Complejidad Computacional en Fundamentos de las Matemticas Computacionales

Luis Miguel Pardo

November 8, 2006

La investigación del equipo que dirijo (MTM2004-01167) está interesada fundamentalmente en los problemas de complejidad computacional relacionados con los Fundamentos de las Matemticas Computacionales. La actividad involucra varios colectivos, sociedades y revistas entre las que podría destacar la sociedad Foundations of Computational Mathematics (FoCM http://www.focm.net/) Effective Methods in Algebraic Geometry (MEGA, http://www.ricam.oeaw.ac.at/mega2007) o SIGSAM (teniendo como principal actividad el congreso ISSAC, http://issac2006.dima.unige.it/) entre otros colectivos.

Desde nuestra perspectiva, la investigación en Matemáticas Computacionales se divide en tres grandes etapas fuertemente interrelacionadas:

- Diseño y Análisis de Algoritmos. Es en esta etapa donde nuestro grupo de investigación incide de manera más persistente. Previo a cualquier implementación de un algoritmo, procedimiento o método se ha de hacer un *análisis de su eficacia*. Esto involucra considerar dos funciones asociadas a cada algoritmo y problema susceptible de ser tratado algorítmicamente: El **Tiempo** de Ejecución y el **espacio**/memoria requerido para su ejecución. Así algoritmos cuyo tiempo de ejecución venga dado por una función exponencial en el tamaño de la entrada se consideran algoritmos intratables (impracticables) y no procede implementarlos. *Algoritmos cuyo tiempo de ejecución sea polinomial (al menos en promedio) son algoritmos tratables y susceptibles de ser implementados.* En el caso de tratamiento de algoritmos numéricos se deben analizar adicionalmente el *condicionamiento*, la *establidad* y la *robustez* entre otros parámetros.

- Implementación y Experimentación de algoritmos de complejidad tratable y la generación de códigos optimizados de prototipos utilizables en diversos ámbitos.

- La Consultoría o puesta al servicio de la comunidad del prototipo ya diseñado.

Nuestro equipo se dedica fundamentalmente a los aspectos relativos al diseño y análisis de algoritmos que, ante todo, involucran la complejidad computacional de los mismos (eficacia).

Algunos de los problemas centrales en nuestra investifación aparecen en algunas de las listas famosas de problemas para el siglo XXI. Por ejemplo, de la lista de Instituto Clay (http://www.claymath.org/) nos interesan fundamentalmente la Conjetura de Cook ($\mathbf{P} \neq \mathbf{NP}$) y, colateralmente, la Conjetura de Swinnerton-Dyer (*Diseño de un algoritmo para decidir la existencia de soluciones racionales de variedades abelianas). Pero quizás sea la lista de problemas propuestos por Steve Smale la que contenga más problemas en los que se encuentra inmerso e interesado nuestro equipo. Así, de los 18 problemas propuestos por Steve Smale para las Matemáticas del siglo XXI (véase por ejemplo http://mathworld.wolfram.com/SmalesProblems.html) estamos interesados en los siguientes:*

- Problem 3. Does $\mathbf{P} \neq \mathbf{NP}$ (i.e., are **P**-problems equivalent to **NP**-problems)?
- Problem 4. Integer zeros of a polynomial. Also Tewnty Questions, also Is Hilbert's Nullstellansatz solvable in Polynomial Time?.
- Problem 5. Height bounds for Diophantine curves.
- Problem 9. The linear programming problem. Is linear programming solvable in strong polynomial time?.
- Problem 17. Solving polynomial equations by a uniforma algorithm in Polynomial time on the average.

Debe señalarse que **nuestro equipo ha resuelto** afirmativamente el **Problema 17** durante el pasado año 2005. Hemos diseñado un algoritmo numérico (basado en deformación homotópica) que en tiempo polinomial en promedio resuelve sistemas de ecuaciones polinomiales multivariadas. En estos días pretendemos comenzar la fase de implementación del algoritmo. En experiencias anteriores, el equipo diseñó el algoritmo más eficiente para la resolución simbólica de sistemas de ecuaciones polinomiales multivariadas y para el Nullstellensatz de Hilbert (véase la implementación en http://www.math.uvsq.fr/ lecerf/software/kronecker/).

En todo caso, como la complejidad está presente en todo diseño de algoritmo que se precie, todos los problemas de Matemáticas Computacionales en los que se desconoce la existencia de algoritmos tratables son de interés de nuestro equipo.