

# Matemáticas prácticas

---

Fernando Chamizo Lorente  
(UAM-ICMAT)

Universidad de Murcia

23 de febrero de 2017

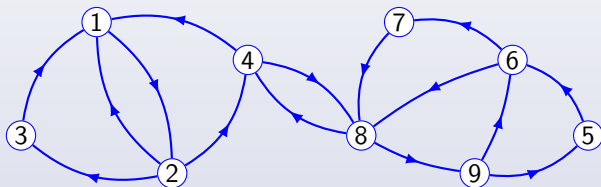
*“ I’m a mathematician, basically. What I do is look around for problems where I can find useful applications for mathematics. All I do, really, is the math, and other people have the ideas. ”*

F. Dyson

## El algoritmo *PageRank*™

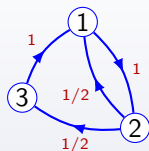
## El modelo

Imaginemos una red de comunicaciones formada por nodos unidos mediante “cables” con una dirección en la que fluye la información: Las páginas web que componen internet con sus enlaces.

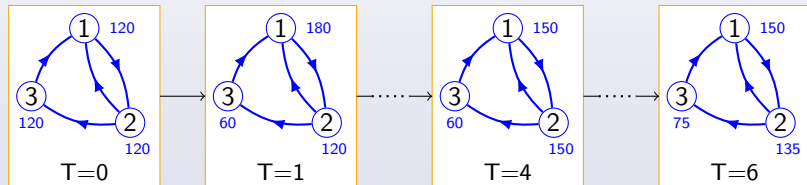


Un modelo natural simple consiste en suponer que un visitante de una página web escoge al azar uno de los enlaces que se le ofrecen para continuar navegando.

Supongamos que en cada unidad de tiempo, los visitantes se reparten equitativamente entre los diferentes enlaces.



Asignando un número de visitantes a cada página,

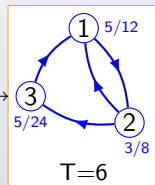
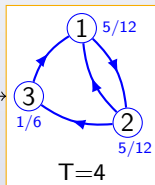
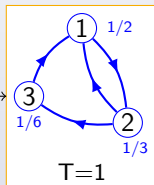
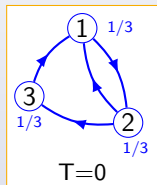
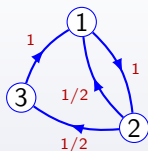


$(144, 144, 72)$

*distribución límite*

Supongamos que en cada unidad de tiempo, los visitantes se reparten equitativamente entre los diferentes enlaces.

Asignando un número de visitantes a cada página,

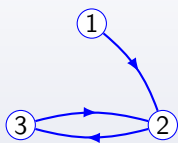


$(2/5, 2/5, 1/5)$

*distribución límite*

¿Es éste un sistema adecuado para ordenar la importancia de las páginas web?

Si uno de los vértices no tiene ninguna importancia, por ejemplo



paso 0  $\rightarrow$   $(1/3, 1/3, 1/3)$

paso 1  $\rightarrow$   $(0, 2/3, 1/3)$

paso 2  $\rightarrow$   $(0, 1/3, 2/3)$

paso 3  $\rightarrow$   $(0, 2/3, 1/3)$

paso 4  $\rightarrow$   $(0, 1/3, 2/3)$

la probabilidad inicial asignada al vértice irrelevante se va transmitiendo periódicamente y la simulación no converge.

Si la detenemos en un paso impar nos parecerá que ② es el doble de importante que ①, mientras que si la detenemos en un paso par, sacaremos la conclusión contraria.

¿Por qué ocurre esto? ¿Cómo remediar este tipo de situaciones conservando el espíritu del modelo?

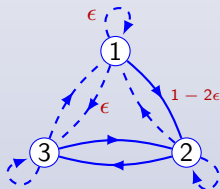
# La teoría

## Cadenas de Markov

- Irreducible  $\rightarrow$  Todas las partes de la red son alcanzables.  
 Regular  $\rightarrow$  Irreducible especificando un número de pasos.

*Para cadenas de Markov finitas regulares, la sucesión  $\{(\pi_n)\}_{n=0}^{\infty}$  de vectores de probabilidades en el paso  $n$  converge a la única distribución estacionaria. En particular, el límite no depende de la distribución inicial.*

Añadiendo enlaces imaginarios “débiles” de probabilidad  $\epsilon$  que unen los nodos no conectados, la cadena se vuelve regular.



En el ejemplo, la distribución límite es

$$\left( \epsilon, \frac{1-2\epsilon}{2-3\epsilon}, \frac{1-3\epsilon+3\epsilon^2}{2-3\epsilon} \right)$$

que tiende a  $(0, 1/2, 1/2)$  si  $\epsilon \rightarrow 0^+$ .



## La práctica

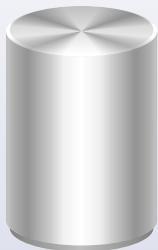
El algoritmo fue patentado en 2001 por uno de los fundadores de la empresa Google.

- Esta empresa afirma que su buscador indexa  $N > 10^{12}$  páginas.  
↪ Teóricamente esto llevaría a considerar más de  $10^{24}$  enlaces.
- Como las probabilidades siempre suman uno, basta hacer la simulación con los enlaces reales y multiplicar en cada paso los resultados por  $1 - N\epsilon$  y sumarles  $\epsilon$ .  
↪ Según los informes, se toma  $\epsilon = 0,15/N$ , se realizan entre 50 y 100 iteraciones en la simulación y se llevan a cabo actualizaciones más o menos mensuales.

# TAC: tomografía axial computarizada

## El modelo

Tenemos una barra metálica maciza homogénea y queremos saber si tiene alguna imperfección en su fabricación.



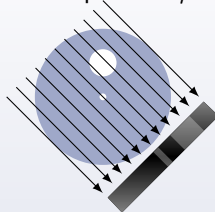
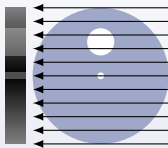
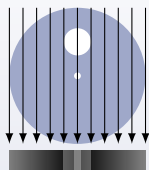
Con una máquina de rayos X, basta tomar una radiografía.

Cualquier mancha que veamos es señal de una imperfección.

¿Podemos ser más precisos con más radiografías y averiguar su situación y forma o ver variaciones suaves de la densidad?

La cuestión es equivalente a saber si podemos deducir la estructura de un cuerpo translúcido a partir de sus sombras.

Consideramos una región no homogénea de densidad superficial  $\rho$

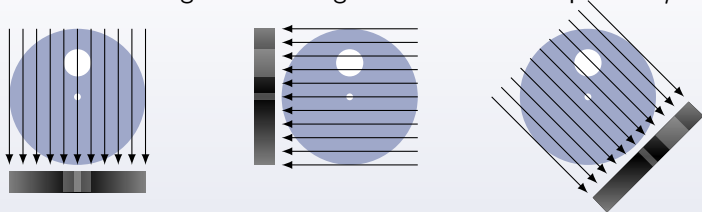


Cada rayo sufre una atenuación medible que depende de

$$P_{\alpha}(t) = \int_{r_{\alpha,t}} \rho \quad \text{con } r_{\alpha,t} \equiv x \cos \alpha + y \sin \alpha = t$$

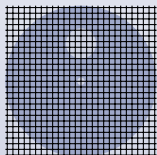
Lo que buscamos es un método que permita recuperar  $\rho$  a partir de las funciones  $P_{\alpha}$  para  $0 \leq \alpha < 2\pi$ .

Consideramos una región no homogénea de densidad superficial  $\rho$



Cada rayo sufre una atenuación medible que depende de

$$P_\alpha(t) = \int_{r_{\alpha,t}} \rho \quad \text{con } r_{\alpha,t} \equiv x \cos \alpha + y \sin \alpha = t$$



$N \times N$  píxeles

**Variante discreta.**  $x_i$  = densidad en el  $i$ -ésimo píxel

Para cada  $t$  y  $\alpha$ ,  $P_\alpha(t) \rightarrow$  ecuación lineal

$$A\vec{x} = \vec{b} \quad \text{con } A \in \mathcal{M}_{L \times N^2}(\mathbb{R}), \quad L \gg N^2$$

Muy dispersa  $\approx N$  incógnitas/ecuación

## La teoría

### Método continuo.

La posibilidad de recuperar  $\rho$  a partir de  $P_\alpha(t)$  está basada en la fórmula de inversión de la transformada de Fourier:

$$f(x, y) = \int_{\mathbb{R}^2} \widehat{f}(\xi, \eta) e^{2\pi i(x\xi + y\eta)} d\xi d\eta$$

donde  $\widehat{f}(\xi, \eta) = \int_{\mathbb{R}^2} f(x, y) e^{-2\pi i(x\xi + y\eta)} dx dy$ .

*Si  $\rho \in C_0^\infty$  entonces se obtiene a partir de  $P_\alpha(t)$  por medio de la fórmula*

$$\rho(x, y) = \iiint_{\Omega} r P_\alpha(t) \cos(2\pi r(x \cos \theta + y \sin \theta - t)) d\theta dr dt$$

*con  $\Omega = [0, 2\pi] \times \mathbb{R}^+ \times \mathbb{R}$ .*

## La teoría

### Método discreto.

**Problema técnico:** si queremos una resolución  $1000 \times 1000$ , tenemos  $10^6$  incógnitas y el método de eliminación de Gauss requeriría del orden de  $10^{18}$  operaciones si el número de ecuaciones iguala al de incógnitas.

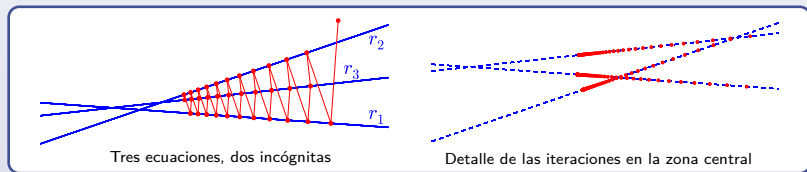
**Problema teórico:** el sistema está sobredeterminado y entonces errores infinitesimales debidos a deficiencias en el modelo o a la toma de datos, lo convertirán en incompatible.

¿Cómo resolver un sistema lineal sin solución?

## Algoritmo de Kaczmarz

Proyectamos sucesivamente sobre cada uno de los hiperplanos que definen las ecuaciones, partiendo de un punto inicial arbitrario.

Para dos incógnitas las iteraciones se acercan a puntos que son “casi soluciones” cuando se tiene un sistema incompatible que es “casi compatible determinado”.



$$A\vec{x} = \vec{b}$$

$$A\vec{x} = \vec{c} \quad \text{con solución } \vec{x} = \vec{s}$$

Llamemos  $\gamma$  a una cota superior de  $|b_i - c_i|/\|\vec{f}_i\|$ , con  $\vec{f}_i$  la fila  $i$ -ésima. Es conveniente elegir los hiperplanos con probabilidad proporcional a  $\|\vec{f}_i\|$ .



## Algoritmo de Kaczmarz

Proyectamos sucesivamente sobre cada uno de los hiperplanos que definen las ecuaciones, partiendo de un punto inicial arbitrario.

(Needell) *Las iteraciones  $\vec{x}_k$  al aplicar el algoritmo de Kaczmarz aleatorio a un sistema incompatible satisfacen*

$$\mathbb{E}\|\vec{x}_k - \vec{s}\| \leq (1 - M^{-2}S^{-2})^{k/2} \|\vec{x}_0 - \vec{s}\| + \gamma MS, \quad \begin{cases} M = \inf \frac{\|\vec{x}\|}{\|A\vec{x}\|} \\ S = \left(\sum |a_{ij}|^2\right)^{1/2} \end{cases}$$

*donde la esperanza es relativa a la elección de los hiperplanos.*

$$A\vec{x} = \vec{b}$$



$$A\vec{x} = \vec{c} \quad \text{con solución } \vec{x} = \vec{s}$$

Llamemos  $\gamma$  a una cota superior de  $|b_i - c_i|/\|\vec{f}_i\|$ , con  $\vec{f}_i$  la fila  $i$ -ésima. Es conveniente elegir los hiperplanos con probabilidad proporcional a  $\|\vec{f}_i\|$ .

## La práctica

**Método continuo.** La hipótesis  $\rho \in C_0^\infty$  no es realista por ejemplo en el cuerpo humano y la falta de regularidad causa complicaciones en la aplicación del análisis de Fourier (fenómeno de Gibbs).

↪ Desde los resultados de J. Radon hasta que las TAC formaron parte de la práctica médica habitual, pasaron más de 60 años.

**Método discreto.** La dificultad práctica consiste en trabajar con grandes volúmenes de datos y tratarlos estadísticamente.

## La práctica

**Método continuo.** La hipótesis  $\rho \in C_0^\infty$  no es realista por ejemplo en el cuerpo humano y la falta de regularidad causa complicaciones en la aplicación del análisis de Fourier (fenómeno de Gibbs).

↪ Desde los resultados de J. Radon hasta que las TAC formaron parte de la práctica médica habitual, pasaron más de 60 años.

**Método discreto.** La dificultad práctica consiste en trabajar con grandes volúmenes de datos y tratarlos estadísticamente.

- ▶ *Tomografía de emisión de positrones (PET).* Detecta la radiación que aparece en forma de rayos gamma cuando un isótopo radiactivo decae por desintegración beta.
- ▶ *Resonancia magnética nuclear (NMR).* Utiliza fenómenos oscilatorios relacionados con el espín de los núcleos de hidrógeno.

# Codificación y compresión

## El modelo

Cuando tecleamos un texto en un ordenador, internamente las letras se codifican en bytes.

	(ASCII)	
M	→	01001101
U	→	01010101
R	→	01010010
C	→	01000011
I	→	01001001
A	→	01000001



Supongamos que tenemos un fichero muy grande compuesto solo por M's y U's.

Si lo codificamos con  $M \mapsto 0$  y  $U \mapsto 1$ , entonces un texto de longitud  $L$  ocupará  $L$  bits.

¿Se puede hacer mejor?

M

Si aproximadamente el 75 % de las veces aparece una M y el resto una U, independientemente, podemos codificar pares de símbolos mediante:

U

MM	→	0
MU	→	10
UM	→	110
UU	→	111

UMMMUUMMMM → 110011100

Es fácil, pero no trivial, comprobar que nunca hay ambigüedades al descodificar.

M

Si aproximadamente el 75 % de las veces aparece una M y el resto una U, independientemente,

U

$9/16$	→	MM	→	0
$3/16$	→	MU	→	10
		UM	→	110
$1/16$	→	UU	→	111

La longitud media que ocupará un par de símbolos tras la codificación es

$$\ell_2 = 1 \cdot \frac{9}{16} + 2 \cdot \frac{3}{16} + 3 \cdot \frac{3}{16} + 3 \cdot \frac{1}{16} = \frac{27}{16}$$

Un texto de longitud  $L$  ocupará en media  $L\ell_2/2 = 0,84375L$  bits tras la codificación.

Sabiendo la estadística de los símbolos que lo componen,  
¿hasta dónde y cómo se puede comprimir un fichero?

## La teoría

Alfabeto:

$$\mathcal{A} = \{s_1, s_2, \dots, s_n\}$$

con probabilidades (independientes)  $p_i$ ,  $\sum p_i = 1$

Una codificación es una forma de transformar los  $s_i$  en tiras de bits que no den lugar a ambigüedades al descodificar cadenas.

Entropía:

$$H = H(p_1, \dots, p_n) = - \sum_{i=1}^n p_i \log_2 p_i$$

Entropía  $\approx 0 \rightarrow$  poca incertidumbre

Esta definición la introdujo C.E. Shannon en su famoso artículo de 1948 “*A mathematical theory of communication*”, donde también introduce la palabra “bit”.



## La teoría

Alfabeto:

$$\mathcal{A} = \{s_1, s_2, \dots, s_n\}$$

con probabilidades (independientes)  $p_i$ ,  $\sum p_i = 1$

Una codificación es una forma de transformar los  $s_i$  en tiras de bits que no den lugar a ambigüedades al decodificar cadenas.

Entropía:

$$H = H(p_1, \dots, p_n) = - \sum_{i=1}^n p_i \log_2 p_i$$

Suponiendo independencia, para la codificación que minimiza la tasa media  $r$  de bits por símbolo cuando se agrupan de  $N$  en  $N$ , se verifica

$$H \leq r < H + \frac{1}{N}.$$

En particular  $r$  tiende a  $H$  cuando  $N \rightarrow \infty$ .

## Método LZ78

Se descompone la cadena en *frases*, donde una frase es una subcadena que añade un carácter nuevo a otra que haya aparecido antes.

salsa\_salada

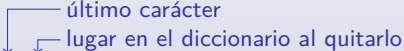
## Método LZ78

Se descompone la cadena en *frases*, donde una frase es una subcadena que añade un carácter nuevo a otra que haya aparecido antes.

|s|a|1|sa|\_|sa1|ad|a#|

La división en frases se corresponde a un *diccionario*, cuya entrada  $k$  es la  $k$ -ésima frase y añadimos la entrada 0 con la cadena vacía.

0	1	2	3	4	5	6	7	8
$\emptyset$	s	a	1	sa	_	sa1	ad	a#

Codificación: 

(s, 0), (a, 0), (1, 0), (a, 1), (\_, 0), (1, 4), (d, 2), (#, 2)

Sea  $\mathcal{L}_N$  la longitud media para bloques de longitud  $N$  con LZ78, suponiendo independencia, entonces  $\mathcal{L}_N/N \rightarrow H$  para  $N \rightarrow \infty$ .

## La práctica

Alfabeto:

$$\mathcal{A} = \{M, U\}$$

con probabilidades  $1/2$

Entropía:

$$H = 1$$

Lo mejor que se puede hacer es la codificación trivial  $M \mapsto 0$  y  $U \mapsto 1$ , con la que un texto de longitud  $L$  ocupará  $L$  bits.

Sin embargo estos compresores generales dan un rendimiento del orden de un 30% peor. Por ejemplo, con  $10^6$  caracteres se tiene:

gzip  $\longrightarrow$  1272544 bits

zip  $\longrightarrow$  1273680 bits

bzip2  $\longrightarrow$  1281704 bits

rar  $\longrightarrow$  1291960 bits

## La práctica

Alfabeto:

*Español*

Entropía:

$H \approx 4,09$

Debemos esperar que  $L$  caracteres en español no se compriman más allá de  $4,09L$  bits.

Para el texto de *La regenta* de 1869610 caracteres, el límite serían 7646705 bits, sin embargo los resultados experimentales son:

gzip  $\longrightarrow$  5840976 bits

zip  $\longrightarrow$  5842112 bits

bzip2  $\longrightarrow$  4273320 bits

rar  $\longrightarrow$  4874552 bits

$\rightsquigarrow$  La paradoja se explica porque en el cálculo teórico de las longitudes medias se supone la independencia de los caracteres de las palabras y eso está lejos de la realidad en un texto.

## El problema de los horarios

## El modelo

Queremos decidir la existencia de un horario al especificar ciertas incompatibilidades y, en caso de que exista, contar el número de posibilidades.

Supongamos que hay alumnos que han escogido simultáneamente algunas asignaturas.

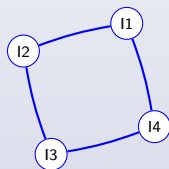
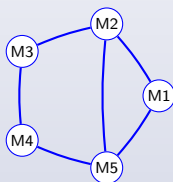
### Matemáticas

$$M1 \cap M2 \cap M5$$

$$M2 \cap M3$$

$$M3 \cap M4$$

$$M4 \cap M5$$



### Informática

$$I1 \cap I2$$

$$I2 \cap I3$$

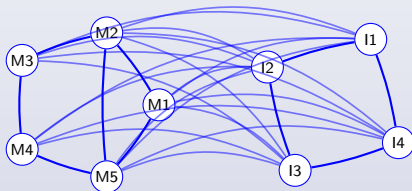
$$I3 \cap I4$$

$$I4 \cap I1$$

Si disponemos de 5 horas lectivas  
¿cuántos horarios se pueden hacer?

## El modelo

Debido al doble grado informática-matemáticas, el decanato impone que no deben coincidir asignaturas de matemáticas e informática, entonces se añaden muchas incompatibilidades.



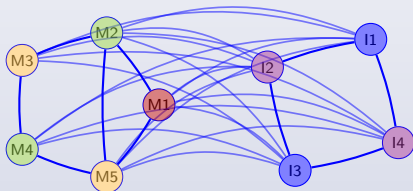
¿Es todavía posible impartir los cursos en 5 horas lectivas?  
¿de cuántas formas?



## El modelo

A cada hora posible del horario se le asocia un color. Una **coloración del grafo** es una forma de asignar colores a los vértices de manera que los vértices conectados por aristas tengan colores diferentes.

Por ejemplo, un horario para las cinco horas lectivas podría ser:



- 10:00-10:50
- 11:00-11:50
- 12:00-12:50
- 13:00-13:50
- 14:00-14:50

## La teoría

*Para cada grafo  $G$  de  $n$  vértices, existe un polinomio  $P_G$  de grado  $n$  tal que el número de formas de colorear los vértices con  $k$  colores es  $P_G(k)$ . Además, para cada arista  $a$  de  $G$  se cumple*

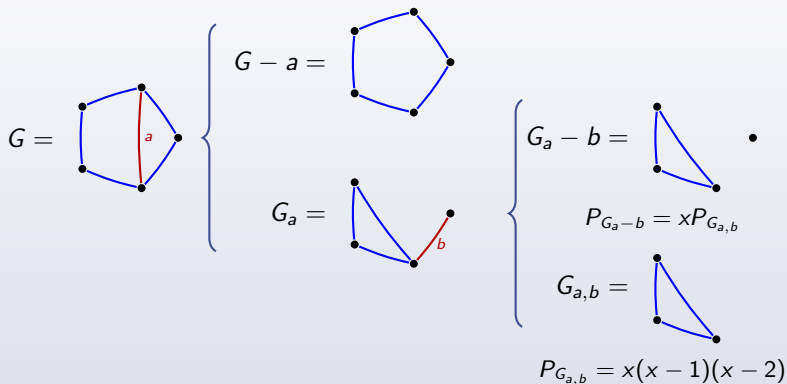
$$P_G = P_{G-a} - P_{G_a}$$

*donde  $G - a$  es el grafo que resulta al eliminar la arista  $a$  y  $G_a$  el que resulta al colapsarla en un punto identificando los vértices de sus extremos.*

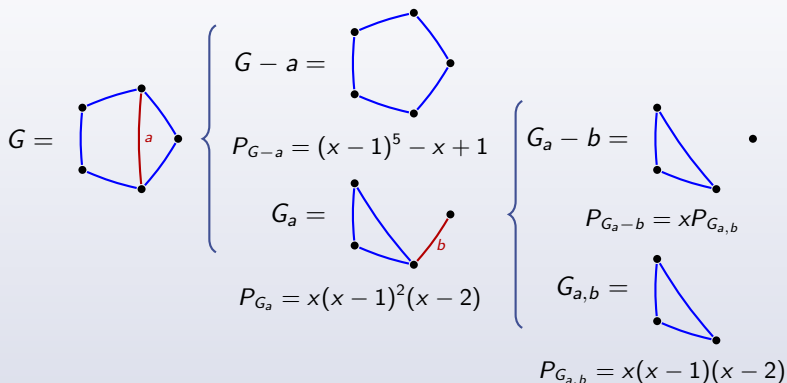
Hay grafos para los que el *polinomio cromático*  $P_G$  sencillo:

$$\begin{array}{ll}
 P_G = x(x-1)^{n-1} & \text{si } G \text{ es un árbol} \\
 P_G = (x-1)^n + (-1)^n(x-1) & \text{si } G \text{ es el } n\text{-ágono} \\
 P_G = x(x-1)(x-2)\cdots(x-n+1) & \text{si } G \text{ es el grafo completo } K_n
 \end{array}$$

## Horario de matemáticas



## Horario de matemáticas



$$P_G = x(x-1)(x-2)(x^2 - 3x + 3)$$

$P_G(1) = P_G(2) = 0$     Se necesitan al menos 3 horas, dando lugar a 18  
 $P_G(3) = 18$             horarios posibles ( $3 = 18/3!$  salvo permutaciones)

Sea  $G$  un grafo y sean  $G_1$  y  $G_2$  subgrafos suyos.

- Si  $G = G_1 \cup G_2$  con unión disjunta,  $P_G = P_{G_1} P_{G_2}$ .
- Si  $G = G_1 \cup G_2$  y  $G_1 \cap G_2 = K_n$ ,  $P_G = P_{G_1} P_{G_2} / P_{K_n}$ .
- Si  $G$  se obtiene uniendo todos los vértices de  $G_1$  con los de  $G_2$ , escribiendo  $P_{G_i} = \sum_j a_{ij} P_{K_j}$  se cumple  $P_G = \sum_k \sum_j a_{1j} a_{2j} P_{K_{j+k}}$ .

Matemáticas

$$x(x-1)(x-2)(x^2-3x+3) = P_{K_5} + 4P_{K_4} + 3P_{K_3}$$

Informática

$$(x-1)^4 + x - 1 = P_{K_4} + 2P_{K_3} + P_{K_2}$$

La última propiedad da para el grafo total:

$$\begin{aligned} P_G &= P_{K_9} + 6P_{K_8} + 12P_{K_7} + 10P_{K_6} + 3P_{K_5} \\ &= x(x-1)(x-2)(x-3)(x-4)(x^4 - 20x^3 + 155x^2 - 546x + 733) \end{aligned}$$

Como es lógico,  $P_G(5)/5!$  vuelve a salir 3.

## La práctica

El gran problema es que el algoritmo no es muy eficiente cuando el número de vértices crece.

- El tiempo de computación utilizando el algoritmo sin atajos es del orden de  $\varphi^s$  donde  $\varphi$  es la razón áurea y  $s$  es la suma de vértices y aristas.
  - ↪ Parece ser que a lo más que se ha llegado es a conseguir tiempos exponenciales en el número de vértices.
- Por otra parte, se conocen algunos resultados negativos que dan pocas esperanzas para que un nuevo algoritmo mágico nos sorprenda.
  - ↪ Es lo que se llama un problema *NP-hard*.

En definitiva, el algoritmo es práctico para pocos vértices o cuando hay simetrías o propiedades especiales.

Colgaré esta presentación en:

---

<https://www.uam.es/fernando.chamizo>

---

**Gracias por la atención**