Gröbner Basis Representations of Sudoku

Elizabeth Arnold, Stephen Lucas, and Laura Taalman

doi:10.4169/074683410X480203

Elizabeth Arnold received her B.A. in Government from Georgetown University in 1989, her M.Ed. in Mathematics education in 1993 from George Mason University, and her Ph.D. from the University of Maryland at College Park in 2000. She is currently an assistant professor at James Madison University. Her research area is in computational commutative algebra, in particular applications and algorithms for Gröbner bases. In her spare time, she enjoys spending time with her family and riding horses.

Stephen Lucas received his B.Math from the University of Wollongong in 1989 and his PhD from the University of Sydney in 1994. In 2002 he received the Michell Medal for Outstanding New Researchers from ANZIAM, Australia. He is currently an associate professor at James Madison University, after a postdoc at Harvard and a faculty position at the University of South Australia. His research interests span a wide range of topics in applied and pure mathematics, usually with a numerical bent.

Laura Taalman received her B.S. in mathematics from the University of Chicago in 1990, and her Ph.D in mathematics from Duke University in 2000. Her research includes singular algebraic geometry, knot theory, and the mathematics of puzzles. She is currently an associate professor at James Madison University, and is a recipient of the MAA Trevor Evans award and the MAA Alder Award. Laura is the author of the textbook *Integrated Calculus* and the puzzle books *Color Sudoku* and *Naked Sudoku*. In her spare time she is a total geek.

A *Sudoku board* is a 9×9 Latin square with an additional block condition. Specifically, the 81 cells of a Sudoku board are filled with the integers 1–9 in such a way that no row, column, or designated 3×3 block contains repeated entries. We will refer to these rows, columns and blocks as *regions*. A *Sudoku puzzle* is a subset of a Sudoku board that *uniquely* determines the rest of the board. For example, the Sudoku puzzle in Figure 1 is one of many puzzles whose unique solution is the Sudoku board on the right in the same figure.

The number of possible Sudoku boards is larger than the number of stars thought to be in the universe. Felgenhauer and Jarvis [**7**] showed that there are 6,670,903,752, 021,072,936,960 different Sudoku boards. Even if we wanted to count only essentially different, nonequivalent Sudoku boards, Russell and Jarvis [**13**] showed that this number is also rather large, namely 5,472,730,538.

For the purposes of illustration, in this paper it will be convenient for us to work with a simpler version of Sudoku called Shidoku. A *Shidoku board* is a 4×4 Latin

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

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

Figure 1. Sudoku puzzle and board.

square whose regions (rows, columns, and designated 2×2 blocks) each contain the integers 1–4 exactly once. In this smaller universe, it is not that difficult to show that there are 288 different Shidoku boards [**17**]. One of the things this paper will discuss is the use of Gröbner bases as an alternate method of counting Sudoku and Shidoku boards.

A *Shidoku puzzle* is a subset of a Shidoku board that uniquely determines the rest of the board. For example, Figure 2 shows a Shidoku puzzle whose unique solution is the Shidoku board in the center. The Shidoku board on the right shows the variable-assignments we use for the cells of a Shidoku board throughout this paper.

			4
4		2	
	3		1
1			

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

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

Figure 2. Shidoku puzzle, board, and variables.

Many different Sudoku solving strategies have been developed and numerous computer programs have been written using these strategies to solve, generate, and rate the difficulty level of Sudoku puzzles. The Sudopedia website [**16**] is an excellent resource for all things Sudoku, and includes dozens of strategies of various levels of sophistication. However, in this paper we are not interested in solution techniques, but rather in the inherent structure of Shidoku and Sudoku puzzles and boards.

In what follows, we develop three different ways of representing the constraints of Shidoku with a system of polynomial equations. In one case, we will explicitly show how a Gröbner basis can be used to obtain a more meaningful representation of the constraints. The Gröbner basis representation can be used to find puzzle solutions or count numbers of boards.

Polynomial representations of Shidoku

There are various ways to represent the constraints in a Shidoku board as a system of polynomials.

Sum-product Shidoku system We start with a simple, but nonetheless new, representation based on the regions of the board. Think of the 16 cells on a Shidoku board

as 16 variables that can each take on only the values 1, 2, 3, or 4. For each of these variables w, we can encode this fact with a polynomial equation of the form

$$(w-1)(w-2)(w-3)(w-4)=0. \tag{1}$$

Now suppose that $\{w, x, y, z\}$ is a set of four cells that make up a region of the Shidoku board, that is, a row, column, or 2×2 block. We need to assign four different values to these cells. It turns out that the only way to choose four numbers that sum to 10 and multiply to 24 from the set $\{1, 2, 3, 4\}$ is to choose each number exactly once. This means that we can represent the row, column, and block conditions of Shidoku by pairs of polynomial equations of the form

$$w+x+y+z-10=0 \quad \text{and} \quad wxyz-24=0. \tag{2}$$

Together with the previous 16 equations, this gives us a total of 40 polynomial equations that encompass the rules of Shidoku. We will call this representation of Shidoku by polynomial equations (1) and (2) the *sum-product Shidoku system*. To represent a given Shidoku puzzle using these polynomials, we simply add more equations as necessary to specify any given cell values. For example, using the variable-assignments in Figure 2 (right), we would add the equations $d-4=0$, $e-4=0$, $g-2=0$, $j-3=0$, $l-1=0$, and $m-1=0$ to encode the Shidoku puzzle shown in Figure 2 (left).

Roots of unity Shidoku system We can represent Shidoku as a system of polynomial equations another way, by considering pairs of cells that share a region rather than by considering entire regions at a time. This is related to the *graph coloring problem* [**11**]: Given a graph (a set of vertices connected by edges), assign each vertex a color so that each pair of vertices joined by an edge has different colors. We think of each cell of a Shidoku board as a vertex, and connect the vertices for two cells exactly when those cells lie in a common region of the board. Now, consider a proper 4-coloring of the vertices of this graph and think of each color as a number in the set $\{1, 2, 3, 4\}$. This corresponds to a valid variable-assignment in Shidoku since in the graph, no two vertices connected by an edge will be assigned the same color, and therefore on the Shidoku board, no two cells sharing a region will be assigned the same number. Figure 3 shows the part of the graph determined by the upper-left cell on the board, and a corresponding partial variable-assignment for Shidoku.

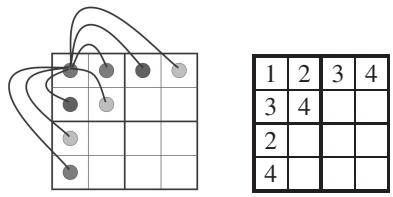


Figure 3. Part of the graph associated to Shidoku.

To represent Shidoku as a system of polynomial equations based on pairs of cells that share a region (i.e., pairs of vertices connected by an edge in Figure 3), we follow the graph coloring method of Bayer described in [**11**]. We start by replacing the entries 1, 2, 3 and 4 by the fourth roots of unity ± 1 and $\pm i$. (Note that the actual symbols used on a Shidoku board have no effect on the rules or the solution.) We can now easily

encode the fact that each cell w takes on values from the fourth roots of unity with 16 polynomial equations of the form:

$$w^4 - 1 = 0. \tag{3}$$

Now consider any two cells w and x on the Shidoku board that lie in the same row, column, or block. We already have $w^4 - 1 = 0$ and $x^4 - 1 = 0$, and therefore $w^4 - x^4 = 0$. Factoring gives $(w - x)(w + x)(w^2 + x^2) = 0$. To force w and x to take on different values, we must have $w - x \neq 0$, which means that we have a polynomial equation of the form:

$$(w + x)(w^2 + x^2) = 0. \tag{4}$$

Combining the 56 equations of this form with the previous 16 equations, we obtain a total of 72 polynomial equations that represent the structure of Shidoku. We call the representation from (3) and (4) the *roots-of-unity Shidoku system*. Gago-Vargas et al. [**9**] take a similar approach, but their system uses integers instead of roots of unity.

Boolean Shidoku system Yet another approach is to introduce four Boolean variables w_1, w_2, w_3, w_4 for each cell on the Shidoku board, where we set $w_k = 1$ when cell w takes the value k, and $w_k = 0$ otherwise. Note that we have now increased from 16 to 64 variables, each satisfying the polynomial equation:

$$w_k(w_k - 1) = 0. \tag{5}$$

This actually simplifies matters, because (5) implies that $w^2 = w$, and therefore any power of w_k can be replaced by w_k during the computation of the solution of the system of equations. Since each cell on the board can only hold one value, for any cell w, we must have exactly one of the four associated variables w_k equal to 1 and the other three equal to 0. Because each w_k can take on only the values 0 and 1, this Boolean condition can be encoded using 16 polynomial equations of the form:

$$w_1 + w_2 + w_3 + w_4 = 1. \tag{6}$$

Finally, we must require that any two cells w and x that lie in a common region have different values. This means that for each possible k, at least one of x_k or w_k must be 0. Because we are dealing with Boolean variables, we can express this requirement on each of the 56 pairs of cells that share a region with polynomial equations of the form:

$$x_1w_1 + x_2w_2 + x_3w_3 + x_4w_4 = 0. \tag{7}$$

We call the system of 136 polynomials defined by (5), (6), and (7) the *Boolean Shidoku system*. Although this system involves many more variables and polynomials than the previous two systems, there are advantages to computing in the Boolean setting. Recent work has been done concerning these methods, in particular [**3, 15**].

This Boolean Shidoku system is derived from the *exact cover problem* [**10**]: Given a set and a collection of its subsets, choose some of the subsets so that every element in the original set is in exactly one of the subsets. Consider building a matrix where each column is associated with an element of the original set, and each row corresponds to a subset with $x_{ij} = 1$ if element j is in subset i. Then the exact cover problem is equivalent to choosing a collection of rows such that each column has exactly one 1 in it. One of the most efficient algorithms to solve exact cover problems is Donald

Knuth's *Algorithm X*, as implemented in an algorithm known as *dancing links* [**12**]. It uses a backtracking, depth first, recursive approach with a particularly efficient data structure, and has been used to construct particularly fast Sudoku solvers.

There are of course many other ways to represent Shidoku, and, similarly, Sudoku, with systems of polynomials. For example, equations (2) can be replaced with the one polynomial $xy + xz + xw + yz + yw + zw = 35$ [**1**]. It is Gröbner bases that will allow us to handle these large systems of polynomials.

Gröbner basics

A Gröbner basis for a system of polynomials is a new system of polynomials with the same solutions as the original, but which is easier to solve and often has additional "nice" properties. An algorithm for computing Gröbner bases was first published by Bruno Buchberger in 1965 in his Ph.D. thesis [**4**]. Gröbner was Buchberger's thesis advisor.

To define Gröbner bases precisely, we need some abstract algebra. A *polynomial ring* is a set of polynomials in a certain number of variables where addition and multiplication of polynomials are defined in the usual way. For our purposes, the coefficients of polynomials will come from the field $\mathbb{Q}$ of rational numbers. An *ideal* in a polynomial ring is a subset of the ring that is closed under polynomial addition and closed under multiplication by all polynomials in the ring. In other words, if I is an ideal in a polynomial ring R, then for any polynomials f and g in I and any polynomial r in R, the polynomials $f + g$ and rf are also in the ideal I. An ideal can be *generated* by a set of polynomials just like a vector space can be spanned by a set of vectors. For example, if $I = \{rf + sg + th \mid r, s, t \in R\}$, then we say that f, g, and h generate I, and write $I = \langle f, g, h \rangle$.

Now, given a system of polynomials, we can look at the ideal generated by these polynomials in the polynomial ring. A Gröbner basis is a "better" generating set for this ideal. Undergraduate mathematics students are familiar with Gröbner bases in two simple cases. If the polynomials in the system are all linear, then the Gröbner basis for the ideal generated by these polynomials is the new system of polynomials in echelon form derived by Gauss-Jordon elimination. The new system has the same solution set as the original system, but it is easier to solve. Also, from the new system we can tell right away whether or not the system has one, infinitely many, or no solutions. Buchberger's algorithm actually generalizes the well-known process of transforming the matrix into echelon form. A second commonly understood example of Gröbner bases is the one-variable case. Suppose we have a system of polynomials in one variable. The greatest common divisor of these polynomials is a single polynomial whose roots encompass all common solutions to the original system. This greatest common divisor is the Gröbner basis of the ideal generated by the original system of polynomials, and once again, Buchberger's algorithm generalizes the Euclidean algorithm for computing the greatest common divisor.

Our Shidoku polynomial systems are more complicated, involving *non*-linear polynomials in *several* variables. The first step towards finding a Gröbner bases is to establish a *term ordering* on the monomials. Establishing an order is also the first step in the linear case with Gauss-Jordan elimination. When performing row reduction with two rows, we use the leading non-zero term (pivot) of one of the polynomials to combine with the other. In Gröbner basis theory we call this the *leading term*. Likewise, in the one-variable case there is a natural ordering for the monomials: that of degree. In the Euclidean algorithm, when we divide one polynomial by another, we only divide

the first (leading) term of the polynomial with the larger degree by the first term of the polynomial with the smaller degree; the lower-degree terms just follow along for the ride. For general Gröbner bases computations we need to divide one multivariate polynomial by another. We do this in the same way as in the one-variable case, by dividing just the leading terms. The term ordering that we choose determines the leading terms of the polynomials.

The term ordering that we use in this paper is the *lexicographical term ordering*, abbreviated *Lex*. Lex is almost exactly as it sounds; it is a dictionary ordering where a's beat b's and c's and the more the better. For example, if we have variables x, y, and z and choose to order the variables as $x > y > z$, then $xy >_{\text{Lex}} yz$, $xy >_{\text{Lex}} xz^2$, and $x^2 >_{\text{Lex}} x$. There are many other orderings on monomials that can be defined—in fact, infinitely many! For more information on term orderings and optimization see [**5**].

Given a chosen term ordering (in our case, Lex), the *leading term* of a polynomial f will be denoted $\text{lt}(f)$. This leading term can be broken down into the *leading coefficient* $\text{lc}(f)$ and the *leading power product* $\text{lp}(f)$, so that $\text{lt}(f) = \text{lc}(f)\text{lp}(f)$. Given any set S of polynomials in a polynomial ring, we define the *leading term ideal of* S to be the ideal $\text{Lt}(S)$ generated by the leading terms of the polynomials in S, or $\text{Lt}(S) = \langle \text{lt}(f) | f \in S \rangle$. Note that the leading term ideal of a *set* of polynomials is not necessarily equal to the leading term ideal of the *ideal* generated by that set of polynomials. For example, if $S = \{x, x + 1\}$ then $\text{Lt}(S) = \langle x \rangle$, but if $I = \langle x, x + 1 \rangle$, then $x + 1 - x = 1 \in I$. So $\text{Lt}(I) = \langle 1 \rangle = R$. When the leading term ideal of a set S of polynomials *is* equal to the leading term ideal of the ideal I generated by S, we say that S is a *Gröbner basis* for the ideal I. In other words, a set of non-zero polynomials $G = \{g_1, g_2, \ldots, g_t\} \subseteq I$ is called a *Gröbner basis* for I if and only if $\text{Lt}(G) = \text{Lt}(I)$.

Why is a Gröbner basis "better" than other generating sets for an ideal? One of the reasons is that if G is a Gröbner basis for an ideal I, then there is a simple way to use G to determine whether or not a given polynomial f is in the ideal I. This is done by the process of *reduction*. Given a polynomial f_1 we can reduce f_1 by f_2 by dividing f_1 by f_2 (that is, writing $f_1 = gf_2 + r_1$ for some polynomials g and r_1 with $\deg r_1 < \deg f_2$) and replacing f_1 by the remainder r_1. Note that f_1 can be divided by f_2 exactly when $\text{lt}(f_2)$ divides $\text{lt}(f_1)$. Note also that if f_1 reduced by f_2 gives remainder r_1, then $f_1 - \alpha f_2 = r_1$. This means that given any polynomial f, if one can reduce f by the polynomials in G until 0 is reached, then f must be *in* the ideal I. On the other hand, if in the process of reduction a polynomial is reached whose leading term is not divisible by any of the leading terms of polynomials in G, then f cannot be in the ideal I. Thus a Gröbner basis answers the "Ideal Membership Problem" in this situation.

Buchberger's algorithm guarantees the existence of a Gröbner basis for a given ideal and term order. The easiest way to find a Gröbner basis is to use a symbolic manipulation package such as Maple, Mathematica, CoCoA, GP/Pari, etc. All these systems employ Buchberger's algorithm. The interested reader can learn more about Gröbner bases and Buchberger's algorithm in [**2**], [**5**], or [**8**].

Thinking back to our goal of investigating Shidoku via analysis of polynomial systems, we can use Gröbner bases to obtain a simpler generating set of polynomials for either the sum-product system, the roots of unity system, or the Boolean system. If we include additional polynomials to represent the given values in a Shidoku puzzle, and that puzzle has a unique solution, then the system of polynomials is completely determined. The resulting Gröbner basis will consist of 16 linear polynomials that explicitly identify the solution. (In other words, Buchberger's algorithm provides us with a Shidoku solver, although not necessarily the most efficient one.) If we start with an inconsistent set of given values for which no solution board is possible, then the Gröbner basis will consist of the single polynomial 1, representing the impossible

equation $1 = 0$. If we start with too few given values to guarantee a unique solution, then the system will be underdetermined, and the Gröbner basis will consist of (possibly) some explicit solutions and some polynomials. If we do not add any additional polynomials to represent given values, then the resulting system will be a model for the structure of the Shidoku board itself.

As an example, consider the sum-product system defined by equations (1) and (2). As in Section , we use the variables $a, b, c, \ldots, p$ (from upper left to lower right) for the 16 cells on the Shidoku board. We use Lex term ordering with variables in *reverse* order $p > o > n > \cdots > a$. We choose the Lex ordering because of a very useful and well-known theorem in Gröbner basis theory (see Corollary 2.2.11 in [**2**]): Given a system of polynomial equations with a finite number of solutions, the reduced Gröbner basis for the ideal generated by these polynomials using the lexicographical term ordering is triangular.

Having a *triangular* set of polynomials is similar to having echelon form. For example, if G is a Gröbner basis in 16 variables $a < b < c < \cdots < p$, then the theorem above says that the polynomials of G can be ordered as $\{g_1, g_2, \ldots, g_s\}$, $s \geq 16$, in such a way that g_1 involves only the smallest variable a, g_2 involves only a and b and has leading term involving only b, g_3 involves only a, b, and c with leading term involving only c, and so forth until g_{16}. There may be more than 16 polynomials in G, but the first 16 will be in a form that allows us to solve the system of equations by back substitution.

Figure 4 shows the Gröbner basis for the ideal generated by the sum-product Shidoku system with no given values, as computed using Maple 12. Since Gröbner bases are all about the leading terms, we have omitted a large number of lower terms in several of the polynomials. Note that the Gröbner basis algorithm has reduced our generating set of 40 polynomials to a Gröbner basis of 17 polynomials. Even more importantly, the first 16 of the 17 polynomials are in triangular form. As we will see in the next section, this triangular Gröbner basis can be used to shed light on certain counting problems in Shidoku.

$$
\begin{aligned}
p_1 &= a^4 - 10a^3 + 35a^2 - 50a + 24 & p_9 &= i^2 + \text{lower terms} \\
p_2 &= b^3 + b^2a + \text{lower terms} & p_{10} &= j^2 - je - ja + ae \\
p_3 &= c^2 + bc + \text{lower terms} & p_{11} &= 18k + \text{lower terms} \\
p_4 &= d + c + b + a - 10 & p_{12} &= 18l + \text{lower terms} \\
p_5 &= e^2 + \text{lower terms} & p_{13} &= m + i + e + a - 10 \\
p_6 &= f + e + b + a - 10 & p_{14} &= n + j - e - a \\
p_7 &= g^2 - gb - ga + ab & p_{15} &= 18o + \text{lower terms} \\
p_8 &= h + g - b - a & p_{16} &= 18p + \text{lower terms} \\
& & p_{17} &= 9gj + \text{lower terms}
\end{aligned}
$$

Figure 4. Gröbner basis for the sum-product Shidoku system with no given values.

Counting boards using Gröbner bases

Although it is not that difficult to use simple counting methods to determine that there are exactly 288 different possible Shidoku boards, the same calculation in the 9×9 Sudoku case is not possible by hand, and requires significant computer time. In this

section, we explore how Gröbner bases can be used to count boards in the 4×4 Shidoku case. Similar methods may prove useful for counting boards of larger dimension.

If we were to count the number of Shidoku boards by hand, we might begin by counting choices starting from the upper-left corner. In that upper-left corner we have 4 choices for a. Once a is chosen we have 3 choices for b, and then 2 choices for c and only 1 choice for d. Moving to the next row, the previous choices of a and b only allow 2 choices for e, and then 1 choice for f. This means that there are $4 \cdot 3 \cdot 2 \cdot 1 \cdot 2 \cdot 1 = 48$ ways to fill in the first six cells on the board; see Figure 5.

(4)	(3)	(2)	(1)
(2)	(1)		

Figure 5. Counting the number of ways to fill the first six cells.

An examination of the leading terms in the Gröbner basis in Figure 4, shows that the six numbers we just used for counting are in fact the powers in the leading terms of the polynomials p_1 to p_6 in that basis! This interesting pattern can be explained algebraically. As we will see, none of the polynomials in the Gröbner basis have repeated roots. Hence, since p_1 is quartic, the equation $p_1 = 0$ has four possible solutions. Since p_1 only involves a, there are four choices for a. Once a choice for a is made and substituted in p_2, the equation $p_2 = 0$ only involves b and is cubic. Therefore there are three possible choices for b. In this way, moving through the first six polynomials p_1 to p_6 in the Gröbner basis we can simply multiply leading term degrees to obtain the same result of $4 \cdot 3 \cdot 2 \cdot 1 \cdot 2 \cdot 1 = 48$ solutions (a, b, c, d, e, f) for the first six variables.

This reasoning seems to suggest that in determining the number of possible Shidoku puzzles, we need only multiply the degrees of the Gröbner basis elements. If we naively did this, we would calculate that there are 384 possible Shidoku boards. But the actual answer is 288. What went wrong? Well, remember p_{17}? This polynomial has a leading term containing both g and j, which represents a branching effect that occurs when we count past the sixth cell in the puzzle.

Counting, again by hand and looking at the board, suppose we have filled in one of the 48 possible arrangements for the first six cells on the board. One such choice is shown in Figure 6, with $(a, b, c, d, e, f) = (1, 2, 3, 4, 3, 4)$. We have a choice of *two* values for the g cell, either $g = 1$ or $g = 2$. If we choose $g = 1$, then the next cell is determined as $h = 2$, and another choice arises at cell i, where we can have $i = 2$ or $i = 4$. Each of these choices for i leads to two possible solution boards, as shown in Figure 6. Note that with the choice of $g = 1$ there are two choices for j after choosing i.

1	2	3	4
3	4		

1	2	3	4
3	4	①	2
②	1_3	4	3_1
4	3_1	2	1_3

1	2	3	4
3	4	①	2
④	1_3	2	3_1
2	3_1	4	1_3

Figure 6. Four possible solutions when $g = 1$.

Now consider what happens if we back up and instead make the choice $g = 2$. This time $h = 1$, but again $i = 2$ or $i = 4$. This time, each of these choices for i leads to

just *one* possible solution board, as shown in Figure 7. Note that with this choice of g there is only one choice for j.

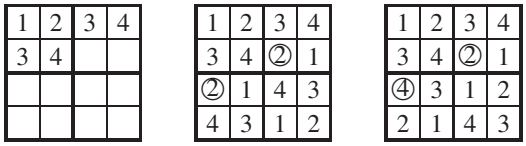


Figure 7. Two possible solutions when $g = 2$.

From the work above we see that there are $48 \cdot 1 \cdot 2 \cdot 2 = 192$ different Shidoku boards that are equivalent (up to permutation of symbols) to those we found in the $g = 1$ calculation above, and $48 \cdot 1 \cdot 2 \cdot 1 = 96$ different Shidoku boards that are equivalent (up to permutation of symbols) to the $g = 2$ case above. This gives us a total of 288 possible different Shidoku boards.

Of course the branching that happened in the calculation above is much more complicated in the 9×9 Sudoku case, so counting solutions by hand will not be feasible in that larger case. It turns out that we *can* use a Gröbner basis to count the solutions. Let J be the ideal generated by the Gröbner basis, $G = \{p_1, \ldots, p_{17}\}$. To count the number of solutions to the system of polynomial equations $p_1 = 0, \ldots, p_{17} = 0$, and hence, the number of Shidoku boards, we can simply count the number of power products that are not divisible by any of the leading power products of G. The nice triangular Lex Gröbner basis will assist us with this. The reason that we can do this involves a bit of algebra and Gröbner basis theory.

Our ideal, J, is *zero-dimensional* which means that there are only a finite number of solutions to the system of polynomial equations, $p_1 = 0, \ldots, p_{17} = 0$. Because each variable can take on only 4 values, we have at most 4^{16} possible solutions. Furthermore, J is what is known as a *radical ideal*, meaning that given any polynomial f such that some power of f is in J, then f is also in J. For example, the ideal $\langle x^2, y^2 \rangle$ is not radical, since x and y are not in the ideal. But the ideal $\langle x, y \rangle$ is radical. The fact that the sum-product Shidoku and Sudoku ideals are radical is well known (see Proposition 2.7 in [**6**]). This fact is what allowed us to assume that we had no repeated roots in the beginning of our Gröbner basis counting argument above.

Since our Shidoku ideal J is zero-dimensional, a well-known theorem in Gröbner basis theory (see Proposition 2.1.6 in [**2**]) allows us to conclude that a basis for the $\mathbb{Q}$-vector space $\mathbb{Q}[a, b, \ldots, p]/J$ consists of the cosets represented by the power products that are not divisible by any leading power product of G. Furthermore, since our ideal J is radical, Proposition 2.10 in [**6**] says that the dimension of this vector space is in fact the number of solutions to the system of polynomials. A similar argument is outlined in [**9**].

While this may sound complicated and technical, in practice it is not that bad! In order to find the number of possible Shidoku boards, we just need to count the power products that are not divisible by any leading power product of G. Let's do that.

Recall that our Gröbner basis was computed with Lex and $a < b < \cdots < p$. The nice format of the Lex Gröbner basis will make our task easy. Consider the first polynomial $p_1 = a^4 - 10a^3 + 35a^2 - 50a + 24$ in Figure 4. This is the *only* polynomial in the basis whose leading term is entirely in a, and therefore a, a^2 and a^3 are power products that are not divisible by any leading power product of a Gröbner basis polynomial. a^4, a^5, a^6 and so on, are divisible by a^4 and are not counted. The polynomial p_2 has leading term b^3 and is the only polynomial in the Gröbner basis whose leading

term is in b, so b^2 and b are also power products that are not divisible by any leading power product of a Gröbner basis polynomial. Putting these together we can also add things like a^3b^2, a^3b, a^2b^2, a^2b, and so on, to our list of power products that are not divisible by any leading power product in the Gröbner basis.

So, how many of these types of power products do we have? Any such power product is of the form $a^{r_1}b^{r_2}\cdots p^{r_{16}}$, and by the argument above we have 4 choices for r_1 (corresponding to the choices $a^0 = 1$, a^1, a^2, and a^3). Similarly we have 3 choices for r_2, 2 choices for r_3, 1 choice for r_4, and so on through p_{16} in the Gröbner basis. The resulting product of choices so far is $4 \cdot 3 \cdot 2 \cdot 1 \cdot 2 \cdot 1 \cdot 2 \cdot 1 \cdot 2 \cdot 2 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 384$, but p_{17} whose leading term is $9gj$ still needs to be considered. We need to remove all of the power products that are divisible by gj from our list. How many of these are there? Each of these non-allowable power products are of the form $a^{r_1}b^{r_2}\cdots p^{r_{16}}$ with $r_7 = 1$ and $r_{10} = 1$, so there are $4 \cdot 3 \cdot 2 \cdot 1 \cdot 2 \cdot 1 \cdot 1 \cdot 1 \cdot 2 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 = 96$ possible such power products to be removed from our list. This leaves $384 - 96 = 288$ power products *not* divisible by any leading term in the Gröbner basis, and thus a count of the 288 different possible Shidoku boards.

Other directions

We have seen how to represent the constraints of Shidoku as a system of polynomial equations, find an associated Gröbner basis for the ideal generated by that system, and use this Gröbner basis to both solve Shidoku puzzles and count the number of Shidoku boards. The natural extension of this work is to reproduce it for 9×9 Sudoku. The roots-of-unity and Boolean Shidoku systems can be extended to Sudoku systems in an obvious way. Gago-Vargas et al. [**9**] have successfully solved Sudoku puzzles using a form of the roots-of-unity system in cases with a large number of given values. Without any given values, however, finding a Gröbner basis for Sudoku systems with these methods is beyond the capabilities of a typical desktop computer. An interesting approach worth considering is the special case of Boolean Gröbner bases, where Buchberger's algorithm is modified to make use of the fact that variables can only take the values 0 or 1. The work of Bernasconi et al. [**3**] and Sato [**15**] suggests that the computational cost of finding Gröbner bases in the Boolean case is greatly reduced, and Sato's conference presentation [**14**] suggests that puzzles with a unique solution could be solved very quickly with Boolean methods.

Our tests with the sum-product system for Shidoku suggest that this new formulation is a useful one in terms of computational efficiency—if only because the initial number of polynomial equations is relatively small. Extending it to Sudoku requires some alterations however. There is more than one choice of a selection of nine (not necessarily distinct) integers from the set $\{1, 2, \ldots, 9\}$ that sum to 45 and add to 362880, namely $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $\{1, 2, 4, 4, 4, 5, 7, 9, 9\}$. It turns out that $\{-2, -1, 1, 2, 3, 4, 5, 6, 7\}$ is the smallest (in magnitude) set of nine integers for which the only way for nine elements of the set to have the sum and product of the set is to choose each number exactly once. Thus for Sudoku, equation (1) is replaced by

$$(w+2)(w+1)(w-1)(w-2)(w-3)(w-4)(w-5)(w-6)(w-7) \qquad (8)$$

for each of the 81 cells w on the board, and equation (2) is replaced by

$$\sum_{k=1}^{9} x_k - 25 = 0 \quad \text{and} \quad \prod_{k=1}^{9} x_k - 10080 = 0 \qquad (9)$$

for each set of cells $\{x_1, \ldots x_9\}$ that make up a row, column, or block region of the board.

There are a large number of open Sudoku problems, including the *minimum givens* problem, which asks for the smallest number of given values that can completely determine a Sudoku board. It is conjectured that 17 is this minimum number, and many Sudoku puzzles with unique solutions from 17 givens are known. Despite extensive computational searches by many people, no valid 16-givens puzzle has been found. Computationally, enumerating all 16-givens possibilities is not realistically feasible. This is where the potential power of the Gröbner basis approach is so appealing. The Gröbner basis techniques outlined in this paper do more than just provide a Sudoku solver. They allow us to represent the inherent structure in the rules of Sudoku in a compact way.

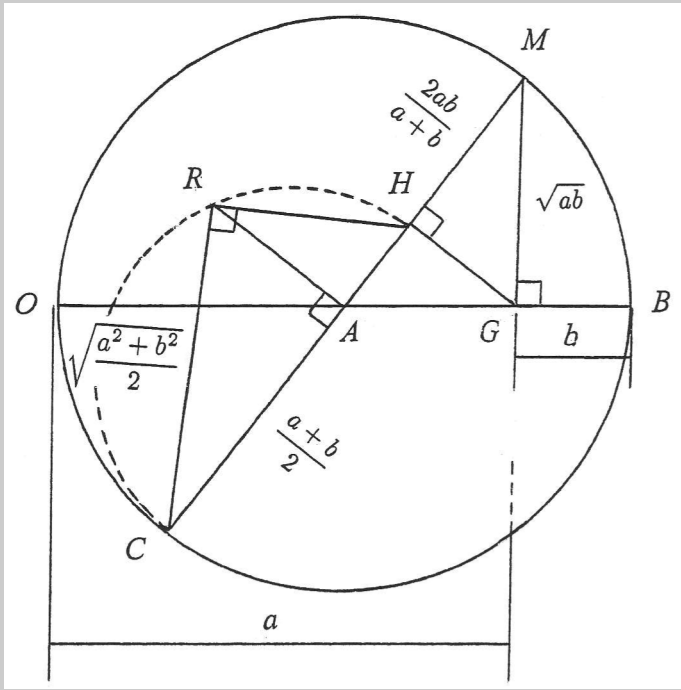
Summary. In this paper we use Gröbner bases to explore the inherent structure of Sudoku puzzles and boards. In particular, we develop three different ways of representing the constraints of Shidoku with a system of polynomial equations. In one case, we will explicitly show how a Gröbner basis can be used to obtain a more meaningful representation of the constraints. The Gröbner basis representation can be used to find puzzle solutions or count numbers of boards.

References

1. Comment by anonymous reviewer.
2. W. W. Adams and P. Loustaunau, *An Introduction to Gröbner Bases*, Graduate Studies in Mathematics, vol. 3, American Mathematical Society, Providence, RI, 1994.
3. A. Bernasconi, B. Codenotti, V. Crespi, and G. Resta, Computing Gröbner bases in the Boolean setting with applications to counting, in *Proceedings of the Workshop on Algorithm Engineering (WAE'97)*, G. Italiano and S. Orlando, eds., University of Venice, Venice, September 11–13, 1997, 209–218.
4. B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. dissertation, Inst. University of Innsbruck, Innsbruck, Austria, 1965.
5. D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms*, Springer-Verlag, New York, 1992.
6. ———, *Using Algebraic Geometry*, Springer-Verlag, New York, 1998.
7. B. Felgenhauer and F. Jarvis, Mathematics of Sudoku I, *Mathematical Spectrum* **39** (2006) 15–22.
8. R. Froberg, *An introduction to Gröbner Bases*, John Wiley, Chichester, UK, 1997.
9. J. Gago-Vargas, I. Hartillo-Hermoso, J. Martín-Morales, and J. M. Ucha-Enríquez, Sudokus and Gröbner bases: not only a divertimento, *Computer Algebra in Scientific Computing*, 155–165, *Lecture Notes in Comput. Sci.*, **4194**, Springer, Berlin, 2006. doi:10.1007/11870814_13
10. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
11. T. R. Jensen and B. Toft, *Graph Coloring Problems*, Wiley, New York, 1995.
12. D. E. Knuth, Dancing links, *Millennial Perspectives in Computer Science: Processings of the 1999 Oxford-Microsoft Symposium in Honour of Sir Tony Hoare*, J. Davies, B. Roscoe, and J. Woodcock, eds., Palgrave, Basingstoke, UK, 2000, 187–214.
13. E. Russell and F. Jarvis, Mathematics of Sudoku II, *Mathematical Spectrum* **39** (2006) 54–58.
14. Y. Sato, Boolean Gröbner bases and Sudoku, presentation at *Applications of Computer Algebra Conference*, Linz, Austria, July 2008.
15. Y. Sato, A. Nagai, and S. Inoue, On the computation of elimination ideals of Boolean polynomial rings, *Computer Mathematics: 8th Asian Symposium, ASCM 2007, Singapore, December 15–17, 2007, Revised and Invited Papers*, D. Kapur, ed., Lecture Notes in Artificial Intelligence, *5081*, Springer-Verlag, Berlin, 2008, 334–348.
16. Sudopedia, the free Sudoku reference guide, available at www.sudopedia.org/wiki, accessed December 8, 2008.
17. L. Taalman, Taking Sudoku seriously, *Math Horizons* (September 2007) 5–9.

Proof Without Words: Harmonic Mean < Geometric Mean < Arithmetic Mean < Root Mean Square < Contraharmonic Mean

Sidney Kung (sidneykung@yahoo.com), Cupertino, CA 95014



$$AM = \frac{a+b}{2},\ GM = \sqrt{ab},\ HM = \frac{2ab}{a+b},\ CH = CM - HM = \frac{a^2+b^2}{a+b}$$

$$RC = \sqrt{CA \cdot CH} = \sqrt{\frac{a+b}{2}}\sqrt{\frac{a^2+b^2}{a+b}} = \sqrt{\frac{a^2+b^2}{2}}$$

$$a > b > 0 \Rightarrow b < \frac{2ab}{a+b} < \sqrt{ab} < \frac{a+b}{2} < \sqrt{\frac{a^2+b^2}{2}} < \frac{a^2+b^2}{a+b} < a$$

$$HM < GM < AM < RC < CH$$

Also,

$$CH - AM = AH = AM - HM.$$

(That is, the contraharmonic mean exceeds the arithmetic mean by the same amount that the arithmetic mean exceeds the harmonic mean.)