posició general per als $v \in \mathcal{N}(u)$. Recordeu que tota l'estona $m = |A|$.

Test$_k(G, u)$
Tria aleatòriament $p < n^5$ i $\mathbf{c} \in (\mathbb{Z}/p\mathbb{Z})^m$, p primer
Computa la $\mathcal{N}_k(u)$-immersió modular $f = f_{\mathbf{c}}$
$\forall v \in V \backslash \mathcal{N}_k(u)$ comprova si $\text{rang}(f(\mathcal{N}_k(v))) = k$
  si $\forall v$ és compleix la condició, retornar *cert*
  altrament retornar *fals*

Finalment, l'algorisme MonteCarlo per determinar si G és *k-vèrtex-connex*,

AlgKvc(G, k)
Repetir $(n \log n)/(n-k)$ passos
  escollir aleatòriament $u \in V$
  si **Test**$_k(G, u)$ torna *fals*,
    escriure G no és *k-vèrtex-connex*
Si per tots els u triats, **Test**$_k(G, u)$ torna *cert*,
  escriure G és *k-vèrtex-connex*
altrament escriure G NO és *k-vèrtex-connex*

LLW demostren que **AlgKvc**(G, k) retorna la resposta correcta amb probabilitat $> 1 - \frac{1}{n}$. La complexitat de l'algorisme és $O(n^{2.37} + nk^{2.37})$, on l'exponent 2.37 ve de la multiplicació de matrius. Als anys 80, **AlgKvc**(G, k) va ser l'algorisme més eficient per valors de $k \geq \sqrt{n}$.

Respecte a l'evolució posterior de la complexitat del problema, l'any 1991 Cheriyan i Reif van estendre el resultat de LLW a digrafs, utilitzant les mateixes tècniques i amb la mateixa complexitat [3]. El 2000, Henzinger *et al.* [7] tornant a utilitzar tècniques de fluxos van produir un algorisme aleatoritzat amb una millora de la complexitat a $O(kn^2)$. El 2019 Nanogkai *et al.* van dissenyar un algorisme aleatoritzat amb una complexitat $O(m+k^{7/3}n^{4/3})$ per a valors de k fins a $\sqrt{n}$ [17]. El que és remarcable en aquest darrer treball és que és el primer algorisme que, per $m = O(n)$, trenca la barrera de complexitat $O(n^2)$, i la rebaixa a $O(n^{4/3})$. Als anys 70, es va conjecturar que el problema es pot resoldre en temps $\Theta(m)$. Molt recentment, Li *et al.* [10] han donat una solució al problema en temps $O(m^\alpha)$, on $\alpha \geq 1$ és l'exponent de l'algorisme més eficient que es coneix per computar el màxim flux, actualment $O(m^\alpha)$, amb $\alpha = \frac{4}{3} + o(1)$ [6]. L'algorisme de Li *et al.* queda molt a prop de resoldre la conjectura oberta des de l'any 70.

L'algorisme LLL

Jordi Guàrdia, Departament Matemàtiques, UPC

L'algorisme LLL va ser dissenyat el 1981 per tres "L" insignes de les matemàtiques: els germans Hendrik i Arjen Lenstra i Lazlo Lovász. Tot i que el seu origen remot és un problema de programació lineal i que la primera aplicació que en van fer els autors va ser un problema aritmètic, d'aleshores ençà ha anat revolucionant molts i diversos camps de les matemàtiques. Si algun moment heu de resoldre un problema amb xarxes, molt probablement l'algorisme LLL us serà molt útil.

Descrit molt breument, l'algorisme ens permet trobar bases *bones* de xarxes. Una *xarxa*[3] en $\mathbb{R}^n$ és el conjunt format per totes les combinacions lineals enteres de n vectors linealment independents. Un exemple molt senzill seria la xarxa generada pels vectors $(1, 0), (0, 1)$ en $\mathbb{N}^2$, que està formada pels punts de $\mathbb{R}^2$ amb coordenades enteres. Naturalment, una xarxa donada admet moltes bases i algunes són molt més convenients que d'altres. Per exemple, la xarxa que acabem de descriure també admet la base $(5341, 393), (4213, 310)$, però amb aquesta base serà bastant més difícil treballar. Per decidir quines bases són millors que d'altres solem tenir en compte la mida dels seus vectors. Ara bé, aquesta "mida" pot ser el seu mòdul en la mètrica euclidiana habitual de $\mathbb{R}^n$ o, més en general, la mida respecte a una altra mètrica qualsevol, donada per una forma quadràtica definida positiva.

Les xarxes són una eina molt potent per a resoldre problemes pràctics d'àmbits molt variats. És molt important, doncs, tenir un bon algorisme per trobar una base "petita" d'una xarxa donada i això és el que fa justament l'algorisme LLL. En particular, l'algorisme calcula una *bona aproximació* del vector més

[3] En altres àmbits, són el que s'anomenen reticles, o *lattices* de l'anglès

curt de la xarxa, la determinació del qual és un dels problemes fonamentals en teoria de xarxes.

Per bé que l'algorisme és força senzill, no el descriurem amb detall i ens limitarem a remarcar que la seva clau és la noció de "*base reduïda*" d'una xarxa respecte a la seva mètrica. L'algorisme LLL troba una base reduïda d'una xarxa en temps polinomial respecte a la dimensió de la xarxa, la qual cosa és una fita sorprenent. De fet, els mateixos autors van dedicar un temps a repassar a fons l'algorisme perquè no els semblava pas possible!

Vegem com aplicar l'algorisme en un problema senzill. Volem factoritzar el polinomi $f(x) = x^5 + 3x^4 + 3x^3 - 5x^2 - 15x - 15$ sobre els enters. Mitjançant l'ús algun mètode numèric, trobem que $x_0 = 1.71$ és una aproximació d'una de les arrels. Per trobar un factor de f a partir de x_0, considerem la xarxa generada pels vectors

$$v_1 = (M, 1, 0, 0, 0), v_2 = (Mx_0, 0, 1, 0, 0),$$
$$v_3 = (M\,x_0^2, 0, 0, 1, 0), v_4 = (M\,x_0^3, 0, 0, 0, 1),$$

on M és una constant prou gran que ajustem a conveniència; en aquest cas, $M = 100$ basta. L'algorisme LLL ens diu que el vector $v_4 - 5v_1$ és un vector molt petit d'aquesta xarxa, és a dir, que $x_0^3 - 5 \approx 0$, i això ens suggereix que $x^3 - 5$ pot ser un factor de f. Només ens cal fer la divisió entera per comprovar que, efectivament $f = (x^3-5)(x^2+3x+3)$. Aquesta tècnica és la base de l'algorisme de factorització de polinomis amb coeficients enters, que és justament el problema que resolen Lovász i els germans Lenstra en l'article on presenten l'algorisme ([9]).

Des d'aquest primer treball, l'algorisme LLL s'ha aplicat en moltes tasques pròpies de la teoria de nombres, des de problemes bàsics com la identitat de Bézout fins a qüestions molt elaborades com la determinació del grup de classes d'un cos de nombres o el càlcul d'equacions modulars. Però les aplicacions de l'algorisme han anat molt més enllà de la teoria de nombres. La criptografia és un altre dels camps on l'impacte de l'algorisme ha estat molt important. Poc temps després de la seva publicació, ja es va utilitzar per trencar el conegut criptosistema *knapsack* de Merkle-Hellman ([21]) i des de llavors altres protocols criptogràfics han sofert atacs basats en LLL, de manera que l'algorisme ha esdevingut una eina molt popular en criptoanàlisi.

A banda de les seves nombroses aplicacions pràctiques, l'algorisme també ha tingut implicacions teòriques significatives. L'estudi de la complexitat computacional del *shortest vector problem* va molt lligat al desenvolupament de l'algorisme. Des del punt de vista merament matemàtic, és la clau de la refutació de la conjectura de Mertens ([20]), que es considerava una via per demostrar la hipòtesi de Riemann. Fins i tot li podem atribuir un cert valor històric: la publicació de l'algorisme en una revista considerada de matemàtiques pures com és *Mathematische Annalen* va ser una proposta de H. Lenstra amb la intenció d'atreure l'atenció dels matemàtics cap a la teoria de la complexitat.

Widgerson i l'atzar

Albert Atserias, Departament de Ciències de la Computació, UPC

Una conseqüència del fet que el conjunt dels nombres reals és de cardinal no numerable és que els nombres transcendentals existeixen. És més, en són l'amplíssima majoria. Ara bé, demostrar que un nombre real concret, com e, π, o $e + \pi$, és transcendental és tota una altra història. Un fenomen semblant, potser encara més (?) dramàtic, es dona en alguns dominis discrets. Una bona part del treball d'Avi Wigderson sobre el rol de l'aleatorietat en l'algorísmia i la computació es pot entendre com l'intent d'explicar aquest fenomen i treure'n profit, reduint-lo a un cas concret. Gràcies al seu treball, avui dia sabem que el problema de construir funcions booleanes de n bits que, per una banda siguin estríctament explícites, però que, per l'altra, no es puguin descriure amb circuits de portes lògiques AND, OR, NOT de mida polinòmica en n és un *problema de construcció universal*. Expliquem-ho amb una mica més de detall.