Bases de Groebner y Aplicaciones en Teoría de Grafos

Daniel Mondurrey Ortín

Trabajo de Fin de Grado en Matemáticas
Universidad de Zaragoza

Directores del trabajo:
José Ignacio Cogolludo Agustín y
Jorge Martín Morales

28 de junio de 2019

Prólogo

Muchos de los problemas que nos rodean a diario que a simple vista son difíciles de resolver (reparto de turnos de trabajo, organización de semáforos y señales, cubrir el transporte público de una ciudad) es posible resolverlos aplicando conocimientos matemáticos y, en general, esos conocimientos provienen de la rama del álgebra. A raíz de estas cuestiones, nos planteamos un problema clásico con diversas utilidades: el problema de coloreado de grafos. Este problema consiste en que a partir de un grafo conexo, no dirigido y sin bucles nos cuestionemos si sus vértices pueden ser coloreados con un determinado número de colores de tal manera que los vértices adyacentes tengan asignados distintos colores.

En cuanto hablamos de álgebra, es inevitable pensar en ecuaciones y sistemas. Hay métodos sencillos de resolución cuando los sistemas son lineales pero en cuanto nos encontramos con que no hay linealidad y numerosas variables el problema se complica. Un concepto clave para solucionar este inconveniente son las bases de Groebner que, de una manera sencilla, nos permitirá conocer todas las soluciones de un sistema. Estas bases de ideales se forman de tal manera que para todo polinomio del ideal, el término principal de este polinomio es divisible por alguno de los términos principales de los polinomios que componen dicha base.

El problema de coloreado de grafos se puede enfocar de dos formas distintas:

- ¿Cuál es el número mínimo de colores necesario para colorear el grafo?
- ¿De cuántas formas se puede colorear el grafo con k colores?

Para estas contestar a estas preguntas, se han dado muchos métodos y algortimos. Nosotros vamos a dar la solución al problema usando bases de Groebner. Para ello debemos buscar la forma de trasladar el problema de coloreado de grafos a un sistema de ecuaciones para poner en práctica las técnicas mencionadas.

De esta forma pondremos solución a un problema bastante complejo, que puede resultar interesante para realizar sistemas criptográficos.

El trabajo se distribuye de la siguiente manera, en el Capítulo 1 recogemos conceptos previos y resultados de polinomios, como el algortimo de la división, para así continuar con los ideales de polinomios e ideales monomiales. Visto esto, explicaremos el concepto de bases de Groebner, sus propiedades, el algoritmo de Buchberger (con él obtendremos estas bases de manera efectiva) y las posibles aplicaciones de estos resultados. El capítulo está basado en el texto [1].

En el Capítulo 2, comenzamos hablando de grafos y conceptos básicos para así plantear el problema de coloreado. Una vez planteado, hablaremos de dos formas distintas de llevar el problema a un sistema de ecuaciones para poder resolverlo usando los conocimientos del capítulo anterior.

Una vez vista la parte más teórica, en el Capítulo 3 obtenemos datos objetivos del problema de coloreado de grafos. Haciendo una reducción de parámetros contrastaremos en gráficos la dificultad del problema utilizando como herramienta principal Sage. Concluiremos con una breve adaptación del problema a la criptografía.

Summary

A graph coloring is an assignment of labels, called colors, to the vertices of a graph such that no two adjacent vertices share the same color. An easy way to address this problem is converting the problem in an equation system and, because of the difficulty, solving it using a polynomial basis called Groebner basis.

This document consists of 3 chapters. A short description of each chapter will be given inmediately after.

Chapter 1: Groebner Bases

Simple definitions about general algebra are given in a brief introduction to begin with the division algorithm for multivariate polynomials in a specific field. An example of this proves an evidence of the remainder, it is not uniquely characterizated as it depends on a monomial ordering and the order of the list of dividing polynomials.

For solving this ambiguity, we need to introduce the concept of Groebner Basis. It is a specifical basis of a polynomial ideal I in which the leading term of any element of I is divisible by one of the leading terms of the basis elements. When a polynomial is divided by a Groebner Basis, the remainder will be unique. Groebner bases have many useful propieties and for that reason we need to konw how to obtain them. This will be done with the Buchberger algorimth.

Chapter 2: Graph Theory and Graph Coloring

Graph coloring is the main topic of this chapter. After defining the kind of graphs we need, we will focus on this problem. Concepts about graph coloring will be given and some examples of special cases.

Finally, by imposing some conditions we obtain two ways of transforming the graph coloring problem into an equation system. This lets us use Groebner Bases to solve this systems. Moreover, the Sage code of this proccess will be given at the end of the project in Appendix 3.3.

Chapter 3: Groebner Bases and Graphs

The main point of the coloring problem is its difficulty. For that reason, we will analize this complexity ploting graphics with Sage. In this graphic we will represent the time of resolution of the problem depending on the number of vertex and edges.

To sum up, the difficulty of the problem make us think in cryptography. A brief explanation of how can we use the coloring problem to make a cryptosystem will be given at the end of the chapter.

Índice general

Capítulo 1

Bases de Groebner

Antes de entrar en materia, es necesario conocer los siguientes conceptos básicos de álgebra.

Definición 1.1. Un *monomio* en $x_1, ..., x_n$ es un producto de la forma

$$x^\alpha = x_1^{\alpha_1} \cdot x_2^{\alpha_2} \cdots x_n^{\alpha_n},$$

donde los exponentes $\alpha = (\alpha_1, ..., \alpha_n)$ son enteros no negativos. El *grado total* de un monomio es $|\alpha| = \alpha_1 + \cdots + \alpha_n$.

Definición 1.2. Un *polinomio* f en $x_1, ..., x_n$ con coeficientes en $\mathbb{K}$ es una combinación lineal finita (con coeficientes en $\mathbb{K}$) de monomios. Se pueden escribir de la siguiente forma:

$$f = \sum_\alpha a_\alpha x^\alpha, \quad a_\alpha \in \mathbb{K},$$

con un número finito de n-tuplas $\alpha = (\alpha_1, ..., \alpha_n)$.

Definición 1.3. Sea f un polinomio en $\mathbb{K}[x_1, ..., x_n]$.

i) a_α se dice que es *coeficiente del monomio* x^α.

ii) $a_\alpha x^\alpha$ se dice que es un *término de* f.

iii) El *grado total* de f es el máximo de los grados totales de los monomios, $|\alpha|$, con coeficiente $a_\alpha \neq 0$.

Definición 1.4. Se dice que un subconjunto $I \subset \mathbb{K}[x_1, ..., x_n]$ es un *ideal* si satisface:

i) $0 \in I$.

ii) Si $f, g \in I$, entonces $f + g \in I$.

iii) Si $f \in I$ y $h \in \mathbb{K}[x_1, ..., x_n]$, entonces $hf \in I$.

El conjunto de polinomios con variables $x_1, ..., x_n$ y con coeficientes en $\mathbb{K}$, siendo $\mathbb{K}$ un cuerpo, forman el anillo de polinomios que denotaremos $\mathbb{K}[x_1, ..., x_n]$.

Definición 1.5. Sean $f_1, ..., f_s$ polinomios en $\mathbb{K}[x_1, ..., x_n]$. Sea:

$$V(f_1, ..., f_s) = \{(a_1, ..., a_n) \in \mathbb{K}^n : f_i(a_1, ..., a_n) = 0 \ \forall 1 \leq i \leq s\}$$

Llamamos a $V(f_1, ..., f_s)$ la *variedad afín* definida por $f_1, ..., f_s$.

Así, una variedad afín $V(f_1,...,f_s) \subset \mathbb{K}^n$ es el conjunto de soluciones del sistema de ecuaciones:

$$\begin{cases} f_1(x_1,...,x_n) = 0 \\ \vdots \\ f_s(x_1,...,x_n) = 0 \end{cases}$$

Esta última definición será bastante útil para resolver sistemas de ecuaciones con las bases de Groebner, que serán definidas más adelante.

1.1. Algoritmo de la división

Lema 1.1. *Sea $\mathbb{K}$ un cuerpo y g un polinomio no nulo en $\mathbb{K}[x]$. Entonces todo $f \in \mathbb{K}[x]$ puede escribirse de la siguiente forma:*

$$f = q \cdot g + r,$$

donde $q, r \in \mathbb{K}[x]$ y, o bien $r = 0$, o bien el grado de r es menor que el de g. Además, q y r son únicos.

El algoritmo de la división en polinomios de una variable nos asegura la existencia de un cociente y un resto únicos. Esto se debe a que en polinomios de una variable hay un único orden monomial natural. Para enunciar el algoritmo de la división en polinomios de dos o más variables es necesario establecer un orden total en el conjunto de monomios que sea compatible con la multiplicación y sustituya al grado para polinomios de una variable.

Definición 1.6. Un *orden monomial* en $\mathbb{K}[x_1,...,x_n]$ es una relación $>$ en $\mathbb{Z}^n_{\geq 0}$, o equivalentemente una relación sobre el conjunto de los monomios x^α, $\alpha \in \mathbb{Z}^n_{\geq 0}$ que satisface:

i) $>$ es un orden total (o lineal) en $\mathbb{Z}^n_{\geq 0}$.

ii) Si $\alpha > \beta$ y $\gamma \in \mathbb{Z}^n_{\geq 0}$, entonces $\alpha + \gamma > \beta + \gamma$.

iii) $>$ es un buen orden en $\mathbb{Z}^n_{\geq 0}$. Esto significa que todo subconjunto no vacío de $\mathbb{Z}^n_{\geq 0}$ tiene un elemento más pequeño respecto del orden $>$.

El siguiente lema ayuda a entender qué significa que $>$ sea un buen orden.

Lema 1.2. *Una relación de orden $>$ en $\mathbb{Z}^n_{\geq 0}$ es un buen orden si y solo si no existen sucesiones infinitas estrictamente decrecientes en $\mathbb{Z}^n_{\geq 0}$*

$$\alpha(1) > \alpha(2) > \alpha(3) > \cdots.$$

Definición 1.7. Sea $\alpha = (\alpha_1,\ldots,\alpha_n), \beta = (\beta_1,\ldots,\beta_n) \in \mathbb{Z}^n_{\geq 0}$. Decimos que $\alpha >_{lex} \beta$ si la primera componente no nula a la izquierda del vector $\alpha - \beta \in \mathbb{Z}^n_{\geq 0}$ es positiva. Escribiremos $x^\alpha >_{lex} x^\beta$ si $\alpha >_{lex} \beta$. A $>_{lex}$ se le llama *orden lexicográfico*.

Definición 1.8. Sean $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$, decimos que $\alpha >_{grlex} \beta$ si

$$\begin{cases} \text{o bien} & |\alpha| > |\beta|, \\ \text{o bien} & |\alpha| = |\beta| \text{ y } \alpha >_{lex} \beta. \end{cases}$$

A $>_{grlex}$ se le llama *orden lexicográfico graduado*.

Definición 1.9. Sean $\alpha, \beta \in \mathbb{Z}^n_{\geq 0}$. Decimos que $\alpha >_{grevlex} \beta$ si o bien $|\alpha| > |\beta|$ o bien $|\alpha| = |\beta|$ y la primera componente no nula a la derecha de $\alpha - \beta$ es negativa. A este orden, $>_{grevlex}$, se le llama *orden graduado reverso-lexicográfico*.

A partir de ahora usaremos a menudo la terminología que se define a continuación:

Definición 1.10. Sea $\sum_\alpha a_\alpha x^\alpha$ un polinomio no nulo en $\mathbb{K}[x_1,...,x_n]$ y sea $>$ un orden monomial.

i) El *multigrado* de f es
$$mgr(f) = \text{máx}(\alpha \in \mathbb{Z}^n_{\geq 0} : a_\alpha \neq 0).$$

ii) El *coeficiente principal* de f es
$$LC(f) = a_{mgr(f)} \in \mathbb{K}.$$

iii) El *monomio principal* de f es
$$LM(f) = x^{mgr(f)}.$$

iv) El *término principal* de f es
$$LT(f) = LC(f) \cdot LM(f).$$

Visto esto, se puede enunciar ya el algortimo que permite dividir polinomios de 2 o más variables:

Teorema 1.3 (Algoritmo de la división en $\mathbb{K}[x_1,...,x_n]$). *Dado un orden monomial $>$ en $\mathbb{Z}^n_{\geq 0}$ y sea $F = (f_1,...,f_s)$ una s-tupla ordenada de polinomios en $\mathbb{K}[x_1,...,x_n]$. Entonces todo $f \in \mathbb{K}[x_1,...,x_n]$ puede escribirse de la siguiente forma:*

$$f = a_1 f_1 + a_2 f_2 + \cdots + a_s f_s + r$$

donde $a_i, r \in \mathbb{K}[x_1,...,x_n]$, y $r = 0$ o bien es un polinomio de cuyos monomios ninguno es divisible por $LT(f_1),...,LT(f_s)$. Además, si $a_i f_i \neq 0$, entonces tenemos que

$$mgr(f) \geq mgr(a_i f_i).$$

Ejemplo 1. Sean $f = x^3y + 2xy - y^2, f_1 = xy + 1, f_2 = y - 1 \in \mathbb{K}[x,y]$. Vamos a aplicar el algoritmo de la división dado el orden monomial lexicográfico (con $x > y$) para dividir f entre (f_1, f_2). Los tres polinomios están ordenados según el orden dado.

El objetivo es encontrar $a_1, a_2, r \in \mathbb{K}[x,y]$ tales que $f = a_1 f_1 + a_2 f_2 + r$. Para encontrarlos, buscaremos términos de f que sean divisibles por $LT(f_1) = xy$ y, en caso de que no haya ninguno, haremos lo mismo con f_2 en vez de f_1:

$$x^3y + 2xy - y^2 = (x^2+2)\cdot(xy+1) - x^2 - y^2 - 2.$$

Una vez hecho esto, repetimos el proceso con $r = -x^2 - y^2 - 2$ en vez de f. Como $LT(f_2) = y$ divide a algunos términos de r, tenemos:

$$-x^2 - y^2 - 2 = (-y-1)\cdot(y-1) - x^2 - 3.$$

Así obtenemos que $a_1 = x^2 + 2,\ a_2 = -y - 1$ y $r = -x^2 - 3$.

Si realizamos la división de f entre (f_2, f_1) para obtener $f = b_2 f_2 + b_1 f_1 + r'$, obtenemos que $b_2 = 2x - y - 1,\ b_1 = 0$ y $r' = x^3y + 2x - 1$.

Luego el orden en que tomamos los polinomios para realizar la división de f determinan los términos a_i, del enunciado. Esto queda de manifiesto en el ejemplo anterior:

$$x^3y + 2xy - y^2 = (x^2+2)\cdot(xy+1) + (-y-1)\cdot(y-1) + (-x^2-3),$$
$$x^3y + 2xy - y^2 = (2x-y-1)\cdot(y-1) + 0\cdot(xy+1) + (x^3y + 2x - 1).$$

Esta ambigüedad quedará resuelta con la introducción del concepto de bases de Groebner.

1.2. Ideales monomiales

El concepto de ideal monomial y sus propiedades son clave para definir bases de Groebner y probar resultados vinculantes.

Definición 1.11. Un ideal $I \subset \mathbb{K}[x_1,...,x_n]$ es un *ideal monomial* si existe un subconjunto $A \subset \mathbb{Z}_{\geq 0}^n$ tal que I está formado por todos los polinomios que son sumas finitas de la forma $\sum_{\alpha \in A} h_\alpha x^\alpha$ con $h_\alpha \in \mathbb{K}[x_1,...,x_n]$. Se denota $I = \langle x^\alpha : \alpha \in A \rangle$.

Proposición 1. Sea $I = \langle x^\alpha : \alpha \in A \rangle$ un ideal monomial. Entonces un monomio x^β está en I si y solo si x^β es divisible por x^α para algún $\alpha \in A$.

Demostración. Si x^β es multiplo de x^α para algún $\alpha \in A$, entonces por la defínción de ideal $x^\beta \in I$.

Ahora, si $x^\beta \in I$, entonces $x^\beta = \sum_{i=1}^s h_i x^{\alpha(i)}$, donde $h_i \in \mathbb{K}[x_1,...,x_n]$ y $\alpha(i) \in A$. Si desarrollamos cada h_i como una combinación lineal de monomios, se ve que cada término es divisible por algún $x^{\alpha(i)}$. Luego x^β es divisible por algún x^α con $\alpha \in A$. □

Proposición 2. Sea I un ideal monomial y sea $f \in \mathbb{K}[x_1,...,x_n]$. Entonces son equivalentes:

i) $f \in I$.

ii) Todo término de f está en I.

iii) f es una combinación lineal en $\mathbb{K}$ de monomios en I.

Demostración. Las implicaciones $iii) \Rightarrow ii) \Rightarrow i)$ son triviales. Para probar que $i) \Rightarrow ii)$ se aplica un argumento similar a la demostración anterior. □

Proposición 3. Dos ideales monomiales son iguales si y solo si contienen los mismos monomios.

Demostración. Esta proposición es consecuencia de la anterior. □

A continuación vamos a ver el lema de Dickson, que nos asegura que todo ideal monomial en $\mathbb{K}[x_1,...,x_n]$ esta finitamente generado. Se incluye su demostración ya que es el resultado más relevante de este apartado:

Lema 1.4 (Dickson). *Un ideal monomial $I = \langle x^\alpha : \alpha \in A \rangle \subset \mathbb{K}[x_1,...,x_n]$ puede escribirse de esta forma: $I = \langle x^{\alpha(1)},...,x^{\alpha(s)} \rangle$ donde $\alpha(1),...,\alpha(s) \in A$. En particular, I tiene una base finita.*

Demostración. Vamos a demostrar el lema procediendo por inducción sobre n, el número de variables:

Si $n = 1$, I está generado por los monomios de la forma x_1^α, con $\alpha \in A$. Dado que $A \subset \mathbb{Z}_{\geq 0}$ existe un elemento minimal para A. Sea β el menor elemento de A, es decir, $\beta \leq \alpha$ para todo $\alpha \in A$. Entonces, x^β divide a los demás generadores x^α. De aquí se sigue que $I = \langle x^\beta \rangle$.

Ahora suponemos que $n > 1$ y el lema es cierto para $n-1$. Escribimos la variable n-ésima como y, distinguiéndola de las $n-1$ variables para las que suponemos cierto el resultado. Así, los monomios en $\mathbb{K}[x_1,...,x_{n-1},y]$ son de la forma $x^\alpha y^m$ con $\alpha \in \mathbb{Z}_{\geq 0}^{n-1}$ y $m \in \mathbb{Z}_{\geq 0}$.

Supongamos que $I \subset \mathbb{K}[x_1,...,x_{n-1},y]$ es un ideal monomial. Para encontrar sus generadores, definimos J como un ideal en $\mathbb{K}[x_1,...,x_{n-1}]$ generado por los monomios x^α tales que $x^\alpha y^m \in I$ para algún $m \geq 0$. Como J es un ideal monomial en $\mathbb{K}[x_1,...,x_{n-1}]$, por hipótesis de inducción tenemos que este está generado por un número finito de monomios x^α, es decir, $J = \langle x^{\alpha(1)},...,x^{\alpha(s)} \rangle$.

Para cada $i \in \{1,...,s\}$, tenemos que $x^{\alpha(i)}y^{m_i} \in I$ para algún $m_i \geq 0$. Sea m el máximo de los m_i. Entonces, para cada $k \in \{1,...,m-1\}$, consideramos el ideal $J_k \subset \mathbb{K}[x_1,...,x_{n-1}]$ generado por los monomios x^β tales que $x^\beta y^k \in I$. Usando otra vez la hipótesis de inducción, tenemos que J_k está generado por un número finito de monomios, es decir, $J_k = \langle x^{\alpha_k(1)},...,x^{\alpha_k(s)}\rangle$.

Podemos afirmar que I está generado por los siguientes monomios:

$$\begin{array}{lcl} \textit{Para } J & : & x^{\alpha(1)}y^m,...,x^{\alpha(s)}y^m, \\ \textit{para } J_0 & : & x^{\alpha_0(1)},...,x^{\alpha_0(s_0)}, \\ \textit{para } J_1 & : & x^{\alpha_1(1)}y,...,x^{\alpha_1(s_1)}y, \\ & \vdots & \\ \textit{para } J_{m-1} & : & x^{\alpha_{m-1}(1)}y^{m-1},...,x^{\alpha_{m-1}(s_{m-1})}y^{m-1}. \end{array}$$

Es fácil ver que cada monomio de I es divisible por uno de la lista anterior. Podemos distinguir dos casos:

- Si $r \geq m$, $x^\alpha y^r$ es divisible por un $x^{\alpha(i)}y^m$.
- Si $r < m$, $x^\alpha y^r$ es divisible por un $x^{\alpha_r(i)}y^m$.

Se sigue de la proposición 1 que la lista de monomios genera un ideal con los mismos monomios que I. Por la proposición 3, tenemos que I es el mismo ideal que el generado por la lista de monomios.

Para concluir, veamos que el conjunto finito de generadores puede elegirse de entre el conjunto de generadores de I. Volvemos a denotar las variables como $x_1,...,x_n$, entonces $I = \langle x^\alpha : \alpha \in A\rangle \subset \mathbb{K}[x_1,...,x_n]$. Hemos visto que $I = \langle x^{\beta(1)},...,x^{\beta(s)}\rangle$ con $x^{\beta(i)}$ ciertos monomios en I. Como $x^{\beta(i)} \in I = \langle x^\alpha : \alpha \in A\rangle$, por la proposición 1 tenemos que cada $x^{\beta(i)}$ es divisible por $x^{\alpha(i)}$ para un $\alpha(i) \in A$. Luego $I = \langle x^{\alpha(1)},...,x^{\alpha(s)}\rangle$, como queríamos demostrar.

□

Corolario 1. Sea $>$ una relación en $\mathbb{Z}^n_{\geq 0}$ satisfaciendo:

i) $>$ es un orden total en $\mathbb{Z}^n_{\geq 0}$.

ii) Si $\alpha > \beta$ y $\gamma \in \mathbb{Z}^n_{\geq 0}$, entonces $\alpha+\gamma > \beta+\gamma$.

Entonces $>$ es un buen orden si y solo si $\alpha \geq 0$ para todo $\alpha \in \mathbb{Z}^n_{\geq 0}$.

La demostración de este resultado la podemos encontrar en [1, pág 71].

1.3. Bases de Groebner

Definición 1.12. Sea $I \subset \mathbb{K}[x_1,...,x_n]$ un ideal distinto de $\{0\}$.

i) Denotamos $LT(I)$ al conjunto de términos principales de los elementos de I. Así,

$$LT(I) = \{cx^\alpha : \text{ si } \exists f \in I \text{ con } LT(f) = cx^\alpha\}.$$

ii) Denotamos $\langle LT(I)\rangle$ al ideal generado por los elementos de $LT(I)$.

Proposición 4. Sea $I \subset \mathbb{K}[x_1,...,x_n]$ un ideal:

i) $\langle LT(I)\rangle$ es un ideal monomial.

ii) Existen $g_1,...,g_s \in I$ tales que $\langle LT(I)\rangle = \langle LT(g_1),...,LT(g_s)\rangle$.

Demostración. i) Los $LM(g)$ con $g \in I \setminus \{0\}$ generan el ideal monomial $\langle LM(g) : g \in I \setminus \{0\}\rangle$. Como $LM(g)$ y $LT(g)$ difieren en una constante no nula, $\langle LT(g) : g \in I \setminus \{0\}\rangle = \langle LT(I)\rangle$. Ya tenemos que $\langle LT(I)\rangle$ es un ideal monomial.

ii) Como $\langle LT(I)\rangle$ está generado por $LM(g),\ g \in I \setminus \{0\}$, tenemos que, por el lema 1.4 (Dickson), $\langle LT(I)\rangle = \langle LM(g_1),...,LM(g_s)\rangle$ por un número finito s de $g_i \in I$. Como $LT(g_i)$ y $LM(g_i)$ difieren en una constante no negativa, se sigue que $\langle LT(I)\rangle = \langle LT(g_1),...,LT(g_s)\rangle$.

□

Teorema 1.5 (Teorema de la base de Hilbert). *Todo ideal $I \subset \mathbb{K}[x_1,...,x_n]$ tiene un conjunto generador finito. Es decir, $I = \langle g_1,...,g_s\rangle$ con $g_1,...,g_s \in I$.*

Demostración. Si $I = \{0\}$, este está generado por el conjunto finito $\{0\}$. Si I contiene algún polinomio no nulo, entonces podemos construir un sistema generador $g_1,...,g_s$ para I como vamos a ver.

Por la proposición 4, existen $g_1,...,g_s \in I$ tal que $\langle LT(I)\rangle = \langle LT(g_1),...,LT(g_s)\rangle$. Vamos a ver que $I = \langle g_1,...,g_s\rangle$.

Tenemos que $\langle g_1,...,g_s\rangle \subset I$ ya que cada $g_i \in I$. Sea $f \in I$ un polinomio cualquiera. Si aplicamos el algoritmo de la división (teorema 1.3) para dividir f entre $(g_1,...,g_s)$, obtenemos la siguiente expresión:

$$f = a_1g_1 + \cdots + a_sg_s + r.$$

Si probamos que $r = 0$ hemos terminado. Utilizaremos reducción al absurdo suponiendo que $r \neq 0$. En tal caso sabemos que ninguno de los monomios de r es divisible por $LT(g_1),...,LT(g_s)$. Despejando:

$$r = f - a_1g_1 - \cdots - a_sg_s.$$

Dado que tanto f como g_i están en I se tiene que $r \in I$, entonces $LT(r) \in \langle LT(I)\rangle = \langle LT(g_1),...,LT(g_s)\rangle$, y por la proposición 1 se sigue que $LT(r)$ tiene que ser divisible por algún $LT(g_i)$ lo cual contradice lo dicho anteriormente. Por tanto, $r = 0$. Así:

$$f = a_1g_1 + \cdots + a_sg_s + 0 \in \langle g_1,...,g_s\rangle,$$

y tenemos que $I \subset \langle g_1,...,g_s\rangle$.

Luego hemos obtenido por doble contenido que $I = \langle g_1,...,g_s\rangle$.

□

Definición 1.13. Dado un orden monomial, un subconjunto finito $G = \{g_1,...,g_s\}$ de un ideal I se dice que es una *base de Groebner* si

$$\langle LT(g_1),...,LT(g_s)\rangle = \langle LT(I)\rangle.$$

De una manera menos formal, un conjunto $\{g_1,...,g_s\}$ es una base de Groebner si y solo si el término principal de cualquier elemento de I es divisible por alguno de los $LT(g_i)$ (se sigue de la Proposición 1).

Corolario 2. Dado un orden monomial, entonces todo ideal $I \subset \mathbb{K}[x_1,...,x_n]$ no nulo tiene una base de Groebner. Además, toda base de Groebner de un ideal I es base de I.

Demostración. Dado un ideal no nulo, el conjunto $G = \{g_1, ..., g_s\}$ obtenido con el procedimiento introducido en la demostración del Teorema 1.5 es una base de Groebner por definición.

Para la segunda parte del enunciado, tenemos que $\langle LT(I)\rangle = \langle LT(g_1), ..., LT(g_s)\rangle$ luego, si nos fijamos en la demostración del Teorema 1.5, $I = \langle g_1, ..., g_s\rangle$. Por tanto G es una base de I.

□

Del teorema de la base de Hilbert 1.5, tenemos una importante consecuencia geométrica relacionada con las variedades afines. A través de estas últimas conseguiremos resolver sistemas de ecuaciones. Primero definimos algo similar a las variedades pero con ideales:

Definición 1.14. Sea $I \subset \mathbb{K}[x_1, ..., x_n]$ un ideal. Denotaremos por $V(I)$ a:

$$V(I) = \{(a_1, ..., a_n) \in \mathbb{K}^n : f(a_1, ..., a_n) = 0 \ \forall \ f \in I\}.$$

Como consecuencia del Teorema de la base de Hilbert 1.5, aunque un ideal contenga infinitos polinomios distintos, el conjunto $V(I)$ se puede definir por un número finito de ecuaciones polinómicas.

Proposición 5. El conjunto $V(I)$ es una variedad afín. En particular, si $I = \langle f_1, ..., f_s\rangle$, entonces $V(I) = V(f_1, ..., f_s)$.

Demostración. Por el Teorema 1.5, $I = \langle f_1, ..., f_s\rangle$. Veamos que $V(I) = V(f_1, ..., f_s)$. Como $f_i \in I$, si $f(a_1, ..., a_n) = 0 \ \forall f \in I$, entonces $f_i(a_1, ..., a_n) = 0$, luego $V(I) \subset V(f_1, ..., f_s)$.

Visto el primer contenido veamos el segundo. Sea $(a_1, ..., a_n) \in V(f_1, ..., f_s)$ y sea $f \in I$. Como $I = \langle f_1, ..., f_s\rangle$, podemos poner f como

$$f = \sum_{i=1}^{s} h_i f_i,$$

con $h_i \in \mathbb{K}[x_1, ..., x_n]$. Entonces:

$$f(a_1, ..., a_n) = \sum_{i=1}^{s} h_i(a_1, ..., a_n) f_i(a_1, ..., a_n) = \sum_{i=1}^{s} h_i(a_1, ..., a_n) \cdot 0 = 0$$

Por lo tanto tenemos que $V(f_1, ..., f_s) \subset V(I)$ y por doble contenido tenemos la igualdad.

□

En el siguiente resultado se enuncian las propiedades especiales que cumplen los restos al dividir polinomios entre una base de Groebner.

Proposición 6. Sea $G = \{g_1, ..., g_s\}$ una base de Groebner del ideal $I \subset \mathbb{K}[x_1, ..., x_n]$ para cierto orden monomial y sea $f \in \mathbb{K}[x_1, ..., x_n]$. Entonces existe un único $r \in \mathbb{K}[x_1, ..., x_n]$ que cumple:

i) Los términos de r no son divisibles por $LT(g_i)$, $i = 1, ..., s$.

ii) Existe $g \in I$ tal que $f = g + r$.

Demostración. Por el teorema 1.3 (algoritmo de la división), tenemos que $f = a_1 g_1 + \cdots a_s g_s + r$, donde r satisface la primera afirmación de la proposición. Se sigue que satisface también la segunda si llamamos $g = a_1 g_1 + \cdots a_s g_s$.

Para probar la unicidad, supongamos que tenemos dos f compuestos de diferente manera, $f = g_1 + r_1 = g_2 + r_2$, satisfaciendo las afirmaciones de la proposición. Entonces $r_1 - r_2 = g_2 - g_1 \in I$. Así que si $r_2 \neq r_1$, tenemos que $LT(r_1 - r_2) \in \langle LT(I)\rangle = \langle LT(g_1), ..., LT(g_s)\rangle$.

En particular, r es el resto de la división de f por G sin importar el orden de los elementos de G al usar el algoritmo de la división. Por la proposición 1, se sigue que $LT(r_1 - r_2)$ es divisible por algún $LT(g_i),\ i = 1,...,s$. Llegamos a contradicción ya que ningun término de r_1, r_2 es divisible por ningún $LT(g_i),\ i = 1,...,s$. Así tenemos que $r_1 = r_2$ y se sigue que $g_1 = g_2$ y esto prueba la unicidad de f.

□

Corolario 3. Sea $G = \{g_1,...,g_s\}$ una base de Groebner de un ideal $I \subset \mathbb{K}[x_1,...,x_n]$ para cierto orden monomial y sea $f \in \mathbb{K}[x_1,...,x_n]$. Entonces $f \in I$ si y solo si el resto de la división de f por G es nulo.

Demostración. La implicación hacia la derecha es trivial ya que si el resto es cero, entonces se tiene que $f \in I$. La otra implicación también es trivial, ya que si $f \in I$, entonces $f = f + 0$ satisface las afirmaciones de la proposición 6. Se sigue que 0 es el resto de la división de f por G.

□

1.4. Algoritmo de Buchberger

El objetivo de esta sección es dar un criterio algorítmico para caracterizar que un sistema de generadores de un ideal sea una base de Groebner. Este criterio permite describir un método alternativo para construir una base de Grobner a partir de un conjunto de generadores de un ideal.

Notación. Fijado un orden monomial denotaremos $\overline{f}^G$ al resto de la división de f entre la s-tupla ordenada $G = (g_1,...,g_s)$. Si G es una base de Groebner de $\langle g_1,...,g_s\rangle$, entonces podemos ver G como un conjunto (no importa el orden de los elementos).

Definición 1.15. Sean $f, g \in \mathbb{K}[x_1,...,x_n]$ no nulos. Si $mgr(f) = \alpha$ y $mgr(g) = \beta$, entonces definimos $\gamma = (\gamma_1,...,\gamma_n)$ con $\gamma_i = \text{máx}(\alpha_i, \beta_i),\ i = 1,...n$. Llamamos a x^γ *mínimo común múltiplo de* $LM(f)$ *y* $LM(g)$, es decir, $x^\gamma = LCM(LM(f), LM(g))$.

Obsérvese que, a diferencia de otros resultados y definiciones anteriores, la definición de mínimo común múltiplo no depende de la elección de un orden monomial.

Definición 1.16. Sean $f, g \in \mathbb{K}[x_1,...,x_n]$ no nulos. El *S-polinomio de* f *y* g se define como:

$$S(f,g) = \frac{x^\gamma}{LT(f)} \cdot f - \frac{x^\gamma}{LT(g)} \cdot g.$$

Teorema 1.6 (Criterio de Buchberger). *Sea I un ideal de polinomios y fijemos un orden monomial. Entonces una base $G = \{g_1,...,g_s\}$ de I es de Groebner para I si y solo si para todo $i \neq j$, el resto de la división de $S(g_i, g_j)$ entre G (en cierto orden) es cero.*

Debido a su extensión, la demostración de este Teorema no consta en el trabajo. La podemos consultar en [1, pág 82].

Teorema 1.7. *Sea $I = \langle f_1,...,f_s\rangle \neq \{0\}$ un ideal de polinomios y fijemos un orden monomial. Entonces una base de Groebner para I se puede construir en un número finito de pasos añadiendo a $G = (f_1,...,f_s)$ el polinomio $0 \neq f_{s+1} = \overline{S(f_i,f_j)}^G,\ \ i \neq j$. Si con la nueva base $G = (f_1,...,f_s,f_{s+1})$ no son todos los $\overline{S(f_i,f_j)}^G = 0\ \ \forall i \neq j$, entonces se reitera el proceso hasta que se cumpla. La base G resultante será la base de Groebner deseada.*

Demostración. La prueba de este teorema es complicada y tiene una notación compleja. A través del llamado teorema de la cadena ascendente que nos asegura teniendo una cadena ascendente de ideales $I_1 \subset I_2 \subset I_3 \subset \cdots$ en $\mathbb{K}[x_1,...,x_n]$, entonces existen un $N \geq 1$ tal que $I_N = I_{N+1} = \cdots$. La demostración se puede seguir en [1, pág 87]. □

Ejemplo 2. Sean $(f_1, f_2) = (x^2y - 1, xy^2 - x)$ dos polinomios en $\mathbb{R}[x,y]$ (con $x >_{lex} y$) que generan $I = \langle f_1, f_2 \rangle$. Vamos a construir una base de Groebner para este ideal.

Vamos a ver que la base dada, $F = \{f_1, f_2\}$ no es de Groebner. Calculamos el mínimo común múltiplo de los monomios directores de f_1 y f_2, $x^\gamma = x^2y^2$, obteniendo así el S-polinomio:

$$S(f_1, f_2) = \frac{x^2y^2}{x^2y} f_1 - \frac{x^2y^2}{xy^2} f_2 = x^2 - y \neq 0,$$

y si aplicamos el algoritmo de la división para dividir $x^2 - y$ entre F obtenemos de resto el propio $x^2 - y$, $\overline{S(f_1, f_2)}^F \neq 0$ en I, por lo que el teorema 1.6 asegura que F no es una base de Groebner.

Visto esto, iniciamos el algoritmo añadiendo a F el polinomio $f_3 = x^2 - y$, quedando $F = \{f_1, f_2, f_3\}$. Comprobemos que los restos de dividir los S-polinomios por F, $\overline{S(f_i, f_j)}^F \ \forall i \neq j$, son nulos en I:

$$\begin{aligned}
&\overline{S(f_1, f_2)}^F = 0, \\
&\overline{S(f_1, f_3)}^F = y^2 - 1 = f_1 - y \cdot f_3 = 0, \\
&\overline{S(f_2, f_3)}^F = y^3 - y = y \cdot f_3 = 0.
\end{aligned}$$

Así, por el Teorema 1.6 (criterio de Buchberger), tenemos que $F = \{f_1, f_2, f_3\}$ es una base de Groebner.

Lema 1.8. *Sea G una base de Groebner para el ideal de polinomios I. Sea $p \in G$ un polinomio tal que $LT(p) \in \langle LT(G \setminus \{p\}) \rangle$. Entonces $G \setminus \{p\}$ también es una base de Groebner para I.*

Demostración. Sabemos que $\langle LT(G \setminus \{p\}) \rangle = \langle LT(I) \rangle$. Si $\langle LT(p) \rangle \in \langle LT(G \setminus \{p\}) \rangle$, entonces tenemos que $LT(G \setminus \{p\}) = LT(G)$ y, por definición, $G \setminus \{p\}$ también es una base de Groebner de I. □

Definición 1.17. Una *base de Groebner minimal* para un ideal de polinomios I es una base de Groebner G que cumple:

i) $LC(p) = 1, \ \forall p \in G$.

ii) Para todo $p \in G, LT(p) \notin \langle LT(G \setminus \{p\}) \rangle$.

Definición 1.18. Una *base de Groebner reducida* para un ideal de polinomios I es una base de Groebner G que cumple:

i) $LC(p) = 1, \ \forall p \in G$.

ii) Para todo $p \in G$, ningún monomio de p está en $\langle LT(G \setminus \{p\}) \rangle$.

Proposición 7. Sea $I \neq \{0\}$ un ideal de polinomios. Entonces, dado un orden monomial, I tiene una única base de Groebner reducida.

Demostración. Sea G una base de Groebner minimal para el ideal I. Decimos que $g \in G$ es un elemento reducido en G siempre y cuando los monomios de g no estén en $\langle LT(G \setminus \{g\})\rangle$. Vamos a modificar G hasta que todos sus elementos sean reducidos.

Notemos que si g es un elemento reducido de G, entonces g también es reducido en cualquier otra base de Groebner minimal de I que contenga a g y tenga los mismos términos directores.

Dado $g \in G$ y sean $g' = \overline{g}^{G\setminus\{g\}}$, $G' = (G \setminus \{g\}) \cup \{g'\}$, entonces G' es una base de Groebner minimal de I. Para verlo, notar que $LT(g') = LT(g)$. Cuando dividimos g entre $G \setminus \{g\}$, $LT(g)$ forma parte del resto ya que no es divisible por los elementos de $LT(G \setminus \{g\})$. Así tenemos que $\langle LT(G')\rangle = \langle LT(G)\rangle$. Como G' obviamente está contenido en I, tenemos que G' es una base de Groebner minimal de I. Además, tenemos que g' es un elemento reducido por construcción.

Ahora, tomamos los elementos de G y aplicamos el algoritmo anterior hasta que todos sean reducidos. La base de Groebner puede cambiar cada vez que hagamos una iteración del algoritmo, pero teniendo en cuenta lo anterior un elemento que ya es reducido sigue siéndolo mientras no se cambien los términos directores. Así, conseguimos nuestra base de Groebner reducida.

Queda ver que esta es única. Suponer que G y $\widetilde{G}$ son dos bases de Groebner reducidas para I. En particular, G y $\widetilde{G}$ son bases minimales y por eso tienen los mismos términos directores: $LT(G) = LT(\widetilde{G})$.

Entonces, dados $g \in G$ y $\tilde{g} \in \widetilde{G}$ tales que $LT(g) = LT(\tilde{g})$, si podemos demostrar que $g = \tilde{g}$, se seguirá que $G = \widetilde{G}$ y estará comprobada la unicidad.

Consideramos $g - \tilde{g} \in I$. Como G es una base de Groebner, se sigue que $\overline{g-\tilde{g}}^{G} = 0$. Pero también tenemos que $LT(g) = LT(\tilde{g})$ y así, los términos directores de cada uno se cancelan en $g - \tilde{g}$ y el resto de términos no son divisibles por los $LT(G) = LT(\widetilde{G})$ ya que G y $\widetilde{G}$ son bases reducidas. Se sigue que $\overline{g-\tilde{g}}^{G} = g - \tilde{g}$, luego $g - \tilde{g} = 0$ y se sigue la unicidad. □

1.5. Aplicaciones de las bases de Groebner

Las bases de Groebner tienen utilidad en problemas como la pertenencia de un elemento a un ideal, encontrar la ecuación implicita de una parametrización,... Nos centramos en la aplicación a la resolución de sistemas de ecuaciones.

Como hemos visto antes, un ideal I puede formar una variedad afín $V(I)$. Sabemos también que una variedad afín $V(f_1,...,f_s) \subset \mathbb{K}^n$ es el conjunto de soluciones del sistema de ecuaciones:

$$\begin{cases} f_1(x_1,...,x_n) = 0 \\ \vdots \\ f_s(x_1,...,x_n) = 0 \end{cases}$$

Así por la Proposición 5 podemos resolver los sistemas de ecuaciones de una manera sencilla, tomando como base del ideal I las ecuaciones y convirtiéndola en una base de Groebner.

Ejemplo 3. Vamos a resolver el siguiente problema aplicando los resultados que hemos visto en este capítulo: un vehículo que transporta un explosivo sale de un punto en dirección noreste en linea recta. Al cabo de z kilómetros explota y dos radares ubicados a 2 kilómetros al este y al oeste del punto de salida detectan la explosión con 6 segundos de diferencia. Calcular el punto exacto donde explota suponiendo que el sonido tarda aproximadamente 3 segundos en recorrer un kilómetro y sabiendo que $z = x\sqrt{y}$, con (x,y) las coordenadas del punto de explosión.

El punto de explosión se encuentra en algún lugar de la circunferencia de radio xy km, es decir, en $x^2+y^2=x^2y$. Como el artefacto explota al noreste del origen, las coordenadas (x,y) del lugar serán las positivas.

Los radares están colocados en $(2,0)$ y $(-2,0)$ respecto al origen de coordenadas. Para determinar la ecuación que describen estos dos focos, vamos a utilizar la condición de que en el foco más lejano se tardó 6 segundos más en detectar la explosión. Sea $\overrightarrow{u}=(x-2,y)$ y $\overrightarrow{v}=(x+2,y)$ los vectores que unen los radares con el punto de explosión, entonces la distancia que recorre el sonido a cada radar se diferencia en 2 km, debido a la velocidad del sonido. Por lo tanto, los módulos de los vectores anteriores deben diferenciarse en 2 unidades, resultando la ecuación así: $\sqrt{(x-2)^2+y^2}=\sqrt{(x+2)^2+y^2}-2$.

Desarrollando la ecuación irracional se obtiene la siguiente ecuación de una hipérbola: $3x^2-y^2=3$.

Así ya nos encontramos en situación de resolver un sistema de ecuaciones:

$$\begin{cases} x^2+y^2=x^2y \\ 3x^2-y^2=3 \end{cases}$$

Vamos a resolver el sistema en $\mathbb{R}$ usando bases de Groebner (utilizando el orden lexicográfico). Llamamos $g_1=x^2+y^2-x^2y$ y $g_2=3x^2-y^2-3$ y serán base de un ideal I. Con el algoritmo de Buchberger obtenemos que:

$$\begin{aligned} g_1 &= x^2+y^2-x^2y \\ g_2 &= 3x^2-y^2-3 \\ g_3 &= x^2-\frac{1}{3}y^2-1 \\ g_4 &= y^3-4y^2+3y-3 \end{aligned}$$

es una base de Groebner de I y, de hecho, $\{g_3,g_4\}$ es la base reducida. Así tenemos el siguiente sistema de ecuaciones:

$$\begin{cases} x^2-\frac{1}{3}y^2-1=0 \\ y^3-4y^2+3y-3=0 \end{cases}$$

La última ecuación es fácil de resolver con métodos númericos, tiene una única solución real, y es $y=3{,}3744...$ Asi, sustituyendo en la otra ecuación obtenemos que $x=2{,}1898..$

Encontrar una base de Groebner para un ideal respecto al orden lexicográfico simplifica nuestro sistema de ecuaciones de manera sustancial. Por ejemplo, si el sistema es finito con este procedimiento conseguimos obtener una ecuación con una única incógnita. Así, utilizando técnicas de resolución de ecuaciones de una variable conseguimos soluciones para el sistema. Este procedimiento se conoce como eliminación de variables.

Como hemos visto antes, podemos ver los sistemas de ecuaciones como variedades afines. Luego por la Proposición 5 tenemos que las variedades afines están determinadas por ideales. Esto significa que, una vez encontradas las soluciones buscando una base de Groebner para el ideal I, tenemos que estas soluciones son todas las que existen.

Resumiendo, hemos encontrado un procedimiento efectivo con el cual no solo obtenemos alguna solución sino que obtenemos todas las existentes para ese sistema de ecuaciones.

Visto todos estos conceptos y procedimientos en el siguiente capítulo buscaremos solución al problema de coloreado de grafos, consiguiendo formar un sistema de ecuaciones que resolveremos con este último procedimiento.

Capítulo 2

Teoría y coloreado de grafos

En este capítulo vamos a definir el concepto de grafo y a dar 2 métodos para obtener sistemas de ecuaciones que permitan, resolviéndolos a través de bases de Groebner, dar respuesta al problema de coloreado de un grafo con un número dado de colores.

Aunque hay definiciones más generales de grafo. Nos vamos a centrar en lo que se conoce como grafo regular.

Definición 2.1. Un *grafo regular* G es un par ordenado (V,E) con:

- V es un conjunto finito. Sus elementos se denominan *vértices* o *nodos* de G.
- E es un subconjunto finito de $\binom{V}{2}$. Sus elementos se denominan *aristas*.

Si dos vértices pertenecen a una arista decimos que *están conectados* o que *son adyacentes*.

En particular estos grafos no tienen bucles (aristas que unen un mismo vértice) ni dobles aristas conectando dos vértices. Las aristas no tienen dirección, y por tanto a veces se les denomina también *grafo regular no dirigido*.

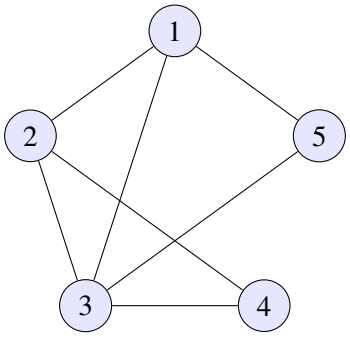
Nos interesa que los grafos además sean conexos. Para definir conexidad, es necesario definir lo siguiente:

Definición 2.2. Un *camino* es una lista $v_0, v_1, ..., v_k$ de vértices de manera que $\{v_i, v_{i+1}\} \in E$ para $0 \leq i < k$. En este caso se dice que existe un camino que conecta v_0 y v_k.

Ahora si que estamos en situación de definir cuándo un grafo se dice conexo:

Definición 2.3. Un grafo G es *conexo* si para todo par $u, v \in V$ existe un camino que conecta u y v.

Ejemplo 4. Grafo no dirigido de 5 vértices $V = \{1,2,3,4,5\}$ y 7 aristas $E = \{\{1,2\},\{1,3\},$ $\{1,4\},\{2,3\},\{3,4\},\{3,5\},\{4,5\}\}$:



2.1. Colorabilidad

Antes de ponernos a hablar de coloración de grafos, enunciamos la siguiente definición:

Definición 2.4. Un grafo G es *bipartito* si V es unión disjunta de dos conjuntos independientes llamados particiones de G.

La definición anterior la podríamos generalizar a *grafo k-partito*, es decir, si V es unión disjunta de k conjuntos independientes. Esta noción nos va a servir para enunciar las siguientes definiciones.

Definición 2.5. Una *k-coloración* de un grafo G es una asignación $f : V \to C$ donde C es un conjunto de k elementos a los que llamamos *colores*. Como a los vértices les asignamos colores, los vértices de un color forman una *clase de color*.

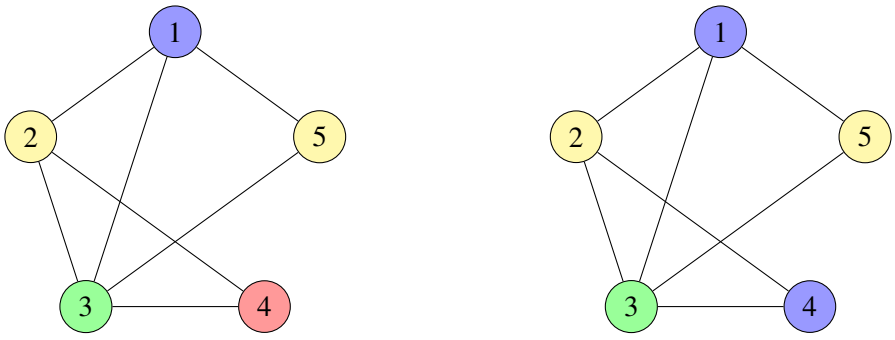
Definición 2.6. Dado un grafo $G = (V,E)$, decimos que una *k-coloración* $f : V \to C$ es *propia* si los vértices adyacentes tienen diferente color asignado, es decir, si $\{u,v\} \in E$, entonces $f(u) \neq f(v)$. De esta forma, un grafo es *k-coloreable* si tiene una k-coloración propia.

A partir de ahora, cuando hablemos de si un grafo se puede o no colorear con k colores estaremos diciendo que admite una k-coloración propia. Observar que una coloración f no se pide que sea suprayectiva, entonces todo grafo k-coloreable es automáticamente $(k+1)$-coloreable usando la misma coloración.

Definición 2.7. El *número cromático* de un grafo G, $\chi(G)$, es el menor k tal que G es k-coloreable. Así, decimos que un grafo es *k-cromático* si $\chi(G) = k$, es decir, si es k-coloreable pero no $(k-1)$-coloreable.

Veamos un ejemplo para aclarar las definiciones anteriores:

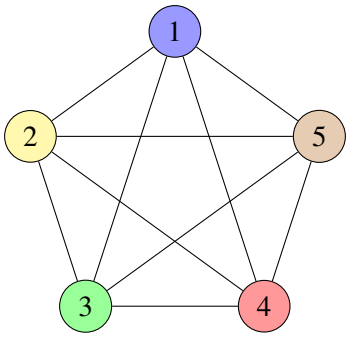
Ejemplo 5. El grafo del ejemplo 4 es 4-coloreable, ya que se puede colorear de la primera forma. Pero el número cromático de este no es 4, ya que se puede colorear también con 3 colores como se ve en el segundo dibujo:



Concluyendo, todo grafo k-cromático es k-coloreable pero no todo grafo k-coloreable es k-cromático.

Definición 2.8. El *clique* de un grafo G es un conjunto de vértives $C \subset V$ tal que todo par de vértices distintos son adyacentes. Al clique que implique el mayor número de vértices en un grafo lo denotaremos por $\omega(G)$.

Ejemplo 6. Un clique de 5 vértices, también llamado grafo completo de 5 vértices, tiene $\binom{5}{2} = 10$ aristas y su representación gráfica es la siguiente:



Además, es fácil ver que si un clique consta de n vértices, su número cromático será n ya que todos los vértices son adyacentes entre sí. Esta definición nos ayuda a marcar una cota inferior al número cromático de cualquier grafo.

Proposición 8. Para todo grafo G tenemos que el número cromático es mayor o igual al número de vértices del clique máximo de ese grafo, es decir, $\chi(G) \geq \omega(G)$.

Demostración. Es consecuencia de que el número cromático de un clique coincide con su número de vértices. □

Para acabar con la parte de coloración de grafos, es conveniente ver unos sencillos ejemplos de coloraciones:

Ejemplo 7. Dependiendo del número de colores, se pueden dar estos casos extremos:

1. Si el grafo tiene n vértices, entonces es n-coloreable ya que existe una coloración con un color diferente para cada vértice.
2. Un grafo es 1-coloreable si y sólo si no contiene aristas.
3. Un grafo que no contiene ciclos cerrados, es decir, un árbol es 2-coloreable. En cambio el recíproco no es cierto, por ejemplo un cuadrado, pero no es un árbol, es 2-coloreable pero un pentágono no es 2-coloreable. Precisamente esta es su caracterización: un grafo es 2-coloreable si y solo si no contiene ningún ciclo con un número impar de vértices.

2.2. De grafos a sistemas de ecuaciones

Para grafos sencillos como los de los ejemplos anteriores, el problema no es muy complicado. Pero en cuanto aumentamos el número de nodos y aristas, el problema es bastante más difícil. Para resolverlo utilizaremos sistemas de ecuaciones polinómicas. A continuación se dan dos formas de conseguir los sistemas de ecuaciones con un ejemplo práctico para cada uno.

A partir de aquí supondremos que G es un grafo de n nodos y nuestro problema consiste en ver si este grafo es k-coloreable.

Primera forma: los nodos van a estar representados por los números naturales, $V = \{1,...,n\}$ y los colores por $C = \{1,...,k\}$. Se definen los sistemas sobre un anillo de polinomios con coeficientes en el cuerpo finito de 2 elementos $\mathbb{Z}_2[\{X_{ij} \mid i \in V,\ j \in C\}]$.

Definimos la *función de colorear* tal que a un nodo i le asigna un único color j, es decir:

$$\begin{array}{rccc} f: & V & \longrightarrow & C \\ & i & \longmapsto & f(i) = j \end{array}$$

Así las incógnitas de nuestro sistema X_{ij} con $i \in V$ y $j \in C$ tales que:

$$X_{ij} = \begin{cases} 0 & \text{si } f(i) = j, \\ 1 & \text{si } f(i) \neq j. \end{cases}$$

Una vez definidas las incógnitas, vamos a establecer las condiciones que deben cumplir estas para así obtener nuestro sistema:

- Cada nodo debe tener al menos un color: que un nodo tenga un color significa que para todo $i \in V$ debe haber al menos un j tal que $X_{ij} = 0$. Luego el producto variando en j y fijando i tiene que ser 0 para que esto suceda. Así nuestro primer conjunto de ecuaciones es:

$$\prod_{j=1}^{k} X_{ij} = 0, \ \forall i \in V.$$

- Cada nodo debe tener como mucho un color: ahora fijamos el subíndice de los nodos, es decir, fijamos un nodo i. Como estamos en $\mathbb{Z}_2$ si tomamos $j,l \in C$ con $j \neq l$, para expresar que el nodo no puede tener dos colores asignado a la vez, nos vale con la ecuación $(X_{ij}+1)(X_{il}+1) = 0$. Si esto lo extendemos a todos los colores y nodos, nos queda:

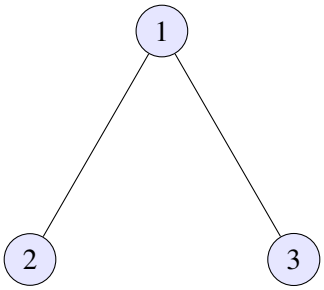
$$(X_{ij}+1)(X_{il}+1) = 0 \ \forall j \neq l, \ j,l \in C.$$

- Los nodos que estén unidos deben tener colores distintos: fijamos un color j y dos nodos i y m que están unidos por una arista, es decir, $[i,m] \in E$. Para que estos dos nodos no estén coloreados por j se tiene que cumplir la ecuación $(X_{ij}+1)(X_{mj}+1) = 0$. Extendiendo esto a todo el grafo obtenemos:

$$(X_{ij}+1)(X_{mj}+1) = 0 \ \forall i \neq m, \ \{i,m\} \in E, \forall j \in C.$$

Por lo tanto, el sistema lo compondrán las ecuaciones anteriores. El número de ecuaciones de los sistemas son $n + n\binom{k}{2} + k|E|$.

Ejemplo 8. Tomamos $G = (V,E)$ con $V = \{1,2,3\}$ y $E = \{\{1,2\},\{1,3\}\}$:



El problema del 2-coloreado produce el siguiente sistema polinómico:

$$\begin{cases} X_{11}X_{12} = 0 \\ X_{21}X_{22} = 0 \\ X_{31}X_{32} = 0 \\ (X_{11}+1)(X_{12}+1) = 0 \\ (X_{21}+1)(X_{22}+1) = 0 \\ (X_{31}+1)(X_{32}+1) = 0 \\ (X_{11}+1)(X_{21}+1) = 0 \\ (X_{12}+1)(X_{22}+1) = 0 \\ (X_{11}+1)(X_{31}+1) = 0 \\ (X_{12}+1)(X_{32}+1) = 0 \end{cases}$$

Para resolver el siguiente sistema aplicaremos lo visto en el último apartado del capítulo 1, es decir, buscamos la base de Groebner del sistema para que sea más sencillo de resolver. Así el sistema queda:

$$\begin{cases} X_{11}+X_{32}=0 \\ X_{12}+X_{32}+1=0 \\ X_{21}+X_{32}+1=0 \\ X_{22}+X_{32}=0 \\ X_{31}+X_{32}+1=0 \\ X_{32}^2+X_{32}=0 \end{cases}$$

Desechando ecuaciones redundantes que provienen de las propiedades del cuerpo se obtienen todas las posibles coloraciones. Ya sabíamos que existían dado que el grafo es un árbol (Ejemplo 7.3). Así $X_{11}=X_{32}=X_{22}=1,\ \ X_{12}=X_{21}=X_{31}=0$ es una solución, pero no es la única ya que $X_{11}=X_{32}=X_{22}=0,\ \ X_{12}=X_{21}=X_{31}=1$ también es otra solución.

Segunda forma: consideramos un grafo de n nodos. Asignamos las variables X_i a cada nodo, $i=1,...,n$. Si tenemos k colores, consideramos el cuerpo finito de número de elementos k o el siguiente primo a este y lo llamamos $\mathbb{F}=\{0,1,...,l-1\}$ con $l=k$ si k es primo o el siguiente primo a k. Así el sistema de polinomios estará definido en $\mathbb{F}[X_1,...,X_n]$.

Volvemos a usar la *función de colorear* que hemos usado antes:

$$\begin{array}{rccc} f: & V & \longrightarrow & C \\ & i & \longmapsto & f(i)=j \end{array}$$

Las incógnitas toman los siguientes valores:

$$X_i=\begin{cases} j & \text{si } f(i)=j, \\ 0 & \text{si } f(i)\neq j. \end{cases}$$

Definimos las funciones:

$$g(x)=\prod_{i=1}^{k}(x-i), \quad h(x,y)=\frac{f(x)-f(y)}{x-y}.$$

Ahora vamos a establecer condiciones para conseguir nuestro sistema:

- Cada nodo tiene que tener asignado un color: por la definicón de la función g, si $g(X_i)=0$ significa que uno de los factores es 0, es decir, ese es el color asignado a ese nodo. Luego nuestras ecuaciones son:
$$g(X_i)=0,\ \ \forall i\in\{1,...,n\}.$$
- Cada par de nodos conectados por una arista tienen que tener asignados colores diferentes: por cómo hemos definido las aplicaciones g y h, si $h(X_i,X_j)=0$ significa que X_i y X_j tienen colores asignados ya que $g(X_i)-g(X_j)=0$ y son colores diferentes ya que $X_i-X_j\neq 0$ ya que sino habría indeterminación. Así, nuestras ecuaciones son:
$$h(X_i,X_j)=0,\ \text{si } \{i,j\}\in E.$$

Así conseguiríamos nuestro sistema de ecuaciones. Notar que este sistema consta de $n+|E|$ ecuaciones.

Ejemplo 9. Considerar el grafo del ejemplo 8. Como nuestro número de colores es 2, este es primo. Luego estamos trabajando en $\mathbb{Z}_2[X_1, X_2, X_3]$ y los colores vienen representados por los elementos de este $\{0, 1\}$. Nuestro sistema de ecuaciones es:

$$\begin{cases} (X_1-0)(X_1-1)=0 \\ (X_2-0)(X_2-1)=0 \\ (X_3-0)(X_3-1)=0 \\ \\ \dfrac{(X_1-0)(X_1-1)-(X_2-0)(X_2-1)}{X_1-X_2} = X_1+X_2+1=0 \\ \\ \dfrac{(X_1-0)(X_1-1)-(X_3-0)(X_3-1)}{X_1-X_3} = X_1+X_3+1=0 \end{cases}$$

Como antes, para resolver el sistema, aplicaremos lo visto en el último apartado del capítulo 1, es decir, buscamos la base de Groebner del sistema para que sea más sencillo de resolver. Así el sistema queda:

$$\begin{cases} X_1+X_3+1=0 \\ X_2+X_3=0 \\ X_3(X_3+1)=0 \end{cases}$$

Luego las posibles soluciones son: $X_3 = X_2 = 0, X_1 = 1$ o $X_3 = X_2 = 1, X_1 = 0$.

Para la resolución del problema, hemos programado en Sage las dos maneras de obtener los sistemas de ecuaciones de los grafos así como la solución. El código del programa lo encontramos en el Anexo 3.3.

Con toda esta información ya estamos en disposición de analizar la complejidad del problema, como vamos a hacer en el Capítulo 3.

Capítulo 3

Bases de Groebner y grafos

En este capítulo vamos a estudiar la complejidad efectiva de resolución del problema de coloreado de grafos con el fin de ver si es posible utilizarlo como base de un sistema criptográfico. Estableciendo una serie de ideas y comparando los tiempos de ejecución de la resolución del problema en SageMath [4] podremos obtener unos gráficos que representen el tiempo que tarda el programa en resolverlo en función de ciertos parámetros. Para calcular estos tiempos, nos vamos a basar en el código del Anexo3.3.

3.1. Reducción de parámetros

El objetivo por tanto es representar los tiempos de ejecución en función de n el número de nodos, de ℓ el número de aristas y de k el número de colores. Para poder representar algo visible reduciremos el número de parámetros relevantes a dos y acotaremos el rango de estos.

Los parámetros a establecer son el número de vértices, el número de aristas y el número de colores. Nuestra intención es dibujar un gráfico en el que podamos comparar fácilmente los tiempos de ejecución según los 3 parámetros mencionados anteriormente. Dado que el 2-coloreado es relativamente sencillo de decidir (Ejemplo 7. 3), consideraremos el problema del 3-coloreado dado que ya genera sistemas de complejidad suficientemente grande.

Una vez fijado el número de colores establecemos cotas para el número de vértices. Estas van a comprender desde un número en el que el problema sea suficientemente complejo hasta otro que tolere Sage para hacer los cálculos. Este número lo hemos estimado heurísticamente entre 8 y 15 vértices. Por debajo de 8 nos encontramos con grafos pequeños y más simples y por encima de 15 las limitaciones computacionales para obtener datos con el programa son considerables.

En cuanto al número de aristas, en principio partimos del intervalo $[n-1, \binom{n}{2}]$, ya que por debajo de $n-1$ aristas el grafo no es conexo y por encima de $\binom{n}{2}$ el grafo no es regular. Este intervalo se puede mejorar del siguiente modo. Podemos dar una cota inferior basándonos en el número mínimo de aristas que hacen falta para que el grafo sea 3-coloreable y no sea 2-coloreable. Para ello hace falta que no sea un árbol, en particular, el número de aristas ha de ser de al menos tantas como el número de vértices n.

En resumen, fijando el número de colores en $k = 3$, las cotas para el resto de parámetros son:

- El número de vértices, n, varía entre 8 y 15.
- El número de aristas, ℓ, varía entre n y $\binom{n}{2}$.

Así, con Sage, podemos dibujar gráficos del coste computacional del problema en función del número de aristas y vértices fijando el número de colores del problema.

3.2. Tiempos de ejecución según parámetros

En el capítulo anterior, deducimos dos maneras de obtener el sistema de ecuaciones del grafo. Ahora vamos a utilizar solo la segunda forma ya que, comparándolas, la primera es demasiado lenta debido al elevado número de incógnitas y ecuaciones que genera.

Es interesante ver qué tanto por ciento de los grafos son 3-coloreables según su número de vértices y aristas. Para ello, hemos realizado en SageMath un recuento de grafos 3-coloreables en una muestra de 100 para cada número de vértices y aristas, utilizando la función coloreado del código del Anexo 3.3. Así, obtenemos el siguiente gráfico:

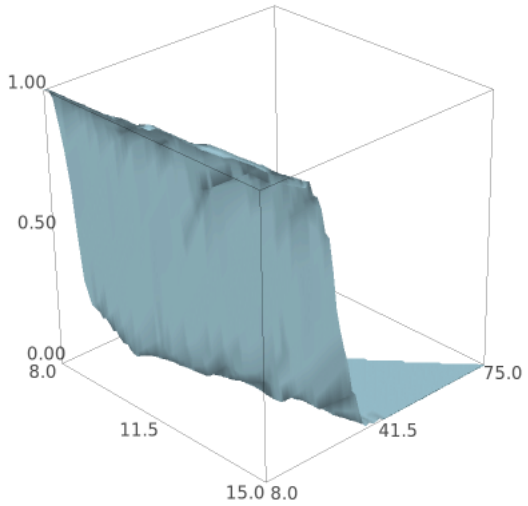


Figura 3.1: Probabilidad de grafos que son 3-coloreables según el número de vértices y aristas.

Para entender la Figura 3.1 observemos que en las coordenadas horizontales se describen el número de vértices y de aristas. La coordenada vertical indica la probabilidad de grafos aleatorios 3-coloreables con dicho número de vértices y aristas.

La tendencia general, independientemente del número de vértices, es que cuanto mayor es el número de aristas menos probable es que el grafo sea 3-coloreable y además la probabilidad se reduce rápidamente.

Veamos ahora el gráfico de tiempos de resolución del problema de 3-coloreado según el número de vértices y aristas de los grafos que sí son 3-coloreables. Para obtenerlo, hemos medido el tiempo que le cuesta a Sage resolver 100 problemas aleatorios por cada número de vértices y aristas, basandonos en el código de la función coloreado incluida en el Anexo 3.3.

Para entender la Figura 3.2 observemos que en las coordenadas horizontales se describen el número de vértices y de aristas mientras que la coordenada vertical indica el tiempo de resolución del problema de 3-coloreado de un grafo aleatorio según el número de vértices y aristas.

Lo más relevante de la gráfica es que el tiempo de resolución crece de forma exponencial conforme subimos el número de vértices. Además, fijando el número de estos, se observa que la gráfica se asemeja a una campana desviada a la izquierda, dejando los mayores valores de coste computacional en el primer tercio del número de aristas.

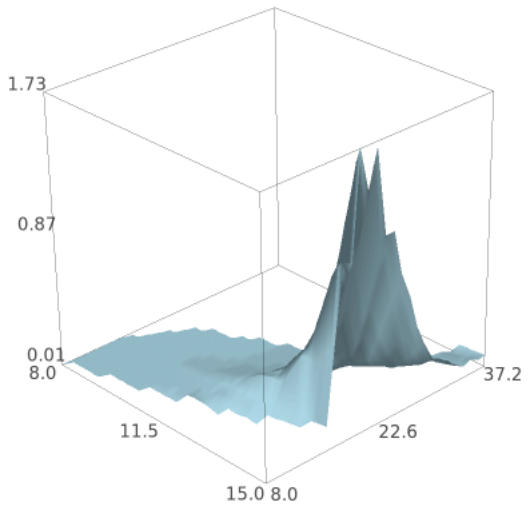


Figura 3.2: Tiempos de resolución del problema de 3-coloreado.

En resumen, la complejidad del problema crece con el número de vértices. En cuanto al número de aristas, esta es mayor cerca de la tercera parte del valor máximo de aristas. Luego un valor orientativo para encontrar grafos de mayor complejidad según aristas es: $\left[\frac{\binom{n}{2}}{3}\right]$.

3.3. Conclusiones y aplicación en criptografía

El problema de la factorización de números es muy sencillo cuando hablamos de números pequeños, pero conforme escogemos números relativamente grandes nos encontramos con que es un problema prácticamente imposible y muy laborioso. Esto mismo ocurre con el problema del coloreado de grafos.

Como hemos visto, un programa informático como Sage no es capaz de obtener datos del 3-coloreado de un grafo de más de 15 nodos. También hemos visto en la Figura 3.2 el crecimiento exponencial del timepo de resolución que se produce conforme aumentamos el número de vértices. Esto nos hace ver lo difícil que sería resolver este problema con una enorme cantidad de nodos y más importante aun el tiempo que costaría. Además, si elegimos un número concreto de aristas el tiempo de resolución se eleva todavía más.

Lo que hace difícil nuestro problema es la obtención de bases de Groebner para sistemas de polinomios con coeficientes en cuerpos finitos. Es decir, hemos conseguido transformar el problema de coloreado de grafos en una obtención de una de estas bases. Nuestro objetivo ha sido desde el principio resolver un problema dificíl a través de dichas bases y analizar su dificultad. Todo esto nos lleva a hablar de criptografía.

La criptografía engloba el conjunto de técnicas utilizadas para cifrar mensajes de tal manera que solo consigan descifrarlo el destinatario de estos. La criptografía se nutre de problemas difíciles de resolver y cada vez son más necesarios por la evolución de la informática e Internet. Por ello, nuestro problema resulta interesante para generar un sistema criptográfico.

Los criptosistemas de Polly Craker son de clave pública y se basan en el álgebra multivariante, es decir, trabaja con polinomios en $\mathbb{K}[x_1,...,x_n]$ siendo $\mathbb{K}$ un cuerpo finito y n el número de variables

implicadas.

La clave pública vendrá dada por un conjunto de polinomios G que serán base de un ideal I y su clave privada serán todos los α tales que $\alpha \in V(F)$. Así, se puede cifrar el mensaje y descifrar a través de las claves pública y privada.

Visto esto es fácil pensar en un criptosistema que se base en nuestro problema de k-coloreado de grafos. La base del ideal I la van a formar las ecuaciones que obtenemos del grafo y los colores. Así, nuestro criptosistema tiene como clave pública el grafo y como clave privada un coloreado del grafo.

En resumen, a través de las bases de Groebner y sus resultados hemos conseguido resolver el problema de coloreado de grafos. Al ser un problema con una dificultad notable variando una serie de parámetros, hemos decidido analizar gráficamente su dificultad en el caso de 3 colores utilizando en SageMath [4] el código recogido en el Anexo 3.3. Después de ver el crecimiento exponencial que sufre el tiempo de resolución variando el número de aristas y aumentando el número de nodos, acabamos hablando de una posible aplicación a la criptografía.

Bibliografía

[1] DAVID COX, JOHN LITTLE, DONAL O'SHEA, *Ideals, Varieties and Algorithms, Colección Springer, Tercera Edición, 2007*.

[2] SARAH BENNETT, *Aplications of Gröbner Bases*, https://math.uchicago.edu/~may/VIGRE/VIGRE2008/REUPapers/Bennett.pdf.

[3] ANGELA M. HENNESSY, *Gröbner Bases With Applications In Graph Theory*, https://drum.lib.umd.edu/bitstream/handle/1903/4203/umi-umd-4013.pdf?sequence=1&isAllowed=y.

[4] SAGEMATH, *The Sage Mathematics Software System (Version v8.6)*, https://www.sagemath.org.

[5] MASSIMILIANO SALA, TEO MORA, LUDOVIC PERRET, SHOJIRO SAKATA, CARLO TRAVERSO, *Gröbner Bases, Coding, and Cryptography, Colección Springer, 2009*.

Anexo

A través del compilador matemático Sage hemos generado funciones que permitan obtener sistemas de ecuaciones a partir de grafos de las dos maneras explicadas anteriormente. Para ello fue necesario introducir la siguiente función que cuenta el número de vértices de un grafo cualquiera ya que la predeterminada por Sage solo cuenta los que tienen vértices adyacentes.

```
def numverts(gr):
    ar=gr.get_vertices()
    aux=max(ar)
    return aux+1
```

A continuación se muestra el código de la función que a través de un grafo y el número de colores obtiene el sistema de ecuaciones que hemos llamado antes primera forma:

```
def eqn1(gr,k):
    aux=[]
    n=numverts(gr)
    cifras=ceil(log(k+1,10))
    xs=list(var(['X_%d'% (10^cifras*i+j) for i in range(n) for j in range(k)]))
    S = PolynomialRing(GF(2),n*k,xs,order='lex')
    xs=[S(_) for _ in xs]
    Xs=[[xs[k*i+j] for j in range(k)] for i in range(n)]
    aux=aux+[prod([Xs[i][j] for j in range(k)]) for i in range(n)]
    for i in range(n):
        for j in range(k):
            for l in range(j):
                aux=aux+[(Xs[i][j]+1)*(Xs[i][l]+1)]
    M=gr.adjacency_matrix()
    for j in range(k):
        for i in range(n):
            for l in range(i):
                if M[i][l]==1:
                    aux=aux+[(Xs[i][j]+1)*(Xs[l][j]+1)]
    return aux
```

Se puede observar que la estructura es primero generar una lista de variables acorde con el número de vértices y colores y después se obtienen las ecuaciones imponiendo las condiciones correspondientes. De la misma forma se obtiene el código del segundo método:

```
def eqn2(gr,k):
    n=numverts(gr)
    xs=list(var(['X_%d'%i for i in range(n)]))
```

```
    if k.is_prime()=='True':
        F=GF(k)
    else:
        F=GF(k.next_prime())
    S=PolynomialRing(F,n,xs,order='lex')
    l=F.list()
    l=[S(_) for _ in l]
    xs=[S(_) for _ in xs]
    colores=[l[i] for i in range(k)]
    eqf=[]
    for j in xs:
        eqf=eqf+[S(prod([(j-i) for i in colores]))]
    eqg=[]
    edges_aux=[[_[0],_[1]] for _ in gr.edges()]
    for j in edges_aux:
        aux=[S(prod([(xs[j[0]]-i) for i in colores])),S(prod([(xs[j[1]]-i)
        for i in colores]))]
        eqg=eqg+[S((aux[0]-aux[1])/(xs[j[0]]-xs[j[1]]))]
    return eqf+eqg
```

Así ya estamos en disposición de crear una función capaz de decirnos si un grafo es k-coloreable y obtener su base de Groebner para resolverlo de una manera sencilla. Además del grafo y el número de colores, la función permite elegir el método de obtención de ecuaciones, viniendo como predeterminado el método 1:

```
def coloreado(gr,k,metodo=1):
    n=numverts(gr)
    if metodo==1:
        cifras=ceil(log(k+1,10))
        xs=list(var(['X_%d'% (10^cifras*i+j) for i in range(n) for j
        in range(k)]))
        S = PolynomialRing(GF(2),n*k,xs,order='lex')
        I=S.ideal(eqn1(gr,k))
    elif metodo==2:
        xs=list(var(['X_%d'%i for i in range(n)]))
        if k.is_prime()=='True':
            F=GF(k)
        else:
            F=GF(k.next_prime())
        S=PolynomialRing(F,n,xs,order='lex')
        I=S.ideal(eqn2(gr,k))
    else:
        return "metodo no existente"
    gb=I.groebner_basis()
    if gb==[1]:
        print("el grafo no es ",k," coloreable")
        return 1
    else:
        print("el grafo si es ",k," coloreable")
        return 0, gb
```