12

Dez algoritmos que abalaram o mundo

Quando o matemático islâmico Muhammad al-Khwarizmi publicou, na primeira metade do século IX, a sua obra *Hisab al-jabr wa'l muqabalah*, não podia imaginar que quer o seu nome quer o do seu livro seriam imortalizados durante milénios pelos então bárbaros do Ocidente.

O seu livro sintetizava o conhecimento sobre a resolução analítica de equações, apresentava a fórmula resolvente do segundo grau e introduzia um formalismo matemático avançado e o sistema de numeração árabe que hoje usamos. A partir de al-Khwarizmi, o termo *al-jabr* tornou-se sinónimo de resolver equações (Álgebra). E o nome de al-Khwarizmi deu, por outro lado, origem a *algarismo* — para designar cada um dos dígitos da numeração árabe — e *algoritmo* — o termo moderno que designa um procedimento sistemático para resolver problemas matemáticos.

Se vivesse nos nossos dias, al-Khwarizmi decerto ficaria apaixonado pelo número especial da revista *Computing in Science and Engineering* de Janeiro de 2000. Jack Dongarra, da Univer-

sidade do Tennessee, e Francis Sullivan, do Institute for Defense Analysis, elaboraram um trabalho notável: a lista dos dez algoritmos com maior influência no desenvolvimento e na prática da ciência e da engenharia no século XX. Em suma, o Top Ten dos algoritmos do século XX.

Eis então, por ordem cronológica, os dez algoritmos que abalaram o Mundo.

1. O Método de Monte Carlo (Metropolis, Ulam e von Neumann, 1946). O matemático polaco Stan Ulam estava no hospital a recuperar de um esgotamento provocado pelo esforço de guerra (Ulam tinha feito parte do Projecto Manhattan, que em Los Alamos desenvolveu a bomba atómica). Para se entreter, tentou calcular a probabilidade de obter uma mão perfeita no jogo de cartas Solitário. Como o problema é difícil, teve a ideia de aproximar essa probabilidade simulando a distribuição de cartas e observando a frequência relativa. Estava criado o Método de Monte Carlo, MMC (que foi herdar o nome precisamente ao seu carácter aleatório): a obtenção de resultados numéricos sobre sistemas complexos por simulação e não por resolução detalhada das equações que regem o comportamento físico do sistema. Hoje o MMC é utilizado em áreas como a Física de Partículas, astrofísica, radioterapia, fluxo de tráfego automóvel, previsão de índices de mercados de capitais, e muitas mais.

2. O Método do Simplex (George Dantzig, 1947). Na sequência da Segunda Guerra Mundial e em vésperas da guerra da Coreia, o planeamento da logística militar estava na ordem do dia. Como optimizar uma função objectivo (minimizar custos de transporte de tropas, por exemplo) em face de restrições lineares? O matemático George Dantzig (veja-se, a propósito de Dantzig, o capítulo 15, *A lenda do problema insolúvel*), então na Força Aérea americana, obteve uma solução particularmente elegante, criando o algoritmo do simplex: o ponto de optimização da função objectivo em Programação Linear está num vértice da fronteira do domínio de acessibilidade. Hoje em dia é quase impossível encontrar um ramo de actividade industrial ou finan-

ceira que não utilize o Método do Simplex no seu planeamento e atribuição de recursos.

3. Métodos iterativos de subespaços de Krylov (Hestenes, Lanczos e Stiefel, 1950). A resolução de sistemas de equações lineares é, pelo menos desde Gauss, um problema de extrema importância para aplicações científicas. O próprio Gauss criou o primeiro método verdadeiramente eficiente, a eliminação, que continua hoje a ser ensinado nas nossas universidades. O que fazer, no entanto, quando os sistemas têm milhões de variáveis, e as matrizes têm uma estrutura especial (por exemplo, muitos zeros)? Nestes casos, os métodos mais expeditos são de tipo iterativo, utilizando a estrutura algébrica subjacente, e não de força bruta. Uma das primeiras aplicações destes métodos a problemas de fusão, por David Kershaw, baixou o número de iterações necessárias de 208 000 para o método de Gauss-Seidel para... 25! A complexidade algorítmica baixa de $O(N^3)$, típica de método de Gauss, para $O(N^2)$ ou menos, dependendo do tipo de matrizes.

4. Métodos de decomposição para cálculos matriciais (Householder, 1951). As operações matriciais constituem a maior parte do trabalho de análise numérica em problemas científicos e técnicos. Torna-se assim da maior importância conceber métodos que permitam realizar todo o tipo de cálculos matriciais da forma mais eficiente possível. Householder inaugurou a era dos métodos de decomposição: estes problemas podem ser muito simplificados escrevendo as matrizes sob a forma de produto de outras matrizes mais simples (triangulares, por exemplo). A decomposição mais famosa é a LU (*lower-upper*, inferior-superior). A complexidade algorítmica pode melhorar de $O(N^4)$ para $O(N^2)$! Sem métodos de decomposição não existiria hoje *software* científico como o *Mathematica* ou o *Maple*.

5. O compilador de Fortran (Backus, 1957). A criação do Fortran foi descrita como o acontecimento singular mais importante na história da programação de computadores. Por impossível que nos possa parecer hoje, antes do aparecimento do compilador de Fortran a programação de um computador era feita directamente em código máquina ou *assembler*, por vezes mesmo ajustando

interruptores no *hardware*! A revolução conceptual das linguagens de alto nível e o abrir de portas da era da informação deu-se com o Fortran. Grande parte da Torre de Babel de linguagens de programação dos anos 60 e 70 desapareceu hoje; mas o Fortran actual, quase 50 anos depois da sua criação, é ainda a linguagem favorita para aplicações científicas mais pesadas. Uma piada antiga mas sempre actual diz que não se sabe como vai ser a linguagem mais utilizada daqui a dez anos, mas vai chamar-se Fortran!

6. O algoritmo QR (Francis, 1959-1961). A determinação de valores próprios é o problema mais importante, do ponto de vista das aplicações científicas e tecnológicas, do cálculo matricial. Os métodos básicos de Álgebra Linear, embora resolvam o problema, tornam-se impraticáveis quando as matrizes têm ordem superior, digamos, a 10. Por outro lado, são extremamente instáveis do ponto de vista numérico. O método QR fornece uma alternativa: é uma forma de factorização de matrizes que permite métodos iterativos para cálculo eficiente e, acima de tudo, estável, de valores próprios. Sem este algoritmo os problemas de ressonância, por exemplo, no cálculo de estruturas ou na concepção das asas de aviões, seriam quase impossíveis de resolver numericamente.

7. Quicksort (Tony Hoare, 1962). Em face dos problemas anteriores, o problema da ordenação parece quase pateticamente mundano: dada uma lista de N objectos, como ordená-los de forma eficiente? A questão muda de figura se pensarmos na omnipresença deste problema: é raro ele *não* se colocar! O algoritmo *Quicksort* para a ordenação utiliza uma estratégia desarmantemente simples, chamada «dividir para reinar»: tomando um elemento da lista, separa os restantes em duas pilhas, os «grandes» e os «pequenos» (comparados com o elemento seleccionado). Itera depois o procedimento em cada uma das pilhas. No final, a lista está ordenada. Esta estratégia possui uma complexidade computacional média de O(N log N), muito melhor do que a complexidade $O(N^2)$ de outros algoritmos mais complicados. As folhas de cálculo dos computadores pessoais de hoje usam o *Quicksort*.

8. A tranformada rápida de Fourier, FFT (Cooley e Tukey, 1965). Este é, provavelmente, o algoritmo do século. Com um

pequeno artigo de cinco páginas, *An algorithm for the machine calculation of complex Fourier series,* Cooley e Tukey entraram para a História. O problema é simples de descrever: calcular transformadas discretas de Fourier. Este problema é omnipresente na ciência e tecnologia actuais, no processamento de sinais ou na Física Matemática. É impossível descrever todas as aplicações, que vão das telecomunicações à imagiologia médica, de remoção de ruído à espectroscopia. A FFT é um algoritmo que reduz a complexidade computacional da transformada de Fourier de $O(N^2)$ para $O(N \log N)$. Este facto fez literalmente explodir novas áreas científicas e tecnológicas. Por exemplo, técnicas que utilizam intensivamente a transformação de Fourier, como a TAC ou a imagiologia por Ressonância Magnética Nuclear, não poderiam existir *de todo* se não utilizassem o algoritmo FFT de Cooley-Tukey — mesmo utilizando o *hardware* actual. Mesmo algo tão trivial como o formato MP3 para música seria impossível sem FFT. O artigo de Cooley-Tukey ainda hoje é citado centenas de vezes por ano.

9. A detecção de relações inteiras (Ferguson e Forcade, 1977). Durante muito tempo, os cientistas sonharam com um método que lhes permitisse reconhecer uma constante numérica a partir da equação que ela satisfaz. Esse sonho foi realizado com o algoritmo das relações inteiras. Uma relação inteira é um algoritmo que, dados n números reais, constrói uma relação linear inteira entre eles (se existir), ou prova que ela não existe dentro de certos limites. As consequências do algoritmo de relações inteiras são incríveis. Existe uma enorme massa de novos resultados em Teoria de Números impossíveis de obter por métodos clássicos, particularmente relativos a números algébricos. Os progressos no estudo das funções zeta forneceram contribuições insuspeitadas à teoria quântica de campos, nomeadamente no cálculo de diagramas de Feynman, e a problemas combinatóricos. A detecção de relações inteiras originou um novo campo matemático: a Matemática experimental. O leitor interessado em explorar as relações inteiras pode utilizar a interface *Integer Relations,* do canadiano Jonathan Borwein, em http://www.cecm.sfu.ca/projects/IntegerRelations/.

10. O algoritmo rápido multipolar (Greengard e Rokhlin, 1987). Até há pouco tempo era impraticável simular o comportamento de agregados de *N* partículas em interacção gravitacional ou electrostática, a não ser para *N* muito baixo. A razão é simples de compreender: sendo necessário entrar em conta com todos os pares de interacções, a simulação de um agregado de *N* partículas exige o cálculo de $O(N^2)$ interacções. Simular a formação de uma modesta galáxia com um milhão de estrelas, ou de um troço de ADN minimamente realista com 200,000 átomos, era assim computacionalmente impossível. Os algoritmos multipolares vieram revolucionar o panorama. Em vez de se calcularem as interacções partícula a partícula, calculam-se interacções entre expansões multipolares truncadas de agregados de partículas. A complexidade computacional foi reduzida para O(N). A Biologia Molecular e a Astrofísica estão hoje em plena revolução.

Estes são os dez algoritmos que, na opinião de Dongarra e Sullivan, mudaram o mundo. Merece um pouco de atenção, no entanto, compreender *porque é que* os algoritmos mudam o mundo, e porque é que eles são muito mais do que esperteza aplicada ou uma questão de eficiência na utilização de recursos.

Todos estamos habituados à chamada «lei de Moore», segundo a qual a tecnologia de computadores nos faz ganhar um factor de 2 em velocidade de processamento a cada 1,5 anos. Assim, mesmo admitindo acelerações nesta tendência, daqui a dez anos teremos computadores no máximo cem vezes mais rápidos do que os actuais. Não será isto suficiente para desvalorizar a questão da eficiência dos algoritmos?

Nem pensar! A lei de Moore altera a constante que multiplica a estimativa do tempo de cálculo em função da dimensão do problema. Um algoritmo mais eficiente altera, como em todos os exemplos acima, o *expoente* da complexidade computacional! Suponhamos, por exemplo, que um algoritmo possui uma complexidade computacional $O(N^2)$. Isto significa que, se o problema B tiver uma dimensão mil vezes maior do que o problema A, demorará *um milhão de vezes mais tempo a ser resolvido.*

No entanto, se dispusermos de um outro algoritmo par mesmo problema com complexidade computacional O(N), o c muda radicalmente de figura. O problema B passa a ser ape mil vezes mais demorado do que o problema A. E o factor aceleração é tanto maior quanto maior a diferença de esc entre os problemas. Um bom algoritmo pode, de um dia pa outro, transformar problemas intratáveis em acessíveis e c literalmente novos domínios de investigação científica e tec lógica.

E isto é apenas a ponta do véu. O século XXI promete traz computação quântica, que provavelmente nos obrigará a fundir T ria da Computação, Física e Lógica. Ainda antes de existirem c putadores quânticos existem já algoritmos quânticos, como o factorização de inteiros de Peter Shor, à espera deles. Se ningu pode imaginar qual será o Top Ten dos algoritmos do século fica uma certeza: terá pouco que ver com o do século XX!