Research Statements

Shang-Hua Teng

Over the past two decades, I have been working on problems at the intersection of Algorithm Design, Optimization, Algorithmic Game and Economics Theories, Scientific Computation, and Parallel Processing. I am a theoretican who cares a lot about the practical impact of mathematical and algorithmic work and who appreciates interdisciplinary research.

My main research interests include Smoothed Analysis of Algorithms and Heuristics, Spectral Graph Theory, Computational Game Theory, Combinatorial Scientific Computing, Combinatorial Optimization, Mathematical Programming, Parallel Scientific Computing, Graph Partitioning and Data Mining, Computational Geometry with Applications in Computer Graphics and Mesh Generation, and Graph Embedding.

My secondary interests include Cryptography and Computer Security, Internet Algorithms and Software, On-Line Scheduling, Cache-Oblivious Algorithms, VLSI and Circuit Simulation, Large-scale Information Processing and Organization, Compiler Optimization, String Matching, Regression and Robust Statistics, Percolation and Phase Transition, and Bioinformatics.

Although these topics above appear to be diverse, the underlying principle of my research has been the same, that is, to understand the mathematical structure of these problems in order to design efficient algorithms and software.

In addition to algorithm design and analysis, I have been very active in industry participation and collaboration with engineers and scientists who develop real-world products. In the past decades, I have contributed to software products in Thinking Machine, NASA, Intel, IBM, and Akamai, including transistor-level logic simulation software, numerical simulation, internet searching, network routing and large-scale data analysis. I have received 8 patents for my work.

Algorithm Analysis, Complexity, and Mathematical Programming

Daniel Spielman and I developed the theory of Smoothed Analysis to explain why some algorithms and heuristics work well in practice in spite of having poor worst-case complexity. We proved that the smoothed time complexity of the simplex method for linear programming is polynomial despite its exponential worst-case complexity.

Our work was quoted as one of the three significant accomplishments funded by the computer science division of the NSF in the National Science Fundation Summary of FY 2003 Budget Request to Congress and covered in Technology Review, by Megan Vandre.

The theory I developed with Spielman solved a long standing open question in Optimizations and Algorithm Analysis. In particular, it answered the following grand challenge posted in "Challenges for Theory of Computing: Report for an NSF-Sponsored Workshop on Research in Theoretical Computer Science", 1999, authored by Condon, Edelsbrunner, Emerson, Fortnow, Haber, Karp, Leivant, Lipton, Lynch, Parberry, Papadimitriou, Rabin, Rosenberg, Royer, Savage, Selman, Smith, Tardos, and Vitter.

> "While theoretical work on models of computation and methods for analyzing algorithms has had enormous payoff, we are not done. In many situations, simple algorithms do well. Take for example the Simplex algorithm for linear programming, or the success of simulated annealing of certain supposedly intractable problems. We don't understand why! It is apparent that worst-case analysis does not provide useful insights

on the performance of algorithms and heuristics and our models of computation need to be further developed and refined. Theoreticians are investing increasingly in careful experimental work leading to identification of important new questions in algorithms area. Developing means for predicting the performance of algorithms and heuristics on real data and on real computers is a grand challenge in algorithms".

Our work has since led many researchers to consider Smoothed Analysis in different algorithmic contexts such as in mathematical programming, large-scale data-clustering, local search, combinatorial optimization, computational geometry, and theory of random graphs and 0-1 matrices.

In numerical analysis, with Arvind Sankar and Dan Spielman, we proved that Gaussian elimination with partial pivoting requires only a logarithmic number of bits in the smoothed model. Our result matches the experimental observation about the needed precision of Gaussian elimination in solving linear systems. It may partially answer the long-term fundamental question in numerical analysis, as posted by Nick Trefethen in his 1993 Essay "The Definition of Numerical Analysis":

> "I have mentioned that Ax = b has the unusual feature that it can be solved in a finite sequence of operations. In fact Ax = b is more unusual than that, for the standard algorithm for solving it, Gaussian elimination, turns out to have extraordinarily complicated stability properties. Von Neumann wrote 180 pages of mathematics on this topic; Turing wrote one of his major papers; Wilkinson developed theory that grew into two books and a career. Yet the fact remains that for certain n by n matrices, Gaussian elimination with partial pivoting amplifies rounding errors by a factor of order 2n, making it a useless algorithm in the worst case. It seems that Gaussian elimination works in practice because the set of matrices with such behavior is vanishingly small, but to this day, nobody has a convincing explanation of why this should be so."

Algorithmic Game and Economics Theories

With Xi Chen and Xiaotie Deng, I solved two major open questions in Algorithmic Game Theory. We proved that the two-player Nash equilibrium does not have a fully polynomial-time approximation scheme, unless PPAD is in P, where PPAD (Polynomial Parity Argument, Directed version) is a complexity class introduced by Papadimitriou in 1991 to capture the difficulty of fixed-point computation. We also proved that the smoothed complexity of the classic Lemke-Howson algorithm, and in fact, of any algorithm for computing a Nash equilibrium in a two-person game is not polynomial, unless PPAD is in RP. Li-Sha Huang and I then extended the result to the problem of finding a market equilibrium in a Leontief exchange economies.

Very recently, Chen and I gave a tight bound of $(\Omega(n))^{d-1}$ on the randomized query complexity for computing a fixed point of a discrete Brouwer function over $[1:n]^d$, the d dimensional grid graph. This result contrasts interestingly with Aldous's elegant finding that randomization can speed up local search over $[1:n]^d$ and reduce the number of queries from $\Theta(n^{d-1})$ to $O(d^{1/2}n^{d/2})$. Our result may lead to further understandings of fixed-point and equilibrium computations.

Nearly-Linear Time Graph Algorithms

With Spielman, I developed the first nearly linear-time algorithm for spectral partitioning, graph sparsification, and local clustering. Our local clustering algorithm has several desirable features for processing very large-scale graphs, such as web graphs. The most important is its ability to compute a sparse cut and hence produces a well-connected cluster in time nearly linear in the size of this cluster, rather than in the size of the whole graph.

With Michael Elkin, Yuval Emek, and Spielman, we solved an open question of Alon, Karp, Peleg, West of 1992 on graph embedding. We proved that every weighted connected graph has a subgraph spanning tree of average stretch $O(\log^2 n \log\log n)$. We also developed a nearly linear-time algorithm for construting the low-stretch spanning tree.

Combinatorial Scientific Computing

Spielman and I developed the first nearly linear-time algorithm for preconditioning and solving symmetric diagonally-dominant linear systems. Our work solved an open question posted by Pravin Vaidya in 1990. Our result also implies that the Fiedler vector of a graph, the eigenvector associated with the second smallest eigenvalue of the Laplacian of the graph, can be approximated in nearly linear time.

Mesh Generation

The three-dimensional mesh generation problem is one of the most fundamental and difficult problems in numerical simulation. With Cheng, Dey, Edelsrunner, and Facello, I introduced a technique called sliver exudation which is the first topological approach for removing slivers from three-dimensional Delaunay meshes. Based on this technique, Xiang-Yang Li and I developed the first provably good Delaunay meshing algorithms for general three dimensional domains, solving a long term open question in mesh generation. Our algorithms extended the earlier work of Delaunay refinement pioneered by Paul Chew, Jim Ruppert, and Jonathan Shewchuk and resolved their major technical challenge, the removal of slivers. Software based on this development was used at the University of Illinois for the simulation of advanced rockets. I also developed the sphere-packing based Delaunay and Voronoi meshing and coarsening algorithm in two and three dimensions with Gary Miller, Dafna Talmor, and Noel Walkington. Spielman and Alper Ungor and I designed one of the first provably good parallel Delaunay meshing algorithms.

Spectral Graph Theory and Graph Partitioning

An encouraging interdisciplinary interaction with Horst Simon, the current director of the NERSC Division of Lawrence Berkeley National Laboratory, helped to jump-start my work with Dan Spielman on spectral graph theory. We obtained a mathematical proof of why spectral partitioning methods work in practice, solving a major challenge in combinatorial optimization and numerical linear algebra. Our work also established a new connection between the geometry of graph and its eigenvalues. In particular, we proved that from the second eigenvector of a planar graph, we can compute, in linear time, a partition whose the quality is as good as that can be achieved by the classic Lipton-Tarjan's planar separator algorithm. Our result also shows that the Fiedler value of a bounded-degree planar graph of n vertices in $O(1/n)$.

Geometric Graph Partitioning and N-Body Simulation

With Gary Miller, William Thurston, and Steve Vavasis, I pioneered the geometric mesh partitioning techniques for finite-element computation in three dimensions. The partitioning algorithm that we developed is the first provably good algorithm for partitioning finite-element meshes in three dimensions. With John Gilbert, I built a Matlab mesh partitioning toolbox based on this theoretical algorithm. Since then various extensions of this software has been used and incorporated in practical software at NASA Ames Research Center and CMU. Building upon this method, I also developed the first provably the first optimal load balancing scheme for non-uniform N-body simulation in three dimensions.

Computational Geometry and Robust Statistics

Alan Frieze, Miller, and I developed the first randomized $O(\log n)$ time, n-processor algorithm for computing the k-nearest neighborhood graphs in any fixed dimensions. Shortly after, Marshall Bern, David Eppstein and I found the first optimal parallel algorithm for generating quadtrees and nearest neighborhood graphs.

Nina Amenta, Bern, Eppstein, and I proved the Rousseeuw-Hubert conjecture on regression depth, providing an important theorem for Robust Statistics and Inference.