DAGSTUHL
REPORTS

Volume 13, Issue 8, August 2023

**ISSN 2192-5283**

*Published online and open access by*
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at https://www.dagstuhl.de/dagpub/2192-5283

*Publication date*
March, 2024

*Bibliographic information published by the Deutsche Nationalbibliothek*
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at `https://dnb.d-nb.de`.



Digital Object Identifier: 10.4230/DagRep.13.8.i

*Aims and Scope*
The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.
In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:

- an executive summary of the seminar program and the fundamental results,
- an overview of the talks given during the seminar (summarized as talk abstracts), and
- summaries from working groups (if applicable).

This basic framework can be extended by suitable contributions that are related to the program of the seminar, e. g. summaries from panel discussions or open problem sessions.



*Contact*
Schloss Dagstuhl – Leibniz-Zentrum für Informatik
Dagstuhl Reports, Editorial Office
Oktavie-Allee, 66687 Wadern, Germany
reports@dagstuhl.de
https://www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 23331

# Recent Trends in Graph Decomposition

**George Karypis[*1], Christian Schulz[*2], Darren Strash[*3], Deepak Ajwani[4], Rob H. Bisseling[5], Katrin Casel[6], Ümit V. Çatalyürek[7], Cédric Chevalier[8], Florian Chudigiewitsch[9], Marcelo Fonseca Faraj[10], Michael Fellows[11], Lars Gottesbüren[12], Tobias Heuer[13], Kamer Kaya[14], Jakub Lacki[15], Johannes Langguth[16], Xiaoye Sherry Li[17], Ruben Mayer[18], Johannes Meintrup[19], Yosuke Mizutani[20], François Pellegrini[21], Fabrizio Petrini[22], Frances Rosamond[23], Ilya Safro[24], Sebastian Schlag[25], Roohani Sharma[26], Blair D. Sullivan[27], Bora Uçar[28], and Albert-Jan Yzelman[29]**

1 **University of Minnesota – Minneapolis, US.** karypis@umn.edu
2 **Universität Heidelberg, DE.** christian.schulz@informatik.uni-heidelberg.de
3 **Hamilton College – Clinton, US.** dstrash@hamilton.edu
4 **University College Dublin, IE.**
5 **Utrecht University, NL.**
6 **Humboldt University Berlin, DE.**
7 **Georgia Institute of Technology – Atlanta, US & Amazon Web Services, US.**
8 **CEA, DAM, DIF – Arpajon, FR.**
9 **Universität zu Lübeck, DE.**
10 **Universität Heidelberg, DE.**
11 **University of Bergen, NO.**
12 **Karlsruher Institut für Technologie, DE.**
13 **Karlsruher Institut für Technologie, DE.**
14 **Sabanci University – Istanbul, TR.**
15 **Google – New York, US.**
16 **Simula Research Laboratory – Oslo, NO & University of Bergen, NO.**
17 **Lawrence Berkeley National Laboratory, US.**
18 **Universität Bayreuth, DE.**
19 **Technische Hochschule Mittelhessen – Gießen, DE.**
20 **University of Utah – Salt Lake City, US.**
21 **University of Bordeaux, FR.**
22 **Intel Labs – Menlo Park, US.**
23 **University of Bergen, NO.**
24 **University of Delaware – Newark, US.**
25 **Sunnyvale, US.**
26 **MPI für Informatik – Saarbrücken, DE.**
27 **University of Utah – Salt Lake City, US.**
28 **CNRS and LIP ENS de Lyon, FR.**
29 **Huawei Technologies – Zürich, CH.**

[*] Editor / Organizer

Recent Trends in Graph Decomposition, *Dagstuhl Reports*, Vol. 13, Issue 8, pp. 1–45
Editors: George Karypis, Christian Schulz, and Darren Strash

Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

**Abstract**

This report documents the program and the outcomes of Dagstuhl Seminar 23331 "Recent Trends in Graph Decomposition", which took place from 13. August to 18. August, 2023. The seminar brought together 33 experts from academia and industry to discuss graph decomposition, a pivotal technique for handling massive graphs in applications such as social networks and scientific simulations. The seminar addressed the challenges posed by contemporary hardware designs, the potential of deep neural networks and reinforcement learning in developing heuristics, the unique optimization requirements of large sparse data, and the need for scalable algorithms suitable for emerging architectures. Through presentations, discussions, and collaborative sessions, the event fostered an exchange of innovative ideas, leading to the creation of community notes highlighting key open problems in the field.

**Seminar** August 13–18, 2023 – https://www.dagstuhl.de/23331
**2012 ACM Subject Classification** Theory of computation → Design and analysis of algorithms
**Keywords and phrases** combinatorial optimization, experimental algorithmics, parallel algorithms
**Digital Object Identifier** 10.4230/DagRep.13.8.1
**Funding** We acknowledge support by DFG grant SCHU 2567/5-1.
*Deepak Ajwani*: Supported in part by a grant from Science Foundation Ireland under Grant number 18/CRT/6183.
*Johannes Meintrup*: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 379157101.
**Acknowledgements** We acknowledge Dagstuhl Seminar 23331 on "Recent Advances in Graph Decomposition".

## 1 Executive Summary

*George Karypis (University of Minnesota – Minneapolis, US)*
*Christian Schulz (Universität Heidelberg, DE)*
*Darren Strash (Hamilton College – Clinton, US)*



Large networks are useful in a wide range of applications. Sometimes problem instances are composed of billions of entities. Decomposing and analyzing these structures helps us gain new insights about our surroundings. Even if the final application concerns a different problem (such as traversal, finding paths, trees, and flows), decomposing large graphs is often an important subproblem for complexity reduction or parallelization. With even larger instances in applications such as scientific simulation, social networks, or road networks, graph decomposition becomes even more important, multifaceted, and challenging. The seminar was an international forum to discuss recent trends as well as to set new goals and new directions in this research area. The goal of this Dagstuhl Seminar was to bring algorithmic researchers from different communities together who implement, optimize, and/or experiment with algorithms running on large data sets or use techniques from the area frequently, thereby stimulating an exchange of ideas and techniques. The seminar focus was on graph decomposition. We chose the main topics of our seminar to bring experts together from a wide range of areas to tackle some of the most pressing open problems in the area of graph decomposition:

**Hardware Design for Dealing with Graphs.** Modern processors are optimized for computations that are floating point intensive and have regular memory accesses. Computations performed by sparse graph algorithms do not benefit from such optimizations. To address this mismatch, new processor and system architectures are being developed.

**Beyond Smart Heuristics.** In heuristic approaches, improvements in solution quality are often the result of a significant research effort. In recent years, to reduce this effort, develop better heuristics, and ultimately find better solutions, researchers have started developing approaches that use deep neural networks and reinforcement learning in order to learn those heuristics in an end-to-end fashion.

**Formulations.** Applications that process large sparse data generally have a unique set of optimization requirements for achieving the best performance. Parallelizing such applications on different architectures and/or using different frameworks introduces new performance issues that pertain to these architectures and frameworks. While graphs offer a rich ground for modeling such problems with different requirements, traditional graph decomposition tools may fall short to target those specific issues.

**Scalable Parallel Algorithms for Future Emerging Architectures.** Scalable high quality graph partitioning (with quality comparable to sequential partitioners) remains an open problem. With the advent of exascale machines with millions of processors and possibly billions of threads, the situation is further aggravated. Moreover, traditional "flat" partitions of graphs for processing on such machines implies a huge number of blocks. Efficient implementation is also a big issue since complex memory hierarchies and heterogeneity (e.g., GPUs or FPGAs) make the implementation complicated.

## Summary of Seminar

The seminar convened 33 distinguished participants from both academic and industrial sectors worldwide. The majority of attendees arrived on Sunday evening. In total, the seminar showcased 19 presentations. The proceedings began with a welcome and ice-breaker session, facilitating introductions among participants. There were two dedicated sessions to address open problems. Ample time was set aside for collaboration. Additionally, a social event in the form of a hike was organized for attendees. The majority of the presentations were concise, spanning approximately 30 minutes, while a select few were of longer duration. Active discussions were a hallmark of this seminar, both during and after presentations, with highlights captured below. Spontaneous working groups emerged during the event, and their details are documented in the subsequent sections. During the week, we collaborated on a joint document that captures a large variety of open problems in the field. These so-called community notes which will be released/published in a separate document. Throughout and following the sessions, participants engaged in lively conversations. These discussions enriched the entire event and set it apart from typical conference formats. Beyond the research-focused activities, numerous social events like board games, poker, music evenings, hiking, and ping pong added a fun dimension. This made the seminar an excellent networking venue, especially for the many attendees experiencing Schloss Dagstuhl for the first time.

## 2 Table of Contents

# 3 Overview of Talks

## 3.1 Scalable Graph Clustering at Google

*Jakub Lacki (Google – New York, US)*



Graph clustering has numerous applications in classification, near-duplicate detection, data partitioning, community detection and privacy. In this talk we present a graph clustering library which supports all of these use cases and powers over a hundred applications at Google. One of the main design goals of the library is high scalability. In the offline setting, the library can handle up to trillion-edge graphs by leveraging distributed processing, or up to 100 billion edges using single-machine parallelism. Moreover, the library contains online algorithms that can update clustering with sub-second latency upon vertex insertions.

**Discussions.** There was a long discussion about applying ideas of TeraHAC to solving maximum weighted matching. In particular, both algorithms in the exact version find edges xy which are the highest weight incident edges for both x and y. In the approximate version of TeraHAC we relax this and allow almost-highest weight incident edges. We discussed whether this can be used to speed up greedy maximum weight matching and it looks like the idea may apply. However, showing asymptotic improvement in the running time bound seems to be somewhat harder. Another question that came up was how do we evaluate a near-dup clustering? We often use human raters that produce verdicts of the form: items x and y should/should not be in the same cluster. Lastly, when deciding about the similarity of two clusters x and y do we only look at edges between them, or do we also look at edges that go from x to outside of y (and vice versa)? Only edges between the clusters are considered.

## 3.2 Local Objectives for Graph Clustering

*Katrin Casel (HU Berlin, DE)*



In many situations clustering tasks do not involve (just) a global optimization goal, but (additionally) request local properties for clusters. Such local objectives are often particularly challenging. This talk gives some examples of such objectives for graph clustering with a brief overview of what is known and what is (surprisingly) open. In particular, these examples are connectivity and fairness as local objectives that are added to a global objective, density as difficult-to-check local objective, and chromatic edges where locally only the most prominent color counts.

**Discussions.** The talk gave rise to some new open questions: What happens when the dense partition objective is combined with a lower bound on the cluster size? Is connected partition in planar graphs easier than in general graphs? Discussions afterwards revealed: MaxMin/MinMax Balanced Partition inherits several hardness results from the Equitable Partition problem. Further, MaxMin/MinMax Balanced Partition is closely related to the Neighborhood Partitioning problem. The implications of this relationship remain to be investigated. Future studies should investigate, in particular from a computational point of view, which notion of fairness is most reasonable for Correlation Clustering (vertex vs edge fairness).

## 3.3 Parallel Incremental Clustering Algorithms for Massive Dynamic Graphs

*Johannes Langguth (Simula Research Laboratory – Oslo, NO)*



We consider the problem of incremental graph clustering where the graph to be clustered is given as a sequence of disjoint subsets of the edge set. The problem appears when dealing with graphs that are created over time, such as online social networks where new users appear continuously, or protein interaction networks when new proteins are discovered. For very large graphs, it is computationally too expensive to repeatedly apply standard clustering algorithms. Instead, algorithms whose time complexity only depends on the size of the incoming subset of edges in every step are needed. At the same time, such algorithms should find clusterings whose quality is close to that produced by offline algorithms. We discuss the computational model and present an incremental clustering algorithm, along with its parallel implementation. The scalability results suggest that our method is well suited for clustering massive graphs with acceptable running times while retaining a large fraction of the clustering quality.

**Discussions.** The performance of the incremental clustering algorithm depends heavily on the structure of the parts. So far, test instances have only been created by assigning edges to parts uniformly at random. Michael Fellows pointed out that different probability distributions would likely match actual growth of online social networks. Tobias Heuer pointed out that NCLiC does not actually compute modularity gains and that modularity retention could probably be improved by modifying the algorithm, although this might come at the cost of increased runtime. Jakub Lacki suggested to extend the incremental clustering algorithm from modularity maximization to correlation clustering which is certainly possible since the link-counting approach should work with many different clustering objectives. Ruben Mayer investigated the possibility of using NCLiC for streaming edge partitioning. Further discussion also suggested a way to replace the current way of randomly skipping expensive updates with an exact method by finding a suitable data structure.

## 3.4 Parameterized Approximation Schemes for Clustering with General Norm Objectives

*Roohani Sharma (MPI für Informatik – Saarbrücken, DE)*



We consider the well-studied algorithmic regime of designing a $(1+\epsilon)$-approximation algorithm for a $k$-clustering problem that runs in time $f(k,\epsilon)\text{poly}(n)$. Our main contribution is a clean and simple EPAS that settles more than ten clustering problems (across multiple well-studied objectives as well as metric spaces) and unifies well-known EPASes. Our algorithm gives EPASes for a large variety of clustering objectives (for example, $k$-means, $k$-center, $k$-median, priority $k$-center, $\ell$-centrum, ordered $k$-median, socially fair $k$-median aka robust $k$-median, or more generally monotone norm $k$-clustering) and metric spaces (for example, continuous high-dimensional Euclidean spaces, metrics of bounded doubling dimension, bounded treewidth

metrics, and planar metrics). Key to our approach is a new concept that we call bounded $\epsilon$-scatter dimension–an intrinsic complexity measure of a metric space that is a relaxation of the standard notion of bounded doubling dimension.

**Discussions.** After the talk, with Darren and Katrin we briefly discussed about the implementability of the presented algorithm. We summarized that though the power of the result in theory is that it gives a generalized framework for dealing with various clustering objective functions simultaneously, in order to implement it faster we need to do the proposed steps (for example finding a new center of the partial cluster) with a metric-sensitive algorithm.

## 3.5 Leveraging Learning-to-prune and reinforcement learning for solving combinatorial optimisation problems

*Deepak Ajwani (University College Dublin, IE)*



In recent years, machine learning (ML) techniques are being increasingly used for solving combinatorial optimisation problems. This often requires a deep integration between techniques from optimisation literature, algorithm engineering and machine learning. For instance, while the optimisation and algorithmic literature guides the feature engineering in learning models, the learning models can guide crucial design steps in exact MILP solvers as well as heuristics. Specifically, I would like to talk about the research done in my group on a range of combinatorial optimisation problems in graphs such as variants of vehicle routing problems, Max Cut, Max Clique, Steiner tree etc. We have used a combination of supervised techniques such as learning-to-prune, reinforcement learning techniques such as CombOptZero and some recent unsupervised learning techniques to compute high quality solutions to optimisation problems in an efficient and scalable manner. In addition, we have also explored if learning techniques can speed up local search heuristics.

**Discussions.** There seemed to be a general agreement about my main argument that in order for a learning technique to (i) have a simple, interpretable architecture, (ii) generalise to larger sizes and (iii) work effectively with limited training data, it will likely have to be integrated with techniques from optimisation and algorithm engineering communities. The questions were mostly geared towards better understanding of the learning models and the associated details, such as how to label the data in case of multiple optimal solutions and what to do with integer linear programs that are non-binary etc. There were some discussions on how to apply these techniques to reduce the problem size and improve the refinement part (some of this is also noted in the open problem section). There were also discussions on how effectively reduction rules and kernelization techniques can be leveraged in the learning frameworks that I presented. Also, I was pointed to interesting references in this area such as Frederic Manne's work on using GNN for learning vertex cover [1], the linked paper on heuristics for hitting set[2] and on "Graph Partitioning and Sparse Matrix Ordering using Reinforcement Learning and Graph Neural Networks"[3].

[1] https://drops.dagstuhl.de/opus/volltexte/2022/16546/
[2] https://epubs.siam.org/doi/pdf/10.1137/1.9781611977042.17
[3] https://arxiv.org/abs/2104.03546

## 3.6 Using Steiner Trees in Hypergraph Partitioning

*Tobias Heuer (Karlsruhe Institute of Technology, DE)*



Minimizing wire-lengths is one of the most important objectives in the realization of modern circuits. The design process involves initially placing the logical units (cells) of a circuit onto a physical layout, and subsequently routing the wires to connect the cells. Hypergraph partitioning (HGP) has been long used as a placement strategy in this process. However, it has been replaced by other methods due to limitations of existing objective functions for HGP, which only minimizes wire-lengths implicitly. In this talk, we present a novel HGP formulation that maps a hypergraph $H$, representing a logical circuit, onto a routing layout represented by a weighted graph $G$. The objective is to minimize the total length of all wires induced by the hyperedges of $H$ on $G$. To capture wire-lengths, we compute minimal Steiner trees – a metric commonly used in routing algorithms. For this formulation, we present a direct $k$-way multilevel algorithm that we integrate into the shared-memory hypergraph partitioner Mt-KaHyPar. Mt-KaHyPar is a highly scalable partitioning algorithm that achieves the same solution quality as the best sequential algorithms, while being an order of magnitude faster with only ten threads. Our experiments demonstrate that our new algorithm achieves an improvement in the Steiner tree metric by 7% (median) on VLSI instances when compared to the best performing partitioning algorithm that optimizes the mapping in a postprocessing step. Although computing Steiner trees is an NP-hard problem, we achieve this improvement with only a 2–3 times slowdown in partitioning time compared to optimizing the connectivity metric.

**Discussions.** In the talk, Tobias used wire-length minimization as the main objective for VLSI design and motivation for my novel hypergraph partitioning formulation. However, attendees rightly noted that this is just one of several important metrics, such as, e.g., the perimeter of bounding boxes around nets. Additionally, some mentioned that placement challenges have been addressed using reinforcement learning recently.

## 3.7 Directed Acyclic Partitioning from Graphs to Hypergraphs

*Ümit V. Çatalyürek (Georgia Institute of Technology – Atlanta, US)*



Data transfer continues to be the biggest obstacle to efficient computation. The de facto abstraction for modeling computations has been directed acyclic graphs (DAGs). When scheduling computational tasks, an effective load balance and data locality trade-off is required. The ordering and mapping of the DAG's vertices (i.e., tasks) to computational resources are significantly benefited by acyclicity. As a result, it is preferable to maintain acyclicity at various levels of computation. In this talk, we demonstrate how acyclic partitioning of DAGs – partitioning where the inter-part edges of vertices from different parts should preserve an acyclic dependency among the parts – can be investigated to reduce redundant data movement on two-level memory settings. We also present the challenges of developing acyclic partitioning methods for directed hypergraphs, where they provide more elegant and accurate abstractions than graph counterparts.

**Discussions.** George Karypis was suggesting a potential alternative model for DAG partitioning for quantum simulation, with one fixed partition, to model the problem as edge-cut partitioning. In offline discussions, it turned out that the model doesn't fit the problem that is solved here.

## 3.8 Approximate Modular Decomposition

*Yosuke Mizutani (University of Utah, US)*



One fruitful avenue for designing efficient graph algorithms has been to parameterize using a variety of structural parameters (e.g. treewidth, clique-width) in addition to the natural parameter (i.e. solution size). Modular-width is a structural parameter introduced by Gajarsky et al. (2013) in an effort to generalize simpler notions on dense graphs while avoiding the intractability that often came with the existing clique-width parameter. Modular-width has several additional advantages – it can be computed in linear-time, and the associated modular decomposition (MD) tree has applications in visualization and parallel processing. Unfortunately, real-world graphs tend to have large modular-width. This leads us to the following natural questions: given a graph $G$, is there a useful notion of approximate MD trees that preserves fast computation, exhibits much lower widths, and enables solution of the downstream problems with quality guarantees? There are several possible avenues of attack. For example, we could define a graph editing problem: can we make a small number of changes to $G$ to produce a $G'$ with low modular-width? Another formulation would be to relax the definition of modules, which is likely closely interwoven with the idea of twin-width. Finally, we consider the option of taking a data-driven approach. What kind of graphs admit a nice approximation of modular decomposition?

## 3.9 Exact $k$-way sparse matrix partitioning

*Rob H. Bisseling (Utrecht University, NL)*

**Joint work of** Engelina Jenneskens



To minimize the communication in parallel sparse matrix-vector multiplication while maintaining load balance, we need to partition the sparse matrix optimally into $k$ disjoint parts, which is an NP-complete problem. We present an exact algorithm and an implementation called General Matrix Partitioner (GMP) based on the branch and bound (BB) method which partitions a matrix for any $k$, and we explore exact sparse matrix partitioning beyond bipartitioning. We also present an integer linear programming (ILP) model for the same problem, based on a hypergraph formulation. We used both methods to determine optimal 2,3,4-way partitionings for a subset of small matrices from the SuiteSparse Matrix Collection. To answer the question "How good is recursive bipartioning (RB)?", we used the exact results found for $k = 4$ to analyse the performance of RB with exact bipartitioning. Finally, we will discuss how exact methods inspire heuristic methods such as medium-grain partitioning and we will briefly touch on heuristic solvers such as Mondriaan and its hybrid distributed/shared-memory parallel version PMondriaan, which is currently under development.

**Discussions.** George Karypis asked whether there are worst-case results about the quality of recursive bipartitioning. There are papers like by Shang-Hua Teng and Horst Simon about possibly bad partitionings by RB, but the results of the talk show that they do not occur in practice for the small problems tested. Sherry Li asked about the realism of communication volume as a metric reflecting communication time. It is true that you want to minimise total communication volume as wel as per processor volume, and also the latency. Since data words communicated to the same processors can be combined into one message, the latency can be significantly reduced, and will only be a problem for a very large numbers of processors and a relatively small problem size. Tobias Heuer asked what splitting hyperedges exacly means. This feature is incorporated in the version of PMondriaan to be released, and it can be exemplified by converting hyperedges of size 3 into 3 edges of size 2, and then combining them with other edges and correcting where needed for the cost.

## 3.10 What Scotch cannot do yet

*François Pellegrini (University of Bordeaux, FR)*



In 30 years, the Scotch software has seen many increases in its capabilities. However, like its (friendly) competitors, its functional envelope is limited. In this talk, we will discuss problems that Scotch is not yet able to solve, and how to address them in the (near) future.

**Discussions.** By the end of the presentation, two open problems were presented. The first one relates to approximating distances in a family of recursively coarsened graphs, with respect to the distance computed in an initial fine graph. The aim of this method would be to avoid handling $O(p^2)$ distance matrices for all target graph vertices (where $p$ is the number of such vertices). During the discussion, it was mentioned that software for car traffic routing also use multilevel descriptions of the distances between locations [Sebastian], and that partial matrices are built, or local distance computations, are performed for each level. The case for tree-shaped architectures was also discussed [Henning], but answers are straightforward with this architecture. The second open problem relates to the computation of distance-2 coloring of a graph, which is a prerequisite to run a subsequent deterministic, lock-free matching algorithm. The discussion [Bora, Fredrik] converged to the fact that if the final goal of obtaining a deterministic matching was sought, directly using a locking algorithm could do faster for a small number of threads.

## 3.11 Recent Advances in Streaming (Hyper)Graph Decomposition

*Marcelo Fonseca Faraj (Universität Heidelberg, DE)*



There is a gap in (hyper)graph decomposition algorithms. Streaming algorithms, which are adaptable to small machines, partition huge (hyper)graphs quickly, but yield low-quality results. Conversely, in-memory algorithms produce high-quality solutions but require significant memory. Our talk explores recent advances in streaming (hyper)graph decomposition. We

begin with streaming graph partitioning, covering from hash-based to buffered approaches. Next, we discuss the state of the art in streaming process mapping. Finally, we present recent advances in streaming hypergraph partitioning. With their recent introduction and potential for further improvement, these families of streaming algorithms present unexplored avenues for improvement, while their recent strong results can provide new insights into solving the respective in-memory versions of these problems.

**Discussions.** There were some discussions if the degree of a vertex is currently considered in the local opbjective function that is optimized by the streaming graph partitioning algorithm. Currently this is not the case, but this may be interesting to consider in the future. Moreover, Tobias Heuer mentioned that the buffered streaming model for streaming graph partitioning could also be extended to hypergraph partitioning yielding potentially higher solution quality for hypergraph streaming partitioners.

## 3.12 Graph partitioning and distributed graph processing – An end-to-end optimization perspective

*Ruben Mayer (Universität Bayreuth, DE)*



Graph partitioning is often considered as a necessary preprocessing step for distributed graph processing. In doing so, the partitioning quality in terms of cut size and balancing is crucial to the performance of distributed graph processing jobs. However, yielding high graph partitioning quality is a challenging and compute-intensive problem. Many different graph partitioning algorithms have been proposed, which differ both in their achieved partitioning quality as well as their computational costs. How many resources and how much time to invest into partitioning depends on various factors such as the graph size, the resource budget of the user, and the complexity and run-time of subsequent graph processing on the partitioned graph. In my talk, I will elaborate on the problem of optimizing the end-to-end graph processing pipeline.

**Discussions.** There were several questions during the talk. For example, what can we learn from the trained ML models in EASE about the workloads under which certain partitioners perform best and how much of the graph in HEP is high-degree vs. low-degree, especially whether there are significant high-degree parts.

## 3.13 Parameterized complexity and algorithmics – some horizons – and the universal applied paradigm of diverse solutions

*Mike Fellows (University of Bergen, NO)*



The talk will describe in a very accessible way the foundational motivations and brief formal setup of parameterized complexity, a name which has too many syllables, but it is pretty straightforward and can be compared to “coordinatized geometry”. It has always

been nurtured as a theory, with the aspiration to be useful in practice. It goes beyond the one-dimensional P versus NP framework in a quite simple way, much as coordinatized geometry went beyond the Geometry of the Greeks. One natural way to usefully deploy parameterization is to address the working reality of many computing applications domains: that a single mathematically optimal solution is not what is really wanted! In many practical computing applications areas, what is really wanted is a moderate-sized collection of quality-wise pretty good solutions to choose from, often on the basis of side information not included in the strict optimal mathematical model. The talk is based on a recent IJCAI paper with multiple authors that began this direction of research, that fits very neatly with the mathematical algorithm design tools of parameterized complexity.

## 3.14 Combinatorial problems in sparse matrix computations

*Xiaoye S. Li (Lawrence Berkeley National Laboratory, US)*



We will describe the combinatorial algorithms needed in sparse matrix computations for solving algebraic equations. We will focus on the open problems in the graph preprocessing stages, such as ordering, symbolic factorization, and communication schedule, and particularly the speculations on the multi-GPU design.

**Discussions.** George Karypis mentioned that there isn't any refinement used specifically for the vertex separator. There was some discussion that involved multiple iterations of Luby's algorithm to increase the size of the computed independent set (and hence decrease the size of the computed vertex cover). Sherry clarified that, at present, they only use a single iteration, although he acknowledged that there's potential to enhance its quality. The two delved deeper into the concept of multilevel refinement. In this context, there were dicussions if and how combinations of mindegree and nested dissections, which have proven to be effective in practice, had been considered.

## 3.15 Algebraic Programming for Graph Computing: GraphBLAS and beyond

*Albert-Jan Yzelman (Huawei Technologies – Zürich, CH)*



Evolving from GraphBLAS, Algebraic Programming, or ALP for short, requires programmers to annotate their programs with explicit algebraic information. This information is then used in auto-parallelisation and other automatically applied optimisations, ranging from low-level concerns such as vectorisation to more complex algorithmic transformations. Recent work revolves around achieving faster parallel performance chiefly via non-blocking execution, providing more humble programming interfaces beyond GraphBLAS' generalised sparse linear algebra, and introducing a system for structured data representation. This talk will briefly introduce the status of ALP as-is, the guiding ideas behind its design, and a summary of recent advances. It then focuses on challenges towards increasing the usability and applicability

of ALP, as well as on challenges in bridging the state of the art in combinatorial scientific computing algorithms –such as partitioners or schedulers– for improving the performance of ALP-based programs.

**Discussions.** There was a discussion regarding the scope of algebraic programming, in particular if it is exclusively tied to GraphBLAS, Albert-Jan Yzelman clarified that algebraic programming is not restricted to just GraphBLAS. In fact, the domain encompasses a wide variety of applications and tools.

## 3.16 Distributed Landmark Labelling Using Vertex Separators

*Kamer Kaya (Sabanci University – Istanbul, TR)*



Distance queries are a fundamental part of many network analysis applications. Distances can be used to infer the closeness of two users in social networks, the relation between two websites in a web graph, or the importance of the interaction between two proteins or molecules. As a result, being able to answer these queries rapidly has many benefits to the area of network analysis as a whole. Pruned landmark labeling (PLL) is a technique used to generate an index for a given graph that allows the shortest path queries to be completed in a fraction of the time when compared to a standard breadth-first or a depth-first search-based algorithm. Parallel Shortest-distance Labeling PSL reorganizes the steps of PLL for the multithreaded setting and works particularly well on social networks. Unfortunately, even for a medium-size, 50 million vertex graph, the index size can be as large as 300GB. On the same graph, a single CPU core takes more than 12 days to generate the index. This presentation is on a distributed algorithm by partitioning the input graph. The proposed method improves both the execution time and the memory consumption by distributing both the data and the work across multiple nodes of a cluster.

**Discussions.** There was some discussion on how the graph can be better partitioned and how the algorithm can be modified for vertex/edge additions/deletions. The size of the index for larger graphs was questioned and the best possible performance we can was also discussed briefly.

## 3.17 Recent Advances in Ka (Hyper)Graph Partitioning

*Daniel Seemaier and Lars Gottesbüren (KIT – Karlsruher Institut für Technologie, DE)*



In the first part of our talk, we will present a brief overview of our two main solvers Mt-KaHyPar and KaMinPar for balanced (hyper)graph partitioning. Until recently, there was a severe gap in terms of solution quality between sequential and parallel solvers. We have since parallelized all of the techniques in sequential solvers that lead to high solution quality, which are now available in Mt-KaHyPar. Moreover, we tackled new problem domains such as efficiently partitioning into a very large number of clusters with the deep multilevel scheme,

which is available in KaMinPar. Additionally, KaMinPar was recently extended with a high-performance MPI mode, adapting the deep multilevel scheme to the distributed scenario. In the second part of our talk, we will then present some currently ongoing work. Traditionally, local search algorithms perform only balance-preserving moves. Perhaps unsurprisingly, we can achieve huge quality improvements by permitting large balance violations; performing unconstrained moves with some caution and rebalancing the solution later on. The same quality gap that used to exist between sequential and shared-memory parallel, now exists between shared-memory parallel and distributed-memory partitioners. Therefore, we are working on a distributed FM version in KaMinPar, as well as a distributed version of the recently proposed JET algorithm, both of which show promising results in preliminary experiments.

**Discussions.** The discussions revolved around the topics of scalability and problem formulations. Specifically, how many cores are needed for Mt-KaHyPar to be faster than PaToH (2 cores), why does the deterministic version scale better than the non-deterministic versions (because non-determinism may incur additional rounds needed to converge) and matters of parallel programming.

## 3.18 Graph Neural Network Research at AWS AI

*George Karypis (University of Minnesota – Minneapolis, US)*



During just a few years, Graph Neural Networks (GNNs) have emerged as the prominent supervised learning approach that brings the power of deep representation learning to graph and relational data. An ever-growing body of research has shown that GNNs achieve state-of-the-art performance for problems such as link prediction, fraud detection, target-ligand binding activity prediction, knowledge-graph completion, and product recommendations. As a result, GNNs are quickly moving from the realm of academic research involving small graphs to powering commercial applications and very large graphs. This talk will provide an overview of some of the research that AWS AI has been doing to facilitate this transition, which includes developing the Deep Graph Library (DGL)–an open source framework for writing and training GNN-based models, improving the computational efficiency and scaling of GNN model training for extremely large graphs, developing novel GNN-based solutions for different applications, and making it easy for developers to train and use GNN models by integrating graph-based ML techniques in graph databases.

## 3.19 An MPI-based Algorithm for Mapping Complex Networks onto Hierarchical Architectures

*Henning Meyerhenke (HU Berlin, DE)*



Processing massive application graphs on distributed memory systems requires to map the graphs onto the system's processing elements (PEs). This task becomes all the more important when PEs have non-uniform communication costs or the input is highly irregular.

Typically, mapping is addressed using partitioning, in a two-step approach or an integrated one. Parallel partitioning tools do exist; yet, corresponding mapping algorithms or their public implementations all have major sequential parts or other severe scaling limitations. In this paper, we propose a parallel algorithm that maps graphs onto the PEs of a hierarchical system. Our solution integrates partitioning and mapping; it models the system hierarchy in a concise way as an implicit labeled tree. The vertices of the application graph are labeled as well, and these vertex labels induce the mapping. The mapping optimization follows the basic idea of parallel label propagation, but we tailor the gain computations of label changes to quickly account for the induced communication costs. Our MPI-based code is the first public implementation of a parallel graph mapping algorithm; to this end, we extend the partitioning library ParHIP. To evaluate our algorithm's implementation, we perform comparative experiments with complex networks in the million- and billion-scale range. In general our mapping tool shows good scalability on up to a few thousand PEs. Compared to other MPI-based competitors, our algorithm achieves the best speed to quality trade-off and our quality results are even better than non-parallel mapping tools.

**Discussions.** The discussion after the talk revolved around various aspects: (i) the advantages and disadvantages of modeling the system architecture by a tree, (ii) implementation aspects regarding high-degree vertices, and (iii) the straightforward extension to reordering with the proposed approach. Furthermore, some pointers to additional related work were provided.

# 4 Working Groups

## 4.1 Balanced Edge Partitioning for Distributed Graph Processing

*Ruben Mayer (Universität Bayreuth, DE)*



The generally accepted formulation of the edge partitioning problem imposes a load balancing constraint on the number of *edges* per partition: $\forall p_i \in P : |p_i| \leq \alpha * \frac{|E|}{k}$ for a given $\alpha \geq 1, \alpha \in \mathcal{R}$, where $|p_i|$ denotes the number of edges in partition $p_i$. However, balancing only the number of edges does not always lead to good load balancing in distributed graph processing. In some cases, it is better to balance the number of vertex replicas. The open problem is to achieve an edge partitioning that is balanced both in the number of edges and vertices while minimizing the vertex replication factor. First thoughts in that direction may lead to the formulation of a multi-constraint partitioning problem. The working group discussed various models that could be used to achieve that, i.e. modelling the problem as a multi-constraint hypergraph or matrix partitioning problem.

## 4.2 Edge-Colored Clustering

*Blair D. Sullivan (University of Utah – Salt Lake City, US)*



This working group discussed the Edge-Colored Clustering problems presented in Thursday morning's Open Problem session, which relate to issues in clustering hypergraphs with categorical edge labels. The group began by trying to relate the non-overlapping variant of the problem to hypergraph partitioning without balance constraints. They also considered whether the locally budgeted overlap problem is equivalent to a generalization of vertex cover (in particular, b-coloring was discussed), eventually resulting in a family of graphs for which the two problems diverge arbitrarily. In addition to trying to relate ECC to previously-studied problems, the group also discussed (in)approximability, data reduction/kernelization, the possibility of using ILP solvers, and practical use-cases.

## 4.3 Exact $k$-way sparse matrix partitioning

*Rob H. Bisseling (Utrecht University, NL)*



Robust ILP methods have consistently demonstrated superior performance over Brand and Bound when it comes to matrix partitioning. Delving deeper into the intricacies of these solvers reveals a myriad of complex processes at work. The question arises: what unique advantages do ILPs offer in the realm of sparse matrix and graph partitioning? What insights can we glean from their application and functionality? This was discussed in part in the working group. Interestingly, when ILP is applied during the initial partitioning phase, it often paves the way for potential improvements. However, it's worth noting that these enhancements, while promising, can come at a significant cost. And surprisingly, in the context of initial partitioning, they might not always translate into better overall results. Could there be something valuable to extract from the pre-solve and branch-and-bound techniques employed in these solvers? Such insights could be revolutionary, especially given the current limitations of solvers when dealing with dense networks. Furthermore, there was an emerging discourse surrounding parallel solvers. It's essential to emphasize this area as it holds significant promise and potential for future developments in matrix partitioning techniques.

# 5 Open Problems

In the last four decades, there has been a tremendous amount of research in the area of the seminar. See for example the book by Bichot and Siarry [7], the survey by Schloegel et al. [67] or Kim et al. [32] as well as last generic surveys on the topic by Buluç et al. [10] and more recently Çatalyürek et al. [14]. However, a wide range of challenges remain in the area. Thus we now report currently open problems and future directions in the area of (hyper)graph decomposition that have been presented during Dagstuhl Seminar 23331 on "Recent Trends in Graph Decomposition".

## 5.1 Preliminaries

A *weighted undirected hypergraph* $H = (V, E, c, \omega)$ is defined as a set of $n$ vertices $V$ and a set of $m$ hyperedges/nets $E$ with vertex weights $c : V \to \mathbb{R}_{>0}$ and net weights $\omega : E \to \mathbb{R}_{>0}$, where each net $e$ is a subset of the vertex set $V$ (i.e., $e \subseteq V$). The vertices of a net are called *pins*. We extend $c$ and $\omega$ to sets in the natural way, i.e., $c(U) := \sum_{v \in U} c(v)$ and $\omega(F) := \sum_{e \in F} \omega(e)$. A vertex $v$ is *incident* to a net $e$ if $v \in e$. $\mathrm{I}(v)$ denotes the set of all incident nets of $v$. The set $\Gamma(v) := \{u \mid \exists e \in E : \{v, u\} \subseteq e\}$ denotes the neighbors of $v$. The *degree* of a vertex $v$ is $d(v) := |\mathrm{I}(v)|$. We assume hyperedges to be sets rather than multisets, i.e., a vertex can only be contained in a hyperedge *once*. Nets of size one are called *single-vertex* nets. Given a subset $V' \subset V$, the *subhypergraph* $H_{V'}$ is defined as $H_{V'} := (V', \{e \cap V' \mid e \in E : e \cap V' \neq \emptyset\})$.

A *weighted undirected graph* $G = (V, E, c, \omega)$ is defined as a set of $n$ vertices $V$ and a set of $m$ and edges $E$ with vertex weights $c : V \to \mathbb{R}_{>0}$ and edge weights $\omega : E \to \mathbb{R}_{>0}$. In contrast to hypergraphs, the size of the edges is restricted to two. Let $G = (V, E, c, \omega)$ be a weighted (directed) graph. We use *hyperedges/nets* when referring to hypergraphs and *edges* when referring to graphs. However, we use the same notation to refer to vertex weights $c$, edge weights $\omega$, vertex degrees $d(v)$, and the set of neighbors $\Gamma$. In an undirected graph, an edge $(u, v) \in E$ implies an edge $(v, u) \in E$ and $\omega(u, v) = \omega(v, u)$.

## 5.2 Balanced (Hyper)graph Decomposition and Variations

### Balanced Hypergraph Partitioning

**Multilevel Scheme.** Although traditional coarsening algorithms work particularly well for mesh graphs, their extension to hypergraphs has revealed a lack of understanding and can easily destroy the structure of the hypergraph. It is important to invest in better coarsening techniques tailored specifically to hypergraphs to preserve their structural properties. An interesting avenue in that direction could be incorporating embeddings during coarsening. By leveraging embeddings, coarsening algorithms could potentially achieve better representations of hypergraph structures, leading to improved partitioning outcomes. While spectral techniques were popular in the pre-multilevel era, pure spectral partitioning was not deemed competitive afterwards – mainly due to high running times. With the increased performance of today's machines and GPUs, it might be worthwhile to revisit these approaches as multilevel refinement techniques. Recently, unconstrained refinement (ignoring balance constraint while performing node moves) with subsequent rebalancing has shown promising results. However, the design space of these types of algorithms is far from being explored.

**Methodology.** Currently, we are lacking evaluations of real-world applications and workflows that use partitioning for load balancing and communication volume minimization. Therefore, the impact of quality gains in partitioning in terms of running time improvements for the applications are somewhat unclear. Moreover, in both graph and hypergraph partitioning, we still don't have a uniformly accepted balance constraint definition that works well in the case of vertex weighted (hyper)graphs. There exist definitions that enforce a lower bound on the block weights, add the weight of the heaviest vertex to the balance definition, or simply require that each block must be non-empty. Given that finding a balanced partition is an NP-hard problem even without optimizing an objective function, we should investigate in a balance definition that guarantees the existence of a feasible solution without providing too much leeway in the maximum allowed block weights.

**High-Quality Distributed-Memory Partitioning.** In recent years, several publications demonstrated that shared-memory partitioning algorithms can achieve the same solution quality as their sequential counterparts. However, the same quality gap still exists between sequential and distributed-memory solvers.

**Bottleneck Objective Functions.** For parallel computations, we assign the nodes of a (hyper)graph evenly to processors in a computing cluster. This should balance the computational load across the cluster. However, this does not bound the communication between processors, which can also become a sequential bottleneck if some PEs have to communicate significantly more than others. Therefore, we should investigate in techniques for optimizing bottleneck objective functions.

**The One Partitioning Tool Idea.** The graph- and hypergraph partitioning problems come in many different flavors: weighted vs. unweighted (hyper)graphs, directed vs. undirected hypergraphs, different objective functions, single vs. multi-objective, single vs. multi-constraint, partitioning with fixed vertices, partitioning with variable block weights, etc. Can we join forces and build (upon) a single open-source multilevel framework that is easily extensible to foster the research and development of new partitioning heuristics such that we can have a single tool that actually is able to solve all of these problems?

## (Hyper)DAG Scheduling

Let a HyperDAG (or, alternatively, a DAH – directed acyclic hypergraph) represent a computation and be given by a set of vertices and directed hyperedges, i.e., $\mathcal{H} = (\mathcal{V}, \mathcal{N})$. Here, $\mathcal{V} = \mathcal{S} \cup \mathcal{T} \cup \mathcal{O}$ while every directed hyperedge $n \in \mathcal{N}$ consists of a source and an arbitrary number of destination vertices; i.e., $n \in \mathcal{V} \times \mathcal{P}(\mathcal{V})$, where $\mathcal{P}(\mathcal{V})$ is the power set of $\mathcal{V}$. The vertices $\mathcal{S}$ are the input (source) data of the computation, the outputs are in $\mathcal{O}$, while intermediate computations are captured by computing *tasks* in $\mathcal{T}$.

Scheduling the computation on a parallel system with $p$ processing units requires assigning each vertex $v \in V$ a time step $t_v$ and a location $s_v \in \{0, 1, \ldots, p-1\}$ that define when and where to execute the intermediate computation in the case of $v \in \mathcal{T}$, or when and where an input (or output) should be available in the case of $v \in \mathcal{S}$ (or $v \in \mathcal{O}$).

Let us initially consider a machine model that costs communication and computation, though does not consider weights for simplicity of presentation – i.e., each data element $v \in \mathcal{S} \cup \mathcal{O}$ uniformly costs some unit storage; $v \in \mathcal{T}$ generates intermediate data that costs the same unit storage; and $v \in \mathcal{T}$ costs some unit time to compute.

Approaching the scheduling problem from a hypergraph partitioning point of view generates a mutually disjoint $\mathcal{V}_0, \ldots, \mathcal{V}_{p-1}$ partition of $\mathcal{V}$ under some allowed load imbalance $\epsilon$, and minimizes the traditional $\lambda-1$-metric $\sum_{n_i \in \mathcal{N}} (\lambda_i - 1)$; i.e., minimizes the communication volume of data units between parts of the partition[4]. However, even a perfectly balanced and optimal partitioning may lead to a division of the HyperDAG across the $p$ compute units that exposes no parallelism whatsoever. One solution is to divide the HyperDAG into $s$ layers $\mathcal{L}_0, \ldots, \mathcal{L}_{s-1}$, where vertices $v \in \mathcal{L}_{\leq i}$ are predecessors of those in $\mathcal{L}_{>i}$, and then to partition each layer separately. Recent results show, amongst other results, that the resulting HyperDAG partitioning problem is NP-hard, and also that no polynomial-time approximation algorithm exists [52]. An additional problem is determining an appropriate $s$, which is a hard problem on its own.

[4] Here, $\lambda_i$ is defined as the *connectivity* of the $i$th hyperedge, i.e., the number of parts of the partition the vertices in that hyperedge span.

**Optimal Scheduling.** Similar in motivation to the sparse matrix partitioning problem in Section 5.2 in this paper, one challenge is to find optimal schedules for real-world HyperDAGs. A collection of problems may be found in open HyperDAG_DB repository[5], which welcomes additional problem submissions. Determining optimal schedules enables efficient execution of oft-repeated computations, such as those in training neural networks; enables gauging the effectiveness of current heuristics for online scheduling, such as those used within run-time systems like OpenMP or Cilk; and enables inspiring better on-line heuristics by looking at optimal examples. This direction implies finding better ILP formulations and improved pruning strategies for use with optimal scheduling algorithms. Pruning strategies may furthermore rely on data-driven methods, see e.g., Juho et al. [39], trained using entries of the HyperDAG database that have been solved to optimality.

**Models and Hardness.** The hardness results previously presented depend on assumptions on the underlying machine and cost models. Indeed, other choices may reveal differing hardness results– for example, the same recent work shows that removing the layer-wise constraint in favor of a makespan constraint on the HyperDAG partitioning results in an optimization problem where *evaluating* whether said constraint has been violated is an NP-hard problem in itself [52]. A fundamental challenge thus is to identify what machine and cost model choices

1. result in significantly harder optimization problems,
2. affect the search space underlying the optimization problem and how, and
3. have optimal schedules that relate to one another and how.

Example modeling choices include whether time step assignment takes place at the unit vertex granularity or in bulk (e.g., assigning multiple tasks to a single layer); whether data between vertices are moved individually or in bulk; whether communication in a time step charges constant latency, a cost proportional to a size, or both; whether communication size corresponds to volume or $h$-relations[6]; whether communication may overlap with computation; or whether communication throughput and latency (when costed) are uniform across the $p$ processing units, or instead hierarchical or even topology-dependent. More detailed initial considerations on such modeling options appear in a pre-print [51].

To make each of the three above challenges more concrete, we briefly follow with known examples: 1+2) electing a machine model where vertex-to-time-step assignment happens in bulk and layer-wise, leads to fewer variables in an ILP formulation and thus to a reduced search space; yet, paradoxically, also has stronger known hardness results compared to non-layered hypergraph partitioning [52]; 3) there is at most a factor two difference between optimal BSP solutions[7] with overlapping communications versus those without.

## (Hyper)graph Algorithms and ALP

Algebraic programming, or ALP for short, enables writing programs with explicit algebraic information passed into the programming framework. Examples of such algebraic information are binary operators and their properties such as associativity, commutativity, etc., as well as richer algebraic structures such as monoids and semirings. A semiring embodies the

[5] `https://github.com/Algebraic-Programming/HyperDAG_DB/`

[6] the maximum of incoming and outgoing messages to or from any partition at a given time step.

[7] with BSP, compute tasks and communication are considered in bulk, communication charges both latency and size, communication size is given by $h$-relations, and latency as well as throughput parameters are uniform [78].

rules under which linear algebra takes place, but allows its generalization to any pair of additive and multiplicative operations under which those rules hold; for example, while the plus-times semiring enables standard numerical linear algebra, the min-plus semiring enables shortest-paths computations. The following two observations are core to GraphBLAS: a) most graph algorithms can be expressed in (generalized) linear algebra; and b) our deep understanding of optimizing sparse linear algebra (thus) applies to graph computations.

The recent nonblocking mode of ALP/GraphBLAS performs fusion of linear algebraic primitives under any algebraic structure, at run-time. It achieves up to 16.1× and 12.2× speedup over the similar state-of-the-art frameworks of SuiteSparse:GraphBLAS and Eigen on ten matrices for the PageRank algorithm, with similar results for a Conjugate Gradient (CG) solver and sparse deep neural network inference [46, 45]. Other recent work introduces support for dense linear algebra, matrix structures (e.g., triangular), and views (e.g., permutations or outer products) [71]. It furthermore enables automatic distributed-memory execution of sequential ALP code [86, 68].

**ALP Accelerating Graph Algorithms.** With the new extensions, both the applicability and performance of the ALP framework has increased, and should enable the acceleration of graph algorithms that previously only had sequential or otherwise un-optimized representations. To aid porting efforts, ALP not only supports the auto-parallelization of linear algebraic formulations of graph problems, but also that of vertex-centric ones [82].

One challenge is to find graph algorithms that, despite recent advances, remain hard to express using generalized linear algebra, or graph algorithms that are expressible yet do not achieve high performance. Examples include $k$-core decomposition and $p$-spectral clustering, the former done successfully [41] and the latter still partially relying on non-ALP code [53].

**Graph Algorithms Accelerating ALP.** Techniques exist to accelerate sparse matrix computations using hypergraph partitioning, either on distributed-memory [13, 80, 57], shared-memory [83, 84], or both simultaneously [87, 85]. However, based on the computation, either the hypergraph representation of a sparse matrix must be adapted, the minimization objective modified, or both; see, e.g., Ballard et al. [5] who consider sparse matrix–matrix (SpMSpM) multiplication rather than sparse matrix–vector (SpMV) multiplication, as most preceding cited works. Thus for ALP as a programming model, the challenge lies in how these models and optimization techniques may be combined – preferably transparently to the programmer – to optimize the arbitrary sequences of computations and inputs that ALP encounters.

For example, while we may readily reuse known techniques to optimize any program consisting of SpMV multiplications with the same input matrix and some vector operations, the framework must be smart to select a different model and optimization objective when it concerns SpMSpM multiplication instead. Furthermore, relying on such existing work requires ALP to translate between different partitionings whenever differing operations or differing input matrices are encountered.

Ideally, however, the framework co-optimizes across multiple primitives and inputs that it encounters. The following avenues seem possible:

1. dynamically building fine-grained (hyper)DAG representations, followed by partitioning, (Hyper)DAG partitioning [50, 59], or scheduling (see also the related problem 5.2);
2. employing a coarse-grained parameterized representation of the computation and employing analytic choices, or
3. some mixture of the preceding avenues.

Difficulties with the first solution likely relate to the scale of the resulting optimization problem. For the second, while work in nonblocking ALP/GraphBLAS execution shows that such approaches may be effective [45], it is unclear how they may extend to arbitrary

primitives and inputs. A successful solution thus may well lie with the third option, and require mixed fine- and coarse-grained representations combined with both combinatorial and analytic techniques.

### Sparse Matrix Partitioning

Given an $m \times n$ matrix $A$ with $N$ nonzeros, the sparse matrix partitioning problem seeks a partition of $A$ into $p$ disjoint parts $A = \cup_{i=0}^{p-1} A_i$ such that the number of nonzeros in part $A_i$ satisfies $|A_i| \leq (1+\varepsilon) \left\lceil \frac{N}{p} \right\rceil$ for $0 \leq i < p$, where $\varepsilon \geq 0$ is a given load-imbalance parameter. Important questions for the area of sparse matrix partitioning as well as graph and hypergraph partitioning are: *How good is heuristic bipartitioning compared to exact bipartitioning? How good is recursive bipartitioning into k parts compared to direct k-way partitioning?*

To answer these questions, we can solve a set of small- and medium-size problem instances to optimality using an exact algorithm either based on the branch-and-bound (BB) approach, or on an integer linear programming (ILP) approach. To answer the first question, a set of 839 matrices has been bipartitioned by the programs MondriaanOpt [58] and MatrixPartioner [34]. To answer the second question, an exact bipartitioner has been employed within a recursive bipartitioning program for $k = 4$ and it has been compared with an exact direct 4-way partitioner using the program General Matrix Partitioner (GMP) [31] and the commercial ILP solver CPLEX.

We would like to scale up these initial results, to reach larger problems and be able to answer the main questions with more confidence. Since for $k = 2$ the bipartitioner MP works best, we ask whether its algorithm and implementation can be further improved. Parallelization should also help to enlarge our database of solved problems. One might interpret this database as a training set for learning (by either machines or humans) about properties of optimal solutions.

For $k > 2$, we surprisingly found that a basic formulation as an ILP solved by a commercial ILP solver was far superior to the BB solver GMP, and this poses the question what we can learn from the ILP solvers. Furthermore, can we use them for certain types of sparse matrix/graph partitioning? Finally, can we improve the basic formulation of the ILP to solve even larger problems?

### Scalable Distributed Memory Partitioning

Scalability of high quality parallel (hyper)graph partitioning remains an active area of research. In particular, achieving good scalability and quality on large distributed memory machines is still a challenge, but even on shared-memory machines, scalability to a large number of threads seems difficult. Even more difficult is aligning the inherent complexity and irregularity of state-of-the-art algorithms with the restrictions of GPUs or SIMD instructions. Another conundrum is that, for good memory access locality during partitioning, (hyper)graphs need to already be partitioned reasonably well. Hierarchies of supercomputers have to be taken into account during partitioning. This can be done by using multi-recursive approaches taking the system hierarchy into account or by adapting the deep multilevel partitioning approach sketched above to the distributed memory case. When arriving at a compute-node level, additional techniques are necessary to employ the full capabilities of a parallel supercomputer. For example, many of those machines have GPUs on a node level. Recently, researchers started to develop partitioning algorithms that run on GPUs and, while of independent interest, partitioning algorithms developed for this type of hardware can help in that regard. Hence, future parallel algorithms have to compute partitions on and for

heterogeneous machines. On the other hand, algorithms should be energy-efficient and performance per watt has to be considered. Lastly, future hardware platforms have to be taken into consideration when developing such algorithms. One way to achieve this will be to use performance portable programming ecosystems like the Kokkos library [76].

### Balanced Edge Partitioning for Distributed Graph Processing

The generally accepted formulation of the edge partitioning problem imposes a load balancing constraint on the number of *edges* per partition (cf., e.g., [47], [48]): $\forall p_i \in P : |p_i| \leq \alpha * \frac{|E|}{k}$ for a given $\alpha \geq 1, \alpha \in \mathcal{R}$, where $|p_i|$ denotes the number of edges in partition $p_i$. However, balancing only the number of edges does not always lead to good load balancing in distributed graph processing, as shown in [49]. In some cases, it is better to balance the number of vertex *replicas* – vertex copies produced whenever incident edges are placed in different partitions. The open problem is to achieve an edge partitioning that is balanced both in the number of edges and vertices while minimizing the vertex replication factor. First thoughts in that direction may lead to the formulation of a multi-constraint partitioning problem.

### Provably Effective Graph/Hypergraph Coarsening

Today almost all of the state-of-the-art graph and hypergraph partitioning tools utilize multi-level approaches that are comprised of three phases: coarsening, initial partitioning and uncoarsening/refinement. Even though numerous different coarsening techniques have been proposed and many are shown to be effective in multi-level partitioning, we still do not have a provably effective and efficient coarsening technique for graph or hypergraph partitioning problems. The situation is more dire for directed graph and hypergraph partitioning for acyclic partitioning. Keeping acyclicity during coarsening is a desirable property, yet, it is computationally expensive to ensure and maintain acyclicity, with flexible coarsening techniques.

Today we also do not have a well-defined objective for coarsening. In other words, we do not have well-agreed upon desirable properties of the coarsened graph. Overall we want to solve a partitioning problem, but in multi-level partitioning the overall success of the algorithm is a complex function of the three phases of the multi-level approach. We have many counterexample results showing that the best initial partitioning solution does not always yield the best result. Hence, it is even more difficult to define a goal for coarsening.

In undirected graph and hypergraph partitioning, many successful tools use randomized heavy edge matching/clustering techniques, where vertices are visited in random order, and they are matched with their unmatched neighbor with the heaviest connection. This randomization helps to "maintain" the graph's structure. Hence, one potential direction for successful coarsening techniques is randomized algorithms (see Section 5.2).

### Randomized Algorithms for Graph Sparsification and/or Coarsening

Randomized algorithms on networks often involve sampling nodes, edges, or subnetworks [24, 25, 35, 72]. These sampling techniques are used as subroutines (1) for the solution of fundamental problems on networks, such as connected components, the assignment problem, breadth-first search on long-diameter graphs, or global minimum cut, and (2) for graph learning problems trained with variants of mini-batch stochastic gradient. These problems are widely encountered in the US DOE applications, e.g., the use of the assignment problem in optimal transport (transforming probability distributions) for cosmology, the use of connected components in genomics problems. Sketching and sparsification, which are other randomization techniques used for networks, can be used to find approximate solutions

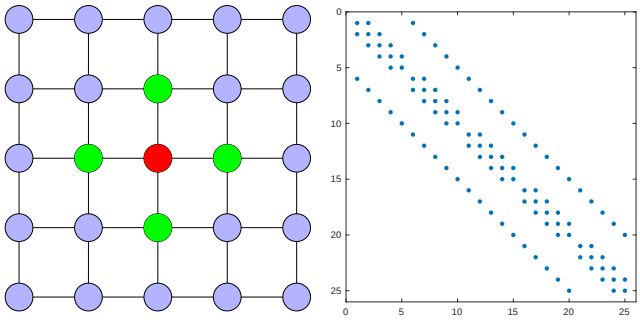


**Figure 1** The 2D mesh resulting from the discretization of a square domain with the five-point stencil, using 5 points in each dimension, and the corresponding matrix after a row-by-row ordering of the mesh points. . Left: A 2D mesh. Right: The matrix of the mesh.

for higher-level problems where the network problem is a subroutine. The computational science applications include domain-decomposition solvers, iterative solvers, preconditioning for sparse systems using approximate factorization. Sampling and sketching can also be used as a coarsening technique in multilevel graph partitioning. Unfortunately, the impressive advances in the theory of randomized algorithms for networks has not been translated into practical demonstrations. There are ample research opportunities to bridge this gap between theory and practice, and hence, to produce high-quality software running on modern HPC hardware with demonstrations on application codes.

### Balanced Streaming Partitioning

In streaming edge partitioning, edges are presented one at a time in a stream and must be assigned to a partition irrevocably at the moment they are encountered. Degree-based hashing has been proven effective for streaming edge partitioning, however, its potential benefits in the context of streaming *vertex* partitioning remain to be explored. An open problem is to investigate the benefits and challenges associated with using degree-based hashing techniques specifically for streaming vertex partitioning. Another notable limitation in the current literature on streaming partitioning algorithms is the predominant focus on common ordering strategies for the input (hyper)graph, such as random ordering, breadth-first search, and depth-first search orders. While these ordering strategies provide valuable insights into the performance of partitioning algorithms, they may not fully capture the challenges posed by real-world scenarios. Hence, it is an open problem to test streaming partitioning algorithms under adversarial node and edge orderings, particularly in the context of buffered streaming algorithms, where locality has a large impact on the quality of the result.

An open problem in the field of streaming process mapping is to address the problem when the underlying topology cannot be faithfully represented as a hierarchy, but only as a graph or hypergraph. Existing streaming algorithms either ignore the topology, i.e., solve the graph partitioning problem, or optimize directly for hierarchical topologies. However, many real-world scenarios involve complex interconnections that form graph-based topologies.

### Exact Solvers for Large $k$ (Hyper)graph Decomposition

Solving the graph bipartitioning problem to optimality using branch-and-bound algorithms has recently been shown to be highly effective if the optimum solution value is very small. For example, Delling et al. [20] can solve instances with millions of vertices to optimality in

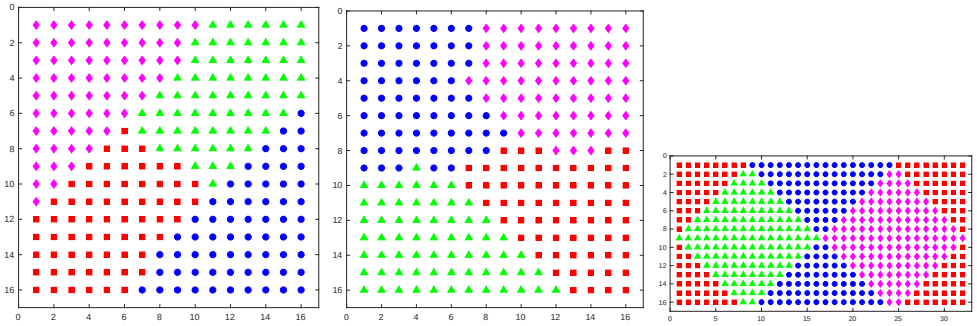

**Figure 2** Partitioning the $16 \times 16$ mesh with different routines; the basic diamonds cannot partition this mesh, but can partition a larger one. Left: The $K = 4$-way partition of the $16 \times 16$ obtained by the special routines [26]. Connectivity$-1$ is 58. Middle: The $K = 4$-way partition of the $16 \times 16$ mesh obtained by PaToH, suboptimal (connectivity$-1$ is 61). Right: The $K = 4$-way partition of the $16 \times 32$ mesh by the basic diamonds [8, Section 4.8], which is believed to be asymptotically optimal. Here though the connectivity$-1$ is 126 and PaToH obtains 93.

a reasonable amount of time. It is still open whether such solvers also exist for balanced (hyper)graph partitioning problems where $k$ is significantly larger than 2. The difficulty here comes from the fact that larger values of $k$ introduce a large amount of symmetry in the problem, i.e., if you have some partition of the graph, then any permutation of the block ids is also a partition of the graph that has the same balance and objective. Another problem of current solvers is how to handle dense instances or, more precisely, instances in which the objective function is large.

### Partitioning Stencils and Analyzing The Performance of Partitioning Tools

We investigate the scalability of the graph- and hypergraph-based sparse matrix partitioning methods in terms of being able to obtain high quality solutions in large problem instances. The quality measure that we are interested is the connectivity$-1$ metric, which usually measures the total volume of communication when vertices represent data items/computations and the hyperedges represent the dependencies. Ideally, theoretical investigations help explain the scalability or success of the methods. However, the current algorithms in use are too sophisticated to lend themselves to such an approach. We are thus looking for sound experimental methodology.

One approach is to take a subclass of problems, develop special partitioners, and compare the hypergraph partitioners with those. We take five-point stencil computations in two-dimensions (2D) for which we have a special linear-time partitioner [26, 77] and see how good the current partitioning methods are on these cases. We compare the performance of hypergraph partitioners on these. A rectangular 2D domain is discretized with the five point stencil and a mesh of size $X \times Y$ is obtained whose points are placed at integer locations. Two points $(x_1, y_1)$ and $(x_2, y_2)$ of the resulting mesh are neighbors iff $|x_1 - x_2| + |y_1 - y_2| = 1$. A sample mesh resulting from the discretization of a square domain with five points in each dimension is shown in Figure 1. The figure also shows the connections of a point. After an ordering of the mesh points, one can obtain an $X^2 \times Y^2$ matrix. The matrix obtained from the shown mesh after a row-major ordering is shown in Figure 1.

A partitioning of the points of the mesh $\mathcal{M}$ corresponds to a row-wise or a column-wise partitioning of the associated matrix $\mathbf{A}_{\mathcal{M}}$. Without loss of generality let us focus on the row-wise partitionings of $\mathbf{A}_{\mathcal{M}}$. The standard column-net hypergraph $\mathcal{H}_{CN}$ model [11] can

**Table 1** The partitioning of the $X \times X$ mesh of five-point stencils with perfect balance using the method of Grandjean and Uçar [26] obtains the numbers given in row "Vol" as connectivity−1 metric of the cut. PaToH's results are in the row "PaToH" – they do not have perfect balance. Apart from 58 in the last line, the other numbers are claimed to be optimal.

| $X$ | 16 | 64 | 256 | 1024 | 2048 |
|---|---|---|---|---|---|
| PaToH | 61 | 249 | 985 | 4034 | 7999 |
| Vol | 58 | 222 | 878 | 3500 | 6996 |

be used for this purpose. Partitioning the vertices of the hypergraph $\mathcal{H}_{CN}$ among $K$ parts will therefore correspond to partitioning the stencil computations among $K$ processors; the connectivity−1 metric of the cut will measure the total communication volume; and the balance of part weights in terms of vertices will correspond to balance of the loads of the processors. We assume unit vertex weights here, the effect of having less operations on the border points of the mesh will be ignored for simplicity (and is negligible).

With the special partitioning methods by Grandjean and Uçar, we obtain the total communication volume listed in Table 1. While the communication volume listed in the table is obtained with the routine itself, we note that it is given by the formula

$$2 \times \left( \left\lfloor \frac{n}{\sqrt{2}} \right\rfloor + n \right) + 4 \,, \tag{1}$$

which is claimed to be optimal for $X > 16$ in the table (for $X = 16$, the optimal communication volume is claimed to be 57).

This formula requires some conditions on $X$, which we do not give here. Are the connectivity−1 values given in Table 1 optimal for a perfectly balanced 4-way partitioning of the two mesh of $X \times X$ points discretizing a square domain with a five point stencil? For the $16 \times 16$ mesh, Gurobi solver [28] also finds a cut of value 57 in a few minutes under an additional constraint to assign the four corners to four different parts). The solution obtained by the Gurobi solver is at the close neighborhood of what is shown in Figure 2; by the method of Grandjean and Uçar is easily updated to mimic Gurobi's result.

Another point that arise from the given 4-way partitioning is that these partitions cannot be obtained in a recursive bisubsection scheme where each step greedily optimizes the cut hyperedges with perfect balance. This is so, as the first cut vertically cuts the mesh into two equally sized parts with perfect balance and optimal cut. Simon and Teng [70] delve more into this point in the context of graph partitioning.

We note that more results of the sort are given elsewhere [26]. The same reference also surveys some results from the literature, including references on discrete isoperimetric problems [81], which can be used to guide algorithms. Two things are of particular note: in the corners, the optimal parts are triangle-like, as in two corners in Figure 2 (left), and in the interior the optimal parts are diamond-like as in Figure 2 (right).

Bisseling and McColl [9] propose digital diamonds to partition similar meshes with wrap-around connections. Digital diamonds are $\ell_1$-spheres defined with a center $(c_x, c_y)$ and a radius $\rho$. Such a diamond contains all mesh points $(p_x, p_y)$ where with $|p_x - c_y| + |p_y - c_y| \leq \rho$. Grandjean and Uçar give formulas for the total communication volume when one uses digital diamonds. They also specify conditions on mesh and part sizes under which a partition by digital diamonds are possible. Basic diamonds proposed by Bisseling [8, Section 4.8] trim off two borders from the digital diamonds to address partitioning of another set of mesh and part sizes – these conditions as well as the total volume of communication are also specified by Grandjean and Uçar. Digital diamonds and basic diamonds are believed to be

asymptotically optimal in terms of the total communication volume, but obtain disconnected partitions on the borders of the mesh when there are no round-around connections – which is not desirable in certain applications.

Another approach is to take general sparse matrices boost the data in a way, reason about it and evaluate the performance of hypergraph partitioners on these. For example, suppose we partition a matrix $A$ row-wise into $k$ parts, and obtain a total communication volume of $T_V$ units in SpMxV. Then, if we partition the matrix $B = [A, A]$ row-wise again into $k$ parts, then the first partition should be good for $B$ with $2 \times T_V$ communication volume. If our partitioning tool is good, such a performance is expected; if the answers were not $T_V$ vs $2 \times T_V$, we could either improve the partition of $A$ or $B$. Similarly, a $k$-way row-wise partition of $C = [A; A]$ – this time two copies of $A$ are stacked to have twice as many rows – should have a communication volume of $T_V$ units. Some experimental investigation with PaToH [12] using these matrix repetition schemes [77] reveals a good behavior. What else can we say about the behavior of partitioning tools on more general problems?

### Highly Spread Out Weights in Mesh Partitioning

Many distributed numerical simulations rely on mesh partitioning to improve the balance of their computations on every computing unit, thus increasing their efficiency and scalability. A mesh is modeled by its dual graph or hypergraph as input to a partitioner: each vertex corresponds to a cell of the mesh, and the vertex weight is the computing cost of this cell. Good quality hexahedral meshes have a reasonably regular topology, mostly looking like a 2D or 3D grid. However, the vertex weight distribution can be highly spread out for various applications like Monte-Carlo particle transport simulations. For this kind of instance, classic multi-level approaches of the existing graph partitioner can have some quality issues, symmetric to the ones observed in Section 5.2. Multi-level graph partitioners focus on topological properties, and here, taking more into account vertex weight distribution should lead to better and faster obtained partitions.

### Cartesian Mesh Partitioning

Directly addressing data in memory is crucial in achieving high performance when running on modern architectures, especially on GPU. Grids allow direct access to neighbor cells for mesh computations, making stencil computations like the one presented in Section 5.2 very efficient. However, standard partitioning approaches lead to non-rectangular parts, making distributed applications less efficient. Thus, a new problem is partitioning a grid into parts that are all a subgrid or a set of subgrids. Such a partitioning model will also work to partition block structured meshes that often arise for hexahedral meshes.

### Problems in Multilevel Graph Partitioning With Star Graphs

The partitioning community has long focused on instances with a regular structure, e.g., mesh graphs or instances from circuit design. However, it becomes more and more important to find high-quality solutions for instances with an irregular structure, such as those derived from social networks. Surprisingly, we found a subclass of these instances where current state-of-the-art partitioning algorithms compute solutions that are far from optimal. The identified instances – referred to as *star instances* – are characterized by a core of a few highly-connected nodes (core nodes) with only sparse connections to the remaining nodes (peripheral nodes). One example of such an instance is the Twitter graph. Here, we found that partitioning the nodes into low- and high-degree vertices ($\leq$ median degree)

induces a bipartition that cuts half the edges as any of the existing multilevel partitioning tools. We identified several other social networks where we observed the same behavior. Thus, it becomes increasingly important to develop efficient partitioning techniques that can handle such instances. From a theoretical perspective, we were already able to present an $(R+1)$-approximation for star instances, where $R$ is the ratio of an approximation algorithm for the min-knapsack problem. This is a remarkable result since there exists no constant factor approximation for the general graph partitioning problem.

### Graph Partitioning with Ranked Vertices

For some graph algorithms, there is an implicit rank over the vertices. For instance, 2-hop indexing generates a *label cover* that can be used for answering pairwise shortest-distance queries. Classical algorithms, e.g., Pruned Landmark Labeling (PLL) [1] and its variants, leverage a ranking that has a drastic impact on the number of entries stored at local vertex indexes. In the distributed setting, the amount of entries in the *cut*, i.e., the ones replicated and/or communicated among the nodes, depends on this ranking. Especially when the number of nodes is high, this communication can create a bottleneck. For distributed execution, ranking the vertices also changes the loads on each part. In that sense, another problem at hand is given the rank, the amount of data stored at each vertex needs to be estimated well enough so that the part weights incur an acceptable level of imbalance. Hence, the problem is given a graph $G = (V, E)$, *what is the best vertex ranking and partitioning pair that yields the best performance in terms of execution time and maximum memory used at each node?*

### Designing Multilevel Algorithms

In a variety of fields, computational optimization challenges often arise when modeling large and complex systems, presenting significant hurdles for solving algorithms, even when high-performance computing resources are deployed. These obstacles are frequently due to a multitude of factors such as an extensive number of variables and the complexity in describing each variable or interaction. Problems involving combinatorial and mixed-integer optimization add extra layers of complexity. Specifically, the presence of integer variables frequently results in NP-hard problems, particularly in contexts where nonlinearity and nonconvexity are factors.

A widely adopted strategy for tackling these challenges involves the use of iterative algorithms. While these algorithms may be grounded in divergent algorithmic paradigms, they often exhibit a similar pattern: rapid improvement during initial iterations followed by a phase of slower progress. In the realm of iterative algorithms, utilizing first-order optimization techniques like gradient descent or methods that rely on limited observable data, such as local search, often leads to a local optimum that is usually suboptimal when compared to the true global optimum. Additionally, the algorithms employed within each iterative cycle are not always exact, further complicating the optimization process. To speed up these algorithms at each iterative step, various strategies including heuristics, parallelization, and different ad hoc techniques are commonly employed, albeit often at the expense of solution quality. Being trapped in local optimum of unacceptable quality is one of the most important issues of such algorithms.

Multilevel methodologies have been introduced to address the challenges of large-scale optimization, offering a strategy that reduces the chances of being trapped in low-quality local optimum. These techniques are complementary to stochastic and multistart approaches, which

also help the algorithm escape local optima. While there's no one-size-fits-all prescription for designing multilevel algorithms, their core philosophy revolves around global considerations while executing local actions based on a hierarchy of increasingly simplified representations of the original complex problem.

In practice, a multilevel algorithm initiates the optimization process by generating a hierarchy of progressively simplified (or coarser) problem representations. Each subsequent coarser level aims to approximate the problem at the current level but with fewer degrees of freedom, facilitating a more efficient solution process. After solving the coarsest problem, its solution is extrapolated back to the more detailed level for further refinement – a phase termed as "uncoarsening." Employing this multilevel approach frequently results in substantial improvements in both computational efficiency and the quality of solutions. There are many broad impact open questions in designing multilevel algorithms for (hyper)graphs some of which we mention here.

**Distance between vertices.** In order to coarsen the problem, a critical issue is to design a distance (or similarity) function between nodes. The question is simple: how to introduce a similarity function that will effectively find subsets of nodes that share the same solution (e.g., in the context of graph partitioning it is about predicting that nodes will be assigned the same part)? Incorrectly chosen subsets of nodes will mislead coarsening and will make the uncoarsening to work much harder which will result in increased complexity and poor results. In the same time, sophisticated distance functions are not supposed to destroy the overall complexity of the multilevel algorithm. Examples of such advanced solutions are spectral-based [16, 61, 69] and low-dimensional representations [73]. They work very well on the partitioning, ordering [66] and clustering multilevel schemes. However, there is also a lot of evidence that these algorithms are not perfect and do not fit all scenarios.

**Density of coarse levels.** This remains one of the most crucial issues in multilevel algorithms. In many problems and coarsening schemes the more we coarsen the problem, the more dense graphs are obtained unless we deliberately take actions to sparsify them. On the one hand, such dense representations often may approximate the original problem better. On the other hand, the complexity of refinement at the corresponding levels of uncoarsening becomes prohibitive. For example, in the algebraic multigrid inspired multilevel approach for graph linear ordering this issue was simply patched by reducing the interpolation order [64] which is a pretty blind solution. It was slightly improved in graph partitioning [65] by using a better node distance function in combination with the small interpolation order but more sophisticated and theory-grounded approaches are required. In a similar 2-dimensional layout problem, the authors switched to more regular coarsening with the geometric multigrid [60]. In general, dense graphs are problematic for most existing multilevel algorithms that mostly designed for sparse instances and require special treatment such as other coarsening schemes or special hardware [43].

**Maximization problems.** A particularly interesting class of problems for which multilevel algorithms have not reached their advanced stage is maximization problems such as max cut, maximum independent set, and maximum dominating set. A traditional coarsening approach quickly generates dense coarse levels and becomes impractical. Recent work on sophisticated node distance functions and sparsification improve the situation [3] but after a certain number of levels the quality of coarse levels becomes either poor (if sparsified) or intractable (otherwise). Rethinking of the coarsening ideas is required for this class of problems as such approaches as inverting graphs quickly become impractical.

**When to stop the refinement?** Perhaps there is no multilevel algorithm whose developers have not asked this question. Overall, there is no theory-grounded work related to optimization on (hyper)graphs that answers this question. Apart from the complexity issue, on the first glance it may look trivial that in the ideal refinement, the employed local optimization solvers should be optimal. However, there is a lot of practical evidence that terminating refinement before reaching the best possible local solution is beneficial to the final global suboptimal solution.

**Advanced types of multilevel cycles.** In multilevel schemes, the V-cycle coarsening-uncoarsening is the most basic and widely used cycle for this purpose, but several other advanced cycles aim to improve the efficiency and effectiveness of multilevel methods. Most widely used of them are: (1) the W-cycle is a more advanced version of the V-cycle that provides a more aggressive approach to solving the coarser problems. In a W-cycle, a refinement and full deeper W-cycle is performed at each coarser level before moving back to the finer level. This allows for more thorough refinement at lower levels, often leading to better convergence properties compared to the V-cycle. (2) The F-cycle method creates the hierarchy of coarse representations and starts at the coarsest level and works its way up to the finest grid, solving the problem at each level by applying another full V- or W-cycle. It combines with V-cycles or W-cycles at each level for better optimization of coarse levels. Both F- and W-cycles are particularly effective for problems where an initial coarse approximation is not easy to obtain. Both cycles usually exhibit better then in V-cycle quality which comes at additional cost of complexity. The W-cycles are usually more expensive but do exhibit a good quality [62, 63]. Finding robust criteria on when to recursively apply one or another type of advanced cycle (if at all) is very important in multilevel algorithms as their running time is increased with the advanced cycles.

## 5.3 (Hyper)graph Clustering

### Correlation Clustering

In the correlation clustering problem the input is a graph with edges labeled with + and − (or simply with +1 and −1). + indicates that the endpoints of the edge should be in the same cluster, and − means that the endpoints of the edge should be in different clusters. The goal of correlation clustering is to find a clustering that respects as many of these requirements as possible. Of course respecting all of them is in general not possible, and so a commonly studied objective is to minimize the number of disagreements.

There is a big discrepancy between the theory work on correlation clustering and what is done in practical solutions. For example, while the famous PIVOT algorithm provides 3-approximation for complete graphs, if the algorithm is run on a sparse graph (i.e., one where + edges induce a sparse graph) the algorithm often gives a solution that is worse than leaving each node in a cluster of size 1. Better approximation algorithms are known, but they are not as scalable, as they rely on solving an LP or SDP. In the case of weighted or not-complete graphs the best known approximation ratio is $O(\log n)$.

Despite all of these theoretical advances, the solutions that are implemented in practice are based on local swaps and a multilevel approach. In particular, the basic operation that these algorithms make is moving a node to a neighboring cluster, only if this increases the overall objective. This way, the algorithm essentially treats the objective function as a blackbox and does not leverage all the structural properties of the problem, which are used to give approximation algorithms.

While the practical implementations are quite scalable, there is probably room for improvement, as the number of logical rounds needed to obtain a good solution goes in hundreds. This in particular makes these algorithms not easy to use in distributed settings.

An interesting open problem is to bridge the gap between theory and practice for correlation clustering with the goal of obtaining better practical implementations. Specifically, it would be interesting to develop algorithms requiring fewer rounds, which will make them amenable to an efficient distributed implementation.

### Overlapping Edge-Colored Clustering

Edge-Colored Clustering is a categorical clustering framework [2] whose input is an edge-colored hypergraph and output is an assignment of colors to nodes which minimizes the number of edges where any vertex has a color different from its own (*mistakes*). We are interested in variants of this problem which allow budgeted overlap. Specifically, the following three notions were defined in [17]. LocalECC allows up to $b$ of color assignments at each node. GlobalECC allows one "free" color assignment for each node, plus $b$ additional assignments across all nodes. RobustECC allows each node to either receive exactly 1 color, except that at most $b$ nodes are assigned *every* color. Equivalently, at most $b$ nodes are deleted.

Each of these problems generalizes ECC, with equivalence for the first coming at $b = 1$ and for the latter two at $b = 0$. Consequently, they are each NP-hard (Angel et al. [2]). We (Crane et al. [17]) showed that greedy algorithms give an $r$-approximation on the number of edge mistakes, where $r$ is the maximum hyperedge size. Further, for LocalECC, a $(b + 1)$-approximation can be achieved with LP-rounding. More generally, we ask about bicriteria $(\alpha, \beta)$-approximations, where $\alpha$ is the approximation factor on edge mistakes and $\beta$ is the approximation factor on the budget $b$, and show that all three variants have such approximation algorithms, though the factors are no longer constants for GlobalECC.

**Open Problems.** Are these ideas relevant for any practical applications? Where? What can we assume about the inputs in those settings? Are there constant-factor single-criteria approximations for the Global and Robust versions? Does GlobalECC have a constant-constant bicriteria approximation? More generally, bicriteria inapproximability is an interesting and relatively unexplored direction. We saw that empirically these approximations performed much better in practice than the guarantees. Is there some sort of structure in real-world instances that we can model to improve our analysis?

### Dense Graph Partition

Dense Graph Partition, introduced by Darley et al. [19], models finding a community structure in a social network. Formally, given an undirected graph $G = (V, E)$, the task is finding a partition $\mathcal{P} = \{P_1, \ldots, P_k\}$ of $V$, for some $k \geq 1$, of maximum density. With $E(P_i)$ denoting the number of edges among vertices in $P_i$, the density of $\mathcal{P}$ given by $d(\mathcal{P}) = \sum_{i=1}^{k} \frac{|E(P_i)|}{|P_i|}$. Note that there is no restriction on the number of communities which yields some difference to the problem of partitioning into cliques. While there exists a partition into exactly $k$ sets of density $(n - k)/2$ if and only if the input graph can be partitioned into $k$ cliques [6], there can be a partition into less than $k$ sets with a density higher than $(n-k)/2$ even if the input cannot be partitioned into $k$ cliques. Alternatively, Dense Graph Partition can be modeled from a game theoretic perspective. Aziz et al. [4] study the Max Utilitarian Welfare problem where the vertices in a graph $G = (V, E)$ are agents, and each agent $x \in V$ validates its coalition $P \subseteq V$ with $x \in P$ by $\frac{1}{|P|}|\{u \in P \mid \{u, v\} \in E\}|$. Maximizing social welfare for this model is equivalent to Dense Graph Partition.

It is known that maximum matching is a 2-approximation [4], and there are a few improvements on specific graph classes: polynomial-time solvability on trees [19], $\frac{4}{3}$-approximation on maximum degree 3 graphs, and EPTAS for everywhere dense graphs [6]. This in particular gives rise to the questions: Can the 2-approximation be improved, at least on some more non-trivial graph classes? Does there exist a polynomial-time approximation scheme on general instances? Is it true that there is always an optimum solution where all parts induce a graph of diameter at most 2, a so-called *2-club clustering*? What is the complexity on graphs of bounded treewidth?

### Streaming Graph Clustering

Streaming Graph Clustering is commonly defined as follows: given a graph $G = (V, E)$, find a clustering $\mathcal{C} : V \to \mathcal{N}$ that maximizes a quality score such as modularity, using at most $O(|V|)$ memory. In the one-pass version, $E$ is an ordered list of edges and each edge can be read only once. A popular heuristic for this problem is SCoDA [29].

This matches well with real-world applications where graphs are discovered over time, e.g. in online social networks, as well as for graphs which are too large to cluster using standard $O(|E|)$ memory algorithms. However, the one-pass version is quite limiting and often results in low clustering quality [38].

The Incremental Graph Clustering model [38] is a *buffered* variant of the one-pass model where the ordered edge list is subdivided into *batches*. Unlike in buffered streaming graph partitioning [21], the batches are assumed to be given and not selected by the algorithm. Edges in a batch are read and processed in memory together. The algorithm can use $O(|E|)$ memory, but we require that running time for processing each batch does not depend on $|E|$, only on the size of the batch.

The Neighborhood-to-community link counting (NCLiC) [38] is a heuristic for this variant. For modularity clustering, it provides strong modularity retention compared to offline algorithms. It applies the Leiden Algorithm to each new batch and then merges it with the already processed graph.

**Open Problems.** The NCLiC algorithm keeps track of the approximate number of neighbors in each cluster. If a vertex changes community it will update its neighbors with a probability that depends on its degree. Skipping some updates allows maintaining the required running time, but introduces a reduction in clustering quality. Is there a data structure that allows keeping exact counts of the neighboring clusters without violating the running time constraint?

Another open question is: is it possible to modify NCLiC to use at most $O(|V|)$ memory while retaining most of the modularity retention qualities?

## 5.4 Data Reductions and Learning

### Data Reductions for (Hyper)Graph Decomposition

Most balanced (hyper)graph partitioning formulations are NP-hard: it is believed that no polynomial-time algorithm exists that always finds an optimal solution. However, many NP-hard problems have been shown to be fixed-parameter tractable (FPT): large inputs can be solved efficiently and provably optimally, as long as some problem parameter is small. Over the last two decades, significant advances have been made in the design and analysis of fixed-parameter algorithms for a wide variety of graph-theoretic problems. Moreover, in recent years a range of methods from the area have been shown to improve implementations

drastically. For example for the maximum (weight) independent set problem [27]. Here, data reductions rules transform the input into a smaller one that still contains enough information to be able to recover the optimum solution. For the maximum independent set problem, this enabled highly scalable exact solvers that can solve instances with millions of vertices to optimality. For balanced partitioning this has currently not been carefully investigated. However, here are some very simple data reduction rules. For example, removing a vertex of degree one, then solving the smaller subproblem with same balance constraint and afterwards assigning the vertex to a block with leftover capacity, is a valid data reductions rule. This yields the natural open questions: are there more and highly effective data reduction rules for balanced (hyper)graph partitioning problems? These rules could be helpful in two ways: they could speedup current heuristic solvers, e.g. multilevel (hyper)graph partitioning algorithms, and they could help to build more scalable exact partitioning algorithms (see Section 5.2).

After all reduction rules for kernel computations have been applied, the final smaller instance can still be too large to be solved to optimality within a reasonable time bound. This is a serious problem as the overall goal of the algorithms is to solve the given problem instance. The idea of lossy kernelization is as follows: when no more reductions can be applied, i.e. a problem core has been computed, one may shrink the input further while guaranteeing that the optimal solution value changes only slightly. Then a good approximate solution of the reduced input can be lifted to a good approximate solution of the original input. This has recently been done for the vertex cover problem [40]. The natural question that arises is can these techniques be applied to balanced (hyper)graph partitioning as well?

As the (lossy) kernel/core still contains the optimum solution (or some approximation thereof) in some sense, this has a large potential to speed up the (multilevel) heuristic while not sacrificing solution quality. Additionally, running a fast algorithm on the large kernel can help to identify parts of the instances that are likely to be in a good solution. Those parts can then be put into a partial solution and the remaining instance can be reduced recursively.

It could also be possible to use machine learning to learn lossy reductions for a wide-range of problems in this area. For example, one could use learning to predict if two vertices should be clustered together or to decide if an edge is a cut edge or not. The basic idea is then to use a classification model to learn which parts of the input can be pruned, i.e. are unlikely or highly likely in an optimum solution. In the first case, a solution omits this part of the input, in the latter case this part of the input will be included in the solution. For example, [39] propose to use machine learning frameworks to automatically learn lossy reductions for the maximum clique enumeration problem and [74] shows that this learning-to-prune framework is effective on a range of other combinatorial optimization problems. The classification model can be a deep neural network in an end-to-end framework or a classifier with significantly fewer parameters such as SVM or random forest if a deeper integration of machine learning and algorithmic techniques is done. The latter will require carefully engineered features based on existing heuristics.

### Learning for Local Search in Multi-level (Hyper)graph Partitioners

Machine learning techniques can also be used to learn more efficient refinement steps. Existing refinement steps in multi-level graph partitioning techniques rely on solving a flow problem or iterative moves of Kernighan–Lin or Fiduccia–Mattheyses heuristic. However, solving flow problems can be quite slow (given the number of times it is called). Similarly, the number of possible moves that need to be explored for finding a good step using Kernighan–Lin or Fiduccia–Mattheyses can be quite high. It is worthwhile exploring if learning techniques can be used to predict good regions where the flow algorithm can focus. This can improve the

trade-off between the time to solve the flow problem and the gain from it for the refinement part. For the case of the Kernighan–Lin or Fiduccia–Mattheyses heuristic, the interesting question is whether learning techniques such as reinforcement learning can be used to learn a good sequence of moves for these local search heuristics. This has the potential to reduce the search space that needs to be explored to find good local moves. For training the learning techniques, the R-MAT graph generator from the Graph500 benchmark can be used.

## 5.5 Embeddings

### Distance Estimation for Process Mapping

Process mapping is a super-problem of graph partitioning, in which vertices of some source graph $S$ have to be assigned (i.e., mapped) to vertices of some target graph $T$, by way of a mapping function $\tau_{S,T} : V(S) \longrightarrow V(T)$, so that an objective function is minimized. In the field of parallel computing, source graphs commonly represent computations to be performed, usually multiple times in sequence, while target graphs represent processing elements and interconnection networks of multi-processor and/or multi-computer hardware architectures. The objective function to minimize is the amount of data to be exchanged across the interconnection network, so as to reduce its congestion, provided that every processing element in $V(T)$ receives roughly the same number of vertices of $S$ (or, more generally, equivalent vertex weights with respect to its compute power), to minimize computation imbalance. In this context, partitioning some graph $S$ into $k$ parts amounts to mapping $S$ onto $K(k)$, the complete graph of order $k$, since in this case all processing elements are at the same distance from all the others.

In the Dual Recursive Bipartitioning (DRB) algorithm [54] used by the Scotch software, computing the mapping of $S$ onto $T$ requires to be able to estimate the shortest-path distance in $T$ between any two vertex subsets of $V(T)$ called the *subdomains* of $V(T)$. These subdomains are not arbitrary, since they result from recursively bipartitioning the graph $T$ into pieces of roughly the same size in a way that minimizes the cut of the interconnection network. Being able to compute the distance between any two subdomains allows the DRB algorithm to estimate the penalty of assigning some vertex $v$ of $S$ to either one of two sibling subdomains of $T$, by estimating the distance between these subdomains and those to which all the neighbor vertices $v$ of $u$ have already been mapped. When the recursive bipartitioning of $T$ is perfectly balanced, the number of subdomains of $T$ is $2|V(t)| - 1$.

A way to quickly obtain the distance between any two subdomains of some target graph $T$ is to pre-compute a distance matrix between all of them, of a size in $O(|V(T)|^2)$. While this solution works for small target graphs, it is no longer applicable when mapping onto big parts of very big target architectures. To solve this problem, one has to find a more compact (in terms of data storage) and quick (in terms of retrieval time) method to produce these distance estimates. An important condition on these approximations is that distances should become more accurate as subdomains are smaller and closer to each other in $T$.

In [55, 56], it has been shown that, for target architectures for which the recursive bipartitioning of subdomains, and the distances between subdomains, can be computed algorithmically, by way of explicit functions (e.g. , for regular vendor architectures such as meshes, butterfly graphs, etc.), a bipartition tree, created by way of recursive matching and coarsening of the whole target graph, allows one to represent any subset, even disconnected, of the processing elements of these target architectures. The DRB algorithm can therefore be applied to them.

However, for irregular architectures (e.g. , those represented by irregular graphs), the question remains open. It can be expressed in the following form: "How can one get cheaply (both in terms of memory and computation time) approximate distances between any pair of the subgraphs yielded by the recursive bipartition of some irregular graph?"

### Space-Efficient Planar Graph Embedding

When one opens up a publication regarding planar graph bisubsections, one often reads a sentence akin to: Without loss of generality, assume that the input graph is embedded in the plane and maximal planar. Famous works that makes use of this specific property is the balanced separator theorem due to Lipton et al. [42], which states that every planar graph has a balanced vertex separator of size $O(\sqrt{n})$. Standard recursive bisubsection algorithms for planar graphs are based on this theorem, which are able to construct the entire recursive bisubsection in linear time [33]. Often it is easy to assume such an embedding, as it can be computed in linear time using $O(n \log n)$ bits of space, i.e., a linear number of words. In sub-linear space settings one can compute an embedding in polynomial time (albeit with an extremely large polynomial degree). Now, the question remains: what can one achieve when aiming for a linear time algorithm, while using $o(n \log n)$ bits, or ideally, $O(n)$ bits? The standard linear time algorithms are quite involved, but on the most basic level many of them use a simple depth-first tree and compute a constant number of, but seemingly critical, variables per vertex. Even when aiming for a much lower goal: check if the input graph is planar within $O(n \log n)$ time while using $o(n \log n)$ bits, there is no obvious way to tackle this problem. As graphs grow larger and larger, such questions of space-efficiency become of higher interest. Especially with the direct application of graph partitioning algorithms that rely on such embeddings.

### Finding Moore-Bound-Efficient Diameter-3 Graphs

In graph theory, given a graph with degree $d$ and diameter $k$, the largest number of vertices in that graph can be determined using the Moore bound. Recent technological advances in photonics technology have greatly increased the number of links – or degree $d$ – of the network routers, improving the scalability of large supercomputers. While Moore-bound-optimal diameter-2 graphs have recently been engineered to span a few thousand nodes [36], emerging AI and graph applications are demanding larger configurations. Unfortunately, diameter-3 graphs are still elusive, with Moore's bound efficiencies of only 15%. The construction of more efficient diameter-3 graphs would directly impact the design of emerging photonics systems for large scale graphs [36, 37], data analysis, and AI applications.

## 5.6 Parameterized Complexity

### Parameterized Complexity of Layered Giant Graph Decomposition

**Direction 1.** An important theme – or challenge – for theoretical computer science, that has been recognized for decades, is the observation that has been made prominently by Richard Karp and others that we don't really understand very well natural input distributions. *It is remarkable how well sometimes very simple heuristics work in practice for problems that are known to be NP-hard.* There must be some sort of structure, but what is it? And if we knew, could we exploit that in designing algorithms?

A striking example of this was described by Karsten Weihe in an old paper entitled, "On the Differences Between Practical and Applied" which was about Weihe's experience doing quite practical computing for a simple Hitting Set application in real-world computing where his project was tasked with computing a minimum number of stations that could service all of the trains of Germany.

The model is a straightforward bipartite graph, with trains on one side, and stations on the other, and an edge if a train $t$ stops at a station $s$. There are two simple pre-processing vertex deletion rules: (1) If $N(s)$ is a subset of $N(s')$, then delete $s$. (2) If $N(t)$ is a subset of $N(t')$ then delete $t'$. Weihe found that these two simple reduction rules cascade back-and–forth on the gigantic real-world train graphs, and one ends up with (using PC terminology) a kernelized instance that consists of disjoint connected components that have size at most around 50, so the problem can be solved optimally by analyzing the connected components separately.

From the standpoint of parameterized complexity theory, we could simply declare the structural parameter of interest to be: $k$ ="the maximum connected component size of the network $G'$ that results when opportunities to apply the two reduction rules have been exhausted". This would be perfectly legal in the mathematical framework of PC – we could call the parameter the Weihe-width of the Hitting Set instance and have a pretty good FPT algorithm for computing what we could call a Weihe-width decomposition. This is legal, but from a traditional parameterized algorithms and complexity perspective, not entirely satisfying.

At the expense of quadratic blowup one can combinatorially reduce the very important medical- and bio-informatics problem of Feature Selection to Hitting Set in the following natural way. We now have enormous amounts of information concerning the genes that are being expressed into RNA, and so each patient in our hypothetical hospital has a gene activation profile. And each patient either does, or does not, have cancer.

We want to know a small subset of the genes to pay attention to so that we can accurately predict the outcome. On the one side, we have a vertex for each pair of patients that have differing outcomes, and on the other side, we have one vertex for each gene. It is surprising that Weihe's two reduction rules work quite practicably, in this very different real-world large data context. What is going on, and how can we generalize?

**Open Problem.** Can a Weihe-width $k$ decomposition of a graph of size $n$ be computed in truly linear FPT time?

A second open problem begins by reconsidering the most central example of an FPT graph problem, Vertex Cover, that has inspired in various ways a surprising amount of theoretical work in the parameterized complexity research community. For example, the recent work reported at IJCAI 2020 on the parameterized complexity *paradigm of solution diversity* began with an initial FPT result about the naturally parameterized Diverse Vertex Cover problem.

A (parameterized) vertex cover of a graph is a set of $k$ vertices $V'$ of $G = (V, E)$ such that the largest connected component of $G' = G - V'$ is size one! In other words, deleting the vertices of $V'$ kills off all the edges of $G$, yielding, if we want to call it that, a very nice decomposition of $G'$ into clusters of extremely high data-integrity and coherence, as each connected component consists of a single vertex.

It might seem that the Vertex Cover problem is so simple that it might be irrelevant for giant graphs. But by setting thresholds for declaring edges, it has been used very effectively in stages in very large dataset bioinformatics, e.g., Dehne's CLUSTAL W package for multiple sequence alignment [15].

A key point is that the successful CLUSTAL W algorithm begins by decomposing a sparse graph constructed by making edges between vertices (data objects) that are emphatically NOT similar. The impulse would be to seek cliques of compatible vertices, but here is exploited that the naturally parameterized Clique and Vertex Cover problems are parametrically dual, and from that point the CLUSTAL W algorithm proceeds in stages with an initial decomposition based on a vertex cover cutset on a sparse graph based on thresholding the NOT similarity that makes an edge in the initial graph.

The following problem explores a generalization where the resulting connected components ("clusters") satisfy other simple integrity requirements. It is called *Vertex Decomposition into Small Dominator Clusters*: given a graph $G = (V, E)$. and parameter $(k, d)$, the question is can we delete $k$ vertices from $G$, obtaining $G'$ such that every connected component of $G'$ has domination number at most $d$? It is interesting to start by asking if this might be FPT for the vector parameter $(k, d)$. But, if we fix $k = 0$, then the problem is W[2] - complete. We can still hope for a parameterized tractability result, where $d$ is allowed to play an XP-role in the exponent of the polynomial and for fixed $d$, with parameter $k$ we get FPT.

**Open Problem.** Is this FPT? And if so, can the corresponding decomposition be computed in truly linear FPT time for $d = 1$?

Note that we could define endlessly many interesting and largely unexplored parameterized problems in a similar manner where the decomposition is modeled by connected components formed by essentially a cutset. And there is also the possibility of interestingly layered decompositions of this kind. For example in the *Layered Vertex Cover* problem: given a graph $G = (V, E)$. and parameter $(k, k', k'')$ the question is can we delete $k$ vertices from $G$, obtaining $G'$ such that every connected component $C$ of $G'$ has the property that: $k'$ vertices can be deleted from $C$ resulting in a graph $C''$ such that each connected component of $C''$ has a vertex cover of size at most $k''$?

Or perhaps our particular application intention might be naturally served by deleting $k$ vertices so that the resulting connected components have nice properties governed by a parameter $t$, and these components can be further decomposed into connected components with a different nice property governed by $t'$ and so on.

**Direction 2.** The theme of fairly simple and elemental decompositions based on vertex- and edge-cutsets is important.

**Direction 3.** Since the size $n$ of the networks (graphs) targeted in this application area is huge, the attention should be focused on *truly linear-time FPT*, that is, processing that is simply of $O(n)$ cost, regardless of any parameterization $k$ that we might want to consider. Polynomial time $O(n^c)$ with no exponential costs associated to the parameter $k$, is the best kind of FPT. For very large graphs, we need $c = 1$. Slightly more generally, we might consider reasonable FPT processing time-costs of the form $O(n^c + f(k))$, where again the exponent of the polynomial part is $c = 1$, which we will call truly linear time FPT. This is an area of PC structural complexity theory little explored. There is a small amount of relevant recent work by Jianer Chen and coauthors.

## FPT Approximation of Vertex Bisubsection

Edge (resp. Vertex) Bisubsection is one the fundamental graph partitioning problems, where given a graph $G$ and an integer $k$, the goal is to find a set of at most $k$ edges (resp. vertices), say $S$, such that the vertex set of $G \setminus S$ can be partitioned into two almost equal parts $V_1$ and $V_2$, that is $||V_1| - |V_2|| \leq 1$, and there are no edges between a vertex of $V_1$ and

$V_2$, that is $E(V_1, V_2) = \emptyset$. In the regime of parameterized complexity, Edge Bisubsection admits a fixed-parameter tractable (FPT) algorithm parameterized by the solution size $k$. In particular, it admits an algorithm running in time $2^{O(k \log k)} n^{O(1)}$ [18], where $n$ is the number of vertices in the input graph.

**Open Problem 1.** One can solve Edge Bisubsection, for fixed $k$, in linear time? Preferably, is there an algorithm solving Edge Bisubsection in $2^{O(k \log k)} n$-time?

**Open Problem 2.** Does Edge Bisubsection admit an algorithm with running time $2^{O(k)} n^{O(1)}$? Or can one show that there is no algorithm for this problem that runs in time $2^{o(k \log k)} n^{O(1)}$ under reasonable complexity assumptions?

In contrast to Edge Bisubsection, Vertex Bisubsection is known to be W[1]-hard [44], that is it is unlikely that it admits an FPT algorithm parameterized by $k$. On the kernelization front, Edge Bisubsection cannot admit a polynomial kernel under reasonable complexity assumptions [79]. This leads of interesting questions regarding the fixed-parameter tractability and/or kernelization *with approximations* for these problems. In particular, the following questions remain intriguing.

**Open Problem 3.** Does Edge Bisubsection admit a polynomial $\alpha$-lossy kernel, for some $\alpha > 1$? Are there lossy reduction rules that help in solving the problem practically?

**Open Problem 4.** Does Vertex Bisubsection admit an FPT-approximation algorithm? That is, in time $f(k, \epsilon) n^{O(1)}$, can one find a set of at most $(1 + \epsilon)k$ vertices, say $S$, such that $V(G \setminus S) = V_1 \uplus V_2$, $||V_1| - |V_2|| \leq 1$ and $E(V_1, V_2) = \emptyset$, or report that there is no such set $S$ of size at most $k$?

**Open Problem 5.** In scenarios where Vertex Bisubsection pops up in practical usage, can we identify some structure on the instances? For example, can we say that the graphs in interesting instances belong to some nice graph class, or "is close to" being in a graph class (this could, for example, be formalized using distance to triviality measures), or have some bounded parameter. If such an identification is possible, studying these scenarios theoretically may lead to interesting insights about the problem.

### FPT in Decomposition

*The first open question here is if* Densest $k$-Subgraph *FPT parameterized by modular-width?* Given a graph $G$ and an integer $k$, the Densest $k$-Subgraph problem asks for a subgraph of $G$ with at most $k$ vertices maximizing the number of edges. It is known that this problem is FPT by stronger parameters such as neighborhood diversity and twin cover, yet it is W[1]-hard by weaker clique-width. Modular-width is defined using the standard concept of modular decomposition [23]. Any graph can be produced via a sequence of the following operations:

1. Introduce: Create an isolated vertex.
2. Union $G_1 \oplus G_2$: Create the disjoint union of two graphs $G_1$ and $G_2$.
3. Join: Given two graphs $G_1$ and $G_2$, create the complete join $G_3$ of $G_1$ and $G_2$. That is, a graph $G_3$ with vertices $V(G_1) \cup V(G_2)$ and edges $E(G_1) \cup E(G_2) \cup \{(v, w) : v \in G_1, w \in G_2\}$.
4. Substitute: Given a graph $G$ with vertices $v_1, \ldots, v_n$ and given graphs $G_1, \ldots, G_n$, create the *substitution* of $G_1, \ldots, G_n$ in $G$. The substitution is a graph $\mathcal{G}$ with vertex set $\bigcup_{1 \leq i \leq n} V(G_i)$ and edge set $\bigcup_{1 \leq i \leq n} E(G_i) \cup \{(v, w) : v \in G_i, w \in G_j, (v_i, v_j) \in E(G)\}$. Each graph $G_i$ is substituted for a vertex $v_i$, and all edges between graphs corresponding to adjacent vertices in $G$ are added.

These operations, taken together in order to construct a graph, form a *parse-tree* of the graph. The width of a graph is the maximum size of the vertex set of $G$ used in operation (O4) to construct the graph. The *modular-width* is the minimum width such that $G$ can be obtained from some sequence of operations (1)-(4). Finding a parse-tree of a given graph, called a *modular decomposition*, can be done in linear-time [75].
*The second open question is whether we can develop a framework of approximate modular decomposition applicable to real-world datasets?* Unfortunately, most real-world graphs tend to have larger modular-width. It would be beneficial if we can efficiently build non-exact parse trees with much lower width but without losing much information. Possible avenues of exploration include graph editing, a relaxed definition of the parse-tree, and a data-driven approach.

## Advancing the Parameterized View on Graph Modification

One of the most explored topics in parameterized complexity are so called *distance to triviality problems* (see, for example, [22, 30]). The intuitive question behind these problems is always "can we make a small change to our input so that it takes on some property?". In terms of graph problems, one can state a meta-problem as follows, where $\mathcal{P}$ is a graph property *Vertex-Deletion-To-*$\mathcal{P}$: given a graph $G$, an integer $k$ as well as a parameter $k$, the question is can we delete at most $k$ vertices from $G$, such that the resulting graph has property $\mathcal{P}$?

For many graph properties for which one can consider this meta-problem, either tractable algorithms or complexity lower bounds are known. On the other hand, in some application areas it could be useful to delete as many vertices as possible, while ensuring that the resulting graph has a certain property. This leads to the *Max-Vertex-Deletion-To-*$\mathcal{P}$ problem: given a graph $G$, an integer $k$ as well as a parameter $k$, the question is can we delete *at least* $k$ vertices from $G$, such that the resulting graph has property $\mathcal{P}$?

While the change to the problem statement is deceptively simple, we have to this date no complexity-theoretic insight into this class of problems. Note that this problem also differs from the widely used kernelization techniques, as in kernelization, we ask for the resulting input size to be bounded by, for example, $f(k)$. As parameterized complexity can be seen as providing a mathematically rigorous framework of preprocessing through the rich methods of kernelization techniques and algorithmics for distance to triviality problems, extending this framework to further variants of preprocessing seems very natural and could provide further complexity-theoretic and algorithmic insights. These techniques could potentially be useful in the area of (hyper)graph decomposition.

## References


1 T. Akiba, Y. Iwata, and Y. Yoshida. Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In *Proc. 2013 ACM SIGMOD Int. Conf. on Management of Data*, page 349–360, New York, NY, USA, 2013.
2 E. Angel, E. Bampis, A. Kononov, D. Paparas, E. Pountourakis, and V. Zissimopoulos. Clustering on $k$-edge-colored graphs. *Discrete Applied Mathematics*, 211:15–22, 2016.
3 A. Angone, X. Liu, R. Shaydulin, and I. Safro. Hybrid quantum-classical multilevel approach for maximum cuts on graphs. *IEEE High-Performance and Extreme Computing*, 2023.
4 H. Aziz, S. Gaspers, J. Gudmundsson, J. Mestre, and H. Täubig. Welfare maximization in fractional hedonic games. In Q. Yang and M. J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 461–467. AAAI Press, 2015.
5 G. Ballard, A. Druinsky, N. Knight, and O. Schwartz. Hypergraph partitioning for sparse matrix-matrix multiplication. *ACM Transactions on Parallel Computing (TOPC)*, 3(3):1–34, 2016.

6 C. Bazgan, K. Casel, and P. Cazals. Dense graph partitioning on sparse and dense graphs. In A. Czumaj and Q. Xin, editors, *18th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT*, volume 227 of *LIPIcs*, pages 13:1–13:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

7 C. Bichot and P. Siarry, editors. *Graph Partitioning*. Wiley, 2011.

8 R. H. Bisseling. *Parallel Scientific Computation: A Structured Approach using BSP and MPI*. Oxford University Press, Oxford, UK, Mar. 2004.

9 R. H. Bisseling and W. F. McColl. Scientific computing on bulk synchronous parallel architectures. In B. Pehrson and I. Simon, editors, *Technology and Foundations: Information Processing '94, Vol. I*, volume 51 of *IFIP Transactions A*, pages 509–514. Elsevier Science Publishers, Amsterdam, 1994.

10 A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz. *Recent Advances in Graph Partitioning*, pages 117–158. Springer International Publishing, Cham, 2016.

11 Ü. V. Çatalyürek and C. Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 10(7):673–693, Jul 1999.

12 Ü. V. Çatalyürek and C. Aykanat. *PaToH: A Multilevel Hypergraph Partitioning Tool, Version 3.0*. Bilkent University, Department of Computer Engineering, Ankara, 06533 Turkey. PaToH is available at `http://bmi.osu.edu/~umit/software.htm`, 1999.

13 Ü. V. Çatalyürek and C. Aykanat. A fine-grain hypergraph model for 2D decomposition of sparse matrices. In *IPDPS*, pages 118–123. IEEE, 2001.

14 Ü. V. Çatalyürek, K. D. Devine, M. F. Faraj, L. Gottesbüren, T. Heuer, H. Meyerhenke, P. Sanders, S. Schlag, C. Schulz, D. Seemaier, and D. Wagner. More recent advances in (hyper)graph partitioning. *ACM Comput. Surv.*, 55(12):253:1–253:38, 2023.

15 J. Cheetham, F. K. H. A. Dehne, S. Pitre, A. Rau-Chaplin, and P. J. Taillon. Parallel CLUSTAL W for PC clusters. In *ICCSA (2)*, volume 2668 of *Lecture Notes in Computer Science*, pages 300–309. Springer, 2003.

16 J. Chen and I. Safro. Algebraic distance on graphs. *SIAM Journal on Scientific Computing*, 33(6):3468–3490, 2011.

17 A. Crane, B. Lavallee, B. D. Sullivan, and N. Veldt. Overlapping and robust edge-colored clustering in hypergraphs. *arXiv preprint arXiv:2305.17598*, 2023.

18 M. Cygan, P. Komosa, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, S. Saurabh, and M. Wahlström. Randomized contractions meet lean decompositions. *ACM Trans. Algorithms*, 17(1):6:1–6:30, 2021.

19 J. Darlay, N. Brauner, and J. Moncel. Dense and sparse graph partition. *Discret. Appl. Math.*, 160(16-17):2389–2396, 2012.

20 D. Delling, D. Fleischman, A. V. Goldberg, I. P. Razenshteyn, and R. F. Werneck. An exact combinatorial algorithm for minimum graph bisection. *Math. Program.*, 153(2):417–458, 2015.

21 M. F. Faraj and C. Schulz. Buffered streaming graph partitioning. *ACM Journal of Experimental Algorithmics*, 27:1–26, 2022.

22 F. V. Fomin, P. A. Golovach, and D. M. Thilikos. On the parameterized complexity of graph modification to first-order logic properties. *Theory Comput. Syst.*, 64(2):251–271, 2020.

23 J. Gajarský, M. Lampis, and S. Ordyniak. Parameterized algorithms for modular-width. In G. Gutin and S. Szeider, editors, *Parameterized and Exact Computation*, pages 163–176. Springer International Publishing, 2013.

24 S. Ganguly, T. McKenzie, S. Mohanty, and N. Srivastava. Many nodal domains in random regular graphs. *arXiv preprint arXiv:2109.11532*, 2021.

**25** S. Ganguly and N. Srivastava. On non-localization of eigenvectors of high girth graphs. *International Mathematics Research Notices*, 2021(8):5766–5790, 2021.

**26** A. Grandjean and B. Uçar. On partitioning two dimensional finite difference meshes for distributed memory parallel computers. In M. Aldinucci, D. D'Agostino, and P. Kilpatrick, editors, *Proceedings of the 22th Euromicro Conference on Parallel, Distributed, and Network-based Processing*, pages 9–16, Turin, Italy, February 2014. IEEE Computer Society, Conference Publishing Services.

**27** E. Großmann, S. Lamm, C. Schulz, and D. Strash. Finding near-optimal weight independent sets at scale. In *GECCO*, pages 293–302. ACM, 2023.

**28** Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

**29** A. Hollocou, J. Maudet, T. Bonald, and M. Lelarge. A linear streaming algorithm for community detection in very large networks. *arXiv preprint arXiv:1703.02955*, 2017.

**30** B. M. P. Jansen, J. J. H. de Kroon, and M. Włodarczyk. Vertex deletion parameterized by elimination distance and even less. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, page 1757–1769, New York, NY, USA, 2021. Association for Computing Machinery.

**31** E. L. Jenneskens and R. H. Bisseling. Exact $k$-way sparse matrix partitioning. In *Proceedings 12th IEEE Workshop Parallel/Distributed Combinatorics and Optimization (PDCO 2022)*, pages 754–763. IEEE Press, 2022.

**32** J. Kim, I. Hwang, Y. H. Kim, and B. R. Moon. Genetic Approaches for Graph Partitioning: A Survey. In *Proc. of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO'11)*, pages 473–480. ACM, 2011.

**33** P. N. Klein, S. Mozes, and C. Sommer. Structured recursive separator decompositions for planar graphs in linear time. Association for Computing Machinery, 2013.

**34** T. E. Knigge and R. H. Bisseling. An improved exact algorithm and an NP-completeness proof for sparse matrix bipartitioning. *Parallel Computing*, 96:102640, 2020.

**35** I. Koutis, A. Levin, and R. Peng. Faster spectral sparsification and numerical algorithms for SDD matrices. *ACM Trans. Algorithms*, 12(2), 2015.

**36** K. Lakhotia, M. Besta, L. Monroe, K. Isham, P. Iff, T. Hoefler, and F. Petrini. PolarFly: A cost-effective and flexible low-diameter topology. In F. Wolf, S. Shende, C. Culhane, S. R. Alam, and H. Jagode, editors, *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis, Dallas, TX, USA, November 13-18, 2022*, pages 12:1–12:15. IEEE, 2022.

**37** K. Lakhotia, L. Monroe, K. Isham, M. Besta, N. Blach, T. Hoefler, and F. Petrini. PolarStar: Expanding the scalability horizon of diameter-3 networks. *CoRR*, abs/2302.07217, 2023.

**38** J. Langguth, A. Tumanis, and A. Azad. Incremental clustering algorithms for massive dynamic graphs. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 360–369. IEEE, 2021.

**39** J. Lauri, S. Dutta, M. Grassia, and D. Ajwani. Learning fine-grained search space pruning and heuristics for combinatorial optimization. *J. Heuristics*, 29(2):313–347, 2023.

**40** B. Lavallee, H. Russell, B. D. Sullivan, and A. van der Poel. Approximating vertex cover using structural rounding. In *ALENEX*, pages 70–80. SIAM, 2020.

**41** L. Li, H. Chen, P. Li, J. Han, G. Wang, and G. Zhang. The $k$-core decomposition algorithm under the framework of GraphBLAS. In *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–7. IEEE, 2021.

**42** R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980.

**43** X. Liu, H. Ushijima-Mwesigwa, I. Ghosh, and I. Safro. Partitioning dense graphs with hardware accelerators. In *International Conference on Computational Science*, pages 476–483. Springer, 2022.

**44** D. Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.

**45** A. Mastoras, S. Anagnostidis, and A. N. Yzelman. Design and implementation for non-blocking execution in GraphBLAS: Tradeoffs and performance. *ACM Trans. Archit. Code Optim.*, 20(1), Nov 2022.

**46** A. Mastoras, S. Anagnostidis, and A. N. Yzelman. Nonblocking execution in GraphBLAS. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 230–233, 2022.

**47** R. Mayer and H.-A. Jacobsen. Hybrid edge partitioner: Partitioning large power-law graphs under memory constraints. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 1289–1302, New York, NY, USA, 2021. Association for Computing Machinery.

**48** R. Mayer, K. Orujzade, and H.-A. Jacobsen. Out-of-core edge partitioning at linear run-time. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2629–2642, 2022.

**49** N. Merkel, R. Mayer, T. A. Fakir, and H.-A. Jacobsen. Partitioner selection with ease to optimize distributed graph processing. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2400–2414, 2023.

**50** M. Y. Özkaya, A. Benoit, and Ü. V. Çatalyürek. Improving locality-aware scheduling with acyclic directed graph partitioning. In R. Wyrzykowski, E. Deelman, J. Dongarra, and K. Karczewski, editors, *Parallel Processing and Applied Mathematics*, pages 211–223, Cham, 2020. Springer International Publishing.

**51** P. A. Papp, G. Anegg, and A. N. Yzelman. DAG scheduling in the BSP model. *arXiv preprint arXiv:2303.05989*, 2023.

**52** P. A. Papp, G. Anegg, and A. N. Yzelman. Partitioning hypergraphs is hard: Models, inapproximability, and applications. In *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '23, pages 415–425, New York, NY, USA, 2023. Association for Computing Machinery.

**53** D. Pasadakis, O. Schenk, V. Vlačić, and A. N. Yzelman. Nonlinear spectral clustering with C++ GraphBLAS, 2023. Pre-print.

**54** F. Pellegrini. Static mapping by dual recursive bipartitioning of process and architecture graphs. In *Proceedings of IEEE Scalable High Performance Computing Conference*, pages 486–493, 1994.

**55** F. Pellegrini and C. Lachat. Process Mapping onto Complex Architectures and Partitions Thereof. In *SIAM Conference on Computational Science & Engineering*, Salt Lake City, United States, Mar. 2015. SIAM.

**56** F. Pellegrini and C. Lachat. Process Mapping onto Complex Architectures and Partitions Thereof. Research Report RR-9135, Inria Bordeaux Sud-Ouest, Dec. 2017.

**57** D. M. Pelt and R. H. Bisseling. A medium-grain method for fast 2D bipartitioning of sparse matrices. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, pages 529–539. IEEE, 2014.

**58** D. M. Pelt and R. H. Bisseling. An exact algorithm for sparse matrix bipartitioning. *Journal of Parallel and Distributed Computing*, 85:79–90, 2015.

**59** M. Popp, S. Schlag, C. Schulz, and D. Seemaier. *Multilevel Acyclic Hypergraph Partitioning*, pages 1–15. SIAM, 2021.

**60** D. Ron, I. Safro, and A. Brandt. A fast multigrid algorithm for energy minimization under planar density constraints. *Multiscale Modeling & Simulation*, 8(5):1599–1620, 2010.

**61** D. Ron, I. Safro, and A. Brandt. Relaxation-based coarsening and multiscale graph organization. *Multiscale Modeling & Simulation*, 9(1):407–423, 2011.

**62** I. Safro, D. Ron, and A. Brandt. Graph minimum linear arrangement by multilevel weighted edge contractions. *Journal of Algorithms*, 60(1):24–41, 2006.

**63** I. Safro, D. Ron, and A. Brandt. A multilevel algorithm for the minimum 2-sum problem. *J. Graph Algorithms Appl.*, 10(2):237–258, 2006.

**64** I. Safro, D. Ron, and A. Brandt. Multilevel algorithms for linear ordering problems. *Journal of Experimental Algorithmics (JEA)*, 13:1–4, 2009.

**65** I. Safro, P. Sanders, and C. Schulz. Advanced coarsening schemes for graph partitioning. *Journal of Experimental Algorithmics (JEA)*, 19:1–24, 2015.

**66** I. Safro and B. Temkin. Multiscale approach for the network compression-friendly ordering. *Journal of Discrete Algorithms*, 9(2):190–202, 2011.

**67** K. Schloegel, G. Karypis, and V. Kumar. Graph Partitioning for High-Performance Scientific Simulations. In *Sourcebook of parallel computing*, pages 491–541. Morgan Kaufmann Publishers, 2003.

**68** A. Scolari and A. N. Yzelman. Effective implementation of the High Performance Conjugate Gradient benchmark on GraphBLAS, 2023. Accepted for publication.

**69** R. Shaydulin, J. Chen, and I. Safro. Relaxation-based coarsening for multilevel hypergraph partitioning. *Multiscale Modeling & Simulation*, 17(1):482–506, 2019.

**70** H. D. Simon and S.-H. Teng. How good is recursive bisection? *SIAM Journal on Scientific Computing*, 18(5):1436–1445, 1997.

**71** D. G. Spampinato, D. Jelovina, J. Zhuang, and A. N. Yzelman. Towards structured algebraic programming. In *Proceedings of the 9th ACM SIGPLAN International Workshop on Libraries, Languages and Compilers for Array Programming*, ARRAY 2023, pages 50–61, New York, NY, USA, 2023. Association for Computing Machinery.

**72** D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

**73** J. Sybrandt, R. Shaydulin, and I. Safro. Hypergraph partitioning with embeddings. *IEEE Transactions on Knowledge and Data Engineering*, 34(6):2771–2782, 2020.

**74** D. Tayebi, S. Ray, and D. Ajwani. Learning to prune instances of $k$-median and related problems. In C. A. Phillips and B. Speckmann, editors, *Proceedings of the Symposium on Algorithm Engineering and Experiments, ALENEX 2022, Alexandria, VA, USA, January 9-10, 2022*, pages 184–194. SIAM, 2022.

**75** M. Tedder, D. Corneil, M. Habib, and C. Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In *Automata, Languages and Programming*, pages 634–645, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

**76** C. R. Trott, D. Lebrun-Grandié, D. Arndt, J. Ciesko, V. Dang, N. Ellingwood, R. Gayatri, E. Harvey, D. S. Hollman, D. Ibanez, N. Liber, J. Madsen, J. Miles, D. Poliakoff, A. Powell, S. Rajamanickam, M. Simberg, D. Sunderland, B. Turcksin, and J. Wilke. Kokkos 3: Programming model extensions for the exascale era. *IEEE Transactions on Parallel and Distributed Systems*, 33(4):805–817, 2022.

**77** B. Uçar and Ü. V. Çatalyürek. On the scalability of hypergraph models for sparse matrix partitioning. In M. Danelutto, J. Bourgeois, and T. Gross, editors, *Proceedings of the 18th Euromicro Conference on Parallel, Distributed, and Network-based Processing*, pages 593–600. IEEE Computer Society, Conference Publishing Services, 2010.

**78** L. G. Valiant. A bridging model for parallel computation. *Communications of the ACM*, 33(8):103–111, 1990.

**79** R. van Bevern, A. E. Feldmann, M. Sorge, and O. Suchý. On the parameterized complexity of computing balanced partitions in graphs. *Theory Comput. Syst.*, 57(1):1–35, 2015.

**80** B. Vastenhouw and R. H. Bisseling. A two-dimensional data distribution method for parallel sparse matrix-vector multiplication. *SIAM Review*, 47(1):67–95, 2005.

**81** D.-L. Wang and P. Wang. Discrete isoperimetric problems. *SIAM Journal on Applied Mathematics*, 32(4):860–870, 1977.

**82** A. N. Yzelman. Humble heroes, 2022. Pre-print.

**83** A. N. Yzelman and R. H. Bisseling. Cache-oblivious sparse matrix–vector multiplication by using sparse matrix partitioning methods. *SIAM Journal on Scientific Computing*, 31(4):3128–3154, 2009.

**84** A. N. Yzelman and R. H. Bisseling. Two-dimensional cache-oblivious sparse matrix–vector multiplication. *Parallel Computing*, 37(12):806 – 819, 2011.

**85** A. N. Yzelman, R. H. Bisseling, D. Roose, and K. Meerbergen. MulticoreBSP for C: a high-performance library for shared-memory parallel programming. *International Journal on Parallel Programming*, 42:619–642, 2014.

**86** A. N. Yzelman, D. Di Nardo, J. M. Nash, and W. J. Suijlen. A C++ GraphBLAS: specification, implementation, parallelisation, and evaluation. Preprint, 2020.

**87** A. N. Yzelman and D. Roose. High-level strategies for parallel shared-memory sparse matrix–vector multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 25(1):116–125, 2014.

## Participants

- Deepak Ajwani
University College Dublin, IE
- Cevdet Aykanat
Bilkent University – Ankara, TR
- Rob Bisseling
Utrecht University, NL
- Katrin Casel
HU Berlin, DE
- Ümit V. Çatalyürek
Georgia Institute of Technology – Atlanta, US & Amazon Web Services, US
- Cedric Chevalier
CEA – Bruyères-le-Châtel, FR
- Florian Chudigiewitsch
Universität zu Lübeck, DE
- Michael R. Fellows
University of Bergen, NO
- Marcelo Fonseca Faraj
Universität Heidelberg, DE
- Lars Gottesbüren
KIT – Karlsruher Institut für Technologie, DE
- Tobias Heuer
KIT – Karlsruher Institut für Technologie, DE
- George Karypis
University of Minnesota – Minneapolis, US
- Kamer Kaya
Sabanci University – Istanbul, TR
- Jakub Lacki
Google – New York, US
- Johannes Langguth
Simula Research Laboratory – Oslo, NO
- Xiaoye Sherry Li
Lawrence Berkeley National Laboratory, US
- Fredrik Manne
University of Bergen, NO
- Ruben Mayer
Universität Bayreuth, DE
- Johannes Meintrup
THM – Gießen, DE
- Henning Meyerhenke
HU Berlin, DE
- Yosuke Mizutani
University of Utah – Salt Lake City, US
- Francois Pellegrini
University of Bordeaux, FR
- Fabrizio Petrini
Intel Labs – Menlo Park, US
- Frances A. Rosamond
University of Bergen, NO
- Ilya Safro
University of Delaware – Newark, US
- Sebastian Schlag
Apple – Cupertino, US
- Christian Schulz
Universität Heidelberg, DE
- Daniel Seemaier
KIT – Karlsruher Institut für Technologie, DE
- Roohani Sharma
MPI für Informatik – Saarbrücken, DE
- Darren Strash
Hamilton College – Clinton, US
- Blair D. Sullivan
University of Utah – Salt Lake City, US
- Bora Uçar
ENS – Lyon, FR
- Albert-Jan Yzelman
Huawei Technologies – Zürich, CH

Report from Dagstuhl Seminar 23332

# Synergizing Theory and Practice of Automated Algorithm Design for Optimization

**Diederick Vermetten*[1], Martin S. Krejca†[2], Marius Lindauer†[3], Manuel López-Ibáñez†[4], and Katherine M. Malan†[5]**

1 **Leiden University, NL.** d.l.vermetten@liacs.leidenuniv.nl
2 **LIX, Ecole Polytechnique, IP Paris, Palaiseau, FR.** martin.krejca@polytechnique.edu
3 **Institute of AI, Leibniz University Hannover, DE.** m.lindauer@ai.uni-hannover.de
4 **Alliance Manchester Business School, University of Manchester, UK.** manuel.lopez-ibanez@manchester.ac.uk
5 **Department of Decision Sciences, University of South Africa – Pretoria, ZA.** malankm@unisa.ac.za

## Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23332, which focused on automated algorithm design (AAD) for optimization. AAD aims to propose good algorithms and/or parameters thereof for optimization problems in an automated fashion, instead of forcing this decision on the user. As such, AAD is applicable in a variety of domains. The seminar brought together a diverse, international set of researchers from AAD and closely related fields. Especially, we invited people from both the empirical and the theoretical domain. A main goal of the seminar was to enable vivid discussions between these two groups in order to synergize the knowledge from either domain, thus advancing the area of AAD as a whole, and to reduce the gap between theory and practice. Over the course of the seminar, a good mix of breakout sessions and talks took place, which were very well received and which we detail in this report. Efforts to synergize theory and practice bore some fruit, and other important aspects of AAD were highlighted and discussed. Overall, the seminar was a huge success.

**Seminar** August 13–18, 2023 – https://www.dagstuhl.de/23332

**2012 ACM Subject Classification** Theory of computation → Mathematical optimization; Computing methodologies → Machine learning; Theory of computation → Design and analysis of algorithms

**Keywords and phrases** automated algorithm design, hyper-parameter tuning, parameter control, heuristic optimization, black-box optimization

**Digital Object Identifier** 10.4230/DagRep.13.8.46

* Editorial Assistant / Collector
† Editor / Organizer



Synergizing Theory and Practice of Automated Algorithm Design for Optimization, *Dagstuhl Reports*, Vol. 13, Issue 8, pp. 46–70
Editors: Martin S. Krejca, Marius Lindauer, Manuel López-Ibáñez, and Katherine M. Malan

Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

# 1 Executive Summary

*Martin S. Krejca (LIX, Ecole Polytechnique, IP Paris, FR, martin.krejca@polytechnique.edu)*
*Marius Lindauer (Institute of AI, Leibniz University Hannover, DE, m.lindauer@ai.uni-hannover.de)*
*Manuel López-Ibáñez (Alliance Manchester Business School, University of Manchester, UK, manuel.lopez-ibanez@manchester.ac.uk)*
*Katherine M. Malan (Department of Decision Sciences, University of South Africa – Pretoria, ZA, malankm@unisa.ac.za)*



Automated algorithm design (AAD) is a prevalent and highly important domain, as design-dependent algorithms are used in a plethora of very different areas. Determining which algorithms and/or their parameters to choose for each given problem is a task that is infeasible to solve in this magnitude solely by humans. Computers can greatly assist in this process in a multitude of ways. Consequently, many different branches of AAD exist. This Dagstuhl Seminar brought together a diverse set of researchers from the AAD domain as well as closely related domains, from all over the world. The aim of the seminar was to provide a common platform to exchange ideas about all of the different established AAD techniques and theories, how to advance the field of AAD, and how to join forces. Most prominently, the seminar aimed at a strong exchange between researchers who work empirically and those who work theoretically. We believe that this seminar was a great success, reaching all of the intended goals.

Due to the diverse backgrounds of the participants, the seminar started with elaborate introduction rounds followed by two overview talks – one detailing the theoretical perspective of AAD, given by Carola Doerr, the other one the empirical perspective, given by Katherine M. Malan. From then on, the seminar consisted of a mix of breakout sessions and short, inspiring talks as participants felt the need to share their experiences or ideas based on the discussions. Immediately from day one, interesting breakout sessions emerged, some of which had multiple iterations, due to the interest of the participants and due to the scope of the subject. In the morning of each day, we made sure to provide a summary of all breakout sessions of the previous day in order to update everyone and to discuss what other topics could be further explored.

Topics that came up on day one were discussions about the terminology and about benchmarks, which had multiple sessions throughout the week. Both of these topics are very important when aiming to unite the community and especially when trying to bridge the gap between theory and practice. The sessions about terminology showed that different communities use different terms for similar concepts and, perhaps more worrying, terminology is not used consistently across communities, making communicating results between different areas harder. Potential solutions include to provide a freely accessible overview of the different terms and define them well, or to use whatever terminology feels appropriate for each article but specify the terms very well in the article. The sessions about benchmarks highlighted that more diverse sets of benchmark functions are desired and that it is not straightforward to transfer the existing benchmarks into a setting that is well suited for theoretical investigations. The latter point sparked other interesting discussions, summarized in this report.

An inspiring talk was given by Kevin Leyton-Brown, who proposed that optimization should be guided based on users' utilities rather than on expected runtime of the algorithms. This led to an interesting breakout group in which various useful utility functions were discussed which could be used for realistic AAD scenarios. Another very interesting breakout group was inspired by Benjamin Doerr, who was looking for the easiest possible AAD scenario to study theoretically. In a fruitful exchange, a scenario was constructed that seemed interesting and approachable from a theoretical point and also useful from a practical point of view. The discussion emerging from this group delved into an open question that was discussed throughout the seminar, namely, what are the characteristics that make AAD scenarios a special case of the more general field of black-box optimization.

Besides these success stories, further talks led to potential collaborations and new ideas. One example was the talk by Johannes Lengler, in which he proposed how existing theoretical results for randomized optimization heuristics could be potentially interpreted as results for AAD. The ensuing discussion was about to what extent such generalizations make sense, with no clear consensus being reached so far. Nonetheless, the talk and the following in-depth discussion showed that there is significant work to be done on this topic.

The seminar also gave members who are not part of the core AAD community the opportunity to present their work and to integrate into the community. This was met with very positive feedback. Overall, we thank all of the participants for the great discussions, valuable talks, and the support, all of which made this seminar a great success. In addition, we thank the Dagstuhl staff, who supported us incredibly well and helped us run this seminar as smoothly as it did.

## 2 Table of Contents

**Working Groups**

# 3 Overview of Talks

## 3.1 Reflections on Synergizing Theory and Practice of Automated Algorithm Design for Optimization

*Carola Doerr (Sorbonne Université, CNRS, LIP6 – Paris, FR, carola.doerr@lip6.fr)*



This seminar is aimed at bringing together theoreticians in (black-box) optimization with researchers developing and using automated algorithm design techniques. In order to create a joint understanding for the topics of this seminar, I will briefly present some reflections on what I consider useful to know to get to know each other. In particular, I will emphasize that some participants may have a strong interest in optimizing automated algorithm design techniques, while others may have a stronger interest in using automated algorithm design techniques to identify efficient optimization algorithms.

I will also present some ideas for possible breakout sessions, including using benchmarks with proven theoretical guarantees to understand behavior and performance limits of state-of-the-art automated algorithm design techniques, the selection of instances to use during the design (training) process, proxies (e.g., for performance measures) that can be used in the training process to make it more efficient, and automated algorithm configuration techniques that output an explicit instance feature – parameter map.

## 3.2 Runtime Analysis for the NSGA-II

*Benjamin Doerr (Ecole Polytechnique, IP Paris, FR, lastname@lix.polytechnique.fr)*


**Joint work of** Weijie Zheng, Yufei Liu, Benjamin Doerr
**Main reference** Weijie Zheng, Yufei Liu, Benjamin Doerr: "A First Mathematical Runtime Analysis of the Non-dominated Sorting Genetic Algorithm II (NSGA-II)", in Proc. of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 – March 1, 2022, pp. 10408–10416, AAAI Press, 2022.
**URL** http://dx.doi.org/10.1609/AAAI.V36I9.21283

Recently, the mathematical runtime analysis of evolutionary algorithms made a huge step forward by conducting several analyses of the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) [1], the most prominent multi-objective evolutionary algorithm intensively used in practice (the work [1] is cited more than 50,000 times). In this talk, I touch upon three results in this research direction.

- The first such runtime analysis [2] proved that the NSGA-II easily computes the Pareto front of the bi-objective OneMinMax and LOTZ benchmarks when the population size is at least four time the size of the Pareto front. In contrast, when the population size is only equal to the size of the Pareto front, regularly desirable solutions are lost and consequently, it takes at least exponential time to reach a population that does not miss a constant fraction of the Pareto front.
- In sharp contrast to these and several other positive results for bi-objective problems [3, 4, 5, 6, 7, 8, 9, 10], in three or more objectives the NSGA-II cannot even optimize the simple OneMinMax problem [11].
- On the positive side, as the first runtime analysis for this algorithm shows, the NSGA-III [12] can efficiently optimize the 3-objective OneMinMax problem [13].

**References**

1 Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.
2 Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In *Proc. of AAAI '22*, pages 10408–10416, 2022.
3 Chao Bian and Chao Qian. Better running time of the non-dominated sorting genetic algorithm II (NSGA-II) by using stochastic tournament selection. In *Proc. of PPSN '22*, pages 428–441, 2022.
4 Sacha Cerf, Benjamin Doerr, Benjamin Hebras, Jakob Kahane, and Simon Wietheger. The first proven performance guarantees for the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) on a combinatorial optimization problem. In *Proc. of IJCAI '23*, pages 5522–5530, 2023.
5 Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. Analysing the robustness of NSGA-II under noise. In *Proc. of GECCO '23*, pages 642–651, 2023.
6 Duc-Cuong Dang, Andre Opris, Bahare Salehi, and Dirk Sudholt. A proof that using crossover can guarantee exponential speed-ups in evolutionary multi-objective optimisation. In *Proc. of AAAI '23*, pages 12390–12398, 2023.
7 Benjamin Doerr and Zhongdi Qu. A first runtime analysis of the NSGA-II on a multimodal problem. *IEEE Transactions on Evolutionary Computation*, 27(5), 2023.
8 Benjamin Doerr and Zhongdi Qu. From understanding the population dynamics of the NSGA-II to the first proven lower bounds. In *Proc. of AAAI '23*, pages 12408–12416, 2023.
9 Benjamin Doerr and Zhongdi Qu. Runtime analysis for the NSGA-II: provable speed-ups from crossover. In *Proc. of AAAI '23*, pages 12399–12407, 2023.
10 Weijie Zheng and Benjamin Doerr. Better approximation guarantees for the NSGA-II by using the current crowding distance. In *Proc. of GECCO '22*, pages 611–619, 2022.
11 Weijie Zheng and Benjamin Doerr. Runtime analysis for the NSGA-II: proving, quantifying, and explaining the inefficiency for three or more objectives. *CoRR*, abs/2211.13084, 2022.
12 Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. 18:577–601, 2014.
13 Simon Wietheger and Benjamin Doerr. A mathematical runtime analysis of the Non-dominated Sorting Genetic Algorithm III (NSGA-III). In *Proc. of IJCAI '23*, pages 5657–5665, 2023.

## 3.3 A graphical overview of applied research in automated algorithm design

*Katherine Malan (University of South Africa – Pretoria, ZA, malankm@unisa.ac.za)*



With the aim of building a common understanding among the participants of seminar 23332, I have attempted to unpack the components of automated algorithm design (AAD) to visually illustrate the goals of different scenarios in AAD. After describing the different components (problem space, algorithm space, problem feature space, algorithm feature space, and performance space) and their specialized types, I describe some recent studies in AAD visually overlayed onto the framework. Studies from the three scenarios of automated

algorithm configuration, composition and selection are covered. The purpose of the talk is to generate discussion with the hope that this leads to a common understanding of some of the aspects of AAD.

## 3.4 Modular Optimization Algorithms: Testing AAD for Continuous Optimization

*Diederick Vermetten (Leiden University – NL, d.l.vermetten@liacs.leidenuniv.nl)*



I present a use case of AAD in the domain of continuous optimization where we found that modularizing popular algorithms and applying configuration techniques can yield clear improvements in anytime performance. We do however have problems when testing generalizability, as our test suites have not been designed for this purpose. We thus propose a new function generator: MA-BBOB, which we hope can provide a playground for testing the generalizability of AAD methods in optimization.

## 3.5 A OneMax for Automated Algorithm Design?

*Johannes Lengler (ETH Zürich, CH, johannes.lengler@inf.ethz.ch)*



In this talk I present one candidate for a "OneMax" function in automated algorithm composition, i.e., a simple benchmark that we would expect an automated algorithm composer to solve.

The suggestion is that the potential algorithm consists of n components. For each of them there are two components, type 0 and type 1. When the composer wants to compare two or more algorithms, it draws a random permutation of size n, and gives credit $2^i$ to the $i$-th component of the algorithm if this is of type 1, and credit 0 if it is of type 0. The total fitness performance of an algorithm is then the sum of the credits of all its components.

I discussed connections of this benchmark to the class of monotone functions (like HotTopic functions) and of dynamic BinVal functions that have been used as benchmarks in the context of randomized optimization heuristics.

## 3.6 Success Story: Power of Hyper-Heuristics and Transfer Learning in AAD

*Nelishia Pillay (University of Pretoria, ZA, nelishia.pillay@up.ac.za)*



Automated design (AutoDes) encompasses automated configuration, automated composition, automated generation and automated selection. We focus on automated configuration (AutoCon), automated composition (AutoCom) and automated generation (AutoGen) for

finding scalable solutions to complex problems in agriculture, health, education and industry. We investigate hyper-heuristics and transfer learning in these three AutoDes areas. We measure success in terms of the human competitiveness, i.e., do the solutions outperform human solutions, as well as in terms of providing improvements over existing techniques used for solving these problems. Hyper-heuristics work in an alternate space that maps to the design space and as such allows for areas that could not be reached by searching the design space directly to be explored. Transfer learning involves transferring knowledge between design spaces. We have found that using hyper-heuristics and transfer learning in AutoCom, AutoCon and AutoGen provide human competitive results with reduced computational cost.

## 3.7 Bimodal Parameter Landscapes in EDAs, Phase Transitions, and Stability

*Carsten Witt (Technical University of Denmark, Kgs. Lyngby, DK, cawi@dtu.dk)*



We consider the simple estimation-of-distribution algorithm cGA on the OneMax benchmark function. Previous research by Lengler, Sudholt and Witt (Algorithmica 2021) shows that its expected runtime depends in a non-trivial way on its update parameter $K$, resulting in a bimodal landscape and two widely different settings minimizing the expected runtime. Moreover, we discuss phase transitions in algorithm behavior related to the choice of $K$, with chaotic behavior of marginal probabilities and high genetic drift below the phase transition, and very stable behavior above it.

## 3.8 A few open questions related to automated algorithm selection

*Mario Andrés Muñoz (OPTIMA, The University of Melbourne, AU, munoz.m@unimelb.edu.au)*



From over ten years of work on algorithm selection for continuous black-box optimization, I have more open questions than answers. In this talk, I discussed some of my most recent projects and, what they are to me, important challenges worth tackling. First, it is the understanding of the relationships between real-world problems and benchmarks, by being able to place them in a common instance space. Related, is the need to have more informative descriptions of problems that are interpretable to humans and not only machine learning algorithms. Second, we need to describe algorithm behaviors, and strategies to find suitable algorithms for hard problems, where selection seems insufficient. All of this must be supported by an infrastructure that facilitates sharing of results, data and software.

## 3.9 Searching Problem Spaces: Automated Problem Design?

*Marcus Gallagher (The University of Queensland – Brisbane, AU, marcusg@uq.edu.au)*



Problem instances are a key component of automated algorithm design in optimization, particularly when we are concerned with benchmarking and comparing the performance of different techniques. Problem instances may be produced via manual design, randomized generators, feature space or derived from real-world domains. Alternatively, we can produce problem instances via some kind of search over some appropriate problem space. The key challenges lie in how to search this space effectively. Some existing work in this space includes:

- Using racing algorithms across problem instances.
- Creating problem transformations (e.g. sphere to Rastrigin function), implicitly defining a path through the problem space.
- Search the (hyper)parameter space of a problem instance generator.
- Search the (hyper)parameter space of a surrogate model.
- Perturbation of a dataset (when the objective function is data driven).
- Use of some type of generative model.

There seems to be much room for further research in this direction. Seminar attendees began a collaborative document to collate references in this area.

## 3.10 Automated algorithm configuration/selection and constraint programming

*Nguyen Dang (University of St Andrews, UK, nttd@st-andrews.ac.uk)*



Constraint programming (CP) can offer an accessible means to non-expert users to solve their combinatorial problems. More concretely, constraint modeling and solving pipelines such as MiniZinc and Essence provide an expressive modeling language to describe a problem and the relevant data (problem instance) and a toolchain to translate/reformulate the problem (and instance) into the input understandable by a generic (CP/SAT/SMT/MIP) solver. In this talk, I will talk about two applications we have done in our CP group in St Andrews which link automated algorithm configuration/selection (AC/AS) and CP. The first one is an application of AC/AS on automated generation and selection of streamliner constraints, which was shown to significantly speed up the solving of constraint/SAT solvers. The second one is about combining AC/AS and CP to create an automated instance generation system for benchmarking purposes.

## 3.11 A Comparative study of NCO and heuristics on TSP

*Shengcai Liu (Agency for Science, Technology and Research, Singapore, SG, liu_shengcai@cfar.a-star.edu.sg)*


**Joint work of** Shengcai Liu, Yu Zhang, Ke Tang, Xin Yao
**Main reference** Shengcai Liu, Yu Zhang, Ke Tang, Xin Yao: "How Good is Neural Combinatorial Optimization? A Systematic Evaluation on the Traveling Salesman Problem", IEEE Computational Intelligence Magazine, Vol. 18(3), pp. 14–28, 2023.
**URL** http://dx.doi.org/10.1109/MCI.2023.3277768

Neural Combinatorial Optimization (NCO) is a recent research area in deep learning that uses deep reinforcement learning to train DNNs (policies) to solve combinatorial optimization problems (e.g., TSPs, VRPs). However, the experiments presented in existing works are generally non-conclusive due to several reasons: 1) The state-of-the-art heuristic solvers are usually missing, e.g., EAX for TSPs, HGS for VRPs; 2) For heuristic solvers, their default parameter values are used which neglects the fact of parameter tuning; 3) Unfair comparison of parallel NCO solvers vs. sequential heuristic solvers; 4) The benchmark instances are quite limited in terms of both types and sizes.

In this talk, I introduce our recent comparative study of NCO solvers and heuristic solvers on the TSPs. The study is the first that 1) considers five different problem types, 2) involves problem instances with up to 10000 nodes, 3) includes tuned traditional solvers in the comparison, and 4) investigates five different performance aspects including the energy consumption of the solvers. Four main conclusions are drawn from the experiments: 1) For all the TSP problem sizes and types considered in the experiments, heuristic solvers still significantly outperform NCO solvers in nearly all the five performance aspects; 2) A potential benefit of NCO solvers would be their superior time and energy efficiency for small-size problem instances when sufficient training instances are available; 3) current NCO approaches are not suitable for handling large-size problem instances, structural problem instances, and mixed problem instances; 4) When the training instances have different problem characteristics (problem types and sizes) from those of the testing instances, both NCO solvers and tuned traditional solvers exhibit performance degradation, and NCO solvers suffered from far more severe performance degradation

## 3.12 A Utilitarian Foundation for Algorithm Configuration

*Kevin Leyton-Brown (University of British Columbia – Vancouver, CA, kevinlb@cs.ubc.ca)*


**Joint work of** Devon R. Graham, Kevin Leyton-Brown, Tim Roughgarden
**Main reference** Devon R. Graham, Kevin Leyton-Brown, Tim Roughgarden: "Formalizing Preferences Over Runtime Distributions", in Proc. of the 40th International Conference on Machine Learning, Proceedings of Machine Learning Research, Vol. 202, pp. 11659–11682, PMLR, 2023.
**URL** https://proceedings.mlr.press/v202/graham23a.html

When trying to solve a computational problem, we are often faced with a choice between algorithms that are guaranteed to return the right answer but differ in their runtime distributions (e.g., SAT solvers, sorting algorithms). This work aims to lay theoretical foundations for such choices by formalizing preferences over runtime distributions. It might seem that we should simply prefer the algorithm that minimizes expected runtime. However, such preferences would be driven by exactly how slow our algorithm is on bad inputs, whereas

in practice we are typically willing to cut off occasional, sufficiently long runs before they finish. We propose a principled alternative, taking a utility-theoretic approach to characterize the scoring functions that describe preferences over algorithms. These functions depend on the way our value for solving our problem decreases with time and on the distribution from which captimes are drawn. We describe examples of realistic utility functions and show how to leverage a maximum-entropy approach for modeling underspecified captime distributions. Finally, we show how to efficiently estimate an algorithm's expected utility from runtime samples.

## 3.13 From Synthetic to Real-World to Application Benchmarks

*Marcel Wever (LMU Munich, DE, marcel.wever@ifi.lmu.de)*



Algorithm selectors and algorithm configurators are most commonly benchmarked on algorithms for computationally hard problems, e.g., SAT, TSP, or MIP. While such benchmarks facilitate unleashing the power of algorithm selection and configuration systems, results underlie limited interpretability. More specifically, if an algorithm configurator fails to yield a substantially better-performing configuration, it is unclear whether it is nonexistent or hard to find. From the viewpoint of a practitioner, however, it is not clear to what extent algorithm selection or configuration can be applied to a broader scope of software systems. In this presentation, I invite the community to create theoretically inspired, synthetic benchmarks that exhibit certain characteristics inherent to algorithm configuration problems on the one hand, and more applied benchmarks on the other hand.

## 3.14 On the Probabilistic Model-Based Evolutionary Algorithm for Mixed-Variables

*Shinichi Shirakawa (Yokohama National University, JP, shirakawa-shinichi-bg@ynu.ac.jp)*



In automated machine learning and automatic algorithm design, mixed types of design variables, including continuous, integer, and categorical variables, should be optimized. I briefly introduce the extension of CMA-ES, called CMA-ES with margin, to handle integer variables. CMA-ES with margin prevents stagnation of the search distribution to a certain integer value in mixed-integer problems by introducing lower-bounded marginal probabilities of integer variables. I also describe information geometric optimization (IGO) with categorical distribution, which is a probabilistic model-based evolutionary algorithm for categorical domains. CMA-ES with margin and IGO with categorical distribution update the parameters of Gaussian and categorical distributions, respectively, to search for better solutions. Then, I mention a possible extension of these algorithms for handling mixed-variable problems consisting of continuous, integer, and categorical variables.

## 3.15 Landscape Features in AAS/C

*Niki van Stein (Leiden University, NL, n.v.stein@liacs.leidenuniv.nl)*



We propose DoE2Vec, a variational autoencoder (VAE)-based methodology to learn optimization landscape characteristics for downstream meta-learning tasks, e.g., automated selection of optimization algorithms. Principally, using large training data sets generated with a random function generator, DoE2Vec self-learns an informative latent representation for any design of experiments (DoE). Unlike the classical exploratory landscape analysis (ELA) method, our approach does not require any feature engineering and is easily applicable for high dimensional search spaces. For validation, we inspect the quality of latent reconstructions and analyze the latent representations using different experiments. The latent representations not only show promising potentials in identifying similar (cheap-to-evaluate) surrogate functions, but also can significantly boost performances when being used complementary to the classical ELA features in classification tasks.

## 3.16 Automated Algorithm Design from the Bottom Up

*Lars Kothoff (University of Wyoming – Laramie, US, larsko@uwyo.edu)*



Recent advances in algorithm selection allow to automatically characterize algorithms based on their source code. This offers new directions for automated algorithm design, enabling researchers to quantify existing and generated code in novel ways and compare and contrast different sources. Most existing approaches for automated algorithm design are top-down, i.e. algorithms are designed or modified based on high-level descriptions. In this talk, I will give an overview of the new methods for characterizing source code and sketch future research directions that could leverage this, highlighting the results of a preliminary analysis of existing solvers for hard AI problems.

## 3.17 UNSAT Solver Synthesis via Monte Carlo Forest Search

*Chris Cameron (University of British Columbia – Vancouver, CA, cchris13@cs.ubc.ca )*



We introduce Monte Carlo Forest Search (MCFS), a class of reinforcement learning (RL) algorithms for learning policies in tree MDPs, for which policy execution involves traversing an exponential-sized tree. Examples of such problems include proving unsatisfiability of a SAT formula; counting the number of solutions of a satisfiable SAT formula; and finding the optimal solution to a mixed-integer program. MCFS algorithms can be seen as extensions of Monte Carlo Tree Search (MCTS) to cases where, rather than finding a good path (solution) within a tree, the problem is to find a small tree within a forest of candidate trees. We instantiate and evaluate our ideas in an algorithm that we dub Knuth Synthesis, an

MCFS algorithm that learns DPLL branching policies for solving the Boolean satisfiability (SAT) problem, with the objective of achieving good average-case performance on a given distribution of unsatisfiable problem instances. Knuth Synthesis leverages two key ideas to avoid the prohibitive costs of policy evaluations in an exponentially-sized tree. First, we estimate tree size by randomly sampling paths and measuring their lengths, drawing on an unbiased approximation due to Knuth (1975). Second, we query a strong solver at a user-defined depth rather than learning a policy across the whole tree, to focus our policy search on early decisions that offer the greatest potential for reducing tree size.

# 4 Working Groups

## 4.1 Working Group 1: Terminologies of Automated Algorithm Design

*Nelishia Pillay (University of Pretoria, ZA, nelishia.pillay@up.ac.za)*



This focus of this working group was to find a standardization of terminology for AAD. In the literature there is no consistency in terms defining AAD concepts.

### 4.1.1 Discussed Problems

The first session of the working group highlighted the following challenges that need to be addressed regarding the terminology used for AAD:

- Standardize terminology for AAD – different researchers use different terms for the same concepts, thus a standardization is needed.
- Remove confusion regarding AAD terms in the literature – due to different terms being used for the same concept this leads to confusion.
- The need of an extensible framework/standardization as the field develops further – AAD is a rapidly developing field and as such there is a need for an extensible standardization.
- Principle components of design need to be identified – a starting point of the standardization would be to identify the main/principle components of designs.

### 4.1.2 Possible Approaches

Criteria Considered for Categorization

- Underlying design problem – what is the design problem being solved, e.g. determining hyperparameters, creating a new operator.
- Outcome – What is the design that is produced? Does it have to be reusable?
- High-level methodology – What optimization technique is used to generate the design?

Proposed Categorization

- Automated configuration – Involves determining the parameters and hyper-parameters, e.g. learning rate, population size, operator probabilities. Related terms: HPO, parameter tuning, parameter control.
- Automated Selection – Involves selecting algorithms. Same underlying problem in automated configuration.
- Automated Composition – Involves combining existing processes/algorithms as black box components.
- Automated Generation – Involves creating new algorithms, e.g. new move operators, new genetic operators.

**Textbox 2: Automatic Algorithm Configuration**

In the machine learning community, **automatic algorithm configuration** is a method for determining a well-performing parameter configuration $\theta^* \in \Theta$ of a given algorithm $A \in \mathrm{P}$ across a given set of instances $i \in \Pi = \{i_1, \ldots, i_l\}$. Given a cost metric $c: \mathrm{P} \times \Pi \to \mathbb{R}$ that computes a cost $c(A(\theta), i)$ for a given algorithm and its configuration on an instance, a distribution $\boldsymbol{C}(A(\theta), \Pi)$ of such cost values over instances drawn from $\Pi$, and a function $m(\boldsymbol{C}(A(\theta), \Pi))$ that computes a statistical population parameter (e.g., mean, median, …) of $\boldsymbol{C}(A(\theta), \Pi)$, the **algorithm configuration problem** is to find the optimal parameter configuration $\theta^*$ for algorithm $A$ for the objective $m$, over the instance set $\Pi$:

$$\Theta^* = \arg\max_{\theta \in \Theta} m\left(\boldsymbol{C}(A(\theta), \Pi)\right)$$

**Figure 1** Proposed Mathematical Formulation.

Proposed Principle Components:

- Problem input – What is the input to the design program, e.g. values for parameters, potential components of an operator?
- Search space – What space is being explored to solve the problem, e.g. parameter space, program space, heuristic space.
- Methods – The optimization approaches used for search.
- Objective – What is the purpose of solving the problem, e.g. reduced workload, better design leading to better results.
- Outcome – The design produced.

Areas for further discussion:

- Is automated selection and automated configuration the same problem?
- Is the search space a principle component?

### 4.1.3 Conclusions

The discussions held provided a starting point to develop a standardization for AAD. There are some areas for further discussion where consensus was not reached. The aim is for this standardization proposal generated from the discussions to be taken further by interested members of the community for further discussion and possible formalization.

## 4.2 Working Group 2: Automated Algorithm Design Benchmarks

*Johannes Lengler (ETH Zurich, CH, johannes.lengler@inf.ethz.ch)*



This working group focused on how benchmark suites for Automated Algorithm Design should look like.

### 4.2.1 Discussed Problems

- What are landscape features? Do theoretical benchmarks capture them?
- Useful benchmarks for algorithm configuration
- Benchmarks on AAD

- AAD across combinatorial problem domains
- How to bridge theory and practice? What about theoretical empirics?
- Benchmarking on real-world applications
- Representativeness of benchmark data
- Theory-inspired benchmarks for understanding AAD
- Improving experimental methods for AAS etc.
- How to select instances for the transfer from training to testing
- Understanding problem spaces and problem properties better

### 4.2.2 Conclusions

- Benchmark sets in optimization and in AAD often tend to be small.
- Benchmarking is probably more instrumental in automated algorithm design than in "manual" algorithm design however we could not point to in which way benchmarking needs to be distinctively different.
- It is not clear whether we want to increase the size and diversity of the benchmark suites. In principle this is desirable (ML), but we do want our benchmark set to still reflect the real world. Data augmentation was successful in ML.
- Perhaps optimization heuristics even have a strength in being able to work with little training data (flexibility, adaptability).
- This touches on the question at which level the automation should take place. For one problem class? For everything? That decides the scope of the benchmark set.
- It is controversial how useful or arbitrary a feature space representation of benchmark problems is.
- Invariance properties may be a relevant and desirable aspect for features.

## 4.3 Working Group 3: New Methods for Automated Algorithm Configuration

*Marius Lindauer (Leibniz University Hannover, DE, m.lindauer@ai.uni-hannover.de)*



This working group focused on current and new methods for automated algorithm configuration (AAC).

### 4.3.1 Open and Understudied Problems

As a first discussion point, we discussed what is missing or not well-studied in AAC. This includes:

1. Multi-objective AAC guided by a surrogate (e.g., Bayesian optimization) s.t. users can optimize for complementary objectives and obtain an approximated Pareto front at the end [7];
2. AC methods with feature-dependent results (a.k.a per-instance algorithm configuration or instance-specific algorithm configuration) where the results are (I) interpretable (e.g., by providing symbolic formulas) or (II) derived from a grammar to be more expressive compared to common bounded configuration spaces;
3. Pre-trained surrogate models that are inspired the current trend of foundation models;

4. Text-based interfaces for AAC based on large language models (LLMS) to provide an easy user interface where domain experts can easily interact with [28];
5. Allowing more complicated constraints in the configuration space – currently this is dominated by rejection sampling that cannot deal with highly constrained spaces;
6. Interactive methods where one can aggregate several preferences across multiple interactions – the major difficulty lies in the aggregation;
7. Statistical guarantees s.t. users can trust that the results are close to optimal w.r.t. the evaluated configurations (e.g., the configuration returned is not worse than any other seen with 95% certainty);
8. How to deal with low signal-noise ratios where there is a lot (maybe even heteroscedastic) noise on algorithm evaluations?
9. How can theoretically-driven racing schemes (e.g. [30]) and user-defined utility-functions [13] be combined with Bayesian-optimization guided AAC?
10. How can we configure our AAC methods in view of how expensive it is to run AAC methods? Simply applying AAC to AAC will be too expensive in most cases or diminishing returns have to be taken into account [21];
11. Maybe surprisingly, the commonly used AAC packages (e.g., irace [22] and SMAC [18, 20]) do not implement all state-of-the-art ideas from other AI subfields (e.g., foundation models, constraint solving, LLMs and theoretical sound approaches).

### 4.3.2 Promising Next Steps

At the end of the discussions, several promising directions were identified:

1. What are actually the most important components of an AAC method? A systematic study across different AAC methods and application domains is still missing. This could be done based on AClib [19]. This will also require updating AClib with new scenarios, algorithms and instances.
2. How can we make results from AAC methods more interpretable to (I) increase the users' trust into these methods and to (II) enable theory-driven results based on empirical results? The common approaches include either using more interpretable models (e.g., linear models or symbolic models) or model-agnostic post-hoc explanations.
3. How can we increase the robustness of AAC methods where robustness can refer to noisy algorithm evaluations, heterogeneous instance sets or stochasticity of the AAC method?

### 4.3.3 Dedicated Discussion on Dynamic Algorithm Configuration

In contrast to the traditional approach of wrapping AAC around existing algorithms, an alternative is to tightly integrate AAC into the algorithms. This enables dynamic configuration (DAC) [1] while the algorithm is running and can, e.g., dynamically adapt to new challenges in the optimization landscape.

There are two major lines of work in this direction:

1. Configuring the parameters of the algorithm dynamically. This follows the idea of the traditional AAC methods that takes the algorithm as it is and only controls its parameters [27, 5]. With DACBench [10], there is a benchmark library that allows to benchmark such approaches on various AI subfields efficiently. In addition, there is a benchmark that is on the intersection of empirical experiments and theory [6].
2. Algorithm components can also be completely replaced by AI components and thus increase efficiency [26, 17, 8]. This can come with reduced theoretical guarantees since the learned components are typically based on neural networks.

Open problems for dynamic configurations include:

1. How to scale DAC to large configuration spaces (i.e. many algorithm parameters)? Most current approaches deal with less than five parameters and some even struggle to perform well on more than two parameters.
2. How to decide on which instances one should learn in a dynamic AAC strategy? An efficient learning strategy is required since dynamic configuration needs to perform well on new instances. Approaches from curriculum and self-paced learning could be promising [9].
3. When should DAC be applied? How to efficiently determine whether dynamic configuration is beneficial beyond static configuration?

### 4.3.4 Conclusions

Although there is active research on AAC methods for more than a decade and there is substantial progress (e.g., tremendous speedups, efficient scaling to more algorithm parameters (at least for static AAC) and dealing with different kinds of instances), there are still many open challenges that are not yet addressed fully and open up many further research directions.

## 4.4 Working Group 4: Utility Functions

*Kevin Leyton-Brown (University of British Columbia, Canada, kevinlb@cs.ubc.ca)*



This working group focused on how to apply a utilitarian perspective to the problem of algorithm configuration. We discussed the benefits of different (random vs stratified) sampling strategies to estimate utility value from samples. We considered the impact of a utilitarian approach on algorithm configuration methods that offer theoretical guarantees. We also discussed candidate utility functions that could be suggested to practitioners. In the end, we recommend a simple approach:

1. Is there a minimum amount of time $t_0$ below which all algorithm runs should be considered equivalent?
2. Is there a maximum amount of time $\bar{\kappa}$ above which the end user would never be willing to wait?
3. In between $t_0$ and $\bar{\kappa}$, how does our preference for time change? We foresee three common cases:
   a. $t_0 = \bar{\kappa}$: we have a step function
   b. linearly (e.g., we face a fixed cutoff and pay per hour of compute)
   c. geometrically (utility falls by a constant fraction per unit time)

## 4.5 Working Group 5: New Estimation-of-Distribution Algorithms

*Manuel López-Ibáñez (University of Manchester, UK, manuel.lopez-ibanez@manchester.ac.uk)*



This working group focused on estimation-of-distribution algorithms (EDAs) and how they could be used for AAD.

### 4.5.1 Discussed Problems

- What are the challenges for EDAs in the AAD scenarios:
  1. The search space is mixed (continuous, integer, categorical).
  2. There exists a hierarchy between the parameters, in the form of known *a priori* conditions.
  3. There might be unknown stochastic dependencies between parameters.
  4. The problem can be noisy, with non-Gaussian noise.
  5. Since we often consider multiple instances, this is another area in which a different kind of noise gets introduced.
- How can the sampling be made more efficient? In sig-cGA, some variables are fixed when you are sure they should have some particular value (good theoretical results but unclear experimental results).
- How to create good synthetic functions?
- How to handle the conditional parameters? Can we just ignore the condition and sample values which have no effect (which results in a random walk of the respective values)?

### 4.5.2 Possible Approaches

- Can we use the Bayesian optimization algorithm (BOA) with Bayesian networks (or P3 or GOMEA)? Can the dependency network be created, and could this handle mixed-integer variables?
  - The algorithms in questions should be capable to represent the dependency network well up to a degree.
  - EDAs are also capable to handle mixed-integer problems, but we came to no conclusion how well this would work for our purpose.
- Univariate EDAs require far less work to build a model. For certain contexts, e.g., where it is very hard to figure out actual dependencies, this might already be sufficient. We have some theoretical guarantees for how to choose the parameters of univariate EDAs, preventing them from converging prematurely to bad values.
- Density estimation trees or forests (e.g., the `mlpack` package): how to update the trees?
  - We did not come to a good conclusion.
- Use CMA-ES with handling of mixed-integer spaces (except categorical). There is a version of CMA-ES that can adapt the size of the covariance matrix.
- We can use techniques from deep reinforcement learning to learn the update function of CMA-ES. Is there a structural prior to use to speed up this training? Could we apply algorithm design, define a grammar, and apply genetic programming to this context?
  - We were indecisive about the next steps.

### 4.5.3 Conclusions

This area is heavily underexplored so far. Some AAD approaches, like irace, exist that can be thought of as EDAs. However, they do not make use of theoretical guarantees. Partially because the theoretical results are too recent, partially because we have no guarantees for multivariate EDAs. Hence, employing complex EDAs should follow proper empirical tests.

## 4.6 Working Group 6: The OneMax Problem of AAD

*Benjamin Doerr (Ecole Polytechnique, IP Paris, FR, lastname@lix.polytechnique.fr)*



The aim of this working group was to bring together researchers in AAD and in the theory of evolutionary computation to discuss what could be an AAD problem that is simple enough to allow analyses via mathematical means, but for which a deeper understanding promises to lead to useful insights on AAD methods used in practice.

The hope is that such a first simple problem can trigger a success story similar to the theoretical analysis of evolutionary algorithm, which has started with toy problems like how the $(1 + 1)$ evolutionary algorithm optimizes the OneMax problem [2, 12, 23], but which now, for example, is able to analyze complex algorithms intensively used in practice like the NSGA-II [4], the most prominent multi-objective evolutionary algorithm (50,000 citations).

Obviously, what is such a good first problem is highly non-obvious and the group discussed various directions. It was noted that some first works already exist [14, 15, 16], however, no participant of the breakout session had a deeper understanding of these works. This together with short preliminary readings of these works were interpreted in the way that most likely the problems regarded in these works do not yet perfectly satisfy the needs of the automated algorithm design community.

Finally, the following problem set-up was proposed. We try to optimize a pseudo-Boolean function $f : \{0, 1\}^n \to \mathbb{R}$. This function is defined via two functions $g_0, g_1$ defined on $n - 1$ bits via $f(x) = g_0(x_2, \dots, x_n)$ if $x_1 = 0$ and $f(x) = g_1(x_2, \dots, x_n)$ if $x_1 = 1$. This models an automated algorithm design problem in which we have the choice between two algorithms $A_0$ and $A_1$. This choice is described via the variable $x_1$. We further have $n - 1$ binary algorithm parameters. For $i = 0, 1$, if we run algorithm $A_i$ with parameters values $x_2, \dots, x_n$ on our problem, the performance is $f(i, x_2, \dots, x_n) = g_i(x_2, \dots, x_n)$. In this model, we assume that we have access to a precise and noise-free performance of the algorithms on our problem instance. We do, a priori, not assume that $g_0$ and $g_1$ are in some sense related, though most likely the more interesting results will be obtained if there is some correlation between the influences of the parameters on the result. An example for such a setting could be that both functions are linear functions with coefficients that mostly have the same sign. Also, it was pointed out that this model captures the situation that the two algorithms have different numbers of parameters, namely by letting one of the two functions only depend on some of the variables.

A particular research question that could be studied in this model is the following. To try to find out which algorithm with which parameters to use for our problem, is it better to optimize the parameters of the two algorithms separately, that is, we optimize $g_0$ and $g_1$ separately and take the better of the two results, or is it preferable to just optimize $f$, that is, let the algorithm design process switch between the algorithms.

It was later noted that a special case of the function $f$ was regarded in [29].

## 4.7 Working Group 7: Explainability of AAD Methods

*Pascal Kerschke (Technical University of Dresden, DE, pascal.kerschke@tu-dresden.de)*



This working group focused on the explainability of AAD, starting from the question of whether / why we would want to have explainable AAD.

### 4.7.1 Discussed Problems

- Everyone interprets explainability in their own way, ranging from understanding complementarity of algorithms and explaining the autoML process itself to explaining experimental data such as identification of flaws or biased data/setup of the process.
- Overarching question: when / why care for explainability if we just want better-performing algorithms?
- There are different levels of explainability: designers vs. users. For example, users want explainability to gain trust in the AAD frameworks / tools. Explanations are also always contextualized in your own (domain) knowledge, so different users might need different explanations.
- Can explainability help identify what parts of a problem are relevant, or help to identify useful guidelines from the vast amount of data collected by AAD approaches? Explanations might help understand the complexity of a model, e.g. that some HPO problems have low effective dimensionality [3].
- What are the limits of explainability: can we sufficiently explain very complex models with high fidelity? Do explanation techniques have sufficient explanatory power?
- To what extent are features useful for explainability? If we use features, should we limit ourselves only to those we fully understand and build simple models based on those?
- We need to be careful of the Rashomon effect: many possible explanations might exist based on spurious interactions, high correlation of features,...
- Diagnostic tools for the existing interpretable/explainable ML methods and theoretical proofs are still missing. Many of the IML methods are still very new, and not yet fully trusted.

### 4.7.2 Possible Approaches

- One potential way to help explainability is to use functional ANOVA or Generalized Additive Models for decomposition of the model.
- There exists a wide range of techniques which might be useful for the explainability of AAD: fANOVA, Symbolic Formulas, (automatic) Ablations, Partial Dependency Plots, Item Response Theory, Instance Footprint plots, Local Parameter Importance, landscape features, feature selection, simple explainable models, regularized models in expensive settings...

### 4.7.3 Conclusions

Explainability in general is gaining more attention in many communities, and there are ways in which AAD might also benefit from more explainable techniques. It is however still unclear when to focus on explainability over just improving the performance of the final algorithm. The quick growth of explainability techniques gives rise to a lot of opportunities, but we should not lose sight of the overall picture and reason about what we want to explain, why and to whom.

## 4.8 Working Group 8: Automated Machine Learning

*Marius Lindauer (Leibniz University Hannover, DE, m.lindauer@ai.uni-hannover.de)*
*Bernd Bischl (LMU, DE, bernd.bischl@stat.uni-muenchen.de)*



This working group focused on current challenges and promising future directions for automated machine learning (AutoML).

### 4.8.1 Discussed Problems

1. Although the efficiency of AutoML has improved by orders of magnitude in recent years, AutoML might still be too inefficient for very expensive models (e.g., LLMS).
2. AutoML is not always robust; e.g., a simple default pipeline sometimes (rarely) outperforms a complex AutoML system. How can users trust AutoML systems that the invested compute (and waiting) time is worth it?
3. AutoML still does not handle all kinds of pitfalls of ML. For example, (I) class imbalance is not properly dealt with in all tools, (II) wrong or inefficient validation schemes were used or (III) ensembling is not beneficial.
4. The user interfaces of many AutoML tools are not user-friendly. It is an open question of how verbose the output of an AutoML tool should be and what kind of warnings are required (e.g., class imbalance and fairness). Generally, designing good user interfaces without specifying the target user group is hard, but AutoML aims to be an approach for everyone.
5. AutoGluon [11] is very fast and delivers strong performance (in particular on tabular data for supervised tasks). Is AutoML for this subtask already solved or can we outperform it by using drastically different approaches?
6. Large language models (LLMS) pose a major challenge since they are expensive and combine several learning paradigms. Current AutoML approaches are hardly applicable [28].
7. The hidden objectives of practitioners are not always explicitly expressed. For example, some users care only about predictive performance and others want to have rather small and interpretable models. Not all AutoML tools can address different user preferences; e.g., AutoGluon returns neither small nor interpretable models.
8. How can AutoML help to understand the data in a better way (e.g., in data-centric approaches)?
9. How can we design an interactive AutoML system that (I) allows users to learn from the AutoML system about the AutoML process and the data, and (II) to specify their preferences, (III) benefits from users' input to improve the overall performance of the system (given the users' objectives)? This might be important s.t. practitioners do not get the impression that these systems can get out of control, but they are still in charge and thus can trust the system.
10. Surprisingly, there seems to be a large group of practitioners rather using optimizers for AutoML problems but fewer the full-fledged AutoML systems. These optimizers offer much more flexibility but require more expertise and effort to set them up. What is missing in full AutoML systems s.t. more practitioners want to use them? Learning search spaces on the fly could be one promising direction [25, 24].
11. Large language models (such as ChatGPT) could offer new ways to interact with AutoML systems (or partially replace them) s.t. users will need even less knowledge about how to code ML systems [31].

### 4.8.2 Conclusions

Although AutoML is much more mature than it was a few years ago, there are still major challenges in applying AutoML in practice without expertise in ML and AutoML. One of the most crucial next steps will be to bridge the gap between the expectations, expertise, and workflows of ML practitioners and AutoML tools.

### References


1 Steven Adriaensen, André Biedenkapp, Gresa Shala, Noor H. Awad, Theresa Eimer, Marius Lindauer, and Frank Hutter. Automated dynamic algorithm configuration. *Journal of Artificial Intelligence Research*, 75:1633–1699, 2022.

2 Thomas Bäck. Optimal mutation rates in genetic search. In *Proc. of ICGA '93*, pages 2–8, 1993.

3 James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

4 Weijie Zheng, Yufei Liu, and Benjamin Doerr. A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In *Proc. of AAAI '22*, pages 10408–10416, 2022.

5 André Biedenkapp, H. Furkan Bozkurt, Theresa Eimer, Frank Hutter, and Marius Lindauer. Dynamic algorithm configuration: Foundation of a new meta-algorithmic framework. In *Proc of ECAI '20*, pages 427–434, 2020.

6 André Biedenkapp, Nguyen Dang, Martin S. Krejca, Frank Hutter, and Carola Doerr. Theory-inspired parameter control benchmarks for dynamic algorithm configuration. In *Proc. of GECCO '22*, pages 766–775, 2022.

7 Aymeric Blot, Holger H. Hoos, Laetitia Jourdan, Marie-Éléonore Kessaci-Marmion, and Heike Trautmann. Mo-paramils: A multi-objective automatic algorithm configuration framework. In *Proc. of LION '10*, pages 32–47, 2016.

8 Chris Cameron, Jason S. Hartford, Taylor Lundy, Tuan Truong, Alan Milligan, Rex Chen, and Kevin Leyton-Brown. Monte carlo forest search: UNSAT solver synthesis via reinforcement learning. *CoRR*, abs/2211.12581, 2022.

9 Theresa Eimer, André Biedenkapp, Frank Hutter, and Marius Lindauer. Self-paced context evaluation for contextual reinforcement learning. In *Proc. of ICML '21*, pages 2948–2958, 2021.

10 Theresa Eimer, André Biedenkapp, Maximilian Reimer, Steven Adriaensen, Frank Hutter, and Marius Lindauer. DACBench: A benchmark library for dynamic algorithm configuration. In *Proc. of IJCAI '21*, pages 1668–1674, 2021.

11 Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *CoRR*, abs/2003.06505, 2020.

12 Josselin Garnier, Leila Kallel, and Marc Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7:173–203, 1999.

13 Devon R. Graham, Kevin Leyton-Brown, and Tim Roughgarden. Formalizing preferences over runtime distributions. In *Proc. of ICML '23*, pages 11659–11682, 2023.

14 George T. Hall, Pietro S. Oliveto, and Dirk Sudholt. On the impact of the cutoff time on the performance of algorithm configurators. In *Proc. of GECCO '19*, pages 907–915. 2019.

15 George T. Hall, Pietro S. Oliveto, and Dirk Sudholt. Analysis of the performance of algorithm configurators for search heuristics with global mutation operators. In *Proc. of GECCO '20*, pages 823–831. 2020.

16 George T. Hall, Pietro S. Oliveto, and Dirk Sudholt. Fast perturbative algorithm configurators. In *Proc. of PPSN '20*, pages 19–32. 2020.

**17** André Hottung and Kevin Tierney. Neural large neighborhood search for the capacitated vehicle routing problem. In *Proc. of ECAI '20*, pages 443–450, 2020.

**18** Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Proc. of LION '11*, pages 507–523, 2011.

**19** Frank Hutter, Manuel López-Ibáñez, Chris Fawcett, Marius Lindauer, Holger H. Hoos, Kevin Leyton-Brown, and Thomas Stützle. AClib: A benchmark library for algorithm configuration. In *Proc. of LION '14*, pages 36–40, 2014.

**20** Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23:54:1–54:9, 2022.

**21** Marius Lindauer, Matthias Feurer, Katharina Eggensperger, André Biedenkapp, and Frank Hutter. Towards assessing the impact of bayesian optimization's own hyperparameters. *CoRR*, abs/1908.06674, 2019.

**22** Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58, 2016.

**23** Heinz Mühlenbein. How genetic algorithms really work: mutation and hillclimbing. In *Proc. of PPSN '92*, pages 15–26, 1992.

**24** Felix Neutatz, Marius Lindauer, and Ziawasch Abedjan. Automl in heavily constrained applications. *CoRR*, abs/2306.16913, 2023.

**25** Valerio Perrone and Huibin Shen. Learning search spaces for bayesian optimization: Another view of hyperparameter transfer learning. In *Proc. of NeurIPS '19*, pages 12751–12761, 2019.

**26** Gresa Shala, André Biedenkapp, Noor H. Awad, Steven Adriaensen, Marius Lindauer, and Frank Hutter. Learning step-size adaptation in CMA-ES. In *Proc. of PPSN '20*, pages 691–706, 2020.

**27** David Speck, André Biedenkapp, Frank Hutter, Robert Mattmüller, and Marius Lindauer. Learning heuristic selection with dynamic algorithm configuration. In *Proc. of ICAPS '21*, pages 597–605, 2021.

**28** Alexander Tornede, Difan Deng, Theresa Eimer, Joseph Giovanelli, Aditya Mohan, Tim Ruhkopf, Sarah Segel, Daphne Theodorakopoulos, Tanja Tornede, Henning Wachsmuth, and Marius Lindauer. AutoML in the age of large language models: Current challenges, future opportunities and risks. *CoRR*, abs/2306.08107, 2023.

**29** Ingo Wegener and Carsten Witt. On the analysis of a simple evolutionary algorithm on quadratic pseudo-boolean functions. *Journal of Discrete Algorithms*, 3:61–78, 2005.

**30** Gellért Weisz, András György, Wei-I Lin, Devon R. Graham, Kevin Leyton-Brown, Csaba Szepesvári, and Brendan Lucier. ImpatientCapsAndRuns: Approximately optimal algorithm configuration from an infinite pool. In *Proc. of NeurIPS '20*, 2020.

**31** Lei Zhang, Yuge Zhang, Kan Ren, Dongsheng Li, and Yuqing Yang. MLCopilot: Unleashing the power of large language models in solving machine learning tasks. *CoRR*, abs/2304.14979, 2023.

## Participants


- Thomas Bäck
Leiden University, NL
- Bernd Bischl
LMU München, DE
- Chris Cameron
University of British Columbia – Vancouver, CA
- Nguyen Dang
University of St Andrews, GB
- Benjamin Doerr
Ecole Polytechnique – Palaiseau, FR
- Carola Doerr
Sorbonne University – Paris, FR
- Tome Eftimov
Jozef Stefan Institute – Ljubljana, SI
- Marcus Gallagher
The University of Queensland – Brisbane, AU
- Nikolaus Hansen
INRIA Saclay – Palaiseau, FR
- Pascal Kerschke
TU Dresden, DE
- Lars Kotthoff
University of Wyoming – Laramie, US
- Martin S. Krejca
Ecole Polytechnique – Palaiseau, FR
- Johannes Lengler
ETH Zürich, CH
- Kevin Leyton-Brown
University of British Columbia – Vancouver, CA
- Marius Lindauer
Leibniz Universität Hannover, DE
- Shengcai Liu
A*STAR – Singapore, SG
- Manuel López-Ibáñez
University of Manchester, GB
- Katherine M. Malan
UNISA – Pretoria, ZA
- Mario Andrés Muñoz
The University of Melbourne, AU
- Nelishia Pillay
University of Pretoria, ZA
- Shinichi Shirakawa
Yokohama National University, JP
- Thomas Stützle
Free University of Brussels, BE
- Niki van Stein
Leiden University, NL
- Diederick Vermetten
Leiden University, NL
- Marcel Wever
LMU München, DE
- Carsten Witt
Technical University of Denmark – Lyngby, DK


Report from Dagstuhl Seminar 23341

# Functionally Safe Multi-Core Systems

**Georg von der Brüggen**[*1], **Ian Gray**[*2], **and Catherine Nemitz**[†3]

1 **TU Dortmund, DE.** georg.von-der-brueggen@tu-dortmund.de
2 **University of York, GB.** ian.gray@york.ac.uk
3 **Davidson College, US.** canemitz@davidson.edu

**Abstract**

This report documents the program and the outcomes of Dagstuhl Seminar 23341 "Functionally Safe Multi-Core Systems". The seminar took place at a time when there is significant debate and disagreement in both academia and industry on how future safety-critical systems can be developed, certified, and deployed. This process is increasingly complex, as on the one hand, modern systems must provide more services while, on the other hand, analysing such systems becomes significantly more challenging in a multi-core scenario than it was in the single-core era. It is therefore a vital question to determine how the same level of certainty can be provided in the future.

The seminar brought together experts from academia and industry for the three major layers involved in safety-critical systems: application, middleware, and platform. They discussed the different perspectives, which problems are deemed specifically important, and potential solutions. One main focus when organizing the seminar was to not only present the different positions but also to provide space for lengthy discussion and disagreement.

**Seminar** August 20–25, 2023 – https://www.dagstuhl.de/23341
**2012 ACM Subject Classification** Computer systems organization → Architectures; Computer systems organization → Embedded and cyber-physical systems; Computer systems organization → Real-time systems; Computer systems organization → Dependable and fault-tolerant systems and networks; Theory of computation → Models of computation
**Keywords and phrases** eda and micro-architectures, middleware, multi-core, safety-critical applications
**Digital Object Identifier** 10.4230/DagRep.13.8.71

## 1 Executive Summary

*Georg von der Brüggen*
*Ian Gray*



There is a significant problem on the horizon in the field of embedded and real-time systems. The traditional approach for certifying high-integrity systems is no longer possible due to the increased complexity of modern applications, and the hardware platforms on which they execute are becoming heterogeneous multi-core systems on chip.

Considering how functional safety can be provided for multi-core systems, this Dagstuhl Seminar aimed to bring together experts from both academic and industry as well as participants from the three relevant layers, namely, application, middleware, and platform. The goal was to look at these topics from the individual perspective and to inspire interesting and fruitful discussion.

[*] Editor / Organizer
[†] Editorial Assistant / Collector

## Motivation

High-integrity domains such as automotive and avionics represent an important success story for academia. There is a long and storied history of knowledge transfer from academia into these domains, taking theory, approaches, and tooling and using it to build and certify complex systems in which safety is paramount.

This seminar is organised at a time when the domain is in an unprecedented moment of flux. Application complexity due to increasing consumer demand has skyrocketed over the last decade, and so to meet these demands, hardware manufacturers have created increasingly complex hardware platforms. Where previously software would be used for basic control, engine management, and rudimentary driver assist systems, increasing amounts of automation and high-fidelity entertainment are being deployed. Current estimates suggest a high-end vehicle currently runs over one hundred million lines of code, and that number will only increase. A corresponding increase in complexity has also been observed in the most traditionally conservative domains, such as avionics and space.

The traditional approach to such systems would use simple processing devices upon which small sets of tasks could be allocated. Existing certification of real-time systems could work well in these domains, using CPU models and scheduling theory in order to determine the worst-case response times of all tasks running on each device. The use of timing-aware interconnects, such as the CAN bus, could allow system integrators to ensure that safety requirements could be discharged at the top level.

Unfortunately, the theory has not kept pace with reality. The next generation of safety-critical systems will undoubtedly use multicore processors because there is an increasing need for performance and the availability of single-core processors is reducing. It simply is no longer possible to deliver the kinds of applications that are being demanded by consumers on simple, single-core devices over which we have a high degree of certainty.

The currently available techniques for analysing the timing of such multicore devices are limited. Equally, certification authorities are only just beginning to catch up with this new reality meaning that the certification needs of systems using multi-core are not clear. Recently, the civil aviation industry has produced some guidance in the form of CAST-32A, however, this is raising more questions than it is answering.

Safety-critical systems need strong guarantees of their timing behaviors which includes evidence of when the timing requirements are met, and then evidence for the loss of availability of certain functions in the other cases. It is crucially important that both the timing requirement and loss of service are commensurate with the system safety function that comes from the application. The challenge in providing such evidence comes from the platform's shared resources, e.g., caches and buses. With the introduction of multicore, this has become more complex due to reduced predictability. The unpredictability can be managed through the middleware where the resource management exists, however with appropriate consideration across the three layers during their design and subsequent composition.

This Dagstuhl Seminar aimed to bring together practitioners from three disciplines which represent the three layers (application, middleware, and platform) relevant to safety-critical systems that use multicore to understand how the safety of a system using multicores may be argued; the achievable evidence that can be produced; and how said systems might then be developed.

## Program and Structure

This seminar stands at the inflection point for high-integrity systems. There is significant debate and disagreement in both industry and academia about the manner in which future development should progress. It is unknown to what extent it is possible to maintain the same hard real-time guarantees that we have been used to in the past. Modern systems and modern devices simply do not provide the same level of certainty that we have relied upon. Simultaneously, systems are being asked to do even more things for which safety and certification are arguably more important than before.

This seminar was therefore structured to attempt to capture and advance this discussion. The main goal was to put academic leaders and industrial practitioners together in the same room, and to provide space for discussions and disagreements. The overarching goal was to agree as a room on what we believe the most important open questions were, and how research can be best structured over the coming decade to support the growing needs of the industry. Reciprocally, we hoped to capture what industry needs to provide academia, so that its needs can be met.

The first day of the seminar was arranged in advance, and then all subsequent sessions were arranged based on the discussions and the results that were obtained on the previous day. Sessions ran for 75 minutes with two sessions in the morning and two in the afternoon. These sessions usually started with a short talk or a controversial statement followed by extensive discussion. Therefore, we report a brief summary of the discussion in the sections instead of a summary of invited talks. When two sections discussed the same topic they are summarized together. One topic that was of specific interest to the participants from academia was the industry perspective; thus, multiple sessions had an industry focus.

## Takeaways

Most takeaways from this seminar came from the open-ended structure and heavy focus on collaboration. A key outcome was the codification of a range of the open research challenges in this area and the way that we might as a community work towards them – these are detailed in Section 14. Other important points that were highlighted through the discussions were:

- Just because a problem is academically interesting to solve, doesn't mean that it is necessarily a key research question. Instead of focussing on specific application domains, it is important to remember the value in more generalised system models and approaches.
- A system designer doesn't specifically care about WCRT, they care about the argumentation chain that will help them discharge their safety requirements. This often involves priorities, deadlines, criticality, and timing analysis, but they are means to an end.
- Industry is very eager to use academic results, but certification requirements mean that they are often unable to do so without mature tool support. Finding a way to solve this problem will massively increase the impact of research.
- The entire certification process is under-served by both academia and industry. Academics have little visibility into the system, and industry is incentivised to remain insular in its approach to the problem. There is a possibility for opening this process up through collaboration and public funding.

- Hardware vendors are unclear on what is actually required by the application layer and the theory. Important future work will attempt to answer just how much predictability we actually need in any given circumstance. This is important because current high-end hardware displays fundamentally unpredictable timing from the perspective of the end user, and so exact timing models cannot exist.
- There is a need to develop more architectural description technologies that can provide more appropriate guarantees that can support timing analysis without being unnecessarily detailed or overly specific. This would involve the standardisation of performance counters which are currently ad hoc.

Feedback from the seminar shows that this intent was captured well. Participant feedback said that "this seminar has had one of the most successful industry-academic collaborations I've experienced in a Dagstuhl Seminar" and others praised the networking opportunities and the "engaging environment for our industrial participants" with "eye-opening panels and discussions". The main weakness of this approach was that the less formalised structure was noted by participants, indicating that a possibility for greater balance exists between discussion and networking opportunities and more formalised talks.

## 2 Table of Contents

## 3 Introductory Sessions

The purpose of the introductory sessions was to explain the way in which the seminar was being arranged and focus on the idea that the participants can decide what the outcomes should be, and the sessions that therefore should be scheduled in order to make that happen. The introduction was led by Ian Gray from the University of York. This session explained the background to the original Dagstuhl proposal, which is that critical systems based on multicore processes are an inevitability. Very few OEMs design and build their own processors, and there are very few single-core processors that can be purchased with a supply guarantee for over 20 years – the sort of guarantee that would be required in many industrial systems. Therefore, these new multicourse systems must be used, and must be made safe.

The seminar participants were chosen from three main areas:

- Industry
- Academic
- Junior researchers

Unlike many similar seminars in which the audience is academic with industrial representation, we aimed for a much more even split. Problems and difficulties in this area are encountered from both the theoretical and from the practical side, and so input from both areas was vital. Equally, we ensured that new researchers were also represented to capture a wide range of academic thought. The participants were selected across the Americas, Europe, and Asia, and we encouraged leadership and representatives who identify as genders other than male.

The introductory session contextualised some of the work that has already been done by the certification authorities toward functional safety in multicore architectures. Particularly CAST-32A, which became AMC 20-193, that codifies:

- Measurement-based approaches to timing analysis can only deliver approximate knowledge of a system's timing properties.
- Knowledge has uncertainties (known and unknown).
- There may be extreme behaviour that has not been seen.
- Static analysis approaches make an assumption that valid knowledge is available for all hardware and software, and that there are no anomalies.

This means any design and assurance may need to be stochastic in nature allowing for questionable confidence measures and significant uncertainties. Indeed, the balance between stochastic assurance and proof-based assurance remained an important topic of discussion throughout the entire seminar.

The introductory session used live audience polling technologies to determine the feeling of the room and to start to collect information that would be used to design upcoming sessions. The question was asked "To what extent do we think exact analysis for these systems is still possible?". As can be seen from the results in Figure 1, a wide range of views were held by the participants. This ranged from a complete rejection of the feasibility of exact analysis, through some level of stochastic analysis, to a few believers that exact analysis will always be mandatory.

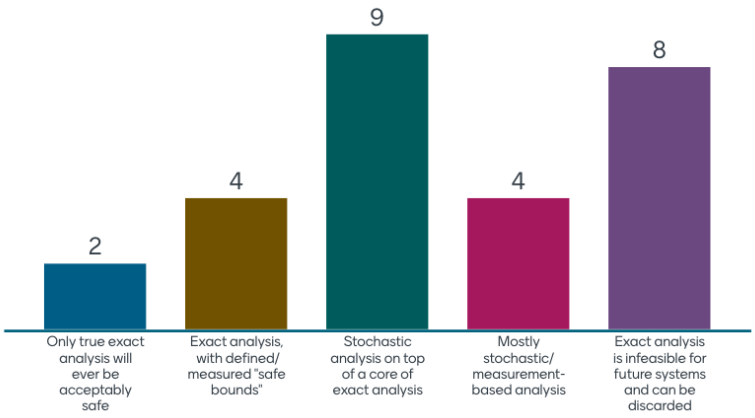


**Figure 1** To what extent do we think exact analysis for these systems is still possible?

## 4 Application, Middleware, and Platform

The second section of the day attempted to break down the singular problem of functionally safe multicore systems into a range of levels at which the issues can be defined and solved. The organisers deemed that these levels were "application", "middleware", and "platform", and so, this section sought to define these layers and solicit beliefs about the biggest challenges at each level.

### 4.1 Platform

The platform layer encompasses all of the issues that come from the hardware upon which the system software is running. This layer asks a range of questions such as:

- How should the micro-architecture provide properties for timing abstractions and safety assurance?
- How can hardware components be configured to support functional safety?
- How should timing predictability be examined?

The audience was asked to come together online and in-person to determine the issues at this level. The following main points were identified:

- Features enabling performance often impact the timing behaviour. 'Fast' is often hard to analyse.
- Getting the appropriate level of information to perform analysis. What is the resolution that is necessary by the real-time community and how do you know you've identified the right model?
- The composability of analysis to allow larger systems to be analysed from smaller components.
- Machine learning and how it interacts with timing.
- Reliable detection of errors and how they should be accommodated.
- Partitioning of hardware, both spatially and temporally.
- Corner cases, unexpected events, unlikely events, and unknown unknowns – all of which are more likely as systems get more complex.
- How to create usable timing models from low-level architectural models, i.e. VHDL.

## 4.2 Middleware

The middleware layer encompasses the fact that systems are being developed with off-the-shelf middleware components alongside bespoke in-house development. This can lead to difficult interactions between code that the developer controls and code that they do not. Equally modern middleware has to manage the shared resources of multicore systems, and ensure that access to shared resources can be carefully timed and budgeted. Low-level shared resources and failures can break otherwise well-designed abstractions. This layer is defined by issues of failure propagation, isolation, resource management, robustness, and uncertainty.

The following main issues were identified by the participants:

- Strategies for segregation and isolation so that the limits of failure propagation are understood.
- Runtime monitoring and runtime management.
- How many different middlewares are needed? How can they be combined?
- Bridging the gap between theory and implementation.
- Security, and how it interacts with other issues of system safety.
- Resource management, composition, and run-time monitoring and control.

## 4.3 Application

The application needs to build on top of the previous layers, collecting all the uncertainties while perhaps adding its own. At this layer, a key problem is that abstractions tend to 'leak'. Lower levels of the system may claim a certain amount of performance or a certain amount of isolation, but it is quite common that at the application level these assumptions can be tested and broken. It is necessary for the application level to understand what it can, and cannot rely upon, and what may cause the system to behave in an unexpected way.

Participants deemed the following issues important at this level:

- To determine an acceptable contribution for timing in the mitigation of hazardous events.
- How to present application and platform co-design throughout certification/stages of involvement.
- Developers should write the least code possible. Can a model-based approach help here?
- How to formally express application requirements?
- Can a metalanguage be developed that captures and expresses the application requirements, and can be used to communicate (and configure) lower layers (middleware/platform)?
- Is it always possible to abstract hardware timing issues (e.g. interference) from the application point of view?
- How do you design systems that are resilient to faults, whether they be logical, temporal, or security-related, while still maintaining predictability?
- What tool support can be provided in order to ensure the developers can avoid common problems, can express application requirements in a consistent manner, and can implement theories and knowledge from lower levels?

## 4.4 Desired Outcomes

Finally, this section concluded by collecting together the outcomes that participants intended to create from this seminar. Lots of different possibilities were discussed, but chief among them was a desire to establish the key research challenges that could help to drive academic

research for the next 10 years. Through this, a route map towards the effective achievement of assurance arguments for modern multi-core systems could be detailed, and therefore, the areas of theory that are not yet sufficiently developed become clear. In particular, the academic participants saw great value in hearing the views of practitioners about the real-world issues that they encounter and how they are beginning to tackle the problem of multicore. Accordingly, the organisers began arranging sessions around this with a view to culminating on the final day with a collection of research challenges.

## 5 Properties Needed and Functionalities Provided to Ensure Functional Safety and its Verification

After the previous session focused on the perspectives of applications, middleware, and platforms, the two final sessions on Monday started by collecting the thoughts of the participants on the properties needed and the functionalities that must be provided to allow verification.

Observability was determined as a key factor – otherwise, a system may be functionally safe but the safety cannot actually be argued and verified. However, it is not clear how observability can be achieved in practice, especially when abstractions need to cross layers. As a result, model composability is challenging, and if not appropriately scoped, impossible.

It was also mentioned that partitioning on the hardware level may be better than software partitioning due to potential overhead and granularity. However, while this may be an approach for static applications, it may not work well if applications have more dynamic needs. It was also pointed out that partitioning is not always possible – for instance, parallelization is often necessary but avoiding race conditions is non-trivial.

Research on multi-core systems will rely on results in other areas. However, these results may either not be known or assumptions and details may be unclear for people who are not experts in the field. For instance, it was (to the participants) unclear what the status of research into stochastic modeling of network traffic is. In this context, it has been pointed out that incorrect analyses had previously been used in practice, but that the overall over-approximations were still sufficiently pessimistic to prevent errors from occurring in practice. This was seen as an argument for a composable analysis compared to a holistic analysis.

An alternative approach could be to flip the problem – by imposing constraints on software development and preventing worst-case overlaps. For instance, some real-time operating systems do not impose constraints while others require applications to follow a given partition scheme.

This approach could play out in a way that industry is given a list of constraints to validate (without knowing the underlying analysis), which could result in systems that are easier to certify. In addition, tooling to show if meeting these constraints could be provided.

Furthermore, the following issues were selected as key by participants:

- Is there any hope of treating the path as deterministic (e.g., when looking at main memory access due to cache misses)?
- Literature of real-time analysis and WCET analysis is established – but some busy window assumptions do not compose and have been the source of challenges.

For the second session on the topic, the participants were split into 4 groups, each discussing and reporting their thoughts and ideas on what could be necessary to have functionally safe multi-cores and which other major problems still exist.

- Data sheets are sometimes inconsistent and incorrect which makes it hard to determine the "correct" behaviour and to determine where the uncertainties are. Indeed, the lack of accuracy from official documentation would return as an important issue many times over the course of the seminar.
- Related to this point, how do we verify correctness when we utilize functions, algorithms, and libraries? For instance, even seemingly simple things like the implementation of the EDF scheduler in some RTOSes were shown to be incorrect.
- Functional safety may mean different things on different levels. On a general level, a system may be considered safe if there is no harm to humans. Yet, on the system level, safety may mean, for instance, avoiding race conditions or fulfilling timing requirements. It must be made clear which definition or level is considered.
- Is functional safety the same as being certifiable?
- What is the difference between the properties we care about in safety-critical real-time systems and *functionally safe*? When can we say we have a sufficient set to ensure safety and how can we guarantee nothing is missing?
- Safety can be argued either by design or by verification. Is it easier to achieve when coming from one direction – and how can these approaches possibly be combined?
- There is a shift from static to dynamic analysis.
- Will hardware at some point reach sufficient performance with full predictability? Or, will this never be the case meaning that we should therefore focus on resilience and safe degradation?
- Is it always necessary to verify that all functional requirements are satisfied? Or, should we determine how much violation of properties can be tolerated or mitigated (e.g., by looking at fault tolerance approaches)?

# 6 System Certification

A topic that participants wanted to cover was that of system certification, the process by which high-integrity and safety-critical systems are certified for use by authorities. Academics are rarely involved in the complicated and time-consuming certification process. Yet, if academic results in the embedded system and real-time domain should be utilized in real-world products, they must usually be certified by the relevant authorities. Therefore, knowledge about the certification process itself, how these results can be transferred, and which assumptions, constraints, and techniques are seen as reasonable (or even at all achievable) by industry is highly relevant for academic research. In these two sessions, we discussed the certification process based, on the one hand, on the industry's perspective of Victor Jegu from Airbus and, on the other hand, on the perspective of a collaboration between academia and industry, given by Claire Pagetti from ONERA.

## 6.1 Industry's Perspective

Airplanes are very complex systems and also have strong certification requirements. As a result, they have a long cycle of development and certification; thus, moving from single-core to multi-core is also a lengthy process, which is just starting. The main driver for this move is not the need for more computation power – to ease certification this need could more comfortably be fulfilled by using multiple single-core CPU boards. However, airplanes must be maintained for decades and the availability of single-core CPUs is becoming an issue.

This lack of availability of other options while multi-core systems are highly available is the main driver for the transition to multi-core systems. One main focus of discussion was how the transfer from (interconnected) single-core systems to (interconnected) multi-core systems can be achieved. From Victor Jegu's perspective, key characteristics of the before-multi-core systems and ideal multi-core systems are as follows:

- Before multi-core:
  - one or more standalone cores with private resources
  - few COTS/IP components
  - ideally in-house hardware development
  - minimal shared resources, static WCET analysis possible
- Ideal multi-core:
  - distributed cores and resources as well as isolated interconnects – to minimize the shared usage to usage strictly needed for communication
  - cores may run in clusters for distributed applications (or when the workload is not real-time critical)
  - domains with large/intensive memory needs should also have dedicated memory controllers

To enable certification, systems should be completely deterministic in the data flow. Thus, tasks should be periodic, and scheduling decisions should be as static as possible. Yet, current multi-core systems are far from this, as there is minimal isolation of interconnects. As a result, there are multiple, often unpredictable, interference channels:

- sharing cache
- sharing buses and interconnects
- sharing targets (like SRAM and DRAM)

The problem of providing a certifiable system gets more complex since not all dependencies visible are in a processor's documentation. Furthermore, software constraints (like semaphores, locks, and rendezvous) have to be considered.

Solutions for these problems may involve:

- more formalism, e.g., by using languages like Rust for development
- lock-free architectures
- inter-core communication only through non-blocking interfaces
- static resource allocation by configuration tables

Restricting critical applications to one core is often not possible, since some resources are shared by the platform by force:

- memory controller
- interconnects
- secondary buses
- interrupt controller

## 6.2 Collaboration Perspective

Claire Pagetti is a contributor of the PHYLOG [1] project and its follow-up PHYLOG2, with both projects supporting aeronautic industry partners (e.g., Airbus). While PHYLOG focused on defining a certification strategy for multi-core architectures with respect to CAST 32A (now part of the AMC 20-193), PHYLOG2 considers the certification of hybrid architectures – which are multi-core architectures incorporating accelerators (e.g., GPU or FPGA).

The aeronautical certification process considers airworthiness regulations and provides means of compliance summarised in the AC (FAA Advisory Circulars) / AMC (EASA Applicable Means of Compliance) for applicants to define their certification strategy. The A(M)C themselves point to more detailed ARP/DO standards, each detailing objectives and activities to be fulfilled in order to demonstrate the compliance of a final product to regulations. An applicant can alternatively define their own path to compliance, called CRI (Certification Review Item), as long as they have an agreement with the certification authorities. Thus, certification can be seen as an argumentation to convince the certification authorities that compliance with regulations has been reached by validating and/or verifying that all requirements are met.

In particular, for designing an embedded multi-core-based system, which is the scope of PHYLOG, the applicable requirements are those specified by the AMC 20-152A, AMC 20-193, ED 79/ARP 4754, ARP 4761, ED 20x, ED 80/DO 254, and ED-12C/DO 178. Tools used during the design and implementation may have to be qualified as well, particularly if they generate embedded code. PHYLOG outcome is a model-based certification framework based on:

- Assurance cases to organize arguments through structured notations or diagrams. Most of the CAST 32A objectives as generic assurance cases have been translated to be instantiated for each specific system;
- A model-based approach as a replacement or complement to the usual text-based documentation. Therefore, a very generic modeling language, PML (PHYLOG meta-model), has been defined that allows representation of the minimal components that contribute to three types of analyses (interference, capacity, and safety) required by the CAST 32A.
- Formal and automatic analyses to support the certification argumentation. Amongst them, the identification of interference and the characterization of their temporal effect is of great importance. A Monosat + Scala tool that automatically computes the interference from a PML model has been deployed.
- A stressing benchmark strategy is associated with the modeling activity to validate the PML model and characterize the effect of interference.

The purpose of PHYLOG2 is to delve deeper into modeling hybrid architectures.

### References

1 PHYLOG project webpage – `https://w3.onera.fr/phylog`

## 7 Industry Panel

The industry panel was organized as a Q & A session where the audience could ask questions which then were answered by Matteo Andreozzi from ARM, Jan Micha Borrmann from Bosch, Victor Jegu from Airbus, Kevin Quinn from General Dynamics, and Zoë Stephenson from Rapita Systems. The panel was moderated by Mitra Nasri from TU Eindhoven. We paraphrase the questions and summarize the provided answers.

- **Q1:** Do you look at academic papers, e.g., when you have a problem at hand?
  - Industry is more about patents than papers. Unfortunately, already published results can usually not be used in patents.
  - It would be easier if there was a shared wiki with people's research domains, considered models, etc.

- **Q2:** What academic assumptions are (un-)reasonable?
  - Academic papers often focus on meeting timing deadlines exactly, but safety-critical systems have to deal with faults. Hence, they are designed to be deterministic – when a deadline is missed, we know what will happen.
  - Shut down is usually not an option when a failure occurs – systems need to be fail-operative.
  - Mixed-criticality for us is about partitioning the electronics.
  - We don't care specifically about WCET, just that we meet safety requirements and end-to-end response time constraints. WCETs are just the best model we have.
  - Data freshness is important – one should annotate data with timestamps.
  - Focus is more on making correct decisions, not on periodic behavior of polling – it is less helpful to keep polling stale data.
  - For data freshness, the DAG model can be useful. Yet, industry cares about data flow graphs, academics are asking more about control flow graphs.
- **Q3:** What steps are taken to help increase timing predictability? (Question 19 from a recent survey [1] on industry practice in real-time systems. So, the panelists discussed the answers provided in the survey.)
  - Partitioning caches – should be expanded to more, just general "partitioning shared resources".
  - If disabling cache, maybe buy a more predictable machine instead.
  - A partitioning argument must be clearly communicable for certification.
  - A bandwidth servers type thing for highest assurance/integrity parts, can simplify RM analysis, one task every 20ms.
  - Regarding cache replacement policies, LRU is well explored in real-time literature and used by some platforms. Yet, ARM (and other) commercial platforms use a proprietary cache replacement mechanism – details on the mechanism cannot be given but some predictability guarantees can be provided.
  - Tools are used for WCET computation. It would be helpful to have an analysis showing the worst-case cache misses as this could reduce the testing needs.
  - One of the biggest needs is observability.
- **Q4:** Hardware performance counter standardization?
  - PMUs are not designed for dynamic monitoring.
  - Advanced OSes look at splitting up data gathering from monitoring.
- **Q5:** What do you see as a roadmap for the future, what questions should be answered, and what are future trends?
  - One very important thing when moving real-time support is observability.
  - Can we get to a useful workload modeling abstraction that enables sharing needed info without needing to disclose IP?
  - SHIM: software-hardware interface model.
  - Demos are beneficial as they can show the vision of others.
  - What could we do instead of WCET analysis? Is there another way to meet the safety requirements? Especially when considering the out-of-order execution of "modern" architecture. How does out-of-order execution impact jitter?
  - Partitioning can be a means to manage cost. Yet, it is not 100% reliable – could we get some bounds around that unreliability?
  - Software development takes a long time, at a huge cost and drivers can be as large as the primary code base.
  - Hardware reuse is expected, software needs a huge recertification effort.

**References**

1 Benny Akesson, Mitra Nasri, Geoffrey Nelissen, Sebastian Altmeyer, and Robert I. Davis, *A comprehensive survey of industry practice in real-time systems*, Real-Time Systems 58, 2021.

## 8 An "Ask" from Static Analysis

After focussing on the industrial perspective in the previous sessions, this session primarily considered potential analysis directions from academia. Rodolfo Pellizzoni from the University of Waterloo provided a general overview from the perspective of static analysis and Sanjoy Baruah from the Washington University in St. Louis moderated the discussion.

A multi-core resource interference analysis is often considering different parts of the system individually. Multiple cores (and other components) thus result in multiple analyses which a performed in a divide-and-conquer approach. For each core, a static analysis and a resource analysis are performed.

The conceptual resource model may be as follows:

- Takes some flow as input (target) and output (initiator) some flows.
- These are the requests (which behave differently for different types).
- Latency depends on the flow under analysis and the interfering ones.
- The process could be (conceptually) iterated to obtain end-to-end latency.

A basic, possibly provable, analysis could be structured as follows:

- Assuming that a model of the resource is available (which is not always the case).
- For each resource and each request, based on the type, an access pattern that causes the worst-case latency is determined.
- Sum the latency over all requests.
- For each program under analysis, sum over resources and add the total latency.

This approach could be safe (with caveats) but would be extremely pessimistic. At least the following sources of pessimism must be considered.

1. Request Distribution
   - There is often no information on request distribution. Yet, these distributions may potentially be obtained by program analysis or by observation. If pathological cases are triggered they may result in a 100x increased latency. Many hardware structures that are designed to not saturate under a "normal" operation flow stall once such a case happens.
   - A key issue is to capture the burstiness over time. Potential modeling approaches:
     a. Determining the number of requests over a given time window – for both the flow under analysis and for interfering ones. Problems with this approach include that it is imprecise, it is unclear how to cover bursts correctly, that large time windows must be considered, and how windows are synchronized.
     b. Request curves as in, for instance, network calculus. These curves separate burst and rate and can be deterministic or stochastic. However, determining arrival curves for network calculus is non-trivial and it must be guaranteed that the backlog is smaller than the buffer size to avoid "backpressure".
2. Latency Sensitivity
   - General idea: sum latency of all requests to the WCET.
   - Yet, not all requests necessarily add latency to the core (e.g., reads can be prefetched).

   - Out of Order cores can reorder operations and tolerate some load latency. This can potentially be accounted for by computing a tolerance term per request if alignment issues can be solved.
3. Missing Pipeline Effects
   - One interfering request may add delay on multiple resources.

Additional problems often arise since a precise resource model is either not available or too complex to be analyzed in practice. One potential solution, often used in academia, is to use a simpler, approximated model where the input-output characterization ensures that the output curve is always "safe". Yet, this property can likely only hold of system configured through partitioning (best in hardware).

Unfortunately, static analysis may not be the solution for the multi-core era, as finding a correct abstract interpretation of a modern superscalar out-of-order core may be (nearly) impossible. Instead, one may try other directions by examining the following questions.

- Is some kind of hybrid analysis possible?
- Could added observability help strengthen the case?
- Can we reach hard guarantees in soft systems?
- What dependencies do we need to look at?
- If we do not need worst-case, what do we need? Can we rely on statistical arguments?
- Could we look at a set of possible WCET values?

## 9 A Short Tour of Mostly Relevant Parts of DO-178C

As noted earlier in the report, undergoing the lengthy certification process is not something that academics generally take part in. Participants invited Zoö Stevenson from Rapita Systems to give an overview of her experience in supporting certification through DO-178C. It is the main certification document that describes the approval process for commercial aerospace systems. It is used by a range of countries including the US, Canada, and the EU and conformance to DO-178C is a key requirement for many tool vendors. Rapita Systemes provides both software tooling and analysis of user software, with the aim of certifying complex systems to this level.

A key point to highlight is that there is often a disconnection between software-level concerns and certification-level concerns. A feature of key importance to the software developer, may not necessarily be linked to system component of correspondingly high integrity. Software level does not imply failure level, so it is not possible to directly use software reliability as part of the certification argument.

A common misconception about certification documents, such as DO-178C, is that they give a list of system requirements which must be fulfilled and a checklist of tests to be performed. Rather, requirements come from system design, and so certification is concerned with the overall system and the evidence that is necessary to discharge safety requirements. For example, if the developer is using Ada, which contains a range of exception-handling systems, DO-178C will still require such systems are tested, and their efficacy demonstrated, if such systems are linked to the discharging of a safety requirement.

The applicant is responsible for oversight of all of its suppliers. This has wide-ranging implications for the way the software is developed integrated and analysed. The system must be analysed as a whole, with an analysis performed on the integrated software stack. All tool vendors and software vendors in this domain work to support this process.

The resulting certification argument is a combination of reviews, analyses, and tests. Significant training for employees is needed to support all stages of this process. A Software Quality Assurance process (SQA) will often form part of this evidence, through compliance to a relevant standard such as AS9100.

The final observation was on the semantics of certification documents. The previous version of 178C, 178B, was often used by vendors as prescriptive. If the document didn't mention something then it was presumed that you could not use it. This had wide-ranging implications for the range of development techniques, such as object-orientation, and of hardware platforms, such as FPGAs. More recent certification documents do not take this stance, but progress in these areas remains slow.

In the following Q&A, participants noted that it would be useful as a community to have an entire certification packet as an example. This is difficult because as has been noted, certification is a very lengthy and incredibly expensive process and necessarily reveals large amount of commercially sensitive information. Accordingly, it would have to be undertaken as part of a publicly funded project and while there are some burgeoning projects in this area, nothing has as of yet been completed.

## 10 Programming Languages and (Machine-Checked) Models: What to do About Multi-Core Platforms?

Timothy Bourke from INRIA & ENS in Paris discussed programming languages and (machine-checked) models with a focus on the additional challenges that arise when examining the timing behaviour in a multi-core environment. The process has a wide set of potential requirements and strategies. Hence, there is a tradeoff between opportunities and risks, for instance, for the exploitation of hardware, concurrency and related issues (like race conditions), or when considering inter-core interference sources.

The verification process results in a large argumentation tree, where the correct branch needs to be selected based on the component and the confidence. The interference analysis model can be built using PML. An appropriate documentation must be provided for the purpose of timing analysis. If one wants to do the same for schedulability, it must be determined what guarantees each test provides. In addition, to apply such an approach in multi-core systems, it might be necessary to utilize hardware designed for predictability or hardware that provides increased observability.

## 11 Combined HW/SW Formal Verification of Timing Behaviour

Mathieu Jan from CEA LIST talked about opportunities for combined hardware/software modeling for the formal verification of timing behaviour, using RISC-V as an example. He highlighted that it is important to identify timing anomalies – these do not necessarily need to be eliminated but they must be accounted for. This can be achieved by considering multiple traces. The 6-stage RISC-V rocket pipeline must be modeled (either manually or with an automatic tool). The TLA+ pipeline specifications are extended with instructions and control path.

## 12 How Much Predictable Computing Do We Need?

This session was lead by Jian-Jia Chen TU Dortmund University and aimed to answer the question, "how much predictable computing do we actually need?" A key theme that was emerging from the discussions was that it is possible for academics to solve problems that do not actually need to be solved. Fundamentally this comes from the fact that academics want to *solve* problems, and engineers want to *avoid* problems.

Consider the commonly used terminology to determine the real-time constraints of a given system component:

- Hard – the component should never violate timing.
- Firm – the component may violate timing, but this would never be seen at the application level, i.e. violations are handled.
- Soft – this component is allowed timing violations up to some stated level of assurance.

Academics tend to focus on 100% hard real-time systems, but real-world workloads are a mixture of different levels of assurance. This varies heavily from domain to domain; the needs of the automotive industry are very different from that of avionics or manufacturing, for example.

The application domains that should be considered most useful for focusing academic research were then discussed. There is currently a lot of interest in the field of autonomous driving and the interesting problems that it represents, but if pursued too much may result in too narrow a focus.

Instead of putting applications into 'buckets' and considering them in isolation, we should instead attempt to develop a range of system models that can be later matched to application classes. The reason for this is that application classes will change over time, and the speed of the industry means that this can sometimes be quite rapid. Cloud computing, big data, and autonomous driving are all examples of application classes that have appeared rapidly and driven a large amount of academic interest.

An example given was that of the Mixed Criticality System (MCS) model. Instead of focusing on a specific domain or application, MCS aims to be a way of building systems that can switch between different levels of assurance. It is a well-studied theory, and its generality means that it is starting to see use in a wide range of areas.

Our system models must recognise that industry is less focused on characterising workloads and more on the system-level requirements that they must fulfill. Timing correctness is an important tool for the verification of these requirements, but it is merely a *derived* requirement.

## 13 What Properties Do We Need From the Hardware/Middleware?

Chris Gill from Washington University in Saint Louis discussed the properties needed from hardware and middleware based on example projects he participated in. He emphasized that due to the large scope of embedded and cyber-physical systems applications have different needs and constraints; thus, application-specific semantics matter more than ever. For instance, sensor types, energy requirements, memory and cache sizes, the number of cores as well as the core speed, and the overhead to interconnect these cores may differ largely. As a result, general abstractions and generalizations are hard to find and it seems problematic to find a general approach for decomposing systems. Yet, it would be nice to have a list of constraints and construct adaptable scheduling to meet those.

Micha Borrmann from Bosch GmbH then discussed how these issues are addressed in real-world hardware development by enumerating some of the challenges in highly-available automotive computing. Automotive systems are moving from using a large network of low-power devices towards greater centralisation of computing resources in a few high-powered multicore platforms. This results in a large number of safety and security requirements that must be addressed. This necessitates a complete rethink in the way that automotive systems can be built, and is an active area of development.

## 14 Open Research Challenges

The seminar was concluded with an roundtable session in which the open research challenges as well as questions and thoughts were collected. The goal of this was to collect the thoughts of the participants as they had evolved over the course of the seminar, and to ask the questions would should drive the direction of research for the next decade. We collected top level observations, and the questions that stem from each of these.

- Maybe hard real-time is not the most accurate model.
  - Time is a means to some end – not the end goal.
  - Timing is also still an unsolved problem, and getting more complex
  - In real-time, when we specify tasks, we focus on time parameters. Should we also look at logical correctness? Are the correct values produced? Usually, we assume functional/logical correctness – this assumption may need to be challenged.
  - Is reliability a fundamental thing we must provide?
  - Trust and probability may be interesting metrics to consider.
  - Distinguish functional correctness and fault tolerance.
  - Non-functional properties are important: composability, extensibility, robustness, parametric simplicity.
- Applications are very complex, but platforms are amazingly complex.
  - We consider algorithms (scheduling algorithms, protocols), we need to make sure these are actually correctly implemented – separate concerns: analysis is correct, implementation is correct. For instance, PROSA vs. given implementation – does the implantation match the model?
  - When it comes to partitioning: how do we ensure two components do not interfere? This is a huge engineering effort, with hours of time per line of code.
  - Why should one use more complex architecture, requiring hugely complex analysis? One could consider building a simpler architecture.
  - Is the right way to look at uniprocessor techniques and determine how we can extend them to multicore? OR, should we do an entire redesign of analyses?
  - Safely-critical applications cannot simply just apply a patch, they need recertification.
- There is a value of consensus in the research community.
  - Dependability and fault-tolerance research communities study these and are working with some industries. Do we need to integrate? Can we rely on other work?
  - Do we analyze too complex systems? Engineers often repeatedly simplify problems.
  - It would be nice to know where we are losing. For example, why is a system using $X$ unable to pass certification?
  - Instead of asking if a model is good, try to use it.

- What is the set of requirements?
  - Specifications and documentation will often have flaws.
  - Does the OS meet specs – industry finds this is not always the case. Is a verified OS a solution?
  - There are assumptions even in abstractions, simulations for physical hardware – we cannot fully capture physics in the model.
  - Models need to be as simple as possible for real use vs. expressive, complex models.

Many discussions centred around the system models that we create. It was observed that there is a clear tradeoff between accuracy and model simplicity. Formal models for safety certification ideally should satisfy or include:

- criticality
- environment assumptions
- developer-friendly tooling
- predictions about (worst-case) system behavior
- features acceptable to the certification authority

However, the process of modeling can have violations throughout this process:

- model compliance
- assumption validity
- analysis soundness
- implementation correctness
- analysis accuracy
- explainability

There clearly is therefore an identified need to keep people talking. This seminar showed the unique value in something which is critically lacking in our discipline, the ability for practitioners and academic experts to get together in the same room and to pass ideas to each other. Models need refining which can support certification, be reflected by hardware, be used by developers, and be analysed for correctness by tooling. Only with a cross-cutting integrated approach from all levels of the development stack can this succeed.

## Participants


- Sebastian Altmeyer
Universität Augsburg, DE
- Tanya Amert
Carleton College – Northfield, US
- Matteo Andreozzi
Arm – Cambridge, GB
- Sanjoy Baruah
Washington University – St. Louis, US
- Jan Micha Borrmann
Robert Bosch GmbH – Stuttgart, DE
- Timothy Bourke
INRIA & ENS Paris, FR
- Björn B. Brandenburg
MPI-SWS – Kaiserslautern, DE
- Jian-Jia Chen
TU Dortmund, DE
- Christian Ferdinand
AbsInt – Saarbrücken, DE
- Julien Forget
University of Lille, FR
- Anna Friebe
Mälardalen University – Västeras, SE
- Chris Gill
Washington University – St. Louis, US
- Ian Gray
University of York, GB
- Arpan Gujarati
University of British Columbia – Vancouver, CA
- Robin Hapka
TU Braunschweig, DE
- Mathieu Jan
CEA LIST – Gif-sur-Yvette, FR
- Victor Jegu
Airbus S.A.S. – Toulouse, FR
- Eric Jenn
IRT Antoine de Saint Exupéry – Toulouse, FR
- Mitra Nasri
TU Eindhoven, NL
- Geoffrey Nelissen
TU Eindhoven, NL
- Catherine Nemitz
Davidson College, US
- Claire Pagetti
ONERA – Toulouse, FR
- Sri Parameswaran
University of Sydney, AU
- Rodolfo Pellizzoni
University of Waterloo, CA
- Kevin Quinn
General Dynamics – St Leonards on Sea, GB
- Jan Reineke
Universität des Saarlandes – Saarbrücken, DE
- Benjamin Rouxel
University of Modena, IT
- Selma Saidi
TU Dortmund, DE
- Matheus Schuh
Kalray – Montbonnot-Saint-Martin, FR
- Zoë Stephenson
Rapita Systems Ltd. – York, GB
- Jürgen Teich
Universität Erlangen-Nürnberg, DE
- Georg von der Brüggen
TU Dortmund, DE
- Bryan Ward
Vanderbilt University – Nashville, US
- Reinhard Wilhelm
Universität des Saarlandes – Saarbrücken, DE
- Houssam-Eddine Zahaf
University of Nantes, FR


Report from Dagstuhl Seminar 23342

# Computational Geometry of Earth System Analysis

**Susanne Crewell[*1], Anne Driemel[*2], Jeff M. Phillips[*3], and Dwaipayan Chatterjee[†4]**

1 **Universität Köln, DE.** susanne.crewell@uni-koeln.de
2 **Universität Bonn, DE.** driemel@cs.uni-bonn.de
3 **University of Utah – Salt Lake City, US.** jeffp@cs.utah.edu
4 **Universität Köln, DE.** dchatter@uni-koeln.de

## Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23342 "Computational Geometry of Earth System Analysis". This seminar brought together experts of algorithms and the Earth sciences to foster collaborations that can tackle algorithmic problems in the Earth system by the crossover of expertise in these different areas. The Earth sciences include a manifold of disciplines that deal with atmospheric, oceanic and terrestrial observations to further our understanding of climate processes. New generations of observation systems that are being developed right now provide novel data about the atmospheric and surface conditions at increasing spatial and temporal resolution. This provides unique information to improve weather and climate prediction but cannot always be handled by traditional numerical models. Computational Geometry is rooted in a strong tradition of algorithm and complexity analysis applied to practical geometric problems. Efficient algorithmic methods developed in this field are often tailored to the low-dimensional geometric settings that arise in a multitude of application areas, but have until recently not been applied to problems arising in the Earth system sciences – and in particular not in meteorology.

**Seminar** August 20–25, 2023 – https://www.dagstuhl.de/23342
**2012 ACM Subject Classification** Theory of computation → Computational geometry; Theory of computation → Data structures and algorithms for data management
**Keywords and phrases** Data reduction, Event detection, Feature tracking, Geometric algorithms, Interpolation methods, Sensor placement
**Digital Object Identifier** 10.4230/DagRep.13.8.91

## 1 Executive Summary

*Anne Driemel (Universität Bonn, DE)*
*Susanne Crewell (Universität Köln, DE)*
*Jeff M. Phillips (University of Utah – Salt Lake City, US)*



Various disciplines within the Earth sciences deal with measuring and representing the geometry of the Earth's land and sea surface, as well as atmospheric and oceanic conditions, to further our understanding of dynamic processes occurring on the Earth. All around the world society and economy is becoming more and more vulnerable to changing weather. The recent extreme precipitation and flooding in western Germany and Libya, the strong tornado in the Czech Republic or dry spells leading to severe fires in Canada are just few of

[*] Editor / Organizer
[†] Editorial Assistant / Collector


Computational Geometry of Earth System Analysis, *Dagstuhl Reports*, Vol. 13, Issue 8, pp. 91–105
Editors: Susanne Crewell, Anne Driemel, and Jeff M. Phillips
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

many examples which illustrate how anthropogenic climate change is going to influence our weather. Climate describes the statistics of all-weather events and is typically predicted using physical models representing the relevant processes in the Earth system. Understanding these processes is paramount to anticipating and addressing the challenges posed by climate change today. A key aspect of meteorological research and Earth sciences in general is to develop methods to turn atmospheric, oceanic and terrestrial observations into regional information for weather and climate prediction. The observations are being collected with different types of sensors at increasing spatial and temporal resolution which poses computational challenges to meteorologists that cannot always be addressed with traditional methods. More and more observation systems (e.g. from commercial aircraft or new satellite series including radio occultation) but also opportunistic crowd-sourced measurements are currently exploited. In addition, there are ground-based remote-sensing networks for operational atmospheric profiling, which can shed light on small-scale processes such as turbulence and cloud physics that cannot be resolved in detail from satellites, but can be used to improve model parameterizations. Thus, on the horizon there is a wealth of new, voluminous observation systems providing unprecedented possibilities for improving weather and climate models, but require innovative and explorative approaches in the areas of data handling, data assessment, information extraction, and data assimilation. The field of computational geometry is concerned with the design, analysis, and implementation of efficient algorithms for geometric and topological problems, which arise naturally in a wide range of application areas. Computational geometry is a vibrant and mature field of research, with several dedicated international conferences and journals and strong intellectual connections with other computing and mathematics disciplines. Within computer science and mathematics, computational geometry lies in the intersection of the theory of algorithms and combinatorial geometry. Despite its theoretical nature, the research in this field is strongly oriented towards and motivated by concrete practical problems that arise in various application areas that deal with geometric data. The related emerging field of geometric data analysis deals with the efficient statistical analysis of geometric data by providing sketches and data summaries with provable guarantees.

## Outcome

This Dagstuhl Seminar brought together 26 researchers from the fields computational geometry and the Earth sciences to provide a forum to discuss the unique computational challenges that need to be dealt with and how the geometry underlying the input data can be exploited to obtain efficient algorithms. The 5-day seminar was initially focused on three problem areas (1) data assimilation of weather-related measurements for numerical simulation, (2) tracking and clustering of moving atmospheric features, and (3) the planning and optimization of sensor placements. In addition to these topics, seminar participants contributed research questions from the current state of the art in their fields. There were 13 research talks throughout the seminar. Several longer talks gave an overview over the proposed seminar topics, either from the point of view of Earth scientists or from the point of view of algorithmic techniques. Organized group discussions resulted in several breakout groups on emerging topics. Each group was composed of a mix of participants from the different fields. In addition, the program left ample room for discussions and for new research directions to emerge.

The seminar was seen as a success by organizers and participants. This is nicely summarized by a quote from the survey:

"This seminar brought together a highly motivated group of excellent researchers. It felt as if all of them wanted to make the best use of this seminar to make a difference, in an interdisciplinary application domain. Since this was an interdisciplinary seminar, there was certainly a natural spread in terms of how willing people were to learn about new topics and/or to (re-)explain their work/ideas to researchers from other fields. However, overall this worked well and the atmosphere stimulated new ideas and interdisciplinary exchange."

## 2 Table of Contents

# 3 Overview of Talks

## 3.1 Partitioning a Polygon Into Small Pieces

*Mikkel Abrahamsen (University of Copenhagen, DK)*


**Joint work of** Mikkel Abrahamsen, Nichlas Langhoff Rasmussen
**Main reference** Mikkel Abrahamsen, Nichlas Langhoff Rasmussen: "Partitioning a Polygon Into Small Pieces", CoRR, Vol. abs/2211.01359, 2022.
**URL** https://doi.org//10.48550/ARXIV.2211.01359

We study the problem of partitioning a given simple polygon $P$ into a minimum number of connected polygonal pieces, each of bounded size. We describe a general technique for constructing such partitions which works for several notions of 'bounded size,' namely that each piece must be contained in a unit square or unit disk, or that each piece has bounded perimeter, straight-line diameter or geodesic diameter. The problems are motivated by practical settings in manufacturing, finite element analysis, collision detection, vehicle routing, shipping, laser capture microdissection and perhaps earth system analysis.

It seems out of reach to compute optimal partitions; even in extremely restricted cases such as when $P$ is a square. Our main result is to develop constant-factor approximation algorithms, which means that the number of pieces in the produced partition is at most a constant factor larger than the cardinality of an optimal partition. Existing algorithms [Damian and Pemmaraju, 2004] do not allow Steiner points, which means that all corners of the produced pieces must also be corners of $P$. This has the disappointing consequence that a partition does often not exist, whereas our algorithms always produce meaningful partitions. Furthermore, an optimal partition without Steiner points may require $\Omega(n)$ pieces for polygons with n corners where a partition consisting of just 2 pieces exists when Steiner points are allowed. Other existing algorithms [Arkin, Das, Gao, Goswami, Mitchell, Polishchuk and Tóth, 2020] only allows $P$ to be split along chords, whereas we make no constraints on the boundaries of the pieces.

## 3.2 Flood Risk Analysis on Terrains

*Pankaj Kumar Agarwal (Duke University – Durham, US)*



An important problem in terrain analysis is modeling how water flows across a terrain and creates floods by filling up depressions. This talk discusses a number of flood-risk related problems: Given a terrain T, represented as a triangulated xy-monotone surface, a rain distribution R, and a volume of rain V, determine which portions of T are flooded and how water flows across T. Efficient algorithms are presented for flood-risk analysis under both single-flow-direction (SFD) as well as multi-flow-directions (MFD) models – in the former, water at a point can flow along one downward slope edge while in the latter, it can flow along multiple downward slope edges; the latter more accurately represent flooding events but it is computational more challenging.

## 3.3 Algorithms for Movement Analysis

*Maike Buchin (Ruhr-Universität Bochum, DE)*



Nowadays a lot of movement data of animals, people and weather phenomena are being collected. To analyse this data requires efficient geometric algorithms. Some fundamental tasks such as clustering, segmentation, and simplification occur often in this. I will discuss three of these problems: similarity, clustering, and grouping. Each of these, I will motivate by movement analysis, sketch the algorithmic challenges of solving these, and present some experiments of our algorithms on concrete movement data sets.

## 3.4 Earth System Modelling

*Peter Dueben (ECMWF – Bonn, DE)*



I will provide an overview on Earth system modelling discussing the ten most relevant questions from Earth system modelling for Computational geometry. This includes what the Earth system is, how it is modelled, what good Earth system models are, how good they are, how uncertainty is represented, what geometry the models use, how they use high performance computing, and what machine learning will change in Earth system modelling in the future.

## 3.5 Between Drought and Heavy Rain: Digital Methods for Extreme Water Events

*Sándor Fekete (TU Braunschweig, DE)*


URL https://www.exdimum.org

In the course of global climate change, not only the averages of precipitation are shifting, but also the extremes: Drought and local heavy rainfall events have become more pronounced. This development leads to challenges that occur simultaneously on different scales. Extreme drought is a long-term process that extends over large regions; conversely, floods are often short-term and localized. It is important that the extremes do not balance out, but often reinforce each other.

In this talk, we give an overview of a number of different aspects related to extreme water events. In particular, we describe challenges and methods within the interdisciplinary research cluster EXDIMUM, which is part of a federal funding program WaX that targets extreme water events.

## 3.6 A network approach to cloud organization

*Franziska Glassmeier (TU Delft, NL)*


**Joint work of** Franziska Glassmeier, Graham Feingold
**Main reference** Franziska Glassmeier, Graham Feingold: "Network approach to patterns in stratocumulus clouds", Proceedings of the National Academy of Sciences, Vol. 114(40), pp. 10578–10583, 2017.
**URL** https://doi.org//10.1073/pnas.1706495114

We discuss a network analysis of the organization of cloud constellations based on data from detailed cloud simulations (LES) of a stratocumulus cloud deck. We find the network structure to be neither random nor characteristic to natural convection. It is independent of the typical length scale (atmospheric boundary layer height). The latter is a consequence of entropy maximization (Lewis's Law with parameter 0.16). Non-6 sided cells occur according to a neighbor-number distribution variance of about 2. Reflecting the continuously renewing dynamics of Sc fields, large (many-sided) cells tend to neighbor small (few- sided) cells (Aboav-Weaire Law with parameter 0.9). By developing a heuristic model, we show that stratocumulus cell dynamics can be mimicked by versions of cell division and cell disappearance and are biased towards the expansion of smaller cells.

## 3.7 Integrated Modeling of Terrestrial Systems

*Stefan Kollet (Forschungszentrum Jülich, DE)*



Groundwater-to-atmosphere simulations are useful for closing the terrestrial water and cycle at the continental scale and interrogating anthropogenic impacts at different space and time scales. We want to understand how human water use, in this case groundwater pumping and irrigation, changes the natural terrestrial cycles over the European continent including local effects, such as changes in water table depths, evapotranspiration, and air temperature, and non-local effects, such as base flow, continental discharge and precipitation. In particular, we study whether these changes are systematic in space and time, and ultimately impact and potentially redistribute water resources across the continent. We present technical aspects of our work related to model coupling and high-performance computing technologies, and results illustrating the significant impact of human water use beyond individual watersheds.

## 3.8 SCALGO Live: Computational geometry algorithms and data structures in practice

*Thomas Moelhave (SCALGO – Aarhus, DK)*



By harnessing the power of global geographical digitisation and using the latest advances in big data processing technology, SCALGO builds innovative digital tools that enable our users to create better, more liveable and sustainable environments where there is space for water.

Our technology is based on decades of basic and applied research in algorithms for processing big geographic data at leading universities in both Europe and the USA. In this talk I will discuss some of the technical challenges we have faced in transitioning from basic computer science research to a powerful user-friendly product for users with varied backgrounds.

## 3.9 Machine learning for climate science and meteorology: new opportunities for computational geometry to shine?

*Peer Nowack (KIT – Karlsruher Institut für Technologie, DE)*



Machine learning techniques have gained widespread use in meteorology and climate science. In this talk, I will put forward a few ideas on how computational geometry could help address key methodological and scientific challenges in these application domains. In particular, computational geometry could lead to better input feature design for machine learning applications, or could contribute to the development of new methods to detect and track important weather phenomena.

These ideas could open up new routes for close, solution-oriented, collaborations between computer scientists and domain scientists. For example, I will suggest ways in which such approaches could improve results of machine learning methods to observationally constrain climate change projection uncertainties. I will further discuss important atmospheric dynamical phenomena which are currently still difficult to identify automatically with both high precision and recall. Here, I will emphasize the potential of methods from computational geometry to overcome longstanding event definition and detection barriers. A related advance would be the creation of a wider collection of new event-focused datasets that dynamically follow and characterize important weather systems, to be made available for future use in event-focused studies by domain scientists. Motivating examples include Ceppi and Nowack (2021) and Thomas et al. (2021).

### References

1 Thomas, C., Voulgarakis, A., Lim, G., Haigh, J., and Nowack, P. An unsupervised learning approach to identifying blocking events: the case of European summer, Weather and Climate Dynamics 2, 581 – 608 (2021). https://doi.org/10.5194/wcd-2-581-2021
2 Ceppi P. and Nowack P. Observational evidence that cloud feedback amplifies global warming, PNAS 118, e2026290118 (2021). https://doi.org/10.1073/pnas.2026290118
3 Nowack, P., Ceppi, P., Davis, S.M. et al. Response of stratospheric water vapour to warming constrained by satellite observations. Nature Geoscience 16, 577 – 583 (2023). https://doi.org/10.1038/s41561-023-01183-6

## 3.10 Challenges in analyzing and evaluating next-generation (km-scale) global climate simulations

*Vera Schemann (Universität Köln, DE)*



In global climate modelling a new generation of models is about to become feasible. By approaching km-scale resolutions, more information about regional and temporal variability of a changing climate system becomes possible. Furthermore, an improved representation of clouds and precipitation is expected and with that more insights into extreme events and their potential changing frequency and properties could be provided. But this new generation is also posing new challenges: The pure amount of data as well as the chaotic structure of the climate system will challenge our traditional way of analyzing climate model output or make it even impossible. New methods for statistical comparisons or identification of specific cluster and pattern have to be developed or adapted. This talk gave a short overview about this topic by starting with our traditional models and methods, introducing ongoing projects on developing new models and workflows and formulating upcoming questions and challenges.

## 3.11 Coresets for Data Reduction

*Christian Sohler (Universität Köln, DE)*



After a gentle introduction into analysis of algorithms, I survey some results from the area of coresets in the context of clustering. A coreset is a small summary of a data set with respect to an optimization problem. In the context of clustering, the input is a set of points and a coreset is a much smaller point set such that every solution to the clustering problem has roughly the same objective value for the coreset and the original point set. After introducing the basic definitions, I explain some applications of coresets and techniques to compute them.

## 3.12 Data-driven stochastic modelling of unresolved scales for weather and climate models

*Nikki Vercauteren (University of Oslo, NO)*



Limited computer resources lead to the need for a reduced representation of the weather and climate system when studying its dynamical behaviour. We discussed different approaches used to reduce the complexity of atmospheric models. One standard approach is to divide the dynamics into slow, resolved processes for which the fluid dynamical and thermodynamics equations are solved on a numerical grid, and fast or unresolved processes that are described through a simplified model or parameterisation. Another strategy is to simplify the model itself by projecting the equations of motion onto a smaller dimensional state space. We

discussed alternative, data-driven strategies to identify clusters, or coarse-grain patterns, in atmospheric datasets, and how to use such coarse-grain patterns, or atmospheric regimes, to inform model reduction approaches. Finally we discussed how to combine such a clustering framework with a stochastic modelling strategy. In this approach, a stochastic parameterisation of unresolved processes is learned from observations. In the learning process, the underlying, unobserved atmospheric regimes that lead to different dynamics of the unresolved variables to be modelled are clustered. The resulting parameterisation of the unresolved or fast degrees of freedom is hence adapted to the occurring atmospheric regimes. The example of application analysed turbulence modelling in context where turbulent mixing can be very intermittent. The stochastic formulation accommodates both the short-term intermittent behavior of turbulent mixing and the long-term average mixing. It could present a way forward for dealing with the complexities of unsteady flows in numerical weather prediction or climate models.

## 3.13 Topological Characterization and Uncertainty Visualization of Atmospheric Rivers

*Bei Wang Phillips (University of Utah – Salt Lake City, US)*


**Joint work of** Fangfei Lan, Brandi Gamelin, Lin Yan, Jiali Wang, Hanqi Guo, Bei Wang

Atmospheric rivers (ARs) are long, narrow regions in the atmosphere that transport water vapor from the Earth's tropics. ARs have been of great interest to climate scientists because they cause a large percentage of precipitation worldwide. In North America, ARs contribute significantly to water supply and flooding risk, especially in the western regions. However, ARs are difficult to characterize due to the lack of a universal definition and their varying shapes and sizes. Many AR detection tools (ARDTs) have been developed for different purposes, producing distinct AR boundaries. In this work-in-progress, we study the ARs detected by an ensemble of AR detection algorithms, quantify and visualize the uncertainty that arises from the output of these algorithms. We propose an uncertainty visualization framework that captures both the exterior and interior variability of an ensemble of ARs. This is build upon a collaboration with the Argonne National Laboratory (ANL). This research is partially supported by grants DOE DE-SC0021015 and NSF IIS-2145499.

# 4 Working groups

## 4.1 Sensor placement and path planning problems

*Mikkel Abrahamsen (University of Copenhagen, DK), Sándor Fekete (TU Braunschweig, DE), Jürgen Kusche (Universität Bonn, DE), Petra Mutzel (Universität Bonn, DE), and Vera Schemann (Universität Köln, DE)*



Problems of and methods for geometric optimization play an important role in the interface between application areas and fundamental theory. In this work group, we discussed a number of such questions. In particular, we discussed the problem of planning flight paths

with a Lagrangian focus (sampling the same air mass several times) and the issues rising during the comparison of measured and simulated parameter. Furthermore, we learned about the problems of placing stationary and floating sensors to measure rising sea levels.

## 4.2 Tracking of cloud fields

*Franziska Glassmeier (TU Delft, NL), Kevin Buchin (TU Dortmund, DE), Dwaipayan Chatterjee (Universität Köln, DE), Sándor Fekete (TU Braunschweig, DE), Siddharth Gupta (University of Warwick – Coventry, GB), André Nusser (University of Copenhagen, DK), Frank Staals (Utrecht University, NL), and Bei Wang Phillips (University of Utah – Salt Lake City, US)*



Atmospheric phenomena evolve internally, while at the same time being transported by the prevailing winds. The study of internal evolution is simplified by isolating it from transport, i.e., by studying the Lagrangian evolution of atmospheric phenomena along trajectories. We focused on cloud fields, i.e., constellations of many clouds in spatial proximity. The state-of-the-art approach for obtaining trajectories of cloud fields makes use of observationally constrained, simulated wind fields (reanalysis data). Different, purely-observational approaches for identifying the Lagrangian evolution of atmospheric phenomena through tracking are used within the Atmospheric Sciences. We discussed methods for identifying the Lagrangian evolution of cloud fields by tracking constellations of clouds in time-series of satellite images. Based on a range of example case studies from the tropical Atlantic, we identified two different strategies. Both approaches rely on tracking cloud fields as represented by the scalar field of cloud-optical depth and do not require a segmentation into cloud objects and surrounding cloud-free space.

The first approach is mostly topological with some geometric hybridization. It relies on keeping track of specific features of the cloud field (like local maxima of the intensity distribution). This results in a relatively fast and scalable approach for mapping the shape development, with readily available code by some of the participants. The cost is that this feature-based approach relies on some simplifications that may potentially cause discontinuities, so some post-processing of outcomes is indicated.

The second is purely geometric and relies on an Earth mover's distance. Specifically, it uses a weighted matching between consecutively frames to the relocation of the geometric distribution itself. This promises to more closely map the continuous process of shape development, in particular when for the relocation of the cloud field an underlying translation is assumed (either one for the whole field or local translations guided by optical flow). At this point, the challenge is to make the algorithmic realization scalable enough to make it useful for larger data sets.

Thanks to an existing implementation, the topological approach could already be tested during the Dagstuhl Seminar and showed promising results. The implementation of the geometric approach was also started during the seminar. Going forward, we have made arrangements to compare our two approaches with each other and to state-of-the-art approaches in cloud research. This comparison will introduce new methodology from computational geometry to cloud research. In addition, it will establish a benchmarking dataset and metrics and thus provide the foundation for future improvements.

## 4.3 Detection of blocking events

*Peer Nowack (KIT – Karlsruher Institut für Technologie, DE), Maike Buchin (Ruhr-Universität Bochum, DE), Susanne Crewell (Universität Köln, DE), Anne Driemel (Universität Bonn, DE), Jan-Henrik Haunert (Universität Bonn, DE), Benjamin Raichel (University of Texas at Dallas – Richardson, US), Melanie Schmidt (Heinrich-Heine-Universität Düsseldorf, DE), and Bei Wang Phillips (University of Utah – Salt Lake City, US)*



Atmospheric blocking events are key drivers of extreme weather in the mid-latitudes, including Europe and the US. The high-pressure systems "block" the normal zonal flow of air masses, and are often associated with major heatwaves during summer and cold spells during winter. Blocking patterns are also coupled to low-pressure systems and extreme rainfall in other regions. For example, this was the case in September 2023 when extreme flooding in Greece was coupled to a heatwave over Central to Northern Europe. An important science question is how the frequency and character of blocking events might change in the future in the presence of anthropogenic climate change. For this purpose, climate scientists need to analyse blocking events in century-scale climate model simulations, which requires automatic blocking detection methods.

In this break-out group, we discussed how computational geometry approaches could use a recently published ground-truth dataset for European summer blocking (Thomas et al., 2021) to develop more reliable blocking detection methods, compared to current blocking indices (which commonly disagree in their detection).

### References

1 Thomas, C., Voulgarakis, A., Lim, G., Haigh, J., and Nowack, P. An unsupervised learning approach to identifying blocking events: the case of European summer, Weather and Climate Dynamics 2, 581 – 608 (2021). https://doi.org/10.5194/wcd-2-581-2021

## 4.4 Data reduction

*Nikki Vercauteren (University of Oslo, NO), Dwaipayan Chatterjee (Universität Köln, DE), Peter Dueben (ECMWF – Bonn, DE), Siddharth Gupta (University of Warwick – Coventry, GB), Jeff M. Phillips (University of Utah – Salt Lake City, US), and Christian Sohler (Universität Köln, DE)*



Despite the complexity of the climate system, the existence of repeating and persisten patterns or weather regimes is known from observations. Hence one should expect that the very high dimensionality of data provided by climate simulations can be reduced to study such quasi-persistent weather regimes. In this working group we discussed possible data-driven approaches to identify persistent weather regimes and derive reduced models for the dynamics, based on climate reanalysis datasets. Different approaches to reduce the

dimension of input data where discussed. Once identified from data, the reduced models can be used to study the dynamics of transitions between weather regimes and identify factors favouring regime transitions. That can be pursued based on statistical causality approaches that help to investigate the impact of weather regimes on other atmospheric features such as cloud organization, or the occurrence of atmospheric rivers.

## Participants


- Mikkel Abrahamsen
University of Copenhagen, DK
- Pankaj Kumar Agarwal
Duke University – Durham, US
- Kevin Buchin
TU Dortmund, DE
- Maike Buchin
Ruhr-Universität Bochum, DE
- Dwaipayan Chatterjee
Universität Köln, DE
- Susanne Crewell
Universität Köln, DE
- Anne Driemel
Universität Bonn, DE
- Peter Dueben
ECMWF – Bonn, DE
- Sándor Fekete
TU Braunschweig, DE
- Franziska Glassmeier
TU Delft, NL
- Siddharth Gupta
University of Warwick – Coventry, GB
- Jan-Henrik Haunert
Universität Bonn, DE
- Stefan Kollet
Forschungszentrum Jülich, DE
- Jürgen Kusche
Universität Bonn, DE
- Thomas Moelhave
SCALGO – Aarhus, DK
- Petra Mutzel
Universität Bonn, DE
- Peer Nowack
KIT – Karlsruher Institut für Technologie, DE
- André Nusser
University of Copenhagen, DK
- Jeff M. Phillips
University of Utah – Salt Lake City, US
- Benjamin Raichel
University of Texas at Dallas – Richardson, US
- Vera Schemann
Universität Köln, DE
- Melanie Schmidt
Heinrich-Heine-Universität Düsseldorf, DE
- Christian Sohler
Universität Köln, DE
- Frank Staals
Utrecht University, NL
- Nikki Vercauteren
University of Oslo, NO
- Bei Wang Phillips
University of Utah – Salt Lake City, US


Report from Dagstuhl Seminar 23351

# Algorithms and Complexity for Continuous Problems

**Dmitriy Bilyk[*1], Michael Gnewuch[*2], Jan Vybíral[*3], Larisa Yaroslavtseva[*4], and Kumar Harsha[†5]**

1 **University of Minnesota – Minneapolis, US.** dbilyk@math.umn.edu
2 **Universität Osnabrück, DE.** michael.gnewuch@uni-osnabrueck.de
3 **Czech Technical University – Prague, CZ.** jan.vybiral@fjfi.cvut.cz
4 **Universität Graz, AT.** larisa.yaroslavtseva@uni-passau.de
5 **Universität Osnabrück, DE.** kumar.harsha@uos.de

**Abstract**

The Dagstuhl Seminar 23351 was held at the Leibniz Center for Informatics, Schloss Dagstuhl, from August 27 to September 1, 2023. This event was the 14th in a series of Dagstuhl Seminars, starting in 1991. During the seminar, researchers presented overview talks, recent research results, work in progress and open problems. The first section of this report describes the goal of the seminar, the main seminar topics, and the general structure of the seminar. The third section contains the abstracts of the talks given during the seminar and the forth section the problems presented at the problem session.

**Seminar** August 27 – September 1, 2023 – https://www.dagstuhl.de/23351
**2012 ACM Subject Classification** Mathematics of computing → Approximation; Mathematics of computing → Numerical analysis; Mathematics of computing → Probabilistic algorithms
**Keywords and phrases** computational stochastics, infinite-variate problems, quasi-Monte Carlo, sampling, tractability analysis
**Digital Object Identifier** 10.4230/DagRep.13.8.106

## 1 Executive Summary

*Michael Gnewuch (Universität Osnabrück, DE)*
*Dmitriy Bilyk (University of Minnesota – Minneapolis, US)*
*Jan Vybíral (Czech Technical University – Prague, CZ)*
*Larisa Yaroslavtseva (Universität Graz, AT)*



The goal of the Dagstuhl Seminar was to bring together researchers that work in various fields united by a common theme of complexity of continuous problems. In this context, complexity refers to the computational effort required to solve the problem approximately, up to a given error. For these, often high- or even infinite-dimensional, problems it is desirable to have theoretical bounds on the complexity as well as explicit constructions of (nearly) optimal algorithms. Their applications range over natural sciences, economics, statistics, engineering, computer science (including computer graphics, global optimization, and machine learning) and other areas of interest. The focus of the seminar was on the following highly interrelated topics:

[*] Editor / Organizer
[†] Editorial Assistant / Collector


Algorithms and Complexity for Continuous Problems, *Dagstuhl Reports*, Vol. 13, Issue 8, pp. 106–128
Editors: Dmitriy Bilyk, Michael Gnewuch, Jan Vybíral, and Larisa Yaroslavtseva
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

**Tractability analysis of high-dimensional problems.** Tractability analysis has its roots in information-based complexity. It studies continuous problems, where the instances are typically $d$-variate functions. The focus is on how the problem complexity depends on the error tolerance and on $d$. Topics discussed during the seminar were the study of recent notions of tractability and of new problem settings.

**Computational stochastics.** The focus was on strong approximation and quadrature of stochastic differential equations (SDEs) with non-globally Lipschitz continuous coefficients. Systematic investigations on the numerical approximation of such SDEs have started in the last decade and are still in their infancy.

**Infinite-variate problems: algorithms and complexity.** Many applications rely on models with countably many variables. The complexity analysis of the resulting continuous problems may be viewed as the limit of tractability analysis for d-variate functions, where d tends to infinity. For many problems sharp complexity bounds have been proved with the help of generic types of algorithms, as multilevel algorithms and multivariate decomposition methods, but mainly in the case where the spaces of input functions are weighted reproducing kernel Hilbert spaces (RKHSs) based on product weights or RKHSs of increasing smoothness. A major question was how to tackle important function spaces, which exhibit a different underlying structure.

**Well-distributed point configurations: discrepancy, quasi-Monte Carlo methods, dispersion.** Many problems in approximation theory and complexity rely on the existence and construction of good (random or deterministic) point distributions with respect to discrepancy, integration errors, strength of cubature formulas, dispersion, or discrete energy. Numerous questions in this area are still open: in various regimes sharp bounds for discrepancy and dispersion still remain unknown and explicit efficient constructions are still lacking. These problems were intensely discussed during the seminar.

**Linear and standard information in approximation theory.** A classical problem of approximation theory, learning theory and data analysis is the approximation of an unknown multivariate function from incomplete information. The information can be given by a limited number of function evaluations (standard information) or by evaluating a finite number of arbitrary linear functionals (linear information). Even in the simplest setting of approximation in a norm of a Hilbert space, the relation between using standard information and linear information was understood only recently and there is still a number of challenging open questions in this area. In several seminar talks interesting ideas to tackle these questions and new results were presented.

During the seminar we had five **overview talks** of 60 minutes (incl. discussion; one for each of the main research topics) given by leading experts in the respective area to enable reseachers from different fields to follow the regular talks. The speakers and the titles of these lectures were (ordered according to the list of main research topics):

- Peter Kritzer (RICAM Linz), *Tractability analysis.*
- Thomas Müller-Gronbach (Unversität Passau), *On the complexity of strong approximation of SDEs with a non-Lipschitz drift coefficient.*
- Klaus Ritter (RPTU Kaiserslautern-Landau), *Infinite-variate integration and $L^2$-approximation.*
- Stefan Steinerberger (University of Washington – Seattle), *Well-distributed point configurations.*
- Mario Ullrich (JKU Linz), *On optimal approximation based on random samples.*

These talks were delivered at the first three days of the seminar. Additionally, we had 24 **regular talks** of 30 minutes (incl. discussion), many of them presented by young researchers. Due to the time format, there was plenty of time for discussions and collaborations in smaller groups.

Originally we planned an additional poster session for the participants who do not present a talk. Since finally the number of time slots for talks was sufficient to ensure that everyone who wanted to present a talk could actually do so, there was no need for a poster session anymore.

On Wednesday morning we had a **problem session** to share interesting open problems and conjectures and to initiate further discussions. Several researchers used the opportunity to present interesting and challenging open problems. One problem was even solved in a discussion of some participants directly after the problem session. The detailed problem formulations can be found at the end of this report.

## 2 Table of Contents

# 3 Overview of Talks

## 3.1 Constructing Low-Discrepancy Point Sets: Subset Sampling and Optimal Sets

*François Clément (Sorbonne University – Paris, FR)*


**Joint work of** François Clément, Carola, Doerr, Kathrin Klamroth, Luís Paquete

Low-discrepancy sequences such as Sobol' or Halton have been designed to have small discrepancy values asymptotically. However, for practical applications, only a finite number of points will be possible, often too small to reach the asymptotic regime. In this talk, I present two different approaches to construct point sets with low discrepancy values.

The first is via the Star Discrepancy Subset Selection Problem, which consists in selecting from a fixed point set $P$ of size $n$ the best subset of size $k$, best being the subset with the smallest $L_\infty$ star discrepancy. We tackle this problem both via exact methods in dimensions 2 and 3 [1] and via heuristics in higher dimensions [2]. In particular, we observe that we are able to obtain sets with 10-50% smaller discrepancy for all tested dimensions and number of points.

The second is the construction of optimal point sets for the $L_\infty$ star discrepancy. This has been a rather overlooked problem, with the main previous work by White in 1977 solving for up to 6 points in dimension 2. We provide non-linear programming formulations to solve the problem for n smaller than 20 in dimension 2 and smaller than 8 in dimension 3 [3]. In particular, the structure of the local discrepancy values over $[0,1)^2$ for our optimal sets is very different to that of previously known sets such as Fibonacci or Sobol'.

### References

1 François Clément; Carola Doerr; Luís Paquete, *Star discrepancy subset selection: Problem formulation and efficient approaches for low dimensions*, Journal of Complexity, 70:101645, 2022
2 François Clément; Carola Doerr; Luís Paquete, *Heuristic approaches to obtain low-discrepancy point sets via subset selection*, 2023
3 François Clément; Carola Doerr; Kathrin Klamroth, Luís Paquete, *Constructing Optimal $L_\infty$ Star Discrepancy Sets*, https://arxiv.org/abs/2311.17463, 2024

## 3.2 On the existence of optimal shallow networks

*Steffen Dereich (Universität Münster, DE)*


**Joint work of** Steffen Dereic, Arnulf Jentzen, Sebastian Kassing

In this talk, we discuss existence of global minima in optimisation problems over shallow neural networks. More explicitly, the function class over which we minimise is the family of all functions that can be expressed as artificial residual or feedforward neural networks with one hidden layer featuring a specified number of neurons with ReLU (or Leaky ReLU) activation. We give existence results. Moreover, we provide counterexamples that illustrate the relevance of the assumptions imposed in the theorems.

## 3.3 Data compression using lattices for machine learning

*Kumar Harsha (Universität Osnabrück, DE)*


**Joint work of** Kumar Harsha, Michael Gnewuch, Marcin Wnuk

The mean squared error is one of the standard loss functions in supervised machine learning. However, calculating this loss for enormous data sets can be computationally demanding. Modifying a earlier approach [1], we present algorithms to reduce extensive data sets to a smaller size using rank-1 lattice rules. With this compression strategy in the precomputation step, every lattice point gets a pair of weights depending on the original data and responses, representing its relative importance. As a result, the compressed data makes iterative loss calculations in optimization steps much faster. The proposed compression strategy is highly beneficial for regression problems without an analytical solution. Our derivation of the formulas for the weights assumes that input functions have a convergent Fourier series and that the relevant Fourier coefficients form a hyperbolic cross. Accordingly, we have analyzed our algorithms' error for functions whose Fourier coefficients decay sufficiently fast such that they lie in Wiener algebras.

### References

1 Josef Dick; Michael Feischl, *A quasi-Monte Carlo data compression algorithm for machine learning*, Journal of Complexity, 67:101587, 12 2021

## 3.4 Results and open questions around the adaption problem

*Stefan Heinrich (RPTU – Kaiserslautern, DE)*



In the framework of Information-Based Complexity we consider the adaption problem for linear solution operators in the randomized setting. Along with surveying recent advances we discuss a number of remaining/arising/related open questions.

The results presented are contained in the following papers:

### References

1 Stefan Heinrich, *Randomized complexity of parametric integration and the role of adaption I. Finite dimensional case*, arXiv:2306.13471
2 Stefan Heinrich, *Randomized Complexity of Parametric Integration and the Role of Adaption II. Sobolev spaces*, arXiv:2306.13499
3 Stefan Heinrich, *Randomized Complexity of Vector-Valued Approximation*, to appear in: A. Hinrichs, P. Kritzer, F. Pillichshammer (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2022, Springer-Verlag, see also arxiv:2306.13697

## 3.5 Instant Neural Graphics Primitives

*Alexander Keller (NVIDIA – Berlin, DE)*


**Joint work of** Alexander Keller, Thomas Müller, Alex Evans, Christoph Schied
**Main reference** Thomas Müller, Alex Evans, Christoph Schied, Alexander Keller: "Instant neural graphics primitives with a multiresolution hash encoding", ACM Trans. Graph., Vol. 41(4), pp. 102:1–102:15, 2022.
**URL** https://doi.org//10.1145/3528223.3530127

Neural graphics primitives, parameterized by fully connected neural networks, can be costly to train and evaluate. We reduce this cost with a versatile new input encoding that permits the use of a smaller network without sacrificing quality, thus significantly reducing the number of floating point and memory access operations: a small neural network is augmented by a multiresolution hash table of trainable feature vectors whose values are optimized through stochastic gradient descent. The multiresolution structure allows the network to disambiguate hash collisions, making for a simple architecture that is trivial to parallelize on modern GPUs.

## 3.6 Tractability in unweighted function classes

*David Krieg (Johannes Kepler Universität Linz, AT)*


**Main reference** David Krieg: "Tractability of sampling recovery on unweighted function classes", CoRR, Vol. abs/2304.14169, 2023.
**URL** https://doi.org//10.48550/ARXIV.2304.14169

It is well-known that the integration problem as well as the problem of sampling recovery in the $L_2$-norm suffer from the curse of dimensionality on many classical function classes. This includes unweighted Korobov spaces (Sobolev spaces with mixed smoothness) as well as classical smoothness classes such as Hölder classes or the classes $C^k([0,1]^d)$. We show that those problems become (polynomially) tractable, even for $L_p$ approximation with $1 \leq p < \infty$, if we further impose the condition that the sum of Fourier coefficients is bounded.

The classes where we obtain tractability are unweighted (invariant under a reordering of the variables) and bigger than the corresponding weighted Korobov spaces. Tractability is achieved by the use of non-linear algorithms, while linear algorithms cannot do the job.

In fact, the tractability result is a relatively simple implication of powerful results by Rauhut and Ward [Appl. Comput. Harmon. Anal. 40 (2016), pp. 321-351] on $\ell_1$-minimization. The approach is not limited to the Fourier system, but it remains to be seen whether any "interesting" classes of nonperiodic functions can be tackled.

## 3.7 Tractability analysis

*Peter Kritzer (Österreichische Akademie der Wissenschaften – Linz, AT)*


**Joint work of** Peter Kritzer, Josef Dick, Adrian Ebert, Onyekachi Emenike, Fred J. Hickernell, Christian Irrgeher, Gunther Leobacher, Freidrich Pillichshammer, Henryk Wozniakowski

The information complexity of the problem of approximating an operator $S_d : \mathcal{F}_d \to \mathcal{G}_d$ for normed spaces $\mathcal{F}_d$ and $\mathcal{G}_d$ is defined as the minimal number $\mathrm{comp}(\varepsilon, d)$ of information evaluations needed to find an $\varepsilon$-approximation. A large literature specifies conditions under which the information complexity for a sequence of numerical problems defined for dimensions $d \in \{1, 2, \ldots\}$ grows at a moderate rate when the error threshold $\varepsilon$ decreases and/or the dimension increases, i.e., the sequence of problems is tractable.

In this talk, we give an overview of basic ideas in tractability analysis, highlight classical results for linear problems defined on Hilbert spaces, and outline some more recent results on this subject.

## 3.8 Optimal confidence intervals for randomized quadrature

*Robert J. Kunsch (RWTH Aachen, DE & TU Chemnitz, DE)*


**Main reference** Robert J. Kunsch: "Linear Monte Carlo quadrature with optimal confidence intervals", CoRR, Vol. abs/2309.09059, 2023.
**URL** https://doi.org//10.48550/ARXIV.2309.09059

We study the numerical integration of functions from isotropic Sobolev spaces $W_p^s([0,1]^d)$ using finitely many function evaluations within randomized algorithms, aiming for the smallest possible probabilistic error guarantee $\varepsilon > 0$ at confidence level $1 - \delta \in (0, 1)$. This error criterion is more challenging than the classical root-mean-squared error which is usually taken for Monte Carlo algorithms within information-based complexity and where only $n$ and $\varepsilon$ are put in relation. For spaces consisting of continuous functions, non-linear Monte Carlo methods with optimal confidence properties have already been known, in few cases even linear methods that succeed in that regard.[1] In this talk we promote a method called *stratified control variates* (SCV), see [2], and by it show that already linear methods achieve optimal probabilistic error rates in the high smoothness regime without the need to adjust algorithmic parameters to the uncertainty $\delta$. We also analysed a version of SCV in the low smoothness regime where $W_p^s([0,1]^d)$ may contain functions with singularities. Here, we observe a polynomial dependence of the error on $\delta^{-1}$ which cannot be avoided for linear methods. This is worse than what is known to be possible using non-linear algorithms where only a logarithmic dependence on $\delta^{-1}$ occurs if we tune in for a specific value of $\delta$. This new way of looking at randomized integration still leaves many questions open, for instance, precise lower bounds for linear methods in the low smoothness regime, as well as universal methods or optimality in mixed-smoothness spaces.

### References

1 Robert J. Kunsch; Daniel Rudolf, *Optimal confidence for Monte Carlo integration of smooth functions*, Advances in Computational Mathematics, 45:3095–3122, 2019
2 Robert J. Kunsch, *Linear Monte Carlo quadrature with optimal confidence intervals*, arXiv:2309.09059 [math.NA], 2023

## 3.9 High-dimensional approximation: Transforming periodic Approximations vs. Rando Fourier Features

*Laura Lippert (TU Chemnitz, DE)*



We study the problem of scattered-data approximation on $\mathbb{R}^d$, where we have given sample points and the corresponding function evaluations. We compare two approaches. In the first one we transform functions on $\mathbb{R}^d$ to $\mathbb{T}^d$, apply an approximation based on hyperbolic wavelet regression and transform the resulting function back. In fact, we transform the sample points to points on $\mathbb{T}^d$, evaluate the periodic basis functions at these points, create a matrix and solve the matrix equation with an LSQR-algorithm to get an approximation.
The other approach involves random Fourier features, where we draw frequencies $\omega_j \in \mathbb{R}^d$ at random and learn coefficients $a_j$ from the given data to construct the approximant, i.e.

$$f(\cdot) \approx \sum_j a_j \exp\left(\langle \omega_j, \cdot \rangle\right).$$

We give error estimates for both cases, involving different function spaces. We truncate the ANOVA decomposition to approximate high-dimensional functions of low effective dimension.

### References

1 Laura Lippert; Daniel Potts, *Variable Transformations in combination with Wavelets and ANOVA for high-dimensional approximation*, arXiv:2207.12826, 2022
2 Laura Lippert; Daniel Potts; Tino Ullrich, *Fast Hyperbolic Wavelet Regression meets ANOVA*, Numerische Mathematik 154, 155–207, 2023

## 3.10 On the minimal dispersion in the unit cube

*Alexander Litvak (University of Alberta – Edmonton, CA)*



We discuss ideas which lead to recent improvements of upper bounds for the minimal dispersion of a point set in the unit cube and its inverse. We also discuss sharpness of bounds. The talk is partially based on a joint work with G. Livshyts.

## 3.11 Fixed-radius spherical cap discrepancy

*Michelle Mastrianni (University of Minnesota – Minneapolis, US)*


**Joint work of** Dmitriy Bilyk, Stefan Steinerberger
**Main reference** Dmitriy Bilyk, Michelle Mastrianni, Stefan Steinerberger: "Single radius spherical cap discrepancy via gegenbadly approximable numbers", CoRR, Vol abs/2308.00694, 2023
**URL** https://doi.org/10.48550/arXiv.2308.00694

A seminal result of Beck shows that for any set of $N$ points on the $d$-dimensional sphere, there always exists a spherical cap such that the number of points in the cap deviates from the expected value by at least $N^{(1/2-1/2d)}$. We refine the result by removing a layer of

averaging: we show that, when $d$ is not 1 mod 4, there exists a set of real numbers such that for each $r > 0$ in the set one is always guaranteed to find a spherical cap with radius $r$ for which Beck's result holds. The main ingredient is a generalization of the notion of badly approximable numbers to the setting of Gegenbauer polynomials, which we call Gegenbadly approximable numbers. These are fixed numbers $x$ such that the sequence of Gegenbauer polynomials evaluated at $x$ avoids being close to 0 in a precise quantitative sense.

## 3.12 Complexity of composite linear problems

*Peter Mathé (Weierstraß Institut – Berlin, DE)*


**Joint work of** Peter Mathé, Bernd Hofmann
**Main reference** Bernd Hofmann, Peter Mathé: "The degree of ill-posedness of composite linear ill-posed problems with focus on the impact of the non-compact Hausdorff moment operator", CoRR, Vol abs/2111.01036, 2021.
**URL** https://doi.org/10.48550/arXiv.2111.01036

We consider compact composite linear operators in Hilbert space, where the composition is given by some compact operator followed by some non-compact one possessing a non-closed range. Focus is on the impact of the non-compact factor on the overall behavior of the decay rates of the singular values of the composition.

## 3.13 Comparison of Two Search Criteria for Lattice-based Kernel Approximation

*Weiwen Mo (KU Leuven, BE)*


**Joint work of** Weiwen Mo, Frances Y. Kuo, Dirk Nuyens, Abirami Srikumar, Ian H. Sloan
**Main reference** Frances Y. Kuo, Weiwen Mo, Dirk Nuyens, Ian H. Sloan, Abirami Srikumar: "Comparison of Two Search Criteria for Lattice-based Kernel Approximation", CoRR, Vol. abs/2304.01685, 2023.
**URL** https://doi.org//10.48550/ARXIV.2304.01685

The kernel interpolant in a reproducing kernel Hilbert space is optimal in the worst-case sense among all approximations of a function using the same set of function values. In this talk, we compare two search criteria to construct lattice point sets for use in lattice-based kernel approximation. The first candidate, $P_n^*$, is based on the power function that appears in machine learning literature. The second candidate, $S_n^*$, is a search criterion used for generating lattices for approximation using truncated Fourier series. We find that the empirical difference in error between the lattices constructed using the two search criteria is marginal. The criterion $S_n^*$ is preferred as it is computationally more efficient and has a bound with a superior convergence rate.

## 3.14 On the complexity of strong approximation of SDEs with a non-Lipschitz drift coefficient

*Thomas Müller-Gronbach (Universität Passau, DE)*



We study the complexity of pathwise approximation in $p$-th mean of the solution of a stochastic differential equation at the final time based on finitely many evaluations of the driving Brownian motion. First, we briefly review the case of equations with globally Lipschitz continuous coefficients, for which an error rate of at least $1/2$ in terms of the number of evaluations of the driving Brownian motion is always guaranteed by using the equidistant Euler-Maruyama scheme. Then we illustrate that giving up global Lipschitz continuity may lead to non-polynomial error rates for the Euler-Maruyama scheme or even for any method based on finitely many evaluations of the driving Brownian motion. Finally, we turn to recent complexity results in the case of equations with a drift coefficient that is not globally Lipschitz continuous. Here we focus on scalar equations with a Lipschitz continuous diffusion coefficient and a drift coefficient that satisfies piecewise smoothness assumptions or has fractional Sobolev regularity.

## 3.15 Training of DNNs with lattice rules

*Dirk Nuyens (KU Leuven, BE)*


**Joint work of** Dirk Nuyens, Alex Keller, Frances Kuo, Ian Sloan

Following Mishra, Rusch (2021) and Longo, Mishra, Rusch, Schwab (2021) we consider the so-called "generalization error" which describes how well a trained DNN can approximate a function on the whole domain. In fact this is just the L2 approximation error if measured in the L2-norm. We start from the assumption that the underlying function is periodic and in any Korobov space with smoothness alpha. This is motivated by the periodic model of the random diffusion field when doing uncertainty quantification.

## 3.16 Super-polynomial Accuracy of Median-of-means

*Zexin Pan (Stanford University, US)*


**Joint work of** Zexin Pan, Art Owen
**Main reference** Zexin Pan, Art B. Owen: "Super-polynomial accuracy of multidimensional randomized nets using the median-of-means", CoRR, Vol. abs/2208.05078, 2022.
**URL** https://doi.org//10.48550/ARXIV.2208.05078

Digital net is an important class of Quasi-Monte Carlo methods used for multidimensional integration. In the talk, I am going to show digital net randomized by linear scrambling and digital shift exhibits surprising concentration behavior. Taking the median of several digital net averages can exclude outliers and boost the convergence rate from nearly cubic to super-polynomial when the integrand is smooth. I will end with some discussions on the difficulties in building confidence intervals.

## 3.17 Sampling recovery in the uniform norm

*Kateryna Pozharska (National Academy of Sciences of Ukraine – Kyiv, UA & TU Chemnitz, DE)*


**Joint work of** David Krieg, Kateryna Pozharska, Mario Ullrich, Tino Ullrich
**Main reference** David Krieg, Kateryna Pozharska, Mario Ullrich, Tino Ullrich: "Sampling recovery in the uniform norm", CoRR, Vol. abs/2305.07539, 2023.
**URL** https://doi.org//10.48550/ARXIV.2305.07539

In the talk, I will present the results of our joint work with David Krieg, Mario Ullrich and Tino Ullrich [1] on the recovery of functions based on function evaluations.

The main emphasis is made on the uniform recovery, however we provide a method to transfer results for $L_2$-approximation to rather general seminorms (including $L_p$, $1 \leq p \leq \infty$). The underlying (least squares) algorithm is based on a random construction and subsampling based on the solution of the Kadison-Singer problem.

Besides an explicit bound for the corresponding sampling widths, we also obtain some interesting inequalities between the sampling, Kolmogorov and Gelfand widths. Namely, we show that there is an absolute constant $c > 0$, such that for a compact topological space $D$ and a compact subset $F$ of $C(D)$ the following holds

$$g_{2n}^{\mathrm{lin}}(F, L_\infty) \;\leq\; c\sqrt{n}\, d_n(F, L_\infty). \tag{1}$$

The bound in (1) is optimal up to constants, also if we consider only convex and symmetric $F$ and replace the Kolmogorov width $d_n$ by the Gelfand width $c_n$ on the right hand side. This means, that there are linear algorithms using $2n$ samples that are as good as all algorithms using arbitrary linear information up to a factor of at most $\sqrt{n}$. A result that cannot be true without oversampling [2]. Moreover, our results imply that linear sampling algorithms are optimal up to a constant factor for many reproducing kernel Hilbert spaces.

**Acknowledgements.** KP would like to acknowledge support by the Philipp Schwartz Fellowship of the Alexander von Humboldt Foundation.

### References

1 David Krieg; Kateryna Pozharska; Mario Ullrich; Tino Ullrich, *Sampling recovery in the uniform norm*, arXiv: 2305.07539v2, 2023
2 Erich Novak, *Deterministic and stochastic error bounds in numerical analysis*, Lecture Notes in Mathematics 1349, Springer-Verlag, 1988

## 3.18 Randomized Euler algorithm for SDEs with drift in integral form and its connection with Perturbed SGD

*Pawel Przybylowicz (AGH Univ. of Science & Technology-Krakow, PL)*



We will present a version of randomized Euler algorithm for approximation of solutions of stochastic differential equations when the drift is in integral form. We will show upper estimates for the error and some lower bounds in the worst-case model. We will show some relationship of the randomized Euler algorithm with perturbed SGD used in machine learning.

## 3.19 Infinite-variate Integration and $L^2$-Approximation

*Klaus Ritter (RPTU – Kaiserslautern, DE)*



In this survey we consider two basic computational problems, integration and $L^2$-approximation, for functions of countably infinite real variables. For motivation we briefly refer to stochastic models based on iid-sequences of random variables, e.g., series expansions of stochastic processes and random fields.

Thereafter we discuss appropriate cost models (information cost) for infinite-variate problems as well as the general structure of the functions spaces (RKHS of tensor product form) that have been studied so far. Finally, we present results (decay of the minimal errors) and key ingredients of the proofs for tensor products of weighted spaces and of spaces of increasing smoothness.

## 3.20 Integration and $L^2$-Approximation on Gaussian Spaces

*Robin Rüßmann (RPTU – Kaiserslautern, DE)*



We study integration and approximation on reproducing kernel Hilbert spaces with Gaussian kernels of tensor product form. We find new results in the infinite-variate case, and we improve some known results in the finite-variate case.

Our most important tool is a close relation between Gaussian spaces and Hermite spaces of infinite smoothness and tensor product form.

This relation allows us to transform any algorithm for integration or approximation on the Gaussian space into an algorithm for the same problem on the Hermite space and vice versa, preserving error and cost, which means that known upper and lower error bounds for one function space setting also apply to the other setting.

## 3.21 Large holes in large point sets

*Mathias Sonnleitner (Universität Passau, DE)*



Dispersion is a measure of equidistribution of point sets. We give a brief introduction and survey known results with focus on a large number of points compared to the dimension. From the literature on coverings of Euclidean space and convex body approximation, we deduce implications for the spherical case and point to open problems.

## 3.22 Multi-level quasi-Monte Carlo methods for kernel interpolation in uncertainty quantification

*Abirami Srikumar (UNSW Sydney, AU)*


**Joint work of** Abirami Srikumar, Alexander D. Gilbert, Michael Giles, Frances Y. Kuo, Ian H. Sloan

As high-dimensional problems become increasingly prevalent in many applications, the effective evaluation of these problems within the limits of our current technology poses a great hurdle due to the exponential increase in computational cost as dimensionality increases. One class of strategies for evaluating such problems efficiently are quasi-Monte Carlo (QMC) methods.

In this talk, we explore the effectiveness of multi-level quasi-Monte Carlo methods for approximating solutions to elliptic partial differential equations with stochastic coefficients that depend periodically on the stochastic parameters. In particular, we are interested in fast approximation using kernel-based lattice point interpolation. We present some regularity results, theory on the convergence properties of errors of such approximations and the results of numerical experiments.

## 3.23 Well-distributed Point Configurations

*Stefan Steinerberger (University of Washington – Seattle, US)*



This is a survey talk to describe several different problems in the broad framework of well-distributed point configurations. Problems discussed include: low-discrepancy sets of points, minimizers of energy functionals, greedy sequences, problems on the sphere $S^2$ and minimizer of the logarithmic energy as well as the connection to the crystallization conjecture.

## 3.24 Improving diversity in StableDiffusion with genetic crossover and well distributed point configurations

*Olivier Teytaud (Meta AI Research – Tournon-sur-Rhone, FR)*


**Joint work of** Olivier Teytaud, Mariia Zameshina, Laurent Najman, Mathurin Videau

Latent diffusion models excel at producing high-quality images from text. Yet, concerns appear about the lack of diversity in the generated imagery. To tackle this, we introduce Diverse Diffusion, a method for boosting image diversity beyond gender and ethnicity, spanning into richer realms, including color diversity. Diverse Diffusion is a general unsupervised technique that can be applied to existing text-to-image models. Our approach focuses on finding vectors in the Stable Diffusion latent space that are distant from each other. We generate multiple vectors in the latent space until we find a set of vectors that meet the desired distance requirements and the required batch size. To evaluate the effectiveness of our diversity methods, we conduct experiments examining various characteristics, including color

diversity, LPIPS metric, and ethnicity/gender representation in images featuring humans. The results of our experiments emphasize the significance of diversity in generating realistic and varied images, offering valuable insights for improving text-to-image models. Through the enhancement of image diversity, our approach contributes to the creation of more inclusive and representative AI-generated art.

We also present genetic crossovers for combining latent variables into a better latent variable.

After the seminar, we start a cool collaboration with Carola, Stefan, Dmitry, Laurent, for producing fantastic point configurations in high-dimensional point spaces.

Joint work ESIEE and Meta.

## 3.25 On optimal approximation based on random samples

*Mario Ullrich (Johannes Kepler Universität Linz, AT)*


**Joint work of** Mario Ullrich, Matthieu Dolbeault, David Krieg, Katerina Pozharska, Mathias Sonnleitner, Tino Ullrich
**URL** https://arxiv.org/search/?query=Ullrich%2C+Mario&searchtype=author&abstracts=hide&order=-announced_date_first&size=50

In this talk we give an overview of some recent and not so recent developments in the area of approximation of functions based on function evaluations. The emphasis is on information-based complexity and the worst-case setting, i.e., we ask for the minimal number of information (aka measurements) needed by any algorithm to achieve a prescribed error for all inputs, basically ignoring implementation issues.

However, it turned out that in many cases, certain (unregularized) least squares methods based on "random" information, like function evaluations, can catch up with arbitrary algorithms based on arbitrary linear information, i.e., the best we can do theoretically.

## 3.26 Sampling numbers of smoothness classes via $l_1$ minimization

*Tino Ullrich (TU Chemnitz, DE)*


**Joint work of** Thomas Jahn, Tino Ullrich, Felix Voigtlaender
**Main reference** Thomas Jahn, Tino Ullrich, Felix Voigtlaender: "Sampling numbers of smoothness classes via $l_1$-minimization", Journal of Complexity, Vol. 79, p. 101786, 2023.
**URL** https://doi.org//10.1016/j.jco.2023.101786

Using techniques developed recently in the field of compressed sensing we show new upper bounds for general (nonlinear) sampling numbers of (quasi-)Banach smoothness spaces in $L_2$. In particular, we show that in relevant cases such as mixed and isotropic weighted Wiener classes or Sobolev spaces with mixed smoothness, sampling numbers in $L_2$ can be upper bounded by best n-term trigonometric widths in $L_\infty$. We describe a recovery procedure from m function values based on $l_1$-minimization (basis pursuit denoising). With this method, a significant gain in the rate of convergence compared to recently developed linear recovery methods is achieved. In this deterministic worst-case setting we see an additional speed-up of $m^{-1/2}$ (up to log factors) compared to linear methods in case of weighted Wiener spaces. For

their quasi-Banach counterparts even arbitrary polynomial speed-up is possible. Surprisingly, our approach allows to recover mixed smoothness Sobolev functions belonging on the $d$-torus with a logarithmically better rate of convergence than any linear method can achieve when $1 < p < 2$ and $d$ is large. This effect is not present for isotropic Sobolev spaces.

## 3.27 Some Computational Approaches to the Pair Correlation Statistic

*Christian Weiß (Hochschule Ruhr-West – Mülheim, DE)*



The pair correlation statistic measures the behavior of gaps between the first elements of a sequence on a local scale. Although a generic independent, identically distributed random sequence drawn from uniform distribution has so-called Poissonian pair correlations, there are only few explicitly known such examples. The main reason why they are difficult to find is that it is in general hard to calculate the pair correlation statistic. In this talk, we therefore concentrate on computational aspects of the pair correlation and present some possible approaches. For instance, geometric properties like the gap structure, an underlying lattice structure of the sequence or the discrepancy turn out to be useful tools. We also present some recent combinatorial results.

## 3.28 The fixed vector randomised lattice algorithm for high-dimensional integration

*Laurence Wilkes (KU Leuven, BE)*


**Joint work of** Frances Y. Kuo, Dirk Nuyens, Laurence Wilkes
**Main reference** Frances Y. Kuo, Dirk Nuyens, Laurence Wilkes: "Random-prime-fixed-vector randomised lattice-based algorithm for high-dimensional integration", J. Complex., Vol. 79, p. 101785, 2023.
**URL** https://doi.org//10.1016/J.JCO.2023.101785

The fixed vector algorithm offers a very simple solution to the problem of producing a lattice-based randomised algorithm for numerical integration with the optimal randomised error. We shift the construction of the generating vector of the lattice to a precomputation so that the only randomised element is to choose the number of function evaluations from a suitable range. In the talk, we will look at the existence result for such a fixed generating vector, a method to construct the vector and finally look at how the algorithm can be generalised to work in the half-period cosine space and with a lower smoothness parameter.

## 3.29 Randomized approximation of summable sequences: adaptive and non-adaptive

*Marcin Wnuk (Universität Osnabrück, DE)*


**Joint work of** Marcin Wnuk, Erich Novak, Robert Kunsch

We are considering approximation of the identical embedding of finite-dimensional spaces $S_M : \ell_1^M \to \ell_\infty^M$ by randomized algorithms. We establish a lower bound on the $n$-th minimal error which is, roughly speaking, of the form: there exists an $\varepsilon > 0$ such that if for all $M \in \mathrm{N}$

$$e^{\text{ran-non}}(n(M), S_M) < \varepsilon$$

then

$$n(M) \geq C\sqrt{\log(M)}.$$

Here $e^{\text{ran-non}}(n, S)$ denotes the $n$-th minimal error which can be obtained when using randomized non-adaptive algorithms. This has at least two interesting consequences:

1. We are able to give an example of a sequence of linear problems for which the $n$-th minimal error for randomized adaptive algorithms decreases much faster than the minimal error for randomized non-adaptive algorithms-a phenomenon which was only recently observed by S.Heinrich in the context of vector-valued mean computation.
2. It enables us to show that the only operators which can be arbitrarily well approximated by randomized non-adaptive algorithms are compact operators- i.e. exactly the same operators which can be arbitrarily well approximated by deterministic algorithms.

# 4 Open problems

## 4.1 Low discrepancy sets in 2D

*Dmitriy Bilyk (University of Minnesota – Minneapolis, US)*



### Problem 1: Permutation sets

This problem is concerned with new constructions of two-dimensional low discrepancy sets. Two standard examples of sets achieving the lowest possible growth of the discrepancy $\mathcal{O}(\log N)$ are the van der Corput set and the Fibonacci lattice.

The digit-reversing van der Corput set is a set of $N = 2^n$ points of the following form

$$\big(0.x_1x_2\dots x_n, 0.x_nx_{n-1}\dots x_1\big),$$

where the coordinates are written as binary fractions of length $n$, and the binary digits $x_i$ take values zero or one. Many other variations are known (digit shifts and digit scrambling, cyclic shifts, constructions in bases other than two etc).

The Fibonacci lattice consists of $N = F_n$, where $F_n$ is the $n^{th}$ Fibonacci number, and has the form

$$\left(\frac{k}{F_n}, \left\{\frac{kF_{n-1}}{F_n}\right\}\right), \quad k = 0, 1, \dots, F_n - 1.$$

This can be viewed as an approximation of the Kronecker lattice, i.e. points $(k/N, \{k\alpha\})$, where $\alpha$ is the golden ratio. This generalizes to similar constructions using approximants to other badly approximable numbers.

One can easily observe that both of the examples presented above are of the form

$$\left(\frac{k}{N}, \frac{\sigma(k)}{N}\right), \quad k = 0, 1, \ldots, N-1,$$

where $\sigma$ is a permutation of an $N$-elements set $\{0, 1, \ldots, N-1\}$. Therefore, natural questions arise:

- Which other permutations $\sigma$ produce low discrepancy sets?
- Which structural and combinatorial properties of the permutation $\sigma$ are responsible for low discrepancy of the arising point sets?
- Do minimizers of the discrepancy always have a structure of such "permutation sets"? (at least, if restricted to the grid $(i/N, j/N)$?)
- Are $L^2$ (or more generally, $L^p$ discrepancies) minimized by "permutation sets"?
- Closely related question: For a set of points $z_1, \ldots, z_N \in \mathbb{T}^2$, where $z_i = (x_i, y_i)$, consider the tensor-product interaction energy of the form

  $$E(z_1, \ldots, z_N) = \sum_{i,j} f(x_i - x_j) f(y_i - y_j).$$

  Under which conditions on the function $f$ are the minimizers of this energy "permutation sets"? And which permutations produce minimizers? (Through Warnock-type formulas, the periodic $L^2$ discrepancy can be rewritten exactly as such an energy, and other versions of the $L^2$ discrepancies are closely related to similar energies.)
- Are there generalizations of permutation constructions to higher dimensions?

#### Problem 2: Projections of the vertices of the unit cube

The aforementioned van der Corput set can be realized as an image of the $2^n$ vertices of the $n$-dimensional unit cube $\{0, 1\}^n$ under a linear mapping $A : \mathbb{R}^n \to \mathbb{R}^2$ given by the matrix

$$A = \begin{pmatrix} \frac{1}{2} & \frac{1}{2^2} & \cdots & \frac{1}{2^n} \\ \frac{1}{2^n} & \frac{1}{2^{n-1}} & \cdots & \frac{1}{2} \end{pmatrix}.$$

- Can other mappings be constructed which similarly produce low discrepancy sets?
- Can probabilistic results be proved for the discrepancy of random projections?
- Can this approach be extended to construct well distributed sets in higher dimensions?

## 4.2 Standard information versus linear information for $L_p$-approximation

*David Krieg (Johannes Kepler Universität Linz, AT)*



Let $D$ be a compact domain of unit volume and let $C(D)$ be the space of continous complex-valued functions on $D$. Consider a compact, convex and symmetric subset $F$ of $C(D)$. For $1 \leq p \leq \infty$, we want to compare the rate of convergence of the numbers

$$g_n(F, L_p) = \inf_{\substack{x_1, \ldots, x_n \in D \\ \phi\colon \mathbb{C}^n \to L_p(D)}} \sup_{f \in F} \big\| f - \phi\big(f(x_1), \ldots, f(x_n)\big) \big\|_p$$

and

$$c_n(F, L_p) = \inf_{\substack{L_1,\dots,L_n \in C(D)' \\ \phi\colon \mathbb{C}^n \to L_p(D)}} \sup_{f\in F} \big\| f - \phi\big(L_1(f), \dots, L_n(f)\big)\big\|_p$$

for $n \to \infty$. These numbers describe the minimal worst case error for $L_p$-approximation on $F$ if $n$ function values, respectively $n$ arbitrary continuous linear measurements are allowed. The rate of convergence of a non-negative and non-increasing sequence $(x_n)$, denoted by $\deg(x_n)$, is the supremum of all $t > 0$ such that $(x_n) \in \mathcal{O}(n^{-t})$.

How much can we loose in the rate when restricting from arbitrary linear information to function evaluation? That is, the task is to study the loss function

$$\mathrm{loss}_p(\alpha) = \sup_{\substack{D \text{ \& } F \text{ as above:} \\ \deg(c_n(F,L_p)) = \alpha}} \Big[\deg(c_n(F, L_p)) - \deg(g_n(F, L_p))\Big].$$

How does the loss function behave as a function of $\alpha > 0$ and $1 \le p \le \infty$?

Results until 2012 are collected in [4, Section 29.1]. The univariate Sobolev spaces $W_1^s([0,1])$ of integrability one show that $\mathrm{loss}_p(\alpha) \ge 1/2$ for all $p \ge 2$, at least if $\alpha > 1$. Recent upper bounds imply that in fact $\mathrm{loss}_p(\alpha) = 1/2$ if either $p = 2$ (see [2] together with [1]) or $p = \infty$ (see [3]). Is it true that $\mathrm{loss}_p(\alpha) = 1/2$ holds for all $p \ge 2$ and $\alpha > 1$?

### References


1 J. Creutzig, P. Wojtaszczyk. Linear vs. nonlinear algorithms for linear problems. Journal of Complexity **20** (2004), 807–820.

2 M. Dolbeault, D. Krieg, M. Ullrich. A sharp upper bound for sampling numbers in $L_2$. *Applied and Computational Harmonic Analysis* **63** (2023), 113–134.

3 D. Krieg, K. Pozharska, M. Ullrich, T. Ullrich. Sampling recovery in the uniform and other norms, preprint, `https://arxiv.org/abs/2305.07539`.

4 E. Novak, H. Woźniakowski. Tractability of multivariate problems. Volume III: Standard information for operators. Volume 18 of *EMS Tracts in Mathematics*. EMS, Zürich, 2012.


## 4.3 Are linear algorithms non-adaptive?

*Mathias Sonnleitner (Universität Passau, DE)*



Let $F, G$ be normed spaces (over $\mathbb{R}$) and $S : F \to G$ be linear and bounded. Suppose that for each $f \in F$, we have $n$ pieces of adaptive information

$$N(f) = \left(L_1(f), L_2\left(f, L_1(f)\right), \dots, L_n\left(f, L_2\left(f, L_1(f)\right), \dots\right)\right), \quad f \in F$$

based on linear and continuous functionals on $F$. Let $A = \varphi \circ N : F \to G$ be a linear algorithm with $\varphi : \mathbb{R}^n \to G$.

Can we find $N^{\mathrm{non}}(f) = \left(L_1(f), \dots, L_n(f)\right), f \in F$, where $L_1, \dots, L_n : F \to \mathbb{R}$ are linear and (linear) $\widetilde{\varphi} : \mathbb{R}^n \to G$ such that $A = \widetilde{\varphi} \circ N^{\mathrm{non}}$?

This basic problem was solved completely using elementary means in a discussion with David Krieg, Robert Kunsch and Marcin Wnuk during the seminar.

## 4.4 4-regular graphs from the van der Corput sequence

*Stefan Steinerberger (University of Washington – Seattle, US)*



Consider the following test for pseudorandomness: to any distinct, real numbers $x_1, \dots, x_n$, we associate a 4-regular graph $G$ as follows: using $\pi$ to denote the permutation ordering the elements, $x_{\pi(1)} < x_{\pi(2)} < \dots < x_{\pi(n)}$, we build a graph on $\{1, \dots, n\}$ by connecting $i$ and $i+1$ (cyclically) and $\pi(i)$ and $\pi(i+1)$ (cyclically).

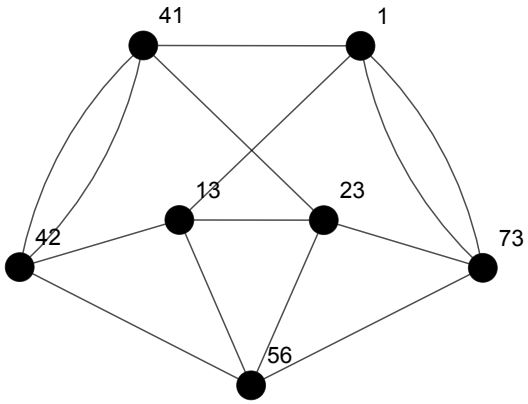


**Figure 1** The Graph for the first few digits of $\sqrt{2}$: $1, 41, 42, 13, 56, 23, 73$. If the digits of $\sqrt{2}$ behave like truly random numbers, these graphs have optimal expansion properties.

One reason why these graphs are interesting is the following result for the second eigenvalue $\lambda_2$ of the adjacency matrix of such a graph.

▶ **Theorem 1** (Friedman, Theorem 1.2. in [1]). *If $x_1, \dots, x_n$ are i.i.d. random variables chosen from an absolutely continuous distribution and $G$ is the 4-regular graph constructed from them as above, then for any $\varepsilon > 0$ there exists $c > 0$ such that with likelihood at least $1 - c/n$, we have*

$$|\lambda| \leq 2\sqrt{3} + \varepsilon.$$

As it turns out, this graph construction can indeed be effectively used for understanding the strength of a random number generator: if the numbers are truly random, the arising graphs are random in a "near-"optimal way [3].

One could now wonder what happens to these graphs when the underlying real numbers are far from "random" and two canonical examples are the Kronecker sequence and the van der Corput sequence. Fig. 2 shows the type of graph obtained from the van der Corput sequence in base 2

$$\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \dots$$

We observe the emergence of some strange topological features that are currently unexplained.

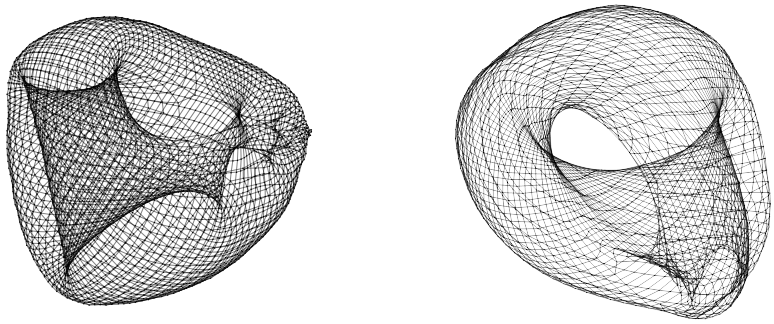

**Figure 2** The first 4096 elements of the van der Corput sequence in base 2 (left) and the first $2187 = 3^7$ elements of the van der Corput sequence in base 3 (right).

**Problem.** What is this emerging manifold? How does it depend on the base $b$ of the van der Corput sequence?

Something similar happens to the Kronecker sequence but there it is perhaps slightly less mysterious (the continued fraction expansion can help to explain it). It has since become clear that this graph structure tends to exhibit all sorts of strange patterns for structured sequences and many examples are given in [2]. Very few of these cases are understood.

### References


1 J. Friedman, A proof of Alon's second eigenvalue conjecture and related problems, Mem. Amer. Math. Soc. 195 (2008), no. 910

2 D. Korssjoen, B. Li, S. Steinerberger, R. Tripathi, R. Zhang, Finding Structure in Sequences of Real Numbers via Graph Theory: a Problem List, Involve 15, 251-270 (2022).

3 S. Steinerberger, Using Expander Graphs to test whether samples are i.i.d, arXiv:2008.01153


## 4.5 On a problem of classes of optimal information

*Mario Ullrich (Johannes Kepler Universität Linz, AT)*


**Joint work of** Mario Ullrich, Mathias Sonnleitner
**Main reference** Mathias Sonnleitner, Mario Ullrich: "On the power of iid information for linear approximation", CoRR, Vol. abs/2310.12740, 2023.
**URL** https://doi.org//10.48550/ARXIV.2310.12740

New open problem arise after recent results on the power of standard information (function values), and more general classes $\Lambda$, indicate that already under mild assumptions on $\Lambda$ we might hope for optimal approximations, i.e., that the corresponding minimal errors are of the same order as minimal errors based on arbitrary information. Several of related open problems can be found in the survey paper [1], which was nearly finished during the Dagstuhl Seminar 23351. A (simple, and too optimistic) version of an open problem was presented at the seminar, and already disproved in a subsequent discussion with David Krieg and Erich Novak.

### References


1 Mathias Sonnleitner, Mario Ullrich. *On the power of iid information for linear approximation*, arXiv:2310.12740, 2023.

## Participants


- Dmitriy Bilyk
University of Minnesota – Minneapolis, US
- François Clément
Sorbonne University – Paris, FR
- Ronald Cools
KU Leuven, BE
- Steffen Dereich
Universität Münster, DE
- Benjamin Doerr
Ecole Polytechnique – Palaiseau, FR
- Carola Doerr
CNRS, Sorbonne University – Paris, FR
- Michael Giles
University of Oxford, GB
- Michael Gnewuch
Universität Osnabrück, DE
- Kumar Harsha
Universität Osnabrück, DE
- Stefan Heinrich
RPTU – Kaiserslautern, DE
- Alexander Keller
NVIDIA, DE
- David Krieg
Johannes Kepler Universität Linz, AT
- Peter Kritzer
Österreichische Akademie der Wissenschaften – Linz, AT
- Thomas Kühn
Universität Leipzig, DE
- Robert J. Kunsch
RWTH Aachen, DE & TU Chemnitz, DE
- Frances Y. Kuo
UNSW Sydney, AU
- Laura Lippert
TU Chemnitz, DE
- Alexander Litvak
University of Alberta – Edmonton, CA
- Michelle Mastrianni
University of Minnesota – Minneapolis, US
- Peter Mathé
Weierstraß Institut – Berlin, DE
- Weiwen Mo
KU Leuven, BE
- Thomas Müller-Gronbach
Universität Passau, DE
- Erich Novak
Friedrich-Schiller-Universität Jena, DE
- Dirk Nuyens
KU Leuven, BE
- Zexin Pan
Stanford University, US
- Kateryna Pozharska
National Academy of Sciences of Ukraine – Kyiv, UA & TU Chemnitz, DE
- Pawel Przybylowicz
AGH Univ. of Science & Technology-Krakow, PL
- Klaus Ritter
RPTU – Kaiserslautern, DE
- Daniel Rudolf
Universität Passau, DE
- Robin Rüßmann
RPTU – Kaiserslautern, DE
- Mathias Sonnleitner
Universität Passau, DE
- Abirami Srikumar
UNSW Sydney, AU
- Stefan Steinerberger
University of Washington – Seattle, US
- Olivier Teytaud
Meta AI Research – Tournon-sur-Rhone, FR
- Mario Ullrich
Johannes Kepler Universität Linz, AT
- Tino Ullrich
TU Chemnitz, DE
- Jan Vybíral
Czech Technical University – Prague, CZ
- Christian Weiß
Hochschule Ruhr-West – Mülheim, DE
- Laurence Wilkes
KU Leuven, BE
- Marcin Wnuk
Universität Osnabrück, DE


Report from Dagstuhl Seminar 23352

# Integrating HPC, AI, and Workflows for Scientific Data Analysis

**Rosa M. Badia*[1], Laure Berti-Equille*[2], Rafael Ferreira da Silva*[3], and Ulf Leser*[4]**

1 **Barcelona Supercomputing Center, ES.** rosa.m.badia@bsc.es
2 **Research and Development Institute, IRD Montpellier, FR.** laure.berti@ird.fr
3 **Oak Ridge National Laboratory, US.** silvarf@ornl.gov
4 **Humboldt University of Berlin, DE.** leser@informatik.hu-berlin.de

## Abstract

The Dagstuhl Seminar 23352, titled "Integrating HPC, AI, and Workflows for Scientific Data Analysis," held from August 27 to September 1, 2023, was a significant event focusing on the synergy between High-Performance Computing (HPC), Artificial Intelligence (AI), and scientific workflow technologies. The seminar recognized that modern Big Data analysis in science rests on three pillars: workflow technologies for reproducibility and steering, AI and Machine Learning (ML) for versatile analysis, and HPC for handling large data sets. These elements, while crucial, have traditionally been researched separately, leading to gaps in their integration. The seminar aimed to bridge these gaps, acknowledging the challenges and opportunities at the intersection of these technologies. The event highlighted the complex interplay between HPC, workflows, and ML, noting how ML has increasingly been integrated into scientific workflows, thereby enhancing resource demands and bringing new requirements to HPC architectures, like support for GPUs and iterative computations. The seminar also addressed the challenges in adapting HPC for large-scale ML tasks, including in areas like deep learning, and the need for workflow systems to evolve to leverage ML in data analysis fully. Moreover, the seminar explored how ML could optimize scientific workflow systems and HPC operations, such as through improved scheduling and fault tolerance. A key focus was on identifying prestigious use cases of ML in HPC and understanding their unique, unmet requirements. The stochastic nature of ML and its impact on the reproducibility of data analysis on HPC systems was also a topic of discussion.

**Seminar** August 27 – September 1, 2023 – https://www.dagstuhl.de/23352

**2012 ACM Subject Classification** Computing methodologies → Distributed computing methodologies; Computing methodologies → Machine learning; Computing methodologies → Parallel computing methodologies

**Keywords and phrases** Large scale data presentation and analysis, Exascale class machine optimization, Performance data analysis and root cause detection, High dimensional data representation

**Digital Object Identifier** 10.4230/DagRep.13.8.129

* Editor / Organizer



Integrating HPC, AI, and Workflows for Scientific Data Analysis, *Dagstuhl Reports*, Vol. 13, Issue 8, pp. 129–164
Editors: Rosa M. Badia, Laure Berti-Equille, Rafael Ferreira da Silva, and Ulf Leser



Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

# 1 Executive Summary

*Rosa M. Badia (Barcelona Supercomputing Center, ES)*
*Laure Berti-Equille (IRD – Montpellier, FR)*
*Rafael Ferreira da Silva (Oak Ridge National Laboratory, US)*
*Ulf Leser (HU Berlin, DE)*



The Executive Summary for the Dagstuhl Seminar 23352 on "Integrating HPC AI and Workflows for Scientific Data Analysis" encapsulates a comprehensive discussion on the integration of High-Performance Computing (HPC), Artificial Intelligence (AI), and workflow technologies. The seminar, held from August 27 to September 1, 2023, was pivotal in highlighting the interdependence of these technologies for modern Big Data analysis. With a focus on bridging the gaps between these historically siloed areas, the seminar addressed the augmentation of resource demands due to the integration of AI into scientific workflows, the challenges posed to HPC architectures, and the exploration of AI's potential in optimizing workflow systems and operations, including scheduling and fault tolerance.

The seminar proffered a nuanced understanding of AI+HPC integrated workflows, elaborating on the different modes in which AI and HPC components could be coupled within workflows. These ranged from AI models replacing computationally intensive components (AI-in-HPC) to AI models that operate externally to steer HPC components or generate new data (AI-out-HPC), and to concurrent AI models that optimize HPC runtime systems (AI-about-HPC). Such integration is vital for the future of scientific workflows, where AI and HPC not only coexist but also co-evolve to foster more effective and intelligent scientific inquiry.

A shift in the paradigm of HPC systems towards real-time interaction within workflows was another focal point of the seminar. Moving away from the traditional batch-oriented systems, the seminar shed light on the emerging need for workflows that support dynamic, on-the-fly interactions. These interactions are not only vital for the real-time steering of computations and the runtime recalibration of parameters but also for making informed decisions on cost-value trade-offs, thereby optimizing both computational and financial resources.

The discussion also ventured into the realm of federated workflows, distinguishing them from the conventional grid computing model. Federated workflows, or cross-facility workflows, emphasize the orchestration of workflows across different computational facilities, each with distinct environments and policies. This paradigm advocates for a seamless execution of complex processes, underscoring the necessity of maintaining coherence and coordination throughout the workflow life cycle.

Contractual and quality-of-service (QoS) considerations in federated workflows, especially when crossing organizational boundaries, were identified as critical areas of focus. The seminar highlighted the need for formal contracts to manage the intricate bindings and dynamic interactions between various entities. The role of a federation engine was emphasized as a tool for translating requirements, ensuring compliance, and resolving disputes, thereby ensuring the workflow's needs are met at each federation point.

Moreover, the seminar identified key challenges and opportunities at the intersection of these technologies, such as the stochastic nature of ML and its impact on the reproducibility of data analysis on HPC systems. It highlighted the need for holistic co-design approaches, where workflows are introduced early and scaled from small-scale experiments to large-scale

executions. This approach is essential for integrating the "full" workflow environment, including ML/AI components, early in the process, thereby replacing expensive simulation with fast-running surrogates and enabling interactive exploration with the entire software environment.

In summary, the Dagstuhl Seminar 23352 provided an in-depth exploration of the synergistic relationship between HPC, AI, and scientific workflows. It paved the way for future research directions and practical implementations, aiming to revolutionize scientific data analysis by harmonizing computational power with intelligent, data-driven analysis. The discussions and outcomes of the seminar are poised to influence the development of workflow systems and technologies in the years to come, signaling a shift towards more integrated, adaptive, and efficient scientific computing paradigms.

## 2 Table of Contents

# 3 Challenges

## 3.1 Workflow Dynamics and Management

*Rosa M. Badia (Barcelona Supercomputing Center, ES)*
*Silvina Caino-Lores (INRIA – Rennes, FR)*
*Kyle Chard (University of Chicago, US)*
*Wolfgang E. Nagel (TU Dresden, DE)*
*Fred Suter (Oak Ridge National Laboratory, US)*
*Domenico Talia (University of Calabria, IT)*



A broad definition of an integrated AI+HPC workflow is that of a workflow in which at least one HPC task (i.e., an HPC simulation) and at least one AI task (e.g., a surrogate model) coexist in the same workflow application. We also consider that such workflows can run on HPC systems, although not necessarily all their components are executed on HPC systems.

We can then refine this definition according to how the AI and HPC components are effectively coupled, ordered, and placed within the workflow. The first coupling mode is when an AI model is used to replace a computationally intensive component, or the whole HPC simulation itself, of the workflow (AI-in-HPC). In this scenario, only the inference part of the model is part of the workflow, the training of the model being done offline. The second coupling mode captures scenarios where AI is used to steer the HPC components or generate new data or parameterization. The AI model thus resides "outside" of the main HPC simulation (AI-out-HPC). Here both training and inference can be part of the AI+HPC workflow, e.g., when using reinforcement learning techniques. The last coupling mode is when the AI models are concurrent and coupled to the main HPC tasks and run synergistically with simulations (AI-about-HPC). For instance, AI models can be used to optimize the performance of the HPC runtime system/workflow manager/resource manager/scheduler. Results of the HPC component are thus used to train the AI component as it runs and allow for system-wide predictions and optimization. Note that these three coupling modes are not mutually exclusive and are commonly combined within the same workflow.

### 3.1.1 Dynamic behavior in AI+HPC workflows

Most managers still treat static programming and execution. Expressing and handling dynamicity will become critical for AI+HPC workflows.

1. Adding components. Users might want to add or interact with components of the workflow. How to address this from a resource allocation and management perspective?
2. Alternative path in the workflow (branching). Workflows can have a traditional HPC component and a surrogate model. These options have different computational complexities and might be triggered by different conditions (e.g., low model accuracy). Dynamic resource allocation is needed in order to run the HPC simulation, for example.
3. Feedback loop (i.e., steering, control loops). Ability to manage detection of errors, thresholds, hooks in general, and change the behavior of the workflow accordingly. This can be due to the behavior of the AI model (bias, overfitting, etc.). For example, monitoring malicious (or not malicious, just because of data corruption) behavior in federated learning workflow, which are becoming increasingly present in HPC (e.g., Integrated circuit model building at the advanced photon source, biomedical workflows across Argonne Leadership Class Facility and Broad with APPFL).

4. Fault management, exception management. Ability to manage faults at the task level (software or hardware, division by error, etc) and change the behavior of the workflow accordingly.
5. Other things to consider: security, privacy, provenance, etc.

Although points 3 and 4 are conceptually different (the former is motivated by application functionality, the latter by application errors or runtime state), underlying mechanics in the runtime might be the same to manage these events. These mechanics might not necessarily be exposed to the same degree to the end user/application.

An additional problem can be how to detect these errors/faults. The workflow manager should provide this functionality.

### 3.1.2 Resource Management

One of the main activities performed by Workflow Management Systems is resource management, taking into account that the term resources include hardware infrastructure, software components and data.

We consider the following set of stages as the more common set for the workflow lifecycle under current workflow management systems:

- *Stage 1:* Preparation of environment: performs the software dependency deployment (i.e., using containers).
- *Stage 2:* Data and artifacts discovery/staging in. With artifacts we consider, for example, a pre-processed dataset, a trained AI model, etc. This can be for example performed using Apache airflow data pipelines that define the data movements that are needed before the actual computing execution.
- *Stage 3:* Compute resource provisioning. Current practices do not include elements of I/O scheduling and allocation that favor AI workflow executions. Other extensions that can be considered for improving convergence with AI are exploiting locality and reusing data blocks (for example reusing models for inference tasks), both on disk and in memory, but currently there is insufficient locality exposure to other elements in the system. Other ideas that can be leveraged in HPC come from the cloud, like expressing data affinity.
- *Stage 4:* Deployment of the pilot job(s). The dynamicity of the pilot shapes/types to match dynamic workload (e.g., MPI, single-core, GPUs, etc.).
- *Stage 5:* Execution of the actual computational workflows.
- *Stage 6:* Data and artifacts staging out.
- *Stage 7:* Cleanup.

The order of the stages might depend on the actual solution (e.g. using Conda environments, stage 1 is executed after stage 3), or some stages will not exist in some cases (i.e., stage 4 maybe is not needed in some cases).

We consider the following approaches are applied in workflows' resource management:

1. Use of AI to inform the scheduler: AI can be used in multiple forms to assist the resource management. We are listing a few below:
   - To estimate the required resources and run time and manage faults.
   - To learn from previous executions.
   - To perform uncertainty management (knowing that user-provided estimates are inaccurate, mainly for walltime, memory and, storage).

   The challenges arise on how to acquire, clean and assimilate the data necessary to train these models. This includes, Identifying available data sources: user knowledge, scheduler logs, application/workflow monitoring, hardware monitoring (e.g., bluefish, starfish, etc.).

2. Use of elasticity, heterogeneity and performance of AI in HPC systems. Optimally, workflow management systems should be able to request heterogeneous resources to suit the AI and HPC workload on demand (not just the amount of resources but the type of resource that will provide the desired level of performance for a specific task). Currently, some systems implement workarounds to provide elasticity within a static allocation (supported with Slurm and LSF, for example). Existing AI libraries may require reengineering to improve their performance and unlock high scalability on HPC environments (this might involve revisiting their interaction with the HW, I/O system, etc.). For example, some of this is happening e.g., in the DOE world for Intel and AMD GPUs. Also, vendors are working on this for specific hardware (e.g., cerebras, sambanova, graphcore, etc.).

### 3.1.3 Integration of Different Platforms and Systems

AI motivates the need for further integration of systems:

1. The integration of HPC, which typically operates on a batch-queue basis, with cloud technologies like Kubernetes (K8s), presents a unique challenge in workflow management. A work in progress approach is currently being explored, aiming to delegate parts of workflows to existing managers within each facility, such as Zambeze and Fluence, without necessitating oversight of the entire computation. This raises a pertinent question about the need to reevaluate and possibly redimension storage services in HPC facilities to effectively accommodate the increasing demands of AI workloads.
2. The management and locality of data in distributed or cross-facility workflows present significant challenges, particularly in terms of authentication, authorization, and policy adherence. Initiatives like EU's Gaia-X and Fenix, as well as the US's OneID, are making strides in addressing these issues. However, policy constraints often emerge as the most limiting factor in these contexts, impacting how data is accessed and shared across different facilities.
3. Also workflow management (see example of federated learning).
4. In the Edge-to-HPC paradigm, a key strategy is to position tasks, such as training, close to the data source at the edge, optimizing the use of edge nodes and HPC nodes. This involves adjusting the granularity of tasks and the size of data in relation to the computing capabilities of these nodes. A critical aspect of this approach is investigating the trade-offs between time-to-prediction and accuracy, especially when implementing compression and filtering techniques. These methods can impact the accuracy of AI models but are beneficial in reducing the data transfer and processing load. This approach is not only applicable to AI models but can also extend to other data generation stages, like HPC simulations, and is particularly relevant in scenarios requiring urgent computing and real-time steering, where efficiency and response time are crucial.
5. The integration of programming models and environments for workflows that combine HPC and AI presents a unique challenge, given the significant differences between HPC and AI programming environments. To bridge this gap, there are ongoing efforts such as PyCOMPSs [1], which is based on Python, aiming to harmonize these distinct environments. This initiative represents a step towards creating a more unified and efficient programming model that can cater to the diverse requirements of both HPC and AI, facilitating smoother integration and more effective workflow management in these complex computational domains.

In conclusion, while leveraging AI for scheduling in HPC systems promises efficiency gains, it also raises the critical question of its impact on energy consumption. The trade-off between enhanced scheduling performance and increased energy demands necessitates careful consideration. The path forward lies in steering AI towards green computing, where AI not only optimizes computational tasks for performance but also aligns with energy-efficient practices. This approach requires a delicate balance, ensuring that the benefits of AI in HPC do not inadvertently escalate energy consumption, but rather contribute to a more sustainable and environmentally conscious computing paradigm.

### References

1 Natalia Poiata, Javier Conejero, R Badia, and Jean-Pierre Vilotte. Data-streaming workflow for seismic source location with pycompss parallel computational framework. *EGU General Assembly*, 10, 2021.

## 3.2 Sustainability Concerns

*Laure Berti-Equille (IRD – Montpellier, FR)*
*Timo Kehrer (Universität Bern, CH)*
*Christine Kirkpatrick (San Diego Supercomputer Center, US)*
*Dejan Milojicic (HP Labs, US)*
*Sean R. Wilkinson (Oak Ridge National Laboratory, US)*



In this section, we delve into the crucial questions raised during our brainstorming sessions focusing on the integration of sustainability in HPC and AI-driven scientific workflows. These discussions centered around three main queries: (1) how to cultivate sustainability awareness, (2) the categorization of key sustainability challenges specific to HPC+AI workflows, and (3) pinpointing where these challenges predominantly arise in the workflow lifecycle. To tackle the first question on building sustainability awareness, we drew insights from existing initiatives such as the Sustainability Awareness Framework (SusAF) [1], adopting a multi-faceted approach that encompasses environmental, economic, social, and technical dimensions of sustainability [2]. Addressing the second question, Fig. 1 provides a structured categorization of these sustainability challenges, recognizing that the list is not exhaustive and acknowledging the existence of cross-cutting challenges that span multiple dimensions.

For example, among cross-cutting challenges, **sustainability awareness** is a central challenge as people (social dimension) don't always understand how to measure the energy they're consuming (this is especially difficult on a shared system) (technical dimension) and also the carbon footprint of their workflows (environmental dimension) [3]. This is also difficult to measure if the amounts (energy, CO2) are not even exposed somehow.

The **re-usability** challenge also belongs to both technical and social as for the latter, how to convince potential users to re-use (parts of) existing workflows and actually make this re-use technically possible and easy for various kinds of users (DevOps, domain scientist, data scientist, etc.). Another example of central challenges, at the governance level and at the convergence of social, technical, economic and environmental dimensions of sustainability is the **deployment of sustainability-rated multi-objective policies** where some priorities can be defined adaptively depending of the application context and needs to favor either environmental, social or economic considerations for example. As it's commonly the case

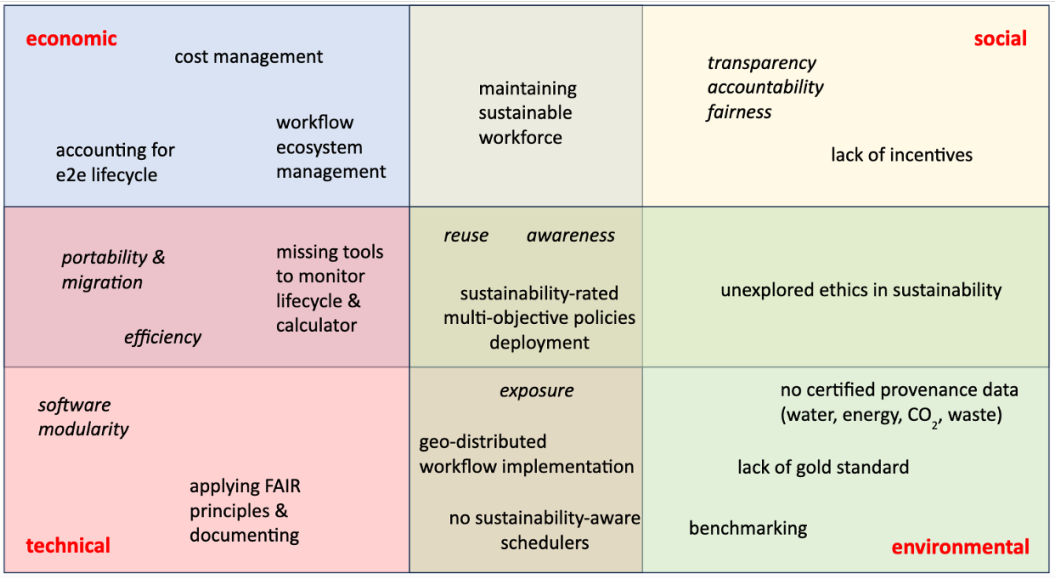


**Figure 1** Categorized sustainability challenges in HPC+AI workflows.

in many sustainability studies since the UN 2030 agenda and its SDGs [4], we can also distinguish between high-level objectives and activities which need to be improved to reach the objectives since both are challenging per se. In the figure, we note in italic the objectives whereas the activities are in normal font.

### 3.2.1 Main Sustainability Challenges

- **Environmental Sustainability.** The main challenge we identified under the environmental sustainability dimension is how to get **certified provenance data** to quantify or estimate the impact on the environment and natural resources, i.e., provenance data about CO2 emissions, energy or water consumption from authoritative sources [5]. The lack of traceability metadata makes it very difficult to know, for example, where the electricity comes from and distinguish between renewable vs non-renewable energy, clean or green energy powering some given HPC+AI workflows [6]. Similarly, reliable and continuously up-to-dated data about water-usage, gas emission or waste management is currently missing, not detailed enough, or not trustworthy and we cannot drill-down and quantify the impact of a single workflow on the environment (or drill-up for a set of workflows of a given application) [7]. As a consequence, there is a **lack of benchmark** and a **lack of gold standard** that could be used as a reference for virtuous practice. More generally, this leads to a lack of **exposure** and **awareness** regarding environmental impacts of the workflow lifecycle.
- **Social Sustainability.** A central challenge is the **lack of incentives** individual researchers, service providers, organizations, and funders have for nurturing sustainability. This is especially true for researchers, who are incentivized to experiment, analyze, and publish results. Nowhere in the academic process that relates to tenure and promotion are sustainability concerns accounted for, such as making computing choices that are economically and environmentally sustainable. Similarly, funders have little incentive to promote sustainability as they do not directly suffer the consequences of poor economic or environmental choices. Another challenge in the social realm is making choices that

**maintain a sustainable workforce**. For example, a researcher has no incentive to use the workflow package or machine learning (ML) software with the largest market share or that is most widely used and supported at a computing center. If more researchers stayed with market share and understood platforms, it is easier on research computing staff to support fewer packages, as well as software that is maintained with security patches and bug fixes. The incentives for such dimensions are with service providers, who have few ways to motivate researchers to factor other variables into their choice of research computing components. One way to promote social sustainability values, including **transparency, accountability, and fairness**, is to embed these principles in shared community values. The US National Science Foundation (NSF) funded EarthCube initiative brought together geoscientists and cyberinfrastructure builders to build innovative tools and invigorate the community with advanced computing techniques. The EarthCube community, led by leaders elected by the membership and a funded coordination office, wished to imbue the context of the collective work with shared principles. The principles included Responsibility, Dependability, Service, Openness-Transparency (among others). By defining the shared values, EarthCube was able to tune strategic actions to model and encourage these seemingly intrinsic qualities. This led to impactful work as shown in the recently released EarthCube retrospective [8].

- The **Technical sustainability** of HPC+AI workflows can be considered in (at least) three ways: (1) that of the HPC system itself, (2) that of the AI models used as tools and constructed by the workflow, and (3) that of the workflow itself when considered as research software. Because current HPC systems are short-lived and highly specialized, HPC workflows are usually tailored specifically to the systems where they will run, and this causes problems in workflow portability (running a workflow on another current HPC system) and workflow migration (running a workflow on a future HPC system).
- The **Economic Sustainability** relates to cost management of HPC+AIworkflows, as well as the entire workflow ecosystem management, accounting for the end-to-end lifecycle of the workflows. Some of the above challenges can be dependent (correlated or anti-correlated). Therefore, corrective actions may have various indirect positive or negative impacts beyond what we can expect and there is not yet a principled way of estimating and predicting the collateral effects of improving one dimension or one objective over the others. For example, if we decrease the energy consumption of a workflow the overall cost can however be increasing; loose-coupling of softwares can have a positive effect and facilitate reuse, but may also increase the computation cost [9]. Nevertheless, we should not compromise in making research progress.

### 3.2.2 Sustainability Challenges Across the Workflow Lifecycle

Finally, we attempt to address the third question "Can we pinpoint where the sustainability challenges are predominant in the workflow lifecycle?" We decompose the workflow lifecycle into several stages and consider the human, data, and AI planes (Fig. 2) [10].

Although it is not represented as a cycle, the workflow lifecycle is iterative and can have multiple feedback loops. The data processing block can be decomposed into multiple blocks (data acquisition/collection, storage, preprocessing, analytics, use, distribution, archival), and again not necessarily executed in a sequential manner and some blocks (such as archival) may not be present in some workflows. Similarly, the ML model development block can be decomposed into various blocks such as feature engineering, model building, training, tuning, validation and exec monitoring. Data and AI planes are predominantly supported by HPC and AI softwares respectively. Our exercise consisted in mapping and locating the challenges

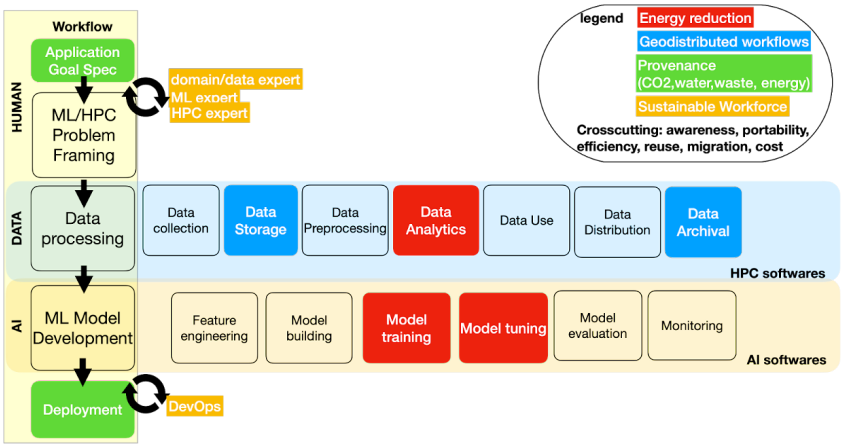


**Figure 2** Quasi reference model for sustainability in HPC+AI workflows.

we identified previously into the generic workflow lifecycle because all the challenges may not be not occurring at every stage and when they occur they may not be predominant or have a critical impact on sustainability. Empirical experiments will surely be needed to verify our assumptions here.

For example, energy consumption could be reduced essentially targeting the data analytics stage with more frugal methods, in particular during model training and tuning stages. Ensuring that the workforce (such as domain and data scientists as well as Devops engineers) is sustainable is important during the early stage of the application domain and goal specifications for designing the adequate workflow and framing the ML on HPC problem and also during the deployment as well as continuously gathering reliable data characterizing the environmental impact of the workflow (energy, gas emission, etc.). Reallocating computing resources or data storage (vicinity) can save energy and reduce technical or environmental costs related to computation, data storage and archival.

A follow-up of this preliminary discussion can be to identify the leverages in the workflow lifecycle where a small improvement of some targeted tasks can have a huge positive impact on the sustainability dimensions. Next, we could define Whatif scenarios and experiments to simulate the gain/loss in terms of sustainability and support sustainability-rated multi-objective policies.

### 3.2.3 Use Cases

The following are some use cases that drive sustainability concerns:

- *Physical limits:* some sites cannot bring more power than they are designed for, therefore it is required to do power-throttling and minimizing power consumption to conduct computation. Both this and the next use case mean doing more (computation) with less (energy)
- Economically-driven: energy costs a lot of money, running an exascale computer costs even more than buying it. Therefore it is required to be mindful when leveraging these computers to get most out of large scale computations.
- Save the planet: many executives are making pledges to make their corporations net-zero or even net-positives. It is non-trivial to achieve this and in many cases it means payments towards clean energy. It entails both upstream and downstream explorations.
- Broader good: there is a public pressure for less consumption and for clean energy use. HPC and AI computers consume lots of energy, so being mindful to use clean energy most of the time is one way to alleviate the problem.

## References


1 Sustainability Awareness Framework (SusAF). https://www.suso.academy/en/sustainability-awareness-framework-susaf/, 2023.
2 Cullen Bash, Kirk Bresniker, Paolo Faraboschi, Tiffani Jarnigan, Dejan Milojicic, and Pam Wood. Ethics in sustainability. *IEEE Design & Test*, 2023.
3 Baolin Li, Siddharth Samsi, Vijay Gadepally, and Devesh Tiwari. Sustainable hpc: Modeling, characterization, and implications of carbon footprint in modern hpc systems. *arXiv preprint arXiv:2306.13177*, 2023.
4 Transforming our world: the 2030 Agenda for Sustainable Development. https://sdgs.un.org/2030agenda, 2023.
5 Sadasivan Shankar and Albert Reuther. Trends in energy estimates for computing in ai/machine learning accelerators, supercomputers, and compute-intensive applications. In *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–8. IEEE, 2022.
6 Dan Zhao, Nathan C Frey, Joseph McDonald, Matthew Hubbell, David Bestor, Michael Jones, Andrew Prout, Vijay Gadepally, and Siddharth Samsi. A green (er) world for ai. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 742–750. IEEE, 2022.
7 Sirui Qi, Dejan Milojicic, Cullen Bash, and Sudeep Pasricha. Shield: Sustainable hybrid evolutionary learning framework for carbon, wastewater, and energy-aware data center management. *arXiv preprint arXiv:2308.13086*, 2023.
8 EarthCube: A Retrospective 2011-2022. https://library.ucsd.edu/dc/object/bb0180794q, 2023.
9 Amir Nassereldine, Safaa Diab, Mohammed Baydoun, Kenneth Leach, Maxim Alt, Dejan Milojicic, and Izzat El Hajj. Predicting the performance-cost trade-off of applications across multiple systems. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, pages 216–228. IEEE, 2023.
10 Nicolas Dube, Duncan Roweth, Paolo Faraboschi, and Dejan Milojicic. Future of hpc: The internet of workflows. *IEEE Internet Computing*, 25(5):26–34, 2021.

## 3.3 Integration and Standardization

*Ilkay Altintas (San Diego Supercomputer Center, US)*
*Rosa Filgueira (University of St Andrews, GB)*
*Ana Gainaru (Oak Ridge National Laboratory, US)*
*Shantenu Jha (Rutgers University, US & Brookhaven National Laboratory, US)*
*Ulf Leser (HU Berlin, DE)*
*Bertram Ludäscher (University of Illinois at Urbana-Champaign, US)*
*Jeyan Thiyagalingam (Rutherford Appleton Lab., GB)*



### 3.3.1 Reference Architectures From a Workflow Perspective

To date, no established reference architecture for scientific workflow systems exists. Furthermore, there neither exists a characterization of the precise functionalities a workflow system should encompass nor where it should interface (and with which API) with other components of a distributed infrastructure. The general components of a distributed infrastructure steered by a workflow system are depicted in the idealized architecture shown in Fig. 3. From top to bottom, these include:

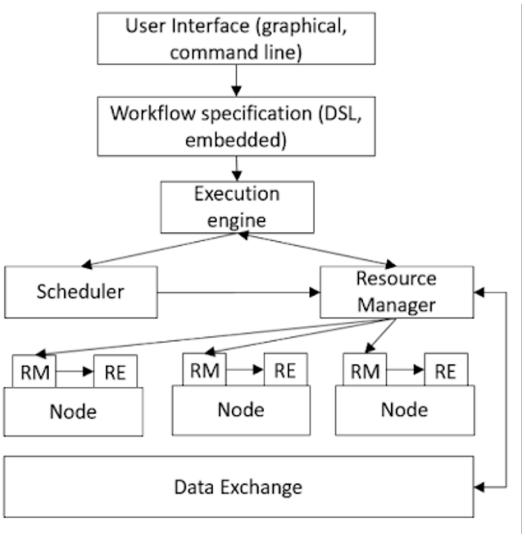


**Figure 3** Main components of a typical distributed infrastructure steered by a workflow system.

- **User interface.** While many systems target developers and favor command line interfaces [1], others offer comprehensive graphical interfaces using the DAG-structure of workflows as metaphor [2]. Graphical interfaces often are also associated with access to libraries of available workflow tasks or control structures of a workflow language [3].
- **Workflow specification.** Many systems use specific domain specific languages for specifying a workflow, which might resemble programming languages [4] or come in the form of flat file formats [5]. Other systems offer workflow functionality as extensions to a host programming language without having a proper syntax [TBA+17]. Workflow specifications describe an abstract workflow; during execution, tasks defined abstract workflow often lead to multiple physical instances.

- **Workflow engine.** The workflow specification (or workflow program) is executed by a workflow engine, whose main purpose is the control of the dependencies between workflow tasks. As such, the workflow engine, at every stage of a workflow execution, must be able to determine the set of currently executable tasks, which requires some form of bidirectional communication with the scheduler to be informed about finished tasks. At this stage, typically also the compilation of the abstract workflow into a physical one is performed. In dynamic workflow systems, where the set and structure of tasks is data dependent, the communication must include further aspects, such as number of files in a directory (for scatter operations) or intermediate data files themselves (for conditionals) [6].
- **Scheduler.** The scheduler is informed by the workflow engine about the set of ready -to-run tasks and determines their assignment to the set of available (virtual) compute nodes. To this end, it communicates with a resource manager to obtain free resources and to access the characteristics of free nodes (e.g., main memory, accelerators etc.). Schedulers typically are not individual components, but their functionality instead is included in that of another system (see below).
- **Resource Managers and Runtime Environment.** Resource managers have two different purposes (RM). At the global level, the RM oversees the status of all registered nodes together with their functional characteristics and offers controlled access to them for its clients, possibly with an associated QoS. At the local level, each node runs an instance of the RM to control the node itself and to provide a local runtime environment (RE) for workflow tasks (such as Docker or Singularity containers).
- **Data exchange.** Finally, the tasks of a workflow execution must exchange data along their data dependencies. There are different ways how this can be achieved. In a streaming setting, all data exchange is handled through the network [13]. Batch-processing often assumes availability of a shared data space, for instance provided by a parallel file system like Lustre [21] or CePH [7].

However, concrete systems often deviate heavily from this architecture. For instance, scheduling usually is not implemented as a separate component, but instead is performed inside the workflow engine or by the resource manager – and sometimes by both [18]. Systems with graphical user interfaces might not have an explicit language or format to express a workflow specification but instead directly interpret the workflow graph. Workflow engines are typically tightly coupled to a workflow specification language and not capable of executing any other specifications; moving from one system to another therefore requires program translation. Systems trying to co-optimize task placement and data locality need more expressive interfaces to resource managers, schedulers and file systems and explicit ways of manipulating placement [14]. Furthermore, there is no agreement on the interfaces between components, such as between workflow engines, schedulers, and resource managers or between a graphical user interface and a workflow engine. Even at interfaces where standards exists, such as POSIX for file access or DRMAA for resource managers, these are not implemented by all systems. For instance, HFDS is not Posix compliant, and Kubernetes does not support DRMAA.

### 3.3.2 Role of AI in workflows and HPC

The role of AI in scientific workflows and HPC is increasingly prominent and multifaceted. Here is a summary of the role of AI in these both of domains:

1. Enhancing Scientific Workflows:
   - *Workflow Optimization:* AI techniques, such as reinforcement learning [19, 40], can optimize the execution of complex scientific workflows.
   - *Discovering Similar Workflows:* Semantic code search powered by AI can assist researchers in finding workflows that are functionally or structurally similar to their own. Semantic techniques [31, 30] can be especially beneficial when looking for existing solutions to similar scientific problems, thus promoting knowledge sharing and collaboration.
   - *Identifying Workflow Components:* Researchers can use AI and LLMs models to perform semantic code searches [38] and identify specific components or tasks within workflows that perform certain tasks.. This enables reusing and adapting existing components to build new workflows more efficiently.
   - *Automated Documentation:* AI and LLMs can generate automated documentation for scientific workflows [38]. This documentation enhances the understanding of the workflow, and also fosters reproducibility, making it easier for researchers to share and replicate their work.
2. Data-Driven HPC-workflows:
   - *Data Analysis:* AI and machine learning enable sophisticated data analysis within scientific workflows, extracting valuable insights from large datasets [16, 26].
   - *Pattern Recognition:* AI [34] can identify patterns and correlations in scientific data, aiding researchers in discovering hidden relationships and making data-driven decisions.
   - *Real-time Insights:* In HPC simulations [24, 22], AI [27] can provide real-time insights, facilitating adaptive simulations based on changing conditions. This is especially relevant in fields like seismology, climatology, etc, where quick responses are critical.
3. HPC and AI Synergy:
   - *Resource Allocation:* AI optimizes resource utilization in HPC clusters by dynamically [8] allocating computing resources based on workload demands or previous runs [25]. This leads to cost savings and improved efficiency in resource usage.
   - *Reducing Energy Consumption for Exascale:* One of the critical challenges in exascale computing is the immense energy consumption of supercomputers. AI contributes to energy-efficient [29] computing by optimizing cooling systems, and power usage. Machine learning models can predict workload patterns and dynamically adjust power consumption to match computing demands, significantly reducing energy waste.
   - *Enhancing Fault Tolerance:* HPC systems are prone to hardware failures and errors due to their sheer complexity. AI-driven [23] fault tolerance mechanisms can detect anomalies in real-time and initiate corrective actions. Machine learning models can predict hardware failures before they occur, allowing for proactive maintenance and minimizing downtime.
   - *Predictive Analytics:* AI models can diagnose [32] and predict IO bottlenecks [37], data transfer issues, or compute node failures [DA+18] in advance, allowing for proactive mitigation and improved workflow efficiency.
   - *Deep Learning:* HPC infrastructures are crucial for training deep learning models [11, 20] on vast datasets. The synergy allows scientists to use AI for tasks like image recognition, natural language processing, and simulation optimization.

In summary, AI plays a vital role in scientific workflows and HPC by optimizing processes, extracting insights from data, and fostering synergy between these domains, ultimately advancing research and innovation in numerous fields.

### 3.3.3 AI within the Reference Architectures: Issues and Challenges

AI can play an important role for many components of a workflow system running on HPC. However, integrating such functionality also faces a number of challenges for which no good solutions exist today. These are:

- **Scheduling** in many systems is not an operation performed once at a single component, but instead often is accomplished in an iterative, hierarchical fashion. For instance, in a dynamic workflow at every point in time when a scheduler needs to take a decision, only the immediate next steps are known, while further downstream tasks are not. Decisions considering long-range implications, which are typical for AI-based schedulers, are thus impossible [15]. Another example for PilotJobs, which are scheduled by the resource managers as single tasks, but which during execution actually are expanded into multiple tasks which then need to be scheduled by the PilotJob itself [33]. Such two level scheduling currently is neither supported by AI-based solutions nor adequately reflected in the reference architectures.
- **Resource predictions.** Many advanced solutions for scheduling and resource management rely on precise predictions for the resource the execution of a task will require, such as runtime, memory, bandwidth, or energy. While many methods for performing such predictions are currently developed [35, 12], from an architectural point of view it is not clear where this functionality should be placed. The methods often assume access to current or past log files, which requires a positioning deep in the stack; on the other hand, their results are required by the workflow engine, the scheduler, and the resource manager. In pure online prediction systems, which try to predict resource requirements only based on the currently run workflow, there exists a dependency between predictions and scheduling, as the scheduler must take into account for which tasks and at which accuracy predictions are possible, for instance to prefer tasks for which this is not possible yet [36]. A further dependency that is missing in the reference architectures exists between resource predictions and resource managers, as a prediction regarding, for instance, main memory, are an important input for right-sizing of containers and virtual machines.
- **Metrics.** Our discussion so far focused much on conventional features of workflow/-workload execution, such as runtime and resource requirements. However, AI-based workflows also bring entirely new metrics that must be taken into account. Two particularly important ones are accuracy and transparency. Accuracy describes the quality of the result produced by an AI-based workflow. From a user perspective, optimizing accuracy actually might be more important than optimizing runtime, but requires entirely different means of user support [17]. Transparency describes the property of a workflow answer to be explainable from a user perspective, and is an important cornerstone of trust in data analysis results. Again, optimizing for transparency calls for different actions than optimizing resource demands.
- **Cloud-based architectures.** Our reference architecture does not consider the typical properties of cloud infrastructures. For instance, elasticity, i.e., a growing and shrinking of the available compute nodes during workflow execution, is not considered, but could be ideally combined with AI based workload predictions. It would require that tasks during their execution can acquire more computational resources, which breaks the hierarchical nature of the architecture sketch in Fig. 3. In cloud environments, function-as-a-service has become popular recently (also in workflow systems [28]) as a means to provide serverless and thus easier to maintain workflow executions. Such approaches require an adaptation of the reference architectures, as not anymore discrete tasks are the basic unit of operations, but instead asynchronous function calls, which requires a different understanding of resource management and scheduling.

- **Changing user groups.** The recent "democratization of data science[1]" results in a drastic change in the types of users that workflow and HPC systems must support. IN a nutshell, the user base grows enormously, while at the same time the typical technical capabilities and resources associated to a user or user project shrinks. This calls for new and easier to use interfaces (e.g., graphical metaphors, integration with research data management, improved result visualization, interactive and human-in-the-loop interfaces etc.) and expanded user support (e.g., other means of workflow design and adaptations, workflow testing, and workflow debugging; personalized assistance; improved and context-dependent documentation etc.). AI can play an important role here to make interfaces personalized and more context-dependent.

### 3.3.4 The Role of Benchmarking for Workflows, AI and HPC

Given the broader role of AI, HPC and workflows on data analysis, it is almost difficult to ascertain the suitability of a workflow engine, or workflow, or AI or HPC technique(s) for a given data analysis task. This becomes further complicated if performance (either runtime or scientific task performance) becomes a qualifying metric for the final decision around a particular AI technique or workflow or choice of a workflow engine. This complexity entails a need for a mechanism that can aid scientists (or stakeholders) in making such a decision. Benchmarking has been the cornerstone of solving such issues since the inception of software systems. As such, it is conceivable that a mechanism akin to benchmarking would be ideal to understand the interplay between these aspects.

Although one can resort to simply benchmark a workflow (or any aspect of interest), in the absence of a well-defined or well-established basis, such efforts would become meaningless. One option is to establish a common set of benchmarks that would capture a number of realistic mix of cases of different data analysis problems. A set of applications (whether realistic or synthetic) make up a benchmark suite, and are very specific to the case in hand, and in our case, workflows.

The notion of benchmarking for scientific workflows is not a novel concept, and in fact, can be widely found in the literature [9]. However, a benchmark suite that captures the integration of AI and HPC for data analysis workflows, especially in the context of recent developments in AI (such as LLMs), adds additional complexities. This is further complicated with the recent developments around detector rates, capability of modern facilities (such as AI at the edge), and modern AI-specific architectures. However, given the diverse range of scientific applications, it would be a monumental effort if the suite were to provide a full coverage of all application cases. Instead, one can envisage building a benchmark suite based on application classes or beamline types.

Identifying such a class or themes of benchmarks is not only useful for quantifying the performance capabilities of different workflow solutions, but also very instrumental in for understanding and assessing the functional capabilities of different workflow engines.

### 3.3.5 Benchmarking Techniques

In this section, we delve into the intricacies of benchmarking within Research Agendas (RAs), particularly focusing on workflow engines and their deployment across various infrastructures. The benchmarking process in this context is multifaceted, encompassing every component of a RA. This includes conducting integration tests, which are crucial for ensuring that different components of a workflow engine function cohesively.

[1] https://hbr.org/2018/07/the-democratization-of-data-science

One effective approach is to run multiple workflows within the same workflow engine on different infrastructures. This method is not only beneficial for testing the robustness and scalability of the system but also helps in identifying potential bottlenecks and optimization points. Such tests are exemplified by initiatives like nf-core [10], which demonstrate the practical application of these benchmarking strategies. However, the challenge arises when attempting to implement the same problem across multiple workflow engines, each in their respective language, and on different or similar infrastructures. This approach is invaluable for comparative analysis, providing insights into the relative strengths and weaknesses of various systems.

Despite its benefits, this methodology is not without its drawbacks. The primary challenge lies in the significant effort and resources required to implement and manage these benchmarks. Setting up the same problem across different workflow engines involves considerable time and expertise, particularly in adapting the problem to the nuances of each engine's language and infrastructure compatibility. Moreover, the need to run these benchmarks on various infrastructures adds another layer of complexity, requiring careful planning and coordination to ensure accurate and meaningful comparisons. Thus, while these benchmarking techniques offer substantial benefits in evaluating and improving research agendas, they also demand a considerable investment in terms of effort and resources.

### 3.3.6 Benchmarking Targets

In this section, we explore the diverse aspects and environments where workflow benchmarking is applied, emphasizing the need for a comprehensive approach to adequately assess and improve these systems. The benchmarking targets are multifaceted, ranging from the nature of the workflows (stream, batch, task-based) to the underlying data handling mechanisms (main memory versus file-based). Each of these aspects presents unique challenges and opportunities for optimization, making them critical targets for benchmarking efforts.

One of the key considerations in workflow benchmarking is the approach to data processing, with distinctions between synchronous and asynchronous workflows. Synchronous workflows, where tasks are executed in a predetermined order, and asynchronous workflows, which allow tasks to run independently and often concurrently, each have their own performance characteristics and optimization needs. The benchmarking process also needs to account for the diversity in programming languages used in workflow systems. Multi-language support is essential to cater to the varied requirements and preferences of different user groups. Additionally, the infrastructure on which these workflows are executed plays a crucial role. This includes the type of resource manager, the availability and use of GPUs, and the size and configuration of clusters. Benchmarking must therefore encompass a wide range of infrastructure setups to ensure comprehensive evaluation and optimization.

Another important target for benchmarking is the expressiveness of the languages used to define workflows. This includes evaluating how well a language or system supports complex constructs like conditionals and recursion. The ability of a workflow system to handle these elements effectively can significantly impact its usability and efficiency, making it a crucial aspect of benchmarking. By focusing on these varied targets, workflow benchmarking can provide critical insights into the performance and capabilities of these systems, guiding improvements and ensuring they meet the diverse needs of their users.

### 3.3.7 Issues with Benchmarking

In this section, we address the complexities and challenges that arise in accurately evaluating workflow performance. A primary concern is identifying and scrutinizing the performance-critical parts of workflow execution. This includes assessing the efficiency of various components such as the scheduler, the methods used for dependency resolution, and the workflow interpreter, as well as the quality of the individual task implementations. Each of these elements can significantly impact the overall performance of a workflow, making their thorough assessment essential for a comprehensive understanding of the system's capabilities.

Another pivotal issue in workflow benchmarking is the choice between real measurements and simulations. Real measurements offer tangible data on system performance under actual conditions, but they may have limitations in terms of scalability and broader applicability. In contrast, simulations, like those executed using WorkflowSim on platforms such as CloudSim and NetSim, are invaluable for scalability tests and can provide insights that are not feasible in real-world settings. However, simulations might not capture all the intricacies of real-world performance. The challenge is to balance the insights gained from simulations with the practicalities of real-world measurements, especially considering the risk of overfitting in benchmarks. This issue is evident in benchmarks like Linpack for high-performance computing systems, where optimization often focuses more on the benchmark than on practical applications.

Furthermore, the diversity of applications complicates the process of workflow benchmarking. Workflows vary greatly, from being heavily reliant on AI, to focusing on streaming data, to being based on complex simulations, or even being hybrids of these types. Each category requires specific approaches to benchmarking to accurately assess performance and efficiency. Therefore, there's a growing need for a benchmarking standard that can adapt to these diverse requirements, providing a balance between broad applicability and specific, actionable insights. This standard differs from internal benchmarks used for tuning or procurement, which often focus on optimizing specific aspects of a system's performance. Developing effective benchmarking strategies that can navigate these issues is critical for guiding the optimization and development of workflow systems in a way that is both efficient and applicable to a wide range of real-world scenarios.

## References


1 Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B Hall, Christopher H Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O Twardziok, Alexander Kanitz, et al. Sustainable data analysis with snakemake. *F1000Research*, 10, 2021.

2 Jeremy Goecks, Anton Nekrutenko, James Taylor, and Galaxy Team team@ galaxyproject. org. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 11:1–13, 2010.

3 Paolo Missier, Stian Soiland-Reyes, Stuart Owen, Wei Tan, Alexandra Nenadic, Ian Dunlop, Alan Williams, Tom Oinn, and Carole Goble. Taverna, reloaded. In *Scientific and Statistical Database Management: 22nd International Conference, SSDBM 2010, Heidelberg, Germany, June 30–July 2, 2010. Proceedings 22*, pages 471–481. Springer, 2010.

4 Paolo Di Tommaso, Maria Chatzou, Evan W Floden, Pablo Prieto Barja, Emilio Palumbo, and Cedric Notredame. Nextflow enables reproducible computational workflows. *Nature biotechnology*, 35(4):316–319, 2017.

5 Peter Couvares, Tevfik Kosar, Alain Roy, Jeff Weber, and Kent Wenger. Workflow management in condor. *Workflows for e-Science: Scientific Workflows for Grids*, pages 357–375, 2007.

6 Marc Bux, Jörgen Brandt, Carl Witt, Jim Dowling, and Ulf Leser. Hi-way: Execution of scientific workflows on hadoop yarn. In *20th International Conference on Extending Database Technology, EDBT 2017, 21 March 2017 through 24 March 2017*, pages 668–679. OpenProceedings. org, 2017.

7 Ceph. `https://docs.ceph.com/en/quincy/`, 2023.

8 Wei-Che Chien, Chin-Feng Lai, and Han-Chieh Chao. Dynamic resource prediction and allocation in c-ran with edge artificial intelligence. *IEEE Transactions on Industrial Informatics*, 15(7):4306–4314, 2019.

9 Taina Coleman, Henri Casanova, Ketan Maheshwari, Loïc Pottier, Sean R Wilkinson, Justin Wozniak, Frédéric Suter, Mallikarjun Shankar, and Rafael Ferreira da Silva. Wfbench: Automated generation of scientific workflow benchmarks. In *2022 IEEE/ACM International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pages 100–111. IEEE, 2022.

10 Philip A Ewels, Alexander Peltzer, Sven Fillinger, Harshil Patel, Johannes Alneberg, Andreas Wilm, Maxime Ulysse Garcia, Paolo Di Tommaso, and Sven Nahnsen. The nf-core framework for community-curated bioinformatics pipelines. *Nature biotechnology*, 38(3):276–278, 2020.

11 Diogo R Ferreira and JET Contributors. Using hpc infrastructures for deep learning applications in fusion research. *Plasma Physics and Controlled Fusion*, 63(8):084006, 2021.

12 Rafael Ferreira da Silva, Gideon Juve, Mats Rynge, Ewa Deelman, and Miron Livny. Online task resource consumption prediction for scientific workflows. *Parallel Processing Letters*, 25(03):1541003, 2015.

13 Rosa Filguiera, Amrey Krause, Malcolm Atkinson, Iraklis Klampanos, and Alexander Moreno. dispel4py: A python framework for data-intensive scientific computing. *The International Journal of High Performance Computing Applications*, 31(4):316–334, 2017.

14 Salvatore Giampà, Loris Belcastro, Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. A data-aware scheduling strategy for executing large-scale distributed workflows. *IEEE Access*, 9:47354–47364, 2021.

15 Naveen Kumar Gondhi and Ayushi Gupta. Survey on machine learning based scheduling in cloud computing. In *Proceedings of the 2017 International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence*, pages 57–61, 2017.

16 Shantenu Jha, Vincent R Pascuzzi, and Matteo Turilli. Ai-coupled hpc workflows. *arXiv preprint arXiv:2208.11745*, 2022.

17 Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, 2023.

18 Fabian Lehmann, Jonathan Bader, Friedrich Tschirpke, Lauritz Thamsen, and Ulf Leser. How workflow engines should talk to resource managers: A proposal for a common workflow scheduling interface. *arXiv preprint arXiv:2302.07652*, 2023.

19 Chin Poh Leong, Chee Sun Liew, Chee Seng Chan, and Muhammad Habib Ur Rehman. Optimizing workflow task clustering using reinforcement learning. *IEEE Access*, 9:110614–110626, 2021.

20 Xiaoyi Lu, Haiyang Shi, Rajarshi Biswas, M Haseeb Javed, and Dhabaleswar K Panda. Dlobd: A comprehensive study of deep learning over big data stacks on hpc clusters. *IEEE Transactions on Multi-Scale Computing Systems*, 4(4):635–648, 2018.

21 Lustre. `https://www.lustre.org/`, 2023.

22 Federica Magnoni, Emanuele Casarotti, Pietro Artale Harris, Mike Lindner, Andreas Rietbrock, Iraklis Angelos Klampanos, Athanasios Davvetas, Alessandro Spinuso, Rosa Filgueira, Amy Krause, et al. Dare to perform seismological workflows. In *AGU Fall Meeting Abstracts*, volume 2019, pages IN13C–0726, 2019.

23 Avinab Marahatta, Qin Xin, Ce Chi, Fa Zhang, and Zhiyong Liu. Pefs: Ai-driven prediction based energy-aware fault-tolerant scheduling scheme for cloud data center. *IEEE Transactions on Sustainable Computing*, 6(4):655–666, 2020.

24 Manolis Marazakis, Marc Duranton, Dirk Pleiter, Giuliano TAFFONI, and Hans-Christian Hoppe. Hpc for urgent decision-making. 2022.

25 Jargalsaikhan Narantuya, Jun-Sik Shin, Sun Park, and JongWon Kim. Multi-agent deep reinforcement learning-based resource allocation in hpc/ai converged cluster. *Computers, Materials & Continua*, 72(3), 2022.

26 Azita Nouri, Philip E Davis, Pradeep Subedi, and Manish Parashar. Exploring the role of machine learning in scientific workflows: Opportunities and challenges. *arXiv preprint arXiv:2110.13999*, 2021.

27 Scott Dale Peckham. How can advances in ai help the working geoscientist? In *AGU Fall Meeting Abstracts*, volume 2018, pages IN12A–05, 2018.

28 Sebastián Risco, Germán Moltó, Diana M Naranjo, and Ignacio Blanquer. Serverless workflows for containerised applications in the cloud continuum. *Journal of Grid Computing*, 19:1–18, 2021.

29 Alberto Scionti, Paolo Viviani, Giacomo Vitali, Chiara Vercellino, Olivier Terzo, et al. Enabling the hpc and artificial intelligence cross-stack convergence at the exascale level. In *HPC, Big Data, and AI Convergence Towards Exascale: Challenge and Vision*, pages 37–58. CRC Press, 2022.

30 Nikolay A Skvortsov and Sergey A Stupnikov. A semantic approach to workflow management and reuse for research problem solving. *Data Intelligence*, 4(2):439–454, 2022.

31 Johannes Starlinger, Bryan Brancotte, Sarah Cohen-Boulakia, and Ulf Leser. Similarity search for scientific workflows. *Proceedings of the VLDB Endowment (PVLDB)*, 7(12):1143–1154, 2014.

32 Ozan Tuncer, Emre Ates, Yijia Zhang, Ata Turk, Jim Brandt, Vitus J Leung, Manuel Egele, and Ayse K Coskun. Diagnosing performance variations in hpc applications using machine learning. In *High Performance Computing: 32nd International Conference, ISC High Performance 2017, Frankfurt, Germany, June 18–22, 2017, Proceedings 32*, pages 355–373. Springer, 2017.

33 Matteo Turilli, Mark Santcroos, and Shantenu Jha. A comprehensive perspective on pilot-job systems. *ACM Computing Surveys (CSUR)*, 51(2):1–32, 2018.

34 Dan Wang, Qiang Du, Tong Zhou, Antonio Gilardi, Mariam Kiran, Bashir Mohammed, Derun Li, and Russell Wilcox. Machine learning pattern recognition algorithm with applications to coherent laser combination. *IEEE Journal of Quantum Electronics*, 58(6):1–9, 2022.

35 Carl Witt, Marc Bux, Wladislaw Gusew, and Ulf Leser. Predictive performance modeling for distributed batch processing using black box monitoring and machine learning. *Information Systems*, 82:33–52, 2019.

36 Carl Witt, Dennis Wagner, and Ulf Leser. Feedback-based resource allocation for batch scheduling of scientific workflows. In *2019 International Conference on High Performance Computing & Simulation (HPCS)*, pages 761–768. IEEE, 2019.

37 Michael R Wyatt, Stephen Herbein, Todd Gamblin, and Michela Taufer. Ai4io: A suite of ai-based tools for io-aware scheduling. *The International Journal of High Performance Computing Applications*, 36(3):370–387, 2022.

38 Zaynab Zahra, Zihao Li, and Rosa Filgueira. Laminar: A new serverless stream-based framework with semantic code search and code completion. In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 2009–2020, 2023.

39 Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158*, 2023.

40 Shuo Zhang, Zhuofeng Zhao, Chen Liu, and Shenghui Qin. Data-intensive workflow scheduling strategy based on deep reinforcement learning in multi-clouds. *Journal of Cloud Computing*, 12(1):125, 2023.

## 3.4 Human Interaction and Accessibility

*Rafael Ferreira da Silva (Oak Ridge National Laboratory, US)*
*Daniel Laney (Lawrence Livermore National Laboratory, US)*
*Paolo Missier (Newcastle University, GB)*
*Jedrzej Rybicki (Jülich Supercomputing Centre, DE)*
*Matthias Weidlich (Humboldt University of Berlin, DE)*



In the rapidly evolving landscape of HPC, the integration of modern science applications necessitates the reevaluation of traditional workflow paradigms, particularly in the context of real-time interacting during workflow execution. Whether mediated by humans, AI, or specialized algorithms, these interactions introduce a new layer of complexity and opportunity across the experiment lifecycle when deployed on HPC environments. Such interactions serve critical roles, from real-time steering of ongoing computations to runtime recalibration of parameters, enabling adaptive adjustments that can feed back into the experiment itself for iterative improvement. Furthermore, these interactions enable data-driven decisions on cost/value trade-offs, thereby optimizing both computational and financial resources.

Building on the notion of real-time interactions, an example is the iterative process of workflow creation aimed at solving intricate problems based on initial specifications. In such cases, multiple workflow versions often emerge and are refined through expert-mediated (or possibly AI- or algorithm-mediated) interactions. The main goal is to perform an exploration of the solution space of a given problem rather than obtaining a single solution. Two key scenarios emerge in this context: (1) the independent exploration of alternative designs that can execute in parallel; and (2) a phased exploration where designs are sequentially refined based on observed results. To understand these challenges, we examine them within a framework of stakeholder interactions that occur throughout the standard lifecycle of modern HPC workflows.

Fig. 4 suggests a reference framework for placing three main stakeholders: Scientists, SysAdmins, and Data Scientists, in the context of current HPC-based workflow practice, and how their interaction contributes to extending the current batch-oriented model of workflow execution. Specifically, Scientists include a class of users who "own" the scientific problems that the workflow aims to address, and are responsible for the design of workflows that contribute to their solution. Examples of these are given below. A validation step is then typically required to ensure that the workflow fits the system requirements of the underlying HPC infrastructure. Optimizers and SysAdmin roles with expert knowledge of the HPC architecture and its resource allocation policies mediate this step, as suggested in the Figure. In a standard HPC batch submission model, the resulting validated workflow maps to jobs that are scheduled through a process of resource allocation.

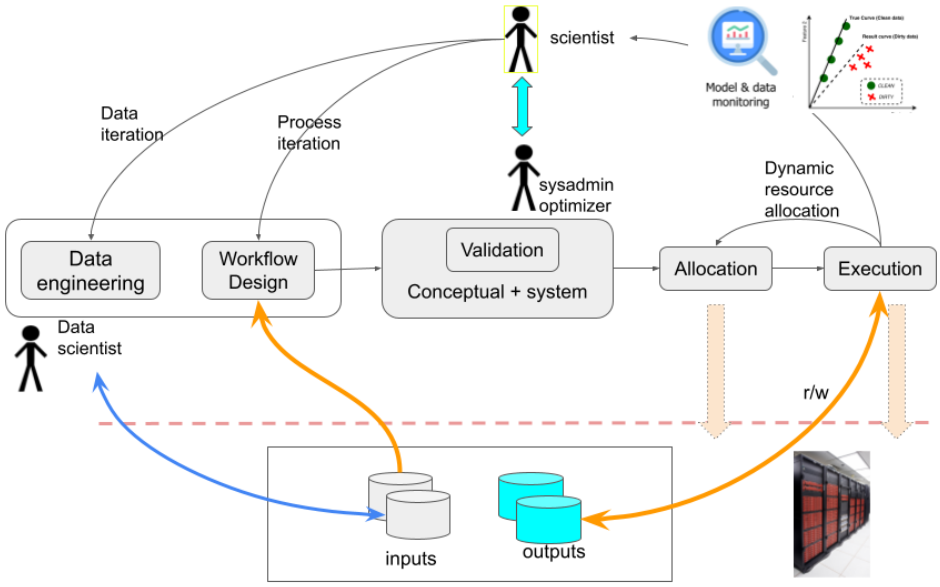


**Figure 4** Simplified Framework of Stakeholder Roles in HPC Workflows, highlighting the collaboration between Scientists, SysAdmins, and Data Scientists to advance beyond standard batch-oriented workflow models.

We suggest that data-centric workflow design may bring new and potentially disruptive elements into the framework. Two main scenarios are common. Firstly, a familiar iterative exploration of the solution space by the scientists, which is required to converge on a stable workflow design for a new problem. This involves a repeated interleaving of four steps: batch execution, analysis of results, workflow refinement, and resubmission.

Workflow design naturally accounts for the type and structure of the underlying datasets (indicated as "inputs" in the Figure), however a realistic iterative refinement process needs to account for the need to refine the data, in addition to the workflow, at each iteration. To achieve this, we envision an additional Data Scientist role, who is responsible for data engineering and pre-processing tasks as required to align the input datasets to the workflow requirements. A notable class of problems where this additional step is needed is in the context of so-called Data-Centric AI [39], where the workflow is designed to deliver a model (for example, to predict some outcome), and the corresponding data tasks at each iteration include cleaning, correcting for bias, generating synthetic data to complement a training set, experimenting with alternative data imputation strategies, and more.

A second scenario occurs when each batch execution itself may be interrupted and broken down into separate components, with a decision process in the middle. For instance, during a process of parameter sweeping (exploration) that consists of an array of parallel tasks, it may be possible to identify promising regions of the parameter space and thus to steer the exploration, by pruning some of the tasks and starting new ones. In many cases this can be determined algorithmically, however we can also envision a human-in-the-loop scenario where the roles identified in the Figure directly participate in the decision process.

### 3.4.1 An Overview of the Challenges

Since HPC systems are primarily batch oriented, users evolve a cadence of job submission, analysis, and new submissions, sometimes over periods of days or weeks. With advances in ML/AI, we expect both human-in-the-loop and AI-driven workflows will become more

common. The challenge is that this new paradigm, of dynamic workflows, is at odds with existing HPC center policies and capabilities in many cases. Furthermore, human interaction with large scale HPC resources means that the benefit of the interaction or dynamism must be quite high to justify the costs associated with pausing, restarting, or spawning computations.

A key observation is that users often begin at a small scale where interaction is far less costly in terms of "wasted" compute, in order to understand and begin to design their workflow. However, at these small scales workflow execution tools are not needed, and can be an impediment due to their complexity, and are thus not introduced into the process until it is time to scale up the problems and number of simulations. ML/AI provides a way to introduce the "full" workflow environment early in the process, replacing expensive simulation with fast-running surrogates, and enabling interactive exploration with the entire software environment. We believe that a holistic, co-design approach is needed, in which workflows can be introduced early and scaled through the entire process of workflow creation, validation, and execution, from laptop scale to exascale.

In this section, we survey the challenges to human-in-the-loop and AI-driven workflows in current HPC centers:

- **Resource Allocation:** The integration of dynamic, AI-driven workflows into HPC systems faces a significant challenge due to the discrepancy between traditional HPC resource allocation policies and the requirements of these modern workflows. Traditional HPC centers, primarily designed for batch-oriented tasks, struggle to accommodate non-batch workflows like human-in-the-loop or AI-driven, data-centric processes. The absence of elasticity in resource allocation further exacerbates this issue, limiting the adaptability essential for dynamic workflows and leading to workload unpredictability. Consequently, there is an urgent need to develop HPC policies that support on-demand, AI-directed workflows, distinguishing between "compute projects" suited for batch processing and 'data projects' that demand a more flexible, interactive approach. Adopting a holistic, co-design strategy is crucial, allowing workflows to be introduced early and scaled efficiently from small-scale experiments to extensive executions, integrating ML/AI to create fast-running surrogates for interactive exploration and efficient resource utilization from the outset.
- **Data Management:** Data management and workflows overlap in many ways. Data are triggers, inputs, and outputs of scientific workflows; they also serve as an "integration layer" between workflow steps, and thus can help debug their execution. Workflows are also helpful in understanding the provenance of a particular dataset by describing what kind of processing led to its creation. For the intelligence (human or artificial) in the loop, the data plays a critical role. Decisions about how to proceed with the workflow are based on data, such as intermediate results. All this poses many challenges in terms of data management and workflow execution. The availability and placement of data plays a role in the execution plan of a workflow. The data created in the workflow must be made available to the external entity (intelligence in the loop) in a timely manner to enable interaction with the workflow execution. Finally, workflows and data have different lifetimes (the data remains important and valid even after the workflow execution has ended), so the resource manager responsible for workflow execution must be able to make both short-term and long-term decisions.
- **Debugging and Provenance:** This challenge centers around the need for deep checkpointing and meticulous action tracking. Essential to this challenge is the provision of provenance data in real-time, which would significantly enhance decision-making during workflow execution. However, this raises potential issues, such as the risk of real-time

provenance tracking interfering with the execution of workflows. Additionally, there are concerns regarding trust and the integrity of provenance data, which are crucial for reliable and verifiable scientific computations. Addressing these challenges requires a careful balance between providing detailed, real-time insights into workflow processes and ensuring that these mechanisms do not disrupt the efficient execution of complex computational tasks.

- **AI Integration:** This challenge revolves around preparing for high levels of unforeseen automation and the complexities it brings. To future-proof systems against this, a co-design approach is essential, where both hardware and software stacks are developed with workflow applications and their unique requirements in mind. Another critical aspect is ensuring that these AI-integrated workflows are explainable, which can be achieved by leveraging provenance and execution traces to provide clarity and understanding of the AI's decision-making process. However, this integration is not without its difficulties, as competing AIs within the same ecosystem may attempt to optimize at the application level for their own benefit, leading to uncertainties and complexities in workflow management. Addressing these challenges requires a sophisticated balance between advancing AI capabilities and maintaining control and transparency over automated processes in computational workflows.
- **Abstractions for Human Interaction:** This challenge involves facilitating human involvement at every stage, including creation, deployment, resource allocation, and execution. This requires designing roles and interfaces that cater to different stakeholders such as users, workflow experts, facility personnel, and data scientists, ensuring their input is valuable and feasible at various stages of the workflow. Additionally, AI can be integrated as a surrogate for "any human in the loop," performing tasks or making decisions in places where human intervention is typically required. Key types of user interaction that need to be abstracted include changing parameters of the workflow, starting or killing jobs based on real-time needs, and modifying the data, such as adjusting the samples used in the computation. The design of these abstractions must be intuitive and flexible, allowing for efficient and effective human-AI collaboration in managing complex computational processes.

In confronting the challenges of integrating advanced workflows into HPC systems, lessons from the history of HPC and its applications are invaluable. History shows that as HPC evolved, its increasing complexity often created barriers to automation and higher-order reasoning, a pattern now emerging in workflow systems as they accrue complex notations and concepts. This parallel suggests the need for a careful approach to developing workflow systems, ensuring they do not become so intricate that they hinder the very progress they are designed to facilitate. As we design these systems, it is crucial to consider the cost/value trade-offs in every decision, recognizing that "cost" can be multifaceted, encompassing computational resources, ease of use, and adaptability to future technologies. The key lies in learning from the past to build workflow systems that are robust, efficient, and accessible, thereby enabling them to be powerful tools in the advancement of scientific research and applications, rather than becoming cumbersome obstacles.

# 4 Future Perspectives

## 4.1 Optimization and Efficiency

*Ana Gainaru (Oak Ridge National Laboratory, US)*
*Shantenu Jha (Rutgers University, US & Brookhaven National Laboratory, US)*
*Christine Kirkpatrick (San Diego Supercomputer Center, US)*
*Daniel Laney (Lawrence Livermore National Lab, US)*
*Wolfgang E. Nagel (TU Dresden, DE)*
*Jedrzej Rybicki (Jülich Supercomputing Centre, DE)*
*Domenico Talia (University of Calabria, IT)*



For many years, HPC has been driven by extremely skilled individuals who are able to capture the complete execution of their scientific code, debug it, and optimize it to squeeze the last drops of performance out of the underlying infrastructures. We argue that this model is no longer sustainable. Scientific endeavors require multi-step workflows, the underlying infrastructures are becoming heterogeneous, and there are constant advances in methods that require the integration of new codes. As a first step, we need a cultural change that incentivizes the sharing of codes and workflows. In addition, the technical solutions should be designed so that the replacement of individual computational steps in workflows to incorporate new implementations or methods should be seamless. Workflow performance optimization will become a team effort, with scientists supported by HPC experts who consider not only the maximum performance of individual steps, but also a holistic view of the execution, including, for example, data movement. Future HPC centers will evolve in much the same way that cloud providers have evolved from offering Infrastructure-as-a-Service to higher abstractions such as Software-as-a-Service or Function-as-a-Service. Workflow descriptions can become the vehicle that drives this movement to higher abstractions and performance optimization as a team effort.

### 4.1.1 Performance Models and Reasoning

- We still do not reason well about performance of workflows, and facilities and procurements still focus primarily on individual application performance patterns.
- We often exclude data considerations, location, movement, etc.
- We need to build workflow benchmarks to help the community understand the more complex performance characteristics of workflows.
  - Throughput, makespan, responsiveness, time to solution
  - Understanding how these relate to each other so you can reason about them.
- Collective performance measures are different than for individual user/app.
  - We need to be able to see the entire workflow and reason.
  - We need to take a closer look at data, which can become bottleneck point.
- Empiricism: observing workflow behavior, and incrementally improving via learning from the behavior of the system.
  - Develop best practices.
  - Converge on principals.
- We thus need to have telemetry data on our workflows that allows this learning to occur; Post-mortem & Digital Twins.

### 4.1.2 Optimizing the Quality of Science

- We have to find definition on how to measure scientific quality for our workflows
- We need to define API's and architectures that enable future workflows to optimize under these currently unknown measures of quality
- We need to reason from examples:
  - Protein folding: standard, simple approach, is the baseline, approaches exist for interrogating the simulation and adjusting it to accelerate the folding.
    - Result of optimization is the same structure.
    - But time to solution can be 1000x shorter.
    - Still users often default to the simple, easy approach.
    - A Workflow system, perhaps augmented by AI.
- Preconditioners: many codes of iterative linear solvers in them (sometimes many such solvers).
  - Experts research preconditioners to accelerate solution.
  - Expert users choose among these for their problem, sometimes via experimentation.
  - We envision that a workflow system with AI could detect or predict solver behavior and choose preconconders to optimize execution.
- Coupled multi-application workflows under propagation of uncertainties.
  - In these workflows, users often set up and optimize each application in the workflow independently.
  - AI + workflow could understand parameter sensitivities and uncertainties, and optimize for scientific output, potential reducing cost of simulations.

### 4.1.3 A Vision for HPC & Data Centers

- **We have to think beyond simple scheduling and move towards multi-dimensional + temporal scheduling.**
  - It is not just about when to run but where you get the data from.
  - It needs to be spatio-temporal, co-location.
  - There is scheduling at the workflow level, workload level, and task level
  - The challenge is that it is relatively easy to schedule at one level or in one dimension, but how do enable usage of information across all levels to optimally schedule work.
  - Open research question: should information flow only upwards from lower levels (separation of concerns), or should it be globally available.

## 4.2 Lifecycle Management in Federated Workflows

*Rosa M. Badia (Barcelona Supercomputing Center, ES)*
*Kyle Chard (University of Chicago, US)*
*Timo Kehrer (Universität Bern, CH)*
*Dejan Milojicic (HP Labs – Milpitas, US)*
*Fred Suter (Oak Ridge National Laboratory, US)*
*Sean R. Wilkinson (Oak Ridge National Laboratory, US)*



"Federated workflows," a term also known as "cross-facility workflows" [1], represent a paradigm shift from traditional "grid computing." Grid computing typically refers to a distributed computing model where resources from various locations are pooled together to work on large-scale problems, often with a focus on maximizing resource utilization and computational power across a network of machines. In contrast, the concept of federated workflows extends beyond just resource sharing. It involves the integration and orchestration of workflows across different computational facilities, each possibly with its own unique environment, policies, and capabilities. This approach is not only about aggregating computational power but also about seamlessly executing complex processes that span across these varied environments, maintaining coherence and coordination throughout the workflow life cycle.

Understanding how federated workflows differ from traditional grid computing models, such as those exemplified by the Worldwide LHC Computing Grid (WLCG), is crucial. While grid computing primarily addresses the challenge of resource scarcity through distributed computing, federated workflows focus on the integration and interoperability of distinct computational workflows across various facilities. This distinction underscores a more nuanced approach to handling diverse computational tasks, data management practices, and workflow optimizations in a unified manner.

While it is possible to manage simpler federated workflows manually, the development of a dedicated federated engine could significantly enhance the efficiency and scalability of these systems. Such an engine would automate the execution of federated tasks, eliminating the need for building ad-hoc workflows for each new project. This would not only streamline the workflow execution across different facilities but also reduce the time and effort required for coordination and integration, thereby enabling more complex and dynamic federated workflows to be executed with greater ease and efficiency. The move towards federated workflows marks a significant evolution in how we approach distributed computing, emphasizing the need for sophisticated integration and orchestration tools to manage the complexities of modern computational tasks.

### 4.2.1 Use Cases for Federation

There are numerous use cases which demonstrate the value of federated workflows. We list a few here, but the list is not exhaustive.

- **Overflow / offload / cloud bursting.** Our first use case involves temporarily providing more compute and data resources upon demand by users, potentially in ways that are seamless and invisible to the user. For example, a user of an on-premise HPC system might require more resources than are currently available to run a workflow, and a potential solution might be to offload their workflow to execute on cloud resources.

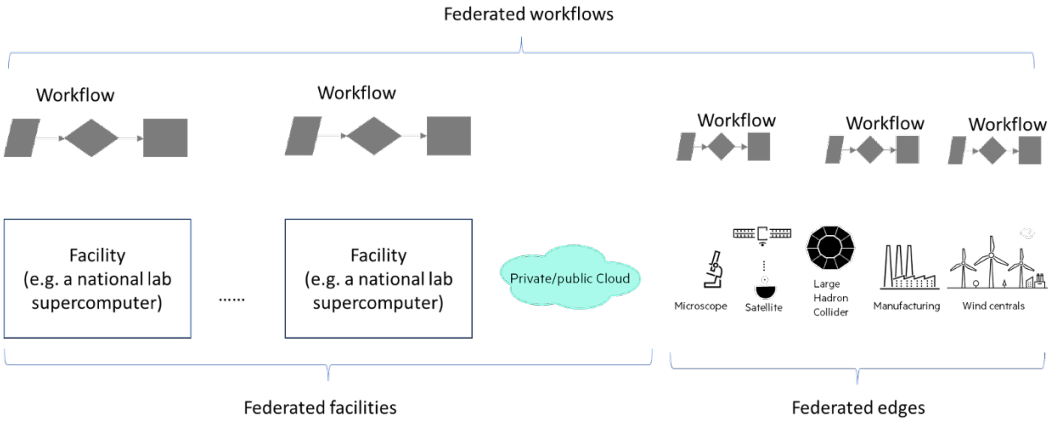


**Figure 5** Conceptual Diagram of Federated Workflow Architecture. This figure illustrates the flow of data and tasks across a federated system, incorporating both private and public cloud resources.

- **Redundancy.** Similarly, during outages at HPC facilities, which may be planned or unplanned, users still need to execute their workflows. Federated workflows which execute across multiple facilities need to be capable of retargeting sites for execution dynamically in cases where facilities may be completely unavailable.
- **Urgent computing.** Sometimes, even offloading to the cloud may be insufficient to provide resources because the situation demands all possible resources from HPC facilities, clouds, and the like. Truly urgent situations like wildfires [2] and the recent COVID-19 pandemic [3] need to minimize time-to-solution in order to save lives, so they need to execute at high priority and the largest scales.
- **Internet of Things.** Federated resources also include very "small" compute and data resources like Internet of Things (IoT) devices. This use case involves edge devices which may have slow processors, power constraints, and reduced network connectivity as compared with HPC facilities. These devices can still be sufficient for federated learning, however, and this use case is therefore important for certain kinds of AI workflows [4].
- **Coupling edge computing with HPC.** There are also architectures in which edge devices can be used to process data in near-real-time, while coupling with analyses executed on true HPC resources. For example, a scientist at a light source might run parts of a workflow locally while collecting observational data at the same time that HPC resources are crunching numbers to help steer the data collection process [5].
- **Exotic hardware like quantum computers.** Federated workflows also take advantage of unique capabilities available at different facilities, such as quantum computers or classical HPC resources with exotic architectures. For example, OLCF has on-premises classical HPC resources, but the quantum computing resources are provided to its users by the cloud; these can still be used in the same workflow [6]. Different HPC facilities' flagship systems often have different architectures which lend themselves well to different kinds of problems, such as CPU-intensive or GPU-intensive tasks.
- **Collaborative data analysis.** In this use case, scientists use a shared dashboard to display analyses and visualizations of data produced by an HPC simulation. New analyses and visualizations can be dynamically added to that dashboard, potentially triggering/steering new computations (link). This use case thus relies on federated computing resources – supercomputer, analysis/visualization cluster, and cloud service for the dashboard.

- **Multi-instrument federation.** Another use case for federating facilities would allow not only for federating compute resources, but also for federating the instruments themselves. For example, an event-driven workflow in multi-messenger astrophysics could react to a neutrino detection event by DUNE by aiming telescopes at a region of interest.

### 4.2.2 Policies

In this section, we address the complex and essential role that policies play in the successful implementation and management of federated workflows. While technical aspects often dominate the concerns of engineers, policies are equally critical in making federated workflows viable and effective. These policies encompass a wide range of considerations, including governmental regulations like the General Data Protection Regulation (GDPR), which dictates who can access and execute workflows, as well as more technical aspects like the duration and frequency of workflow connections and invocations. Another key policy area involves identity verification, such as the requirements for Multi-Factor Authentication (MFA) and the timeframe within which it must be validated. Additionally, policies regarding user accounts at various facilities, including the necessity for signed agreements, play a crucial role in governing how federated workflows operate within and across different organizational domains.

The management of policies in federated workflows builds upon existing policies for individual workflows. However, when these workflows cross organizational boundaries, they begin to resemble aspects of past Grid computing work, albeit with unique challenges and considerations. Policies in federated workflows are critical not only for their deployment but also for their widespread adoption. Without well-defined and enforced policies, it becomes challenging to manage key aspects such as security, quality of service (QoS), sustainability, and availability in a manner that aligns with both organizational objectives and cross-organization collaboration. These challenges make the policy landscape for federated workflows a fertile ground for research, where the development of new policies and the adaptation of existing ones can significantly impact the efficiency and effectiveness of these complex computational ecosystems.

### 4.2.3 Workflow Federation Patterns/Motifs

We identified a series of execution patterns for future federated workflows. These patterns are built incrementally, starting from an existing and well defined scientific use case and then adding new features and/or constraints to this initial scenario.

- **One to many facilities.** Our initial use case corresponds to the execution pattern underlying the processing of data produced by the four High Energy Physics experiments deployed on the Large Hadron Collider at CERN (i.e., ATLAS, CMS, ALICE, and LHCb). These experiments rely on the worldwide LHC computing grid (WLCG) for about two decades. This computing grid comprises multiple computing centers across the world that are federated to execute a single workload composed of millions of independent jobs (MC simulations). Each center has a contractual commitment to provide resources with a given availability from other sites to be able to run on their sites. There is also an upstream control of the distribution of the jobs among the sites according to their current load and a common authentication overlay based on proxy certs to enable the federated infrastructure. However, the management of the federation is human-engaged.

- **Federated facility.** To automate the management of a "one to many facilities" federation and form a truly federated facility, we consider collecting data to train an AI model that optimizes what jobs to run and where to run them, in terms of performance. Additionally, we propose that all contractual agreements across the different participants of the federated facility are verified at any point of time during the execution of the workload.
- **Sustainable federated facility.** The next stage is then to add a sustainability dimension, in the environmental and social meanings of the term, to the proposed federated facility. The objective there is to complement the mapping and scheduling decisions based solely on performance by using AI to help decide not only where to execute the workload according to the environment sustainability profile of the different sites but also whether it is better to not run some subset of the workload than wasting precious resources.
- **Federatable workflows.** We then introduce the concept of federatable workflow, i.e., a workflow published by one of the facilities belonging to the federation that can be composed with other federatable workflows to form a more complex federated workload. A federatable workflow can be considered as a end point of the end-to-end federation along with computing and experimental (e.g., telescopes, microscopes, light sources, etc) facilities.
- **Data-constrained federated workflows.** Sometimes data cannot move due to governance/policy constraints, such as data produced in the European research space or in the case of biomedical data). In that case, it becomes necessary to "send compute to the data". Such constraints on data movements add geographical constraints that shape the workflow and further motivate a federation of workflows.

### 4.2.4 Contracts for Federated Workflow

Because federated workflows can cross organizational boundaries, some support for contracts is required to formalize the bindings between invocations that can cross administrative, legal, financial, and other boundaries. These contracts could be very simple, such as offline agreement that all dynamic bindings between certain entities are either permissible or not, or they could be substantially more complex in terms of financial, legal and other dynamic bindings that need to take place at any location.

The non-trivial aspect comes from the fact that QoS can have different meanings in different organizations. For this reason this remapping can be built into the federation engine which will be able to translate requirements, observe compliance and address disputes, violations, etc. We can express QoS and authorization as a part of the federation at federation time, so that we know whether from responsiveness, throughput, resources, etc. the federation point will work for the needs of the workflow.

Addressing privacy concerns of regional areas (Europe, US, Asia) or individual entities (e.g., farms (about their crops), businesses (e.g., medical)) can also be done at the federation engine. Federation engines can become a basis for federation of facilities, workflows, and anything in between (devices, workflow tasks).

### References

1 Katerina B Antypas, DJ Bard, Johannes P Blaschke, R Shane Canon, Bjoern Enders, Mallikarjun Arjun Shankar, Suhas Somnath, Dale Stansberry, Thomas D Uram, and Sean R Wilkinson. Enabling discovery data science through cross-facility workflows. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3671–3680. IEEE, 2021.

2 Ilkay Altintas, Jessica Block, Raymond De Callafon, Daniel Crawl, Charles Cowart, Amarnath Gupta, Mai Nguyen, Hans-Werner Braun, Jurgen Schulze, Michael Gollner, et al. Towards an integrated cyberinfrastructure for scalable data-driven monitoring, dynamic prediction and resilience of wildfires. *Procedia Computer Science*, 51:1633–1642, 2015.

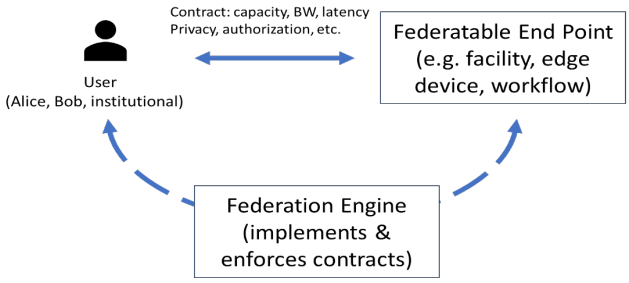


**Figure 6** Schematic Representation of a Federation Engine within a Federated Workflow System enforcing contracts.


3 Hyungro Lee, Andre Merzky, Li Tan, Mikhail Titov, Matteo Turilli, Dario Alfe, Agastya Bhati, Alex Brace, Austin Clyde, Peter Coveney, et al. Scalable hpc & ai infrastructure for covid-19 therapeutics. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–13, 2021.

4 Omer Rana, Theodoros Spyridopoulos, Nathaniel Hudson, Matt Baughman, Kyle Chard, Ian Foster, and Aftab Khan. Hierarchical and decentralised federated learning. *arXiv preprint arXiv:2304.14982*, 2023.

5 James E McClure, Junqi Yin, Ryan T Armstrong, Ketan C Maheshwari, Sean Wilkinson, Lucas Vlcek, Ying Da Wang, Mark A Berrill, and Mark Rivers. Toward real-time analysis of synchrotron micro-tomography data: accelerating experimental workflows with ai and hpc. In *Driving Scientific and Engineering Discoveries Through the Convergence of HPC, Big Data and AI: 17th Smoky Mountains Computational Sciences and Engineering Conference, SMC 2020, Oak Ridge, TN, USA, August 26-28, 2020, Revised Selected Papers 17*, pages 226–239. Springer, 2020.

6 Samuel T Bieberich, Ketan C Maheshwari, Sean R Wilkinson, In-Saeng Suh, Rafael Ferreira da Silva, et al. Bridging hpc and quantum systems using scientific workflows. *arXiv preprint arXiv:2310.03286*, 2023.


## 4.3 Advanced Techniques and Innovations

*Laure Berti-Equille (IRD – Montpellier, FR)*
*Rosa Filgueira (University of St Andrews, GB)*
*Ulf Leser (HU Berlin, DE)*
*Bertram Ludäscher (University of Illinois at Urbana-Champaign, US)*
*Paolo Missier (Newcastle University, GB)*
*Jeyan Thiyagalingam (Rutherford Appleton Lab, GB)*



### 4.3.1 LLMs for workflows

Large Language Models (LLMs), such as UniXCoder, codeBERT, codex, GPT-4, GraphCodeBERT, CodeT5, have already proven their utility in enhancing developers' capabilities to manage and produce more efficient code across a range of domains. These models have opened up exciting possibilities for automating and optimizing software development processes. However, it is equally pertinent to recognize the immense potential LLMs hold in the

domain of scientific workflows. Here, we summarise the different ways LLMs can be used to improve the management and efficiency of scientific workflows. Below, we present a list of use cases in which LLMs can offer benefits to the scientific workflow community:

- **Discovering Similar Workflows.** Large language models can help researchers discover similar workflows in their domain. By analyzing descriptions or code snippets of existing workflows, these models can identify patterns and similarities, aiding researchers in finding relevant examples and benchmarks for their own projects. This functionality can significantly reduce the time and effort required to design new workflows.
- **Identifying Similar Workflow Components.** Within a given scientific workflow, there are often recurring components such as data preprocessing, analysis modules, and visualization tools. Language models can assist in identifying similar components across different workflows. This capability can enhance code reusability and promote the sharing of standardized components within the scientific community.
- **Autocompleting Tasks and Workflows.** Language models can serve as intelligent code completion tools, making it easier for researchers to write and refine workflow scripts. As scientists input their desired tasks or steps, these models can suggest code snippets, offer parameter recommendations, and assist in handling dependencies, resulting in more efficient and error-free workflow development.
- **Describing and Summarizing Workflows and Tasks.** Scientific workflows often involve intricate data manipulation and analysis procedures that can be challenging to document comprehensively. Large language models can automatically generate human-readable descriptions and summaries of workflows and individual tasks. This enhances the accessibility of the workflow for collaborators and future reference.
- **Interoperability Across Workflow Engines/Frameworks.** Diverse workflow management systems often feature distinct syntax and structures. Large language models can simplify the process of translating workflows from one platform to another. Researchers can input their workflow in one system's language, and the model can generate equivalent code in the target platform's format, streamlining the migration process and minimizing errors.
- **Workflow Optimization and Tuning.** Large language models could also assist in optimizing and fine-tuning scientific workflows. By analyzing performance data and user-defined goals, these models can suggest improvements in resource allocation, parallelization strategies, and parameter tuning to achieve faster and more efficient execution.

## 4.4 Collaboration and Community Building

*Ilkay Altintas (San Diego Supercomputer Center, US)*
*Silvina Caino-Lores (INRIA – Rennes, FR)*
*Rafael Ferreira da Silva (Oak Ridge National Laboratory, US)*
*Christine Kirkpatrick (San Diego Supercomputer Center, US)*
*Matthias Weidlich (HU Berlin, DE)*



### 4.4.1 Community Building Initiatives

In the burgeoning field of HPC+AI workflow integration, there is a notable absence of dedicated venues for community engagement and knowledge exchange. To fill this void, initiatives can potentially evolve from Workflow Community Initiatives (WCI) and interdisciplinary-oriented venues, creating spaces where best practices for workflows, including pattern submissions linked to workflow module or pattern commons, can be discussed and refined. These venues could operate on a model of continuous submission, allowing for the real-time tracking of emerging or popular patterns. However, publishing these insights in a manner that engages communities unfamiliar with workflow intricacies remains a question. Furthermore, identifying key stakeholders within this ecosystem is essential to understand how systems should be architected to facilitate the interactions between these parties.

Addressing cultural differences in workflow structures is another significant challenge. Scientists often create monolithic workflows that are resistant to the modular, adaptable approaches emerging in modern workflow composition. To encourage the adoption of new practices, there must be tangible incentives for end users, and equitable methods to transfer development best practices from the computer science community to practitioners. Advocating for these cultural shifts at higher levels, such as funding agencies, is also critical and requires champions who can effectively communicate the value of these changes. Capturing the varied needs of intersecting communities, such as domain science and AI, presents both a challenge and an opportunity to distill knowledge from specialized groups.

Defining a specification for workflow assessment and evaluation tailored to specific use cases is an ongoing challenge. Distinguishing "workflow challenges" from benchmarks could help, with metrics that measure not only technical performance but also innovation, as exemplified by the DEBS community's approach to streaming datasets. Competitions, like those on Kaggle or potential student contests at focused venues, could concentrate on workflow patterns and their applications. The suitability of existing repositories for static workflows, such as WorkflowHub, Snakemake, BioBB, and MyExperiment, needs to be evaluated against the requirements of modern, dynamic workflows. The Workflow Module Commons on top of WCI could serve as an equitable service for building blocks, as suggested by Ilkay's work, but this raises the policy challenge of engaging institutions to host such services. AI could support automatic workflow generation, optimizing performance, and composing workflows from existing building blocks based on descriptions. However, this introduces the challenges of collecting training data, defining workflow requirements—particularly in terms of data and system specifications—and ensuring the certification of generated workflows and predictions for resource allocation.

**Recommendation.** build a community ecosystem for adding value to the collection, documentation and sharing of workflows, use cases and building blocks, promoted by supporting publications to derive best practises and dedicated venues hosting dedicated events (e.g., hackathons to apply best practises and patterns, workflow challenges).

### 4.4.2 Technical Challenges for Adoption of Data-Centric AI+HPC Approaches

The shift towards data-centric AI+HPC approaches poses several technical challenges for adoption, particularly in environments that have traditionally favored a task-centric perspective. A key question arises: how can the HPC community, which typically prioritizes tasks, be incentivized to adopt a data-centric approach that is essential for integrating AI components into workflows? This paradigm shift requires not only a recognition of the efforts involved in creating data-centric building blocks and generating valuable datasets but also strategies to democratize data within HPC, making it more accessible for use, sharing, and retrieval. Currently, many computing facilities operate in silos, with distinct responsibilities for computing or data management. Some institutions, like UCSD, have successfully developed institutional repositories, but this is not yet widespread. The challenge extends to the discovery of data and repositories linked to HPC, which often falls prey to resource prioritization issues rather than technical feasibility. To address these hurdles, there is a critical need to educate the next generation of HPC professionals and researchers in data-centric workflow thinking, laying a foundational understanding that will drive the future of integrated AI+HPC solutions.

**Recommendation.** Incensitivize integrating the role of data at the same level as tasks in HPC communities. Promote this idea in workflow venues, promote BOFs integrating HPC/data gap, foster partnerships with industry, incentivise publishing in data journals systematic description of datasets (e.g., DOIs for workflow components). Respond to RFIs, disseminate this Dagstuhl report, try to influence solicitations to ask for data-driven architectures, give incentives to be data forward, focusing on workflows as the interface/application layer for users/researchers.

**Recommendation.** User-centric software (workflows) should have simplified ways of showing provenance and habituating researchers to looking for verified/trustworthy data via provenance iconography that can be expanded for full details.

Current workflow systems are generally unprepared to handle the intricacies of the substantial hurdles in transitioning to workflows that are not just propelled by input or intermediate data, but also by metadata. This raises the question of whether there is an opportunity to reshape the design of these systems to integrate a data-focused perspective more thoroughly. A significant challenge lies in breaking away from established practices in HPC workflow development and system design, which have traditionally undervalued the role of data. Overcoming this will require a paradigm shift towards recognizing data not only as a passive element but as a dynamic driver of workflow processes, necessitating a fundamental re-evaluation of how data is integrated and leveraged within the HPC environment.

**Recommendation.** Machines will need to evolve in their architecture to accommodate hybrid workloads. Workflow managers shall be enabled to incorporate enriched data with metadata capturing properties of data to drive workflow management decisions in an ecosystem that encompasses system and task execution information.

## Participants


- Ilkay Altintas
San Diego Supercomputer Center – La Jolla, US
- Rosa Maria Badia
Barcelona Supercomputing Center, ES
- Laure Berti-Equille
IRD – Montpellier, FR
- Silvina Caino-Lores
INRIA – Rennes, FR
- Kyle Chard
University of Chicago, US
- Rafael Ferreira da Silva
Oak Ridge National Laboratory, US
- Rosa Filgueira
University of St Andrews, GB
- Ana Gainaru
Oak Ridge National Laboratory, US
- Shantenu Jha
Rutgers University – Piscataway, US & Brookhaven National Laboratory – Upton, US
- Timo Kehrer
Universität Bern, CH
- Christine Kirkpatrick
San Diego Supercomputer Center – La Jolla, US
- Ulf Leser
HU Berlin, DE
- Bertram Ludäscher
University of Illinois at Urbana-Champaign, US
- Dejan Milojicic
HP Labs – Milpitas, US
- Paolo Missier
Newcastle University, GB
- Wolfgang E. Nagel
TU Dresden, DE
- Jedrzej Rybicki
Jülich Supercomputing Centre, DE
- Frédéric Suter
Oak Ridge National Laboratory, US
- Domenico Talia
University of Calabria, IT
- Jeyan Thiyagalingam
Rutherford Appleton Lab. – Didcot, GB
- Matthias Weidlich
HU Berlin, DE
- Sean R. Wilkinson
Oak Ridge National Laboratory, US