Graph Colouring and Sudoku

Applications of Gröbner Bases

Yangda Bei

A report presented for
Advanced Studies Course SNCN3101

ANU Mathematical Sciences Institute
The Australian National University
Country
Date

Abstract

The Hilbert Basis Theorem states that ideals in a polynomial ring over a Noetherian ring noetherian??? has a finite generating set. This allows us to use a terminating algorithm to find a unique set of finite elements that generate a polynomial ideal, namely, a reduced Gröbner basis. Using Gröbner bases to describe a system is very powerful and has wide applications in the field of algebraic geometry. We present methods that use Gröbner bases to solve the graph colouring problem as well as Sudoku and its variants.

Contents

1 Introduction

2 Gröbner Bases

Finding a special generating set for the ideal I. For $f \in$

Definition 2.1. A *monomial* in $x_1, \dots, x_n$ is a product of the form

$$x_1^{\alpha_1} \cdot \dots \cdot x_n^{\alpha_n},$$

where all of the exponents $\alpha_1, \dots, \alpha_n$ are non-zero

Definition 2.2. A *monomial ordering* $\succ$ on $k[x_1, \dots, x_n]$ is a relation on the set of monomials $x^\alpha, \alpha \in \mathbb{Z}^n_{\geq 0}$ (i.e., the monomial exponents), such that:

1. The relation $\succ$ is a total (or linear) ordering.
2. If $\alpha \succ \beta$ and $\gamma \in \mathbb{Z}^n_{\geq 0}$, then $\alpha + \gamma \succ \beta + \gamma$.
3. The relation $\succ$ is a well-ordering on $\mathbb{Z}^n_{\geq 0}$ (every non-empty subset has a smallest element)

Definition 2.3. A *monomial ideal* is a polynomial ideal generated by monomials.

Theorem 2.1 (Division Algorithm)**.** *Let $\succ$ be a monomial ordering and let $f_1, \dots, f_s, \in k[x_1, \dots, x_n]$ be. Then every $f \in k[x_1, \dots, x_n]$ can be written as*

$$f = q_1 f_1 + \dots q_s f_s + r,$$

where $q_i, r \in k[x_1, \dots, x_n]$ where no $LT(f_i)$ is divisible by any term of the remainder r.

Theorem 2.2 (Hilbert Basis Theorem)**.** *Every polynomial ideal $I \subseteq k[x_1, \dots, x_n]$ is finitely generated.*

Definition 2.4 (Gröbner Basis)**.** Let $I \subseteq k[x_1, \dots, x_n]$ be an ideal. Given a monomial ordering $<$, a finite subset $G = \{g_1, \dots, g_t\} \subseteq I$ different from $\{0\}$ is said to be a **Gröbner basis** if

1. G generates I, and
2. $\langle LT(g_1), \dots, LT(g_t) \rangle = \langle LT(I) \rangle$, where $LT(I)$ is the set of leading terms of non-zero elements of I.

Theorem 2.3.

3 Graph Colouring

Key Definitions

A *graph* is an ordered pair $G = (V, E)$, which consists of a nonempty set V of *nodes* and a set E of paired vertices whose elements are called *edges*

Let $G = (V, E)$ be an n-node graph. For a node $v \in V$, the *neighbourhood* of v is given by $N(v) = \{u \in V \mid \{u, v\} \in E\}$.

Theorem 3.1. *The degree sum formula states that for a given graph $G = (V, E)$,*

$$\sum_{v \in V} \deg(v) = 2|E|.$$

Definition 3.1. For a graph $G = (V, E)$, a *colouring* is a function $\mathcal{C} : V \to \{1, 2, \dots\}$ such that for all $u \in N(v)$, $\mathcal{C}(u) \neq \mathcal{C}(v)$. G is k-colourable if G can be coloured with k distinct colours.

Note that we use the terminology *colours* for node labels and labels such as *red* and *blue* are used when the number of colours are small.

Definition 3.2. The *graph polynomial* f_G associated to the graph $G = (V, E)$ is an element of the ring $\mathbb{C}[x_1, \dots, x_n]$, given by:

$$f_G := \prod_{\{u,v\} \in E} (u - v)$$

Theorem 3.2 (Hilbert's Nullstellensatz). *Let $I \subseteq \mathbb{C}[k_1, \ldots, x_n]$. Then $f \in \mathbf{I}(\mathbf{V}(I))$ if and only if there exists some integer m such that $f^m \in I$.*

Theorem 3.3. *Fix k a positive integer. Let I be the ideal generated by the polynomials $v^k - 1$ for $v \in V$. The graph G is k-colourable if and only if $f_G \notin I$.*

Proof. Let G be k-colourable. Then there is an assignment of colours to the vertices such that no two adjacent vertices have the same colour. This corresponds to a point $a \in \mathbf{V}(I)$ such that $f_G(a) \neq 0$. Hence, $f_G \notin I$.

Conversely, if G is not k-colourable, then there is at elast one pair of adjacent vertices that share a colour. This means that f vanishes for any assignment of colours, that is, f vanishes on $\mathbf{V}(I)$. By Hilbert's Nullstellensatz, there is some m such that $f^m \in I$. Since I is a radical ideal (PROVE) $\square$

For each $v \in V$, the polynomials that generate I represent the k-th roots of unity

We can also equivalently formulate the question as follows. Let $x = e^{\frac{2\pi i}{k}} \in \mathbb{C}$ be the k-th root of unity. Represent the k colours by the k distinct roots of unity, so each node is assigned $1, x, x^2, \ldots, x^{k-1}$. We can model this as

$$x_i^k - 1 = 0,\ 1 \leq i \leq n \tag{1}$$

if x_i and x_j are connected by an edge, they need to be a different colour. Since $x_i^k = x_j^k$, we have that $(x_i - x_j)(x_i^{k-1} + x_i^{k-2}x_j + \cdots + x_i x_j^{k-2} + x_j^{k-1})$. We require x_i and x_j to be different k-th roots of unity so

$$x_i^{k-1} + x_i^{k-2}x_j + \cdots + x_i x_j^{k-2} + x_j^{k-1} = 0. \tag{2}$$

Let the ideal I be generated by the polynomials in Equation 1 and for each pair of adjacent vertices x_i, x_j y the polynomials in Equation 2

The following definition relates these polynomial equations to an ideal that we can analyse.

Definition 3.3. The k-colouring of an n-node graph G is the ideal $I_{G,k} \subseteq \mathbb{C}[x_1, \ldots, x_n]$ generated by

$$\begin{aligned} \text{for all } i \in V(G): &\quad x_i^k - 1 \\ \text{for all } \{i,j\} \in E(G): &\quad x_i^{k-1} + x_i^{k-2}x_j + \cdots + x_i x_j^{k-2} + x_j^{k-1} \end{aligned}$$

3.1 3-Colouring

For $k = 3$, the ideal $I_{G,3}$ is generated by the following polynomials:

$$\begin{aligned} \text{for all } i \in V(G): &\quad x_i^3 - 1 \\ \text{for all } ij \in E(G): &\quad x_i^2 + x_i x_j + x_j^2 \end{aligned}$$

The 3-colouring probem is known to be NP-Complete.

3.2 The Chromatic Number

Another interesting question that arises from the colouring problem is what the smallest number of colours required to colour a graph. We give the

Definition 3.4. The *chromatic number* $\chi = \chi(G)$ is the smallest number such that the graph can be coloured with χ colours. A k-colourable graph is k-chromatic if its chromatic number is k.

We first give a few definitions.

Definition 3.5 (Independent Set, Vertex Cover). For a graph $G = (V, E)$, an *independent set* is a set of vertices $U \subseteq V$ such that there are no edges between any two vertices in U. The independence number $\alpha = \alpha(G)$ is the size of the largest independent set. A subset $W \subseteq V$ is a *vertex cover* such that for all $\{u, v\} \in E$, $u \in W$ or $v \in W$.

We see that a vertex cover of a graph is the complement of an independent set. Hence, a maximal independent set corresponds to a minimal vertex cover. Furthermore, each node in an independent set can be coloured the same colour.

INSERT EXAMPLE

Definition 3.6 (Cover Ideal)**.** For a graph G, the *colour ideal* $I_{cover}(G)$ is the monomial ideal

$$I_{cover}(G) := \bigcap_{\{u,v\}\in E} (u)$$

4 Sudoku

In this section, we will explain how the solutions of a Sudoku puzzle can be represented as the points in the vanishing locus of a polynomial in 81 variables. The unique solution of a well-posed Sudoku puzzle can be read off from the reduced Gröbner basis of that ideal. We can formulate the algebraic approach to solving a Sudoku puzzle as a graph colouring problem where the aim is to construct a 9-colouring of a particular graph given a partial solution. It is worth noting that since we can regard solving a Sudoku as a graph problem, graph theoretical methods are much more efficient at solving a Sudoku puzzle than using Gröbner bases. The general problem of solving a Sudoku puzzle on an $n^2 \times n^2$ grid with $n \times n$ blocks is NP-complete and has been subsequently converted to various other NP-complete problems such as constraint satisfaction [6], integer programming [1], boolean satifiability [4], and the Hamiltonian cycle problem [3].

A *Sudoku board* is a particular example of a Latin square. A *Latin square* of order n is an $n \times n$ square grid filled with n distinct symbols such that no symbol appears more than once in each row and column. Typically, Sudoku boards are 9×9 Latin squares filled with the integers 1 to 9 with the additional constraint that the numbers appears only once in each of the nine distinguished 3×3 blocks. We say that a *Sudoku puzzle* is a partial solution to a Sudoku board. A *well-posed* Sudoku puzzle is one that uniquely determines the rest of the board. We take Sudoku puzzle to mean well-posed Sudoku puzzle in the remainder of the report. Figure 1 is an example of a Sudoku puzzle and its corresponding Sudoku board.

minimum number of clues needed to solve is 17 but not mathematically proven

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

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

Figure 1: Well-posed Sudoku puzzle and board

We can formulate a Sudoku puzzle as a graph where the vertices are coloured with the numbers in the 81 squares. This gives us a graph with 81 nodes. From here on, *nodes variables*, and *cells* will be used interchangeably as they are equivalent unless otherwise stated. There is an edge between two vertices if the corresponding squares are

- in the same row,
- in the same column, and
- in the same 3×3 block.

Hence, the degree of each node is $6+6+8 = 20$. By Theorem 3.1, a typical Sudoku graph has $810 \times 2 = 810$ edges. PROVE THAT STARTING POSITION NEEDS ALL NUMBERS

Because of the high number of variables that naturally arise from such a large graph, we investigate and apply techniques to solve a smaller variant of Sudoku.

4.1 Shidoku

A *Shidoku board* is a 4×4 Latin square that whose regions (rows, columns, and designated 2×2 blocks), and similarly, a *well-posed Shidoku puzzle* is a partial solution to a Shidoku board that uniquely determines the board.

3			
	2		4
	1		
		4	

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

Figure 2: Well-posed Shidoku puzzle and board

dfd

Each of the methods presented for solving the puzzle uses values from different number systems (?) to represent the n variables given by the n cells: the roots of unity method solves for solutions in $\mathbb{C}^n$, sum-product method in $\mathbb{Z}^n$, and boolean method in $\mathbb{Z}_2^n$.

4.1.1 Roots of Unity Method

We can reformulate the techniques used to solve graph colouring in the context of sudoku. Similar to the graph colouring solution in Section 3.1, we can represent pairs of cells that share a region rather than a whole region in itself.

To start, we replace the numbers $1, 2, 3$ and 4 with the fourth roots of unity, ± 1 and $\pm i$. Note that we can make any arbitrary choice for the root of unity associated with a number. Then, we can encode this information in each cell x_i for $1 \leq i \leq 16$ which takes on values from the fourth roots of unity in 16 polynomial equations of the form

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

Next, if we fix x_i, for x_j in the same region as x_i, we have another set of polynomial equations that encodes the puzzle. Since $x_i^4 - x_j^4 = 0$, factoring gives $(x_i - x_j)(x_i + x_j)(x_i^2 + x_j^2) = 0$. Now, x_j cannot be the same number as x_i because they must be distinct roots of unity so $x_i - x_j \neq 0$. Hence, we also have polynomials of the form

$$(x_i + x_j)(x_i^2 + x_j^2) = 0. \tag{4}$$

The 56 polynomials of this form, along with the 16 from Equation 3, gives 72 polynomials total that we can generate our ideal I_{RoI}. We can then check whether the corresponding variety has solutions.

PUT EXAMPLE AND CODE

4.1.2 Sum-Product Method

We can also have a representation of the board based on its regions. Every cell can take on a number from $\{1, 2, 3, 4\}$. Fix i for $1 \leq i \leq 16$. For each x_i, we can encode this in the polynomial equation

$$(x_i - 1)(x_i - 2)(x_i - 3)(x_i - 4) = 0 \tag{5}$$

since the x_i must be one of $\{1, 2, 3, 4\}$. Now suppose we have four cells w, x, y, z in the same region. 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 use each number once. (Note that : unique set of numbers for arbitrary $n \times n$. For example, in a normal Sudoku, there is more than one choice of selecting numbers from $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ that sum to 45 and multiply to $9! = 362880$, namely $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ and $\{1, 2, 4, 4, 4, 5, 7, 9, 9\}$. We can instead assign each cell a number from $\{-2, -1, 1, 2, 3, 4, 5, 6, 7\}$ since it is the smallest set in magnitude for which each of the

0	1	2	3
4	5	6	7
8	9	1	1
012345	1	1	1

1

nine elements are picked exactly once to make the sum and product). This means that for w, x, y, z in each region, we have polynomial equations of the form

$$w + x + y + z - 10 = 0, \text{ and} \tag{6}$$

$$wxyz - 24 = 0. \tag{7}$$

We get 16 equations from Equation 5 and 24 equations from Equation 6 and Equation 7 since there are 12 regions, giving a total of 40 initial polynomials that we can use to generate the ideal I_{SP}. We also add constraints based on the cells that have been filled on any given Shidoku puzzle. For example, if we consider the Shidoku puzzle in Figure 4.1 (left), we would also add $x_1 - 3 = 0$, $x_6 - 2 = 0$, $x_8 - 4 = 0$, $x_{10} - 1 = 0$, and $x_{15} - 4 = 0$.

give example + code

4.1.3 Boolean Variable Method

Lastly, we explain the Boolean method. For each cell, we introduce four variables for each cell $x_{i,1}, x_{i,2}, x_{i,3}$, and $x_{i,4}$ for $1 \leq i \leq 16$ (note that now colorred dfdfd), where we set $x_{i,k} = 1$ when cell x_i takes the value k, and $x_{i,k} = 0$ otherwise. Encoding the individual cells for the puzzle now takes 64 variables instead of 16 (it is suggested by Bernasconi et al. [2] and Sato et al. [5] that the cost of finding a Gröbner basis is greatly reduced). For each i, we then get polynomials of the form

$$x_{i,k}(x_{i,k} - 1) = 0 \tag{8}$$

Because each $x_{i,j}$ can only take on values 0 and 1, we also get 16 polynomials of the form

$$x_{i,1} + x_{i,2} + x_{i,3} + x_{i,4} - 1 = 0 \tag{9}$$

Finally, we require any two cells x_i and x_j in the same region to have different values. Therefore, for each possible k, at least one of $x_{i,k}$ and $x_{j,k}$ must be 0. We have 56 polynomial equations of the form

$$x_{i,1}x_{j,1} + x_{i,2}x_{j,2} + x_{i,3}x_{j,3} + x_{i,4}x_{j,4} = 0. \tag{10}$$

We get a total of 136 initial polynomials that we can use to generate our ideal I_{BV} and find a reduced Gröbner basis for.

5 Origami

Appendices

fdfd

References

[1] Andrew Bartlett, Timothy P Chartier, Amy N Langville, and Timothy D Rankin. An integer programming model for the sudoku problem. *Journal of Online Mathematics and its Applications*, 8(1), 2008.

[2] Anna Bernasconi, Bruno Codenotti, Valentino Crespi, and Giovanni Resta. Computing groebner bases in the boolean setting with applications to counting. In *WAE*, pages 209–218, 1997.

[3] Michael Haythorpe. Reducing the generalised sudoku problem to the hamiltonian cycle problem. *AKCE International Journal of Graphs and Combinatorics*, 13(3):272–282, 2016.

[4] Ines Lynce Ist, Inês Lynce, and Joël Ouaknine. Sudoku as a sat problem. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (AIMATH)*, pages 1–9, 2006.

[5] Yosuke Sato, Akira Nagai, and Shutaro Inoue. On the computation of elimination ideals of boolean polynomial rings. In *Computer Mathematics: 8th Asian Symposium, ASCM 2007, Singapore, December 15-17, 2007. Revised and Invited Papers*, pages 334–348. Springer, 2008.

[6] Helmut Simonis. Sudoku as a constraint problem. In *CP Workshop on modeling and reformulating Constraint Satisfaction Problems*, volume 12, pages 13–27. Citeseer, 2005.