2
		6		3	

$\Longrightarrow$

The solution board

1	2	3	4	5	6
4	6	5	2	1	3
3	1	2	5	6	4
6	5	4	3	2	1
5	3	1	6	4	2
2	4	6	1	3	5

Since the formulation of the rules of these puzzles by a system of Boolean polynomial equations are similar, we take 4-doku puzzles for simplicity and formulate their rules.

Table 3: Assignment of 16 variables

a_{11}	a_{12}	a_{13}	a_{14}
a_{21}	a_{22}	a_{23}	a_{24}
a_{31}	a_{32}	a_{33}	a_{34}
a_{41}	a_{42}	a_{43}	a_{44}

We first assign 16 variables $a_{11}, a_{12}, \ldots, a_{44}$ as in Table 3. We then consider the Boolean polynomial ring $(\mathbb{F}_2)^4(a_{11}, a_{12} \ldots, a_{44})$ with lex order $a_{11} < a_{12} < \cdots < a_{44}$. We abbreviate as $0 = (0,0,0,0), 1 = (1,1,1,1)$ and set $e_1 := (1,0,0,0), e_2 := (0,1,0,0)$ etc.. Let us take the first row. Then the 7 equations below express the rules of 4-doku for the first row:

$$a_{11} + a_{12} + a_{13} + a_{14} + 1 = 0 \tag{1}$$

$$a_{11} \cdot a_{12} = 0, a_{11} \cdot a_{13} = 0, a_{11} \cdot a_{14} = 0, a_{12} \cdot a_{13} = 0, a_{12} \cdot a_{14} = 0, a_{13} \cdot a_{14} = 0 \tag{2}$$

For example, $(a_{11}, a_{12}, a_{13}, a_{14}) = (e_1, e_2, e_3, e_4)$ satisfies these equations. We note that there are lots of solutions in $(\mathbb{F}_2)^4$ other than this. For example, $(a_{11}, a_{12}, a_{13}, a_{14}) = (0, e_1 + e_2, e_3, e_4)$ is also a solution, which of course is not admissible as a solution for 4-doku puzzles.

There are 4 rows, 4 columns and 4 blocks so that there are $7 \times 12 = 84$ equations (or generators of an ideal) in all. Adding the clues (initial values) to the above generators, we can represent the rules of 4-doku puzzles by Boolean polynomials. We call the ideal generated by the above 84 polynomials together with the clues *the ideal of the given 4-doku puzzle*.

Let S be a puzzle of Sudoku type and I its ideal. In [5], we have defined the mathematical difficulty of S as $\mathrm{Ino}(I)$, which is supported by experimental data.

5 Main results

In this section, we state our main results and give the outlines of their proofs. For more details, see Appendix [6]. We first need a definition.

Definition 22 (Redundant and irredundant puzzles)
Let S be a puzzle of Sudoku type with a unique solution. If the deletion of any one number from S yields a puzzle which has more than one solution, we say S is an irredundant puzzle. A puzzle is called redundant if S is not irredundant.

The following proposition lessens the amount of computation very much.

Proposition 23
Let S be a redundant puzzle of Sudoku type with a unique solution, and S' a puzzle with several numbers deleted from S. We assume S' still has a unique solution. If the Inoue invariant of S' is trivial, then that of S is trivial too.

Proof Let I (resp. I') be the ideal of the puzzle S (resp. S'). Suppose we are applying the Inoue algorithm to I'. Since the Inoue invariant of I' is trivial, we reach the unique solution by applying ASPTransform once. Since we have $I \supset I'$, it holds that $\mathrm{ASP}(I) \supset \mathrm{ASP}(I'), \mathrm{SP}(I) \supset \mathrm{SP}(I'), \mathrm{Sol}(\mathrm{ASP}(I)) \supset \mathrm{Sol}(\mathrm{ASP}(I'))$.

Thus we have the following diagram:

$$\begin{array}{ccc} I' = I'_0 & \subset & I = I_0 \\ \cap & & \cap \\ I'_1 & \subset & I_1 \\ \cap & & \cap \\ & \cdots & \\ \cap & & \cap \\ J' = I'_k & \subset & I_k \end{array}$$

Here I'_j (resp. I_j) is the ideal obtained by adding to I'_{j-1} (resp. I_{j-1}) all the solution polynomials associated to the ASP's contained in I'_{j-1} (resp. I_{j-1}). Further $J' = \mathrm{ASPTransform}(I')$ is the ideal which is a solution leaf (namely, J' contains the solution polynomials of all the variables and does not contain non-zero constants).

We show that I_k also is the solution leaf. Indeed, since $I'_k \subset I_k$, I_k also contains the solution polynomials of all the variables. Further, I_k does not contain non-zero constants. Indeed, if I_k contains a non-zero constant, then $\mathbb{V}(I_k) = \phi$ and I (namely the puzzle S) does not have a solution, a contradiction to the assumption. Thus I_k satisfies the two conditions of the solution leaf. Therefore, starting from I, we reach a solution leaf I_k by one ASPTransform without branching, which means that the Inoue invariant of S is trivial. ∎

The following is the first main result of this note.

Theorem 24 (Inoue invariants of 4-doku)
Any 4-doku puzzle with a unique solution has the trivial Inoue invariant $(2, 1, 1)$.

Proof In the case of 4-doku, the irredundant puzzles (with a unique solution) exist only if the number of clues is 4,5,6 ([4]). Hence, by Proposition 23, it is enough to see that the Inoue invariant of the puzzles (with a unique solution) with 4,5,6 clues is trivial.

Further, there exist only 2 essentially different (namely different modulo the action of the 4-doku symmetry group) solution boards as shown in Table 4 ([4]):

Table 4: The essentially different 2 solution boards

No.1

1	2	3	4
3	4	1	2
2	1	4	3
4	3	2	1

No.2

1	2	3	4
3	4	1	2
2	3	4	1
4	1	2	3

Now, let S be a 4-doku puzzle with k ($k = 4, 5, 6$) clues with a unique solution T. By a suitable 4-doku symmetry transformation, we may assume T is No.1 or No.2 above. It is immediate to check that all the puzzles with a unique solution obtained by deleting l ($l = 10, 11, 12$) cells from these 2 solution boards have the trivial Inoue invariant by a computation with Magma (see Appendix [6, Section 1]). Thus our theorem is proved. ∎

We now turn to the diagonal 5-doku puzzles. We first enumerate the essentially different solution boards of diagonal 5-doku.

Let S_5 be the symmetric group of degree 5 and D_4 the dihedral group of order 8. D_4 is the symmetry group of the square with center at the origin. We note that S_5 acts on the set $\mathcal{X}$ of the solution boards of the diagonal 5-doku as the permutation of numbers. D_4 also acts naturally on $\mathcal{X}$ and actually, $S_5 \times D_4$ is the symmetry group of the diagonal 5-doku. The following theorem classifies the set $\mathcal{X}$ of solution boards modulo the action of $S_5 \times D_4$.

Theorem 25 (Essentially different solution boards of diagonal 5-doku)
There are only three different solution boards modulo $S_5 \times D_4$-action as shown below.

NSB No.1

1	2	3	4	5
2	4	5	3	1
5	3	2	1	4
3	1	4	5	2
4	5	1	2	3

NSB No.3

1	2	3	4	5
3	4	5	1	2
5	1	2	3	4
2	3	4	5	1
4	5	1	2	3

NSB No.6

1	2	3	4	5
4	5	2	3	1
5	3	4	1	2
3	1	5	2	4
2	4	1	5	3

For the proof of Theorem 25, we first consider only S_5-action, forgetting D_4-action.

Definition 26 (Normalized solution boards of diagonal 5-doku)
A normalized solution board (NSB for short) is the one such that the first row is given by $a_{11} = 1, a_{12} = 2, a_{13} = 3, a_{14} = 4, a_{15} = 5$.

We note that each equivalence class of solution boards under the S_5-action contains a unique normalized solution board.

Proposition 27 (8 NSB's of diagonal 5-doku)
There are exactly 8 normalized solution boards as shown below.

NSB No.1

1	2	3	4	5
2	4	5	3	1
5	3	2	1	4
3	1	4	5	2
4	5	1	2	3

NSB No.2

1	2	3	4	5
2	5	4	1	3
4	3	2	5	1
5	4	1	3	2
3	1	5	2	4

NSB No.3

1	2	3	4	5
3	4	5	1	2
5	1	2	3	4
2	3	4	5	1
4	5	1	2	3

NSB No.4

1	2	3	4	5
3	5	2	1	4
5	1	4	3	2
4	3	5	2	1
2	4	1	5	3

NSB No.5

1	2	3	4	5
4	5	1	2	3
2	3	4	5	1
5	1	2	3	4
3	4	5	1	2

NSB No.6

1	2	3	4	5
4	5	2	3	1
5	3	4	1	2
3	1	5	2	4
2	4	1	5	3

NSB No.7

1	2	3	4	5
5	3	1	2	4
2	5	4	3	1
4	1	2	5	3
3	4	5	1	2

NSB No.8

1	2	3	4	5
5	3	4	1	2
4	5	2	3	1
2	4	1	5	3
3	1	5	2	4

Proof Let I be the ideal of the puzzle with the initial condition $a_{11} = 1, a_{12} = 2, a_{13} = 3, a_{14} = 4, a_{15} = 5$. Then we get the desired 8 solution boards by applying the Inoue solver (algorithm) to I (see Appendix [6, Subsection 2.1]). ∎

Proof of Theorem 25. If we transform the 8 NSB's with the D_4-action and then normalize, it is easy to check that there are 3 orbits: $\{\text{No.}1, 2, 4, 7\}, \{\text{No.}3, 5\}, \{\text{No.}6, 8\}$. Thus NSB No.1,3,6 are the complete representatives of the solution boards under $S_5 \times D_4$ action. ∎

We next enumerate the diagonal 5-doku puzzles with a unique solution.

Theorem 28 (Number of the diagonal 5-doku puzzles with a unique solution)
(i) There exist exactly 30964554720 diagonal 5-doku puzzles with a unique solution.
(ii) The irredundant puzzles exist only if the number of clues is 4,5,6, and the number of them is 7639680 in all.

Proof (i) Let S be a diagonal 5-doku puzzle with a unique solution T. By a suitable transformation, we may assume T is one of the three NSB's in Theorem 25. Now, by a time-consuming computation by Magma (see Appendix [6, Subsection 2.2]), we find that the number of puzzles which has the NSB No.1 (resp. No.3, No.6) as the unique solution is 32258030 (resp. 32246636, 32256282). Thus there are

$$(32258030 \times 4 + 32246636 \times 2 + 32256282 \times 2) \times 5! = 30964554720$$

puzzles with a unique solution in all. Note that we do not take the $S_5 \times D_4$-action into account for this enumeration.
(ii) By a similar computation as in (i), we can check that there are no irredundant puzzles which have NSB No.1, No.3, No.6 as a unique solution with k clues ($k \geq 7$). Also there are exactly 7996 (resp. 7920, 7920) irredundant puzzles which has No.1 (resp. No.3, No.6) as a unique solution (see Appendix [6, Subsection 2.2]). Thus there are

$$(7996 \times 4 + 7920 \times 2 + 7920 \times 2) \times 5! = 7639680$$

irredundant puzzles in all. ∎

We have a following impressive corollary.

Corollary 29 (The number of minimal and maximal clues of diagonal 5-doku)
(i) The minimal number of clues of the diagonal 5-doku puzzles with a unique solution is 4.
(ii) The maximal number of the clues of the irredundant ones is 6.

Remark 30
(i) Theorem 28 is significant in itself since this kind of precise enumeration seems to be known only for 4-doku so far ([4]).
(ii) To accomplish the computation for Theorem 28, we spent about a month using several Windows PC's simultaneously.

The following theorem is our second main result.

Theorem 31 (Inoue invariants of diagonal 5-doku)
All the diagonal 5-doku puzzles with a unique solution have the trivial Inoue invariant except the following two puzzles W_1, W_2 *(modulo* $S_5 \times D_4$*-action). These two have the Inoue invariant* $(4, 2, 2)$.

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

W1

1			4	5
3	1		5	

W2

1				
	4			
3	1			
4	5			

Proof By Theorem 28 (ii), the irredundant diagonal 5-doku puzzles exist only if the number of clues is $4, 5, 6$.

Now, let S be a puzzle with k ($k = 4, 5, 6$) clues with a unique solution T. By a suitable 5-doku symmetry transformation, we may assume T is one of the 3 NSB's in Theorem 25. We can check that all the puzzles with a unique solution obtained by deleting l ($l = 19, 20, 21$) cells from these 3 solution boards have the trivial Inoue invariant except W_1 and W_2 by a computation with Magma (see Appendix [6, Subsection 2.3]).

Furthermore, if we add any number contained in the solution board to W_1 and W_2, it is easy to check that the resulting puzzles all have the trivial Inoue invariant. Thus by Proposition 23, we are done. ∎

Remark 32
We note that W_i *(*$i = 1, 2$*) has 6 clues, whereas the minimal number of clues of the puzzles with a unique solution is 4. Since it is natural to expect that the fewer the clues, the more difficult the puzzles are, this is an interesting phenomenon.*

We finally report a partial result on the Inoue invariants of the diagonal 6-doku puzzles, whose proof we omit since it is similar to that of Theorem 31.

Theorem 33 (Inoue invariants of diagonal 6-doku restricted to the 5-clues case)
(i) The minimal number of clues of the diagonal 6-doku puzzles with a unique solution is 5.
(ii) There exist exactly 44542080 puzzles with 5 clues which have a unique solution. Among them, there exist 10540800 puzzles with a non-trivial Inoue invariant. The biggest Inoue invariant of them is $(20, 12, 5)$*, and the puzzle with this Inoue invariant* $(20, 12, 5)$ *is the following one in Table 6.*

Table 6: The diagonal 6-doku puzzle with 5 clues with the biggest Inoue invariant $(20, 12, 5)$

				5	
					4
3					
		1	6		

Remark 34
(i) As Theorem 33 shows, there are many diagonal 6-doku puzzles with a non-trivial Inoue invariant unlike 4-doku and diagonal 5-doku puzzles, even restricted to the case of minimal 5 clues.

We also note that, as seen from the case of diagonal 5-doku puzzles, the number of clues of the puzzle with the biggest Inoue invariant may be larger than the minimal number of clues (Remark 32). Hence it may well happen that the diagonal 6-doku puzzle with the biggest Inoue invariant (unknown so far) has p clues where $p > 5$.
(ii) The reason that Theorem 33 refers only to the case of 5 clues is that it takes too much time for this computation. We estimate that it will take several years (maybe more) to achieve the same computation for all the number of clues. Thus, in the case of diagonal 6-doku, we have not obtained a complete result as in the case of diagonal 5-doku.

References

[1] Bosma, W., Cannon, J., Fieker, C. and Steel, A. (eds.), *Handbook of Magma functions*, Edition 2.18, http://magma.maths.usyd.edu.au/magma/ (2011).

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

[3] Inoue, S., Efficient Singleton Set Constraint Solving by Boolean Gröbner Bases, *Communications of JSSAC* **1**(2012), 27-37.

[4] Minami, S., Harikae, S. and Nakano, T., Enumeration of the 4-doku puzzles with a unique solution (in Japanese), *Bulletin of JSSAC* **18**(2012), No.2, 21-24.

[5] Nakano, T., Minami, S., Harikae, S., Arai, K. and Watanabe, H., On the Inoue invariant of a system of Boolean polynomial equations and its applications to puzzles of Sudoku type, *preprint* (2012).

[6] Nakano, T., Arai, K. and Watanabe, H., Appendix to the note "On the Inoue invariants of puzzles of Sudoku type", http://math.ru.dendai.ac.jp/~nakano/research.html (2013).

[7] Rosenhouse, J. and Taalman, L., *Taking Sudoku Seriously*, Oxford University Press (2011).

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

[9] Sato, Y., Inoue, S., Suzuki, A. and Nabeshima, K., Boolean Gröbner Bases and Sudoku, *preprint* (2008).

[10] Sato, Y., Inoue, S., Suzuki, A., Nabeshima, K. and Sakai, K., Boolean Gröbner bases. *J. of Symbolic Computation* **46**(2011), 622-632.

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