Report from Dagstuhl Seminar 25181

Learned Predictions for Data Structures and Running Time

Inge Li Gørtz*[1], Benjamin J. Moseley*[2], Shikha Singh*[3], and Sergei Vassilvitskii*[4]

1 **Technical University of Denmark - Lyngby, DK.** inge@dtu.dk
2 **Carnegie Mellon University - Pittsburgh, US.** moseleyb@andrew.cmu.edu
3 **Williams College - Williamstown, US.** ss32@williams.edu
4 **Google - New York, US.** sergeiv@gmail.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 25181 "Learned Predictions for Data Structures and Running Time". The focus of the seminar was applying the new algorithms-with-predictions framework to improve worst-case running time of algorithms and data structures.

This seminar brought together researchers from the data structures, combinatorial optimization and learned predictions communities to address the challenges of adopting learned machine-learned predictions for improving running time guarantees.

Seminar April 27 – May 2, 2025 – https://www.dagstuhl.de/25181
2012 ACM Subject Classification Theory of computation
Keywords and phrases algorithms with predictions, approximation algorithms, beyond-worst-case analysis, data structures, learning-augmented algorithms
Digital Object Identifier 10.4230/DagRep.15.4.112

1 Executive Summary

Inge Li Gørtz
Benjamin J. Moseley
Shikha Singh
Sergei Vassilvitskii

Traditionally, the performance of algorithms and data structures is measured in terms of the worst-case analysis of their *cost*, e.g. running time, approximation ratio or memory usage. Algorithms designed in the worst-case model are optimized to perform well against all instances, even carefully contrived adversarial instances. While the worst-case model leads to extremely strong guarantees, it can be a pessimistic predictor of the algorithm's performance on instances typically seen in practice. In particular, it is frequently observed that the theoretically best worst-case algorithm may not necessarily be the fastest algorithm in experimental evaluations. An emerging line of work, called **algorithms with predictions** or learning-augmented algorithms, uses machine learning to design improved algorithms that circumvent worst-case lower bounds. The majority of prior work in this area has focused on improving the quality of the solution (that is, the competitive ratio) of optimization algorithms. How to leverage learned predictions to speed up the *running time* of optimization algorithms and data structures has received less attention.

* Editor / Organizer

Learned Predictions for Data Structures and Running Time, *Dagstuhl Reports*, Vol. 15, Issue 4, pp. 112–125
Editors: Inge Li Gørtz, Benjamin J. Moseley, Shikha Singh, and Sergei Vassilvitskii
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Seminar Overview

The goal of this Dagstuhl Seminar was to develop the theoretical foundations of using predictions for faster data structures and optimization. It brought together researchers from different but complementary areas of data structures, combinatorial optimization and learning theory.

The initial days of the seminar consisted of a small number of survey style talks (45 mins) focused on the key areas of learning-augmented and worst-case data structures, machine learning for NP hard problems as well as learning-augmented streaming algorithms. The rest of the schedule included shorter technical presentations, unstructured collaboration time, open-problem working groups as well as group discussions. Overall, the seminar was successful in bridging the gap between the different communities as well as creating new technical collaborations among the participants.

Technical Themes

Overall, the seminar converged on the following key themes.

Beyond-Worst-Case Data Structures. A major theme was how predictions can be used to improve the running time of fundamental data structures such as sorted arrays, priority queues and search trees. The main techniques that were discussed in the talks were: how prediction decomposition can be used for designing learned data structures, how history independence has been surprisingly successful at improving worst-case data structure performance, and how techniques such as compression, sketching and adapting to workload patterns can be leveraged for faster practical performance.

Warm-starting offline optimization. Traditional optimization algorithms assume that problems are solved from scratch each time. However, algorithms are often run on different correlated data sets over time. These prior runs can encode information that can be used to improve run times in the future. An intuitive approach to speed up an algorithm is to start with some previously computed solution, referred to as a *warm start*. Warm start is commonly used by heuristics to improve empirical performance. This seminar focused on how several recent results that provide a theoretical foundation of warm-start heuristics in the algorithms-with-predictions model.

Dynamic and streaming algorithms with predictions. Predictions can be specially useful in dynamic and streaming algorithms to help with the uncertainty with future inputs. The seminar focused on how predicting future updates can be used to efficiently "lift" fully offline algorithms to the online setting. This technique proved to be specially effective in the area of dynamic graph problems where the existing worst-case algorithms have slow update times (polynomial in input size). Similarly in the streaming setting, predictions about future updates enabled efficient sketching algorithms. Finally, the seminar also focused on how dynamic algebraic algorithms have been leveraged predictions to bypass worst-case lower bounds.

Learned combinatorial optimization. Finally, the seminar focused on how correlations in the input distributions can be exploited to design more efficient data-driven algorithms for optimization problems. These techniques have been applied to a variety of domains such as configuring integer programming algorithms, using heat maps and neural net techniques for improved algorithms for NP hard problems such as TSP, and using predicted distributions for portfolio optimization.

Takeaways and Conclusion

We believe the Dagstuhl Seminar was successful in meeting our as well as the participants expectations. The post-seminar survey was highly positive. Attendees rated the scientific quality of the seminar very highly (median 10/11) and said that it inspired new research ideas, interdisciplinary insights, and potential collaborations, particularly due to its successful integration of communities working on predictions and data structures.

In the future, we would like to include more PhD students as well as start working group activities earlier in the week. Overall, we were very happy with the seminar. We would like to thank the entire Schloss Dagstuhl team for making the seminar possible and ensuring that everything from initial planning, to logistics, meals, etc. were impeccably smooth.

2 Table of Contents

Executive Summary

Overview of Talks

Panel discussions

3 Overview of Talks

3.1 Approximation Algorithms with Predictions

Antonios Antoniadis (University of Twente, NL)

We initiate a systematic study of utilizing predictions to improve over approximation guarantees of classic algorithms, without increasing the running time. We propose a systematic method for a wide class of optimization problems that ask to select a feasible subset of input items of minimal (or maximal) total weight. This gives simple (near-)linear time algorithms for, e.g., Vertex Cover, Steiner Tree, Min-Weight Perfect Matching, Knapsack, and Clique. Our algorithms produce optimal solutions when provided with perfect predictions and their approximation ratios smoothly degrade with increasing prediction error. With small enough prediction error we achieve approximation guarantees that are beyond reach without predictions in the given time bounds, as exemplified by the NP-hardness and APX-hardness of many of the above problems. Although we show our approach to be optimal for this class of problems as a whole, there is a potential for exploiting specific structural properties of individual problems to obtain improved bounds; we demonstrate this on the Steiner Tree problem. We conclude with an empirical evaluation of our approach.

3.2 Sorting and Priority Queues with Predictions

Christian Coester (University of Oxford, GB)

Joint work of Xingjian Bai, Ziyad Benomar, Christian Coester

The talk discusses two of the most fundamental problems in algorithms and data structures through the lens of algorithms-with-predictions: sorting and priority queues. We consider different types of predictions and show how these can be used to improve running time and comparison complexity of algorithms. In one setting, the algorithm is given predictions about the rank of items, and in another setting, the algorithm has access to quick-and-dirty comparisons to complement much slower exact comparisons. We obtain simple algorithms with optimal guarantees and favorable experimental performance in sorting tasks as well as when applied to Dijkstra's shortest path algorithm.

3.3 Ski Rental With Distributional Predictions

Michael Dinitz (Johns Hopkins University - Baltimore, US)

Ski rental was one of the first problems to be considered in the algorithms with predictions literature. We revisit this problem from the perspective of distributional predictions. We give an algorithm with optimal additive loss as a function of the earth mover's distance between the predicted distribution and the true distribution, and show both analytically and experimentally that in many natural settings distributional predictions allow for significantly better performance than traditional point predictions.

3.4 Scenario-Based Robust Optimization of Tree Structures

Christoph Dürr (Sorbonne University - Paris, FR)

Joint work of Christoph Dürr, Spyros Angelopoulos, Alex Elenter, Georgii Melidi
Main reference Spyros Angelopoulos, Christoph Dürr, Alex Elenter, Georgii Melidi: "Scenario-Based Robust Optimization of Tree Structures", in Proc. of the AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA, pp. 26895–26903, AAAI Press, 2025.
URL https://doi.org/10.1609/AAAI.V39I25.34894

We initiate the study of tree structures in the context of scenario-based robust optimization. Specifically, we study Binary Search Trees (BSTs) and Huffman coding, two fundamental tech- niques for efficiently managing and encoding data based on a known set of frequencies of keys. Given k different scenarios, each defined by a distinct frequency distribution over the keys, our objective is to compute a single tree of best-possible performance, relative to any scenario. We consider, as performance metrics, the competitive ratio, which compares multiplicatively the cost of the solution to the tree of least cost among all scenarios, as well as the regret, which induces a similar, but additive comparison. For BSTs, we show that the problem is NP-hard across both metrics. We also show how to obtain a tree of competitive ratio $\lceil \log_2(k+1) \rceil$, and we prove that this ratio is optimal. For Huffman Trees, we show that the problem is, likewise, NP-hard across both metrics; we also give an algorithm of regret $\lceil \log_2 k \rceil$, which we show is near-optimal, by proving a lower bound of $\lfloor \log_2 k \rfloor$. Last, we give a polynomial-time algorithm for computing Pareto-optimal BSTs with respect to their regret, assuming scenarios defined by uniform distributions over the keys. This setting captures, in particular, the first study of fairness in the context of data structures. We provide an experimental evaluation of all algorithms. To this end, we also provide mixed integer linear program formulation for computing optimal trees.

3.5 From compression to learning, one more step in data structure design

Paolo Ferragina (Sant'Anna School of Advanced Studies - Pisa, IT)

Key-value stores and search engines are posing a continuously growing need to efficiently store, retrieve and analyze massive sets of keys under the many and different requirements posed by users, devices, and applications. Such a new level of complexity could not be properly handled by known data structures, so that academic and industrial researchers started recently to devise new approaches that integrate classic data structures with various kinds of advanced techniques drawn from data compression, computational geometry and machine learning, hence originating what are currently called "Learned Data Structures". In this talk, I'll survey the evolution of these kinds of data structures, discuss their theoretical and experimental performance, highlight their limits, and point out new challenges worth of future research.

3.6 Nearly Optimal List Labeling

Hanna Komlós (NYU - New York, US)

Joint work of Michael A. Bender, Alex Conway, Martín Farach-Colton, Hanna Komlós, Michal Koucký, William Kuszmaul, Michael E. Saks

Main reference Michael A. Bender, Alex Conway, Martín Farach-Colton, Hanna Komlós, Michal Koucký, William Kuszmaul, Michael E. Saks: "Nearly Optimal List Labeling", in Proc. of the 65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024, pp. 2253–2274, IEEE, 2024.

URL https://doi.org/10.1109/FOCS61266.2024.00132

The list-labeling problem is a fundamental data structural primitive which captures the basic task of storing a dynamically changing set of up to N elements in sorted order in an array of size $M = \Theta(N)$. The goal is to support online insertions and deletions while moving elements around in the array as little as possible. Despite over four decades of study, there has until recently been a longstanding gap between the upper and lower bounds for list labeling ($O(\log^2(N))$ and $\Omega(\log(N))$ element moves per operation, respectively). This gap was narrowed in 2022 by Bender et al. with a randomized history-independent algorithm with $O(\log^{1.5}(N))$ cost, and they showed a matching lower bound for history-independent list-labeling algorithms. In recent work, we nearly close the gap by providing an $\tilde{O}(\log(N))$ algorithm which leverages the workload's history of operations. We do this by building a randomized predictor of future operations that performs well on all inputs in expectation. Combining this with a prediction-augmented algorithm, we achieve an improved worst-case upper bound.

3.7 Faster Data Structures via History Independence

William Kuszmaul (Carnegie Mellon University - Pittsburgh, US)

This talk will survey a recent trend through which history independence – a widely-studied privacy property in the data-structures literature – has been used as an algorithmic tool, not just for privacy, but for building faster and better data structures.

3.8 Data Driven Solution Portfolios

Silvio Lattanzi (Google - Barcelona, ES)

Joint work of Marina Drygala, Silvio Lattanzi, Andreas Maggiori, Miltiadis Stouras, Ola Svensson, Sergei Vassilvitskii

Main reference Marina Drygala, Silvio Lattanzi, Andreas Maggiori, Miltiadis Stouras, Ola Svensson, Sergei Vassilvitskii: "Data-Driven Solution Portfolios", in Proc. of the 16th Innovations in Theoretical Computer Science Conference, ITCS 2025, January 7-10, 2025, Columbia University, New York, NY, USA, LIPIcs, Vol. 325, pp. 46:1–46:15, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

URL https://doi.org/10.4230/LIPICS.ITCS.2025.46

In this paper, we consider a new problem of portfolio optimization using stochastic information. In a setting where there is some uncertainty, we ask how to best select k potential solutions, with the goal of optimizing the value of the best solution. More formally, given a combinatorial problem Π, a set of value functions V over the solutions of Π, and a distribution D over V, our goal is to select k solutions of Π that maximize or minimize the expected value of the *best* of those solutions. For a simple example, consider the classic knapsack problem: given a universe of elements each with unit weight and a positive value, the task is to select r elements maximizing the total value. Now suppose that each element's weight comes from a (known) distribution. How should we select k different solutions so that one of them is likely to yield a high value?

In this work, we tackle this basic problem, and generalize it to the setting where the underlying set system forms a matroid. On the technical side, it is clear that the candidate solutions we select must be diverse and anti-correlated; however, it is not clear how to do so efficiently. Our main result is a polynomial-time algorithm that constructs a portfolio within a constant factor of the optimal.

3.9 Learning Link Deletions

Quanquan C. Liu (Yale University - New Haven, US)

The main bottleneck in designing efficient dynamic algorithms is the unknown nature of the update sequence. In particular, there are some problems, like triconnectivity, planar digraph all pairs shortest paths, k-edge connectivity, and others, where the separation in runtime between the best offline or partially dynamic solutions and the best fully dynamic solutions is polynomial, sometimes even exponential. In this talk, we introduce the predicted-updates dynamic model, one of the first beyond-worst-case models for dynamic algorithms, which generalizes a large set of well-studied dynamic models including the offline dynamic, incremental, and decremental models to the fully dynamic setting when given predictions about the update times of the elements. In the most basic form of our model, we receive a set of predicted update times for all of the updates that occur over the event horizon. We give a novel framework that "lifts" offline divide-and-conquer algorithms into the fully dynamic setting with little overhead. Using this, we are able to interpolate between the offline and fully dynamic settings; when the ℓ_1 error of the prediction is linear in the number of updates, we achieve the offline runtime of the algorithm (up to poly log n factors). The second part of the talk will feature experimental results on how feasible it is to obtain these predictions in real-world graphs. We discuss new results that learn deletion times in dynamic networks using a very simple linear regression framework based on a set of features we find for our testing datasets.

3.10 Learned Data Structures via Decomposition

Samuel McCauley (Williams College - Williamstown, US)

This talk is about recent results for how predictions can improve online algorithms for list labelling, online cycle detection, and incremental shortest paths. In each case, the main idea of the technique is to use predictions to decompose the input into smaller pieces which can be solved more efficiently. I will discuss how this technique works, as well as how to use it in the future: both the potential to apply it to new problems, and common challenges that may arise.

3.11 Machine Learning for Better Algorithms

Benjamin J. Moseley (Carnegie Mellon University - Pittsburgh, US)

This talk explores the emerging area of algorithms with predictions, also known as learning-augmented algorithms. These methods incorporate machine-learned predictions into algorithmic design, allowing the algorithms to adapt to input distributions and achieve improved performance in runtime, space, or solution quality. The presentation will highlight recent advances in leveraging predictions to enhance the efficiency of discrete algorithms. The goal is to overview the breadth of the uses of the model in numerous areas, showcasing prior work and examples.

3.12 From heatmaps to TSP tours

Adam Polak (Bocconi University - Milan, IT)

Joint work of Adam Polak, Marek Eliáš, Fabrizio Grandoni, Eleonora Vercesi

This will be a talk about the travelling salesperson problem. I will show how to improve upon the approximation guarantees of the classic Christofides algorithm by using per-edge predictions as to whether the edge is part of the optimal TSP tour. Our algorithm can be seen as a tool for turning a probabilistic heatmap, produced by a deep neural network, into a discrete solution – which is a crucial step in pipelines of modern neural network solvers for TSP and other combinatorial optimization problems. I will present preliminary empirical results comparing our algorithm to other typical solutions for that step.

3.13 Learning-Augmented Mechanism Design

Golnoosh Shahkarami (MPI für Informatik - Saarbrücken, DE)

In the strategic facility location problem, a set of agents report their locations in a metric space and the goal is to use these reports to open a new facility, minimizing an aggregate distance measure from the agents to the facility. However, agents are strategic and may misreport their locations to influence the facility's placement in their favor. The aim is to design truthful mechanisms, ensuring agents cannot gain by misreporting. This problem was recently revisited through the learning-augmented framework, aiming to move beyond worst-case analysis and design truthful mechanisms that are augmented with (machine-learned) predictions. The focus of this prior work was on mechanisms that are deterministic and augmented with a prediction regarding the optimal facility location. In this paper, we provide a deeper understanding of this problem by exploring the power of randomization as well as the impact of different types of predictions on the performance of truthful learning-augmented mechanisms. We study both the single-dimensional and the Euclidean case and provide upper and lower bounds regarding the achievable approximation of the optimal egalitarian social cost.

3.14 Streaming algorithms with predictions

Ali Vakilian (TTIC - Chicago, US)

In this talk, I'll provide an overview of streaming algorithms with predictions. I'll begin with results for vector-based problems (e.g., frequency estimation, norm estimation, low-rank approximation), and then focus on the use of recent epsilon-accurate predictions in graph streaming settings, specifically for the Max-Cut problem.

3.15 Dynamic Algebraic Algorithms with Predictions

Jan van den Brand (Georgia Institute of Technology - Atlanta, US)

Joint work of Jan van den Brand, Sebastian Forster, Yasamin Nazari, Adam Polak
Main reference Jan van den Brand, Sebastian Forster, Yasamin Nazari, Adam Polak: "On Dynamic Graph Algorithms with Predictions", in Proc. of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024, pp. 3534–3557, SIAM, 2024.
URL https://doi.org/10.1137/1.9781611977912.126

Dynamic algebraic algorithms are data structures that maintain properties of matrices and vector spaces, such as matrix inverse, determinant, rank, etc. They are the main subroutine behind the best algorithms for many dynamic graph problems, such as maintaining reachability, maximum matching size, triangle detection, and many more.

In this talk, we will see techniques that accelerate dynamic algebraic algorithms when predictions about future updates are available. Via reductions, this allows to maintain fully dynamic triangle detection, maximum matching, single-source reachability, in $O(n^{\omega-1} + n\eta_i)$ worst-case update time. Here η_i denotes how much earlier the i-th update occurs than predicted, and $\omega\tilde{2}.372$ is the matrix multiplication exponent. For comparison, there are conditional lower bounds that prove no non-trivial data structures are possible without predictions.

3.16 Machine learning for online matching and integer programming

Ellen Vitercik (Stanford University, US)

When designing algorithms for computational problems in practice, there is often ample data about the application domains in which these algorithms will operate. This data presents a valuable opportunity: it can be leveraged via machine learning (ML) to improve the performance of existing optimization algorithms, help practitioners select among different algorithms, and guide the discovery of entirely new algorithms. However, this promise hinges on several core challenges, including: How to align deep learning architectures with algorithmic tasks, Whether to integrate ML into existing algorithms or train them end-to-end for combinatorial tasks, How to supervise ML models on (NP-hard) algorithmic problems, where ground-truth labels are computationally expensive to obtain, and Whether we can provide robust theoretical guarantees for the resulting algorithms. This talk will showcase solutions through two case studies. The first uses graph neural networks to learn near-optimal online matching policies that generalize beyond their training regime. The second shows how large language models can configure integer programming solvers with only a handful of training instances.

3.17 Beyond-worst-case Data Structures

Sebastian Wild (University of Liverpool, GB)

Lazy search trees [1] are a comparison-based sorted dictionary that smoothly interpolates between binary search trees and efficient priority queues automatically, depending on use. In particular they support faster insertions if many elements are not queried. Lazy B-trees bring this trade-off to external memory. Lazy search trees (of either kind) achieve this interpolation by a beyond-worst-case view of operations on sorted dictionaries, in particular by taking the ranks of queries into account and avoiding insertions to be always preceded by a query. The talk also suggested considering predictions of upcoming queries towards further improvements.

References

1 Sandlund, B., and Wild, S. *Lazy search trees.* In 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS) (pp. 704-715). IEEE.

3.18 CPMA: An Efficient Batch-Parallel Set Without Pointers

Helen Xu (Georgia Institute of Technology - Atlanta, US)

Main reference Brian Wheatman, Randal C. Burns, Aydin Buluç, Helen Xu: "CPMA: An Efficient Batch-Parallel Compressed Set Without Pointers", in Proc. of the 29th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming, PPoPP 2024, Edinburgh, United Kingdom, March 2-6, 2024, pp. 348–363, ACM, 2024.
URL https://doi.org/10.1145/3627535.3638492

This talk introduces the batch-parallel Compressed Packed Memory Array (CPMA), a compressed, dynamic, ordered set data structure based on the Packed Memory Array (PMA). Traditionally, batch-parallel sets are built on pointer-based data structures such as trees because pointer-based structures enable fast parallel unions via pointer manipulation. When compared with cache-optimized trees, PMAs were slower to update but faster to scan. The batch-parallel CPMA overcomes this tradeoff between updates and scans by optimizing for cache-friendliness. On average, the CPMA achieves 3x faster batch-insert throughput and 4x faster range-query throughput compared with compressed PaC-trees, a state-of-the-art batch-parallel set library based on cache-optimized trees. We further evaluate the CPMA compared with compressed PaC-trees and Aspen, a state-of-the-art system, on a real-world application of dynamic-graph processing. The CPMA is on average 1.2x faster on a suite of graph algorithms and 2x faster on batch inserts when compared with compressed PaC-trees. Furthermore, the CPMA is on average 1.3x faster on graph algorithms and 2x faster on batch inserts compared with Aspen. At the end, I will provide some hot takes (opinions expressed here are only my own) on how I think these areas relate to learned indices.

4 Panel discussions

4.1 Industry Research Discussion

Panelists: *Sergei Vassilvitskii (Google – New Yor, US), Rob Johnson (Broadcom – San Jose, US) and Cindy Phillips (Sandia National Labs – Albuquerque, US)*

During the coffee and cake break on Tuesday, we held an informal panel discussion on industry research. Participants asked our panelists questions about their experiences working in their current roles. The discussion was meant to mentor students and junior academics and to offer practical advice for those considering or engaging careers outside academia.

4.2 Women$^+$ in Theoretical Computer Science

During the coffee break on Thursday, we held an informal gathering for all self-identifying women and non-binary participants at the workshop. The goal was to provide mentorship and networking opportunity for the junior members. The topics of discussion included teaching and advising challenges as well as career trajectories in academia and industry.

Participants

- Antonios Antoniadis
University of Twente, NL
- Martin Aumüller
IT University of Copenhagen, DK
- Ioana Oriana Bercea
KTH Royal Institute of Technology – Stockholm, SE
- Philip Bille
Technical University of Denmark – Lyngby, DK
- Gerth Stølting Brodal
Aarhus University, DK
- Christian Coester
University of Oxford, GB
- Alexander Conway
Cornell Tech – New York, US
- Michael Dinitz
Johns Hopkins University – Baltimore, US
- Christoph Dürr
Sorbonne University – Paris, FR
- Martin Farach-Colton
NYU – New York, US
- Paolo Ferragina
Sant'Anna School of Advanced Studies – Pisa, IT
- Inge Li Gørtz
Technical University of Denmark – Lyngby, DK
- John Iacono
ULB – Brussels, BE
- Sungjin Im
University of California – Santa Cruz, US
- Riko Jacob
IT University of Copenhagen, DK
- Rob Johnson
Broadcom – San Jose, US
- Hanna Komlós
NYU – New York, US
- László Kozma
FU Berlin, DE
- William Kuszmaul
Carnegie Mellon University – Pittsburgh, US
- Silvio Lattanzi
Google – Barcelona, ES
- Thomas Lavastida
University of Texas – Dallas, US
- Alexander Lindermayr
Universität Bremen, DE
- Quanquan C. Liu
Yale University – New Haven, US
- Samuel McCauley
Williams College – Williamstown, US
- Ulrich Carsten Meyer
Goethe University – Frankfurt am Main, DE
- Benjamin J. Moseley
Carnegie Mellon University – Pittsburgh, US
- Yasamin Nazari
VU Amsterdam, NL
- Kim Thang Nguyen
University of Grenoble, FR
- Debmalya Panigrahi
Duke University – Durham, US
- Cynthia A. Phillips
Sandia National Labs – Albuquerque, US
- Adam Polak
Bocconi University – Milan, IT
- Rajeev Raman
University of Leicester, GB
- Golnoosh Shahkarami
MPI für Informatik – Saarbrücken, DE
- Shikha Singh
Williams College – Williamstown, US
- Clifford Stein
Columbia University – New York, US
- Ola Svensson
EPFL – Lausanne, CH
- Ali Vakilian
TTIC – Chicago, US
- Jan van den Brand
Georgia Institute of Technology – Atlanta, US
- Sergei Vassilvitskii
Google – New York, US
- Ellen Vitercik
Stanford University, US
- Sebastian Wild
University of Liverpool, GB
- Chenyang Xu
East China Normal University – Shanghai, CN
- Helen Xu
Georgia Institute of Technology – Atlanta, US