Report from Dagstuhl Seminar 13391

Algorithm Engineering

Edited by

Andrew V. Goldberg[1], Giuseppe F. Italiano[2], David S. Johnson[3], and Dorothea Wagner[4]

1 **Microsoft Research – Mountain View, US,** `goldberg@microsoft.com`
2 **Università di Roma "Tor Vergata", IT,** `italiano@disp.uniroma2.it`
3 **AT&T Research – Florham Park, US,** `dstiflerj@gmail.com`
4 **KIT – Karlsruhe Institute of Technology, DE,** `dorothea.wagner@kit.edu`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13391 "Algorithm Engineering". The *algorithm engineering approach* consists of a cycle of algorithm design, analysis, implementation, and experimental evaluation, with the aim of bridging the gap between theory and practice in the area of algorithms. This cycle of phases is driven by falsifiable hypotheses validated by experiments. Moreover, real-world instances often have direct impact on this cycle since they often expose modeling and analysis shortcomings. Algorithm engineering touches other research areas such as algorithm theory, combinatorial optimization, computer architecture, parallel and distributed computing, high-performance computing, and operations research. Prominent success stories in algorithm engineering include the linear program solver CPLEX, the traveling salesman suite CONCORDE, speed-up techniques for shortest paths computation, for example, in route planning, as well as graph partitioning and the computation of Steiner trees. All these topics are driven by the need for efficient algorithms and libraries for problems that appear frequently in real-world applications.

In the last fifteen years, this approach to algorithmic research has gained increasing attention. There is an ACM Journal on Experimental Algorithmics and several annual conferences (WAE/ESA applied track since 1997, Alenex since 1998, and WEA/SEA since 2001) and the series of DIMACS implementation challenges where people meet to compare implementations for a specific problem. From 2007 to 2013 the German Research Foundation also funded a special priority program on algorithm engineering (DFG SPP 1307).

Seminar 23.–27. September, 2013 – www.dagstuhl.de/13391
1998 ACM Subject Classification E.1 Data Structures, E.2 Data Storage Representations, F.2 Analysis of Algorithms and Problem Complexity, G.2 Discrete Mathematics
Keywords and phrases Algorithm Engineering, Science of Algorithmics, Manycore Algorithms, Certifying Algorithms, Web Search, Large Graphs
Digital Object Identifier 10.4230/DagRep.3.9.169
Edited in cooperation with Andrea Kappes

1 Executive Summary

Andrew V. Goldberg
Giuseppe F. Italiano
David S. Johnson
Dorothea Wagner

Topics of the Seminar

The seminar covered all methodological aspects of algorithm engineering. Examples are the scientific method in algorithmics, the use of modern computer architecture in algorithmics, and certifying algorithms. These aspects were also addressed in dedicated discussion sessions.

Science of Algorithmics

One aspect of algorithm engineering is the *scientific method*, where research on algorithms is interpreted as in other disciplines such as physics and life sciences: the observation of a phenomenon that is not yet understood is investigated via falsifiable hypotheses as explanations of the phenomena, and experimental evaluations to test these hypotheses. That way not only empirical evidence on the behavior of algorithms is attained but also new theoretical insights are sought. Experimental algorithmics is already a core component of algorithm engineering from its very beginning. However, the design of reasonable experiments, the use of meaningful test instances, and reproducibility of experiments are still issues to be discussed in order to derive a common understanding and agree on a best practice.

Manycore and GPU Algorithms

Exploiting the full potential of a *modern computer* poses many interesting new challenges for algorithm engineering: ever increasing parallelism, deep memory hierarchies, and heterogeneous architectures. Algorithms should be tailored to utilize multiple cores, but also access memory efficiently, taking into account issues such as data locality. Nowadays the use of GPUs, which are increasingly common in modern servers, is an important issue for efficient algorithm implementation. This is in particular interesting for frequently used and "classical" algorithms.

Certifying Algorithms

An effective way to ensure correct results of algorithm implementations are *certifying algorithms*. The idea is to check each returned result for correctness using a simple checker. It then suffices to test or perhaps verify the checker. Making checking fast implies interesting algorithmic questions when checking is aided by certificates of correctness computed by the main algorithm.

Focus Topic: Web Search and Large Graphs

Experiences from previous Dagstuhl seminars showed that the interaction between different scientific communities stimulates methodological discussions. This exchange is in particular important for neighboring scientific communities who typically meet at separate conferences.

For this seminar, we focus on *web search, large graphs and social networks* in order to also address the scientific WWW and Social Media community. In these fields, methods from algorithm engineering are applied. However, these scientists typically don't publish at the algorithm engineering conferences mentioned above but meet and publish at conferences like the "International World Wide Web Conference", the "ACM Conference on Hypertext and Hypermedia" or the "International AAAI Conference on Weblogs and Social Media".

Search engines work with a large amount of data, making high-performance algorithms and data structures very important. Relevant problems include fast indexing, text and query processing, and relevance computation. The latter involves a large web graph. Web-enabled applications give raise to other large graphs, such as social networks, like "friend graphs" or e-mail graphs induced by message origin-destination pairs. Algorithms on such graphs are of great interest. For example, identifying interest-based sub-communities (e.g., classical music fans) enables better service experience or contextual advertisement.

Aims

The aim of this seminar was to bring together researchers with different backgrounds, e.g., from algorithm and datastructures, computational geometry, combinatorial optimization, parallel algorithms and algorithm engineering in order to strengthen and foster collaborations and to identify key research directions for the future. In particular, the seminar was intended to foster the exchange between algorithm engineering and scientists from the web search community. While the dominant goal of the seminar was the exchange of current research developments and discussion of topical subjects, it also contributed to bring algorithm engineering forward as a still evolving and expanding field in computer science. The seminar program included four dedicated discussion sessions on methodological questions, as well as research related issues like future DIMACS Implementation Challenges.

Conclusion

The organizers decided to schedule talks and discussions not grouped according to topics but provide a vivid mix of different research questions and results. According to the composition of the seminar participants, not all topics were covered equally well. For example, certifying algorithms were not addressed in detail. On Monday, Renato Werneck gave a short report on the "11th DIMACS Implementation Challenge on Steiner Tree Problems"[1] taking place in 2013/14. The program of Monday afternoon was concluded by a panel discussion on "Empirical and Theoretical Approaches to Algorithm Design: Synergies and Opportunities". The second panel discussion on "Benchmarks and reproducibility of experiments" took place on Tuesday. The third panel discussion on Thursday focused on "Promoting and advancing the field" and on Friday a discussion about "Teaching Algorithm Engineering" concluded the program.

The seminar hosted 39 participants. Besides presentations and panel discussions the program offered room for bilateral discussions and working groups. Schloss Dagstuhl and its staff provided a very convenient and stimulating environment. The seminar participants appreciated the cordial atmosphere which improved mutual understanding and inspiration. The organizers of this seminar wish to thank all those who helped make the workshop a fruitful research experience.

[1] http://dimacs11.cs.princeton.edu/home.html

2 Table of Contents

Executive Summary
Andrew V. Goldberg, Giuseppe F. Italiano, David S. Johnson, and Dorothea Wagner 170

Overview of Talks

Algorithm Engineering Issues in Semantic Search
Hannah Bast 174

Engineering Little Algorithms into a Big System
Jon Bentley 174

New Algorithms for Computing the Greedy Spanner
Kevin Buchin 174

Real-Time On-line Analytical Processing (OLAP) On Cloud Architectures
Frank Dehne 175

Customizable Route Planning in Road Networks
Daniel Delling 175

Multi-Modal Route Planning – An Overview
Julian Dibbelt 176

The Hub Labeling Algorithm
Andrew V. Goldberg 176

Machine Learning & Optimisation: Promise and Power of Data-driven, Automated Algorithm Design
Holger H. Hoos 177

A Bootstrap Approach to Analysing the Scaling of Empirical Run-time Data with Problem Size
Holger H. Hoos 177

Fixed-Parameter and Integer Programming Approaches for Clustering Problems
Falk Hüffner 178

Engineering Algorithms for Color Barcodes
Giuseppe F. Italiano 178

The TSP for random points in the unit square: an experimentaland statistical study
David S. Johnson 178

Graph Clustering with the Louvain Method
Andrea Kappes 179

Seeking for the best priority queue: Lessons Learnt
Jyrki Katajainen 179

GraphML-Time: a file format for dynamic graphs
Juergen Lerner 180

The Computational Complexity of Virtual Address Translation
Kurt Mehlhorn 180

Engineering High-Performance Community Detection Heuristics for Massive Graphs
Henning Meyerhenke 181

I/O-efficient Hierarchical Diameter Approximation
Ulrich Carsten Meyer 181

The Art of Discrete Structure Manipulation Based on BDDs and ZDDs
Shin-ichi Minato 182

Teaching Algorithm Engineering
Matthias Müller-Hannemann 182

Enumeration algorithms: polynomial time by branch and bound versus exponential time by ZDDs
Yoshio Okamoto 183

Concurrent data structures in streaming of massive data and overlays in the smart grid context
Marina Papatriantafilou 183

Parallel Algorithm Reconsidered
Peter Sanders 184

Facility Location Problems in Street Networks
Sabine Storandt 185

Consistent Updates in Software Defined Networks
Roger Wattenhofer 186

An Exact Combinatorial Algorithm for Graph Bisection
Renato Werneck 186

Improved Alternative Route Planning
Christos Zaroliagis 186

A new algorithm for solving the All-Pairs-Shortest-Paths problem on positive, real-weighted graphs
Katharina A. Zweig 187

Panel Discussions

Benchmarks and reproducibility of experiments
Andrew V. Goldberg 187

Promoting and Advancing the Field
David S. Johnson 188

Participants 189

3 Overview of Talks

3.1 Algorithm Engineering Issues in Semantic Search

Hannah Bast (Universität Freiburg, DE)

I will talk about our research on semantic search over the past years, with an emphasis on the algorithm engineering perspective.

3.2 Engineering Little Algorithms into a Big System

Jon Bentley (Avaya – Basking Ridge, US)

This talk describes how a few researchers worked closely with a development team to improve the performance of an old yet very important system. It starts with an overview of the project, and then concentrates on the particular technical problem of cache design and implementation (including an interesting study of hashing). This interaction between research and development illustrates some old yet fundamental truths: defining the right problem can be most of the battle, and a little technology, properly applied, can make a big difference. (This talk describes joint work with Duffy Boyle, P. Krishnan and John Meiners.)

3.3 New Algorithms for Computing the Greedy Spanner

Kevin Buchin (TU Eindhoven, NL)

Joint work of Alewijnse, Sander P.A.; Bouts, Quirijn W.; ten Brink, Alex P.; Buchin, Kevin;
Main reference Sander P. A. Alewijnse, Quirijn W. Bouts, Alex P. ten Brink, Kevin Buchin, “Computing the Greedy Spanner in Linear Space”, in Proc. of 21st Annual Europ. Symp. on Algorithms (ESA’13), LNCS, Vol. 8125, pp. 37–48, Springer, 2013; also available as pre-print on arXiv (arXiv:1306.4919v1 [cs.CG]).
URL http://dx.doi.org/10.1007/978-3-642-40450-4_4
URL http://arxiv.org/abs/1306.4919v1

The greedy spanner is a high-quality spanner: its total weight, edge count and maximal degree are asymptotically optimal and in practice significantly better than for any other spanner with reasonable construction time. Unfortunately, all known algorithms that compute the greedy spanner of n points use $\Omega(n^2)$ space, which is impractical on large instances. To the best of our knowledge, the largest instance for which the greedy spanner was computed so far has about 13,000 vertices.

We will present an $O(n)$-space algorithm that computes the same spanner for points in R^d running in $O(n^2 log^2 n)$ time for any fixed stretch factor and dimension. We discuss and evaluate a number of optimizations to its running time, which allowed us to compute the greedy spanner on a graph with a million vertices. To our knowledge, this is also the first algorithm for the greedy spanner with a near-quadratic running time guarantee that has actually been implemented.

Furthermore, we will present two more approaches to obtain efficient algorithms for constructing the greedy spanner. The first approach drastically simplifies our first algorithm. It uses much less space and runs in near-quadratic time under reasonable assumptions on the input. The second approach is input-sensitive and aims at subquadratic running time.

3.4 Real-Time On-line Analytical Processing (OLAP) On Cloud Architectures

Frank Dehne (Carleton University – Ottawa, CA)

Joint work of Dehne, Frank; Q.Kong; A.Rau-Chaplin; H.Zaboli; R.Zhou,
Main reference F. Dehne, Q. Kong, A. Rau-Chaplin, H. Zaboli, R. Zhou, "A Distributed Tree Data Structure For Real-Time OLAP On Cloud Architectures", to appear in Proc. of IEEE Int'l Conf. on Big Data (BigData'13), IEEE CS, 2013.
URL http://www.scs.carleton.ca/~dehne/publications/2-94.pdf

In contrast to queries for on-line transaction processing (OLTP) systems that typically access only a small portion of a database, on-line analytical processing (OLAP) queries may need to aggregate large portions of a database which often leads to performance issues. We introduce CR-OLAP, a Cloud based Real-time OLAP system based on a new distributed index structure for OLAP, the distributed PDCR tree, that utilizes a cloud infrastructure consisting of $(m+1)$ multi-core processors. With increasing database size, CR-OLAP dynamically increases m to maintain performance. Our distributed PDCR tree data structure supports multiple dimension hierarchies and efficient query processing on the elaborate dimension hierarchies which are so central to OLAP systems. It is particularly efficient for complex OLAP queries that need to aggregate large portions of the data warehouse, such as "report the total sales in all stores located in California and New York during the months February-May of all years". We evaluated CR-OLAP on the Amazon EC2 cloud, using the TPC-DS benchmark data set. The tests demonstrate that CR-OLAP scales well with increasing number of processors, even for complex queries. For example, on an Amazon EC2 cloud instance with eight processors, for a TPC-DS OLAP query stream on a data warehouse with 80 million tuples where every OLAP query aggregates more than 50of the database, CR-OLAP achieved a query latency of 0.3 seconds which can be considered a real time response

3.5 Customizable Route Planning in Road Networks

Daniel Delling (Microsoft Research – Mountain View, US)

In this talk, I will present the latest version of our customizable route planning engine. It can compute exact shortest paths in continental road networks (with tens of millions of road segments) and is able to incorporate realistic turn costs, custom height and weight restrictions, personal preferences, and traffic in real time. Moreover, it can efficiently answer extended queries such as computing distance tables or best points-of-interest, which are important building blocks for advanced map applications.

3.6 Multi-Modal Route Planning – An Overview

Julian Dibbelt (KIT – Karlsruhe Institute of Technology, DE)

Joint work of Delling, Daniel; Dibbelt, Julian; Pajor, Thomas; Wagner, Dorothea; Werneck, Renato F.
Main reference D. Delling, J. Dibbelt, T. Pajor, D. Wagner, R. F. Werneck, "Computing Multimodal Journeys in Practice", in Proc. of the 12th Int'l Symp. on Experimental Algorithms (SEA'13), LNCS, Vol. 7933, pp. 260–271, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-38527-8_24

This talk gives an overview of recent approaches to location-to-location route planning for multimodal networks consisting of schedule-based public transit, (unrestricted) walking, driving, and cycling. We discuss challenges arising in the multimodal scenario that go beyond the challenges of each (unimodal) subproblem. We review recent algorithmic approaches to the problem such as preprocessing techniques that enable fast queries for interactive applications and conclude by discussing future work.

3.7 The Hub Labeling Algorithm

Andrew V. Goldberg (Microsoft Research – Mountain View, US)

Given a weighted graph, a distance oracle takes as an input a pair of vertices and returns the distance between them. The labeling approach to distance oracle design is to precompute a label for every vertex so that the distances can be computed from the corresponding labels, without looking at the graph. We survey results on hub labeling (HL), a labeling algorithm that received a lot of attention recently.

HL query time and memory requirements depend on the label size. While some graphs admit small labels, one can prove that there are graphs for which the labels must be large. Computing optimal hub labels is hard, but in polynomial time one can approximate them up to a factor of O(log(n)). This can be done for the total label size (i.e., memory required to store the labels), the maximum label size (which determines the worst-case query time), and in general for an Lp norm of the vector induced by the vertex label sizes. One can also simultaneously approximate Lp and Lq norms.

Hierarchical labels are a special class of HL. For networks with small highway dimension, one can compute provably small hierarchical labels in polynomial time. On the other hand, one can prove that for some graphs hierarchical labels are significantly larger than the general ones. A heuristic for computing hierarchical labels leads to the fastest implementation of distance oracles for road networks. One can use label compression to trade off time for space, making the algorithm practical for a wider range of applications. We give experimental results showing that the heuristic hierarchical labels work well on road networks as well as some other graph classes, but not on all graphs. We also discuss efficient implementations of the provably good approximation algorithms and give experimental results.

Finally, we show that the labels can be stored in a database and HL queries can be implemented in SQL, making the algorithm accessible to SQL programmers.

3.8 Machine Learning & Optimisation: Promise and Power of Data-driven, Automated Algorithm Design

Holger H. Hoos (University of British Columbia – Vancouver, CA)

Computational tools are transforming the way we live, work and interact; they also hold the key for meeting many of the challenges that we face as individuals, communities and societies. Machine learning and optimisation techniques play a particularly important role in this context, and cleverly combined, they can revolutionise the way we solve challenging computational problems – as I will demonstrate in this talk, using examples from mixed integer programming, planning and propositional satisfiability, as well as from prominent supervised machine learning tasks. The fundamental techniques I will cover include automated algorithm selection, configuration and hyperparameter optimisation, as well as performance prediction and Bayesian optimisation. I will also motivate and explain the Programming by Optimisation paradigm for algorithm design and engineering.

3.9 A Bootstrap Approach to Analysing the Scaling of Empirical Run-time Data with Problem Size

Holger H. Hoos (University of British Columbia – Vancouver, CA)

Joint work of Hoos, Holger H.
Main reference H. H. Hoos, “A bootstrap approach to analysing the scaling of empirical run-time data with problem size”, Technical Report TR-2009-16, University of British Columbia, June 2009.
URL http://www.cs.ubc.ca/~hoos/Publ/Hoos09.pdf

In this presentation, I describe a new approach for analysing the scaling of empirical run-time data of an algorithm when applied to sets of inputs of growing size. My method is based on the use of standard numerical techniques for fitting models, which are then challenged by extrapolation to larger problem sizes and statistically validated using bootstrap confidence intervals. It permits not only the automatic construction of predictive models of the given algorithm’s run-time, but also the comparative evaluation of multiple hypothesis on the scaling in the form of different parametric functions. I will illustrate the method using run-time data for Concorde, a state-of-the-art complete algorithm for the travelling salesperson problem (TSP), applied to a class of well-known Euclidean TSP instances. However, the method is generally applicable to most situations where functional (or statistical) models of the dependence of a performance measure of an algorithm on a single, numerical feature of sets of inputs have to be fitted and assessed.

3.10 Fixed-Parameter and Integer Programming Approaches for Clustering Problems

Falk Hüffner (TU Berlin, DE)

We examine two NP-hard clustering problems: Colorful Components, where the goal is to delete a minimum number of edges in a vertex-colored graph such that each remaining connected component contains each color at most once; and Highly Connected Deletion, where the goal is to delete a minimum number of edges in a graph such that each remaining connected component is highly connected. Here, a graph with n vertices is called highly connected if each vertex has degree larger than $n/2$. Both problems have many applications; for example, Colorful Components can be used to find inconsistencies in the mapping of entities between databases, and Highly Connected Deletion can be used to find protein complexes. Using a combination of data reduction and Integer Linear Programming, we are able to solve sizable real-world instances, for example the Wikipedia interlanguage link network with 47 million edges.

3.11 Engineering Algorithms for Color Barcodes

Giuseppe F. Italiano (University of Rome "Tor Vergata", IT)

Joint work of Firmani, Donatella; Italiano, Giuseppe F.; Querini, Marco;

2D color barcodes have been introduced to obtain larger storage capabilities than traditional black and white barcodes. Unfortunately, the data density of color barcodes is substantially limited by the redundancy needed for correcting errors, which are due not only to geometric but also to chromatic distortions introduced by the printing and scanning process. The higher the expected error rate, the more redundancy is needed for avoiding failures in barcode reading, and thus, the lower the actual data density. Our work addresses this trade-off between reliability and data density in 2D color barcodes and aims at identifying the most effective algorithms, in terms of byte error rate and computational overhead, for decoding 2D color barcodes. In particular, we perform a thorough experimental study to identify the most suitable color classifiers for converting analog barcode cells to digital bit streams.

3.12 The TSP for random points in the unit square: an experimentaland statistical study

David S. Johnson (Madison, US)

TSP instances generated by placing cities uniformly at random within the unit square provide reasonable surrogates for the instances arising in many real-world TSP applications. They are also an interesting object of study on their own. The optimal tour length is known to grow as

$c\sqrt{n}$ for some constant c, but the exact value of c and the details of the convergence rate have not been determined. In this talk I report on an ongoing study of this and related questions with David Applegate and Bill Cook, based on experiments with millions of instances using the Concorde software package. Interesting statistical questions arise.

3.13 Graph Clustering with the Louvain Method

Andrea Kappes (KIT – Karlsruhe Institute of Technology, DE)

The Louvain algorithm is a very fast and effective metaheuristic developed for modularity based graph clustering using local vertex moves in a multilevel scheme. I will talk about how the Louvain method can be applied to other objective functions in the presence of constraints on the density of the clustering [2]. Furthermore, it can be effectively transferred to a dynamic setting [1], and to a scenario where multiple clusterings with a varying number of clusters are requested. Finally, it can be modified to take vertex attributes into account. This talk is based on joint work with Philipp Glaser, Robert Görke, Tanja Hartmann, Patricia Iglesias, Uwe Korn, Emmanuel Müller, Christian Staudt, and Dorothea Wagner.

References

1 Robert Görke, Pascal Maillard, Andrea Schumm, Christian Staudt, and Dorothea Wagner. Dynamic Graph Clustering Combining Modularity and Smoothness. *ACM Journal of Experimental Algorithmics*, 18(1):1.5:1.1–1.5:1.29, April 2013.
2 Robert Görke, Andrea Schumm, and Dorothea Wagner. Experiments on Density-Constrained Graph Clustering. In *Proceedings of the 14th Meeting on Algorithm Engineering and Experiments (ALENEX'12)*, pages 1–15. SIAM, 2012.

3.14 Seeking for the best priority queue: Lessons Learnt

Jyrki Katajainen (University of Copenhagen, DK)

Joint work of Bojesen, Jesper; Bruun, Asger; Chen, Jingsen; Edelkamp, Stefan; Elmasry, Amr; Fadel, Ramz; Jakobsen, Kim Vagn; Jensen, Claus; Rasmussen, Jens; Spork, Maz; Teuhola, Jukka; Vitale, Fabio;
Main reference J. Katajainen, "Seeking for the best priority queue: Lessons learnt", CPH STL Report 2013-3, Department of Computer Science, University of Copenhagen, 10 pp., 2013.
URL http://cphstl.dk/Report/Dagstuhl-13391/the-best.pdf

Since 2004, I and my research collaborators have been studying the performance of different priority queues for different sets of operations to be supported. At the theoretical level, we have measured the goodness in terms of the comparison complexity of different operations. As the other optimization criteria, we have considered how to reduce the number of instructions, branch mispredictions, cache misses, and element moves. At the practical level, we have used the actual running time as the key performance indicator. We have done most of our experiments on synthetic operation sequences, but we have also done some application engineering.

In this talk I will briefly survey the theoretical results obtained and I will discuss the methodological issues encountered when performing the practical experiments. As a teaser, consider the following questions:

- What is the best priority queue?
- When to implement a data structure?
- Does a factor of two matter?
- Does the space efficiency matter?
- What is the effect of different hardware phenomena?
- How to measure the practical performance?
- What to do next?

It goes without saying that I do not have complete answers to these questions.

3.15 GraphML-Time: a file format for dynamic graphs

Juergen Lerner (Universität Konstanz, DE)

GraphML (http://graphml.graphdrawing.org/) is an XML-based file format for graphs with arbitrary additional data. During the last ten years GraphML has become an established standard in the graph drawing and network analysis communities. Due to its generality and robustness, it is especially suitable as a format for making graph data available to others. In this talk I propose a GraphML extension for time-varying graph data. The new format is not restricted to represent snapshots of graphs but is also able to efficiently encode dynamic graphs evolving in continuous time. A working draft for the new GraphML version can be found at http://visone.info/wiki/index.php/GraphML-Time

3.16 The Computational Complexity of Virtual Address Translation

Kurt Mehlhorn (MPI für Informatik – Saarbrücken , DE)

Modern computers are not random access machines (RAMs). They have a memory hierarchy, multiple cores, and a virtual memory. We address the computational cost of the address translation in the virtual memory. Starting point for our work on virtual memory is the observation that the analysis of some simple algorithms (random scan of an array, binary search, heapsort) in either the RAM model or the EM model (external memory model) does not correctly predict growth rates of actual running times. We propose the VAT model (virtual address translation) to account for the cost of address translations and analyze the algorithms mentioned above and others in the model. The predictions agree with the measurements. We also analyze the VAT-cost of cache-oblivious algorithms.

3.17 Engineering High-Performance Community Detection Heuristics for Massive Graphs

Henning Meyerhenke (KIT – Karlsruhe Institute of Technology, DE)

Main reference C. L. Staudt, H. Meyerhenke, "Engineering High-Performance Community Detection Heuristics for Massive Graphs," in Proc. of the 42nd Int'l Conf. on Parallel Processing (ICPP'13), pp. 180–189, IEEE CS, 2013; also available as pre-print on arXiv (arXiv:1304.4453v1 [cs.DC]).
URL http://dx.doi.org/10.1109/ICPP.2013.27
URL http://arxiv.org/abs/1304.4453

The amount of graph-structured data has recently experienced an enormous growth in many applications. To transform such data into useful information, high-performance analytics algorithms and software tools are necessary. One common graph analytics kernel is community detection (or graph clustering). Despite extensive research on heuristic solvers for this task, only few parallel codes exist, although parallelism is often necessary to scale to the data volume of real-world applications.

We address the deficit in computing capability by a flexible and extensible clustering algorithm framework with shared-memory parallelism. Within this framework we implement our parallel variations of known sequential algorithms and combine them by an ensemble approach. In extensive experiments driven by the algorithm engineering paradigm, we identify the most successful parameters and combinations of these algorithms. The processing rate of our fastest algorithm exceeds 10M edges/second for many large graphs, making it suitable for massive data streams. Moreover, the strongest algorithm we present includes a parallelization of the well-known Louvain method and yields a very good tradeoff between quality and speed.

3.18 I/O-efficient Hierarchical Diameter Approximation

Ulrich Carsten Meyer (Goethe-Universität Frankfurt am Main, DE)

Joint work of Ajwani, Deepak; Meyer, Ulrich; Veith, David;
Main reference D. Ajwani, U. Meyer, D. Veith, "I/O-efficient Hierarchical Diameter Approximation," in Proc. of the 20th Annual Europ. Symp. on Algorithms (ESA'12), LNCS, Vol. 7501, pp. 72–83, Springer, 2012.
URL http://dx.doi.org/10.1007/978-3-642-33090-2_8

Computing diameters of huge graphs is a key challenge in complex network analysis. As long as the graphs fit into main memory, diameters can be efficiently approximated (and frequently even exactly determined) using heuristics that apply a limited number of BFS traversals. If the input graphs have to be kept and processed on external storage, even a single BFS run may cause an unacceptable amount of time-consuming I/O-operations.

In previous work we proposed the first parameterized diameter approximation algorithm with fewer I/Os than that required for exact BFS traversal. In this talk we present hierarchical extensions of this randomized approach and experimentally compare their trade-offs between actually achieved running times and approximation ratios.

We show that the hierarchical approach is frequently capable of producing surprisingly good diameter approximations in shorter time than BFS. We also provide theoretical and practical insights into worst-case input classes.

3.19 The Art of Discrete Structure Manipulation Based on BDDs and ZDDs

Shin-ichi Minato (Hokkaido University, JP)

Main reference S. Minato, "Techniques of BDD/ZDD: Brief History and Recent Activity," IEICE Transactions on Information and Systems, Vol. E96D, No. 7, pp. 1419–1429, July 2013.
URL http://hdl.handle.net/2115/53121

Discrete structure manipulation is a fundamental technique for many problems solved by computers. Recently, the BDD/ZDD has attracted a great deal of attention, because it efficiently manipulates basic data structures such as logic and sets. In order to organize an integrated method of algebraic operations for manipulating various types of discrete structures, we are executing "ERATO Project", a nation-wide research project in Japan from 2009.

Currently, one of the most interesting research topics of our project is "frontier-based method", a very efficient BDD/ZDD-based algorithm for enumerating and indexing all the subsets of a graph to satisfy various kinds of constraints. For example, we applied this technique for analyzing electricity supply networks and succeeded in generating a ZDD of all solutions for a realistic benchmark with hundreds of control switches. The total number of the solutions was as many as 10^{63}, but the size of ZDD was only 1.1 million (780MByte), and the computation time was about 20 min.

Our project aims to develop efficient software tools based on BDDs/ZDDs and more sophisticated data structures with algebraic operations for various kinds of discrete structure manipulation, which are useful for real- life applications. It would be "the Art" between Science and Engineering. In this talk, we show an overview of our project and recent topics on BDD/ZDD-based discrete structure manipulation.

References

1 Shin-ichi Minato. *Techniques of BDD/ZDD : Brief History and Recent Activity*. IEICE Transactions on Information and Systems, Vol. E96D, No. 7 pp. 1419–1429, July 2013.

3.20 Teaching Algorithm Engineering

Matthias Müller-Hannemann (Martin-Luther-Universität Halle-Wittenberg, DE)

The way how we teach Algorithm Engineering (AE) is of importance for several reasons: it shapes how young researchers entering the field will think about AE and how ideas and techniques are disseminated to other fields. This talk is intended to initiate a discussion on various aspects of teaching. To do so, we give a brief overview of courses on AE taught by colleagues and myself. Issues for discussion include best practices, suggestions for a curriculum, accompanying course material, and ideas for course projects.

3.21 Enumeration algorithms: polynomial time by branch and bound versus exponential time by ZDDs

Yoshio Okamoto (University of Electro-Communications, Tokyo, JP)

License Creative Commons BY 3.0 Unported license
© Yoshio Okamoto
Joint work of Kiyomi, Masashi; Okamoto, Yoshio; Saitoh, Toshiki;
Main reference M. Kiyomi, Y. Okamoto, T. Saitoh, "Efficient Enumeration of the Directed Binary Perfect Phylogenies from Incomplete Data," in Proc. of the 11th Int'l Symp. on Experimental Algorithms (SEA'12), LNCS, Vol. 7276, pp. 248–259, Springer, 2012; also available as pre-print on arXiv (arXiv:1203.3284v1 [cs.DS]).
URL http://dx.doi.org/10.1007/978-3-642-30850-5_22
URL http://arxiv.org/abs/1203.3284

When we design an algorithm to enumerate (or list) all the combinatorial objects with a prescribed property, a branch-and-bound technique is quite useful, and in many cases it runs in worst-case polynomial time per output. This is a theoretically guaranteed algorithm, but it is not satisfactory for the purpose of recently growing interest in, for example, data mining and bioinformatics. We compare an approach by branch and bound, and an enumeration algorithm by zero-suppressed binary decision diagrams (ZDDs) proposed by Minato. As a case study, we consider the enumeration of directed binary perfect phylogenies when the input is incomplete. Our experiment shows that the ZDD approach outperforms the branch-and-bound algorithm. The talk is partially based on joint work with Masashi Kiyomi and Toshiki Saitoh [1].

References

1 M. Kiyomi, Y. Okamoto, and T. Saitoh, Efficient Enumeration of the Directed Binary Perfect Phylogenies from Incomplete Data. Proc. SEA 2012 (2012), pp. 248–259. DOI: 10.1007/978-3-642-30850-5_22
http://arxiv.org/abs/1203.3284arXiv:1203.3284 [cs.DS]

3.22 Concurrent data structures in streaming of massive data and overlays in the smart grid context

Marina Papatriantafilou (Chalmers UT – Göteborg, SE)

License Creative Commons BY 3.0 Unported license
© Marina Papatriantafilou
Joint work of Cederna, Daniel; Georgiadis, Georgios; Gulisano, Vincenzo; Nikolakopoulos, Yiannis; Papatriantafilou, Marina; Tsigas, Philippas
Main reference (collection of papers, see references in the abstract)

In many data gathering applications, information arrives in the form of continuous streams rather than finite data sets. Efficient one-pass and in-memory algorithms are required to cope with a high input load. Stream processing engines provide continuous queries to process data in a real-time fashion and have evolved rapidly from centralized to distributed, parallel and elastic solutions.

The presented work focuses on multiway aggregation, where large data volumes are received from multiple input streams. Multiway aggregation is crucial in contexts such as sensor networks, social media or clickstream analysis applications. We provide three enhanced aggregate operators that rely on new concurrent structures, with domain-related interface and their lock-free implementations, supporting both order-sensitive and order-insensitive aggregation functions. The design of these new data structures balances and parallelizes the work of the aggregate operator's processing steps. We provide an extensive study of the properties of the proposed aggregate operators and the new data structures. We also give empirical evidence of their superiority, running a variety of aggregation queries on two large datasets, one with data extracted from SoundCloud, a music social network, and one with data from a Smart Grid metering network.

In the context of the Smart Grid, network overlays can play a key role, for enabling the management of resources in an adaptive fashion. In this presentation we discussed our work on distributed matching with preferences for dynamic overlays with guarantees, as well as scheduling for demand-response problems in buldings and neighbourhoods in the context of the Smart Grid.

Joint work with Daniel Cederman, Giorgos Georgiadis, Vincenzo Gulisano, Yiannis Nikolakopoulos and Philippas Tsigas.

References

1 Daniel Cederman, Vincenzo Gulisano, Yiannis Nikolakopoulos, Marina Papatriantafilou and Philippas Tsigas. Concurrent Data Structures for Efficient Streaming Aggregation. Technical report, Chalmers University of Technology, 2013.
2 Georgios Georgiadis and Marina Papatriantafilou. A greedy algorithm for the unforecasted energy dispatch problem with storage in smart grids. In Proc. of the 4th Int'l Conf. on Future energy systems (e-Energy'13), ACM, pp. 273–274, 2013. DOI: 10.1145/2487166.2487203
3 Georgiadis, Giorgos, and Marina Papatriantafilou. Adaptive distributed b-matching in overlays with preferences. Experimental Algorithms. Springer, 2012. pp. 208–223.

3.23 Parallel Algorithm Reconsidered

Peter Sanders (KIT – Karlsruhe Institute of Technology, DE)

In this talk we want to motivate researchers in algorithmics in general and in algorithm engineering in particular to work more on parallel algorithms. We begin with a short history of parallel algorithmics – pointing out that a lot of work has been done in the 1980s whereas the area lies almost dormant since the mid 1990s. Since parallelism has moved from a niche application to a mandatory component of any high performance solution this should change. Also since many new developments in algorithmics have happened since then, there are many fundamental results just around the corner. This is particularly true for algorithm engineering. The second part of my talk gives several examples from my group: external sorting [6], string sorting [1], shortest paths [3, 5], multi-objective shortest paths [7], graph matching [2], graph partitioning [4, 9]. The area of communication efficient algorithms is proposed as an entire subfield of parallel algorithm research [8]. Our results there include distributed data structures for approximate membership, duplicate detection, joins, and low-dimensional linear programming. Another interesting area that deserves more attentions are work-efficient algorithms with polylogarithmic critical path length. A surprising new result here is on matrix inversion [10].

References

1 T. Bingmann and P. Sanders. Parallel string sample sort. In *21st European Symposium on Algorithme (ESA)*, volume 8125 of *LNCS*, pages 169–180. Springer, 2013.

2 M. Birn, V. Osipov, P. Sanders, C. Schulz, and N. Sitchinava. Efficient parallel and external matching. In *Europar*, LNCS. Springer, 2013.

3 A. Crauser, K. Mehlhorn, U. Meyer, and P. Sanders. Parallelizing Dijkstra's shortest path algorithm. Technical report, MPI-Informatik, 1998. in preparation.

4 M. Holtgrewe, P. Sanders, and C. Schulz. Engineering a scalable high quality graph partitioner. In *24th IEEE International Symposium on Parallel and Distributed Processing*, 2010.

5 U. Meyer and P. Sanders. Δ-stepping: A parallelizable shortest path algorithm. *Journal of Algorithms*, 49(1):114–152, 2003.

6 M. Rahn, P. Sanders, and J. Singler. Scalable distributed-memory external sorting. In *26th IEEE International Conference on Data Engineering*, pages 685–688, 2010.

7 P. Sanders and L. Mandow. Parallel label-setting multi-objective shortest path search. In *27th IEEE International Parallel & Distributed Processing Symposium*, pages 215–224, 2013.

8 P. Sanders, S. Schlag, and I. Müller. Communication efficient algorithms for fundamental big data problems. In *IEEE Int. Conf. on Big Data*, 2013.

9 P. Sanders and C. Schulz. Distributed evolutionary graph partitioning. In *ALENEX 2012*, 2012.

10 P. Sanders, J. Speck, and R. Steffen. Work-efficient matrix inversion in polylogarithmic time. In *25th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 214–221. ACM, 2013.

3.24 Facility Location Problems in Street Networks

Sabine Storandt (Universität Freiburg, DE)

Joint work of Storandt, Sabine; Funke, Stefan;
Main reference S. Storandt, S. Funke, "Enabling E-Mobility: Facility Location for Battery Loading Stations," in Proc. of the 27th AAAI Conf. on Artificial Intelligence (AAAI'13), AAAI Press, 2013.
URL http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/view/6239
URL http://ad-publications.informatik.uni-freiburg.de/AAAI_evcover_SF_2013.pdf

The main disadvantage of current Electric Vehicles is the short cruising range due to the limited battery supply. Hence for driving longer routes a dense network of loading stations needs to be established. We investigate the problem of placing as few loading stations as possible to achieve mobility goals like global reachability, connectivity and feasibility of shortest paths. Unfortunately, it turns out that the respective optimization problems are all hard to approximate. Moreover naive algorithms require too much space and runtime for practical use. We devise algorithms which compute solutions on the basis of heuristics with incomplete knowledge and a new shortest path enumeration and representation scheme. While these algorithms do not provide any a priori approximation guarantee, we still can prove their results to be close-to-optimal for our real-world inputs with the help of instance-based lower bounds.

3.25 Consistent Updates in Software Defined Networks

Roger Wattenhofer (ETH Zürich, CH)

Computer networks are boring: most aspects are well-understood, and protocols have been standardized a long time ago. However, some people are not happy with the state of the art in networking. Why for instance do networks often waste more than half of the available bandwidth? Introducing software defined networking (SDN), an operator gets more control over its network. With more control, one can possibly achieve results that are not possible with standard protocols. In my talk, I present the SWAN project, an SDN-based effort to raise the bandwidth usage in a corporate wide area network to a mesmerizing 99%. I will focus on some algorithmic questions of the SWAN project, in particular multi-commodity flow and asynchronous updates. My presentation is based on joint work with colleagues at Microsoft Research, published at SIGCOMM 2013 and HotNets 2013.

3.26 An Exact Combinatorial Algorithm for Graph Bisection

Renato Werneck (Microsoft Research – Mountain View, US)

Joint work of Delling, Daniel; Fleischman, Daniel; Goldberg, Andrew; Razenshteyn, Ilya; Werneck, Renato F.;
Main reference D. Delling, D. Fleischman, A. Goldberg, I. Razenshteyn, R. F. Werneck, "An Exact Combinatorial Algorithm for Graph Bisection," submitted for publication.
URL http://research.microsoft.com/apps/pubs/default.aspx?id=200845

We present a novel exact algorithm for the minimum graph bisection problem, whose goal is to partition a graph into two equally-sized cells while minimizing the number of edges between them. Our algorithm is based on the branch-and-bound framework and, unlike most previous approaches, it is fully combinatorial. Besides introducing stronger lower bounds, we propose a new decomposition technique that contracts entire regions of the graph without losing optimality guarantees. In practice, our algorithm works particularly well on instances with relatively small minimum bisections, solving large real-world graphs (with tens of thousands to more than a million vertices) to optimality.

3.27 Improved Alternative Route Planning

Christos Zaroliagis (CTI & University of Patras, GR)

Joint work of Paraskevopoulos, Andreas; Zaroliagis, Christos;
Main reference A. Paraskevopoulos, C. Zaroliagis, "Improved Alternative Route Planning," in Proc. of the 13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'13), OASICS, Vol. 33, pp. 108–122, 2013.
URL http://dx.doi.org/10.4230/OASIcs.ATMOS.2013.108

We present improved methods for computing a set of alternative source-to-destination routes in road networks in the form of an alternative graph. The resulting alternative graphs are characterized by minimum path overlap, small stretch factor, as well as low size and complexity. Our approach improves upon a previous one by introducing a new pruning stage preceding any other heuristic method and by introducing a new filtering and fine-tuning of two existing methods. Our accompanying experimental study shows that the entire alternative graph can be computed pretty fast even in continental size networks.

3.28 A new algorithm for solving the All-Pairs-Shortest-Paths problem on positive, real-weighted graphs

Katharina A. Zweig (TU Kaiserslautern, DE)

Joint work of Zweig, Katharina Anna;
Main reference (unpublished work)

While better algorithms for the all-pairs-shortest-paths-problem (APSP) exist in many cases, the simplicity of Dijkstra's algorithm still makes it the most used one with a runtime of $O(nm + n^2 logn)$. Real-world networks such as street or sensor networks are known to be sparse such that the second term in the runtime of Dijkstra's algorithm is dominating. In this talk I will present a new algorithm for the APSP and show that it can be expected to run in $O(nm)$ for weighted sensor networks. If it is implemented as a distributed algorithm where every sensor computes the distance to all other nodes, the message complexity is also in $O(nm)$ and the algorithm can be implemented in an asynchronous fashion. Finally, I will give empirical evidence that the APSP can also be solved in $O(nm)$ for many street networks.

4 Panel Discussions

4.1 Benchmarks and reproducibility of experiments

Andrew V. Goldberg (Microsoft Research – Mountain View, US)

Common benchmarks are important for the field Algorithm Engineering. They allow researchers to compare experimental results. The discussion included the topics of creating the benchmarks, e.g., via the Challenges. An important issue that has been raised is benchmark evolution: As algorithms become better, benchmarks may become outdated and need to a revision. A related issue is overfitting, when over time the algorithms become tuned to the benchmark instances at the expense of other problems.

Since the Science of Algorithmics applies the scientific method to algorithm design, it is has to deal with the reproducibility and verification issues common to natural sciences. Making code available partially addresses these issues. However, this does not completely settle them: For example, the code may not faithfully implement its description in the corresponding paper. System issues also play a role, especially when large distributed systems and supercomputers with limited access are involved. Another approach to reproducibility is independent reimplementation. It is a great way to reproduce the results, and should be encouraged when happens. However, for sophisticated code, reimplementation is difficult, and often there is little incentive for reimplementation.

4.2 Promoting and Advancing the Field

David S. Johnson (Madison, US)

This panel discussion was lead by David Johnson, and considered two main topics. The first was potential topics for future DIMACS Implementation Challenges. The current Challenge on Steiner Problems, and the previous one on Clustering and Partitioning, were both based on ideas suggested at the previous Dagstuhl workshop on Algorithm Engineering. This time a number of ideas were floated, including a combination of facility location and the k-median problem, but perhaps the most popular one was to reprise the first Challenge from 1990–91, which covered Network Flows and Matching. There have been many new theoretical developments since then, with several major ones in the last two years that seem quite promising but haven't yet been adequately tested. At the first Challenge, a paradigm shift was confirmed, with the then-new "preflow-push" approach of Goldberg and Tarjan dominating the previous state of the art. Could a new paradigm shift be at hand?

The second main topic was that of prizes. At the last workshop, it was suggested that we try to get sponsors for a "Best Algorithm Engineering/Experimentation paper of the Year" award. One suggestion was to get SIGACT and EATCS to jointly sponsor such an award, as they currently do for theoretical papers with their Gödel Prize. Initial responses from the leaders of SIGACT were not encouraging, but new leaders are now in charge, and it is hoped that with a more detailed proposal we might now be successful.

Participants

- Hannah Bast
Universität Freiburg, DE
- Jon Bentley
Avaya – Basking Ridge, US
- Robert E. Bixby
Rice University – Houston, US
- Gerth Stølting Brodal
Aarhus University, DK
- Kevin Buchin
TU Eindhoven, NL
- Markus Chimani
Universität Osnabrück, DE
- Frank Dehne
Carleton Univ. – Ottawa, CA
- Daniel Delling
Microsoft Research – Mountain View, US
- Julian Dibbelt
KIT – Karlsruhe Institute of Technology, DE
- Rudolf Fleischer
German Univ. of Technology – Oman, OM
- Andrew V. Goldberg
Microsoft Research – Mountain View, US
- Holger H. Hoos
University of British Columbia – Vancouver, CA
- Falk Hüffner
TU Berlin, DE
- Giuseppe F. Italiano
University of Rome “Tor Vergata”, IT
- David S. Johnson
Madison, US
- Andrea Kappes
KIT – Karlsruhe Institute of Technology, DE
- Jyrki Katajainen
University of Copenhagen, DK
- Jürgen Lerner
Universität Konstanz, DE
- Kurt Mehlhorn
MPI für Informatik – Saarbrücken, DE
- Ulrich Carsten Meyer
Goethe-Universität Frankfurt am Main, DE
- Friedhelm Meyer auf der Heide
Universität Paderborn, DE
- Henning Meyerhenke
KIT – Karlsruhe Institute of Technology, DE
- Shin-ichi Minato
Hokkaido University, JP
- Rolf H. Möhring
TU Berlin, DE
- Matthias Müller-Hannemann
Martin-Luther-Universität Halle-Wittenberg, DE
- Petra Mutzel
TU Dortmund, DE
- Patrick K. Nicholson
MPI für Informatik – Saarbrücken, DE
- Yoshio Okamoto
University of Electro-Communications – Tokyo, JP
- Marina Papatriantafilou
Chalmers UT – Göteborg, SE
- Alejandro Salinger
Universität des Saarlandes, DE
- Peter Sanders
KIT – Karlsruhe Institute of Technology, DE
- Federico Santaroni
University of Rome “Tor Vergata”, IT
- Sabine Storandt
Universität Freiburg, DE
- Dorothea Wagner
KIT – Karlsruhe Institute of Technology, DE
- Roger Wattenhofer
ETH Zürich, CH
- Renato Werneck
Microsoft Research – Mountain View, US
- Christos Zaroliagis
CTI & University of Patras, GR
- Liang Zhao
KIT – Karlsruhe Institute of Technology, DE
- Katharina A. Zweig
TU Kaiserslautern, DE