Algunos mitos respecto de NP-completos

Horst H. von Brand
vonbrand@inf.utfsm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

Contenido

Ideas equivocadas

Los problemas NP-completos (con mucha razón) ocupan un lugar central en toda discusión sobre problemas difíciles de resolver. Esto los hace también el centro de conceptos erróneos sobre problemas complejos.

Los problemas NP-completos son los más difíciles

Un problema en NP es resuelto en tiempo polinomial por un algoritmo no-determinista. El tiempo de ejecución de una variante determinista que revisa todas las opciones será exponencial en un polinomio en el largo de los datos de entrada. Hay problemas decidibles más difíciles (requieren aun más tiempo). Aún más difíciles son los problemas no decidibles.

Son difíciles porque hay muchos candidatos

Hay problemas similares que son muy fáciles de decidir.

Hamiltonian es NP-completo, pero la pregunta de si el grafo G tiene un circuito euleriano (pasa por cada arco una vez) tiene solución simple. Candidatos potenciales a solución son ciclos para Hamiltonian y circuitos para Eulerian, y de los últimos definitivamente hay más. Al buscar el camino más corto entre dos vértices de un grafo (Shortest Path) hay un algoritmo eficiente, el de Dijkstra; vimos que hallar el camino más largo (Longest Path) es NP-completo. Para ambos el número de candidatos (caminos en el grafo) es el mismo.

Requieren tiempo exponencial

No. Si P $\neq$ NP sabemos que requieren tiempo más que polinomial, que no es lo mismo que exponencial. Por ejemplo, $n^{\log n}$ crece más rápido que cualquier polinomio (el exponente aumenta sin límite) pero más lento que a^n para todo $a > 1$ (calcule el límite para $n \to \infty$ de $a^n / n^{\log n}$).

Y mientras no se dilucide P = NP, simplemente no sabemos.

Los problemas NP-completos son todos difíciles

El que un problema sea NP-completo significa que tiene instancias difíciles, no que todas lo sean.
Por ejemplo, el problema 3Color (¿Tiene un coloreo con tres colores el grafo G?) es NP-completo. Sin embargo, verificar si G es bipartito es muy simple; si es bipartito, con tres colores sobra.
SAT es NP-completo, pero hay algoritmos lineales en el tamaño de ϕ para 1SAT y 2SAT, que son casos particulares.
Incluso puede darse el caso que el problema sea NP-completo, pero los casos de interés práctico tienen solución simple.

Si P = NP, toda la criptografía se hace inútil

Aun si la criptografía se basa en un problema NP-completo, si resulta que P = NP las constantes y exponentes pueden ser tales que igual no hay una solución práctica. En todo caso, la criptografía generalmente se basa en problemas que no se sabe si son NP-completos, o derechamente no lo son. En particular, el cifrado AES usa claves de largo fijo, por lo que el problema de hallar la clave es revisar un número fijo de posibilidades, o sea, es constante. Claro que con la tecnología actual tomaría más que la edad del universo revisar todas las opciones. Por lo demás, en criptografía interesa que *todas* las instancias sean difíciles (o al menos, que las instancias fáciles sean simples de detectar y evitar).

Si P = NP, toda computación se hace trivial

Vimos que muchos problemas difíciles de interés son NP-completos, pero no todos. Y un algoritmo $O(n^{100})$ para, digamos, TSP, hará poca diferencia práctica en todo caso.

Se han llamado *algoritmos galácticos* a grandes avances *teóricos*, algoritmos que demostraron cotas asintóticas impensadas para ciertos problemas; pero los costos involucrados son tan astronómicos que es preferible usar un «algoritmo asintóticamente ineficiente» pero que en los rangos de alcance previsible simplemente son mejores.

La computación cuántica resuelve todo

Se han desarrollado algoritmos para computadores cuánticos teóricos (los que hay concretamente son muy limitados, hay quienes sostienen que hay limitaciones fundamentales a lo que se puede lograr, lejos de la teoría) para *algunos* problemas que están en NP pero que se sospecha fuertemente no son NP-completos. Otros son algoritmos que mejoran tiempos de ejecución a, por ejemplo, la raíz del tiempo determinista tradicional. Un gran avance, pero siguen más que polinomiales.

Resumen

Vimos varias concepciones erradas sobre problemas NP-completos:

- Hay problemas decidibles más complejos. Y los hay no-decidibles...
- Criptografía no depende de $\mathrm{P} \neq \mathrm{NP}$. Criptografía se interesa en problemas con *todas* las instancias difíciles. NP-completo es que *algunas* instancias son difíciles (suponiendo $\mathrm{P} \neq \mathrm{NP}$).
- Un algoritmo polinomial para SAT no hará automáticamente trivial toda computación. ¿Qué si es $O(n^{100})$? Si fuera $O(n^{\log n})$ no sería polinomial, pero posiblemente práctico...