Butlletí de la Societat Catalana de Matemàtiques
Vol. 11, núm. 2, 1996. Pàg. 19–29.

Comprensió geomètrica i eficiència algorísmica

Ferran Hurtado

Resum

Aquest article descriu alguns trets essencials de la geometria computacional, disciplina establerta com a branca diferenciada en temps recents. La primera secció tracta de la constitució de la matèria, la segona en descriu el cos central i la tercera les seves imbricacions bàsiques. El text segueix fidelment en esperit, però no pas al peu de la lletra, el contingut de sengles conferències sobre aquest tema, promogudes per la Societat Catalana de Matemàtiques, que l'autor realitzà el 1995 a la Universitat de Barcelona i a la Universitat de les Illes Balears.

1 Sorgiment de la geometria computacional

A l'inici del segle XX el camí marcat per a les matemàtiques posava l'accent en aspectes de fonamentació i afavoria un creixent procés d'abstracció. Cap als anys setanta, però, la diversificació i la complementació de tendències era ja un fet, del qual fou clar testimoni l'aparició de textos tan importants com *The Art of Counting*, de P. Erdös [Er], i *The Art of Computer Programming*, de D. E. Knuth, [Kn]. Tot i que la incidència ha estat molt variable segons les diverses àrees, és ben clar que la difusió dels ordinadors i la multiplicació del seu ús han comportat grans canvis en molts aspectes de les matemàtiques, han fet tractables problemes que no ho eren, n'han creat molts de nous, i han esperonat, en particular, el desenvolupament espectacular de la matemàtica discreta.

Tal com s'explica a [NH], el significat del terme *computació* s'ha expandit notòriament des de la introducció dels *computadors*, ara fa una cinquantena d'anys. Atenent als objectes que hom processa, destacarem tres tipus d'aplicacions dels ordinadors. La primera generació va ser la dels càlculs numèrics, aplicats sobretot a problemes científics i tècnics. La segona, estimulada per necessitats més comercials i administratives, incorporava llargues llistes de dades (per exemple alfabètiques), amb relació a les quals calia llegir, emmagatzemar, moure, seleccionar. Tot això naturalment perviu, i amb molta força, però hem viscut ja una tercera generació d'aplicacions dominada pel processament d'informació geomètrica i gràfica, present en àrees tan diverses com ara la medicina, la cartografia, el control de robots o el disseny artístic, i que comença a resultar quotidiana per a tothom. La geometria computacional ha emergit per la necessitat de donar resposta des de la matemàtica a aquesta nova i creixent demanda.

En la seva accepció més àmplia i moderna, la geometria computacional consisteix en l'estudi dels problemes geomètrics des del punt de vista de les ciències de la computació. Amb una definició molt més restrictiva, però que descriu prou bé el que ha estat el seu nucli des dels seus inicis a mitjan anys setanta, pot dir-se que la disciplina se centra en el disseny i l'anàlisi d'algorismes geomètrics eficients. Té una forta connexió amb el camp considerablement més antic de la geometria discreta i combinatòria; de fet, molt sovint és impossible (a més d'estèril) de fer-hi una línia divisòria.

És ben sabut que quan hom busca antecedents històrics es fa difícil de frenar, i això en matemàtiques sol conduir, si més no, a l'antiga Grècia. De fet, és absolutament correcte de considerar que les construccions euclidianes amb regle i compàs són essencialment algorismes geomètrics. Però si ens centrem en el sentit modern de la disciplina, pot dir-se que una data clau del seu sorgiment és el període 1975-1978, en què M. I. Shamos va realitzar la seva tesi doctoral [Sh], la qual va concitar molta atenció. Des d'aleshores el desenvolupament de la disciplina ha estat explosiu, tant pel que fa a la seva expansió teòrica com pel nombre d'investigadors involucrats. En l'actualitat són molt nombrosos els congressos i les reunions que s'hi dediquen, i són també moltes les revistes amb atenció exclusiva o parcial a la matèria. La publicació dels primers reculls d'articles, com ara [To] i [Pr] va ser acompanyada per la de textos més sistemàtics com [Me] i [PS] i després seguida per l'aparició de diverses monografies centrades en temes més específics, com ara el de la complexitat combinatòria [Ed], el de la visibilitat [O'R1] o el de la robòtica [SY]. Després d'una dècada de docència a les universitats que lideren el progrés científic i tecnològic, diversos investigadors professors anuncien ara la publicació de nous llibres de text, conseqüència lògica de l'experiència adquirida, però també de l'evolució i la sedimentació de la disciplina. D'aquesta "segona generació", en formen part ja el tractat elemental [O'R2], el [BY], més avançat, un que se centra en les aplicacions dels mètodes probabilístics [Mu], i un que descriu una eina que ha resultat bàsica en la combinatòria dels problemes geomètrics [SA].

Com hem dit abans, en cert sentit pot dir-se que les aplicacions van precedir la disciplina, i ara que aquesta ja té un cos teòric sòlidament constituït, com que els seus vessants "pràctics" corresponen a tecnologia de màxima avantguarda, la demanda de resultats continua amb la mateixa força i exigència que al principi. És per això que diem que en geometria computacional, les aplicacions tenen un protagonisme essencial.

2 Cos central de la disciplina

Tal com hem dit, el que hom pretén és de resoldre problemes geomètrics *de forma eficient.* Però, quin significat té exactament aquest darrer adjectiu? I com es fa per obtenir tal eficiència? Respondre a aquestes dues preguntes és l'objectiu bàsic d'aquest article, ja que equival a descriure en què consisteix la geometria computacional.

Naturalment hi ha molts criteris estètics i pràctics que poden tenir-se en compte per considerar que un algorisme és *bo*. L'ocupació de memòria, la facilitat d'implementació, l'estabilitat numèrica, són aspectes molt importants. Tanmateix, centrarem el nostre comentari en la velocitat de processament —que hom anomena *complexitat en temps*—, tant per simplificar la nostra exposició, com per l'especial rellevància que ha tingut i té aquest factor.

Hauria de ser clar que el que ens convé és una forma de mesurar el temps d'execució d'un algorisme sense fer-ne la programació i independent dels ordinadors que podrien ser usats a la pràctica. El que es fa és utilitzar un *model de computació* per mesurar l'eficiència dels mètodes de resolució, és a dir, un model teòric de màquina en què certs processos tenen assignat un cost (temps) unitat. Aleshores hom compara els algorismes avaluant per a cadascun d'ells la seva complexitat, que en primera versió podem definir com el cost total d'executar l'algorisme en la situació més desfavorable. Aquesta és l'*anàlisi en el cas pitjor*; no comentarem aquí l'*anàlisi del cas esperat*, la qual comporta una presència substancial d'eines probabilístiques.

El model utilitzat més sovint és el RAM real, semblant al descrit en [AHU] però acceptant que un nombre real pot ser emmagatzemat en una unitat de memòria. Les següents operacions es consideren primitives, i poden realitzar-se amb cost unitari:

1. operacions aritmètiques bàsiques ($+$, $-$, $\times$, $/$);
2. comparació de dos nombres reals ($<$, $\leq$, $=$, $\neq$, $\geq$, $>$);
3. accés a la memòria;
4. (ocasionalment) arrels enèsimes, funcions trigonomètriques, l'exponencial i el logaritme.

És habitual analitzar el comportament dels algorismes en ordre de magnitud, ignorant els coeficients constants (que sí que es consideren, però, a efectes pràctics), i tot això segons algun paràmetre n lligat a la mida de les dades rebudes per l'algorisme. Escrivim $T(n) \in O(f(n))$ per fitar superiorment la complexitat temporal $T(n)$ d'un algorisme. Més formalment:

$$T(n) \in O(f(n)) \Longleftrightarrow \exists n_0 \in \mathbf{N}\,,\ \exists \alpha \in \mathbf{R}^+, \text{ tals que } \ T(n) \leq \alpha \cdot f(n) \ \ \forall n \geq n_0.$$

Vegem ara amb més detall alguns exemples per als quals no és difícil fer un primer algorisme i estudiar-ne una mica l'eficiència en el cas pitjor.

Començarem amb un exemple de plantejament elemental, que ens permetrà d'entendre millor el context. Considerem el problema següent: donats n punts del pla $\{P_1, \dots, P_n\}$, per les seves coordenades, determinar-ne el parell situat a distància mínima. Per a aquest problema la quantitat de punts a l'entrada, n, és clarament un bon paràmetre per a comparar algorismes. D'altra banda, hi ha una solució trivial: calcular la distància entre cada dos, tot memoritzant la mínima. Com que cada distància es calcula en un nombre constant d'operacions de cost unitari, i el càlcul es fa $\binom{n}{2}$ vegades, aquest algorisme té un cost $O(n^2)$.

Es pot fer millor? La primera reflexió que us hauríeu de fer és que l'anterior solució no té res de geomètrica: si ens haguessin donat en una matriu una quantitat arbitrària $q_{i,j}$ per a cada dos punts, i volguéssim trobar el parell P_s, P_t, amb mínima $q_{s,t}$, el mètode anterior s'aplicaria idènticament. Però les distàncies no són quantitats arbitràries, al contrari, estan fortament lligades les unes amb les altres! Potser això permetrà, potser no, de batre la complexitat del mètode anterior. La segona reflexió és ara clara, i explica el títol d'aquest article: la geometria computacional no consisteix a "programar", i no es fan prestidigitacions amb cap llenguatge; el que cal és estudiar i demostrar, si es pot, quines propietats geomètriques permeten de trobar mètodes no trivials. La feina crucial consisteix a esbrinar quins teoremes geomètrics poden resoldre un problema i aplicar-los; si no estan disponibles, caldrà, evidentment, imaginar-los i demostrar-los.

Explicar la solució al problema anterior ultrapassa les possibilitats d'aquest article (però vegeu si voleu [PS]). Direm, però, que existeix una solució òptima $O(n \log n)$. És això una gran millora? Rotundament sí: no s'ha de perdre de vista que no estem jugant amb una dotzena de punts, sinó que les aplicacions típiques treballen amb desenes o centenes de milers de punts, situació que fa bo un algorisme $O(n \log n)$ i completament inútil un algorisme $O(n^2)$.

Certament, no tots els problemes són com l'anterior, en què hi ha una solució trivial acceptable per a entrades petites, i en voldríem una altra de millor comportament. En d'altres casos la solució per *força bruta* és de complexitat exponencial, i el sol fet de provar que existeix una solució d'ordre de magnitud polinòmic ja és una gran proesa. Fins i tot hi ha problemes per als quals no es coneix cap solució, i una demostració que existeix algun algorisme ja és una gran notícia. Però vegem ara un segon exemple, per aclarir d'altres aspectes importants.

Sigui S un conjunt de n punts de E^d. S'anomena *envolupant convexa de* S, denotada $conv\, S$, el conjunt de punts expressables com a combinacions convexes de punts de S (combinacions afins amb coeficients a $[0, 1]$). Equivalentment, $conv\, S$ és la intersecció de tots els convexos que contenen S.

A més de ser una eina indispensable en àrees com la de planificació de moviments sense col·lisió, o la de l'anàlisi de la forma, el càlcul de l'envolupant convexa és una etapa preliminar en multitud de processos geomètrics, per la qual cosa és fonamental disposar de mètodes eficients per a calcular-la.

Com que la definició no permet el càlcul, el primer que caldrà serà tenir teoremes que en discretitzin la descripció (podeu veure [Be] si voleu una panoràmica dels resultats que esmentarem, o [Gr], [Zi] per a més detall). Com que S és finit, $conv\, S$ és un politop (polígon en 2D, políedre en 3D) que té com a vèrtexs un subconjunt de S. A més, el teorema de Carathéodory diu que $conv\, S$ és la reunió de $(d + 1)$-símplexs amb vèrtexs a S. Això ens dóna un algorisme: per a cada subconjunt de $d + 1$ punts de S (n'hi haurà $O(n^{d+1})$) considerem el símplex que defineixen i mirem si altres punts són a dins, perquè es podran rebutjar; els símplexs definits pels supervivents donen $conv\, S$. El cost total és $O(n^{d+2})$ (al pla, per exemple, estaríem fent feina $O(n)$ per a $O(n^3)$ triangles).

Magnífic, però es pot fer millor? Bé, hi ha un altre teorema que diu que $conv\, S$ és la intersecció de semiespais tancats que contenen S, tals que l'hiperplà que els defineix conté d punts de S. Això ens dóna un algorisme $O(n^{d+1})$. Pensem-ho en el cas del pla: per a cada recta definida per dos punts de S (i n'hi ha $O(n^2)$) mirem si és que hi ha punts de S pels dos cantons, en temps $O(n)$.

Hem millorat, doncs; però es pot rebaixar aquest cost? Ja veieu quin és el procediment: buscar teoremes, eventualment produir-los, que donin solucions més eficients. La pregunta natural és ara quan ens donarem per satisfets. Hi haurà, potser, un límit infranquejable? Amb d'altres paraules, sabríem donar una fita inferior per a la complexitat del problema? Hauríem de demostrar que *tot algorisme* esmerçaria com a mínim certa quantitat de temps.

En el cas de l'envolupant convexa podem donar una fita inferior com segueix. La descripció de $conv\, S$ inclou no solament la de cadascuna de les cares (vèrtexs, arestes...) sinó també de les incidències entre elles. Alguna de les etapes de qualsevol algorisme haurà d'explicitar aquestes cares i incidències i, per tant, el nombre d'aquestes és una fita inferior per a qualsevol algorisme. El teorema de la fita superior (McMullen, 1971) diu que el total de cares i incidències és $O(n^{\lfloor d/2 \rfloor})$. Aquest resultat és, a més, ajustat, car hi ha politops, com ara els anomenats *cíclics*, que tenen aquest ordre de complexitat. Un argument especial, que no reproduïm, demostra que per a $d = 2$ l'envolupant convexa no pot calcular-se per sota de l'ordre de magnitud de $n \log n$. Tot plegat dóna una fita inferior que expressem en el teorema següent:

1 Teorema *Per a $d \geq 2$, el problema de calcular l'envolupant convexa de n punts de E^d té complexitat $\Omega(n \log n + n^{\lfloor d/2 \rfloor})$.*

(el paper de Ω descrivint una fita inferior es formalitza anàlogament a com hem fet per a la O, en el cas de la fita superior).

Com veiem, això ens deixa un bon forat entre la complexitat del millor algorisme que hem donat i la fita inferior: caldrà continuar! Aquesta situació és comuna a moltíssims problemes, i els investigadors ens esforcem per obtenir o bé fites inferiors més altes o bé algorismes de menys cost. Quan la complexitat d'un algorisme coincideix amb la fita inferior de la complexitat del problema, es diu que l'algorisme és *òptim*. Per a la vostra tranquil·litat, tot i que és impossible de desenvolupar-los aquí, direm que en el cas de l'envolupant convexa ja s'han aconseguit algorismes òptims, començant pel cas $d = 2$ (Graham, 1972), després per a $d = 3$ (Preparata i Hong, 1977), per a $d > 3$ senar qualsevol (Seidel, 1981) i, finalment, per a $d > 2$ parell qualsevol (Chazelle, 1993).

Dissortadament, el mètode suara descrit, consistent a fitar la mida de la informació que ha de donar l'algorisme com a sortida, no sempre dóna fites inferiors de qualitat. En veurem un altre fent una variació del problema anterior. Un punt que pertanyi a un conjunt finit $S \in E^d$ direm que és *extremal* si existeix un hiperplà que el separa de tots els altres punts de S. Els punts extremals de S són els vèrtexs de *conv* S. Suposem que volem un algorisme que calculi *els punts extremals de* S. Si $|S| = n$, la sortida de l'algorisme ja no té ara la fita $\Omega(n^{\lfloor d/2 \rfloor})$ perquè no necessitem descriure tota l'estructura facial de *conv* S, sinó només una llista dels seus vèrtexs, la qual cosa ens deixa tot just amb una fita $\Omega(n)$.

L'obtenció de fites inferiors és un camp d'interacció amb la geometria algebraica real. A principis dels vuitanta Steele i Yao (1982) i Ben-Or (1983) van adaptar al camp semialgebraic (una expressió semialgebraica és una equació o inequació polinòmica) un resultat de Milnor (1964) i Thom (1965) que fita la suma dels nombres de Betti d'una varietat algebraica. En versió sumaríssima el resultat diu que si W és un subconjunt de E^d i denotem per $cc\,W$ el seu nombre de components connexes, aleshores per a tot algorisme capaç de decidir per a tot punt de E^d si pertany o no a W, efectuant decisions semialgebraiques, existeix algun punt de E^d que requereix l'avaluació de $\Omega(\log(cc\,W) - d)$ expressions semialgebraiques.

Retornem ara al problema anterior, quedem-nos, per claredat, en el cas bidimensional: sigui S un conjunt arbitrari de n punts del pla; volem un algorisme que calculi els punts extremals de S. Associem a cada $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ el punt $f(S)$ de E^{2n} definit per $f(S) = \{x_1, y_1, x_2, y_2, \ldots, x_n, y_n\}$. Ara considerem el subconjunt de E^{2n} definit per $W = \{f(S) \mid \text{ tots els punts de } S \text{ són extremals}\}$. Ara és clar que qualsevol algorisme que trobi els punts extremals d'un subconjunt del pla també decidirà si un punt de E^{2n} pertany o no a W, i si sabéssim quantes components té W, el resultat anterior ens donaria una fita inferior per a tal decisió. Es pot demostrar que $cc\,W$ és $\Omega(n!)$ per la qual cosa el problema de calcular els punts extremals dels S del pla veiem, finalment, que té complexitat $\Omega(\log(n!) - 2n)$, o sigui $\Omega(n \log n)$.

Hi ha més mètodes per a obtenir fites inferiors, com també hi ha algunes estratègies genèriques per a tractar d'obtenir algorismes; esperem, però, que per a una primera aproximació, aquestes línies siguin suficients.

Conclourem aquesta secció amb una observació que hauríeu de copsar amb claredat. L'esforç per a millorar l'eficiència del càlcul no ve guiat tan sols per l'afany de perfecció que és característic de les matemàtiques. A l'hora de les aplicacions, la diferència dels temps d'execució és substancial. Cal pensar que un polígon pot tenir, en un cas típic, no menys d'uns quants milers de costats, i on un algorisme $O(n^3)$ hauria de fer milers de milions d'operacions, un altre que sigui $O(n \log n)$ només en fa uns quants milers. No hi ha avenços en el *hardware* que permetin, ni de lluny, acceleracions tan dràstiques. És per això que el disseny d'algorismes eficients, a més de ser una activitat científica, figura a l'avantguarda del progrés tecnològic. De fet, moltes realitzacions recents i espectaculars de la informàtica, que els profans atribueixen a l'augment del poder de càlcul de les màquines, estan lligades a la substitució d'alguns algorismes naïfs per d'altres més eficients.

3 Imbricacions bàsiques

És possible que la secció precedent us hagi donat una mica la idea que per a cada problema particular cal pensar quins fets geomètrics en poden donar una bona solució, i, si no són coneguts, estudiar-los i establir els teoremes oportuns. Aquesta "individualitat" dels problemes podia ser real fins a cert punt en els primers temps però no pas ara, quan la geometria computacional té ja un nucli sòlidament constituït i ha desenvolupat eines, estructures i mètodes que han estat decantats com a bàsics, de forma que molts resultats s'obtenen per l'encadenament adequat de resultats interns de la disciplina, tal com és habitual en matemàtiques. D'altra banda, ha estat necessari desenvolupar fonaments geomètrics allà on no hi havia respostes, on els problemes eren nous, o bé ho era el punt de vista. En particular, la interacció entre la geometria computacional i la geometria discreta ha estat i segueix essent enormement intensa i fructífera. De fet, tal com hem comentat prèviament, sovint hauríem de parlar de fusió d'aquestes disciplines, més que no d'interacció.

A tall d'exemple, considerem la teoria de la transversalitat geomètrica. Una *k-transversal* d'una família de subconjunts de E^d és una k-varietat que els talla tots. La teoria de la transversalitat "clàssica" tracta d'obtenir condicions per a l'existència de transversals, i es va originar amb el Teorema de Helly, un dels grans resultats geomètrics d'aquest segle (i de tots els temps):

2 Teorema (Helly, 1923) *Sigui $\mathcal{A}$ una família d'almenys $d+1$ convexos de E^d, compactes o en nombre finit. Si cada $d+1$ d'ells tenen intersecció no buida, aleshores existeix un punt comú a tots els membres de $\mathcal{A}$.*

Observeu que el teorema pot enunciar-se dient que si cada $d+1$ dels convexos admeten una 0-transversal, aleshores existeix una 0-transversal de tota la família. El resultat no es generalitza a k-transversals sense l'addició de diverses restriccions, i dels successius progressos de la primera època potser el més rellevant fou el de Hadwiger:

3 Teorema (Hadwiger, 1957) *Una família $\mathcal{A}$ de convexos compactes del pla, disjunts dos a dos, admet una línia transversal si, i només si, existeix un ordre lineal en $\mathcal{A}$ tal que cada tres conjunts de $\mathcal{A}$ poden ser tallats per una línia consistentment amb l'ordre.*

Les generalitzacions d'aquest teorema a dimensions superiors són ja dels anys vuitanta. Però ara el que volem és considerar aquest tema des del punt de vista computacional, on també és interessant perquè diversos problemes de computació geomètrica acaben traduint-se en termes de transversalitat. En aquest context el que es vol és *construir* una k-transversal, i de vegades totes. Si $\mathcal{A}$ és una família de convexos de E^d, el conjunt $\mathcal{T}_k^d(\mathcal{A})$ de les k-transversals de $\mathcal{A}$ és un subespai de la varietat grassmanniana que té com a punts les k-varietats de E^d. Construir totes les k-transversals de $\mathcal{A}$ significa donar una descripció completa (però finita) de $\mathcal{T}_k^d(\mathcal{A})$. Vegem uns pocs resultats per fer-nos-en una idea:

4 Teorema (Edelsbrunner, O'Rourke i Seidel, 1986) *Sigui $\mathcal{A}$ una família de politops convexos de E^d amb un total de n_0 vèrtexs. Es pot construir $\mathcal{T}_{d-1}^d(\mathcal{A})$ en temps $O(n_0^d)$.*

5 Teorema (Avis i Doskas, 1990) *Sigui $\mathcal{A}$ una família de n politops convexos de E^d amb un total de n_1 arestes. Es pot trobar un hiperplà transversal de $\mathcal{A}$, si n'hi ha, en temps $O(n\, n_1^{d-1})$.*

6 Teorema (Hershberger, en premsa) *Sigui $\mathcal{A}$ una família de polígons convexos del pla amb un total de n_0 vèrtexs. Es pot construir $\mathcal{T}_1^2(\mathcal{A})$ en temps $O(n_0 \log n_0)$.*

Per a obtenir tots aquests teoremes *constructius* ha estat necessari entrar a fons en la teoria de la transversalitat i, consegüentment, aquest ha estat un camp d'interacció efervescent en els últims quinze anys entre els camps discret i computacional, de forma que no només s'han obtingut algorismes, sinó també una notable multiplicació dels resultats teòrics coneguts.

Aquest ha estat un exemple, però ja us imaginareu que aquesta interacció ha fecundat moltes altres àrees de problemes i és encara molt intensa. Actualment el cos teòric bàsic de la disciplina, a més de coneixements geomètrics que podríem anomenar genèrics, inclou ja diversos capítols ben constituïts de fonaments geomètrics específics.

El mateix s'ha esdevingut amb relació al camp de la geometria combinatòria, íntimament imbricat als aspectes discrets i computacionals. Un dels motius és que l'estructura combinatòria d'un problema geomètric sovint decideix quin mètode algorísmic resol el problema amb la màxima eficiència. Un altre dels motius és que freqüentment la complexitat i l'anàlisi de les solucions estan relacionades amb el recompte, exacte o en ordre de magnitud, de certs objectes geomètrics.

Vegem un exemple d'aquesta relació. Una (n,s)-*successió de Davenport-Schinzel* és una successió finita $S = (s_1, s_2, \ldots, s_m)$ tal que

1. el nombre de símbols diferents a S és n;
2. termes consecutius són sempre diferents;
3. S no conté cap subsuccessió de longitud $s+2$ constituïda per dos símbols diferents alternats.

Denotem per $\lambda_s(n)$ la longitud màxima d'una (n,s)-successió de Davenport-Schinzel; el problema combinatori fonamental és calcular o estimar $\lambda_s(n)$ amb més precisió que la donada per la fita trivial $\lambda_s(n) \in O(n^2)$. Per a $s = 1$ es té trivialment $\lambda_1(n) = n$, car la subsuccessió prohibida per la condició (3) és $\ldots\alpha\ldots\beta\ldots\alpha\ldots$, és a dir, cap símbol no pot repetir-se. Deixem al lector com a exercici la demostració que $\lambda_2(n) = 2n-1$. Però per a $s \geq 3$ ja només es tenen estimacions asimptòtiques, aconseguides fa pocs anys, i amb demostracions molt complicades (vegeu, si voleu, [SA]).

L'esforç de la comunitat d'investigadors per a resoldre el problema des que va sorgir, ara fa uns vint-i-cinc anys, ha estat molt intens en els últims temps per les seves implicacions en el camp geomètric, com ara veurem. Considerem les gràfiques de n funcions

$f_1, f_2, \ldots, f_n$ reals de variable real, contínues a tot $\mathbf{R}$ i tals que cada dues es tallen com a màxim en s punts. Molts problemes de geometria discreta i computacional, i també dels camps de la visió artificial, la informàtica gràfica i la robòtica, exigeixen estudiar la descomposició del pla produïda per la reunió d'aquestes corbes; en particular, molt sovint és necessari utilitzar la regió que queda "per sota" de totes les corbes. Equivalentment, sigui f la funció definida per $f(x) = \min\{f_1(x), \ldots, f_n(x)\}$. Si s'ha de treballar amb f, la pregunta crucial és quina complexitat combinatòria té, o si preferiu, de quantes "peces" està feta f. Considerem ara la successió T de símbols obtinguda tot escrivint els índexs de les corbes f_i en l'ordre en què apareixen al llarg de f d'esquerra a dreta. Aleshores T és composta com a màxim de n símbols diferents, dos de consecutius mai no seran iguals, i no hi pot haver cap $(s+2)$-subsuccessió alternada $\ldots i \ldots j \ldots i \ldots j \ldots$ perquè això implicaria que la gràfica de f_i i la de f_j es tallessin en almenys $s+1$ punts. Per tant el nombre de peces de f és com a màxim $\lambda_s(n)$, i veiem així la connexió amb el problema combinatori. La recíproca és certa, és a dir, per a qualsevol (n, s)-successió S de Davenport-Schinzel es poden construir n funcions com les anteriors tals que la successió d'índexs associada a $f = \min\{f_1, \ldots, f_n\}$ sigui S. Quan els dominis de les funcions són intervals, aleshores la complexitat de la funció mínima és $\lambda_{s+2}(n)$. Per exemple, si ens mirem des de $(0, -\infty)$ una família de n rectes opaques, cadascuna de color diferent, veurem una successió de com a màxim $\lambda_1(n) = n$ colors, però si eren segments la longitud pot pujar a $\lambda_3(n)$ que és de l'ordre $O(n\alpha(n))$, on $\alpha(n)$ és la inversa de la funció d'Ackermann i és sempre creixent, però amb extremada lentitud.

Fins aquí hem parlat dels solapaments més intensos de la geometria computacional amb àrees matemàtiques fins a cert punt "tradicionals", però és clar que, en tant que ciència de la computació, hi ha una presència substancial d'aspectes de la informàtica teòrica, particularment pel que fa al disseny i l'anàlisi d'algorismes i a les estructures de dades. Fins i tot en el cas que no conegueu aquest camp, segurament no us sorprendrà saber que existeixen diverses estratègies algorísmiques abstractes, i mètodes genèrics d'avaluació de complexitats, que cal conèixer i utilitzar; en canvi, potser us resultarà menys esperat saber que l'eficiència dels algorismes també depèn intensament de l'organització i lagestió adequada de la informació. Ho veurem primer amb un exemple no geomètric.

Suposem que tenim un conjunt S de n nombres i que quan rebem un altre nombre x, volem saber quants elements de S són menors que x i després afegir x a S per a futures iteracions. Evidentment això es pot fer en temps $O(n)$ sense més que comparar x amb cadascun dels elements de S. Ara bé, si ens prenem la feina prèvia d'ordenar els elements de S i els posem creixentment en un vector, és clar que per biparticions successives podem respondre quants elements de S són inferiors a en temps $O(\log n)$. Però si aleshores inserim x dins el vector, en la posició que li hagi correspost, haurem de desplaçar en una posició tots els elements més grans que x i això torna a tenir cost $O(n)$. Afortunadament, existeix una *estructura de dades* que permet de mantenir un conjunt totalment ordenat de forma que les localitzacions, les insercions, i també les supressions, es poden fer en temps $O(\log n)$ (vegeu per exemple [CLR] si hi esteu interessats).

Aquest tipus d'operacions s'han de fer sovint durant el denvolupament d'un algorisme, per tant és imprescindible saber quines són les millors estructures de dades disponibles per a assolir la màxima eficiència. D'altra banda, a més d'aquestes estructures genèriques n'existeixen també moltes d'especialment dissenyades per a resoldre problemes geomètrics importants. Considerem, per exemple, els següents:

1. Donats n plans de E^3, per a cadascun dels punts que es rebran posteriorment volem saber quin dels plans té immediatament per sobre (en el sentit de les z).

2. Donat un conjunt S de n punts del pla, per a cadascun dels punts que es rebran posteriorment volem saber quin dels punts de S és més a prop.
3. Donada una descomposició del pla en regions poligonals, amb un total de n costats, per a cadascun dels punts que es rebran posteriorment volem saber a quina de les regions pertany.

És trivial respondre aquestes preguntes en temps $O(n)$, però, no podríem estructurar la informació geomètrica rellevant de forma que es pogués contestar més de pressa? Sense dubte no us escaparà l'interès de la rapidesa en la resposta per a aquests problemes de *localització*, i la seva importància en les aplicacions. L'eficiència s'ha aconseguit fent l'estudi que ha permès de fer aflorar les propietats geomètriques crítiques, però també ha estat necessari emprar o dissenyar estructures de dades que permeten l'acceleració de la resposta.

Naturalment hi ha moltes més disciplines amb interaccions significatives amb la geometria computacional, però certament les tres exposades són les de més importància, i suficients per a aquesta primera aproximació que us hem volgut donar.

No volem concloure aquesta breu presentació sense fer esment de la curiosa situació de la geometria sintètica (també anomenada clàssica), i que té una presència substancial en geometria computacional. El protagonisme que en aquesta tenen les aplicacions prioritza enormement el treball en dues i tres dimensions i, a més, el seu caràcter constructiu ha obligat a recuperar bona part de les tècniques, els conceptes i un esperit, amb llarga tradició, però que havien perdut modernitat. Bona prova del decidit revifament de la geometria sintètica és la recent aparició de llibres dedicats a la presentació de problemes *oberts* de geometria (com ara [CFG] i [KW]), en els quals la restricció al cas del pla és exclusiva o substancial. No deixa de ser curiós que una disciplina bona part de la qual ha estat considerada com de gran bellesa, però de nul·la aplicabilitat pràctica, figuri ara com a principal protagonista d'algunes de les aplicacions més punyents de la matemàtica actual, rotundament implicades amb les tecnologies de màxima avantguarda.

Referències

[AHU] A. V. Aho, J. E. Hopcroft i J. D. Ullman: *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[Be] M. Berger: *Geometry*, Springer-Verlag, 1987.

[BY] J. D. Boissonnat i M. Yvinec: *Géometrie Algorithmique*, Ediscience International, 1995.

[CFG] H. T. Croft, K. J. Falconer i R. K. Guy: *Unsolved Problems in Geometry*, Springer-Verlag, 1991.

[CLR] T. H. Cormen, C. E. Leiserson i R. L. Rivest: *Introduction to Algorithms*, MIT Press, 1990.

[Ed] H. Edelsbrunner: *Algorithms in Combinatorial Geometry*, Springer-Verlag, 1987.

[Er] P. Erdös: *The Art of Counting* (selecció editada per J. Spencer), MIT Press, 1973.

[Gr] B. Grünbaum: *Convex Polytopes*, Wiley & Sons, 1967.

[Kn] D. E. Knuth: *The Art of Computer Programming*, Addison-Wesley, 1973.

[KW] V. Klee i S. Wagon: *Old and New Unsolved Problems in Plane Geometry and Number Theory*, Mathematical Association of America, 1991.

[Me] K. Mehlhorn: *Multidimensional Searching and Computational Geometry*, Springer-Verlag, 1984.

[Mu] K. Mulmuley: *Computational Geometry: an Introduction through Randomized Algorithms*, Prentice Hall, 1993.

[NH] J. Nievergelt i K. Hinrichs: *Algorithms & Data Structures*, Prentice-Hall, 1993.

[O'R1] J. O'Rourke: *Art Gallery Theorems and Algorithms*, Oxford University Press, 1987.

[O'R2] J. O'Rourke: *Computational Geometry in C*, Cambridge University Press, 1994.

[Pr] F. P. Preparata: *Advances in Computing Research: Computational Geometry*, JAI Press, 1983.

[PS] F. P. Preparata i M. I. Shamos: *Computational Geometry. An Introduction*, Springer-Verlag, 1985.

[SA] M. Sharir i P. Aggarwal: *Davenport-Schinzel Sequences and Their Geometric Applications*, Cambridge Univ. Press, 1995.

[Sh] M. I. Shamos: *Computational Geometry*, tesi doctoral, Yale University, 1978.

[SY] J. T. Schwartz i C. K. Yap: *Algorithmic and Geometric Aspects of Robotics*, vol. 1, Hillsdale, NJ: Erlbaum, 1987.

[To] G. T. Toussaint (ed.): *Computational Geometry*, North-Holland, 1985.

[Zi] G. M. Ziegler: *Lectures on Polytopes*, Springer-Verlag, 1994.

Departament de Matemàtica Aplicada II
Universitat Politècnica de Catalunya
Pau Gargallo, 5
08028 Barcelona
hurtado@ma2.upc.es