

GRÖBNER BASES AND SUDOKU PUZZLES

CALEB GEIGER

ABSTRACT. As there are 6,670,903,752,021,072,936,960 distinct Sudoku boards, as computed by Felgenhauer and Jarvis [1], we instead consider the next smallest case of a 4×4 board, which we shall call a Shidoku board [2]. Using Gröbner bases, one can compute both solutions to Shidoku boards, as well as counting the number of possible distinct Shidoku boards.

We will formally explain these solutions with the idea that, although perhaps computationally intensive and with minor tweaks, they can be generalized to the case of the Sudoku board.

1. SUDOKU AND SHIDOKU

To begin, we will give a brief overview of how the games of Sudoku and Shidoku are played.

Sudoku is a game played on a 9×9 Latin square in which every row, column, and every one of the nine 3×3 subgrids that partition the 9×9 grid must contain the numbers 1 through 9 only once. The game begins with some subset of the cells each containing a number, and the objective is the fill in the rest of the cells with the previous constraints in mind.

Shidoku has essentially the same rules as Sudoku, but it is played on a 4×4 grid where the regions are rows, columns, and 2×2 grids that partition the larger grid. Additionally, Shidoku uses only the numbers 1, 2, 3, and 4.

One can imagine that there an astonishingly large quantity of Sudoku boards, and indeed there are 6,670,903,752,021,072,936,960 distinct Sudoku boards, as computed by Felgenhauer and Jarvis [1]. As such, computation on such boards is computationally intensive, and some questions have yet to be answered about Sudoku boards. For example, it is conjectured that a Sudoku board has a unique solution as long as at least 17 cells are filled out initially [3], but this is still an open problem.

Due to the computational intensity associated with the general case of Sudoku, throughout this paper we will be working in the simpler case of the Shidoku board.

2. REPRESENTATIONS OF SHIDOKU BOARDS

There are a number of methods of representing a Shidoku board algebraically, but we will list two of particular interest.

Perhaps the most naïve approach is to consider a variable for each cell, say a_{ij} where i represents the row and j represents the column that a_{ij} is in. Then there are a number of

equations that must be satisfied. Firstly, $\prod_{k=1}^4 (a_{ij} - k) = 0$ for every pair (i, j) , as it must be one of 1, 2, 3, or 4. Additionally, $\sum_{i=1}^4 a_{ij} = 10$ and $\prod_{i=1}^4 a_{ij} = 24$, as the values occurring in a column must be each of 1, 2, 3, or 4, which have this property. One can check that these three equations guarantee that the tuple $(a_{ij})_{i=1}^4$ is some permutation of $(1, 2, 3, 4)$. Similar equations can be imposed on the rows and 2×2 partitions of the grid. In total, this gives you 40 polynomial equations in 16 variables.

A perhaps more interesting representation is one we shall call the Boolean representation. Assign to the cell in the i th row and j th column, which we shall denote a_{ij} , four Boolean variables $\{a_{ijk}\}_{k=1}^4$ for which $a_{ijk} = 1$ if and only if the cell contains the value k , otherwise the variable equals 0. To represent this constraint algebraically, begin by noting that for any tuple (i, j, k) we have $a_{ijk}(a_{ijk} - 1) = 0$. This simplifies future equations considerably, as $a_{ijk}^2 = a_{ijk}$, which implies that our system will be linear in each variable. Since only one of the Boolean variables associated to each cell can be nonzero, we find that $\sum_{k=1}^4 a_{ijk} = 1$. Finally we impose that for any two cells in the same row, column, or 2×2 cell of the partition, say a_{ij} and $a_{i'j'}$ we have $\sum a_{ijk}a_{i'j'k} = 0$, as they cannot be the same value. This gives us a total of 136 equations in 64 variables.

The Boolean representation has a number of advantages over the naïve approach, which requires more time than we have in this paper, but if interested, the reader should look at Computing Groebner bases in the Boolean setting with applications to counting by Bernasconi et al. [4]

3. COUNTING THE NUMBER OF SHIDOKU BOARDS VIA GRÖBNER BASES

Counting the number of Shidoku boards is not particularly difficult using brute-force methods, but using solely brute-force methods is not computationally feasible in the case of a Sudoku board. So we will go through the computation on a Shidoku board, noting that once could do a very similar computation in the general case.

We will perform computations using the naïve representation under lex ordering $a_{ij} < a_{i'j'}$ when $i < i'$ or when $i = i'$ and $j < j'$, but with minor alterations, any reasonable representation can be made to work. We may rewrite all the defining equations of the representation so that we have some polynomials each equal to 0. Then, we can take the ideal $I \subseteq \mathbb{Q}[a_{ij}]_{1 \leq i, j \leq 4}$ generated by these polynomials. We may then calculate the Gröbner basis for I , which we will call $G = \{g_1, \dots, g_{17}\}$. In general, this Gröbner basis is finite, but with given hints there may be more or fewer equations. Due to our ordering, the leading terms of the first sixteen polynomials will be some power of $a_{11}, a_{12}, \dots, a_{44}$, respectively. Denote the power associated to a_{ij} by k_{ij} .

Our goal is to count the number of Shidoku boards; hence, we would like to count the number of solutions to the system $g_1 = 0, g_2 = 0, \dots, g_{17} = 0$. Recall that $\mathbb{Q}[a_{ij}]_{1 \leq i, j \leq 4}/I$ has a basis consisting of monomials that are not divisible by any of the leading terms of g_1, \dots, g_{17} , and moreover, $\dim \mathbb{Q}[a_{ij}]_{1 \leq i, j \leq 4}/I = |V(I)|$, and so we need only calculate this dimension by counting the number of monomials that are not divisible by the leading terms of the Gröbner basis. This computation isn't terribly difficult, but as it's not terribly

enlightening we will merely remark that the computation results in 288 different Shidoku boards.

4. SOLVING A SHIDOKU BOARD

We wrap things up by finally using Gröbner bases to solve a Shidoku board. finding a solution is relatively straightforward at this point. Simply take the ideal $I + J$, where I is the ideal generated by the defining polynomials and J is the polynomials that represent cells that have already been filled out, e.g. $a_{11} - 2$ would be the polynomial representation of the upper left corner having been filled in with a 2. Then we should get a Gröbner basis consisting of sixteen polynomials in upper triangular form, which allows us to use back substitution to find the values for all the variables, solving the board. Note that this is not the most efficient method of finding a solution, and if the reader is interested in one of the most efficient algorithms for solving Shidoku and Sudoku boards, we recommend Knuth's Dancing Links [5].

5. REFERENCES

- [1] B. Felgenhauer & F. Jarvis, **Mathematics of Sudoku I**, *Mathematical Spectrum*, 39, 2006, 15-22
- [2] Arnold E., Lucas S., Taalman L.: **Gröbner Basis Representations of Sudoku**. *The College Mathematics Journal* 41(2): 101-112, 2010. DOI: 10.4169/074683410x480203.
- [3] Forrow, Aden, and John R. Schmitt. "**Approaching the minimum number of clues Sudoku problem via the polynomial method.**" (2013).
- [4] A. Bernasconi, B. Codenotti, V. Crespi & G. Resta, **Computing Groebner bases in the Boolean setting with applications to counting**, in: G. Italiano & S. Orlando, eds., *Proceedings of the Workshop on Algorithm Engineering (WAE'97)*, University of Venice, Venice, September 11-13, 1997, 209-218.
- [5] D.E. Knuth, **Dancing Links**, in: J. Davies, B. Roscoe & J. Woodcock, *Millennial Perspectives in Computer Science: Processings of the 1999 Oxford-Microsoft Symposium in Honour of Sir Tony Hoare*, Palgrave, 2000, 187-214.