Report from Dagstuhl Seminar 18241

High-Performance Graph Algorithms

Edited by

Henning Meyerhenke[1], Richard Peng[2], and Ilya Safro[3]

1 **HU Berlin, DE,** `meyerhenke@hu-berlin.de`
2 **Georgia Institute of Technology – Atlanta, US,** `rpeng@cc.gatech.edu`
3 **Clemson University, US,** `isafro@g.clemson.edu`

Abstract

This report documents the program and outcomes of Dagstuhl Seminar 18241 "High-performance Graph Algorithms". The seminar reflected the ongoing qualitative change how graph algorithms are used in practice due to (i) the complex structure of graphs in new and emerging applications, (ii) the size of typical inputs, and (iii) the computer systems with which graph problems are solved. This change is having a tremendous impact on the field of graph algorithms in terms of algorithm theory and implementation as well as hardware requirements and application areas.

The seminar covered recent advances in all these aspects, trying to balance and mediate between theory and practice. The abstracts included in this report contain and survey recent state-of-the-art results, but also point to promising new directions for high-performance graph algorithms and their applications, both from a theoretical and a practical point of view.

Seminar June 10–15, 2018 – http://www.dagstuhl.de/18241
2012 ACM Subject Classification Mathematics of computing → Discrete mathematicsMathematics of computing → Graph theory, Theory of computation → Design and analysis of algorithms, Theory of computation → Models of computation, Theory of computation → Randomness, geometry and discrete structures
Keywords and phrases algorithm engineering, combinatorial scientific computing, graph algorithms, high-performance computing, theoretical computer science
Digital Object Identifier 10.4230/DagRep.8.6.19
Edited in cooperation with Manuel Penschuck

1 Executive Summary

Henning Meyerhenke
Richard Peng
Ilya Safro

Many presentations in this Dagstuhl seminar emphasized recent trends regarding typical inputs and their effect on graph algorithm development. From a high-level perspective, one can divide the presentations into two categories: either more focused on algorithm theory or more focused on practical algorithmic results. Many talks considered both theoretical and practical aspects. Furthermore, attention was given to intermix talks with theoretical and practically motivated starting points in order to encourage discussions among attendees. We were happy to see such discussions, as well as synergy of both aspects, carrying over to working groups on open problems.

Theory-focused talks were given by Sachdeva, Nanongkai, Jacob, Mouatadid, Kyng, Tsourakakis, and Litvak. They considered numerous topics such as Laplacian solvers and related optimization techniques, dynamic graph algorithms, external-memory graph algorithms, graph decompositions, and generative models.

The talks with emphasis on practical performance can be further subdivided into three subclasses: (i) graph mining, network analysis and optimization, (ii) parallel, distributed and streaming graph algorithms and (iii) graph generation. The talks given by Koutra, Ahmed, Klymko, Angriman, Gleich and Schulz fall into the first subclass, with a wide variation of algorithmic problems under consideration. Likewise, it was interesting to see the variety in computing platforms and tools (for example shared memory, message passing, distributed systems, streaming from databases, GraphBLAS) used in the eight talks of subclass two, presented by Besta, Shun, Predari, Ramachandran, Pothen, Bader, Finocchi and Davis. Finally, the talks by Phillips, Sanders and Penschuck as well as Crescenzi dealt with generating very large graphs with properties also found in real-world graphs – which is important, among others, for convincing scaling studies in algorithm engineering.

2 Table of Contents

3 Overview of Talks

3.1 Sampling from Massive Graph Streams: A Unifying Framework

Nesreen K. Ahmed (Intel Labs – Santa Clara, US)

Joint work of Nesreen K. Ahmed, Nick G. Duffield, Theodore L. Willke, Ryan A. Rossi
Main reference Nesreen K. Ahmed, Nick G. Duffield, Theodore L. Willke, Ryan A. Rossi: "On Sampling from Massive Graph Streams", PVLDB, Vol. 10(11), pp. 1430–1441, 2017.
URL http://dx.doi.org/10.14778/3137628.3137651

The rapid growth of the Internet and the explosion in online social media has led to a data deluge. A growing set of online applications are continuously generating data at unprecedented rates, from the Internet of things (e.g., connected devices, routers), electronic communication (e.g., email, IMs), social media (e.g., blogs), to the vast collection of online social networks and content sharing applications (e.g., Facebook, Twitter). Graphs are a natural data representation in many of these application domains, where nodes represent individuals/entities and edges represent the interaction, communication, or connectivity among them. Consider interaction and activity networks formed from electronic communication between online users. These resulting interaction and activity networks manifest as a stream of edges, where edges (i.e., interactions) occur one at a time, carrying a wealth of behavioral, community, and relationship information. Many of these networks are massive in size, due to the prolific amount of activity data. To keep up with the growing pace of this data, we need efficient methods to analyze dynamic interaction networks as the data arrives in streams, rather than static snapshots of graphs. Sampling provides an attractive approach to quickly and efficiently find an approximate answer to a query, or more generally, any analysis objective. In this talk, I will discuss a novel adaptive general-purpose framework for sampling from graph streams, called graph priority sampling (GPS). From a high-volume stream of edges, our proposed framework maintains a generic sample of limited size that can be used at any time to accurately estimate the total weight of arbitrary graph subsets (i.e., triangles, cliques). To obtain accurate estimates of various graph properties, our proposed framework maintains a weight sensitive sample that devotes sampling resources to edges that are informative for those properties. Unbiasedness of subgraph estimators is established through a new Martingale formulation of graph stream sampling, in which subgraph estimators are unbiased, even when computed at different points in the stream. I will summarize the results of our large-scale experimental study on real-world graphs from various domains.

3.2 Centrality Computation in Real-World Networks: New Challenges

Eugenio Angriman (Universität Köln, DE)

Joint work of Patrick Bisenius, Elisabetta Bergamini, Eugenio Angriman, Henning Meyerhenke
Main reference Patrick Bisenius, Elisabetta Bergamini, Eugenio Angriman, Henning Meyerhenke: "Computing Top-k Closeness Centrality in Fully-dynamic Graphs", in Proc. of the Twentieth Workshop on Algorithm Engineering and Experiments, ALENEX 2018, New Orleans, LA, USA, January 7-8, 2018., pp. 21–35, SIAM, 2018.
URL http://dx.doi.org/10.1137/1.9781611975055.3

The efficient computation of the top-k most central nodes of a graph has been widely studied, especially for well-known centrality metrics such as Closeness, Katz, or Betweenness. We present several techniques that allow us to efficiently recompute the top-k nodes with highest Closeness Centrality after the insertion or removal of an edge in a network without requiring asymptotically more memory than the static algorithms. We also give a brief introduction to NetworKit, an open-source toolkit for large-scale network analysis with shared-memory systems. NetworKit includes a wide range of state-of-the-art graph algorithms implemented in C++, which can be easily used from a Python frontend. Finally, we introduce some open problems related to the extension of single-node centrality metrics to groups of nodes.

3.3 Massive-scale Graph Analytics: A New Algorithmic Model

David A. Bader (Georgia Institute of Technology – Atlanta, US)

Joint work of David A. Bader, E. Jason Riedy, Chunxing Yin

Emerging real-world graph problems include: detecting and preventing disease in human populations; revealing community structure in large social networks; and improving the resilience of the electric power grid. Unlike traditional applications in computational science and engineering, solving these problems at scale often raises new challenges because of the sparsity and lack of locality in the data, the need for research on scalable algorithms and development of frameworks for solving these real-world problems on high performance computers, and for improved models that capture the noise and bias inherent in the torrential data streams.

Focusing on parallel algorithm design and implementation, Bader formalizes a practical model for graph analysis on streaming data. In this model, a massive graph undergoes changes from an input stream of edge insertions and removals. The model supports concurrent updating of the graph while algorithms execute concurrently on the dynamic data structure. The talk introduces a concept of validity: an algorithm is valid if the output is correct for a graph consisting of the initial graph with some subset of concurrent changes. Practical examples of this model are given for valid implementations of breadth first search, connected components, PageRank, and triangle counting, all useful graph kernels in real-world applications.

3.4 Core-periphery clustering and complex networks

Pierluigi Crescenzi (University of Florence, IT)

Joint work of Pierluigi Crescenzi, Pierre Fraigniaud, Zvi Lotker, Paolo Penna
Main reference Pierluigi Crescenzi, Pierre Fraigniaud, Zvi Lotker, Paolo Penna: "Core-periphery clustering and collaboration networks", in Proc. of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2016, San Francisco, CA, USA, August 18-21, 2016, pp. 525–528, IEEE Computer Society, 2016.
URL http://dx.doi.org/10.1109/ASONAM.2016.7752285

In this talk we analyse the core-periphery clustering properties of complex networks, where the core of a network is formed by the nodes with highest degree. In particular, we first observe that, even for random graph models aiming at matching the degree-distribution and/or the clustering coefficient of real networks, these models produce synthetic graphs which have a spatial distribution of the triangles with respect to the core and to the periphery which does not match the spatial distribution of the triangles in the real networks. We therefore propose a new model, called CPCL, whose aim is to distribute the triangles in a way fitting with their real core-periphery distribution, and thus producing graphs matching the core-periphery clustering of real networks.

3.5 Scaling problems with metric constraints

David F. Gleich (Purdue University – West Lafayette, US)

Joint work of David F. Gleich, Nate Veldt, Anthony Wirth, James Saunderson
Main reference Nate Veldt, David F. Gleich, Anthony Wirth, James Saunderson: "A Projection Method for Metric-Constrained Optimization", CoRR, Vol. abs/1806.01678, 2018.
URL http://arxiv.org/abs/1806.01678

We evaluate techniques to solve LP formulations with metric constraints (triangle inequality constraints). We are able to solve these with a-posterori approximation guarantees for problems with 10000 datapoints (or roughly 700 billion constraints) on standard desktop hardware through the careful use of projection methods.

3.6 SuiteSparse:GraphBLAS: graph algorithms in the language of linear algebra

Timothy Alden Davis (Texas A&M University – College Station, US)

Main reference Timothy A. Davis: "Algorithm 9xx: SuiteSparse:GraphBLAS: graph algorithms in the lanuage of sparse linear algebra". Submitted to ACM Transactions on Mathematical Software, June 2018
URL http://suitesparse.com

SuiteSparse:GraphBLAS is a full implementation of the GraphBLAS standard, which defines a set of sparse matrix operations on an extended algebra of semirings using an almost unlimited variety of operators and types. When applied to sparse adjacency matrices, these algebraic operations are equivalent to computations on graphs. GraphBLAS provides a powerful and expressive framework for creating graph algorithms based on the elegant mathematics of sparse matrix operations on a semiring. Key features and performance of the SuiteSparse implementation of GraphBLAS package are described.

3.7 Beyond triangles

Irene Finocchi (Sapienza University of Rome, IT)

Joint work of Irene Finocchi, Marco Finocchi, Emanuele Guido Fusco
Main reference Irene Finocchi, Marco Finocchi, Emanuele G. Fusco: "Clique Counting in MapReduce: Algorithms and Experiments", ACM Journal of Experimental Algorithmics, Vol. 20, pp. 1.7:1–1.7:20, 2015.
URL http://dx.doi.org/10.1145/2794080

The talk addressed the problem of counting the number q_k of k-cliques in large-scale graphs, for any constant $k \geq 3$. This is essential in a variety of applications, including social network analysis. Due to the computationally intensive nature of the clique counting problem, we settle for exact/approximate parallel solutions in the MapReduce framework, discussing both theoretical and experimental contributions. In particular, our algorithms make it possible to compute q_k for several real-world graphs and shed light on its growth rate as a function of k.

3.8 The Parallel External Memory Model: Sparse Matrix Multipy and List Ranking

Riko Jacob (IT University of Copenhagen, DK)

Joint work of Michael Bender, Gerth Brodal, Rolf Fagerberg, Riko Jacob, Elias Vicari, Tobias Lieber, Nodari Sitchinava

The talk surveyed the parallel external memory model (PEM) and some fundamental results on permuting/sparse matrix multiply. It also gave an impression of a very peculiar lower bound for the list ranking problem, meant as an invitation to work on the open problem of generalizing the lower bound to a more standard setting.

References

1 Michael A. Bender, Gerth Stølting Brodal, Rolf Fagerberg, Riko Jacob, and Elias Vicari. Optimal sparse matrix dense vector multiplication in the I/O-model. *Theoretical Computer Science*, 47(4):934–962, 2010.

2 Riko Jacob, Tobias Lieber, and Nodari Sitchinava. On the complexity of list ranking in the parallel external memory model. In *Proceedings 39th International Symposium on Mathematical Foundations of Computer Science (MFCS'14)*, volume 8635 of *Lecture Notes in Computer Science*, pages 384–395. Springer, 2014.

3.9 Theoretically Efficient Parallel Graph Algorithms Can Be Fast and Scalable

Julian Shun (MIT – Cambridge, US)

Joint work of Laxman Dhulipala, Guy E. Blelloch, Julian Shun
Main reference Laxman Dhulipala, Guy E. Blelloch, Julian Shun: "Theoretically Efficient Parallel Graph Algorithms Can Be Fast and Scalable", CoRR, Vol. abs/1805.05208, 2018.
URL http://arxiv.org/abs/1805.05208

There has been significant interest in parallel graph processing recently due to the need to quickly analyze the large graphs available today. Many graph codes have been designed for distributed memory or external memory. However, today even the largest publicly-available real-world graph (the Hyperlink Web graph with over 3.5 billion vertices and 128 billion edges) can fit in the memory of a single commodity multicore server. Nevertheless, most experimental work in the literature report results on much smaller graphs, and the ones that use the Hyperlink graph are done in distributed or external memory. Therefore it is natural to ask whether we can efficiently solve a broad class of graph problems on this graph in memory.

With a graph of this size it is important to use theoretically-efficient parallel algorithms as even minor inefficiencies in the work or parallelism of an algorithm can lead to a significant increase in running time. This talk shows that theoretically-efficient parallel graph algorithms can scale to the largest publicly-available graphs using a single machine with a terabyte of RAM, processing them in minutes. We give implementations of theoretically-efficient parallel algorithms for 13 important graph problems. We also present the optimizations and techniques that we used in our implementations, which were crucial in enabling us to process these large graphs quickly. We show that the running times of our implementations outperform existing state-of-the-art implementations on the largest real-world graphs. For many of the problems that we consider, this is the first time they have been solved on graphs at this scale.

3.10 An Ensemble Framework for Detecting Community Changes in Dynamic Networks

Christine Klymko (LLNL – Livermore, US)

Joint work of Timothy La Fond, Geoffrey Sanders, Christine Klymko, Van Emden Henson
Main reference Timothy La Fond, Geoffrey Sanders, Christine Klymko, Van Emden Henson: "An ensemble framework for detecting community changes in dynamic networks", in Proc. of the 2017 IEEE High Performance Extreme Computing Conference, HPEC 2017, Waltham, MA, USA, September 12-14, 2017, pp. 1–6, IEEE, 2017.
URL http://dx.doi.org/10.1109/HPEC.2017.8091035

Dynamic networks, especially those representing social networks, undergo constant evolution of their community structure over time. Nodes can migrate between different communities, communities can split into multiple new communities, communities can merge together, etc. In order to represent dynamic networks with evolving communities it is essential to use a dynamic model rather than a static one. Here we use a dynamic stochastic block model where the underlying block model is different at different times. In order to represent the structural changes expressed by this dynamic model the network will be split into discrete time segments and a clustering algorithm will assign block memberships for each segment. We show that using an ensemble of clustering assignments accommodates for the variance in scalable clustering algorithms and produces superior results in terms of pairwise-precision and pairwise-recall. We also demonstrate that the dynamic clustering produced by the ensemble can be visualized as a flowchart which encapsulates the community evolution succinctly.

3.11 Scalable Inference and Summarization of Multi-source Network Data

Danai Koutra (University of Michigan – Ann Arbor, US)

Joint work of Danai Koutra, Tara Safavi, Chandra Sripada, U Kang, Jilles Vreeken, Neil Shah, Christos Faloutsos, Yujun Yan, Di Jin, Mark Heimann
Main reference Tara Safavi, Chandra Sripada, Danai Koutra: "Scalable Hashing-Based Network Discovery", in Proc. of the 2017 IEEE International Conference on Data Mining, ICDM 2017, New Orleans, LA, USA, November 18-21, 2017, pp. 405–414, IEEE Computer Society, 2017.
URL http://dx.doi.org/10.1109/ICDM.2017.50
Main reference Danai Koutra, U. Kang, Jilles Vreeken, Christos Faloutsos: "VOG: Summarizing and Understanding Large Graphs", in Proc. of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014, pp. 91–99, SIAM, 2014.
URL http://dx.doi.org/10.1137/1.9781611973440.11
Main reference Neil Shah, Danai Koutra, Tianmin Zou, Brian Gallagher, Christos Faloutsos: "TimeCrunch: Interpretable Dynamic Graph Summarization", in Proc. of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015, pp. 1055–1064, ACM, 2015.
URL http://dx.doi.org/10.1145/2783258.2783321
Main reference Yujun Yan, Mark Heimann, Di Jin, Danai Koutra: "Fast Flow-based Random Walk with Restart in a Multi-query Setting", in Proc. of the 2018 SIAM International Conference on Data Mining, SDM 2018, May 3-5, 2018, San Diego Marriott Mission Valley, San Diego, CA, USA., pp. 342–350, SIAM, 2018.
URL http://dx.doi.org/10.1137/1.9781611975321.39

Networks naturally capture a host of real-world interactions, from social interactions and email communication to brain activity. However, graphs are not always directly observed, especially in scientific domains, such as neuroscience, where monitored brain activity is often captured as time series. How can we efficiently infer networks from time series data (e.g., model the functional organization of brain activity as a network) and speed up the network construction process to scale up to millions of nodes and thousands of graphs? Further, what can be learned about the structure of the graph data? How can we automatically summarize a network 'conditionally' to its domain, i.e., summarize its most important properties by taking into account the properties of other graphs in that domain (e.g., neuroscience, social science)? In this talk I will present our recent work on scalable algorithms for inferring, summarizing and mining large collections of graph data coming from different sources. I will also discuss applications in various domains, including connectomics and social science.

3.12 Optimization on Graphs

Rasmus Kyng (Harvard University – Cambridge, US)

Many of our favorite questions about graphs are answered by solving optimization problems. For example, we can analyze social networks using clustering and regression, and this leads to problems that are solved using optimization techniques. Similarly, planning flows of goods or data in transportation networks boils down to solving optimization problems. Second order methods are a powerful tool in optimization, but they require solving linear equations, which can be prohibitively expensive. However, when the optimization problem comes from a graph, this adds structure to the linear equations. We can leverage this structure to solve the equations quickly, making second order methods tractable.

3.13 Power-law hypothesis for PageRank

Nelly Litvak (University of Twente, NL)

License Creative Commons BY 3.0 Unported license
© Nelly Litvak
Joint work of Nelly Litvak, Remco van der Hofstad, Alessandro Garavaglia, Ninguyan Chen, Mariana Olvera-Cravioto
Main reference Ningyuan Chen, Nelly Litvak, Mariana Olvera-Cravioto: "Generalized PageRank on directed configuration networks", Random Struct. Algorithms, Vol. 51(2), pp. 237–274, 2017.
URL http://dx.doi.org/10.1002/rsa.20700
Main reference Garavaglia, A., van der Hofstad, R., & Litvak: "Local weak convergence for PageRank". arXiv preprint arXiv:1803.06146.
URL https://arxiv.org/abs/1803.06146

PageRank is a well-known algorithm for measuring centrality in networks, originally proposed by Google for ranking pages in the World-Wide Web. One of the intriguing properties of PageRank is the so-called "power-law hypothesis": in a scale-free network the PageRank follows a power law with the same exponent as (in-)degrees. Up to date, this hypothesis has been confirmed many times empirically. However, formalizing and proving this result mathematically turns out to be challenging. In this talk I will discuss recent progress on limiting behavior of PageRank in random graphs. I will first present results on the power-law behavior for PageRank was obtained in the directed configuration model with independent in- and out-degrees. This result follows essentially from the coupling of the graph with a branching tree. I will continue with recent work, where we take a different approach. Instead of focusing on a particular random graph model, we investigate the asymptotic PageRank distribution when the graph size goes to infinity, using the notion of local weak convergence recently developed in the theory of random graphs. To this end, we define an exploration process in the directed setting that keeps track of in- and out-degrees of vertices. Then we use this to prove the existence of the asymptotic PageRank distribution. As a result, the limiting distribution of PageRank can be computed directly as a function of the limiting object. As examples, we apply our results in the directed configuration model with dependent in- and out-degrees, continuous-time branching processes trees, and preferential attachment models.

3.14 To Push or To Pull: On Reducing Communication and Synchronization in Graph Computations

Maciej Besta

License Creative Commons BY 3.0 Unported license
© Maciej Besta
Joint work of Maciej Besta, Michal Podstawski, Linus Groner, Edgar Solomonik, Torsten Hoefler
Main reference Maciej Besta, Michal Podstawski, Linus Groner, Edgar Solomonik, Torsten Hoefler: "To Push or To Pull: On Reducing Communication and Synchronization in Graph Computations", in Proc. of the 26th International Symposium on High-Performance Parallel and Distributed Computing, HPDC 2017, Washington, DC, USA, June 26-30, 2017, pp. 93–104, ACM, 2017.
URL http://dx.doi.org/10.1145/3078597.3078616

We reduce the cost of communication and synchronization in graph processing by analyzing the fastest way to process graphs: pushing the updates to a shared state or pulling the updates to a private state. We investigate the applicability of this push-pull dichotomy to various algorithms and its impact on complexity, performance, and the amount of used locks, atomics, and reads/writes. We consider 11 graph algorithms, 3 programming models, 2 graph abstractions, and various families of graphs. The conducted analysis illustrates surprising differences between push and pull variants of different algorithms in performance, speed of convergence, and code complexity; the insights are backed up by performance data from hardware counters. We use these findings to illustrate which variant is faster for each algorithm and to develop generic strategies that enable even higher speedups. Our insights can be used to accelerate graph processing engines or libraries on both massively-parallel shared-memory machines as well as distributed-memory systems

3.15 A Toolbox to Extract Structure From Graphs

Lalla Mouatadid (University of Toronto, CA)

We survey a few techniques to extract structural properties on certain graph classes. We focus in particular on two techniques: Graph searching and modular decomposition, and illustrate these ideas on certain graph classes. We then introduce a relaxation of modular decomposition (ϵ-modules) to try to lift these techniques to less structured graphs.

3.16 Dynamic Algorithms and Complexity, A Short Survey

Danupon Nanongkai (KTH Royal Institute of Technology – Stockholm, SE)

The purpose of this talk is to give people in other fields a taste of the dynamic graph algorithms research in the theory (FOCS/STOC) community. The talk is a compression of a survey given last week at HALG 2018. The original abstract is below.

In this talk I will attempt to answer the following questions I have been asked quite often: What are the current challenges in dynamic graph algorithms? What are good starting points for PhD students and experienced researchers from other fields, who want to try working in this field? The talk will focus on challenges for basic graph problems (e.g. connectivity, shortest paths, maximum matching), and will survey some existing upper and lower bound results and techniques.

3.17 Generating random massive graphs that mimic real data

Cynthia A. Phillips (Sandia National Labs – Albuquerque, US)

We introduce wrapped BTER, a way to take a single input to the Block Two-Level Erdős-Renyi (BTER) scalable graph generator and produce a family of related graphs which vary by tightness of connection within BTER affinity blocks. This gives an alternative to the Lancichinetti-Fortunato-Radicchi (LFR) community detection benchmark capable of generating billion-edge graphs in less than a minute. We are currently extending this to generate graphs that have the same density, but are scaled up or down in size. We will present the latest results on scaling generation or describe the currently blocking challenges. We feel this technique will provide an important new way to generate realistic data sets for experimental analysis on huge graphs when there is not a lot of real data, or it is millions of nodes instead of billions.

3.18 When Exact Fails to be Parallel: Designing Parallel Algorithms through Approximation

Alex Pothen (Purdue University – West Lafayette, US)

Joint work of Arif Khan, S M Ferdous, Alex Pothen

We describe a paradigm for designing parallel algorithms that employs approximation techniques. Instead of solving a problem exactly, for which efficient parallel algorithms might not exist, we seek a solution with provable approximation guarantees via approximation algorithms. Furthermore, we design approximation algorithms with high degrees of concurrency. We show b-edge covers and b-matchings in graphs as examples of this paradigm.

For b-edge covers, we describe four techniques for designing approximation algorithms that are concurrent. The first is the Greedy algorithm, the second is the use of a primal-dual formalism, and the third and fourth reduce edge cover computations to computing matchings. We show that some of these algorithms obtain high performance on both shared memory and distributed memory computers. We also describe an application of matchings and edge covers to a problem in data privacy.

References

1 Arif Khan, Alex Pothen and S M Ferdous, *Designing Parallel Algorithms via Approximation: b-Edge Cover*, Proceedings of IPDPS, 12 pp., 2018.
2 S M Ferdous, Alex Pothen and Arif Khan, *New Approximation Algorithms for Minimum Weighted Edge Cover,* Proceedings of SIAM Workshop on Combinatorial Scientific Computing, 12 pp., 2018.
3 Arif Khan, Alex Pothen, Mostofa Patwary, Nadathur Satish, Narayanan Sunderam, Fredrik Manne, Mahantesh Halappanavar and Pradeep Dubey, *Efficient approximation algorithms for weighted b-Matching*, SIAM J. Scientific Computing, 38(5), S593-S619, 2016.

3.19 Topology-induced Enhancement of Mappings

Maria Predari (Universität Köln, DE)

Joint work of Maria Predari, Henning Meyerhenke, Roland Glantz
Main reference Roland Glantz, Maria Predari, Henning Meyerhenke: "Topology-induced Enhancement of Mappings", CoRR, Vol. abs/1804.07131, 2018.
URL http://arxiv.org/abs/1804.07131

We propose a new method to enhance a mapping $\mu(\cdot)$ of a parallel application's computational tasks to the processing elements (PEs) of a parallel computer. The idea behind our method TiMEr is to enhance such a mapping by drawing on the observation that many topologies take the form of a partial cube. This class of graphs includes all rectangular and cubic meshes, any such torus with even extensions in each dimension, all hypercubes, and all trees.

Following previous work, we represent the parallel application and the parallel computer by graphs G_a and G_p, respectively. G_p being a partial cube allows us to label its vertices, the PEs, by bitvectors such that the cost of exchanging one unit of information between any two vertices of G_p amounts to the Hamming distance between their labels. By transferring these bitvectors to the vertex set of G_a via $\mu^{-1}(\cdot)$ and extending them to be unique on G_a, we can enhance $\mu(\cdot)$ by swapping labels on G_a in a new way. Pairs of swapped labels are local w.r.t. the PEs, but not w.r.t. G_a. Moreover, permutations of the bitvectors' entries give rise to a plethora of hierarchies on the PEs. Through these hierarchies we turn TiMErinto a hierarchical method for improving $\mu(\cdot)$ that is complementary to state-of-the-art methods for computing $\mu(\cdot)$ in the first place.

In our experiments we use TiMErto enhance mappings of complex networks onto rectangular meshes and tori with 256 and 512 nodes, as well as hypercubes with 256 nodes. It turns out that common quality measure of mappings derived from state-of-the-art tools, such as scotch and KaHIP, can be improved up to 34 %.

3.20 A Round-Efficient and Communication-Efficient Algorithm for Distributed Betweenness Centrality

Vijaya Ramachandran (University of Texas – Austin, US)

License Creative Commons BY 3.0 Unported license
© Vijaya Ramachandran
Joint work of Roshan Dathathri, Gurbinder Gill, Loc Hoang, Keshav Pingali, Matteo Pontecorvi, Vijaya Ramachandran
Main reference Matteo Pontecorvi, Vijaya Ramachandran: "Distributed Algorithms for Directed Betweenness Centrality and All Pairs Shortest Paths", CoRR, Vol. abs/1805.08124, 2018.
URL http://arxiv.org/abs/1805.08124

Betweenness centrality is a graph computation that has many uses in the analysis of networks. Network graphs today consist of billions of nodes and trillions of edges and do not fit in the memory of a single machine, so distributed algorithms must be used to compute betweenness centrality. However, distributed algorithms are more difficult to implement efficiently than their shared-memory counterparts because the cost of synchronization is much higher.

In this talk, we present a round-efficient and communication-efficient algorithm for betweenness centrality and describe its implementation in D-Galois, a state-of-the-art distributed graph analytics framework. Our efficient distributed betweenness centrality algorithm for unweighted directed graphs runs in $2n + \mathcal{O}(\mathcal{D})$ rounds in the Congest model, where n is the number of nodes and D is the diameter of the graph. We adapt and implement this algorithm in D-Galois by further optimizing our Congest distributed algorithm to substantially reduce communication and synchronization costs in the D-Galois execution model. Preliminary experiments with large graphs on big clusters show that this new distributed-memory algorithm is substantially faster than several prior implementations including a state-of-the-art combinatorial Blas implementation.

3.21 Fast Approximate Gaussian Elimination for Laplacians

Sushant Sachdeva (University of Toronto, CA)

License 🅭 Creative Commons BY 3.0 Unported license
© Sushant Sachdeva
Joint work of Rasmus Kyng, Sushant Sachdeva
Main reference Rasmus Kyng, Sushant Sachdeva: "Approximate Gaussian Elimination for Laplacians – Fast, Sparse, and Simple", in Proc. of the IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA, pp. 573–582, IEEE Computer Society, 2016.
URL http://dx.doi.org/10.1109/FOCS.2016.68

Solving systems of linear equations in graph Laplacians is a fundamental primitive in scientific computing and optimization. Starting with the seminal work of Spielman-Teng that gave the first nearly-linear time algorithm for solving Laplacian systems, there has been a long line of work giving faster Laplacian solvers. These solvers have had a large impact on the design of fast graph algorithms.

I'll present a very simple, nearly-linear time Laplacian solver that is based purely on random sampling, and does not use any graph theoretic constructions such as low-stretch trees, sparsifiers, or expanders. Our solver builds a sparse Cholesky factorization for Laplacians – the symmetric version of Gaussian elimination. More precisely, it approximates a Laplacian L as $U'U$, where U is a sparse upper triangular matrix. Since triangular matrices are easy to invert, this immediately implies a fast Laplacian solver via iterative refinement.

3.22 Massively parallel communication-free graph generators

Peter Sanders (KIT – Karlsruher Institut für Technologie, DE) and
Manuel Penschuck (Goethe-Universität – Frankfurt, DE)

License 🅭 Creative Commons BY 3.0 Unported license
© Peter Sanders and Manuel Penschuck
Joint work of Daniel Funke, Sebastian Lamm, Ulrich Meyer, Manuel Penschuck, Peter Sanders, Christian Schulz, Darren Strash, Moritz von Looz
Main reference Daniel Funke, Sebastian Lamm, Peter Sanders, Christian Schulz, Darren Strash, Moritz von Looz: "Communication-Free Massively Distributed Graph Generation", in Proc. of the 2018 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2018, Vancouver, BC, Canada, May 21-25, 2018, pp. 336–347, IEEE Computer Society, 2018.
URL http://dx.doi.org/10.1109/IPDPS.2018.00043
Main reference Manuel Penschuck: "Generating Practical Random Hyperbolic Graphs in Near-Linear Time and with Sub-Linear Memory", in Proc. of the 16th International Symposium on Experimental Algorithms, SEA 2017, June 21-23, 2017, London, UK, LIPIcs, Vol. 75, pp. 26:1–26:21, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.SEA.2017.26

Analyzing massive complex networks yields promising insights about our everyday lives. Building scalable algorithms to do that is a challenging task that requires a careful analysis and extensive evaluation. However, engineering such algorithms is often hindered by the scarcity of publicly available datasets. Network generators serve as a tool to alleviate this problem by providing synthetic instances with controllable parameters.

We present efficient distributed algorithms for generating massive graphs for several popular models of random graphs. By making use of pseudorandomization and divide-and-conquer schemes, our generators follow a communication-free paradigm in the sense that the amount of communication necessary between the processors does not depend on the output size. The resulting generators are often embarrassingly parallel and have a near optimal scaling behavior.

Currently supported models include Erdős-Reny (directed/undirected), Barabasi-Albers scale-free, random geometric (2D/3D), Delaunay triangulations of random point sets (2D/3D), and random hyperbolic.

3.23 Practical Kernelization

Christian Schulz (Universität Wien, AT)

Joint work of Christian Schulz, Sebastian Lamm, Alexander Noe, Peter Sanders, Darren Strash

Many NP-hard graph problems have been shown to be fixed-parameter tractable (FPT): large inputs can be solved efficiently and provably optimally, as long as some problem parameter is small. Here the parameter measures the 'difficulty' of the input in some mathematically well-defined way, for example using the treewidth of the underlying graph. Over the last two decades, significant advances have been made in the design and analysis of FPT algorithms for a wide variety of graph problems. This has resulted in a rich algorithmic toolbox that are by now well-established and are described in several textbooks and surveys. They lead to algorithms that are theoretically efficient: they allow problems of size n with a parameter value of k to be solved in time $f(k)n^c$ for some (exponential) function f and constant c, thereby restricting the exponential dependence of the running time to the parameter k only. However, these theoretical algorithmic ideas have received very little attention from the practical perspective. Few of the new techniques are implemented and tested on real datasets, and their practical potential is far from understood. The rich toolbox of parameterized algorithm theory offers a rich set of algorithmic ideas that are challenging to implement and engineer in practical settings. In this talk, we survey our recent progress in applying reductions/kernelization routines to graph problems in order to obtain algorithms that run fast in practice. Specifically, we look at recent results for the independent set and the minimum cut problem.

3.24 Clustering with a faulty oracle

Charalampos E. Tsourakakis (Harvard University – Cambridge, US)

Joint work of Charalampos E. Tsourakakis, Michael Mitzenmacher, Kasper Green Larsen, Jaroslaw Blasiok, Preetum Nakkiran, Ben Lawson, Vasilis Nakos
Main reference Charalampos E. Tsourakakis, Michael Mitzenmacher, Jaroslaw Blasiok, Ben Lawson, Preetum Nakkiran, Vasileios Nakos: "Predicting Positive and Negative Links with Noisy Queries: Theory & Practice", CoRR, Vol. abs/1709.07308, 2017.
URL http://arxiv.org/abs/1709.07308

Social networks and interactions in social media involve both positive and negative relationships. Signed graphs capture both types of relationships: positive edges correspond to pairs of "friends", and negative edges to pairs of "foes". The *edge sign prediction problem* aims to predict whether an interaction between a pair of nodes will be positive or negative. We provide theoretical results for this problem that motivates natural improvements to recent heuristics for the problem on practical networks.

On the theoretical side, we model the edge sign prediction problem as follows: we are allowed to query any pair of nodes whether they belong to the same cluster or not, but the answer to the query is corrupted with some probability $0 < q < \frac{1}{2}$. Let $\delta = 1 - 2q$ be the bias. We provide an algorithm that recovers all signs correctly with high probability in the presence of noise with $O(\frac{n \log n}{\delta^2} + \frac{\log^2 n}{\delta^6})$ queries. This is the best known result for this problem for all but tiny δ, and improves the recent work of Mazumdar and Saha. Our result naturally generalizes to the case of k clusters as well. We also provide an algorithm that performs $O(\frac{n \log n}{\delta^4})$ queries, and uses breadth first search as its main algorithmic primitive. While both the running time and the number of queries are sub-optimal, our result relies on novel theoretical techniques, and naturally suggests the use of edge-disjoint paths as a feature for predicting signs in online social networks. Specifically, we use edge disjoint s-t paths of short length as a feature for predicting the sign of edge (s, t) in real-world signed networks. Empirical findings suggest that the use of such paths improves the classification accuracy, especially for pairs of nodes with no common neighbors.

4 Working groups

4.1 Time segmentation working group

David A. Bader (Georgia Institute of Technology – Atlanta, US), Irene Finocchi (Sapienza University of Rome, IT), George Karypis (University of Minnesota – Minneapolis, US), Michel A. Kinsy (Boston University, US), Christine Klymko (LLNL – Livermore, US), Danai Koutra (University of Michigan – Ann Arbor, US), Cynthia A. Phillips (Sandia National Labs – Albuquerque, US), and Ilya Safro (Clemson University, US)

This working group explored methods for detecting changing community structure within graph data presented as an edge-stream where edges have time stamps. Within this framework, we divided the problem into two different but related subproblems and discussed each of them, as described below:

1. Burn-in: given the start of a data stream of time-stamped edges, how can one determine at what point they have observed enough data to make running a community detection algorithm reasonable (or any other algorithm concerning higher-order graph properties). Are there other, easier/more concrete metrics that could be used as a proxy?
2. Change detection: assuming one has a given community structure at the start of a data stream, when should one re-partition the network? Due to the fact that many community detection algorithms are computationally expensive, the goal would be to minimized the number of times this repartitioning is done without missing key changes in network structure.
   a. Could a reconstruction-error based approach which projects the network onto a low-rank model and determines the probability of incoming edges fitting the model work?
   b. Could we develop a function f(E) which triggers a repartitioning when it reaches a certain threshold?

4.2 Dynamic Graphs Working Group

Richard Peng (Georgia Institute of Technology – Atlanta, US), Nesreen K. Ahmed (Intel Labs – Santa Clara, US), Rob Bisseling (Utrecht University, NL), George Karypis (University of Minnesota – Minneapolis, US), Danupon Nanongkai (KTH Royal Institute of Technology – Stockholm, SE), Manuel Penschuck (Goethe-Universität – Frankfurt, DE), Alex Pothen (Purdue University – West Lafayette, US), Vijaya Ramachandran (University of Texas – Austin, US), and Julian Shun (MIT – Cambridge, US)

The discussion largely focused on two ways of modeling dynamically changing graphs stored in distributed settings.

The first is a streaming model where both the computation and the stream of updates happen on the same clock. That is, the arrival of updates (from the stream) continues irrespective of how long it took to process the previous one, in contrast with more standard streaming models where each computation (no matter how expensive) is completed before the arrival of the next update. This model can reflect the notion of computing an answer that is correct w.r.t. some snapshot of the graph. Furthermore, the accuracy of the algorithm can be measured by the difference in time between the query and the snapshot where the answer came from. Also, the difficulty of these problems can be quantified by the rate of arrival of the updates.

For the graph connectivity problem, we showed that under a constant arrival rate, a delay of $O(n)$ can be achieved. Then considerable discussion went into the possible tradeoffs between arrival rates and delays in query answers.

We then discussed a parallel setting that focuses on the total communication, specifically ways of extending the bulk synchronous parallel (BSP) model to analyze the communications of dynamic graph algorithms. We discussed problems such as deleting from linked lists, hypergraph partitioning, and list ranking under this communication-centric view. Most of the technical discussions focused on the list ranking problem, which seeks to compute for each element in a linked list its distance to the head of the list. Here rearranging the storage so that consecutive portions of the list are on the same machine leads to gains in communication proportional to the storage associated with each processor. These discussions led us to believe that such partitions / rearrangements of data on machines can lead to provable gains in dynamic updates to the data.

4.3 k-GRIP: Combinatorial Approaches

Blair D. Sullivan (North Carolina State University – Raleigh, US)

Joint work of Blair D. Sullivan, Tim Davis, John Gilbert, Riko Jacob, Fredrik Manne, Lalla Mouatadid, Maria Predari, Christian Schultz, Rasmus Kyng, Alex Pothen, Sushant Sachdeva, Henning Meyerhenke

This group primarily focused on understanding the hardness of k-GRIP, the k-edge Graph Robustness Improvement Problem, where R is an arbitrary measure of "robustness". This is a slight generalization of the k-GRIP problem introduced by Henning Meyerhenke, which sets

R to be the total effective Resistance of the graph. Specifically, the problem asks for the set of k edges whose addition to a graph will minimize the resistance (maximize the robustness).

We considered three notions of robustness: diameter R_{diam}, "natural connectivity" R_{nc} (the sum of the subgraph centralities), and effective resistance R_{eff}. Starting with the diameter, we designed a reduction from k-SetCover to show that this problem is $\mathcal{NP}$-hard. We confirmed that this was previously known (Demaine et al, 2014), and the problem has a $(4 + \varepsilon)$-approximation. The measure R_{nc} was previously shown to be $\mathcal{NP}$-hard in the node/edge removal variant of the problem (Chan et al 2014), but the complexity of edge addition was left open. The group has a tentative $\mathcal{NP}$-hardness reduction for this problem (using k-SetCover again), but the issue of approximation remains open.

Finally, we returned to the original question of R_{eff}, which remains resistant to gadgeteering. Attempts were made in finding faster/better approaches than the naive greedy algorithm – with no immediate conclusive results. Moreover, we established one hurdle in reducing from problems like k-SetCover and k-Clique: in effective resistance, several paths of length x can actually be better than a direct connection, thwarting our initial attempts to force the location of edge additions. In later sessions, we considered the problem of Outlier Detection, which is also an L_2 objective, and was established to be $\mathcal{NP}$-hard by reduction from minimum bisection, but this remains work in progress. The same is true for establishing measures of robustness that are monotone and sub- or supermodular. These properties would admit an immediate greedy approximation.

We also implemented a heuristic method, and tested its performance on many matrices in the SuiteSparse Matrix Collection.

5 Panel discussions

5.1 Applications of the theory of random graphs in algorithmic analysis of large networks

Nelly Litvak (University of Twente, NL)

Random graphs have been introduced by Paul Erdős and Alfréd Rényi at the end of 1950s. Initially, they were invented and used to solve difficult combinatorial problems in graph theory. Notice that this was a very different purpose than modeling real-life networks, such as World Wide Web or Twitter, which did not even exist at that time!

The relevance to the network data motivated enormous developments in the theory of random graphs. More and more mathematicians get involved in this research, their background varying from theoretical probability and statistical mechanics to combinatorics and operations research. This has resulted in spectacular recent developments in the theory of random graphs and their applications to networks.

For example, by now we understand very well how the inhomogeneity in the degrees of nodes and presence of hubs affects graph distances (small-world phenomena) and network algorithms such as PageRank. Many new insightful results have been obtained on spreading phenomenon including competitive spreading of information or products. Many approaches, which did not exist before 2000, have by now become standard, such as applications of martingales and continuous time branching processes in the analysis of preferential attachment model – the famous model popularized by Albert and Barabasi, where new nodes arrive to a network and have preference to connect to nodes with high degrees. Recently several beautiful theoretical concepts have been introduced to formalize a limit of sparse graphs when graph size goes to infinity.

In my own research I analyze algorithms for large networks in the framework of the theory of random graphs. As an outcome of this discussion, I hope to learn about classes of problems where breakthrough in algorithmic analysis of real-life networks can be achieved by building on modern developments in the theory of random graphs.

Participants

- Nesreen K. Ahmed
Intel Labs – Santa Clara, US
- Eugenio Angriman
Universität Köln, DE
- David A. Bader
Georgia Institute of Technology – Atlanta, US
- Maciej Besta
ETH Zürich, CH
- Rob Bisseling
Utrecht University, NL
- Timothy Chu
Carnegie Mellon University – Pittsburgh, US
- Pierluigi Crescenzi
University of Florence, IT
- Timothy Alden Davis
Texas A&M University – College Station, US
- Irene Finocchi
Sapienza University of Rome, IT
- John Gilbert
University of California – Santa Barbara, US
- David F. Gleich
Purdue University – West Lafayette, US
- Riko Jacob
IT University of Copenhagen, DK
- George Karypis
University of Minnesota – Minneapolis, US
- Michel A. Kinsy
Boston University, US
- Marsha Kleinbauer
TU Kaiserslautern, DE
- Christine Klymko
LLNL – Livermore, US
- Yiannis Koutis
NJIT – Newark, US
- Danai Koutra
University of Michigan – Ann Arbor, US
- Rasmus Kyng
Harvard University – Cambridge, US
- Nelly Litvak
University of Twente, NL
- Fredrik Manne
University of Bergen, NO
- Henning Meyerhenke
HU Berlin, DE
- Marco Minutoli
Pacific Northwest National Lab. – Richland, US
- Lalla Mouatadid
University of Toronto, CA
- Danupon Nanongkai
KTH Royal Institute of Technology – Stockholm, SE
- Lorenzo Orecchia
Boston University, US
- Richard Peng
Georgia Institute of Technology – Atlanta, US
- Manuel Penschuck
Goethe-Universität – Frankfurt a. M., DE
- Cynthia A. Phillips
Sandia National Labs – Albuquerque, US
- Alex Pothen
Purdue University – West Lafayette, US
- Maria Predari
Universität Köln, DE
- Vijaya Ramachandran
University of Texas – Austin, US
- Sushant Sachdeva
University of Toronto, CA
- Ilya Safro
Clemson University, US
- Peter Sanders
KIT – Karlsruher Institut für Technologie, DE
- Christian Schulz
Universität Wien, AT
- Julian Shun
MIT – Cambridge, US
- Blair D. Sullivan
North Carolina State University – Raleigh, US
- Charalampos E. Tsourakakis
Harvard University – Cambridge, US