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 estrictament 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.

Fixem un enter positiu k i considerem el conjunt de totes les funcions booleanes de n bits $f : \{0,1\}^n \to \{0,1\}$, per a un enter n prou gran. En nombre, d'aquestes funcions n'hi ha 2^{2^n}. Comparem-ho, ara, amb el nombre de funcions booleanes de n bits que es poden computar amb un circuit amb n^k portes lògiques. D'aquestes n'hi ha, com a molt, n^{4kn^k}, que és una fita superior al nombre de circuits amb n^k portes lògiques. Aquesta és una quantitat certament gran, però negligible respecte de la primera, quan k està fixat. Per tant, si diem que una funció de n bits és *fàcil* quan és computable amb circuits de mida n^k, llavors resulta que gairebé totes les funcions de n bits són *difícils*. Dit això, demostrar que una funció booleana explícitament concreta és difícil és un problema d'una extraordinària dificultat. En aquest context, que una funció sigui explícitament concreta vol dir que sigui computable en temps polinòmic de la seva taula de veritat, o, equivalentment, que sigui computable en temps exponencial $2^{O(n)}$ en la mida n de l'entrada de la funció. Per posar-ne un exemple, considerem la funció $\lambda(N)$ de Liouville, que determina la paritat del nombre de factors primers d'un nombre natural N de n bits. Ningú sap del cert (tot i que se sospita que no), si $\lambda(N)$ es pot computar amb n^k portes lògiques, per a k constant i n arbitrariament gran, on n és el nombre de bits en la representació binària de N. De fet, ara com ara, no es coneix cap funció de n bits que sigui explícitament concreta per a la qual s'hagi demostrat que calguin més que $5n$ (!) portes lògiques (i el !, aquí, *no* és el factorial).

La pregunta és, però, perquè voldríem poder demostrar que la funció $\lambda(N)$, o qualsevol altra de n bits, no es pot computar amb n^k portes lògiques. Part de la resposta és que aconseguir-ho seria demostrar P $\neq$ NP, un dels Set Problemes del Mil·leni. Però, per al cas que ens ocupa, la resposta és que fer-ho ens permetria confirmar, gràcies als treballs de Wigderson i coautors, que l'aleatorietat en algorísmia i, per extensió, en bona part del món físic real, és un recurs que, en principi, resulta substituïble per un increment de la potència de càlcul molt més petit del que es creia fa només dues dècades. Aquest va ser un canvi de paradigma molt important perquè va obrir la porta a atacar alguns problemes que es consideraven fora d'abast. I algun es va poder resoldre sense assumir P $\neq$ NP ni cap altra conjectura.

La definició del concepte clau es remunta a la feina de Yao [27] de principis dels anys 80 del segle passat. Fem servir U_m per denotar la distribució de probabilitat uniforme sobre el conjunt $\{0,1\}^m$ de les cadenes de m bits. Sigui $G : \{0,1\}^m \to \{0,1\}^n$ una funció que converteix m bits en n bits, on $n \gg m$. Considerem ara un conjunt T de tests estadístics, és a dir, un conjunt de funcions booleanes $t : \{0,1\}^n \to \{0,1\}$ que aplicarem a cadenes de n bits. Direm que G és un generador pseudoaleatori respecte del conjunt de tests T si, per a cada $t \in T$, se satisfà que

$$|\Pr[t(U_n) = 1] - \Pr[t(G(U_m)) = 1]| \leq 1/n.$$

A l'esquerra, U_n denota una mostra aleatòria de la distribució uniforme sobre n bits. A la dreta, U_m denota una mostra aleatòria de la distribució uniforme sobre m bits, la qual extenem a n bits aplicant-hi el generador G. Els generadors pseudoaleatoris, per a tests diversos, es van concebre, originàriament, per donar fonaments a la teoria de la criptografia moderna. I és prou evident que els generadors pseudoaleatoris també tenen aplicacions en algorísmia: un generador pseudoaleatori ens permetria *simular* n bits aleatoris generant-ne només $m \ll n$. Si m fos, posem per cas, $2 \log n$, i T fos una classe de tests que garanteix que l'algorisme funciona correctament amb la mostra escrutinada, llavors podríem eliminar la necessitat de bits aleatoris completament. En efecte, amb un increment multiplicatiu en temps de n^2 podríem: (1) recórrer l'espai de totes les possibles llavors $z \in \{0,1\}^{2 \log n}$ per al generador G, (2) executar l'algorisme probabilista fent servir els n bits de $G(z)$ en comptes de la seva font d'aleatorietat genuïna i (3) prendre'n el vot majoritari de les respostes. L'algorisme resultant seria completament determinista, i retornaria la resposta correcta sempre que l'algorisme probabilista inicial tingués probabilitat d'error prou petita (menor que $1/2 - 1/n$ és suficient).

Un repte important, però, és que, per a classes de tests prou generals, com els que es necessiten en criptografia i en algorísmia, no es coneix *cap* generador pseudoaleatori que sigui prou

explícitament computable. Cap ni un. I aquí és on entren els treballs de Nisan-Wigderson [19] i Impagliazzo-Wigderson [8] de finals dels 90. La contribució és que, per tal de construir un generador que extengui m bits a n bits, i garantir que sigui pseudoaleatori respecte dels tests computables amb circuits de mida n^k, és suficient (i necessari) construir una funció booleana de $O(m)$ bits que no sigui computable per circuits de mida $O(n^k)$. Per exemple, si aconseguíssim demostrar que la funció $\lambda(N)$ de Liouville requereix circuits de mida n^3 per a nombres N de n bits, llavors podríem simular n bits aleatoris per a tests de mida n^2 amb, només, uns $O(n^{2/3})$ bits aleatoris reals.

La dualitat hardness vs. randomness que sorgeix de la transformació proposada per Wigderson i coautors té el mèrit de ser universal. De fet, la transformació és tan general, que dona lloc a moltíssimes derivades, algunes de les quals *incondicionals*, és a dir, sense que depenguin de conjectures per ara irresol·lubles com P $\neq$NP. Per exemple, Trevisan [23, 24] va ser el primer a adonar-se que els generadors de Nisan-Wigderson permetien produir *extractors* millors dels que es coneixien fins llavors.

Què és, doncs, un extractor? Un extractor és una funció booleana, eficientment computable, que purifica l'aleatorietat d'una font impura de bits aleatoris i n'extreu gairebé tota l'entropia. El descobriment de Trevisan va ser que el generador de Nisan-Wigderson és, de fet, un extractor explícit en si mateix! La clau de volta és el punt de partida que ens ha dut fins aquí: que gairebé totes les funcions booleanes de n bits són difícils per a circuits de mida $o(2^n/n)$. Fent números rodons, una font impura de n bits que tingui n^ϵ bits d'entropia proporciona una funció booleana d'uns $\epsilon \log n$ bits que no és computable amb circuits de mida $o(n^\epsilon/\log n)$, amb alta probabilitat. Per tant, aplicant-hi Nisan-Wigderson, obtenim un generador pseudoaleatori que proporciona uns $n^\epsilon/\log n$ bits aleatoris per a *tots* els tests. És a dir, proporciona un extractor.

El fet que extractors explícits es poguessin construir incondicionalment va inspirar molts d'altres a produir altres objectes pseudoaleatoris. Una de les fites més espectaculars és potser el treball del mateix Wigderson, en col·laboració amb Barak, Rao i Shaltiel [1], que va acabar produint la primera millora significativa en dècades per al problema de construcció de grafs de Ramsey. De saber construir grafs amb 2^n vèrtexs sense subconjunts homogenis de $2^{n^{1/2}}$ vèrtexs es va passar a saber construir-ne sense subconjunts homogenis de 2^{n^ϵ} vèrtexs per a qualsevol $\epsilon > 0$. El problema de construir grafs de 2^n vèrtexs sense subconjunts homogenis de $O(n)$ vèrtexs, que seria essencialment òptim, segueix obert. De ben segur, la feina de Wigderson en aquest camp inspirarà les generacions futures a resoldre aquest i molts altres dels problemes fonamentals que sorgeixen quan hom es planteja, seriosament, la pregunta de si l'atzar existeix i si es pot simular.

Combinatòria, grafs expansors i grafs límit

Oriol Serra, Departament Matemàtiques, UPC

Exposarem dos exemples de contribucions excepcionals a la matemàtica discreta degudes a Widgerson (grafs expansors) i Lovász (límits de grafs), i que exemplifiquen dues línies de recerca on l'interacció entre combinatòria i informàtica teòrica resulta cabdal.

Expansors

Els grafs expansors són un dels objectes més fascinants de la teoria de grafs. En la seva aplicació a les xarxes de comunicació, una de les característiques importants dels grafs és que no tinguin colls d'ampolla, de manera que el tràfic d'informació a la xarxa es pugui repartir de manera homogènia. Una manera de capturar aquesta propietat és demanar que el nombre de nodes que estan connectats als d'un conjunt X de nodes sigui proporcional al cardinal d'aquest conjunt. Si ∂X denota el conjunt de nodes fora de X que són adjacents a algun node de X la condició s'escriu $|\partial X| \geq c|X|$, on la constant c és la mateixa per tots els conjunts X que tenen com a molt $n/2$ nodes (aquesta darrera condició resulta suficient i és tècnicament millor). Les xarxes de comunicació tenen la característica que cada node té un nombre petit de veïns.