Report from Dagstuhl Seminar 23491

Scalable Graph Mining and Learning

Danai Koutra*[1], Henning Meyerhenke*[2], Ilya Safro*[3], and Fabian Brandt-Tumescheit†[4]

1 **University of Michigan – Ann Arbor, US & Amazon, US.** dkoutra@umich.edu
2 **Humboldt-Universität zu Berlin, DE.** meyerhenke@hu-berlin.de
3 **University of Delaware, Newark, US.** isafro@udel.edu
4 **Humboldt-Universität zu Berlin, DE.** fabian.brandt-tumescheit@hu-berlin.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23491 "Scalable Graph Mining and Learning". The event brought together leading researchers and practitioners to discuss cutting-edge developments in graph machine learning, massive-scale graph analytics frameworks, and optimization techniques for graph processing. Besides the executive summary, the report contains abstracts of the 18 scientific talks presented, descriptions of three open problems, and preliminary results of three working groups formed during the seminar. In summary, the seminar successfully fostered discussions that bridged theoretical research and practical applications in scalable graph learning, mining, and analytics. Several potential outcomes include writing position and research papers as well as joint grant submissions.

Seminar December 03–08, 2023 – https://www.dagstuhl.de/23491

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Computing methodologies → Machine learning algorithms; Computing methodologies → Parallel algorithms

Keywords and phrases Graph mining, Graph machine learning, (hyper)graph and network algorithms, high-performance computing for graphs, algorithm engineering for graphs

Digital Object Identifier 10.4230/DagRep.13.12.1

1 Executive Summary

Danai Koutra (University of Michigan – Ann Arbor, US & Amazon, US, dkoutra@umich.edu)
Henning Meyerhenke (Humboldt-Universität zu Berlin, US, meyerhenke@hu-berlin.de)
Ilya Safro (University of Delaware, Newark, US, isafro@udel.edu)

This Dagstuhl Seminar demonstrated a comprehensive exploration of the latest advancements and research directions in the field of scalable graph analytics and learning. The seminar featured a diverse array of talks that spanned foundational theory, innovative algorithms, and real-world applications. This event brought together leading researchers and practitioners to discuss cutting-edge developments in graph machine learning, massive-scale graph analytics frameworks, and optimization techniques for graph processing.

The seminar highlighted significant contributions to graph neural networks and representation learning, emphasizing such discussions as the problems related to the development of scalable algorithms, a proper benchmarking of the algorithms, several types of data reduction on graphs and various models including static, dynamic, and streaming graph data.

* Editor / Organizer
† Editorial Assistant / Collector

Scalable Graph Mining and Learning, *Dagstuhl Reports*, Vol. 13, Issue 12, pp. 1–23
Editors: Danai Koutra and Henning Meyerhenke and Ilya Safro
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

On the algorithmic front, several participants presented frameworks and algorithms designed for efficient graph analytics at massive scales in regular and streaming contexts. The topics included such broad problems as graph sparsification, various versions of graph partitioning, and matchings in big graphs. They highlighted the ongoing efforts to address the scalability challenges inherent in processing large-scale graph data. Notably, some emphasis was on accelerating optimization on graphs by machine learning.

The seminar also served as a platform for discussing the practical implications and applications of graph mining and learning in various domains. Examples of talks and discussions included such topics as the role of knowledge graphs, causal discovery and inference from social networks, as well as industry level graph mining. All these topics illuminated the broad applicability of graph analytics in social science, industry, and beyond. Moreover, these case studies provided valuable insights into how theoretical advancements translate into applications.

One of the working groups explored using graph neural networks for smaller kernels with an example of the maximum weighted independent set problem, focusing on understanding GNNs' ability to learn existing reduction rules, discovering new rules through GNN insights, and potentially enhancing graph reduction beyond current capabilities. Another working group discussed the ways of formulating the FAIR (Findable, Accessible, Interoperable, and Reusable) principles in an algorithmic context and promoting the ways to generalize future algorithm developement to bridge the gap between specialized practical applications and fundamental algorithms. Another working group focused on causal representation learning, aiming to identify high-level causal variables from observational data within graphs. This area is important for improving machine learning models and understanding causal mechanisms in various networks like social and protein interaction networks. Despite the importance, there's limited research on applying causal representation learning to graph data, a gap that hinders our ability to discern causal relationships in complex systems. The group emphasized the need for advancements in graph representation learning to address this challenge effectively.

In summary, the "Scalable Graph Mining and Learning" Dagstuhl Seminar successfully fostered discussions that bridged theoretical research and practical applications in scalable graph learning, mining, and analytics. Several potential outcomes include writing position and research papers as well as joint grant submissions.

2 Table of Contents

Executive Summary

Overview of Talks

Open Problems

Working Groups

3 Overview of Talks

3.1 Graph Machine Learning and Neural Network Research

George Karypis (University of Minnesota – Minneapolis, US, karypis@umn.edu)

This talk surveys recent accomplishments in graph machine learning and graph neural network research.

3.2 Arachne: An Open-Source Framework for Interactive Massive-Scale Graph Analytics

David A. Bader (NJIT – Newark, US, bader@njit.edu)

Joint work of David A. Bader, Oliver Alvarado Rodriguez, Zhihui Du, Joseph Patchett, Naren Khatwani, Fuhuan Li

A real-world challenge in data science is to develop interactive methods for quickly analyzing new and novel data sets that are potentially of massive scale. In this talk, Bader will discuss his development of graph algorithms in the context of Arkouda, an open-source NumPy-like replacement for interactive data science on tens of terabytes of data. Massive-scale analytics is an emerging field that integrates the power of high-performance computing and mathematical modeling to extract key insights and information from large-scale data sets. Productivity in massive-scale analytics entails quick interpretation of results through easy-to-use frameworks, while also adhering to design principles that combine high-performance computing and user-friendly simplicity. However, data scientists often encounter challenges, especially with graph analytics, which require the analysis of complex data from various domains, such as the cybersecurity, natural and social sciences. To address this issue, we introduce Arachne, an open-source framework that enhances accessibility and usability in massive-scale graph analytics. Arachne offers novel algorithms and implementations of graph kernels for efficient data analysis, such as connected components, breadth-first search, triangle counting, k-truss, among others. The high-performance algorithms are integrated into a back-end server written in HPE/Cray's Chapel language and can be accessed through a Python application programming interface (API). Arachne's back-end server is compatible with Linux supercomputers, is easy to set up, and can be utilized through either Python scripts or Jupyter notebooks, which makes it a desirable tool for data scientists who have access to high performance computers. In this talk, Bader presents an overview of the algorithms his research group has implemented into Arachne and, if applicable, the algorithmic innovations of each. Further, Bader will discuss improvements to our graph data structure to store extra information such as node labels, edge relationships, and node and edge properties. Arachne is built as an extension to the open-source Arkouda framework and allows for graphs to be generated from Arkouda dataframes.

The open-source code for Arachne can be found at `https://github.com/Bears-R-Us/arkouda-njit`. This is joint work with Oliver Alvarado Rodriguez, Zhihui Du, Joseph Patchett, Naren Khatwani, Fuhuan Li, Bader is supported in part by the National Science Foundation award CCF-2109988.

3.3 Data Reductions in Combinatorial Optimization

Ernestine Großmann (Universität Heidelberg, DE, e.grossmann@informatik.uni-heidelberg.de)

Combinatorial optimization problems play a pivotal role in various domains, challenging researchers to find optimal solutions within large solution spaces. One key strategy to tackle the complexity of these problems is data reduction, where instances are simplified to facilitate more efficient algorithms. In this talk, we delve into the world of combinatorial optimization, exploring classic data reduction techniques and their impact on problem-solving efficiency.

Additionally, we will explore different strategies of utilizing machine learning to reduce problem instances in the context of combinatorial optimization and ask how machine learning and exact data reductions can be combined more effectively. In particular, in future work we want to explore potential synergies between these approaches, considering the strengths of each and the challenges in their integration.

3.4 Targeted Branching for the Maximum Independent Set Problem Using Graph Neural Networks

Kenneth Langedal (University of Bergen, NO, kenneth.langedal@uib.no)

Identifying a maximum independent set is a fundamental NP-hard problem. This problem has several real-world applications and requires finding the largest possible set of vertices not adjacent to each other in an undirected graph. Over the past few years, branch-and-bound and branch-and-reduce algorithms have emerged as some of the most effective methods for solving the problem exactly. Specifically, the branch-and-reduce approach, which combines branch-and-bound principles with reduction rules, has proven particularly successful in tackling previously unmanageable real-world instances. This progress was largely made possible by the development of more effective reduction rules. Nevertheless, other key components that can impact the efficiency of these algorithms have not received the same level of interest. Among these is the branching strategy, which determines which vertex to branch on next. Until recently, the most widely used strategy was to choose the vertex of the highest degree. In this work, we present a graph neural network approach for selecting the next branching vertex. The intricate nature of current branch-and-bound solvers makes supervised and reinforcement learning difficult. Therefore, we use a population-based genetic algorithm to evolve the model's parameters instead. Our proposed approach results in a speedup on 73% of the benchmark instances with a median speedup of 24%.

3.5 What is the role of Knowledge Graphs in Graph Mining and Learning?

Davide Mottin (Aarhus University, DK, davide@cs.au.dk)

A knowledge graph (KG) represents a unique convergence of graph structures and natural language. On one side the graph structure is well defined by connection among entities. On the other hand, each entity and each relationship has a type and convey semantic meaning. Over time, research has emphasized different facets – while the machine learning community favors semantic and semi-structured information, the Data Mining and Database community views KGs as labeled graphs. Recently, the raise of Language Models and multimodal learning, seem to have found a new space for KGs to enhance the interpretability and improve the accuracy of models. This talk will survey some research on all sides of the coin and challenge the audience in finding a unification framework for KGs.

3.6 Causal Discovery from Social Networks

Elena Zheleva (University of Illinois – Chicago, US, ezheleva@uic.edu)

Social network data breaks a fundamental assumption of existing causal inference techniques, known as the Stable Unit Treatment Value Assumption (SUTVA), which states that the treatment of one unit cannot influence the outcome of other units. In real-world scenarios, it is common for related units to interact with each other which leads to interference (also known as spillover, or peer effects), in which the outcomes of units are interdependent. For example, the opinion of one person can influence the opinion of their friends, and the health status of one individual can impact the health status of others they interact with. In this talk, I will focus on recent work and problems related to discovering causal insights from social network data.

3.7 Topologies of Reasoning: Demystifying Chains, Trees, and Graphs of Thoughts

Maciej Besta (ETH Zürich, CH, maciej.besta@inf.ethz.ch)

The field of natural language processing has witnessed significant progress in recent years, with a notable focus on improving language models' performance through innovative prompting techniques. Among these, structure-enhanced prompting has emerged as a promising paradigm, with designs such as Chain-of-Thought (CoT) or Tree of Thoughts (ToT), in which the LLM reasoning is guided by a structure such as a tree. We refer to these structures as reasoning topologies, because their representation becomes to a degree spatial, as they are contained within the LLM context. In the first part of the talk, we overview this recent field, focusing on fundamental classes of harnessed structures, the representations of these structures, algorithms executed with these structures, relationships to other parts of the generative AI pipeline such as knowledge bases or Graph Neural Networks, and others. Second, we introduce Graph of Thoughts (GoT): a framework that advances prompting capabilities in LLMs beyond those offered by CoT or ToT. The key idea and primary advantage of GoT is the ability to model the information generated by an LLM as an arbitrary graph, where units of information ("LLM thoughts") are vertices, and edges correspond to dependencies between these vertices. This approach enables combining arbitrary LLM thoughts into synergistic outcomes, distilling the essence of whole networks of thoughts, or enhancing thoughts using feedback loops. We illustrate that GoT offers advantages over state of the art on different tasks such as keyword counting while simultaneously reducing costs. We finalize with outlining research challenges in this fast-growing field.

3.8 Mining and Learning with Graphs at Google Scale

Bryan Perozzi (Google – New York, US, bperozzi@cs.stonybrook.edu)

For over a decade, the Graph Mining team in Google Research has been working on the mission to build the world's most scalable library for graph algorithms and analysis, and use it to improve Google products. Our goal is to make it easy for developers to use graph algorithms in their applications, and to provide them with the tools they need to build scalable and efficient graph processing systems. Our algorithms and systems are used in a wide array of Google products, such as Search, YouTube, AdWords, Play, Maps, and Social. The systems developed by our team have had a significant impact on the way that Google engineers build and deploy graph-based applications.

In this talk I will give an overview of the team, what it works on, and what challenges we view as fundamental for large scale graph systems.

3.9 Bridging Nodes and Edges: The Interdisciplinary Landscape of Graph Mining, Complex Networks, Social Network Analysis and Network Science

Frank Takes (Leiden University, NL, f.w.takes@liacs.leidenuniv.nl)

Various disciplines, from computer science to physics to sociology to economics, have each approached "graphs" or "networks" from a different perspective or interest. Each of these fields has established its specific publishing venues, conferences and societies. Overall, the talk aims to present what (future) challenges and opportunities this division over disciplines brings in terms of the exchange of methods and findings. In addition to highlighting how insights from one field have positively influenced the other, the talk also describes several instances of "multiple discoveries" of network phenomena.

3.10 Matchings in Big Graphs: Approximation and Streaming Algorithms

Alex Pothen (Purdue University – West Lafayette, US, apothen@purdue.edu)

Matchings in graphs are classical problems in combinatorial optimization and computer science, significant due to their theoretical importance and relevance to applications. Polynomial time algorithms for several variant matching problems with linear objective functions have been known for fifty years. However, these algorithms fail to compute matchings in big graphs with billions of edges. They are also not concurrent and thus practical parallel algorithms are not known.

This has led to work in the last twenty years on designing approximation algorithms for variant matching problems with near-linear time complexity in the size of the graphs. Approximation has thus become a useful paradigm for designing parallel matching algorithms and also streaming algorithms. In this talk I will report on an approach to fast approximation algorithms and streaming algorithms for the maximization version of edge-weighted matching. It can be extended to edge-weighted b-matching, and the maximum k-disjoint weighted matching problems.

Matching and related problems could be applied to graph mining, where the matching objective is a submodular function.

3.11 Graph sparsification

Richard Peng (University of Waterloo, CA, peng@uwaterloo.ca)

Graph sparsification is the approximation of dense graphs by sparse ones. Work over the past two decades have exhibited a wide range of objectives that are sparsifiable, such as cut structure, dense subgraphs, and eigenvectors, but also objectives that are not sparsifiable, such as distances and matchings. This talk will overview recent results on sparsification, and also discuss algorithmic ways of blurring the distinctions between sparse and dense problems.

3.12 Towards Foundation Models for Knowledge Graph Reasoning

Mikhail Galkin (Intel AI Lab – San Diego, US, mikhail.galkin@intel.com)

Foundation models in graph learning are hard to design due to the lack of common invariances that transfer across different structures and domains. In this talk, I will give an overview of ULTRA, our new approach for creating foundation models for knowledge graph reasoning that captures relation interactions and does not require any input node or edge features. Experimentally, a single pre-trained ULTRA in the zero-shot inference mode outperforms supervised SOTA models on 50+ diverse graphs and can generalize to any multi-relational graph.

3.13 Improving Generalizability in Link Prediction with Application in Drug Discovery

Ayan Chatterjee (Northeastern University – Boston, US, chatterjee.ay@northeastern.edu)

State-of-the-art link prediction models leveraging neighborhood topology for node representation learning underperform in data-scarce regimes involving low-degree nodes and entities with insufficient metadata. To overcome this challenge, we have proposed a non-end-to-end training approach leveraging unsupervised pre-training on a corpus significantly different from and larger than the training graph to enhance the generalizability of link prediction models. Additionally, we have introduced AI-Bind, a model that combines network science-inspired negative sampling and unsupervised pre-training on molecular representation. AI-Bind not only addresses topological shortcuts and the generalization shortcomings of existing models in predicting drug-target interactions but also excels in identifying binding locations on proteins for novel molecular structures. Furthermore, its integration with AutoDock Tools expedites the molecular docking process.

3.14 Partitioning communication streams into graph snapshots

Cynthia Phillips (Sandia National Labs – Albuquerque, US, caphill@sandia.gov)

Joint work of Jeremy D. Wendt, Richard Field, Cynthia Philips, Arvind Prasadan, Tegan Wilson, Sucheta Soundarajan, Sanjukta Bhowmick

We give a technique for partitioning streaming communication data into static graph snapshots. We use combinatorial statistical models to adaptively find when a snapshot is stable, while watching for significant data shifts – indicating a new snapshot should begin. If snapshots are not found carefully, they poorly represent the underlying data – and downstream graph analytics can fail.

We demonstrate the method on several real-world datasets and show its accuracy against known-answer synthetic datasets. Not surprisingly, snapshot properties change from those created with other methods. For example, our snapshots do not generally "densify" over time, contradicting previous influential results that used simpler partitioning methods.

3.15 Mining Small Patterns in Dynamic Graphs

Kathrin Hanauer (Universität Wien, AT, kathrin.hanauer@univie.ac.at)

Detecting, counting, or even enumerating the copies of a specific subgraph pattern contained in a graph is an important tool in the analysis of various kinds of networks, from social sciences to biology and chemistry, with numerous further applications, e.g. in machine learning on graphs, building graph generators, or graph partitioning. In many use case scenarios, the input graphs are almost naturally subject to change, i.e., edges may newly arrive or disappear, vertices may be introduced or abandoned, or weights may change, which makes the input graph *dynamic*.

In this talk, we will focus on maintaining the number of occurrences of small patterns on three or four vertices in a dynamic graph that undergoes an a priori unknown sequence of edge insertions and deletions. In the first part, we will see how this problem can be solved algorithmically in theory. Afterwards, we will also take a look at the practical side and discuss how the algorithms behave on real-world instances and how they can be further improved experimentally.

The talk concludes with some challenges and open questions in algorithm engineering for dynamic graph problems.

3.16 Architectures, Algorithms, and Applications for Graph Mining and Learning

Johannes Langguth (Simula Research Laboratory – Oslo, NO, langguth@simula.no)

A major recent development in computer hardware was the rise of dedicated accelerator hardware for machine learning applications such as the Graphcore IPUs and Cerebras WSE. These processors have evolved from the experimental state into market-ready products, and they have the potential to constitute the next major architectural shift after GPUs saw widespread adoption a decade ago.

A salient feature of these devices is the use of SRAM for memory, which offers very low latency and high bandwidth, making them attractive for a wide range of graph algorithms. On the other hand, the wide parallelism employed in these devices makes it difficult to use them efficiently for irregular computations.

In this talk we will present the new hardware and discuss the programming techniques that are required to unlock their potential. We present implementations of basic graph algorithms and show early results on the attainable performance, as well as comparisons to other architectures. We follow up by discussing the wider implications of the architecture for algorithm design and programming, along with the wider implications of adopting such hardware, and we discuss some recent graph mining applications.

3.17 Approximating the Diagonal of a Directed Graph Laplacians's Pseudoinverse

Fabian Brandt-Tumescheit (Humboldt Universität zu Berlin, DE, brandtfa@hu-berlin.de)

The ubiquity of massive graph data sets in numerous applications requires fast algorithms for extracting knowledge from these data. We are motivated here by three electrical measures for the analysis of large small-world graphs $G = (V, E)$ – i.e., graphs with diameter in $O(\log |V|)$, which are abundant in complex network analysis. From a computational point of view, the three measures have in common that their crucial component is the diagonal of the graph Laplacian's Pseudoinverse, $L^\dagger$. Computing $diag(L^\dagger)$ exactly by pseudoinversion, however, is as expensive as dense matrix multiplication – and the standard tools in practice even require cubic time. Moreover, the Pseudoinverse requires quadratic space – hardly feasible for large graphs. Resorting to approximation by, e.g., using the Johnson-Lindenstrauss transform, requires the solution of $O(\log |V|/\epsilon^2)$ Laplacian linear systems to guarantee a relative error, which is still costly for large inputs.

In the ongoing project, we work on extending a previous approximation algorithm that requires the solution of only one Laplacian linear system and sampling of uniform spanning trees, which then are related to $diag(L^{\dagger})$ via effective resistances. This previous work so far only supports undirected graphs, and the extension is towards directed graphs. For that, we introduce a new sampling scheme, which relies on the sampling of escape random walks. In theory, the converted algorithm obtains an $\pm\epsilon$-approximation with high probability in a time that is nearly linear in $|E|$ and quadratic in $1/\epsilon$. In addition, the formulation makes the approximation suitable for GPU-based implementation, leading to massive parallelism.

3.18 Contextualizing protein representations learned on protein networks and single-cell data

Michelle Li (Harvard University – Boston, US, michelleli@g.harvard.edu)

Understanding protein function and developing molecular therapies require deciphering the cell types in which proteins act as well as the interactions between proteins. However, modeling protein interactions across diverse biological contexts, such as tissues and cell types, remains a significant challenge for existing algorithms. We introduce PINNACLE, a flexible geometric deep learning approach that is trained on contextualized protein interaction networks to generate context-aware protein representations. Leveraging a human multi-organ single-cell transcriptomic atlas, PINNACLE provides 394,760 protein representations split across 156 cell type contexts from 24 tissues and organs. PINNACLE's contextualized representations of proteins reflect cellular and tissue organization and PINNACLE's tissue representations enable zero-shot retrieval of the tissue hierarchy. Pretrained PINNACLE's protein representations can be adapted for downstream tasks: to enhance 3D structure-based protein representations for important protein interactions in immuno-oncology (PD-1/PD-L1 and B7-1/CTLA-4) and to study the effects of drugs across cell type contexts. PINNACLE outperforms state-of-the-art, yet context-free, models in nominating therapeutic targets for rheumatoid arthritis and inflammatory bowel diseases, and can pinpoint cell type contexts that predict therapeutic targets better than context-free models. PINNACLE is a graph-based contextual AI model that dynamically adjusts its outputs based on biological contexts in which it operates.

4 Open Problems

4.1 Open Problems in Benchmarking Graph Learning via Synthetic Graph Generation

Bryan Perozzi (Google – New York, US, bperozzi@cs.stonybrook.edu)

Graph learning algorithms have attained state-of-the-art performance on many graph analysis tasks such as node classification, link prediction, and clustering. It has, however, become hard to track the field's burgeoning progress. One reason is due to the very small number of datasets used in practice to benchmark the performance of graph learning algorithms. This shockingly small sample size used for standard evaluations allows for only limited scientific insight into how graph learning models perform.

My group has been working on addressing this deficiency through the use of synthetic graph generation for evaluating graph learning models [1]. This approach consists of generating synthetic graphs, and contrasting the behavior of different graph learning algorithms in these controlled scenarios. Our prior work in this area (e.g. GraphWorld [2]) introduced the "GraphWorld paradigm"

1. Define graph metrics to quantify the "universe" of graphs generated by the framework
2. Define a Task (graph generator, feature generator, and prediction) tuple that fully defines a graph learning problem
3. Vary the graphs generated, in order to cover the "universe" as well as possible
4. Evaluate a wide range of graph learning models on the Task to characterize the response surface for each model

Naturally, there is much more to do in the space! For example, our initial investigation focused on community-aware generators (like the Stochastic Block Model) for this evaluation. However, this design decision limited the scope of graphs which could be generated. Further analysis [3], has shown that better control of the degree distribution of the generated graph creates situations which further differentiate graph learning models. Aside from obvious improvements to the components of the framework above, this line of research enables new problem formulations. For example, we also have examined how privacy-aware graph generative models can generate anonymized graph datasets on which a model has similar performance to the real datasets [4]. This is a new and interesting regime for both graph generation and synthetic evaluation. Finally, in very recent work [5], we've illustrated how this style of synthetic analysis can also be used to quantify the reasoning capabilities of Large Language Models (LLMs), providing an interesting connection to the emerging field of Artificial Intelligence.

References

1 Anton Tsitsulin and Benedek Rozemberczki and John Palowitch and Bryan Perozzi. Synthetic Graph Generation to Benchmark Graph Learning. GLB'21, Graph Learning Benchmarks (WWW Workshops).

2 Palowitch, John and Tsitsulin, Anton and Mayer, Brandon and Perozzi, Bryan. GraphWorld: Fake Graphs Bring Real Insights for GNNs. KDD '22.

3 Yasir, Mustafa and Palowitch, John and Tsitsulin, Anton and Tran-Thanh, Long and Perozzi, Bryan. Examining the Effects of Degree Distribution and Homophily in Graph Learning Models. GLB'23, Graph Learning Benchmarks (KDD Workshops).

4 Yoon, Minji and Wu, Yue and Palowitch, John and Perozzi, Bryan and Salakhutdinov, Russ. Graph Generative Model for Benchmarking Graph Neural Networks. International Conference on Machine Learning, 2023.
5 Fatemi, Bahare and Halcrow, Jonathan and Perozzi, Bryan. Talk like a graph: Encoding graphs for large language models. arXiv preprint, 2023.

4.2 Causal Representation Learning

Elena Zheleva (University of Illinois – Chicago, US, ezheleva@uic.edu)

Causal representation learning refers to the discovery of high-level causal variables from low-level observational data and is an important problem in machine learning and causal inference. In machine learning, causal representation learning can help with robustness and learning reusable mechanisms [1]. In causal inference, it can help with discovering the causal mechanisms that gave rise to the observed data and identifying causal effects of interest. While initial work has been done in causal representation learning for IID data (e.g., [1, 2]), little research has been done on causal representation learning for graphs [1]. The proposed discussion is to identify a path towards causal graph representation learning, including challenges and open problems.

References

1 Schölkopf et al. Towards Causal Representation Learning. Proceedings of the IEEE 2021.
2 Lachapelle et al. Disentanglement via Mechanism Sparsity Regularization: A New Principle for Nonlinear ICA. CLeaR 2022.
3 Li et al. Out-Of-Distribution Generalization on Graphs: A Survey. Arxiv 2022.

4.3 Graphs and Matrix accelerators – opportunities and challenges?

Flavio Vella (University of Trento, IT, flavio.vella@unitn.it)

With the growing computational needs requested by deep learning and AI, new libraries and architectures have steadily improved the efficiency of dense matrix multiplication in recent years. Dense-matrix units, such as the Tensor Processing Unit (TPU) (TPU) [1], TensorCore (TC) [2], or, more recently, Advanced Matrix Extensions (AMX) with TMUL by Intel [3] among many others, are hardware accelerators designed to handle large volumes of multiply-accumulate operations efficiently.

The use of such accelerators for other computation is becoming a well-established practice, not only for designing newer building blocks (e.g, *scan and prefix-sum* [4]) and mixed-precision linear solvers for HPC scientific computing [5] but also for various other applications [6]. In the graph analytics realm, there are also preliminary studies on SRAM-centric architecture [7].

Focusing on matrix-accelerators is natural to think about graphs algorithms in terms of matrix operations over semirings [8, 9, 10]. For example, given a fixed semiring $\langle D, \bigoplus, \bigotimes, e\rangle$ (where D is the element set, $\bigoplus, \bigotimes$ the ring operations, and e identity element) and the adjacency matrix A of a graph G, we can compute connectivity, number of paths, shortest paths and graph matching as simple tensor operations.[1]

GraphBLAS potentially provides a suitable abstraction for expressing graph algorithms in terms of *semirings*, however, there are several issues and challenges to face in mapping the algorithm to the matrix accelerator, which range from the sparsity of the graphs to the algorithms we can express to enabling the usage of these newer architectures.

Challenge 1: Sparsity as usual.

As we delve into the field of graph analytics, the first challenge we encounter is how to efficiently handle sparsity. This issue remains prevalent in parallel architectures. Graphs are notoriously sparse or even hyper-sparse while matrix accelerators support dense block-based data structure or structured sparsity. The practice of blocking the adjacency matrix often results in suboptimal performance compared to traditional sparse data structures like CSR.

To mitigate this, two approaches are considered. The first aims to leverage the *inherent density* found in certain graph components (often hidden by their labelling). The second approach involves artificially increasing the density within a block through more sophisticated data structures [11, 12].

Regarding the “inherent density”, reordering algorithms and graph partitioning (see Table 1) can be used [13]. Trotter et al. [13] found that reordering based on graph partitioning provides better SpMV performance than the alternatives for the majority of matrices. However, both approaches are not specific to our purpose: traditional reordering methods (such as Saad’s rendering method etc. [14]) typically are used for finding dense blocks over the diagonal which leads to good properties for linear solvers, and partitioning methods find a cut over the edges by assuming a fixed number of partitions (you want to minimize the cost among the partitions, that in our context correspond to blocks). Nevertheless, matrix accelerators need small (e.g., 4×4) but really dense blocks (order of thousands for medium matrices) to obtain a significant improvement.

Preliminary works [15, 16] introduce a new family of reordering algorithms based on clustering where each cluster corresponds to a distinct block. This algorithm requires $\mathcal{O}(n^2)$ comparisons among rows in the worst case. The main issue of such an approach is still on the final density that depends on the graphs, therefore it seems that, for certain graphs, we need to use ad-hoc data structures that do not allow us to exploit existing vendor-based routines for tensor accelerators.

P1.1. Find a linear algorithm that provides theoretical guarantees on the final density.
P1.2. Find a reordering that fits a structured sparsity schema.

(To solve P1.1 hashing-based algorithms such as Local-Sensitive Hashing (LSH) may be considered [16].)

[1] connectivity → $\langle\{0,1\},\vee,\wedge,0\rangle$, number of paths → $\langle\mathbf{N},+,\cdot,0\rangle$, shortest paths → $\langle\mathbf{R}\cup\{+\infty\},\min,+,+\infty\rangle$, matching → $\langle\mathbf{R}\cup\{-\infty\},\max,+,-\infty\rangle$.

Table 1 State-of-the-art techniques.

Method	Parametric	Process Rectangular	Only symmetric	Objective function (target)	Archs
Reverse Cuthill–McKee [17]	×	×	×	Matrix Bandwidth	CPU
Approximate minimum degree [18]	×	×	×	Row Degree	CPU
Nested dissection[19]	×	×	×	Cholesky Factorization	CPU
Graph partitioning[19]	✓	×	✓	Edge Cut & Load Balance	CPU
Hypergraph partitioning[20]	✓	✓	×	Edge Cut & Load Balance	CPU
Gray code ordering[21]	×	✓	✓	Branch Mispredictions & Data Locality	CPU
Saad's algorithm[22]	×	✓	✓	Row Similarity	CPU

Challenge 2: Graph Algorithms that can benefit from matrix accelerators.

The second issue is related to what algorithm can exploit matrix accelerators. The matrix-centric formulation of typical traversal-based algorithms such as the Breadth-first search (BFS) requires Sparse Matrix-Vector multiplication among A and the source vector B. On matrix accelerators with large capacity for storing the operands, this is not efficient. Other algorithms such as Triangle Counting or unweighted Single(Multi)-source shortest paths fit better (B is this case is the matrix where each column is a search from different sources). However, the mapping is not as straightforward as it seems to be. Firstly, the second operand is typically sparse both on triangle-based (especially for the implementation based on triangular matrices [23]). The consequence is if we apply the reordering to A (adjacency Matrix of G) we need to apply the same permutation to B to make a coherent product. Alternatively, B should be represented in its dense blocked format with the consequence of obtaining poor performance.

Moreover, weighted graphs introduce further difficulties. The issue is the mapping of the tropical semirings (e.g. the $\langle \mathbf{R} \cup \{+\infty\}, \min, +, +\infty\rangle$ and $\langle \mathbf{R} \cup \{-\infty\}, \max, +, -\infty\rangle$ required by the weighted graph computation) on the matrix accelerator that natively only supports a simple algebra $+/\cdot$ (i.e. $\langle \mathbf{N}, +, \cdot, 0\rangle$). In the general case, which means we assume G with arbitrary weights, this could lead to an approximation version of the algorithm. However, within some constraints (e.g., where the weights are positive integers), it is possible to have the exact algorithm through a graph subdivision. Formalize this and find the constraints that have not been solved yet.

So, to summarize, the open problems are:

P2.1. How to design efficient primitives to solve unweighted graph problems by using matrix accelerators.

P2.2. Weighted graphs do not fit the algebraic properties exposed by matrix accelerators, the problem is how to "approximate" a semiring over the LA.

For P2.2. we need to explore theoretical foundations on the equivalences between graphs and matrix [24].

Challenge 3: Dynamic graphs and semirings from a GraphDB perspective.

Modern graph databases employ efficient graph engines to perform queries, with recent interest growing in matrix-centric graph engines. For example, RedisDB [25] adopts a GraphBLAS-based engine.

Databases typically serve multiple queries, including nested ones, and often require or rely on similar information. Currently, query languages translate operations into linear algebra expressions and optimize them using fundamental properties, such as the associative and distributive nature of matrix multiplication. Similarly, can we apply additional optimizations, akin to those long used in relational algebra? The second problem arises when the graph's structure changes. Optimizing in the context of dynamic operations, such as edge insertion/deletion or weight updates, remains a topic of exploration, as highlighted in various works by Giuseppe F. Italiano et al. To the best of our knowledge, there have been no significant enhancements in the processing of dynamic graphs in a truly algebraic form. We have recently started working on dynamic transitive closure and APSP over semirings.

References

1 Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In Proceedings of the 44th Annual International Symposium on Computer Architecture, pages 1–12. ACM, 2017.

2 Nvidia. Nvidia Tesla V100 GPU architecture. `https://www.nvidia.com/content/PDF/nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf`. Accessed: 2018-04-13.

3 Mummidi, Chandra Sekhar and Ferreira, Victor C and Srinivasan, Sudarshan and Kundu, Sandip. Highly Efficient Self-Checking Matrix Multiplication on Tiled AMX Accelerators. ACM Transactions on Architecture and Code Optimization, 2023.

4 Dakkak, Abdul and Li, Cheng and Xiong, Jinjun and Gelado, Isaac and Hwu, Wen-mei. Accelerating reduction and scan using tensor core units. Proceedings of the ACM International Conference on Supercomputing, 2019.

5 Ltaief, Hatem and Hong, Yuxi and Dabah, Adel and Alomairy, Rabab and Abdulah, Sameh and Goreczny, Chris and Gepner, Pawel and Ravasi, Matteo and Gratadour, Damien and Keyes, David. Steering Customized AI Architectures for HPC Scientific Applications. International Conference on High Performance Computing, 2023.

6 Burchard, Luk and Zhao, Max Xiaohang and Langguth, Johannes and Buluç, Aydın and Guidi, Giulia. Space Efficient Sequence Alignment for SRAM-Based Computing: X-Drop on the Graphcore IPU. Association for Computing Machinery, 2023.

7 Burchard, Luk and Cai, Xing and Langguth, Johannes. iPUG for Multiple Graphcore IPUs: Optimizing Performance and Scalability of Parallel Breadth-First Search. 2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC).

8 Kepner, Jeremy and Gilbert, John. Graph Algorithms in the Language of Linear Algebra. Society for Industrial and Applied Mathematics, 2011.

9 Solomonik, Edgar and Besta, Maciej and Vella, Flavio and Hoefler, Torsten. Scaling betweenness centrality using communication-efficient sparse matrix multiplication. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2017.

10 Yiran Chen and Yuan Xie and Linghao Song and Fan Chen and Tianqi Tang. A Survey of Accelerator Architectures for Deep Neural Networks. Engineering, 2020.

11 Lu, Yuechen and Liu, Weifeng. DASP: Specific Dense Matrix Multiply-Accumulate Units Accelerated General Sparse Matrix-Vector Multiplication. Association for Computing Machinery, 2023.

12 Zachariadis, Orestis and Satpute, Nitin and Gómez-Luna, Juan and Olivares, Joaquín. Accelerating sparse matrix–matrix multiplication with GPU Tensor Cores. Computers & Electrical Engineering, 2020.

13 Trotter, James D. and Ekmekçibaşı, Sinan and Langguth, Johannes and Torun, Tugba and Düzakın, Emre and Ilic, Aleksandar and Unat, Didem. Bringing Order to Sparsity: A Sparse Matrix Reordering Study on Multicore CPUs. Association for Computing Machinery, 2023.

14 Chen, Jie and Saad, Yousef. Finding dense subgraphs for sparse undirected, directed, and bipartite graphs. IEEE Trans. Knowl. Data Eng., submitted

15 Labini, Paolo Sylos and Bernaschi, Massimo and Nutt, Werner and Silvestri, Francesco and Vella, Flavio. Blocking Sparse Matrices to Leverage Dense-Specific Multiplication. 2022 IEEE/ACM Workshop on Irregular Applications: Architectures and Algorithms (IA3).

16 Jiang, Peng. Hong, Changwan. and Agrawal, Gagan. 2020. A novel data transformation and execution strategy for accelerating sparse matrix multiplication on GPUs. In Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '20).

17 Liu, Wai-Hung and Sherman, Andrew H. Comparative Analysis of the Cuthill–McKee and the Reverse Cuthill–McKee Ordering Algorithms for Sparse Matrices. SIAM Journal on Numerical Analysis, 1976.

18 Amestoy, Patrick R. and Davis, Timothy A. and Duff, Iain S. Algorithm 837: AMD, an Approximate Minimum Degree Ordering Algorithm. Association for Computing Machinery, 2004.

19 Karypis, George and Kumar, Vipin. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. Society for Industrial and Applied Mathematics, 1998.

20 Catalyurek, U.V. and Aykanat, C. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. IEEE Transactions on Parallel and Distributed Systems.

21 Zhao, Haoran and Xia, Tian and Li, Chenyang and Zhao, Wenzhe and Zheng, Nanning and Ren, Pengju. Exploring Better Speculation and Data Locality in Sparse Matrix-Vector Multiplication on Intel Xeon. 2020 IEEE 38th International Conference on Computer Design (ICCD).

22 Saad, Yousef. Finding Exact and Approximate Block Structures for ILU Preconditioning. SIAM Journal on Scientific Computing.

23 Azad, Ariful and Buluç, Aydin and Gilbert, John. Parallel triangle counting and enumeration using matrix algebra. Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International.

24 Williams, Virginia Vassilevska and Williams, R Ryan. Subcubic equivalences between path, matrix, and triangle problems. Journal of the ACM (JACM).

25 Cailliau, Pieter and Davis, Tim and Gadepally, Vijay and Kepner, Jeremy and Lipman, Roi and Lovitz, Jeffrey and Ouaknine, Keren. Redisgraph graphblas enabled graph database. 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW).

5 Working Groups

5.1 Working Group on Causal Representation Learning

Elena Zheleva (University of Illinois – Chicago, US, ezheleva@uic.edu)
Michelle Li (Harvard University – Boston, US, michelleli@g.harvard.edu)

Causal representation learning refers to the discovery of high-level causal variables from low-level observational data and is an important problem in machine learning and causal inference. In machine learning, causal representation learning can help with robustness and learning reusable mechanisms [1]. In causal inference, it can help with discovering the causal mechanisms that gave rise to the observed data and with identifying causal effects of interest. While initial work has been done in causal representation learning for IID data (e.g., [1, 2]), little research has been done on causal representation learning for graphs [3, 4].

Many real-world systems can be represented as graphs in which nodes represent entities of interest, which can be of various types, and edges represent interactions between these entities. Some examples include social networks, protein interaction networks, molecular structure networks, and drug-protein-disease networks [4]. Graph representation learning has facilitated discoveries across real-world applications on structured data [4, 6]. However, despite the prevalence of causal relationships in real-world systems, existing graph representation learning algorithms are unable to extrapolate causal mechanisms.

References

1 Schölkopf et al. Towards Causal Representation Learning. Proceedings of the IEEE 2021.
2 Lachapelle et al. Disentanglement via Mechanism Sparsity Regularization: A New Principle for Nonlinear ICA. CLeaR 2022.
3 Li et al. Out-Of-Distribution Generalization on Graphs: A Survey. Arxiv 2022.
4 Li et al. Graph representation learning in biomedicine and healthcare. Nature Biomedical Engineering 2022. PDF.
5 Chalupka, Perona, Eberhardt. Multi-level cause-effect systems. AISTATS 2016.
6 Hamilton, William L., Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. IEEE Data Engineering Bulletin 2017.

5.2 Working Group on Making Algorithms Research FAIR

David A. Bader (NJIT – Newark, US, bader@njit.edu)
Hannah Bast[2] *(University of Freiburg, DE, bast@informatik.uni-freiburg.de)*
Ümit V. Çatalyürek (Georgia Institute of Technology – Atlanta, US, umit@gatech.edu)
Joachim Giesen[3] *(Friedrich Schiller University Jena, DE, joachim.giesen@uni-jena.de)*
Kathrin Hanauer (Universität Wien, AT, kathrin.hanauer@univie.ac.at)
Ulrich Meyer (Goethe University – Frankfurt am Main, DE, umeyer@cs.uni-frankfurt.de)
Henning Meyerhenke (Humboldt-Universität zu Berlin, DE, meyerhenke@hu-berlin.de)
Cynthia A. Phillips (Sandia National Labs, Albuquerque, US, caphill@sandia.gov)
Ilya Safro (University of Delaware, Newark, US, isafro@udel.edu)

The need for efficient and tailor-made computer applications is dramatically increasing across all scientific areas. Even if research teams in application domains are acquiring their own computing expertise, they will still rely on powerful building blocks developed by experts. This is especially true for the fundamental algorithmic problems behind their applications – and is even more pronounced when these algorithms ought to exploit special hardware characteristics. In principle, algorithms research can provide these building blocks, but the relevant algorithms are often hard to find or understand or do not match the application needs well. Moreover, implementations are often missing or hard to employ. Users then usually resort to readily available or ad hoc solutions, often with markedly suboptimal performance.

5.2.1 Discussed Problems

Inspired by and based on ideas of an already ongoing initiative in Germany, this working group discussed the promotion of the goal to bridge this long-standing gap and make algorithms research more FAIR = Findable, Accessible, Interoperable, and Reusable; a set of principles originally defined for research data.

5.2.2 Possible Approaches

We argue that the natural way to achieve this is through problem abstractions with the following properties. Each abstraction should be lean, yet cover both a large body of algorithms research and a wide variety of applications. It should be comprehensible to a non-expert, remain stable over time, and come with an implementation.

5.2.3 Conclusions

It is intended to publish a position paper on the subject which presents success stories of the past and the main ideas how to achieve FAIRness in algorithmic research at large in the future.

[2] This co-author was not a seminar participant, but was invited to join the working group after the seminar.
[3] This co-author was not a seminar participant, but was invited to join the working group after the seminar.

5.3 GNNs for smaller kernels, finding the one rule to reduce them all

Kenneth Langedal (University of Bergen, NO, kenneth.langedal@uib.no)
Fredrik Manne (University of Bergen, NO, fredrik.manne@uib.no)
Ernestine Großmann (Universität Heidelberg, DE, e.grossmann@informatik.uni-heidelberg.de)
Christian Schulz (Universität Heidelberg, DE, christian.schulz@informatik.uni-heidelberg.de)
Matthias Schimek (KIT – Karlsruher Institut für Technologie, DE, schimek@kit.edu)
Fabian Brandt-Tumescheit (HU Berlin, DE, fabian.brandt-tumescheit@hu-berlin.de)

Finding the maximum weighted independent set (MWIS) in a graph is a fundamental NP-hard problem. Given an undirected graph $G = (V, E)$ with node weights $w : V \to \mathbb{R}^+$, MWIS is the problem of finding a set $\mathcal{I} \subseteq V$ of pair-wise nonadjacent vertices with the largest possible weight $\sum_{u \in \mathcal{I}} w(u)$. For both exact and heuristic solvers, reduction rules are crucial for tackling large instances. These reduction rules reduce the problem instance while ensuring that an optimal solution on the reduced instance can be lifted to an exact solution for the original graph.

We consider how graph neural networks (GNNs) can be used for solving the MWIS problem. In particular, we investigate how GNNs can be used to both design and apply reduction rules. Our discussions during the Dagstuhl Seminar 23491 mainly focused on three directions for further work.

What are the capabilities and limitations of current GNN architectures for learning existing reduction rules? Known reduction rules have a wide range of complexity [2]. The most straightforward rules compare weights between vertices or neighborhoods. More complicated rules search for patterns such as twins or dominated vertices. Eventually, some reduction rules even require solving the MWIS problem on a subgraph. As a first step, it makes sense to investigate what the existing GNN architectures are capable of in terms of learning known reduction rules. Furthermore, what features should be precomputed as initial input features?

Can we find new reduction rules with the help of a trained GNN model? Reduction rules should be provably exact, meaning we cannot rely on a machine-learning model to decide which vertices to reduce directly. However, a trained model could highlight interesting structures in otherwise irreducible graphs, where a human observer could then derive new reduction rules. We cannot train directly for this task since we do not know what structures could be interesting. Nevertheless, we could use a pre-trained model for predicting vertices that known reduction rules can reduce. To be clear, the outputs from the model should be negative for every vertex here since the graph is irreducible using known reductions. However, we can still rank the outputs and look into the vertices closest to a positive prediction. Alternatively, we could also train a model that predicts what vertices are part of some optimal solution. In this case, we would investigate the vertices that receive the most certain prediction, as in closest to positive or negative. Such models have been trained previously [3].

Can we train a GNN to reduce graphs further than what is currently possible? As mentioned earlier, some reduction rules must be applied cautiously due to their computational cost. Current implementations rely on simple heuristics to decide when to use these rules. Therefore, letting a GNN model decide instead could be a promising direction.

In addition to reduction rules that decrease the size of the problem, there are also transformations that increase the size of the graph [1]. The idea behind these rules is that alternating between phases where we reduce and expand the graph can eventually lead to smaller irreducible graphs. Combined with when to apply expensive rules, this becomes its own search problem. It also fits nicely into the reinforcement learning method with a clear set of actions, environment, and rewards. More specifically, an action would be a rule and area of the graph to apply it, and the goal would be to reduce the graph as much as possible. Alternatively, the goal could be to reduce it with the least amount of work.

References

1 Alexander Gellner, Sebastian Lamm, Christian Schulz, Darren Strash, and Bogdán Zaválnij. "Boosting Data Reduction for the Maximum Weight Independent Set Problem Using Increasing Transformations". In: *2021 Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM. 2021, pp. 128–142.

2 Sebastian Lamm, Christian Schulz, Darren Strash, Robert Williger, and Huashuo Zhang. "Exactly Solving the Maximum Weight Independent Set Problem on Large Real-World Graphs". In: *2019 Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM. 2019, pp. 144–158.

3 Kenneth Langedal, Johannes Langguth, Fredrik Manne, and Daniel Thilo Schroeder. "Efficient Minimum Weight Vertex Cover Heuristics using Graph Neural Networks". In: *20th International Symposium on Experimental Algorithms (SEA 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2022.

Participants

- David A. Bader
NJIT – Newark, US
- Maciej Besta
ETH Zürich, CH
- Aleksandar Bojchevski
Universität Köln, DE
- Konstantinos Bougiatiotis
Demokritos – Athens, GR
- Fabian Brandt-Tumescheit
HU Berlin, DE
- Ümit V. Çatalyürek
Georgia Institute of Technology – Atlanta, US
- Ayan Chatterjee
Northeastern University – Boston, US
- Mikhail Galkin
Intel AI Lab – San Diego, US
- Ernestine Großmann
Universität Heidelberg, DE
- Kathrin Hanauer
Universität Wien, AT
- Mariya Ishteva
KU Leuven – Geel, BE
- George Karypis
University of Minnesota – Minneapolis, US
- Christine Klymko
LLNL – Livermore, US
- Danai Koutra
University of Michigan – Ann Arbor, US & Amazon, US
- Kenneth Langedal
University of Bergen, NO
- Johannes Langguth
Simula Research Laboratory – Oslo, NO
- Michelle Li
Harvard University – Boston, US
- Fredrik Manne
University of Bergen, NO
- Ulrich Carsten Meyer
Goethe University – Frankfurt am Main, DE
- Henning Meyerhenke
HU Berlin, DE
- Davide Mottin
Aarhus University, DK
- Richard Peng
University of Waterloo, CA
- Bryan Perozzi
Google – New York, US
- Cynthia A. Phillips
Sandia National Labs – Albuquerque, US
- Alex Pothen
Purdue University – West Lafayette, US
- Ilya Safro
University of Delaware – Newark, US
- Michael Schaub
RWTH Aachen, DE
- Matthias Schimek
KIT – Karlsruher Institut für Technologie, DE
- Christian Schulz
Universität Heidelberg, DE
- Christian Sohler
Universität Köln, DE
- Frank Takes
Leiden University, NL
- Elena Zheleva
University of Illinois – Chicago, US