Exploiting Real-World Data to Accelerate Computational Science

Darren Strash
Visiting Assistant Professor
Department of Computer Science
Colgate University
dstrash@colgate.edu

December 28, 2017

Viewpoint, Methodology, and Accessibility to Undergraduates

A major challenge in the Information Age is to efficiently process the unprecedented amount of data available to us. While commoditizing this data fuels many industries, what fuels my research is the fact that science and society receive priceless benefits from analyzing such data sets. To accelerate this analysis, I design *practical* algorithms for solving problems on very large graphs, with an emphasis on addressing fundamental computational challenges across the sciences. Many of these challenges center around solving problems that are provably hard, including graph partitioning, subgraph isomorphism, maximum clique, and minimum vertex cover problems. Algorithms for these problems have direct applications in computational biology (phylogenetic tree reconstruction), chemistry (molecular docking), and sociology (community detection); aside from the sciences, algorithms for these problems impact many fields of computer science, including computer graphics, AI, machine learning, data mining, and high throughput computing.

A key component of my research is heavy interdisciplinary collaboration with scientists in the field, and includes exchanging real-world data, as well as fundamental problems and their solutions. Following the algorithm engineering paradigm, I look for ways to exploit the structure of these data sets to obtain algorithms that are efficient in practice, even if these algorithms are considered inefficient in a theoretical computational model. Often, this leads to opportunities to further bridge the gap between theory and practice by improving theoretical analysis for algorithms on "structured" real-world data, which is not possible for arbitrary data.

Given its numerous applications (as well as the mix of theory and practice), my research is accessible to (and appealing to) students with different backgrounds, interests, and even experience levels. For those students who are mathematically-minded, I offer options for theorem-proof style research, and for those who are programming-focused I offer experimentally-driven, project-focused work. Along these lines, in the past year I have supervised eight students in non-thesis related projects, and in the last three years I have supervised five bachelor's theses (one more is in progress) and four master's theses. Three theses have been published in peer-reviewed conferences, with two more publications in progress; one of these received a best paper nomination at the top-tier ACM-sponsored evolutionary algorithms conference, GECCO 2017.

Early Work

In my doctoral work, I developed efficient algorithms for graphs with and without geometry (including shortest paths in road networks, routing, graph drawing, and subgraph listing and counting)

under an interdisciplinary project with computational sociologists and machine learning experts. While mostly theoretical in nature, these early results introduced tools to quickly analyze large networks with real-world structure and helped to establish my real-world viewpoint. Among many results, I designed the first practical (and near-optimal in theory) maximal clique listing algorithm for large sparse networks, which is being actively used by biologists, physicists and computational sociologists around the world. This algorithm is available in popular software packages such as R and NetworkX and earned a best paper award at the International Symposium on Experimental Algorithms. My early research also partially solved a long-standing open problem in computational geometry, that non-simple polygons can be triangulated in linear time, which was published at ACM-SIAM Symposium on Discrete Algorithms, a top-tier conference.

Overall, my early work led to nine conference publications, six of which were significant enough to be invited to special journal issues for their respective conferences.

Recent Work

Over time, my real-world viewpoint has led me to concentrate on algorithms that are practical with reasonable theoretical guarantees. In the last two years, I have begun to address the shortcomings of existing combinatorial optimization algorithms on large networks—which have wide-ranging applications. Beginning with the maximum independent set problem (used in computer vision, machine learning, and routing on road networks), I investigated how to combine inexact techniques, such as genetic and local search algorithms, with exact algorithms. In a series of results, I showed how to find true maximum independent sets hundreds of times faster than with exact methods alone and that these techniques can also find high-quality solutions on graphs that are infeasible to analyze in a timely manner otherwise. These algorithms can be used to analyze graphs with billions of nodes and edges, which are by far the largest graphs considered for this problem.

My investigation into the maximum independent set problem has resulted in five publications in conferences focusing on experimental algorithms and combinatorial optimization, including one publication in the highly-competitive ACM SIGSPATIAL conference, which is the premier conference for geographic information systems. Four of these papers were written with undergraduate and/or master's students, and two of these papers were the direct result of student theses.

The success of these methods for the maximum independent set problem has prompted me to pursue projects with applications spanning the sciences: for instance in structural biology (matching configurations of proteins), computational biology (phylogenetic tree reconstruction) and chemistry, which rely on solving computationally-heavy problems in parallel, by efficiently breaking them up into more manageable subproblems. These projects include scaling up subgraph isomorphism and graph partitioning (including separator computation) algorithms for large networks and has so far led to three publications in IEEE and ACM sponsored venues and a best paper nomination at the highly-competitive genetic algorithms conference, GECCO. Two of these publications were the direct result of student theses.

Ongoing and Future Work

My recent work pushes the boundaries on the size of the inputs that we can solve using exact combinatorial optimization algorithms; however these graphs are still small enough to fit in the memory of a single machine. A natural next step in my research is to design algorithms for even larger data sets. I am in the beginning stages of a long term research project to do just that: develop algorithms for *truly* massive data sets. Given the scale of these data sets, this research will consider both exact and inexact algorithms.

Scaling Up Combinatorial Optimization to Truly Massive Networks. In my current work with a master's student at Karlsruhe Institute of Technology, we are studying how to develop communication-efficient algorithms for generating massive networks in parallel—at the petabyte scale—which will enable researchers to test their algorithms on massive graphs with real-world properties. This work was just accepted at the IEEE-sponsored IPDPS conference—a premier conference on parallel computation.

With bachelor's students at Colgate University I am currently investigating how to scale up maximum clique computation and clique listing algorithms to massive networks. Preliminary results have been promising: we have been able to compute a maximum clique in large networks up to sixty times faster than previous methods.

In addition to exact algorithms, there is further potential to scale up inexact algorithms, which have high-quality in practice, to truly massive networks—those that must be processed with distributed computation or external memory algorithms. Even though many existing algorithms claim to work efficiently on "massive" networks, they typically only target networks that easily fit into the working memory of a single machine—and therefore do not fit the true definition of massive. Those few algorithms that do exist for massive networks typically use the MapReduce framework, which scales well but has slow running time overall. There are two dimensions to this problem that I plan to explore:

1. *Problem Subdivision for Massive Parallelism.* One way to enable high-performance computing is to partition the input problem into pieces that can be solved independently in parallel. Efficient algorithms for combinatorial optimization problems—in this context, called *combinatorial scientific computing*—can be used to for this purpose. A natural next step in my research is to develop efficient parameterized algorithms for graph partitioning and graph coloring, which can be used to partition massive real-world networks. Specific applications of these techniques include identifying graph features to analyze networks for data mining, social interactions, protein-protein interactions, and interaction data in physics and chemistry.

2. *Communication-Efficient Approaches.* Some problems cannot be partitioned into pieces that can be solved independently—in these cases we want to minimize the communication between processing elements. Using graph measure computation (e.g., diameter and betweenness centrality) as a primary problem, I will investigate how techniques from I/O-efficient algorithms can reduce the communication overhead for parallel algorithms. These problems have many applications in sociology, biology, and machine learning, and I believe research in this area will provide many opportunities for collaboration. Further, given the pervasive nature of similar problems in industry, I plan to actively pursue collaborations with companies such as Facebook and Google, who must process such large graphs on a continual basis.

Open Science

Although my main research goal is to produce high-quality results, I further prioritize making my academic output freely available for anyone who wants it. I post preprints of all of my articles on `http://www.arxiv.org`, an open repository for research articles. Further, I release any software that I write for experiments under the GNU GPL license, which requires any release of modified software to also release the modified code. All of my code is freely available on GitHub.[1]

[1] `https://github.com/darrenstrash`