DOI:10.1145/2556329

Technical Perspective
A New Spin on an Old Algorithm

By Michael W. Mahoney

COMMUNICATION—THE COST of moving bits between levels of the memory hierarchy on a single machine or between machines in a network or data center—is often a more precious resource than computation. Although not new, communication-computation tradeoffs have received renewed interest in recent years due to architectural trends underlying high-performance computing as well as technological trends that permit the automatic generation of enormous quantities of data. On the practical side, this has led to multicore processors, libraries such as LAPACK and ScaLAPACK, schemes such as MPI and MapReduce, and distributed cloud-computing platforms. On the theoretical side, this has motivated a large body of work on new algorithms for old problems under new models of data access.

Into this fray enters the following paper by Ballard, Demmel, Holtz, and Schwartz, which considers a fundamental problem, adopting a new perspective on an old algorithm that has for years occupied a peculiar place in the theory and practice of matrix algorithms. In doing so, the work highlights how abstract ideas from theoretical computer science (TCS) can lead to useful results in practice, and it illustrates how bridging the theory-practice gap requires a healthy understanding of the practice.

The basic problem is the multiplication of two $n \times n$ matrices. This is a fundamental primitive in numerical linear algebra (NLA), scientific computing, machine learning, and large-scale data analysis. Clearly, n^2 time is a trivial lower bound—that much time is necessary to read the input and write the output. Moreover, at first glance, it seems "obvious" the ubiquitous three-loop algorithm for multiplying two matrices (given as input two $n \times n$ matrices, A and B, for each i, j, k, do: $C(i, j) += A(i, k) * B(k, j)$) shows that a constant times n^3 time is needed to solve the problem.

Back in 1969, it was surprising when Strassen presented his by-now well-known algorithm. The basic idea is two 2×2 matrices can be multiplied using 7, rather than the usual 8, multiplications. Since the same idea applies to 2×2 block matrices, the natural recursive extension can be used to multiply two $n \times n$ matrices in no more than a constant times n^ω arithmetic operations, where $\omega = \log_2 7 \approx 2.808$. Over the years, the exponent ω has been whittled down to $\omega \approx 2.373$, and many conjecture that there exist Strassen-like algorithms with $\omega = 2$.

Strassen's algorithm highlights the distinction, extremely important in TCS, between problems and algorithms; and it demonstrates that non-obvious algorithms can have better running times, in theory at least, than the obvious algorithm. Although its running time can be better than the usual three-loop algorithm for input matrices larger than ca. 100×100, Strassen's algorithm has, for both technical and non-technical reasons, yet to be widely used in practice.

This paper is part of a larger body of work on minimizing communication in NLA algorithms. Previous work has shown that geometric embedding methods can be used to establish communication lower bounds for three-loop matrix multiplication algorithms in both shared-memory sequential and distributed-memory parallel models. Basically, the algorithm can be modeled as a computation directed acyclic graph (CDAG). Due to the three-loop structure of the algorithm, this graph can be embedded into a 3D cube; and from the isoperimetric properties of that embedding a lower bound on communication can be established. The main result of this paper is a new lower bound on the amount of communication for both sequential and parallel versions of Strassen-like algorithms that is lower than the lower bound of the usual three-loop algorithm.

Since the geometric embedding methods do not seem to apply to the recursive structure of Strassen-like algorithms, the new lower bound is established by considering the edge expansion of the CDAG of Strassen's algorithm. Expanders—graphs that do not have any good partitions and that do not embed well in any low-dimensional Euclidean space—are remarkably useful structures that are ubiquitous within TCS and almost unknown outside TCS. For readers familiar with expanders, this paper will provide yet another application. For readers not familiar with expanders, this paper should be a starting point.

Finally, in a stroke that will make practitioners of numerical analysis and data analysis—as well as lower bound complexity theorists—happy, the authors also show their lower bounds are tight by providing an optimal algorithm. In the sequential case, this is attained by the standard implementation of Strassen's algorithm; and, in the parallel case, the authors, in joint work with Benjamin Lipshitz, have developed a novel Communication Avoiding Parallel Strassen algorithm. This latter algorithm communicates asymptotically less than previous three-loop and Strassen-based algorithms; and its empirical performance exceeds all other known matrix multiplication algorithms, three-loop or Strassen-based, on large parallel machines. Remarkably, this suggests that Strassen's algorithm should be adopted into existing parallel NLA libraries, providing a great example of how to bridge the theory-practice gap, and suggesting that Strassen's algorithm might still see practical use—ironically, though, due to its better communication properties. C

Michael W. Mahoney (mmahoney@icsi.berkeley.edu) is at the International Computer Science Institute and the Department of Statistics at the University of California at Berkeley.