

Más que difícil, complejo

Por Bruno Massare

Resolver un problema puede ser mucho más que un dolor de cabeza, puede ser imposible. Sobre todo cuando la complejidad sobrepasa ciertos límites. Pero ¿dónde se ubican esos límites? Cuadro de situación: debemos dilucidar cómo recorrer 100 ciudades, que están conectadas por caminos, en el menor tiempo posible. Para resolver este

problema se necesitarían miles de años, aun recurriendo al mejor algoritmo conocido y a la computadora más potente. “Quizás lo podríamos resolver en casos de 15 o 20 ciudades, pero si aumenta el número también aumenta el tiempo que requiere su resolución, al punto de que nos damos cuenta de que no somos capaces de resolverlo durante nuestras vidas”, explica Michael Soltys (36), experto en teoría de la complejidad (fue el tema de su tesis de doctorado en la Universidad de Toronto, Canadá).

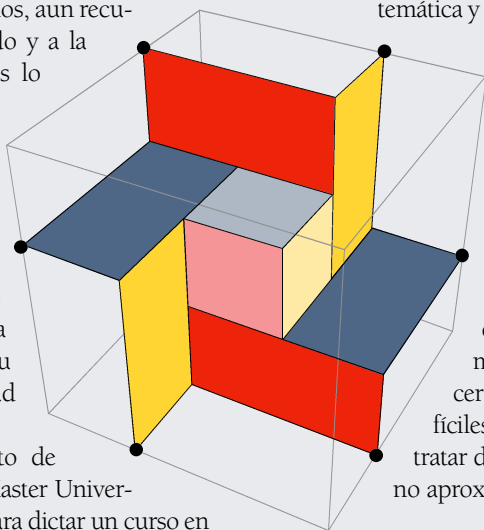
Soltys es docente del departamento de Computación y Software de la McMaster University y en febrero visitó la Argentina para dictar un curso en la Escuela de Verano de la Facultad de Ciencias Informáticas de la Universidad Nacional de Río Cuarto, Córdoba. “Introducción a la Teoría de la Complejidad” fue precisamente el nombre del curso a cargo de Soltys, quien elogia el nivel de los estudiantes argentinos y sostiene que “es importante que un programador pueda distinguir si un problema se puede resolver efectivamente (si es realizable a

La teoría de la complejidad computacional intenta distinguir entre los problemas que pueden resolverse y los que no, y los recursos necesarios para llegar a una solución. Cuáles son sus aplicaciones según un experto canadiense.

través de un algoritmo/programa) o no, a qué algoritmos puede recurrir y que pueda relacionar algoritmos con problemas. Y, entonces, que a un jefe se le pueda explicar que un determinado problema es tan complejo que no se puede resolver de manera completa, aunque sí quizás se pueda aproximar una solución aceptable”.

La teoría de la complejidad computacional nació en el seno de las ciencias de la computación con origen en la matemática y desde los años sesenta ha crecido hasta

independizarse. La mayor parte de sus esfuerzos tratan de entender por qué hay ciertos problemas que son fáciles de resolver y otros que son sumamente difíciles y requieren gran cantidad de recursos computacionales (espacio, tiempo, memoria). Y, también, cómo distinguirlos y estimar qué recursos se necesitan para un determinado problema. “Además, saber qué se puede hacer con estos problemas que son tan difíciles —advierte—. Por ejemplo, quizás tratar de no tener una solución completa, sino aproximarse de a poco a la respuesta.”



PODER DE CÓMPUTO

El problema central de la teoría de la complejidad computacional se conoce como P versus NP. A tal punto es un desafío que el Clay Mathematical Institute de Cambridge, Massachusetts (Estados Unidos), ofrece desde el año 2000 U\$S 1 millón para quien solucione el que se considera uno de los siete problemas aún no resueltos del milenio.

“P” y “NP” son dos clases de complejidad. Ambas clases de problemas pueden resolverse en tiempo polinómico, pero en el segundo a través de un modelo computacional no determinista (más de una posible combinación por cada par estado/símbolo en una máquina de Turing). En el primero de los casos, ejemplos de problemas que pueden resolverse son el de ordenar una lista o buscar un elemento en ella. En el segundo, se trata de otros mucho más complejos de resolver para una computadora (podrían llevar miles de años, como el citado de las ciudades al principio de la nota).

Todo problema de P está en NP, pero no se sabe si hay problemas en NP que no estén en P. Como modelo se ha usado universalmente la máquina de Turing (desarrollada en los años treinta por Alan Turing), un modelo formal de



siado poco, como para saber cuándo va a ser resuelto: son fáciles de explicar, pero las técnicas que existen hoy no son lo suficientemente buenas. Y nadie sabe si dentro de una década un estudiante encontrará una forma de resolverlo”.

Las posibles aplicaciones de la teoría de la complejidad son múltiples. Por ejemplo, para hacer más eficiente a Internet. “Se puede trabajar en encontrar rutas óptimas para que los paquetes de datos circulen más eficientemente. Es un problema muy metido en la complejidad, básicamente de teoría de grafos: tenemos un grafo que está hecho de una determinada manera y debemos encontrar las rutas más cortas, y si las hay, entre cada punto”, dice.

La criptografía es un campo donde esta teoría tiene algo que aportar. En el contexto de la criptografía la complejidad es una ayuda: cuanto más complejo mejor, al revés que en otros campos. “Buena parte de la criptografía se basa en la factorización de enteros, pero a partir de que

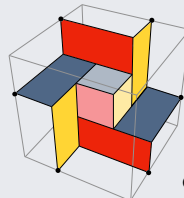
Computación biológica

En cuanto a los avances en el campo de las ciencias de la computación, Michael Soltys dice que la aplicación de modelos computacionales a las ciencias biológicas es el área que más lo sorprende: “Es una especialidad que avanza rápidamente y que está siendo muy requerida. Porque una vez que decodificamos el genoma humano ahora queremos entenderlo bien, saber por qué ciertas enfermedades ocurren más frecuentemente y cómo se relaciona con patrones que se repiten. Es un campo muy interesante, con muchas aplicaciones pese a que es una disciplina prácticamente nueva”, concluye.

asumimos que su complejidad es alta: todos suponemos que la factorización de enteros es muy difícil de vulnerar y esperamos que lo sea, pero sólo lo sospechamos. Nosotros en criptografía intentamos probar que ciertos métodos son realmente difíciles de quebrar. Pero estas pruebas no son pruebas complejas, en el sentido de que siempre asumen algo que no sabemos con certeza”, explica.

Entre los temas que hoy lo ocupan, menciona el problema del razonamiento automático: “Cuando mandamos un vehículo robot a Marte queremos que él tome decisiones por su cuenta, porque la información tarda nueve minutos en ir y volver; necesitamos que si se encuentra con una piedra no nos consulte para esquivarla y seguir su marcha. Así entra a jugar nuestro campo: porque la complejidad de hacer las cosas más simples es tan

grande que para tomar decisiones muy básicas el robot necesitaría años de razonamiento. Nuestro trabajo consiste en que la computadora sea capaz de generar pruebas a teoremas que nosotros le proponemos. Le permitirá elegir el camino más corto para tomar decisiones”.



“ES IMPORTANTE QUE UN PROGRAMADOR PUEDA DISTINGUIR SI UN PROBLEMA SE PUEDE RESOLVER EFECTIVAMENTE”

MICHAEL SOLTYS, experto en teoría de la complejidad

cómputo equivalente a cualquier lenguaje de computación, y que pone en evidencia que hay problemas que esa máquina/modelo no puede resolver.

“No alcanzará con un mayor poder de cómputo, difícilmente eso nos ayude. Por más que la Ley de Moore se siga cumpliendo y así tengamos en los próximos siglos computadoras muchísimo más rápidas, los cambios en problemas tan complejos serían imperceptibles. Lo que sí somos capaces de hacer en este momento es aproximar soluciones, y para eso sí sirve tener toda la capacidad de procesamiento posible, porque son cálculos muy complejos”, sostiene Soltys. Y agrega: “Creo que entendemos el problema dema-