DAGSTUHL
REPORTS

Volume 11, Issue 6, July 2021

Computational Proteomics (Dagstuhl Seminar 21271)
Sebastian Böcker, Rebekah Gundry, Lennart Martens, and Magnus Palmblad 1

Towards Climate-Friendly Internet Research (Dagstuhl Seminar 21272)
Vaibhav Bajpai, Jon Crowcroft, Oliver Hohlfeld, and Srinivasan Keshav 14

Data Structures for Modern Memory and Storage Hierarchies (Dagstuhl Seminar 21283)
Stratos Idreos, Viktor Leis, Kai-Uwe Sattler, and Margo Seltzer 38

Scalable Handling of Effects (Dagstuhl Seminar 21292)
Danel Ahman, Amal Ahmed, Sam Lindley, and Andreas Rossberg 54

Parameterized Complexity in Graph Drawing (Dagstuhl Seminar 21293)
Robert Ganian, Fabrizio Montecchiani, Martin Nöllenburg, and Meirav Zehavi ... 82

Matching Under Preferences: Theory and Practice (Dagstuhl Seminar 21301)
Haris Aziz, Péter Biró, Tamás Fleiner, and Bettina Klaus 124

Approximate Systems (Dagstuhl Seminar 21302)
Eva Darulova, Babak Falsafi, Andreas Gerstlauer, and Phillip Stanley-Marbell 147

ISSN 2192-5283

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

Publication date
November, 2021

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 `http://dnb.d-nb.de`.

Digital Object Identifier: 10.4230/DagRep.11.6.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
http://www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 21271

Computational Proteomics

Edited by

Sebastian Böcker[1], Rebekah Gundry[2], Lennart Martens[3], and Magnus Palmblad[4]

1 **Universität Jena, DE,** sebastian.boecker@uni-jena.de
2 **University of Nebraska – Omaha, US,** rebekah.gundry@unmc.edu
3 **Ghent University, BE,** lennart.martens@ugent.be
4 **Leiden University Medical Center, NL,** n.m.palmblad@lumc.nl

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 21271 "Computational Proteomics". The Seminar, which took place in a hybrid fashion with both local as well as online participation due to the COVID pandemic, was built around three topics: the rapid uptake of advanced machine learning in proteomics; computational challenges across the various rapidlly evolving approaches for structural and top-down proteomics; and the computational analysis of glycoproteomics data. These three topics were the focus of three corresponding breakout sessions, which ran in parallel throughout the seminar. A fourth breakout session was created during the seminar, on the specific topic of creating a Kaggle competition based on proteomics data.

The abstracts presented here first describe the three introduction talks, one for each topic. These talk abstracts are then followed by one abstract each *per* breakout session, documenting that breakout's discussion and outcomes.

An Executive Summary is also provided, which details the overall seminar structure alongside the most important conclusions for the three topic-derived breakouts.

Seminar July 4–9, 2021 – http://www.dagstuhl.de/21271
2012 ACM Subject Classification Applied computing → Bioinformatics
Keywords and phrases bioinformatics, computational mass spectrometry, machine learning, proteomics
Digital Object Identifier 10.4230/DagRep.11.6.1

1 Executive Summary

Lennart Martens (Ghent University, BE)
Rebekah Gundry (University of Nebraska – Omaha, US)
Magnus Palmblad (Leiden University Medical Center, NL)

The Dagstuhl Seminar 21271 "Computational Proteomics" discussed several important developments, challenges, and opportunities that are emerging in the field of computational proteomics. Three core topics were set out at the start, and these were discussed at length throughout the seminar.

These three topics were: (i) the fast evolving use of advanced machine learning approaches in proteomics; (ii) the challenges and opportunities offered by fast developing approaches for structural and top-down proteomics; and (iii) specific issues and computational complications in glycoproteomics.

The machine learning and glycoproteomics topics were each introduced by a dedicated lecture, which set out the current state-of-the-art and presented a tentative set of issues, challenges, or opportunities that could be explored during the seminar. The structural and top-down proteomics topic was introduced by two sequential lectures, one on structural proteomics, and one on top-down proteomics. In total, four introductory talks were thus presented at the start of the seminar. For each of the three main topics, daily Working Group sessions were organised, which took place in the morning and afternoon, with a daily late-night session scheduled each day to wrap up the day's outcomes. This structure was followed to allow maximum involvement by online participants across the various timezones in the hybrid format. The Machine Learning in Proteomics Working Group also spun out another Working Group session during the seminar, which discussed the creation of a machine learning (Kaggle-like) competition based on proteomics data.

Each of these breakout sessions was very actively attended, including by online attendees, and resulted in several interesting research ideas and potential new initiatives. The Machine Learning in Proteomics Working Group was the largest working group, and addressed a number of distinct topics during the seminar. Of particular note were the spin-out effort to establish two machine learning competitions based on proteomics data and challenges to engage the broader machine learning community, and the extensive discussions on the optimal way to represent mass spectrometry data for downstream machine learning.

The Glycoproteomics Working Group was very actively attended, and discussed an exciting set of topics. A first highlight among these topics was provided by the extensive and detailed discussions with the Machine Learning Working Group regarding the potential of, and road towards, the use of state-of-the-art machine learning approaches in glycoproteomics. A second highlight concerned the delineation of a set of high-impact opinion papers to describe the state-of-the-art of the field, and its goals, ambitions, and challenges.

The Structural and Top-Down Proteomics Working Group was very active in detailing the many challenges and opportunities in this fast-evolving field. One noteworthy challenge revolved around the detection, annotation, and biological interpretation of post-translational modifications detected by mass spectrometry. A second challenge concerned the standardization of acquired native mass spectrometry data, the minimal reporting requirements for these experiments, and the dissemination of these data.

Overall, the 2021 Dagstuhl Seminar on Computational Proteomics was extremely successful as a catalyst for careful yet original thinking about key challenges in the field, and as a means to enable downstream progress by setting important, high impact goals to work on in close collaboration. During this Seminar, new topics for a future Seminar were suggested throughout as well, indicating that this active field will continue to yield novel challenges and opportunities for advanced computational work going forward.

2 Table of Contents

Executive Summary
Lennart Martens, Rebekah Gundry, and Magnus Palmblad . . . 1

Overview of Talks

Topic Introduction: Shotgun Cross-Linking Mass Spectrometry and Protein Structure Prediction
Michael Hoopmann . . . 4

Topic Introduction: Future Outlook & Opportunities for Top Down Structural Proteomics
Neil Kelleher . . . 4

Topic Introduction: Machine Learning for MS-based Proteomics
Lukas Käll . . . 4

Topic Introduction: Glycoproteomics Challenges and Opportunities
Frédérique Lisacek . . . 5

Working groups

Working Group Report: Machine Learning in Proteomics
Lukas Käll, Marshall Bern, Sebastian Böcker, Sven Degrove, Bernard Delanghe, Viktoria Dorfer, Daniel Kolarich, Frédérique Lisacek, Magnus Palmblad, Robin Park, Veit Schwämmle, Matthew Smith, and Mathias Wilhelm . . . 5

Working Group Report: Glycoproteomics
Frédérique Lisacek, Kiyoko Aoki-Kinoshita, Marshall Bern, Sebastian Böcker, Robert Chalkley, Bernard Delanghe, Viktoria Dorfer, Patrick Emery, Rebekah Gundry, Michael Hoopmann, Lukas Käll, Neil Kelleher, Joanna Kirkpatrick, Daniel Kolarich, Lennart Martens, Nicki Packer, Magnus Palmblad, Daniel Questschlich, Veit Schwämmle, Matthew Smith, Sabarinath Peruvemba Subramanian, Morten Thaysen-Andersen, Lilla Turiák, and Mathias Wilhelm . . . 7

Working Group Report: Machine Learning (Kaggle) Competitions Based on Proteomics Data
Magnus Palmblad, Viktoria Dorfer, and Veit Schwämmle . . . 9

Working Group Report: Structural and Top-Down Proteomics
Daniel Questschlich, Bernard Delanghe, Viktoria Dorfer, Patrick Emery, Michael Hoopmann, Lukas Käll, Neil Kelleher, Magnus Palmblad, Veit Schwämmle, Matthew Smith, and Mathias Wilhelm . . . 10

Participants . . . 12

Remote Participants . . . 12

3 Overview of Talks

3.1 Topic Introduction: Shotgun Cross-Linking Mass Spectrometry and Protein Structure Prediction

Michael Hoopmann (Institute for Systems Biology – Seattle, US)

Crosslinking and bottom-up mass spectrometry (XL-MS) seeks to aid protein structure prediction and macromolecular structure assembly. The structural prediction community, however, has been slow to adopt and incorporate XL-MS technology. A recent study of CASP13 illustrated that only 12% of participants chose to compete using XL-MS. Better adoption of XL-MS for structural prediction requires improved quality and accuracy of XL-MS results and better computational pipelines for users to incorporate XL-MS into their research. We should consider how to improve the interaction between the XL-MS and structural prediction communities and accelerate the development of methods and pipelines that better integrate these technologies into robust computational tools.

3.2 Topic Introduction: Future Outlook & Opportunities for Top Down Structural Proteomics

Neil Kelleher (Northwestern University – Evanston, US)

In this brief orientation seminar, timely topics in computational proteomics as they relate to denatured and native mode top-down proteomics are presented. This includes the detection of proteoforms, their post-translational modifications (PTMs), and their complexes. Automation platforms for data creation and real-time search, processing new individual ion (i2MS) datatypes, and integration of compositional top-down proteomics with structural proteomics are also discussed. Importantly, the prospect of a Human Proteoform Project and Atlas was proposed, framed, and discussed.

3.3 Topic Introduction: Machine Learning for MS-based Proteomics

Lukas Käll (KTH Royal Institute of Technology – Solna, SE)

Currently, machine learning (ML) is revolutionizing the way we interpret data. Here I will give a brief background to classical ML. I will also point out how ML is helped by various deep learning structures that can learn feature representations of a sample point. Particularly, an encoder-decoder structure known as Transformers promises to change the way we handle sequential data, by enabling transfer learning.

Machine Learning, especially Deep Learning, requires non-trivial amounts of training data. Even though much proteomics data is available in repositories, it is not immediately accessible to ML. Röttger and colleagues (Rehfeldt 2021) recently uploaded a preprint describing how to transform proteomics LC-MS data in public repositories (e.g. PRIDE) to be used for ML.

We can also formulate some potentially relevant questions that we can ask in relation to ML in proteomics, with specific pertinence to this seminar:

- How can transfer learning reduce the need for training data in similar ML applications?
- How can ML be applied to glycomics, for example to predict chromatographic behavior and fragmentation of released glycans or glycopeptides?
- How can ML-based protein structure prediction be combined with top-down or bottom-up strategies for structural proteomics?

3.4 Topic Introduction: Glycoproteomics Challenges and Opportunities

Frédérique Lisacek (Swiss Institute of Bioinformatics – Genève, CH)

First, broad goals, topics of interest, and bottlenecks in the field of glycoprotoemics were identified. From here, three priority areas of potential discussion were defined. These priorities include: 1) outline a white paper that will serve as a tangible outcome of the forthcoming discussions; 2) envision how machine learning could be implemented to improve glycoproteomics analysis; and 3) define strategies to increase accuracy of glycoproteomics results. This latter element is a major challenge in the field of glycoproteomics, as there is a lack of established guidelines for assessing accuracy of search results. While sample preparation, data acquisition, and multiple data search tools have become increasingly accessible to many laboratories, the lack of expertise in basic principles of glycobiology can present challenges to accurate data reporting. Several ideas for increasing accuracy in glycoproteomic results were presented, including 1) strategies to integrate knowledge of biosynthetic pathways into routine data analysis processes, and 2) the need for FDR calculations suitable for intact glycopeptides.

4 Working groups

4.1 Working Group Report: Machine Learning in Proteomics

Lukas Käll (KTH Royal Institute of Technology – Solna, SE), Marshall Bern (Protein Metrics – Cupertino, US), Sebastian Böcker (Universität Jena, DE), Sven Degrove (Ghent University, BE), Bernard Delanghe (Thermo Fisher GmbH – Bremen, DE), Viktoria Dorfer (University of Applied Sciences Upper Austria, AT), Daniel Kolarich (Griffith University – Southport, AU), Frédérique Lisacek (Swiss Institute of Bioinformatics – Genève, CH), Magnus Palmblad (Leiden University Medical Center, NL), Robin Park (Bruker – Rancho Santa Fe, US), Veit Schwämmle (University of Southern Denmark – Odense, DK), Matthew Smith (University of Texas – Austin, US), and Mathias Wilhelm (TU München – Freising, DE)

This abstract summarises the progress made by the Machine Learning in Proteomics Working Group over the course of the entire seminar.

First, the most common scenarios of machine learning in proteomics and computational mass spectrometry were discussed, and from this starting point, future trends were envisioned, including the combination of different models for different acquisition methods, and the prediction of particular mass spectrometry and biological features such as distinction of isobaric glycans *via* retention time, prediction of enzyme activity, and disease association. Particularly, top-down mass spectrometry still faces several important challenges that could be facilitated by machine learning applications, which include prediction of charge distributions of intact proteins as well as specialized applications to decipher non-linear peptides and proteins (cross-linking, cyclic peptides or proteins, and protein ubiquitination).

Several examples of end-to-end prediction *via* machine learning, mostly through currently highly prolific deep learning approaches, were discussed. The most prominent of these were to determine peptide, charge, and modifications from fragmentation mass spectra. Moreover, it was noted that a lack of sufficiently simple use cases for non-proteomics experts are missing, which, if made available, could be used to challenge the machine learning community at large to participate in future developments and innovation. In addition, such use cases could also push existing proteomics informatics community efforts forward by allowing benchmarking studies to take place.

The combination of two common machine learning methods, namely spectral clustering and predictive machine learning, were extensively discussed. Relationships between fragment ions across, e.g., fragmentation techniques could be summarized into a generation function by using experimental and even predicted data that incorporates covariance patterns and thus the variability of the very different types of fragmentation spectra of a peptide as delivered through the various fragmentation techniques. Chimaeric spectra (which contain fragments from more than one fragmented precursor peptide) and modifications could also be easier to distinguish if such information were to be included in identification algorithms.

On the second day, six different topics were explored.

1. Embedding and clustering. Three tasks were discussed that should be achievable using a "simple" representation of a mass spectrum. These tasks were (i) make spectral clustering algorithms run much faster, (ii) improve the power for a particular application, and (iii) make spectral data more readily accessible to machine learning methods.
2. Data sets for competitions: Deep learning challenges in proteomics should be sufficiently simplified to attract the involvement of the broader machine learning community. We discussed two use cases that look into specific problems in peptide MS, and this specific sub-topic became the focus of a separate, spin-out Working Group on a proteomics-based machine learning competition. A distinct abstract is provided for this Working Group, and the interested reader is directed there for more detailed information.
3. Combining models: The discussion started on the differentiation between the development of a single model that covers multiple peptide properties *versus* the combination of multiple predictions via post-processing, and their respective use-cases. A related issue was raised in that some peptides appear to be eluting multiple times in the same chromatogram, and speculation ensued as to the associated consequences on prediction accuracy and downstream data analysis pipelines. The conclusion was that this topic deserves to be investigated in more detail going forward.
4. Metaproteomics: We discussed the different ways in which machine learning could be used in peptide and protein (family) identification, and pathway and gene ontology term enrichment analysis. It was decided that this is a very Interesting and potentially quite fertile topic, and that it will quite likely be possible to transfer machine learning approaches already developed in the sibling fields of metagenomics and metatranscriptomics to make inroads into this issue.

5. Protein inference: different protein inference strategies were discussed, with a particular focus on protein fluorosequencing. It was concluded that there still is ample room for improvement and for new methods to tackle this already well-established challenge. It was also considered at some length whether non-unique peptides (i.e., peptides that match to more than one potential originator protein) could be helpful at all in resolving protein inference, but there no consensus was reached on the utility of such peptides.
6. Reporting standards: the proposed DOME reporting guidelines for supervised machine learning were discussed in the context of mass spectrometry-based proteomics. A potential commentary on the DOME paper was outlined, which will interpret these guidelines specifically for the proteomics community.

Another topic of great interest, concerned the best method to encode mass spectra for downstream machine learning. Despite intense discussions on this topic, there was no consensus on what currently constitutes the optimal method for encoding mass spectra for machine learning. However, a number of potential improvements on existing, naive methods were suggested and discussed to move the field forward. It was also noted that spectral encoding, spectral distance metrics, and spectral clustering are all highly interrelated problems. This because every encoding implicitly suggests a distance metric and a clustering method. Clearly, this topic is worthy of more detailed study as well.

4.2 Working Group Report: Glycoproteomics

Frédérique Lisacek (Swiss Institute of Bioinformatics – Genève, CH), Kiyoko Aoki-Kinoshita (Soka University – Tokyo, JP), Marshall Bern (Protein Metrics – Cupertino, US), Sebastian Böcker (Universität Jena, DE), Robert Chalkley (University of California – San Francisco, US), Bernard Delanghe (Thermo Fisher GmbH – Bremen, DE), Viktoria Dorfer (University of Applied Sciences Upper Austria, AT), Patrick Emery (Matrix Science Ltd. – London, GB), Rebekah Gundry (University of Nebraska – Omaha, US), Michael Hoopmann (Institute for Systems Biology – Seattle, US), Lukas Käll (KTH Royal Institute of Technology – Solna, SE), Neil Kelleher (Northwestern University – Evanston, US), Joanna Kirkpatrick (The Francis Crick Institute – London, GB), Daniel Kolarich (Griffith University – Southport, AU), Lennart Martens (Ghent University, BE), Nicki Packer (Macquarie University – Sydney, AU), Magnus Palmblad (Leiden University Medical Center, NL), Daniel Questschlich (University of Oxford, GB), Veit Schwämmle (University of Southern Denmark – Odense, DK), Matthew Smith (University of Texas – Austin, US), Sabarinath Peruvemba Subramanian (University of Nebraska – Omaha, US), Morten Thaysen-Andersen (Macquarie University – Sydney, AU), Lilla Turiák (Research Centre for Natural Sciences – Budapest, HU), and Mathias Wilhelm (TU München – Freising, DE)

This abstract summarises the progress made by the Glycoproteomics Working Group over the course of the entire seminar.

During the first discussions, a few overall goals were outlined for the Working Group, including the delineation of the contents of a white paper on the current state of the field of glycoproteomics, an effort to integrate with the Machine Learning Working Group, and the definition of outstanding questions related to the bioinformatics in the field.

For the white paper, a few key topics of interest were quickly identified. A first was the need to allow the evaluation of the accuracy of glycoproteomics software, also by non-experts. Another was the need to provide coherent and intuitive data visualisations of the obtained results, which are currently not readily available. A large, unmet need was also identified concerning quantification, where statistical issues such as imputation difficulties and site-specific *versus* modified peptide differential analysis have not yet been addressed. Of course, there is also search space complexity, which is an already well-known problem, with various approaches in use to tackle this issue. It may therefore be relevant to perform a systematic evaluation of the respective benefits and drawbacks of these varied approaches.

As to the integration with the Machine Learning Working Group, it is clear that machine learning is currently having a profound impact on classical proteomics, and continues to make inroads there for some of the most complex problems. It will therefore be highly interesting to connect these efforts more closely with the glycoproteomics field, as there may well be similar benefits to be had here. In this context, the ongoing development, and increased adoption, of ion mobility in state-of-the-art mass spectrometers is a possible starting point for such an integration. However, it will be necessary to consider the creation of gold-standard data sets for this, or at least benchmark data sets for validation and evaluation of such efforts, alongside the necessary large amounts of reliable data needed for model training in the first place.

When discussing the bioinformatics developments, the focus shifted quickly to the integration of known biosynthetic pathways into the automated data analysis process. Currently, any successful analysis in glycoproteomics hinges heavily on the researcher's expertise in glycobiology. It is therefore important to consider whether it would be possible to introduce the principles of glycobiology into the search engines, for instance during the construction of the search space. Another approach that could be relevant would be to construct sample-specific glycan libraries, which could have the same (or even more stringent) effect. At the same time, the limited studies performed so far on unrestricted searches indicate that their performance is not as bad as typically thought, keeping that avenue open for exploration as well.

On the second day, the example provided by the field of top-down proteomics as presented in the corresponding introductory talk was considered. Here, instead of a single white paper, three independent opinion pieces at considerable impact had been written instead. As a result, the overall white paper concept was turned into the planning of three opinion papers focused on: 1) standards for glycoproteomics, 2) the reanalysis of (at least seven) published datasets of the SARS-CoV-2 spike glycoprotein, and 3) ways to address FDR calculation in glycoproteomics.

The content of 1) would span the different ways to optimize for, and ensure generation of, high quality data, while also describing the challenges involved with some of the standards; 2) would promote the multiplicity of methods and data; and 3) would cover the broad diversity of computational issues of intact glycopeptide identification, especially scoring functions.

Furthemore, a discussion was had on the possible input from machine learning into the field, and here several possibilities were proposed. First is the prediction of (relative) retention time prediction of glycosylated peptides and/or glycans. The goal would be to use these predictors as features in either a rescoring approach, and possibly to use these for isomer resolution. Another analyte (glycopeptide or glycan) behaviour to predict would be ion mobillity. Further avenues for possible machine learning input were fune-tuning of false discovery rate calculations, peak picking from raw data (as peak shapes do not follow typical peptide patterns), and fragmentation method optimisation.

The shorter session on the third day focused on the abovementioned list of issues to be discussed with the Machine Learning Working Group, covering several topics in more depth, including retention time prediction. Recent analyses of the HGI challenge data were discussed as an introduction to the topic of FDR calculation.

The final day was first dedicated to a review of the conditions for setting up community challenges. Then, in order to maintain continuity with points developed earlier, the contents of the anticipated manuscripts were detailed further . In particular, a back-to-back presentation of wet and dry glyco-lab issues was decided upon. Moreover, a vigorous discussion developed between the Glycoproteomics and Machine Learning Working Groups, with several participants of the latter joining the former. Much of the discussion focused on ways in which machine learning approaches could be used for relative retention time prediction to increase confidence in glycopeptide assignments, and how this could possibly even add a level of structural detail to the typical compositional information. Experts from both sides of the discussion asked and answered questions regarding the unique challenges associated with glycopeptides and machine learning approaches (one-to-many relationship of peptide to glycans; compositional *versus* structural considerations; features of machine learning that may enable retention time prediction independently of the variability in data acquisition strategies; solutions that work for low complexity samples may not work for high complexity). Strategies discussed include incorporation of iso-electric focusing, knock-out animal data, redundant data (glycoproteomics, glycomics, deglycosylated proteomics), and top-down proteomcis data. And while a consensus on how to solve the overall problems was not achieved, it was agreed that acquisition of data which can then be used for designing and testing machine learning approaches would be an important first step.

Finally, a detailed plan was outlined for a forthcoming manuscript focused on computational issues in glycoproteomics, writing assignments were distributed, and goals for the first follow-up meeting were defined.

4.3 Working Group Report: Machine Learning (Kaggle) Competitions Based on Proteomics Data

Magnus Palmblad (Leiden University Medical Center, NL), Viktoria Dorfer (University of Applied Sciences Upper Austria, AT), and Veit Schwämmle (University of Southern Denmark – Odense, DK)

This Working Group was convened as a spin-out of the Machine Learning in Proteomics Working Group, and focused specifically on the creation of machine learning competitions (as inspired by the Kaggle format) built around proteomics data. The underlying idea being that this will help enlist interest and innovation from the broader machine learning community.

In order to attract the broader machine learning community, deep learning challenges in proteomics should be sufficiently simplified. We therefore discussed in detail two use cases that look into specific problems in peptide mass spectrometry: the prediction of peptide observability (a challenge we nicknamed "SuperPeptide") , and the prediction of the triggering isotope from a fragmentation spectrum (a challenge we nicknamed: "Where did you hit me?").

These two challenges were devised to be posted on platforms such as Kaggle, and can furthermore be advertised throughout the proteomics community *via* organisations such as the European Proteomics Association (EuPA), the Human Proteome Organisation (HUPO), the European Bioinformatics Community (EuBIC), the International Society for Computational Biology (ISCB), and the Association of Biomolecular Resource Facilities (ABRF).

4.4 Working Group Report: Structural and Top-Down Proteomics

Daniel Questschlich (University of Oxford, GB), Bernard Delanghe (Thermo Fisher GmbH – Bremen, DE), Viktoria Dorfer (University of Applied Sciences Upper Austria, AT), Patrick Emery (Matrix Science Ltd. – London, GB), Michael Hoopmann (Institute for Systems Biology – Seattle, US), Lukas Käll (KTH Royal Institute of Technology – Solna, SE), Neil Kelleher (Northwestern University – Evanston, US), Magnus Palmblad (Leiden University Medical Center, NL), Veit Schwämmle (University of Southern Denmark – Odense, DK), Matthew Smith (University of Texas – Austin, US), and Mathias Wilhelm (TU München – Freising, DE)

This abstract summarises the progress made by the Structural and Top-Down Proteomics Working Group over the course of the entire seminar.

Early discussions in this working group focussed on how the different structural mass spectrometry techniques can be integrated with one another, but also more broadly with efforts in the wider structural biology community. In addition, needs for data formats and standardisation for cross-linking mass spectrometry and native mass spectrometry were examined. Moreover, the working group also set out to engage with the Machine Learning in Proteomics Working Group to delineate topics of mutual interest in cross-linking mass spectrometry.

A key discussion point that emerged from this overview, was the overarching theme of how the structural proteomics community should engage with the wider structural biology community. The discussion focussed primarily on cross-linking mass spectrometry and native top-down mass spectrometry strategies. One potential strategy that was explored was to join the Critical Assessment of protein Structure Prediction (CASP) experiments.

Another topic of importance to the structural and top-down proteomics communities relates to the detection of post-translational modifications, and their annotation on existing protein structures. Here, there are specific challenges as well, most notably the issue of having to distinguish between functional and bystander modifications, as both are readily observed in mass spectrometry. Another relevant issue is the determination of the stoichiometry of these modifications across proteoforms. There is also the specific case of proteins with two different conformers that are regulated by complexation or post-translational modifications. Such cases could be interesting targets for computational inference from a combination of native mas sspectrometry and cross-linking mass spectrometry.

Software needs and computational challenges in native mas sspectrometry and native top-down mass spectrometry were discussed in more detail. The first main topic related to the modes of software dissemination. Different, non-exclusive scenarios exist today, ranging from open-source packages over freeware tools, to for-profit software as released by small to

large companies. Specific mention was also made of the need to document available software well, and to provide adequate training opportunities and materials for end users to ensure uptake and proper use.

A delineation of similarities and differences in the acquired data and the analysis approaches employed was made between native top-down mass spectrometry and traditional top-down proteomics. The use of a combination of different types of mass spectrometry analysis for validation was explored as well. One option is to combine data from traditional bottom-up approaches (for instance, affinity purification mass spectrometry or even standard shotgun mass spectrometry), with data from cross-linking experiments, and furthermore add in native (top-down) mass spectrometry data. Conceivably, hydrogen-deuterium exchange mass spectrometry data could be included here too.

Starting points were also formulated for the standardization of data reporting for native mass spectrometry. These standards would need to take the form of standardized data formats, minimal reporting requirements, and relevant terminology in existing or bespoke controlled vocabularies. The Human Proteome Organisation's Proteomics Standards Initiative (HUPO-PSI) creates community standards in the field, but currently lacks strong representation from the native mass spectrometry community. It will therefore be important to motivate more researchers in this community to engage actively in such standardisation efforts. A related aspect is the ability to publicly disseminate native mass spectrometry data, which will require compatibility with proteomics repositories such as PRIDE/ProteomeXchange. This was followed by a lively discussion of what data will need to be recorded to allow the move from proteoform analysis to complexoform analysis.

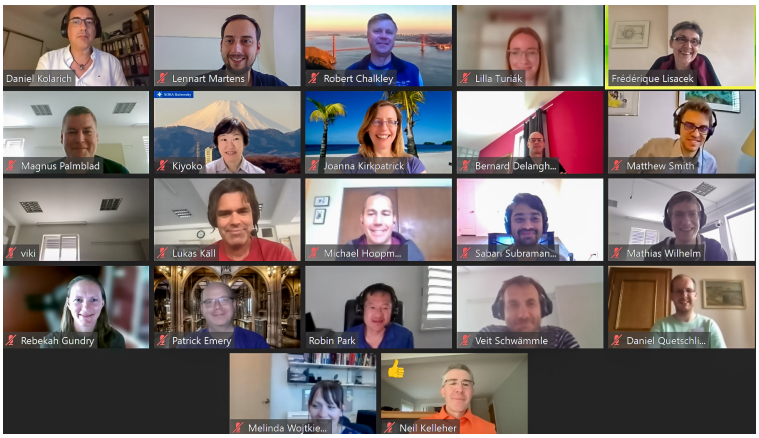
A final topic of discussion centered on ways in which data transfer and integration from structural proteomics experiments into protein knowledgebases like UniprotKB can be optimized.

Participants

- Sebastian Böcker
Universität Jena, DE
- Viktoria Dorfer
University of Applied Sciences Upper Austria, AT
- Lukas Käll
KTH Royal Institute of Technology – Solna, SE
- Frédérique Lisacek
Swiss Institute of Bioinformatics – Genève, CH
- Lennart Martens
Ghent University, BE
- Magnus Palmblad
Leiden University Medical Center, NL
- Robin Park
Bruker – Rancho Santa Fe, US
- Daniel Questschlich
University of Oxford, GB
- Veit Schwämmle
University of Southern Denmark – Odense, DK
- Mathias Wilhelm
TU München – Freising, DE

Remote Participants

- Jeffrey Agar
Northeastern University – Boston, US
- Kiyoko Aoki-Kinoshita
Soka University – Tokyo, JP
- Marshall Bern
Protein Metrics – Cupertino, US
- Robert Chalkley
University of California – San Francisco, US
- Sven Degrove
Ghent University, BE
- Bernard Delanghe
Thermo Fisher GmbH – Bremen, DE
- Patrick Emery
Matrix Science Ltd. – London, GB
- Rebekah Gundry
University of Nebraska – UOmaha, US
- Michael Hoopmann
Institute for Systems Biology – Seattle, US
- Neil Kelleher
Northwestern University – Evanston, US
- Joanna Kirkpatrick
The Francis Crick Institute – London, GB
- Daniel Kolarich
Griffith University – Southport, AU
- Rune Linding
HU Berlin, DE
- Nicki Packer
Macquarie University – Sydney, AU
- Matthew Smith
University of Texas – Austin, US
- Sabarinath Peruvemba Subramanian
University of Nebraska – Omaha, US
- Morten Thaysen-Andersen
Macquarie University – Sydney, AU
- Lilla Turiák
Research Centre for Natural Sciences – Budapest, HU
- Olga Vitek
Northeastern University – Boston, US
- Christine Vogel
New York University, US


Daniel Kolarich
Lennart Martens
Robert Chalkley
Lilia Tunák
Frédérique Lisacek
Magnus Palmblad
Kiyoko
Joanna Kirkpatrick
Bernard Delangh...
Matthew Smith
viki
Lukas Käll
Michael Hoopm...
Sabari Subraman...
Mathias Wilhelm
Rebekah Gundry
Patrick Emery
Robin Park
Veit Schwämmle
Daniel Quetschli...
Melinda Wojtkie...
Neil Kelleher


Frédérique Lisacek

Report of Dagstuhl Seminar 21272

Towards Climate-Friendly Internet Research

Edited by

Vaibhav Bajpai[1], Oliver Hohlfeld[2], Jon Crowcroft[3], and Srinivasan Keshav[4]

1 **TU München, DE,** bajpaiv@in.tum.de
2 **Brandenburg University of Technology, DE,** oliver.hohlfeld@b-tu.de
3 **University of Cambridge, GB,** jon.crowcroft@cl.cam.ac.uk
4 **University of Cambridge, GB,** sk818@cam.ac.uk

Abstract

This report presents guidelines for deciding when virtual or hybrid conferences are suitable and how to design them. The report is the output from a Dagstuhl seminar where the goal was to review the current status of virtual conferences and to develop best practices for hybrid conferences. The participants provided input on the state-of-the-art of virtual conferences: what works, what does not, and what needs improvement. From this discussion, the participants discussed the requirements, implications, and guidelines for designing hybrid conferences. The participants felt that in the future, small research meetings will move entirely online whereas larger ones will be held as hybrid events.

Seminar July 6–9, 2021 – https://www.dagstuhl.de/21272
2012 ACM Subject Classification Networks; Social and professional topics
Keywords and phrases Carbon Footprint, Energy Efficient Networking, Climate Change
Digital Object Identifier 10.4230/DagRep.11.6.14

1 Executive Summary

Vaibhav Bajpai
Jon Crowcroft
Oliver Hohlfeld
Srinivasan Keshav

Goals

The Internet was originally developed to ease collaboration between remote parties, thereby, in principle, reducing carbon emissions by a reduced need for travel. Yet, conducting research on communication networks has typically involved a certain level of carbon footprint. One fundamental reason is the publication and dissemination culture in the field, which focuses on conferences and workshops rather than journals. Not only does every dissemination of a research result therefore involves travel, even the peer-review process to decide which papers to accept, in the form of an in-person technical program committee (TPC) meeting, also requires travel. Moreover, although the standardization of Internet technology within the Internet Engineering Task Force (IETF) largely involves online discussions and audio/video streaming–unlike almost all other standardization bodies–yet regular in-person meetings

are considered critical to converge discussion and build consensus. Thus, conducting and disseminating networking research has resulted in a high level of travel, and a consequent high carbon footprint.

The carbon footprint of these trips (mostly air travel) can, however, be reduced by means of organizational changes and virtual conferences. Recently, as a consequence of the COVID-19 pandemic, we have already witnessed a rapid transition to a virtual mode of operation including remote working, online meetings, and virtual conferences. This has resulted in first-hand experience in carrying out research but with no travel.

In this Dagstuhl Seminar, we initiated a discussion on how to make Internet research more climate friendly. Specifically, we evaluated experiences in running and participating in virtual conferences as a consequence of the COVID-19 pandemic. We wanted to understand what went well and what went badly in implementing and deploying virtual conferences, what challenges were encountered, and what needs to be improved, particularly as we transition to hybrid in-person, online meetings. The broader goal of the seminar is to identify how to transition to a new status quo that continues to reduce the carbon footprint from travel.

Structure

The seminar lasted 2.5 days. It began with an introductory round where each participant presented one slide to give an overview of their experience that was relevant for the seminar and a set of open questions that the participant wished to discuss during the event. These slides were collected from each participant before the seminar. We also had one invited talk (§3.1) that we used as a basis for triggering discussions and identifying areas for group work, while a major portion of the seminar time was dedicated to breakout sessions, whereby participants were split into small groups to discuss specific themes and develop ideas with consensus to propose to larger groups. The morning sessions the following day were dedicated to continuing parallel group work with presentations that reported the outcomes of each breakout session from the previous day. Every evening, we had an online social activity. The afternoon of the third day was spent reviewing and collecting feedback from the participants and for initiating follow up actions identified during the seminar.

2 Table of Contents

Executive Summary
Vaibhav Bajpai, Jon Crowcroft, Oliver Hohlfeld, and Srinivasan Keshav 14

Overview of Talks

Invited Talk: Virtual Conferences (and Climate Change)
Cristina Videira Lopes 17

Retrospective on Online Operation in 2020-2021

Guidelines (Best Practices) for Online Conferences
Srinivasan Keshav, Franziska Lichtblau, Andrew Hines, Henning Schulzrinne, and Michael Menth 18

Financial, Diversity, & Timezone Implications of Online Events
Mirjam Kühne, Jon Crowcroft, Cristel Pelsser, Amr Rizk, and Vaibhav Bajpai . . . 21

Lessons Learned from Online Everything (Group 1)
Georg Carle, Alexander Raake, Oliver Hohlfeld, Colin Perkins, Cristina Videira Lopes, Jörg Ott, Quentin De Coninck, Simone Ferlin, and Jürgen Schönwälder . . 22

Lessons learned from Online Everything (Group 2)
Colin Perkins, Georg Carle, Cristina Videira Lopes, Oliver Hohlfeld, and Jörg Ott 23

Guidelines for Designing Hybrid Conferences

Technical and Social Barriers to Hybrid Conferences
Franziska Lichtblau, Daniel Karrenberg, Jörg Ott, Mirja Kühlewind, and Vaibhav Bajpai 26

Requirements for Hybrid Conferences (Group 1)
Andrew Hines, Colin Perkins, and Mirjam Kühne 27

Requirements for Hybrid Conferences (Group 2)
Jon Crowcroft, Quentin De Coninck, Jari Arkko, Georg Carle, Alexander Raake . 28

Financial, Diversity, and Timezone Implications for Hybrid Events
Henning Schulzrinne, Srinivasan Keshav, Cristel Pelsser, and Sujata Banerjee . . 30

Hybrid Everything: Colloquiums, Hackathons & Research Visits
Amr Rizk, Oliver Hohlfeld, Michael Menth, Jürgen Schönwälder, and Simone Ferlin 33

Conclusions and Next Steps 35

Remote Participants 37

3 Overview of Talks

3.1 Invited Talk: Virtual Conferences (and Climate Change)

Cristina Videira Lopes (University of California – Irvine, US)

Cristina Videira Lopes (UC – Irvine) kicked off the discussion by presenting general aspects of virtual conferences (and climate change). The abstract of her talk was as follows:

"For the past 40 years, research communities have embraced a culture that relies on physical meetings of people from around the world: we present our most important work in conferences, we meet our peers in conferences, and we even make life-long friends in conferences. Also at the same time, a broad scientific consensus has emerged that warns that human emissions of greenhouse gases are warming the earth. For many of us, travel to conferences may be a substantial or even dominant part of our individual contribution to climate change. A single round-trip flight from Los Angeles (LA) to Frankfurt emits the equivalent of about 3.3 tons of carbon dioxide (CO2e) per passenger, which is a significant fraction of the total yearly emissions for an average resident of the US or Europe. Moreover, these emissions have no near-term technological fix, since jet fuel is difficult to replace with renewable energy sources. In this talk, I first raise awareness of the conundrum we are in by relying so heavily in air travel for our work. I will present some of the possible solutions that go from adopting small, incremental changes to radical ones. The talk focuses on one of the radical alternatives: virtual conferences. A year and a half of pandemic has given us a fast introduction to virtual conferences, with mixed results. I am part of a community that has been organizing an annual conference in a virtual environment for many years. Virtual conferences present many interesting challenges, some of them technological in nature, others that go beyond technology. Creating truly immersive conference experiences that make us feel "there" requires attention to personal and social experiences. Those experiences need to be recreated from the ground up in virtual spaces. But in that process, they can also be rethought to become experiences not possible in real life."

4 Retrospective on Online Operation in 2020-2021

Participants were requested to bring one slide to provide their perspective on the topic and all slides were combined together into a block to gather input and for triggering discussions and identifying areas for breakout sessions.

Parallel Group Work

The afternoon sessions were used to discuss certain topics in more depth in smaller groups. This section summarises the discussions of each group.

4.1 Guidelines (Best Practices) for Online Conferences

Srinivasan Keshav (University of Cambridge, GB), Franziska Lichtblau (MPI für Informatik – Saarbrücken, DE), Andrew Hines (University College Dublin, IE), Henning Schulzrinne (Columbia University – New York, US), Michael Menth (Universität Tübingen, DE)

The breakout came up with guidelines for both traditional parts of the conference and social aspects (structured chaos) as described below.

To begin with, we felt that online conferences are different from in-person conferences. As such, it is not a goal that conference participants have an experience as close to a physical conference as possible. Instead, conference participants should be able to learn of new work in their areas of interest, meet with other participants in semi-structured interactions, and be able to present their work both formally and informally to others. These goals should be achieved using tools and procedures that may differ from that used in traditional conferences, but should lead to the same outcomes. To this end, we need to accept that online conferences will never match every aspect of a physical conference, especially for face-to-face (small group and individual) meetings. However, in some areas, they may actually be better than physical meetings, for example, in widening participation. That said, at the moment, there is no silver bullet for online conferences. As such, we need to build on existing tools.

4.1.1 Scheduling

Online conferences require us to revisit the traditional conference schedule with 20-30 minute paper presentation slots and a single timezone. Deciding the schedule is one of the most critical decisions facing conference organizers. Further, the organizers should keep the agenda on time and communicate with participants if there are technical issues.

The goal should be for the conference schedule to provide an overall framework for all conference events. It should allow participants to meet the twin conference goals of learning and interaction. There are three main issues:

- *Multiple time zones:* With multiple time zones, no single participant will be present all the time. So, it is necessary to create structures that allow interaction across time zones. Think about how to maintain continuity for people who come in and out of the conference. Moreover, there may be only a limited number of hours when all participants are present. This time should be used wisely for plenary sessions such as keynote talks, or best papers.
- *Zoom fatigue:* Day-long programs don't work. It is necessary to compress schedules, with perhaps a four-hour limit for each day. We recommend that organizers reduce the number of papers presented, with only the best presented. The rest of the papers can be made into posters or videos that people could watch at their convenience. An alternative is to create a multi-track conference, which has its own challenges and is still online.
- *Limited ability to focus online:* 8-minute talks with Q/A seems to work well, especially with pre-recorded talks. The talks present only the problem statement/conclusion, with details in the paper. Not everyone has liked this approach lately though.

4.1.2 Navigation and signposting

The conference schedule should make it easy for participants to learn about and join or rejoin events. Currently, some program sites are overfilled with information, making it difficult to find papers or events. Signposting and a clearly structured landing page is necessary to avoid

this problem. It would be helpful to be able to jump to the breakout or session with one click. This will require deep linking, which is currently not possible with Zoom breakout rooms. We strongly recommend that the conference schedule explicitly shows participant-tailored time zones so that a participant in each timezone knows exactly where to go.

4.1.3 Poster sessions

Poster sessions can be challenging to hold online. In-person poster sessions allow one to quickly scan a lot of work, with the option to dive deep, or move past. The main issues are:

- *Social awkwardness*: Current solutions do not provide quick skimming and make it awkward to leave if the content is uninteresting.
- *Hard to navigate*: A mechanism is needed to find interesting posters or move across mixed mediums (for example from Zoom to Hubs – specialised spatial-metaphor tools.)

Recently, we have gained some **experience** with online poster events particularly using three different solutions. *Mozilla Hubs* for instance, allow for bi-modal feedback in poster sessions. With *Gather.town*, on the other hand, it is difficult to identify neighbors, i.e., the author or another person standing next to the poster. Finally, *Spatial.chat* lacked good audio quality and overall felt not fit for poster sessions. Some of the **suggestions** when using these tools are listed below:

- A quick skimming is important, since it allows a walk by for a quick yes/no decision. While everyone sees that the person walks in and out. There is a need to increase social ease by openly stating at the conference poster session that it is okay to leave a room.
- Poster sessions should use breakout rooms with one breakout per poster. People can "walk by" and reduce the number of people per session to allow for a more personal interaction with the poster presenter.
- Speed-dating style approaches maximize the use of time, with an excuse to move on. This establishes clear rules, helpful especially for younger community members. Some conferences have also followed a one minute madness approach with an opportunity to arrange longer times if necessary.

4.1.4 Structured chaos

One particularly challenging aspect online conferences struggle to replicate is social and hallway unstructured conversations. These have not been solved by current tools. The main issue here is how to get seed conversations going and also get the conversations to further develop at the conference. There are several motivations, for instance, to catch-up to people, strengthening existing bonds, making new introductions to interesting or important people, building the community and renewing existing relationships, identifying potential research partners, other opportunities and finally recruitment and job hunting.

Although we haven't yet found a perfect solution for social interaction, there are some positive experiences; for instance, for short coffee breaks, dropping people into breakout rooms at random works well. We think there is a need to explicitly identify "social butterflies" who can actively promote social interaction and start off the conversations. This challenge mirrors quite a bit in experiences with online teaching; for example, breakout sessions for students in a class have similar problems. As such, there is a need to bring willing participants together, who want to interact, but also not too many or too few and there is a need to strike a balance. Some difficulties and suggestions to this end are listed below:

- How to decide who starts the conversations? A couple of examples of questions that can be used as conversations starters: "Tell me about your work" or "You use tool/technology X, what do you think of it" or "What have you been up to recently"
- How to ensure implicit social behaviour is made explicit? There is a need to assign people roles so that they do not feel awkward: Explicitly approach specific senior members of the community to steer the communication. At the same time, how to avoid toxic behavior and egotists? We need to make a careful selection of people for dedicated roles.
- How to facilitate cross-pollination? This needs to be explicit perhaps with a special newcomers meeting event, where there is a chance to meet old timers. To this end, how to strongly encourage senior members to participate (such as in SIGCOMM student dinners)? Social interaction can be promoted by assigned seating in physical meetings or by joining a table even if you know no one there, chiming in the conversation is okay. However, the issue of how to balance people in meetings (half known, half new) is still uncharted territory. How to (actively) bootstrap chaos? Perhaps this can be done using social enabler tools and senior community members.

4.1.5 Text channel

A text channel emerged as a good idea for coordinating Q&A and general discussions. Traditionally, the Q&A session at the end of a talk serves as a 'community peer review' tool. Questions provide additional context for the work or expose lacunae that the reviewers did not catch. However, such a session can be somewhat intimidating for shy presenters. A text channel-based Q&A session allows them to participate. It also allows author responses to be captured, unlike the situation in a typical physical conference. Some guidelines when using Slack or similar text channel are outlined below:

- Session chairs need to be strict in enforcing discipline in Slack to prevent discussions from wandering. It is helpful to have a moderator or scribe to capture the Q&A content and turn that into a report published later (with the consent of the relevant parties, who have veto power). It might even be possible for scribes to report on "bits and bytes from the previous day" as is done in RIPE meetings.
- The audience can be encouraged to make use of special markers such as @ to notify authors for pending questions. Authors should be told that questions on Slack should be answered within 24 hours. It is not a good idea to have a generic channel with too much chatter, so one does not know who has to answer. One Slack channel per session is better, though there is still a need to find questions for each paper. On the other hand, one Slack channel per paper has too much granularity, making it difficult to find which channel to attend.

An alternative to Slack is Slido, which allows questions to be posed and voted on, especially for large audiences. We suspect that Slido would be useful for hybrid conferences as well.

4.1.6 Audio, Video and Lighting issues

Audio for virtual events is better than real life for some people, since it allows lip-reading and individual adjustments of audio level. Nevertheless, despite the experience from 2020-2021, bad audio and video quality continues to be a problem. Audio issues are not only serious ('I have no audio'), but more subtle, such as issues with noise, echoing, and audio level. Automated testing of audio intelligibility might help. Alternatively, conferences should provide test sessions for interactive sessions such as panels and keynotes. Meanwhile, testing video submissions in advance of the conference is a good idea, since there are still problems

such as videos that do not work either on Mac or Windows or require specialised tools. An open research topic would be to use automation to judge quality of submitted videos. Finally, lighting can be an issue, especially back-lighting, requiring participants to require prior guidance on how to avoid problematic lighting.

4.2 Financial, Diversity, & Timezone Implications of Online Events

Mirjam Kühne (RIPE – Amsterdam, NL), Jon Crowcroft (University of Cambridge, GB), Cristel Pelsser (University of Strasbourg, FR), Amr Rizk (Universität Duisburg-Essen, DE), Vaibhav Bajpai (TU München, DE)

Costs for online events – There are several costs for online events, namely – meeting platform (such as Zoom, Meetecho, WebEx); although many groups or universities already have licenses, social platforms (such as Gather.town or SpatialChat); meeting give-aways (tokens and T-shirts); stenographers and real-life captions, and finally simultaneous translations, to name a few of the tangible costs.

Being transparent about costs is important – There are some costs for conferences that are hidden in the publication costs of research papers. Established researchers have begun to use free research channels, but the issue exists, because younger researchers have to publish in well-established venues (that charge fees) to build up their CV for instance. As such, financial and business models will have to change. Conferences (and professional societies) who rely on conference fees will have a problem. There are currently several revenue streams for events, (some of which are also used to cover other costs), namely conference registration fees, sponsorships, access to research papers and carbon offsetting whereby some parts could be used to cover costs of events. Organisers could also help to promote environmental projects (which is good for reputation of the event.)

Sponsorships – There seems to be a hesitancy in sponsoring online events by sponsorship organisations. However, visibility still serves a good motivator for sponsors. Meanwhile, other sponsorship benefits have to be found. Further, organisers need to think hard on how to facilitate one-on-one conversations for recruiters, sponsors and peering coordinators in online settings.

Travel funds – It was unclear why and whether would employers fund travel and conference attendance when there is already possibility to attend online for free. It is possible that new participants might experience problems getting funding in the future. To this end, organisations may need to rethink and re-purpose travel funds and scholarships for some.

Conference local hubs – Some large meeting venues (such as the IETF) are proposing to run local hubs in addition to being online. The associate costs for running such local hubs is presently unclear, however, such initiatives could also help people from low-income groups to eventually participate.

Diversity – Online conferences help improve diversity since they encourage participation from attendees who cannot afford travel. Further, online archiving helps broader access to the conference material. Some large venues (such as RIPE) offer stenography to help with inclusion. Meanwhile, smaller (local) events can also be run in local languages and to promote and strengthen local communities. Some venues are also offering child care for attendees to ease participation of parents.

Time zones – SIGCOMM 2020 and 2021 followed a model of pre-recorded presentations together with multiple Q&A sessions for different time zones. Meanwhile, the IETF follows a model of aligning to the timezone of the local venue. It is unclear which model is better or whether one community can easily adapt to the model of the other, since at some venues (such as the IETF) the focus is more on forming consensus and less on presentations. Collaboration that comes naturally with physical settings becomes tricky in online-only mode when participants join from different continents. One option is for conferences to span several weeks with shorter (say two hours per day) venue slots. The focus can also be shifted more towards online interim (topical) meetings rather than concentrating on one or two big events per year.

4.3 Lessons Learned from Online Everything (Group 1)

Georg Carle (TU München, DE), Alexander Raake (TU Ilmenau, DE), Oliver Hohlfeld (BTU Cottbus, DE), Colin Perkins (University of Glasgow, GB), Cristina Videira Lopes (University of California – Irvine, US), Jörg Ott (TU München, DE), Quentin De Coninck (University of Louvain, BE), Simone Ferlin (Ericsson – Stockholm, SE), Jürgen Schönwälder (Jacobs University Bremen, DE)

Types of online meetings that we attended – The group has participated in a broad range of different online meetings. For one, Technical Program Committee (TPC) meetings where meetings for lower-tier venues were traditionally held online or via the phone, while top-tier venues had, by tradition, typically in-person TPC meetings that are now held online. Research visits to other research groups are another variation, where researchers known to a group made a visit while not being physically present at the remote location. This led to joining in-person group meetings and day-to-day discussions to be run largely online. Finally some experiences were gathered with conferences and workshops (virtual and hybrid) and with project meetings.

Experiences with online teaching– Since the COVID-19 pandemic forced universities to move all their teaching activities online, extensive experience with online teaching now exists. We highlight some of the experiences from the past year.

Firstly, online teaching generally can lead to multiple outcomes. First, better grades might be possible. Participants that take the exam are highly motivated while others drop out before and if videos are provided, they can be watched repeatedly. Yet, online teaching sets higher requirements when it comes to self-management and dedication, thus the dropout rate can be higher too (i.e., fewer students register for the exam) and consequently the number of participants can decline over time. Secondly, in a live lecture that is provided as video stream (not pre-recorded), it is usually hard to capture when participants get lost. This may happen in the beginning already (some approaches to catch this in text channels for Q&A, e.g., Slack, exist though). As such, having a dedicated channel for posting questions (e.g., Slack or Tweetback) – even anonymously – that are later sequentially addressed by the lecturer was perceived to work very well. This, however, requires further human resources such as a teaching assistant (TA) that handles the questions. It is hard for a lecturer to give the lecture and follow the chat simultaneously. Thirdly, if and when video recordings are offered, the lecture auditorium lacks sufficient physical presence as

before. Yet, many participants believe that asynchronous teaching material (e.g., videos) will be the future, e.g., explanations of an algorithm can be viewed multiple times, as mentioned before. The most difficult part in online courses are lab sessions, in particular if students need access to lab hardware. For all other cases, virtualization and remote access works well.

Technical Program Committee (TPC) meetings– TPC meetings for lower-tier venues were traditionally held online or via the phone. Top-tier venues had, by tradition typically in-person meetings that are now (during the pandemic) held online as well. In the past (pre-COVID-19 times), some venues organized physical TPC meetings. Meanwhile, TPC meetings are now often held online. They work very well when everyone is prepared for the meeting. However, if there is no travel, researchers tend to over-commit with meetings, but like with other meetings, TPC-meetings are usually hard to squeeze into overall schedule. This is simply a matter of habit, not an issue with online meetings per se. With online TPC meetings, what has worked well is handling conflicts of interest. At an in-person TPC meeting, conflicts need to leave the room (i.e., every few minutes TPC members leave and re-enter the room). In an online environment, conflicts can be sent to a breakout room and easily moved back, which smooths the process. Meanwhile, accessibility of online meetings has (and should be) increased also since no financial participation is required for travel.

Project Meetings – Two categories of project meetings exist: *Administrative* meeting such as general assembly or EU project review meetings in Brussels. Having these meetings online has not made things worse. The second kind of meetings are the *preparatory* meetings to get the project going such as to get teams to start working together, doing content-related work, build community within a given project (no social activities, but still due to different types of contacts). To this end, what helps is the social need that participants have to move things forward, although such interactions are very people-dependent.

4.4 Lessons learned from Online Everything (Group 2)

Colin Perkins (University of Glasgow, GB), Georg Carle (TU München, DE), Cristina Videira Lopes (University of California – Irvine, US), Oliver Hohlfeld (BTU Cottbus, DE), Jörg Ott (TU München, DE)

What did not work well online – Online PhD defenses are sad. They function but lack the celebration aspect which makes it a very unpleasant experience for the candidate.

When it comes to teaching, recording online lectures is a huge time sink. Teaching also feels as if it is performed into the void with no received reactions as to whether the presented content is being understood or whether listeners are falling asleep.

Online meetings on the other hand face their own issues. They can generate churn as participants join and leave. Participants might also leave the computer and stay connected making it hard to identify who is present. There is also a tendency for people to over-do/commit the number of online meetings they attend. Too many meetings also lead to fragmentation and eventual loss of context. Coupled to that, without proper calendar invites, finding meeting information (links, passwords, ...) in emails can sometimes also

become tricky. Time-zones further complicate scheduling and limit available meeting options. Generally, it is also hard to quantify "missed opportunities", but it seems attempts to simulate the in-person experience usually never works.

Current online tools also provide no way to capture social cues. For instance, when it would be okay to interact with participants and when not (e.g., when they are paying attention and are not open to talk). This is very easy in an in-person setting and currently impossible in a virtual format. During in-person meetings, one typically talks to their neighbours, while in online meetings, everyone is a neighbour. As such, the question is whom should you talk to? There currently also exists platform bloat – too many platforms (Where do we meet?, What is a shared platform that everyone has installed?) – When scheduling meetings, this information needs to be captured along with the available times to meet. Scheduling a meeting just becomes a bit harder. There is also no subconscious signal as to what platform needs your attention – one needs to actively check them. It remains unclear whether this increases the cognitive load.

When it comes to social activities, "forced fun-on-demand" is hard. Social activities work if they are prepared, e.g., a birthday party where wine is shipped to everyone or conferences where the ingredients for the social (e.g., mixology at IEEE QoMEX) are made available to the participant before. This is, however, very participant dependent.

What worked well online – Technical Program Committee (TPC) meetings seem to work well in general when held online and can be handled very efficiently. For example, when handling conflicts of interest as discussed before. Group work works well too by using random assignments to breakout rooms. In this mode, participants are assigned to smaller groups (e.g., up to 4 people) at random by using breakout rooms. There are two examples. First, (panel) discussions in smaller groups (e.g., IEEE QoMEX 2021) and Dagstuhl style group discussions. Secondly, getting to know new people by randomly assigning conference participants to smaller breakout rooms works well, some online venues have used this mode for their social activities. Project meetings work just fine, since people know each other. Maybe more productive online than in-person since discussions are more focused with fewer disruptions. The downside here is that people tend to meet too often or schedule too many meetings. Interactive discussion with speakers during talks also work well, but may lead to burnout if they run for too long. Stopping by a conference for just a single session is possible online since no travel is needed. This is a real benefit of virtual conferences. Meanwhile, pre-recorded presentations become part of the proceedings and are although (mostly not as permanently) archived just like papers. This is a real benefit for the scientific community. Q&A discussions work better in online mode, too – more questions are being asked by junior people; the hypothesis is that online is less intimidating than standing in front of a mic. Shared editing of reports is also possible; in an in-person meeting, it is typically considered a bad habit to use a laptop during the meeting, so online note taking is less common. In an online meeting, the notes are just a window next to the video conference. Online mode also opens new meeting opportunities since it is very cheap (also time wise) to interact with new communities that one normally would not attend. Online birthday parties can also work – for example by ordering a bottle of wine or pizza to each participant – same wine and food for everyone creates a joint experience. Playing online interactive games (e.g., escape room) can also provide an immersive real-world experience.

Work life balance and health in general is challenged by online meetings – All participants considered that online meetings *can* challenge work life balance more easily. Preparing digital teaching material and online teaching in general takes much longer

(some participants reported up to ten times as long) as in-classroom teaching. In general, all participants reported that their work became more intensive since more meetings are being scheduled. This is, for example, reflected in the typical gap between meetings: the gap between in-person meetings is five minutes, between online meetings five seconds as many meetings start and end on the hour. Consequently, this more intense schedule can lead to health issues since people move less, e.g., don't leave their chair for ten hours.

What did we learn? – Unstructured activities do not work well online, e.g., random encounters during coffee breaks at conferences. Also creative parts of in-person meetings, e.g., during ITU meetings with side discussions, do not work well in the online world. However, structured activities work very well online. For example, online meetings are more focused with less distractions and are thus very time efficient. On the other hand, less distractions also means no unstructured activities such as no random encounters after a project review, PhD defence or a TPC meeting. A general question concerns how to lower friction? Lower friction activities happen easily online, while higher friction activities get missed. Friction can also be increased artificially. One can consider fetching and sending emails only once per day. This increases the minimum RTT of email noticeable to others and thereby helps to focus on getting work done.

In general, different meetings have different requirements. If online meetings are successful mainly depends on these requirements. Certain meetings do achieve their goal if the agenda is fulfilled, and thus can work very well online. Other meetings have important goals beyond the specific agenda: can be challenging online.

Main takeaways – Different types of meetings have different requirements and audiences. As such it is important to be goal-oriented – structured activities work well online, when the tools meet the needs of the meeting. Meanwhile, unstructured activities (whiteboard-style idea creation, random encounters) do not work well (e.g., what happens after a PhD defense). Online meetings are more focused, have fewer distractions (examples: panels, PhD defenses) but lack the overall social cues.

5 Guidelines for Designing Hybrid Conferences

Participants were requested to bring one slide to provide their perspective on the topic. These slides were combined to trigger discussions and identify areas for breakout sessions.

Defining Hybrid Conferences: A Terminology

Henning Schulzrinne proposed the following terminology for hybrid conferences that the group agreed to adopt in its further discussions:

1. **Passive (inactive) Hybrid** – This model allows only passive remote participation by making videos of talks, demos, panels available to both local and remote attendees. The material can be recorded ahead in time. In this model, decent Internet connectivity is necessary to remotely access the material and therefore could be an issue in regions that censor the Internet in different ways.
2. **Semi-passive (semi-active) Hybrid** – This model supports a limited degree of remote participation including questions, thereby running in a "webinar" mode of operation. The prerequisites are requirements for (1) plus decent audio equipment for interactive

presentations. In this model, capturing local audio could become an issue. The model also risks trolling behaviour from anonymous remote participants during the Q&A. As such, lightweight training is needed for session chairs to handle such cases. Yet another issue is how to implement turn-taking with such a mix of (online/presence) participants. A possibility of professional stenography for speakers can help with written material.

3. **True (fully active) Hybrid –** In this model, both presenters and audience can be either local and remote. The IETF has had experience with such a model, whereby virtual queuing was implemented using QR codes, but it was found that such schemes also break flow. Eavesdropping in online mode is an issue. For small side meetings, traditional Skype also works. It is unclear how to implement two levels of social interactions – one for each mode of participation and whether it would work at all. Yet another concern is how mentoring (and matchmaking of senior academics to students) would work.
4. **Distributed Hybrid –** In this model regional in-person clusters or hubs are created with a shared program and viewing parties. In such a mode, travelling to local hubs has a carbon cost but it is to be explored whether the experience is closer to attending a traditional in-person conference. The Chaos Computer Club (CCC) has been running local hubs for a while, but the experience has not been too positive. On the other hand, running multi-site conferences have the risk of ending up with a multi-conference experience. As such, local hubs still have the advantage of socialising with people at a smaller scale at a much more personal level due to localised nature of languages as well.

Parallel Group Work

The afternoon sessions were used to discuss some selected topics in more depth in smaller groups. This section summarises the discussions of each group.

5.1 Technical and Social Barriers to Hybrid Conferences

Franziska Lichtblau (MPI für Informatik – Saarbrücken, DE), Daniel Karrenberg (RIPE – Amsterdam, NL), Jörg Ott (TU München, DE), Mirja Kühlewind (ERICSSON Eurolab – Herzogenrath, DE), Vaibhav Bajpai (TU München, DE)

Technical barriers – At the moment, the tools themselves appear to be technical barriers. Some of the challenges with current tools include remote and in-presence queue management, ability to see all remote participants at once, or conversely cannot fully see the physical room when remote. Reading the chat and speaking at the same time is a hurdle. Similarly using *Gather.town* for hybrid events presents its own challenges such as how to search for specific people and whether technologies such as "find my ..." or a "tile" attached to conference badges are needed. Such technologies also open privacy concerns and the willingness for attendees to use them. It also opens up challenges on how to synchronise the avatar of an in-person participant as they move physically in the real-world and whether such avatars really work unless they are made fully immersive since latency is also a barrier to immersive interaction.

In terms of equipment – the IETF has used whiteboards before. Meanwhile, online teaching has recently used projections of physical whiteboards during the pandemic times. However, the overall question still is whether we need to adapt to a virtual world? (or) make the virtual world better to mimic the physical world?

Social Barriers – Experience has shown that sustaining creativity in online-only modes has been difficult to achieve. The question is whether we can sustain creativity in hybrid modes? Maybe a new technical environment (using a phone instead of a laptop) is needed to implement social meetings? Large physical coffee breaks usually create the possibility to talk in small groups, but this is really hard to imagine implementing in large Zoom coffee breaks. Artificial background noises (e.g., rain) may help to create some sense of the physical environment, but the problem largely remains unsolved.

5.2 Requirements for Hybrid Conferences (Group 1)

Andrew Hines (University College Dublin, IE), Colin Perkins (University of Glasgow, GB), Mirjam Kühne (RIPE – Amsterdam, NL)

It is assumed that all hybrid events will have a structured and an unstructured component, whereby the social interaction could happen in smaller groups on local hubs. The group focused on the universal requirements for any hybrid conference:

Platforms and technology – One key aspect is good audio. The question here is how to capture good audio from the in-person participants and how to ensure remote participants can clearly be heard. The ability to quickly isolate points of failure and assign responsibility to quickly be able to fix them. The general accessibility of such audio material is also key (via audio transcripts for instance). Further, meeting applications need to be made better to facilitate hybrid conversations together with a usable remote platform to ease participation with in-presence attendees.

Human processes – A successful hybrid event requires session chairs to be effective. This requires management of interactions and on-boarding. Expectation management is also key to this end, whereby fairness needs to be defined as to how events will prioritise the experience of in-person attendees relative to remote participants. Being transparent about privacy and security decisions is also necessary.

Planning – The key question here is the ability to manage the uncertainty of meeting logistics – how many participants attend in-person versus remote since this ratio has direct consequence on the registration fees and is an issue for the organising team.

Integration – What systems need to be put in place to make a smoother integration of remote and in-person attendees? How to organise community introductions and on-boarding? How to ensure long-term mentor-ships (beyond the conference) needed for inclusion of community members are made possible. Would a parallel track or a programme to integrate new people into the community help?

Unexpected Consequences – The last aspect is how to deal with financial models (for organisers and for professional societies) that rely on in person conference registration as a revenue stream and still remains an unresolved challenge.

5.3 Requirements for Hybrid Conferences (Group 2)

Jon Crowcroft (University of Cambridge, GB), Quentin De Coninck (University of Louvain, BE), Jari Arkko (Ericsson – Jorvas, FI), Georg Carle (TU München, DE), Alexander Raake (TU Ilmenau, DE)

In this group, we took a more classification approach to the topic of hybrid conferences. Firstly, we outlined the actors and the technology requirements. Then, we looked at the organisational and interaction effects, moving on to the impact of scale. Finally, we discussed non-technical aspects such as legal and privacy considerations and mapping these considerations on to the five different types of hybrid events, identified earlier in the meeting.

Actors and objects of meetings – Participants can appear as real persons, or be represented by transducers (e.g., local robots), or proxies (e.g., other local participants). They may be rendered via room-attached screen(s)/loudspeaker(s), or even as holographic representations. A participant's actions may appear as real signals from participants, or be mediated through symbolic or other representations that are perhaps technology-mediated. Meanwhile, room(s) can be meeting room(s) or connectives, such as corridors or hallway(s).

Technology components – Effort in setting up hybrid meetings varies widely depending on the level of attempt to achieve fidelity or to provide some valid meeting experience. High-effort approaches may include feature-rich systems, with personal proxies of remote participants, or local video-robots as technical proxies of remote participants. Low-effort might involve laptops to allow local participants in hubs to see and hear others in other hubs.

Technology components include considerations of the variation in hardware and software requirements for local and remote participants, and whether these provide interoperability, e.g., via web-based approaches (e.g., WebRTC). Similarly, low level baseline technical support might be required at all ends, such as YouTube integration

On the media side, most experts agree the primary consideration is audio, and factors include intelligibility, quality, localization, spatialization; echo-cancellation, amongst others. Video quality and localization matter, but A/V sync (time, space) less so. Latency considerations show up when a meeting needs to have more or less interactive and symmetric versus asymmetric delay depends on the meeting organisation (free form or chaired.)

Connectivity also strongly depends on the physical room characteristics, acoustics and lighting setup, with poor room acoustics often contributing to a bad experience, however much effort is put into better microphones and software. Complementary tools such as chat, white board, document camera (physical whiteboard) are useful (also used as meta tools to navigate multiple sessions). Similarly, collaboration tools not integrated into the conferencing applications (such as Google Docs), support the collaboration in the event of poor audio.

Meeting organization – The organization of the meeting is also very important when choosing tools and technology. So gatekeeper roles and mechanisms such as meeting access management, registration fees and other financial considerations, matter. During a meeting, specific individuals acting as moderator(s), or directors (perhaps somewhat like TV/movie directors) of meeting can really help too, for example, choosing the currently

relevant video and audio to be remotely presented; this can also be partly done by a tool (e.g., speaker tracking). The general structure of the event, once running, matters as well. Considerations of human processes and meeting behavior, expectation management, including indications about privacy and security-related matters of meetings and timings are very important. Meetings across multiple time zones, and the impact on individual or multiple, distributed group locations matter a great deal in terms of fatigue, meals and sleep.

Interaction-related effects – There are a number of navigation like activities that are needed if we want the whole event experience of a hybrid meeting to be anything like real life. Some of these matter a great deal more than was realized before we started depending on online meetings. For example establishing ad-hoc communication channels, finding and navigation, groups, individuals; setting up hallway discussions, perhaps through virtual break out assignments – are increasingly valued.

During ongoing encounters, using established communication channels to manage activity and interactivity (e.g., of conversations, participants) all needs – continuity over long periods, including keeping in touch between participants, the need for session control (by a human or by technology) for groups of individuals that have latencies beyond when human conversational group communication paradigms work and of course, support for decision-making tools (voting or IETF hum tools). Collaboration between participants (shared document, shared screen or shared code) also is important that requires integration between tools, and at the very least through (possibly managed) screen and URL sharing.

Larger-scale impact and effects – As events scale up beyond small meetings or workshops, the challenges increase for managing meeting effectiveness and efficiency. Meeting fatigue, jet lag, multitasking, all start to take a toll on participants, and therefore on the overall group. Fatigue is possibly contributed to by the reduction in non-verbal communication. New encounters and meetings in hallways are starting to be supported by virtual reality (VR) environments, and meeting formats are evolving to take advantage of this emerging technological support.

Legal, security and privacy aspects – Meetings need to continue to be recorded, despite technological advances. Handling of sensitive material, pre-meeting, during meeting, post-meeting (e.g., deleting files afterwards) needs to be thought through. Indication of accessibility are a legal requirement in many countries. Implications depend on whether political or private topics may be discussed and should be made clear as part of the pre-meeting management, as should concerns about possible metadata collection by third-party vendors.

Hybrid meeting considerations – We considered this thought experiment: Imagine these extreme points of virtual versus physical co-location of participants: multiple 2-pairs local (total N), all combined virtually versus two large rooms with $N/2$ participants, with virtual connection between two rooms. Now the question is how to measure the attention of participants, (e.g., eye tracking of the participants) and determine the relative value of virtual versus physical human communication protocols. Perhaps a small research program could be based on this. From the QoE perspective, influencing factors include human and technology as previously discussed: Audio (intelligibility, quality, localization); video (quality, localization), A/V sync (time, spatial), delay (symmetric vs. asymmetric), connectivity, physical room characteristics (real vs. virtual), acoustics and lighting setups.

5.3.1 Mapping the requirements to the Hybrid terminology

We now map the requirements to the four kinds of hybrid meetings identified previously based on the discussions in the breakout.

Passive (inactive) hybrid – The goal here is to make video of talks, demos, panels available to all "attendees". This requires no actions from the participants perspective (actors) and objects of meeting are basically depictions of online material. In terms of effort this is low-key and involves low technological involvement too. Meeting organization is easy, with minimal moderation, and not constrained by time-zones. Although interaction-related effects are minimal with no encounters at all leading to no collaboration. As such in terms of impact and effects, it leads to one-way dissemination and minimal feedback. Further, the legal, security and privacy aspects only need to be checked in advance.

Semi-passive (semi-active) hybrid – The goal here is to enable participation (including questions and presentations) by audience, but not full functionality. This involves getting used to the tools (e.g., Discord, Slack) that enable such inclusivety. In terms of technology, the requirements involve all of passive hybrid (see above) plus a possibility to invoke textual chat functionality when needed.

True (fully active) hybrid – The goal here is for presenters and audience to be both local and remote with full ability to participate in all activities such as hallway discussions and others. This mode requires high-quality interaction for all situations and the ability to perceive audience reactions. As such, a well-working system is needed for participating in irregular hallway discussions, and allowing attending individual conversations in a larger gathering (aka "cocktail party effect").

Distributed hybrid – The goal here is to recreate regional in-person clusters, with a shared program and viewing parties. In terms of technology, this most importantly requires keeping interactivity between participants.

5.4 Financial, Diversity, and Timezone Implications for Hybrid Events

Henning Schulzrinne (Columbia University – New York, US), Srinivasan Keshav (University of Cambridge, GB), Cristel Pelsser (University of Strasbourg, FR), Sujata Banerjee (VMware - Palo Alto, USA)

We discussed some implications of hybrid conferences in this session.

5.4.1 Why hybrid conferences could be more attractive

Compared to purely online conferences, semi-passive and fully active hybrid conferences allow for physical presence. Physical presence at conferences is valuable from the perspective of multiple sectors. Participants from industry can meet potential employees and learn of advances in the field. Participants from academia find physical presence critical for high-bandwidth learning and networking and recruiting students (or faculty). Participants from government sector (especially funders such as the NSF) also find physical presence important to learn about the field and where additional economical incentives are needed. Physical presence also leads to multiple positive outcomes, for instance:

- *Face-to-face interaction:* Smaller gatherings allow participants to get a sense of which topics the research community is collectively moving towards.
- *Recruiting:* It is common for employers to send employees to recruit graduating doctoral students at conferences. This is typical for industry that especially run dedicated job fairs at conferences to this end.
- *Forcing attendees to block off time*, with the benefit of getting energized by change of location and refreshed at a conference by change of the environment.
- *"Reward" vacation:* Travel to an attractive venue is a reward (especially for the student authors) for a paper being accepted and all the hard work it entails!

For these reasons, hybrid conferences, which allow physical presence, are preferred to purely online conferences. However, the group noted in passing that the group was not convinced that collaborations materialize from interactions at large conferences, since most collaborations are between students and faculty on the same campus or regional meetings and visits.

Hybrid conferences are important from a financial perspective, as well. Professional societies (such as ACM and IEEE) are being hit with three simultaneous financial shocks: a loss of funds due to open access publishing, decline in membership, and declining conference revenues due to the move to online conference. Thus, they have an incentive to boost revenues using physical conferences, which brings in more revenue than online conferences. This will make them more supportive of hybrid conferences over purely online conferences. Although, the financial shocks are hitting not just the professional societies, but other organisations too (e.g., IETF) since the costs scale in complex ways. Meanwhile, hybrid conferences are also more attractive for sponsors, compared to purely online conferences. Finally, researchers, both faculty and students also can typically access travel funds to travel to hybrid conferences. So, for these financial reasons, it is expected that hybrid conferences would become more common in the future.

5.4.2 Diversity

Diversity has different dimensions, such as differences in geographical regions, under-represented minorities (such as women in computer science), disadvantaged people such as those with disabilities, or being financially constrained. At a high level, hybrid conferences have the potential to increase diversity, and in fact, the measures chosen by hybrid conferences should percolate to physical conferences as well.

We now discuss why we believe this to be the case, as well as specific best practices. To begin with, we advocate moving the location of the conference around the world in consecutive editions, to be more inclusive to different geographies. Most major conferences do this already. However, we need to caution that not all tools work in all geographies – e.g., the ecosystem of Google tools in China. Second, hybrid conferences can be more inclusive using new technology. For instance, hybrid conferences (and online as well) can be more inclusive in terms of different language groups. It is now possible to provide simultaneous translation for non-native English speakers. Other ways where hybrid conferences are more inclusive than physical conferences include video recordings, especially with *automatic captions*, have helped non-native speakers. *Text-to-speech* to do the presentations automatically, where the non-native English presenters simply write the script. This is not necessarily a purely positive outcome! Of course, it has always been possible to hire someone to speak (or record) on your behalf, do the slides (or video) productions, with appropriate disclosures. Finally, accessibility options for various impairments – screen-readers, other accessibility options, speech-to-text translations. It may be also be possible to hire remote video interpreters for

sign language (i.e., not at the main venue but at each local site for a multi-site hybrid.) This is not cheap but then there is no need to pay for travel and multiple interpreters can be used to load balance this effect.

Hybrid events have the potential to increase inclusion but new issues may arise, leading to new dangers. For instance, hybrid conferences will create **first- and second-class attendees**. For example, some faculty may restrict junior students to the remote option. **Funding for the physical portion** of the conference may be more difficult to obtain. Corporate sponsors used to fund student travel. The question is whether they will continue to do so in the hybrid world. Industry may fund students to attend in person, primarily from a recruiting point of view, but only if students are also physically present. As such, there will be need to find new ways to use sponsorship money in the hybrid world. Hybrid events may generate **social pressures to not attend** conferences – e.g., women with young children being pressured by family to not go. For example, more women left the workforce than men during the COVID-19 crisis. Providing childcare at the venue will mitigate this effect. However, it can be difficult to find on-site childcare even for hybrid conferences, especially for services that may need to be provided outside the normal work day. Finally, some folks may have access to better video production resources. As such, there may be a need to transfer travel money to video production costs. Of course, many universities already have video recording studios for remote teaching. These could be made available to graduate students for conference presentations, for instance.

5.4.3 Timezones

It is impossible to avoid the inherent problems that arise from attendees participating from multiple timezones. Attendees of hybrid conferences will need to realize that their experience will never be as good an experience at a fully physical conference. Nevertheless, there is a need to use a combination of strategies to make the experience as good as possible. We now discuss some potential strategies. To begin with, both local and remote attendees will need to show some flexibility to allow the program to spill outside the "normal" workday, potentially answering questions on their work in the middle of the night. In any case, with time-zones, it is critical that there be both synchronous and asynchronous modes of communication and interaction. For attendees who cannot attend some part of the conference, they will need a way to catch up to the event and its content.

Perhaps the only way to deal with time-zones properly is to opt for a multi-location hybrid with no single conference venue. Physical participants at one location would interact virtually and asynchronously with participants at other locations. We could have a 24 hour program or replication of events – perhaps 14-16 hour striped event, as with SIGCOMM 2020. An extreme version would be to have multiple physical conferences that are somewhat independent and translate from each other, if they are run in different local languages. Each version could have different live and recorded content! In this approach, national entities organize events (e.g., COMSNETS, SIGCOMM, APNET) and the top x% translated to international venues, presented on behalf of the authors, creating a federated super conference. SIGGRAPH Asia/Europe/US are examples. For non-local conference editions, papers could be presented by proxies and questions answered live, for instance.

However, this strategy also cuts the community into segments. But this was the case in the past as well, when inter-region collaboration was hard. Local communities were the past and may be the future, as well, if COVID-19 evolves variants. Moreover, political barriers to collaboration also exist and are growing, and may preclude multi-national collaboration (besides the problem with funding international students, which is a problem already.) We

noted that many researchers from some countries (e.g., Japan and maybe Russia, and China in the future) mostly present in their own local forums and do not present at international venues. Perhaps the future is indeed local! If so, there will certainly be a loss of cross-cultural interactions.

To summarize, there is a feeling that multiple time zones in hybrid environments will continue to perpetuate split communities. The group felt that perhaps we are on the cusp of some change, with two simultaneous developments: a decrease in ease of travel and an increasing notion that world is splitting due to political processes. Perhaps we are re-entering a future that looks like that; with the past 50 years being a glorious anomaly!

5.5 Hybrid Everything: Colloquiums, Hackathons & Research Visits

Amr Rizk (Universität Duisburg-Essen, DE), Oliver Hohlfeld (BTU Cottbus, DE), Michael Menth (Universität Tübingen, DE), Jürgen Schönwälder (Jacobs University Bremen, DE), Simone Ferlin (Ericsson – Stockholm, SE)

5.5.1 On deciding for an hybrid event

The question is not how to hybrid everything but for what meeting formats do we need to have an on-site component? There is a need to dissect the different activities at a meeting and design appropriate formats for them. A hybrid meeting is a meeting that you attend online and would have no access otherwise. You get to a hybrid meeting either from adding an on-site component to a fully virtual meeting or allowing remote participants in a usually on-site meeting. Since it is all about the meeting objectives, we discuss them next:

Meeting objectives – Different meetings have different objectives and thus require different hybrid levels. For instance, with IETF/RIPE meetings, the design goal is not to provide equal opportunities to local and remote participants. Such meetings can utilise certain access control mechanisms whereby certain decisions (elections) are specifically made in person at the meetings. On the other hand, teaching has different design goals, whereby local and remote participants must be treated the same. As such, one aspect to consider is what are the goals of the local and remote participants and whether they have same or different goals such as passive participation (listening to talks) versus active participation (meeting people).

Requirements for hybrid meetings – The participants need to be aware that it is a hybrid meeting, so as to adjust their expectation and behaviour. Firstly, the participants must be open and there should be willingness to interact with remote participants. Secondly, there must be a discipline as to make sure that everything that happens locally is remotely accessible, too. There might be instances, where private chats are not made accessible, such as conversations that happen during the in-person only parts of a hybrid event for instance. As such, expectation management is necessary to ensure everybody knows what happens when and who needs to be involved in what.

When is hybrid good (or bad) depends on participant motivation – For instance, with passive and semi-active participation, the goal is simply to listen in to talks and to interact with few people. As such, it is acceptable to not give equal privileges to everyone

and perhaps hybrid is a good alternative. On the other hand, if we want to include everyone (with same privileges) hybrid is difficult since senior academics attend a conference not for the talks but for interacting with others and/or strengthening social ties. Overall, the purpose of the event dictates the level of hybrid nature of the event.

5.5.2 Things to consider when organising hybrid events

There are different financial and technical implications of hybrid events. For instance, a passive hybrid event requires a dedicated video team. This incurs costs as to the video equipment and staff to handle the video and chat functionality during the event, but is largely affordable. On the other hand, "true hybrid" events are way more expensive and technically complex too (e.g., the SIGCHI remote robot experiment) and also do not scale up well to a large number of participants. Some more implications are outline below:

Financial risk for organizers – The question is why should a remote participant, one who is simply interested to sneak into the conference largely to get to know a new community have to pay equally. This poses a financial risk for the organisers since they do not know how many participants will register locally and remotely. As such, the entire business model (sponsoring, and attendance fees) depends on the format of the event whereby remote participants do not get to contribute in case of passive or semi-passive hybrid.

Timezones – The issue of time requires willingness and depends also on how often one needs to participate at odd times. For instance, participants from East Asia, USA and Europe are very challenging to add simultaneously to the remote event.

Lastly, when transitioning from on-site to hybrid, can consequently lead to the majority going online only. As such, it is important to lay out the target audience for whom the event is intended. For instance, as previously mentioned, in teaching, the trend is clear in the sense that online offerings results into very few or none on-site attendees.

5.5.3 Real-world Examples

Teaching – With online teaching, shared material (e.g., asynchronous video material) becomes very relevant. However, designing a hybrid course is hard. Physical teaching is much easier with slides and follow up questions. The key requirement with hybrid mode of teaching is to ensure that remote participants get to have the same experience as local ones. As such, good audio equipment is needed in the lecture halls. A traditional blackboard cannot be used any more – but a digital variant is needed that remains connected to the laptop and streams the content online while also projecting it locally. At the same time, context-switching to help both audiences is tricky where extra help might be necessary in the classroom. A flipped classroom is more fun for everyone (teacher and learners). However the problem is that the format is presently not generally accepted. On the other hand, people need to come on campus to interact and meet people. As such a mixture of both is needed whereby teaching should be online, while all other interactions should be on campus.

Remote IETF Experience – In general, networking (e.g., getting to know new people) was extremely hard in remote-only operation. As such, a hybrid setting may not be the right medium when the goal is to leverage the IETF meeting as an ongoing source to connect with industry. The remote registration is also rather expensive (although fee-waivers are possible without justification) and not proportional to the value that an (academic) gets out of a remote IETF meeting.

Conferences – The experience has been similar to that of remote teaching. The experience has been very positive when it comes to passively participating in other communities with low investment (e.g., meetings that are organised online now and would otherwise be only for a set of participants e.g., operators.). It also allows more equal opportunities to participate, albeit a bit difficult to implement in hybrid mode.

Hybrid meetings – Faculty meetings that were hybrid were horrible. It is not just about audio/video issues but more about the social cues. Perhaps brown bag lunch meetings could be the future since the meeting is online, the participation also goes up.

Project meetings – Online meetings are more structured. The main value comes from the notion that participants prepare ahead and most of the brainstorming goes into the preparation phase. Consequently, project meetings are better prepared, are inherently shorter and produce more output. In-person meetings on the other hand only help to create (or strengthen) social ties more strongly.

Geographically distributed companies – In this scenario, social ties are less relevant, but more important is to get the work done. Meanwhile, many companies already consist of geographically distributed teams. As such, the expectation to work together and get the work done is already in place.

Social ties – Social bonds are usually created out of joint experiences. They can either happen online or offline. For instance, to make online workshops more successful, they should be better at creating such joint experiences. Workshops are usually not as interactive as often people wish they would be. As such, having a good social event where the participants jointly do something is crucial.

5.5.4 Predictions for the future

It will be a gradual process to go online, whereby young people will be driving this change. All the small conferences will go entirely online or will just disappear. Meanwhile, all the big conferences will go hybrid – they have large enough communities that attend locally to survive. We might also see a world of regional events again that largely disappeared. On the academic side there will be a competition in the transition phase – some parts of the world will go in-person earlier, while others join in later. Distributed conferences might become a new way of organizing events. The downside here is the complexity of the organisation due to handling of finances. If we look at how distributed approaches in networking succeed or die out, it will be a question whether distributed conferences will succeed or not. An increasing number of people will chose not to travel to certain places of the world for political, environmental or economic reasons. Once the environmental issues become worse (and the climate models are correct), traveling will become expensive (due to increasing taxes on jet fuel), so increasingly fewer number of people will be able to travel. Consequently, funding agencies might stop affording it. As such, it is quite likely, the IETF will not have three big in-person meetings per year in ten years from now.

6 Conclusions and Next Steps

The goal of this seminar was to first review the current status quo of virtual conferences: what works, what doesn't, what needs improvement (theme of day 1). From this discussion, we discussed the requirements, implications, and guidelines for designing hybrid conferences (theme of day 2). It was generally believed that small venues will move entirely online and

others will be held as hybrid events in the future. Thus, design guidelines for hybrid events are needed. With this seminar we contributed guidelines for deciding when virtual or hybrid conferences are suitable and how to design them. The clear next step is to evaluate these guidelines in practice to provide data points for which designs work and which do not.

The discussions emerged from a group that was biased a bit by more senior colleagues. It is possible, digital natives might see this perspective very differently, since at the end of the day, the younger generation will be driving this effort.

Remote Participants

- Jari Arkko
Ericsson – Jorvas, FI
- Vaibhav Bajpai
TU München, DE
- Sujata Banerjee
VMware – Palo Alto, US
- Georg Carle
TU München, DE
- Jon Crowcroft
University of Cambridge, GB
- Quentin De Coninck
University of Louvain, BE
- Simone Ferlin
Ericsson – Stockholm, SE
- Andrew Hines
University College Dublin, IE
- Oliver Hohlfeld
BTU Cottbus, DE
- Daniel Karrenberg
RIPE – Amsterdam, NL
- Wolfgang Kellerer
TU München, DE
- Srinivasan Keshav
University of Cambridge, GB
- Mirja Kühlewind
ERICSSON Eurolab – Herzogenrath, DE
- Mirjam Kühne
RIPE – Amsterdam, NL
- Franziska Lichtblau
MPI für Informatik – Saarbrücken, DE
- Michael Menth
Universität Tübingen, DE
- Jörg Ott
TU München, DE
- Cristel Pelsser
University of Strasbourg, FR
- Colin Perkins
University of Glasgow, GB
- Alexander Raake
TU Ilmenau, DE
- Amr Rizk
Universität Duisburg-Essen, DE
- Jürgen Schönwälder
Jacobs University Bremen, DE
- Henning Schulzrinne
Columbia University – New York, US
- Georgios Smaragdakis
TU Delft, NL
- Ralf Steinmetz
TU Darmstadt, DE
- Cristina Videira Lopes
University of California – Irvine, US
- Martina Zitterbart
KIT – Karlsruher Institut für Technologie, DE

Report from Dagstuhl Seminar 21283

Data Structures for Modern Memory and Storage Hierarchies

Edited by

Stratos Idreos[1], Viktor Leis[2], Kai-Uwe Sattler[3], and Margo Seltzer[4]

1 **Harvard University – Cambridge, US,** stratos@seas.harvard.edu
2 **Universität Erlangen-Nürnberg, DE,** viktor.leis@fau.de
3 **TU Ilmenau, DE,** kus@tu-ilmenau.de
4 **University of British Columbia – Vancouver, CA,** mseltzer@cs.ubc.ca

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 21283 "Data Structures for Modern Memory and Storage Hierarchies". For decades, computers consisted of a CPU, volatile main memory, and persistent disk. Today, modern storage technologies such as flash and persistent memory as well as the seemingly inevitable migration into virtualized cloud instances, connected through high-speed networks, have radically changed the hardware landscape. These technologies have major implications on how to design data structures and high-performance systems software. The seminar discussed how to adapt data structures and software systems to this new hardware landscape.

Seminar July 11–16, 2021 – http://www.dagstuhl.de/21283
2012 ACM Subject Classification Information systems → Data management systems
Keywords and phrases Cloud, Data Structures, Database Systems, Flash, Near-Data Processing, Persistent Memory
Digital Object Identifier 10.4230/DagRep.11.6.38

1 Executive Summary

Viktor Leis (Universität Erlangen-Nürnberg, DE)

The seminar brought together researchers and practitioners from the data management and systems/storage communities to discuss the implications of the modern hardware landscape on high-performance systems. Due to the pandemic, the seminar was organized as a hybrid event: Virtual participation was limited to one session per day that featured invited talks. The in-person component consisted of free-flowing plenary discussions and several smaller, focused working groups. Some key takeaways from the discussion are:

- **OS/DBMS co-design**: Traditional POSIX-style OS abstractions do not work well for data-intensive systems, leading to complex workarounds and suboptimal performance. While some of these issues could in principle be fixed by optimizing OS implementations, others require new APIs. For example, it is very difficult to implement crash-consistent data structures on top of the mmap system call.
- **Cloud**: The cloud is taking over and cloud-native data processing systems often have a a very different architecture from traditional data management systems. For example, many systems strive to separate storage from compute. This trend is enabled by ever faster networks.

- **Near-data processing**: Separating storage from compute leads to costly data movement, which may be mitigated by pushing down (parts of) the computation close to the data. Major public cloud vendors already to optimize their internal services towards this goal. The challenges is how to program such distributed and specialized hardware components.
- **Persistent Memory**: One major question discussed at the seminar was the role of byte-addressable persistent memory in future systems and whether what the "kill app" for this technology is. While there are several promising applications (e.g., graph processing or systems that require fast recovery times), it is not clear whether wide adoption will occur. Currently, the technology is quite expensive (prices per byte are similar to DRAM) and very hard to program in a crash-consistent way (e.g., writes must be carefully ordered similar to lock-free-style programming).

2 Table of Contents

Executive Summary
Viktor Leis . . . 38

Overview of Talks

An update on the Enzian system
Gustavo Alonso . . . 41

Deep Memory and Storage Hierachies for Scalable and Efficicient DBMSs
Carsten Binnig . . . 41

Reasoning about cloud-native data-structures
Jana Giceva . . . 41

The data systems grammar
Stratos Idreos . . . 42

A Taxonomy of Database and SSDs Co-designs
Alberto Lerner . . . 42

NVM: Bubble Memory all over Again?
Margo Seltzer . . . 43

Working groups

Future databases
Carsten Binnig, Gustavo Alonso, and Alberto Lerner . . . 43

Interface Challenges between Databases and Operating Systems
Christian Dietrich, André Brinkmann, Viktor Leis, and Thomas Neumann . . . 44

Out-of-Memory Data Structures
Viktor Leis and Thomas Neumann . . . 45

A Preview of Upcoming Cache Coherency Technologies
Alberto Lerner, Gustavo Alonso, Kai-Uwe Sattler, and Jens Teubner . . . 46

Near-data processing – State-of-the-art and open problems
Marcus Paradies . . . 46

Non-volatile Memory in Database Systems
Kai-Uwe Sattler, Alexander Baumstark, and Muhammad Attahir Jibril . . . 50

Participants . . . 53

Remote Participants . . . 53

3 Overview of Talks

3.1 An update on the Enzian system

Gustavo Alonso (ETH Zürich, CH)

Joint work of Gustavo Alonso, Timothy Roscoe
Main reference Gustavo Alonso, Timothy Roscoe, David Cock, Mohsen Ewaida, Kaan Kara, Dario Korolija, David Sidler, Zeke Wang: "Tackling Hardware/Software co-design from a database perspective", in Proc. of the 10th Conference on Innovative Data Systems Research, CIDR 2020, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings, www.cidrdb.org, 2020.
URL http://cidrdb.org/cidr2020/papers/p30-alonso-cidr20.pdf

This talk presents the status of Enzian, a research computer being developed at ETHZ. Enzian has been designed to enable research in a wide range of topics related to how systems architecture, from both the hardware and the software perspective, need to evolve in view of developments such as accelerators and cloud computing architectures.

The talk links to several of the topics discussed during the seminar: disaggregated memory, memory hierarchies, distributed systems architecture, etc.

3.2 Deep Memory and Storage Hierachies for Scalable and Efficicient DBMSs

Carsten Binnig (TU Darmstadt, DE)

In recent years, memory and storage hierarchies have become increasingly deeper. On a single machine, additional memory and storage technologies such as PMem or NWMe SSDs have been introduced, allowing DBMSs to scale beyond the data sizes that pure in-memory systems can handle. Moreover, thanks to technologies such as remote direct memory access (RDMA) and NVMe over Fabrics, memory and storage can even scale beyond the capcities of a single machine without sacrificing too much of performance. In this talk, I have been discussing new opportunities (e.g., how to lay out data in an optimal manner across layers) and challenges (e.g., how to keep data copies consistent across layers) that arise for building scalable and efficient DBMSs when exploiting all these different layers of memory and storage on local and remote machines.

3.3 Reasoning about cloud-native data-structures

Jana Giceva (TU München, DE)

The design of data structures should no longer be driven solely by the data layout, the algorithm's access patterns and the properties of the underlying hardware. The premise is that any future data structure must also consider the impact of the relevant cloud metrics, a list that is longer than just performance and cost. This talk is a teaser into what designing

a cloud data structure means in the context of scale, resource disaggregation, and novel cloud-native data system architectures. This entails reasoning in terms of cloud service components, understanding the tradeoffs where must we ensure no-data loss as opposed to good quality of service, as well as considering the impact of the whole system stack when using the underlying network-attached resources.

3.4 The data systems grammar

Stratos Idreos (Harvard University – Cambridge, US)

Data structures are everywhere. They define the behavior of modern data systems and data-driven algorithms. For example, with data systems that utilize the correct data structure design for the problem at hand, we can reduce the monthly bill of large-scale data systems applications on the cloud by hundreds of thousands of dollars. We can accelerate data science tasks by being able to dramatically speed up the computation of statistics over large amounts of data. We can train drastically more neural networks within a given time budget, improving accuracy.

However, knowing the right data structure and data system design for any given scenario is a notoriously hard problem; there is a massive space of possible designs while there is no single design that is perfect across all data, queries, and hardware scenarios. We will discuss our quest for the first principles of data structures and data system design. We will show signs that it is possible to reason about this massive design space, and we will show early results from a prototype self-designing data system which can take drastically different shapes to optimize for the workload, hardware, and available cloud budget using machine learning and what we call machine knowing. These shapes include data structure and system designs which are discovered automatically and do not exist in the literature or industry.

3.5 A Taxonomy of Database and SSDs Co-designs

Alberto Lerner (University of Fribourg, CH)

Joint work of Alberto Lerner, Philippe Bonnet
Main reference Alberto Lerner, Philippe Bonnet: "Not your Grandpa's SSD: The Era of Co-Designed Storage Devices", in Proc. of the SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021, pp. 2852–2858, ACM, 2021.
URL https://doi.org/10.1145/3448016.3457540

This talk discussed the numerous advantages of Co-designing Databases and SSDs. Most notably, co-designing allows a system to offload some database tasks onto storage hardware and thus obtain better performance or resource utilization. These tasks can range from simple behavioral changes in the device, such as scheduling IO operations from a latency-sensitive transaction log with high priority, to moving entire computations into the device, such as executing a portion of a query plan or transforming a log segment into a partial checkpoint. A taxonomy of offload-capable devices was presented, which organizes the devices according to the type of interface they offer. Two of these classes can benefit from further research: devices

with computational features and database-storage co-designed devices. This talk summarizes a join work tutorial with Philippe Bonnet presented at SIGMOD'21 about Databases and SSDs co-design [1].

References

1 Alberto Lerner and Philippe Bonnet. *Not Your Grandpa's SSD: The Era of Co-designed Storage Devices. SIGMOD'21*, June, 2021.

3.6 NVM: Bubble Memory all over Again?

Margo Seltzer (University of British Columbia – Vancouver, CA)

Non-volatile, byte-addressable memory (NVM) has been touted as the next big revolution in persistent storage. With the the load/store access model and performance of DRAM with the persistence of flash, what could be better as a foundation for high-performance data management? In fact, the research community has been prolific in publications touting the amazing systems we'll see; yet, commercial impact has been minimal. Why?

Both the technology and hype harken back to the 1970s and the introduction of a different non-volatile technology: Bubble Memory. Like NVM, pundits predicted that bubble memory would be a game changer in our system stack. It wasn't. This talk explores the lessons we should take away from the bubble memory mania. Our after talk discussion will focus on identifying the Killer Apps that will make NVM a true game changer in the 21st century.

4 Working groups

4.1 Future databases

Carsten Binnig (TU Darmstadt, DE), Gustavo Alonso (ETH Zürich, CH), and Alberto Lerner (University of Fribourg, CH)

This work group discussed the future architecture of database systems from the perspective of modern hardware and cloud computing. The group outlined a novel system running on serverless that takes advantage of all the services the cloud provides without giving up the advantages of an actual database engine. To certain extent, what we designed is a disaggregated data processing engine.

4.2 Interface Challenges between Databases and Operating Systems

Christian Dietrich (TU Hamburg-Harburg, DE), André Brinkmann (Universität Mainz, DE), Viktor Leis (Universität Erlangen-Nürnberg, DE), and Thomas Neumann (TU München, DE)

Databases and operating systems (OS) have in common that they have to serve applications that only reveal their wishes within concrete service request. While database management systems consider users with SQL queries as "applications", they themselves are considered as "applications" by the operating system. Rooted in this dual role, the interaction between the OS and databases suffers from an expectation mismatch: Although database developers want to use general-purpose OS interfaces, they are often disappointed by the supplied performance and the given guarantees (e.g., atomicity). On the other hand, OS developers argue that databases should just use the (right) OS interface (correctly) and give more hints about the intended use of the requested resources. But, since databases are in a similar position as the OS and cannot predict the next application request, this criticism often bears no fruit.

Instead, database developers bypass central OS infrastructure (e.g., by performing direct block-device accesses) and re-implement parts of the OS functionality in user space. While these private re-implementations have the benefit of being more controllable, they make the database a problematic citizen from the OS perspective. One example of this implementation pattern is the *buffer manager* in the DBMS, which has its OS equivalent in the *page cache*; both are caches for the secondary storage, get filled on demand, and evict pages with or without write-back. In contrast to the page cache, the buffer manager allows for fine-grained control about eviction and eviction order, which is necessary for atomicity guarantees on data updates. However, such a process-local buffer manager occupies resident memory, which the OS, in contrast to page-cache pages, cannot easily reclaim when the memory pressure rises.

In our working group, we discussed the `mmap()` OS interface, which would allow the database to rely on the page cache as its buffer manager. However, even if leaving eviction control aside and focusing only on read-only workloads, the current Linux implementation is problematic: When reading random pages from a high-speed NVMe SSD, the OS fills up its page cache and makes the data available in the processes' virtual address space, whereby it nearly reaches the bandwidth limits of the underlying device (around 3 GiB/s). At some point, the page cache reaches its limit, memory becomes scares, and the OS starts to evict data pages. For this, the OS removes the evicted pages from the virtual address space, which requires the OS to ensure mapping consistency on all thread-executing cores. With a TLB shootdown, which is sent as an inter-processor interrupt (IPI), the OS requests TLB flushes on the other CPU cores. In our benchmark, these shootdowns dominate the benchmark performance after second 25 and, after second 40, each 4K read provokes on average 4 IPIs (see Fig. 1).

To tackle this issue, the Linux system call `madvise()` already provides two flags (`MADV_DONTNEED` and `MADV_COLD`), whereby the application can hint that certain pages are not needed in the near future, which would allow for a lazy unmapping of pages without TLB shootdown. However, with the current implementation, it seems that shootdowns are still performed eagerly. Also the vectorized `madvise()` variant `process_madvise()`, which would also eliminate system-call overheads, currently performs one TLB shootdown per unmapped page instead of batching them after the system call.

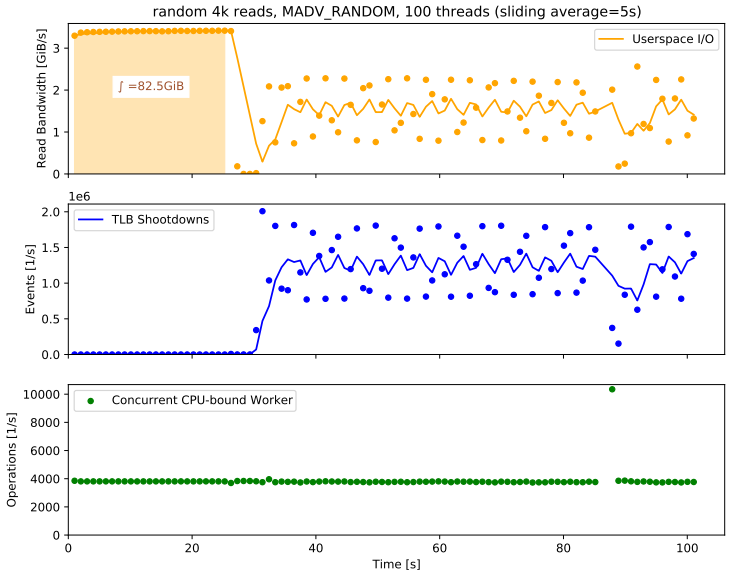


Figure 1 Random 4K Reads from Memory-Mapped Device. Benchmark was executed on a AMD EPYC 7713 64-Core Processor with 512 GiB RAM

A more general idea to improve `mmap()` for database systems could be to introduce process-local (or even mapping-local) page-cache partitions. While these would be under control of the kernel, the user-space application should be able to finely control page eviction and consistency/atomicity requirements. Preferably, this control could be exercised through asynchronous low-overhead kernel interfaces (e.g., `io_uring`) in order to keep up with the bandwidth of NVMe SSDs RAIDs.

4.3 Out-of-Memory Data Structures

Viktor Leis (Universität Erlangen-Nürnberg, DE) and Thomas Neumann (TU München, DE)

Joint work of Alfons Kemper, Viktor Leis, Alberto Lerner, Ulrich Meyer, Thomas Neumann, Alexander van Renen, Kai-Uwe Sattler, Jens Teubner

B-trees are still the most common out-of-memory data structure and perform well when the data is larger than main memory. Specialized in-memory data structures (e.g., radix trees), on the other hand, are often faster for pure in-memory workloads. In this working group, we discussed how to close this gap by designing a data structure that combines ideas from both data structures. The goal is to be as fast as pure in-memory data structures on workloads where the working set fits into main memory, while transparently supporting larger than main memory data sets as well. The key idea behind our data structure, which we code-named Dagstuhl-Tree, is to cache individual keys in a fast in-memory data structure (e.g., a radix tree). The on-disk representation is still similar to a traditional B-tree, but caching and eviction occur at a key granularity. Thus, the proposed data structure can not only speed up in-memory workloads (due to the fast in-memory data structure) but also out-of-memory workloads (due to more fine-grained cache utilization).

4.4 A Preview of Upcoming Cache Coherency Technologies

Alberto Lerner (University of Fribourg, CH), Gustavo Alonso (ETH Zürich, CH), Kai-Uwe Sattler (TU Ilmenau, DE), and Jens Teubner (TU Dortmund, DE)

This working group discussed the Compute Express Link (CXL) protocol [2], an emerging memory coherence standard that allows peripheral devices to manipulate their host's memory seamlessly. The protocol, backed by a large consortium of companies, is expected to soon appear in a new generation of commercial CPUs, GPUs, NICs, accelerators, and potentially SSDs. The group found that CXL, along with other coherence protocols such as CCIX [1] and ETH's Enzian machine's [3], have the potential to open the design space for database systems as follows. Peripheral devices can now read and update data without explicitly moving it first, thanks to the coherent hardware support. New database systems can deploy distributed, *exo-CPU* computations while still benefiting from a shared memory abstraction. The group presented its preliminary research questions and discussed the availability of academic and commercial platforms to support such efforts.

References

1 CCIX Consortium. *Cache Coherent Interconnect for Accelerators – Base Specification Revision 1.a Version 1.0.* `https://www.ccixconsortium.com`, July, 2019.
2 CXL Consortium. *Compute Express Link Specification 2.0.* `https://www.computeexpresslink.org/download-the-specification`, October, 2020.
3 Abishek Ramdas, David Cock, Timothy Roscoe, and Gustavo Alonso. *The Enzian Coherent Interconnect (ECI): opening a coherence protocol to research and applications. LATTE '21*, April, 2021.

4.5 Near-data processing – State-of-the-art and open problems

Marcus Paradies (German Aerospace Center – Jena, DE)

Introduction

The seminal idea of offloading computations close to the data to reduce unnecessary data movement dates back more than three decades. Although the general concept of near-data processing has been around for quite a while already, it only recently got enough momentum to foster an increasing research demand from academia and a more widespread development of near-data processing solutions from the industry. The growing interest in near-data processing is mainly driven by two factors: (1) ever-growing data volumes & an increasing demand for advanced analytics, and (2) increasing heterogeneity & specialization of hardware components (processing, memory & storage, and network) to data-intensive workloads. Growing data volumes pose tremendous challenges to data systems and demand more complex and scalable system architectures, including complex network topologies and deep I/O hierarchies, potentially spanning local storage, remote storage, and cloud storage resources [3]. Therefore, near-data processing can be found today across all hardware components of distributed data systems, in particular network and memory & storage infrastructures. In summary, near-data processing provides the following advantages:

- **Reduction of data movement.** Deep I/O hierarchies in distributed execution environments increase data movement since data has to be moved potentially across multiple storage tiers and the network before it is processed. This can lead to potential bandwidth bottlenecks on the network and storage stacks. Besides contributing to a waste of bandwidth resources, data movement also consumes a considerable amount of energy. Near-data processing reduces the amount of data that needs to be transferred and thus saves bandwidth resources and potentially reduces the overall energy consumption of the system infrastructure.
- **Reduction of access latency.** Offloading computations can also reduce data access latency by avoiding unnecessary data transfers across the network and storage stack.
- **Reduction of load on the host CPU.** Near-data processing enables freeing up scarce host CPU cycles by offloading parts of the computation into the network or the storage stacks. Hardware specialization in the network or in storage might even lead to faster computations compared to running the same operation on general-purpose CPUs.
- **Increase of data privacy & security.** The reduction of data movement increases data privacy and security since only data that is relevant for the processing is moved out of the storage system and across the network.

Types of Near-Data Processing

Near-data processing comes in different flavors and is typically considered in the context of offloading computations into *memory* (e.g., *Processing-in-Memory (PIM)* or *Near-Memory Processing*), *storage* (e.g., *Computational Storage*), or the *network* (e.g., *In-Network Processing (INP)*). Along the entire data path, all components, such as disks, network cards, switches, and memory modules become *active*, i.e., can perform certain operations on the data they handle.

Processing-in-Memory (PIM) and Near-Memory Processing

PIM addresses the *memory wall problem*, i.e., the growing discrepancy between microprocessor performance and DRAM memory speed [8]. By placing a lightweight processing unit in/near memory, PIM helps to alleviate the memory bandwidth limitations of traditional von-Neumann architectures. Recent developments, such as Samsung's HBM-PIM and AxDIMM and UPMEM's PIM solution based on a DRAM processing unit (DPU), demonstrate the increasing interest and momentum from industry to commercialize and embrace PIM-based hardware components for memory-bound applications [9]. While PIM is a promising technology which is gaining more traction lately, challenges for a widespread adoption arise from a limited set of supported operations and the lack of tools and programmability features.

Computational Storage

Computational storage devices (CSD) allow offloading computations into or near the storage device. While in-storage processing has been advertised since the early 90's, only recently commercial CSD products (e.g., Samsung SmartSSD, NGDSystems Newport, and Scaleflux) based on SSDs became available [5, 4]. More broadly, computational storage refers to a family of different technologies, which provide computational resources close to the storage devices. Computational resources might reside within the storage device itself (e.g., some

embedded ARM cores) or are connected via a peripheral interconnect (e.g., a CSD based on reconfigurable hardware (FPGA)). Despite the availability of CSD hardware, there is currently no standard interface mechanism available, which uniformly describes the interaction between the host software and the CSD device. Standardization efforts in the NVMe working group *Computational Storage* discuss extensions of the NVMe protocol for offloading computations (e.g., orchestrated through eBPF).

In-Network Processing

Modern programmable networks create the opportunity for in-network processing, i.e., offloading computations from end hosts into network devices such as programmable switches and smart NICs [1, 7]. Programmable switches, such as Barefoot Networks' Tofino, have a flexible parser and a customizable match-action engine. To process packets at high speed, this architecture has a multi-stage pipeline where packets flow at line rate. Each stage has a fixed amount of time to process every packet, allowing for lookups in memory, manipulating packet metadata and stateful registers, and performing boolean and arithmetic operations [1]. While programmable switches offer impressive performance, they only provide a limited memory size, a limited set of supported actions (e.g., simple arithmetic, data manipulation, and hashing operations), and few operations per packet to guarantee execution at line rate.

Abstractions and Primitives

Near-data processing can be employed for specific usage scenarios, i.e., to offload a well-defined, fixed operation (e.g., SQL filtering & aggregation, regex searches, compression & encryption, etc.) or user-defined operations (e.g., UDF-like operations or kernels). Depending on the offered programming model (e.g., match-action, data-flow, etc.) and offloading mechanism (e.g.,OS/container/VM, bitstream, or eBPF), the expressiveness and composability of operations can vary dramatically. For example, initial PIM solutions only offered simple arithmetic operations to be offloaded, while recent computational storage products allow the execution of arbitrary user code in a containerized manner directly inside the SSD. It remains an open problem, how future programming models for near-data processing will look like. Even promising offloading mechanisms, such as using eBPF in the context of computational storage, struggle to allow general (and potentially complex) offloads of operations into storage devices. In cloud deployments, near-data processing opportunities are usually not directly exposed, but shall be used through well-defined service interfaces (e.g., AWS S3 SELECT), which poses the question of how much control future data systems (DBMSs and data-intensive systems like Spark, Flink, etc.) will have over such abstracted service interfaces. As of today, the most common usage of near-data processing is to offload pre-selected tasks (i.e., operator-level) into memory, storage, or the network. Few examples allow offloading arbitrarily complex operations (i.e., query-level / pipeline-level) or even run the entire DBMS inside the storage device [2].

Open Research Problems

Given the recent excitement about programmable hardware components (memory, storage, and network), there is a large number of open research (and technical) questions that will have to be addressed. The following provides an (incomplete) list of open problems:

- **Programming models and offloading mechanisms:** Programming models are currently mostly kernel-based in some supported programming language (e.g., p4 or eBPF). It is unclear how multiple near-data processing compute units (e.g., programmable switches and programmable SSDs) can be programmed under a unified programming model. The offloading mechanisms are device-specific and usually coupled to a specific protocol (e.g., NVMe in the context of computational storage).
- **Capabilities of near-data processing compute units:** Seminal works on near-data processing already prove the suitability of computation offloading for bandwidth-bound operations, where offloading an operation would lead to a significant reduction in data volume to be transferred. Nevertheless, new hardware devices (e.g., FPGAs, low-energy CPUs, ASICs) for near-data processing with drastically different performance characteristics will have to be evaluated for relevant data-intensive use cases. Besides purely non-functional requirements, also limitations that stem from the programming model have to be taken into account (e.g., p4 and eBPF pose certain restrictions on the types of operations that can be offloaded).
- **Offloading granularity:** It is an open question, at which granularity offloading tasks should be pushed into near-data processing compute units (e.g., sub-operator [6], operator, pipeline, query, or entire DBMS or data system).
- **Scheduling and Offload placement:** Given complex and deep I/O hierarchies with potentially multiple offloading opportunities, offload scheduling and placement become challenging research problems. Imagine a complex three-tier storage hierarchy with programmable SSDs & HDDs and as cold data archive a cloud-based storage service with UDF-like operator offloading. An interesting aspect is the (potentially negative) impact of near-data processing on the usefulness of caches and buffer managers.
- **Security & Performance isolation:** Near-data processing compute units are usually less powerful than full-fledged server CPUs (in particular for low-energy processors in storage devices). Since such resources will be shared by potentially many applications, performance isolation is of utmost importance. Further, unauthorized data access outside of the own local execution context must be prevented (imagine the potential danger of ransomware attacks that could be enabled through computation offloading into storage devices).
- **Cost models:** Cost models can provide a means to steer the scheduling and offloading placement depending on a generic cost metric, which allows pushing the operation to be offloaded to the optimal near-data processing compute unit. Developing such cost models is an open research problem.
- **Dealing with heterogeneous hardware and execution environments:** Data systems (e.g., DBMSs) run in different execution environments (e.g. on-premise or in the cloud), which determines also the opportunities for detecting offloading capabilities in a potentially complex system landscape. In a cloud setting, the entire storage stack (and therefore explicit control over offloading decisions) might be hidden behind an abstract service API. Generic offloading mechanisms and programming models have to be developed in order to allow data systems to leverage potentially diverse near-data processing opportunities in different execution environments.

References

1 Sapio, Amedeo and Abdelaziz, Ibrahim and Aldilaijan, Abdulla and Canini, Marco and Kalnis, Panos. *In-Network Computation is a Dumb Idea Whose Time Has Come*. Proceedings of the 16th ACM Workshop on Hot Topics in Networks, 2017

2 Jong Hyeok Park and Soyee Choi and Gihwan Oh and Sang Won Lee. *SaS: SSD as SQL Database System*, Proc. VLDB Endow., 2021

3 Marcus Paradies. *CryoDrill: Near-Data Processing in Deep and Cold Storage Hierarchies*, 2019, 9th Biennial Conference on Innovative Data Systems Research, 2019.

4 Antonio Barbalace and Jaeyoung Do. *Computational Storage: Where Are We Today?*. 11th Conference on Innovative Data Systems Research, 2021.

5 Gu, Boncheol and Yoon, Andre S. and Bae, Duck-Ho and Jo, Insoon and Lee, Jinyoung and Yoon, Jonghyun and Kang, Jeong-Uk and Kwon, Moonsang and Yoon, Chanho and Cho, Sangyeun and Jeong, Jaeheon and Chang, Duckhyun. *Biscuit: A Framework for Near-Data Processing of Big Data Workloads*, ISCA, 2016.

6 Maximilian Bandle and Jana Giceva. *Database Technology for the Masses: Sub-Operators as First-Class Entities*, Proc. VLDB Endow., 2021

7 Blöcher, Marcel and Ziegler, Tobias and Binnig, Carsten and Eugster, Patrick, *Boosting Scalable Data Analytics with Modern Programmable Networks*, Proceedings of the 14th International Workshop on Data Management on New Hardware. 2018

8 Onur Mutlu and Saugata Ghose and Juan Gomez-Luna and Rachata Ausavarungnirun, *A Modern Primer on Processing in Memory*, CoRR, 2020

9 Juan Gomez-Luna and Izzat El Hajj and Ivan Fernandez and Christina Giannoula and Geraldo F. Oliveira and Onur Mutlu. *Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture*, CoRR, 2021

4.6 Non-volatile Memory in Database Systems

Kai-Uwe Sattler (TU Ilmenau, DE), Alexander Baumstark (TU Ilmenau, DE), and Muhammad Attahir Jibril (TU Ilmenau, DE)

Non-volatile memory (NVM) started as a concept aiming at combining the properties of DRAM (low latency, byte-addressability) with those of storage (persistence, capacity, price). The idea of NVM goes back to the sixties when bubble memory was discussed. Since then, several memory technologies have been proposed, among them Re-RAM, STT-RAM, and PCM – see [1, 2] for surveys. However, only Intel together with Micron have shipped NVM products sitting in existing DRAM slots: Intel Optane DCPMM based on the 3D XPoint technology. Among the the major advantages of its Byte-addressability over SSDs are that it

- allows to use identical data structures both for persistent and transient data and, therefore,
- eliminates the need to transfer data between persistent storage and memory.

For this purpose, NVM permits direct access (in terms of cache lines) to data via standard CPU instructions such as `STORE` and `LOAD`.

Intel Optane DCPMM supports two operating modes: the *memory mode* as a large, but volatile memory pool where DRAM can act as a cache layer on top of NVM, and the *app direct mode* where NVM is used as persistent memory. Operating systems like Linux and Windows on the most recent Intel platforms integrate Optane DCPMM via Direct Access (DAX) interface, i.e., persistent memory is memory-mapped into the address space and, thus, allows to load/store from/to memory directly.

Special cache line flushing instructions are used to guarantee that store operations are persistent because the path to the *persistence domain*, where a store to persistent memory becomes durable, consists of volatile layers like the CPU caches. `CLFLUSHOPT` is an optimized

form of `CLFLUSH`. `CLWB` (cache line write back) is similar to `CLFLUSHOPT` but does not evict the cache line after flushing. These are followed by memory barrier instructions like `SFENCE` to enforce ordering and ensure the cache lines reach the *persistence domain*.

Recently, Intel has announced and shipped the second generation 200 series with increased bandwidth and Enhanced Asynchronous DRAM Refresh (eADR) support. eADR extends ADR to include CPU caches in the *persistence domain*, alongside the persistent memory and the memory controller's write pending queues. This further makes NVM programming easier and eliminates cache line flushes, thereby enhancing performance.

On top of this, additional APIs and development kits such as Intel's PMDK[1] simplify the software development [3].

Over the last few years, researchers have analyzed and benchmarked Intel's NVM Optane technology. The main findings are [4, 5]:

1. Asymmetry between load and store latency.
2. Asymmetry between load and store bandwidth.
3. Sequential IO faster than random IO.
4. Access granularity based on an internal 256-byte buffer.
5. Load bandwidth scale with thread count while store bandwidth does not.
6. Higher latency and lower bandwidth compared to DRAM.

In the database context, main research fields and use cases are:

- instant recovery and logging, e.g. write-behind logging [6], log-free recovery [7] and query recovery [8, 9].
- NVM-optimized data structures including hash tables [10], radix trees [11, 12] and B$^+$-tree variants [13].
- I/O primitives [14], parallel programming models [15], efficient algorithms [16] etc.

However, NVM is not (yet) the promised game changer for several reasons:

- Access latency is still higher than that of DRAM.
- Despite the availability of PMDK, NVM programming is still challenging.
- Finally, the still high costs per GB have hindered the wide adoption.

Overall, NVM has promising prospects as yet another tier of modern memory and storage hierarchies, as it opens up unprecedented opportunities for database systems on future hardware.

References

1 Margo I. Seltzer, Virendra J. Marathe, and Steve Byan. An NVM carol: Visions of NVM past, present, and future. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, pages 15–23. IEEE Computer Society, 2018.

2 Haikun Liu, Di Chen, Hai Jin, Xiaofei Liao, Binsheng He, Kan Hu, and Yu Zhang. A survey of non-volatile main memory technologies: State-of-the-arts, practices, and future directions. *J. Comput. Sci. Technol.*, 36(1):4–32, 2021.

3 Steve Scargall. *Programming Persistent Memory*. Apress, Berkeley, CA, 2020.

4 Shashank Gugnani, Arjun Kashyap, and Xiaoyi Lu. Understanding the idiosyncrasies of real persistent memory. *Proc. VLDB Endow.*, 14(4):626–639, 2020.

5 Joseph Izraelevitz, Jian Yang, Lu Zhang, Juno Kim, Xiao Liu, Amirsaman Memaripour, Yun Joon Soh, Zixuan Wang, Yi Xu, Subramanya R. Dulloor, Jishen Zhao, and Steven Swanson. Basic performance measurements of the intel optane dc persistent memory module, 2019.

[1] `https://pmem.io`

6 Joy Arulraj, Matthew Perron, and Andrew Pavlo. Write-behind logging. *Proc. VLDB Endow.*, 10(4):337–348, 2016.

7 Gang Liu, Leying Chen, and Shimin Chen. Zen: a high-throughput log-free OLTP engine for non-volatile main memory. *Proc. VLDB Endow.*, 14(5):835–848, 2021.

8 Soklong Lim, Tyler Coy, Zaixin Lu, Bin Ren, and Xuechen Zhang. Nvgraph: Enforcing crash consistency of evolving network analytics in NVMM systems. *IEEE Trans. Parallel Distributed Syst.*, 31(6):1255–1269, 2020.

9 Alexander Baumstark, Philipp Götze, Muhammad Attahir Jibril, and Kai-Uwe Sattler. Instant graph query recovery on persistent memory. In Danica Porobic and Spyros Blanas, editors, *Proceedings of the 17th International Workshop on Data Management on New Hardware, DaMoN 2021, 21 June 2021, Virtual Event, China*, pages 10:1–10:4. ACM, 2021.

10 Daokun Hu, Zhiwen Chen, Jianbing Wu, Jianhua Sun, and Hao Chen. Persistent memory hash indexes: An experimental evaluation. *Proc. VLDB Endow.*, 14(5):785–798, 2021.

11 Se Kwon Lee, K. Hyun Lim, Hyunsub Song, Beomseok Nam, and Sam H. Noh. WORT: write optimal radix tree for persistent memory storage systems. In Geoff Kuenning and Carl A. Waldspurger, editors, *15th USENIX Conference on File and Storage Technologies, FAST 2017, Santa Clara, CA, USA, February 27 – March 2, 2017*, pages 257–270. USENIX Association, 2017.

12 Shaonan Ma, Kang Chen, Shimin Chen, Mengxing Liu, Jianglang Zhu, Hongbo Kang, and Yongwei Wu. ROART: range-query optimized persistent ART. In Marcos K. Aguilera and Gala Yadgar, editors, *19th USENIX Conference on File and Storage Technologies, FAST 2021, February 23-25, 2021*, pages 1–16. USENIX Association, 2021.

13 Lucas Lersch, Xiangpeng Hao, Ismail Oukid, Tianzheng Wang, and Thomas Willhalm. Evaluating persistent memory range indexes. *Proc. VLDB Endow.*, 13(4):574–587, 2019.

14 Alexander van Renen, Lukas Vogel, Viktor Leis, Thomas Neumann, and Alfons Kemper. Persistent memory I/O primitives. In Thomas Neumann and Ken Salem, editors, *Proceedings of the 15th International Workshop on Data Management on New Hardware, DaMoN 2019, Amsterdam, The Netherlands, 1 July 2019*, pages 12:1–12:7. ACM, 2019.

15 Guy E. Blelloch, Phillip B. Gibbons, Yan Gu, Charles McGuffey, and Julian Shun. The parallel persistent memory model. In Christian Scheideler and Jeremy T. Fineman, editors, *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures, SPAA 2018, Vienna, Austria, July 16-18, 2018*, pages 247–258. ACM, 2018.

16 Pedro Ramalhete, Andreia Correia, and Pascal Felber. Efficient algorithms for persistent transactional memory. In Jaejin Lee and Erez Petrank, editors, *PPoPP '21: 26th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Virtual Event, Republic of Korea, February 27- March 3, 2021*, pages 1–15. ACM, 2021.

Participants

- Gustavo Alonso
ETH Zürich, CH
- Alexander Baumstark
TU Ilmenau, DE
- Carsten Binnig
TU Darmstadt, DE
- André Brinkmann
Universität Mainz, DE
participant Christian Dietrich
TU Hamburg-Harburg, DE
- Muhammad Attahir Jibril
TU Ilmenau, DE
- Alfons Kemper
TU München, DE
- Viktor Leis
Universität Erlangen-Nürnberg, DE
- Alberto Lerner
University of Fribourg, CH
- Ulrich Carsten Meyer
Goethe-Universität – Frankfurt am Main, DE
- Thomas Neumann
TU München, DE
- Ismail Oukid
Snowflake – Berlin, DE
- Marcus Paradies
German Aerospace Center – Jena, DE
- Kai-Uwe Sattler
TU Ilmenau, DE
- Jens Teubner
TU Dortmund, DE
- Alexander van Renen
Universität Erlangen-Nürnberg, DE

Remote Participants

- Marcos K. Aguilera
VMware – Palo Alto, US
- Raja Appuswamy
EURECOM – Biot, FR
- Manos Athanassoulis
Boston University, US
- Alexander Böhm
SAP SE – Walldorf, DE
- Peter A. Boncz
CWI – Amsterdam, NL
- Mark Callaghan
Rockset – Bend, US
- Khuzaima Daudjee
University of Waterloo, CA
- Jana Giceva
TU München, DE
- Goetz Graefe
Google – Madison, US
- Gabriel Haas
Universität Erlangen-Nürnberg, DE
- Stratos Idreos
Harvard University – Cambridge, US
- Wolfgang Lehner
TU Dresden, DE
- Danica Porobic
Oracle Labs – Redwood Shores, US
- Ken Salem
University of Waterloo, CA
- Wolfgang Schröder-Preikschat
Universität Erlangen-Nürnberg, DE
- Margo Seltzer
University of British Columbia – Vancouver, CA
- Tianzheng Wang
Simon Fraser University – Burnaby, CA
- William Wang
ARM Ltd. – Cambridge, GB

Report from Dagstuhl Seminar 21292

Scalable Handling of Effects

Edited by

Danel Ahman[1], Amal Ahmed[2], Sam Lindley[3], and Andreas Rossberg[4]

1 **University of Ljubljana, SI,** danel.ahman@fmf.uni-lj.si
2 **Northeastern University – Boston, US,** amal@ccs.neu.edu
3 **University of Edinburgh, GB,** sam.lindley@ed.ac.uk
4 **Dfinity – Zürich, CH,** rossberg@mpi-sws.org

Abstract

Built on solid mathematical foundations, effect handlers offer a uniform and elegant approach to programming with user-defined computational effects. They subsume many widely used programming concepts and abstractions, such as actors, async/await, backtracking, coroutines, generators/iterators, and probabilistic programming. As such, they allow language implementers to target a single implementation of effect handlers, freeing language implementers from having to maintain separate ad hoc implementations of each of the features listed above.

Due to their wide applicability, effect handlers are enjoying growing interest in academia and industry. For instance, several effect handler oriented research languages are under active development (such as Eff, Frank, and Koka), as are effect handler libraries for mainstream languages (such as C and Java), effect handlers are seeing increasing use in probabilistic programming tools (such as Uber's Pyro), and proposals are in the pipeline to include them natively in low-level languages (such as WebAssembly). Effect handlers are also a key part of Multicore OCaml, which incorporates an efficient implementation of them for uniformly expressing user-definable concurrency models in the language.

However, enabling effect handlers to scale requires tackling some hard problems, both in theory and in practice. Inspired by experience of developing, programming with, and reasoning about effect handlers in practice, we identify five key problem areas to be addressed at this Dagstuhl Seminar in order to enable effect handlers to scale: Safety, Modularity, Interoperability, Legibility, and Efficiency. In particular, we seek answers to the following questions:

- How can we enforce safe interaction between effect handler programs and external resources?
- How can we enable modular use of effect handlers for programming in the large?
- How can we support interoperable effect handler programs written in different languages?
- How can we write legible effect handler programs in a style accessible to mainstream programmers?
- How can we generate efficient code from effect handler programs?

Seminar July 18–23, 2021 – http://www.dagstuhl.de/21292
2012 ACM Subject Classification Theory of computation → Control primitives; Theory of computation → Program semantics
Keywords and phrases continuations, Effect handlers, Wasm
Digital Object Identifier 10.4230/DagRep.11.7.54
Edited in cooperation with Hillerström, Daniel

Scalable Handling of Effects, *Dagstuhl Reports*, Vol. 11, Issue 06, pp. 54–81
Editors: Danel Ahman, Amal Ahmed, Sam Lindley, and Andreas Rossberg

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

1 Executive Summary

Danel Ahman (University of Ljubljana, SI)
Amal Ahmed (Northeastern University – Boston, US)
Sam Lindley (University of Edinburgh, GB)
Andreas Rossberg (Dfinity – Zürich, CH)

Algebraic effects and effect handlers are currently enjoying significant interest in both academia and industry as a modular programming abstraction for expressing and incorporating user-defined computational effects in programming languages. For example, there are a number of effect handler oriented languages in development (such as Eff, Frank, and Koka); there exist effect handler libraries for mainstream languages (such as C and Java); effect handlers are a key part of languages such as Multicore OCaml (and indeed they are due to appear in the production release of OCaml next year); effect handlers are being increasingly used in statistical probabilistic programming (such as Uber's Pyro tool); and proposals are in the works to include effect handlers in new low-level languages (such as WebAssembly). While effect handlers have solid mathematical foundations and have been extensively experimented in prototype languages and on smaller examples, enabling effect handlers to scale still requires tackling some hard problems. To this end, this Dagstuhl Seminar 21292 "Scalable Handling of Effects" focused on addressing the following key problem areas for scalability: Safety, Modularity, Interoperability, Legibility, and Efficiency.

This seminar followed the earlier successful Dagstuhl Seminars 16112 "From Theory to Practice of Algebraic Effects and Handlers" and 18172 "Algebraic Effect Handlers go Mainstream", which were respectively dedicated to the foundations of algebraic effects and to the introduction of them into mainstream languages. In contrast to these previous two seminars which took place in person at Schloss Dagstuhl, the current seminar was organised fully online due to the SARS-CoV-2 pandemic. As the seminar was attended by participants from a wide range of time zones (ranging from the West coast of the US all the way to Japan), coming up with a schedule that was suitable for everybody was a challenge. In the end, we decided to have three scheduled two-hour sessions each day, with impromptu informal discussions also happening between-times. These sessions were: (i) 15:00-17:00 CEST, which were deemed the Core Hours, where all participants were most likely to be able to present; (ii) 10:00-12:00 CEST, which was most suitable for participants from Asia and Europe; and (iii) 17:30-19:30 CEST, which was most suitable for participants from America and Europe. The Core Hours included talks, breakouts, and discussions of interest to the widest audience, with more specialised talks and breakouts taking place in the other two daily scheduled blocks. Talks were recorded so that participants could catch up due to being in an incompatible time zone, then deleted at the end of the week.

In order to run a successful virtual Dagstuhl seminar we exploited several different technologies. For talks we used Zoom. For breakouts we used a combination of Zoom and Gather.town, and for asynchronous communication and further discussions we used Zulip. For scheduling purposes, we used the wiki page provided by Dagstuhl.

We collected initial lists of proposed talks and breakout topics before the seminar began using an online form. We extended these throughout the week. We scheduled talks and breakout groups daily depending on audience interest and the participant availability. While the first part of the week was dominated by talks, the second part of the week saw more emphasis on breakouts and discussions. During Friday's Core Hours, the leaders of each

breakout group presented a short overview of the discussions and results (11 reports in total). Initially, we were a little unsure about how well breakout sessions would work in a virtual seminar, but as the week went on they became more and more popular and they seemed to go remarkably well. Initially, we mostly used Gather.town and its virtual whiteboards for the breakout sessions. Subsequently, we transitioned to mostly using Zoom breakout rooms (partly because some people had difficulty using Gather.town on their systems).

The seminar was a great success, particularly given the constraints of the virtual format.

There were vibrant discussions around multishot continuations. These are vital for exciting new applications such as probabilistic programming and automatic differentiation, but more research is needed on how to implement them safely and efficiently in different contexts. Flipping perspective, it was mooted that for certain applications, particularly those involving direct interaction with the external world, it might be worthwhile restricting attention to runners, which are even more constrained than effect handlers with singleshot continuations.

There were several discussions relating to usability of effect handlers. These resulted in proposals to design a lecture course on effect handlers and to write a book on how to design effectful programs.

A major area of interest instigated at a prior Dagstuhl Seminar (18172 "Algebraic Effect Handlers go Mainstream") is the addition of effect handlers to WebAssembly. A design is being actively worked on as part of the official WebAssembly development process. At the current seminar we worked out extensions to the existing proposal to accommodate named effect handlers and symmetric stack-switching, both of which promise more efficient execution.

An issue with many existing benchmarks for effect handlers is that they often require installing a range of experimental software and configuring it with just the right settings. In order to make it easier to compare systems and share experimental setups we created the effect handlers benchmarks suite – a repository of benchmarks and systems covering effects and handlers in various programming languages, based on Docker scripts that make it easy for anyone to run the benchmarks and adapt them for their own research. The repository is hosted on GitHub. Since the seminar, 5 systems have been added to the repository and it has been actively updated and maintained by different members of the community.

At the end of the week, there was strong interest among the participants to continue this successful seminar series and submit a proposal for another incarnation, hopefully possible to take place on site in about two years.

2 Table of Contents

Executive Summary
Danel Ahman, Amal Ahmed, Sam Lindley, and Andreas Rossberg 55

Overview of Talks

(Higher-Order) Asynchronous Effects
Danel Ahman 59

The real world cannot be handled
Andrej Bauer 60

Taming Higher-Order Control and State with Precise Effect Dependencies
Oliver Bracevac 60

Higher-order Programming with Effects and Handlers – with First-Class Functions
Jonathan Immanuel Brachthäuser 61

A Separation Logic for Effect Handlers
Paulo Emílio de Vilhena 63

Problems with resources and effects
Stephen Dolan 64

Probabilistic Programming
Maria Gorinova 64

Composing UNIX with Effect Handlers
Daniel Hillerström 64

ParaFuzz: Fuzzing Multicore OCaml Programs
Sivaramakrishnan Krishnamoorthy Chandrasekaran 72

Retrofitting Effect Handlers onto OCaml
Sivaramakrishnan Krishnamoorthy Chandrasekaran 73

Koka update: Compilation to C via generalized evidence passing and Perceus reference counting.
Daan Leijen 73

Handler Calculus
Sam Lindley 73

Efficient Compilation of Algebraic Effect Handlers
Matija Pretnar 74

Programming and Proving with Indexed effects in F*
Aseem Rastogi and Nikhil Swamy 74

Low-level effect handlers for Wasm
Andreas Rossberg 75

Back to Direct Style 3
Philipp Schuster 75

CPS Transformation with Affine Types for Call-By-Value Implicit Polymorphism
Taro Sekiyama 75

Effects with Shifted Names in OCaml
Antal Spector-Zabusky 76

Effects, Interface Types and async APIs
Luke Wagner 76

Working groups

Control Operators Breakout Session
Jonathan Immanuel Brachthäuser, Youyou Cong, Sam Lindley, and Taro Sekiyama 76

UX of Effect Systems Breakout Session
Jonathan Immanuel Brachthäuser, Youyou Cong, Paulo Emílio de Vilhena, and Filip Koprivec 77

Effect Handlers Benchmark Suite
Daniel Hillerström 77

Wasm breakout session
Andreas Rossberg, Sam Lindley, and Luke Wagner 78

Dependent types breakout session
Wouter Swierstra and Robert Atkey 80

Open problems

Efficient stack layout for multishot handlers
Filip Koprivec 80

Participants 81

3 Overview of Talks

3.1 (Higher-Order) Asynchronous Effects

Danel Ahman (University of Ljubljana, SI)

Joint work of Danel Ahman, Matija Pretnar, Janez Radešček

While covering a large body of examples, the operational treatment of algebraic effects has remained synchronous in nature, meaning that when executing code in a language with algebraic effects, an algebraic operation op's continuation is blocked until (i) op is propagated to some implementation of it (such as an effect handler, a runner [1], or some top-level default implementation), (ii) that implementation finishes executing, and (iii) the original program is interrupted with the implementation's result. In this talk, I gave an overview of our work on accommodating asynchrony within algebraic effects based on observing that the different phases (i)–(iii) of an algebraic operation's execution can be split into separate programming abstractions.

The first half of the talk was based on our recent paper [2]. In this part of the talk, I showed how the different phases (i)–(iii) can be captured in a core λ-calculus for asynchrony using the following programming abstractions:

- *signals*, which programmers can issue to indicate that some operation's implementation needs to be executed, and that behave operationally like algebraic operations (in that they propagate outwards);
- *interrupts*, which are propagated to a program as a result of some other program issuing a corresponding signal, and that behave operationally like effect handling (in that they propagate inwards);
- *interrupt handlers*, which programmers can use to react to interrupts, and that (despite their name) behave like (scoped [4]) algebraic operations (in that they propagate outwards, just like signals); and
- *awaiting*, with which programmers can selectively block a program's execution by explicitly awaiting for one of the promises made by interrupt handlers to be fulfilled.

The resulting system achieves asynchrony by ensuring that signals, interrupts, and interrupt handlers never block the execution of their continuations (apart from when asked to do so by explicitly awaiting). In order to model a program's environment, such as the implementation of some algebraic operation, our core calculus also included a simple form of parallel processes. In the talk I also demonstrated the wide applicability of the proposed system: not only can we implement tail-resumptive algebraic operation calls, but we can also implement much more involved examples, such as (cancellable) remote function calls, multi-party web applications, non-blocking post-processing of promises, and preemptive multi-tasking.

In the second part of the talk, I presented our ongoing work on resolving the shortcomings we have since identified in our original system. These included: needing general recursion in the core calculus due to its heavy usage in examples; not being able to pass higher-order values in the payloads of signals and interrupts so as to ensure type safety; and not being able to dynamically spawn new parallel processes. First, in order to remove general recursion from the core calculus, we extended interrupt handlers to a notion of *reinstallable interrupt handlers*, in which the interrupt handler code is allowed to selectively reinstall the given interrupt handler, covering the uses of general recursion in our example programs. Next, in order to support higher-order signal and interrupt payloads in a type-safe manner, we

extended our core calculus and the allowed payload types with a *Fitch-style modal* $\Box$*-type* [3], with which one can box up values of arbitrary types as payloads, while the type system guarantees that these values do not refer to any promise-typed binders in interrupt handlers (whose scope such payloads need to be able to escape). Finally, we also extended the core calculus with a programming abstraction for *spawning new parallel processes*, again using the technology involved in Fitch-style modal types to ensure type safety.

References

1 Ahman D., Bauer A. (2020) Runners in Action. In: Müller P. (eds) Programming Languages and Systems. ESOP 2020. Lecture Notes in Computer Science, vol 12075. Springer, Cham.
2 Danel Ahman and Matija Pretnar. 2021. Asynchronous effects. Proc. ACM Program. Lang. 5, POPL, Article 24 (January 2021), 28 pages.
3 Clouston R. (2018) Fitch-Style Modal Lambda Calculi. In: Baier C., Dal Lago U. (eds) Foundations of Software Science and Computation Structures. FoSSaCS 2018. Lecture Notes in Computer Science, vol 10803. Springer, Cham.
4 Maciej Piróg, Tom Schrijvers, Nicolas Wu, and Mauro Jaskelioff. 2018. Syntax and Semantics for Operations with Scopes. In Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '18). Association for Computing Machinery, New York, NY, USA, 809–818.

3.2 The real world cannot be handled

Andrej Bauer (University of Ljubljana, SI)

Joint work of Danel Ahman, Andrej Bauer
Main reference Danel Ahman, Andrej Bauer: "Runners in Action", in Proc. of the Programming Languages and Systems – 29th European Symposium on Programming, ESOP 2020, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2020, Dublin, Ireland, April 25-30, 2020, Proceedings, Lecture Notes in Computer Science, Vol. 12075, pp. 29–55, Springer, 2020.
URL https://doi.org/10.1007/978-3-030-44914-8_2

The language top level, or its runtime environment, is the interface to external resources, which are not subject to the usual rules of handlers or a monad in the language, and therefore should not be modeled as such. We should take the question "how to properly model and implement the runtime environment" seriously and apply available technology to develop modular and conceptually clean notion of "runtime environment". One such proposal was made by Danel Ahman and myself in our "Runners in action" paper. I would like to take the opportunity to discuss alternatives and to advertise the question as an interesting and important one.

3.3 Taming Higher-Order Control and State with Precise Effect Dependencies

Oliver Bracevac (Purdue University – West Lafayette, US)

This talk presents a novel ownership-style type system that tracks sets of term variables and assigns per-variable usage effects (e.g., reads, writes, kills) as determined by a user-defined effect quantale structure. For instance, through "kill effects," we support linear tracking

of destructive updates to convert data structures between mutable and immutable accesses without copying. Compared to previous works on ownership, our system has a particularly lightweight type- and term-level footprint due subtyping and Scala/DOT-style abstract "self-aliases" to model escaping closures. Combining ownership-style reasoning and effects opens up interesting new avenues for the compilation of impure higher-order languages. Our type system gives rise to a novel typed graph IR that infers precise local effect dependencies, finally leading to affordable and aggressive global optimizations for impure higher-order programs. The graph IR is part of the newest version of the Scala LMS compiler framework and its optimizations enable significant speedups in real-world effectful programs.

3.4 Higher-order Programming with Effects and Handlers – with First-Class Functions

Jonathan Immanuel Brachthäuser (EPFL – Lausanne, CH)

Joint work of Jonathan Immanuel Brachthäuser, Philipp Schuster, Edward Lee, Aleksander Boruch-Gruszecki
Main reference Jonathan Immanuel Brachthäuser, Philipp Schuster, Klaus Ostermann: "Effects as capabilities: effect handlers and lightweight effect polymorphism", Proc. ACM Program. Lang., Vol. 4(OOPSLA), pp. 126:1–126:30, 2020.
URL https://doi.org/10.1145/3428194

3.4.1 Introduction

Reasoning about the use of external resources is an important aspect of many practical applications. Examples range from memory management, controlling access to privileged resources like file handles or sockets, to analyzing the potential presence or absence of side effects.

3.4.1.1 Effect Systems encourage type-based reasoning

Effect systems extend the static guarantees of type systems to additionally track the use of effects [3]. They typically augment the type of functions with information about which effects the function might use. The fundamental idea of enhancing types with additional information is also one of the biggest problems of effect systems. Types quickly become verbose, difficult to understand, and difficult to reason about – especially in the presence of effect-polymorphic higher-order functions [7, 6, 1].

3.4.1.2 Capabilities encourage scope-based reasoning

Capabilities offer an alternative way to control the way resources are used. In this model, one can access resources and effects only through capabilities [4]. Thus, restricting access to capabilities restricts effects: a program can only perform effects of capabilities it can use. Some capabilities have a limited lifetime and should not leave a particular scope – for instance, if they are used to emulate checked exceptions. In these cases, treating capabilities as second-class values [5] provides such static guarantees. From a language designer's perspective, capabilities and second-class values offer an interesting alternative to effect systems: programmers can reason about effects the same way they reason about bindings. Additionally, second-class values admit a lightweight form of effect polymorphism without extending the language with effect variables or effect abstraction [1]. While lightweight, such systems severely restrict expressivity.

3.4.2 Comonadic type systems enable transitioning between type-based and scope-based reasoning

These systems allow programmers to reason about *purity* in an impure languages [2]. A special type constructor Safe witnesses the fact that its values are constructed without using any (impure) capabilities. Values of type Safe are introduced and eliminated with special language constructs. Importantly, explicit box introduction and elimination marks the transition between reasoning about effects by which capabilities are currently in scope, and reasoning about effects by types that witness the potential use of capabilities (that is, *impurity*). The type system presented by [2] only supports a binary distinction between *pure* values and *impure* values, which is not fine-grained enough for many practical applications – for instance, *effect masking*, or local handling of effects.

3.4.3 This Talk

In this talk, we draw inspiration from all three lines of research and present a calculus System C that aims at striking the balance between expressivity and simplicity. In particular, we combine and generalize the work by [5] and [2] to obtain a lightweight, capability-based alternative to effect systems. System C is based on the following design decisions:

3.4.3.1 Second-class values

Following [5], we distinguish between functions that can be treated as first-class values, and functions that are second-class. (To highlight this difference, we explicitly refer to second-class functions as *blocks*.) Thus, we avoid confronting programmers with the ceremony associated with tracking capabilities in types as much as possible. In particular, blocks can freely close over capabilities and effectful computations can simply use all capabilities in their lexical scope, with no visible type-level machinery to keep track of either fact.

3.4.3.2 Capability sets

Based on the work by [5] we annotate each binding in the typing context with additional information. However, we do not only track whether a bound variable is first- or second-class, but also track over which capabilities it closes. That is, we augment bindings (*e.g.*, $f :^{\mathcal{C}} \sigma$) in the typing context with *capability sets* (*e.g.*, $\mathcal{C}$). This information is annotated at the binder and is not part of the type. We will see that this is important for ergonomics as users are never confronted with this information. It is only necessary to check and guarantee effect safety.

3.4.3.3 Boxes

Blocks can freely close over other capabilities. However, they cannot be returned from a function or stored in a field. To recover these abilities we generalize the work by [2]: System C features explicit boxing and unboxing language constructs. Boxing converts a second-class value to a first-class value, reifying the contextual information annotated on the binder into the boxed value's type (*e.g.*, $f :^{\mathcal{C}} \sigma \vdash \mathbf{box}\, f : \sigma\, \mathbf{at}\, \mathcal{C}$). That is, instead of completely preventing first-class values from closing over capabilities, the capabilities they closed over are tracked in their types. To use a boxed block, we have to unbox it. We make sure to only perform this operation when the capabilities (*e.g.*, $\mathcal{C}$) are still in scope, which guarantees effect safety. The **box** and **unbox** constructs allow programmers to freely move between tracking capabilities implicitly, via scope, or explicitly, via type.

3.4.4 Discussion

In the talk, we will show how natural scope-based reasoning and precise type-based reasoning can co-exist in the same language and how programs can switch between them. We initially developed System C as a basis for adding first-class functions back to the Effekt language [1] – hence the title of this proposal. However, we believe that our system has broader applicability and we invite the participants to discuss the calculus, its limitations, and areas of application.

References

1 Jonathan Immanuel Brachthäuser, Philipp Schuster, and Klaus Ostermann. 2020. Effects as Capabilities: Effect Handlers and Lightweight Effect Polymorphism. *Proc. ACM Program. Lang.* 4, OOPSLA, Article 126 (Nov. 2020).

2 Vikraman Choudhury and Neel Krishnaswami. 2020. Recovering Purity with Comonads and Capabilities. *Proc. ACM Program. Lang.* 4, ICFP, Article 111 (Aug. 2020).

3 J. M. Lucassen and D. K. Gifford. 1988. Polymorphic Effect Systems. In *Proc. of the Symposium on Principles of Programming Languages* (POPL '88). ACM, New York, NY, USA, 47–57.

4 Mark Samuel Miller. 2006. Robust Composition: Towards a Unified Approach to Access Control and Concurrency Control. Ph.D. Dissertation. *Johns Hopkins University*, Baltimore, Maryland, USA. AAI3245526.

5 Leo Osvald, Grégory Essertel, Xilun Wu, Lilliam I González Alayón, and Tiark Rompf. 2016. Gentrification gone too far? affordable 2nd-class values for fun and (co-) effect. In *Proc. of the Conference on Object-Oriented Programming, Systems, Languages and Applications*. ACM, New York, NY, USA, 234–251.

6 Lukas Rytz, Martin Odersky, and Philipp Haller. 2012. Lightweight Polymorphic Effects. In *Proc. of the European Conference on Object-Oriented Programming*, James Noble (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 258–282.

7 Yizhou Zhang, Guido Salvaneschi, Quinn Beightol, Barbara Liskov, and Andrew C. Myers. 2016. Accepting Blame for Safe Tunneled Exceptions. In *Proc. of the Conference on Programming Language Design and Implementation*. ACM, New York, NY, USA, 281–295.

3.5 A Separation Logic for Effect Handlers

Paulo Emílio de Vilhena (INRIA – Paris, FR)

Joint work of Paulo Emílio de Vilhena, François Pottier
Main reference Paulo Emílio de Vilhena, François Pottier: "A separation logic for effect handlers", Proc. ACM Program. Lang., Vol. 5(POPL), pp. 1–28, 2021.
URL https://doi.org/10.1145/3434314

A program logic is a pair of a language, for writing the specification of a program, and a set of inference rules, for proving such specifications. In this talk, I present a program logic for a programming language with support for both effect handlers and higher-order state. I will begin with the motivation for this line of work – why is it interesting, or even useful, to conduct this research? I will then give an overview of the logic – how does it extend previous logics and what are its novel notions and main reasoning principles? Finally, I will present my vision for future research based upon this work: the verification of interesting applications of handlers, the design of extensions of the logic and its application to the study of effect systems.

3.6 Problems with resources and effects

Stephen Dolan (Jane Street – London, GB)

Two useful applications for effects are in controlling nondeterministic search (using continuations that are resumed multiple times), and organising programs that use asynchronous I/O (which manipulate stateful resources).

Problems arise when trying to do both in the same language: it is difficult to maintain guarantees of linearity and uniqueness in the presence of continuations that may resume more than once. I will present several tricky programs that mix these features, explain the problems they pose for the current crop of type systems, and leave their solution as a challenge for the audience.

3.7 Probabilistic Programming

Maria Gorinova (University of Edinburgh, GB)

Probabilistic programming aims to democratise Bayesian statistics and inference by providing a programming interface to the problem of probabilistic modelling. The user can specify their model, typically by describing the generative process of the data, and, in a perfect world, obtain inference results automatically. However, Bayesian inference is a challenging task, and it often needs to be tailored to the specific model in order to be efficient.

In this talk, I will discuss how effect handlers have been utilised to implement the backend of a few probabilistic programming languages, including Edward2 and Pyro. I will give several examples of common to probabilistic programming model transformation, which can be easily and compactly implemented using effect handlers. I will argue that such an effect-handling based backed provides the right set of abstractions for probabilistic programming users to be able to write and optimise model-specific and efficient inference strategies.

3.8 Composing UNIX with Effect Handlers

Daniel Hillerström (University of Edinburgh, GB)

URL https://www.youtube.com/watch?v=Ye90HCCG-UA

3.8.1 Introduction

In functional programming effect handlers are often explained in terms of *folds* or *case-splits* over computational trees (depending on whether the handlers in question are *deep* or *shallow*) [7, 11, 12]. In imperative programming effect handlers are often explained as a slight operational extension of exception handlers endowed with the ability to resume exception-raising computations [10]. A compelling programming paradigm-agnostic way to explain effect handlers is to explain them as tiny composable operating systems, where we

may view effectful operations as *system calls*, whose implementations are given by the *ambient environment*. In this analogy effect handlers play the role as the ambient environment. A richer ambient environment may be obtained by composing ever so more effect handlers. One can take this analogy quite literally, and use it to model the essence of an operating system such as UNIX by extending a feature-limited basis with more features by composing more handlers.

In the following sections we will demonstrate how to use effect handlers to model the essence of an UNIX-y operating system, which we shall call Tiny UNIX (the following sections are excerpts from my PhD dissertation [15]).

3.8.2 Basic i/o

The file system is a cornerstone of UNIX as the notion of *file* in UNIX provides a unified abstraction for storing text, interprocess communication, and access to devices such as terminals, printers, network, etc. We shall take a rather basic view of the file system. In fact, our system shall only contain a single file, and moreover, the system will only support writing operations. This system hardly qualifies as a UNIX file system. Nevertheless, it suffices to demonstrate the core idea, and it is not too difficult to grow it into a model of an actual file system [15]. The basic file system serves a crucial role for development of Tiny UNIX, because it provides the only means for us to be able to observe the effects of processes.

As in UNIX we shall model a file as a list of characters, i.e. $\mathsf{File} := \mathsf{List\ Char}$. We will use the same model for strings, $\mathsf{String} := \mathsf{List\ Char}$, such that we can use string literal notation to denote the `"contents of a file"`. The signature of the basic file system will consist of a single operation Write for writing a list of characters to the file.

$$\mathsf{BIO} := \{\mathsf{Write} : \langle \mathsf{FileDescr}; \mathsf{String} \rangle \twoheadrightarrow 1\}$$

The operation is parameterised by a FileDescr and a character sequence. In this note, we will leave the details of FileDescr abstract as they are really only necessary when one considers a file system with multiple distinct files. We shall assume the existence of a term $\mathsf{stdout} : \mathsf{FileDescr}$ such that we can perform invocations of Write. Let us define a suitable handler for this operation.

$$\begin{array}{l}
\mathsf{basicIO} : (1 \to \alpha ! \mathsf{BIO}) \to \langle \alpha; \mathsf{File} \rangle \\
\mathsf{basicIO}\ m := \ \mathbf{handle}\ m\ \langle\rangle\ \mathbf{with} \\
\qquad\qquad \mathbf{return}\ res \qquad\qquad\qquad\qquad \mapsto \langle res; [] \rangle \\
\qquad\qquad \langle \mathsf{Write}\ \langle _; cs \rangle \twoheadrightarrow resume \rangle \mapsto \mathbf{let}\ \langle res; file \rangle = resume\ \langle\rangle\ \mathbf{in} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \langle res; cs +\!\!+ file \rangle
\end{array}$$

The handler takes as input a computation that produces some value α, and in doing so may perform the BIO effect. The handler ultimately returns a pair consisting of the return value α and the final state of the file. The **return**-case pairs the result res with the empty file [] which models the scenario where the computation m performed no Write-operations, e.g. $\mathsf{basicIO}\ (\lambda\langle\rangle.\langle\rangle) \rightsquigarrow^{+} \langle\langle\rangle; \texttt{""}\rangle$. The Write-case extends the file by first invoking the resumption, whose return type is the same as the handler's return type, thus it returns a pair containing the result of m and the file state. The file gets extended with the character sequence cs before it is returned along with the original result of m. Intuitively, we may think of this implementation of Write as a peculiar instance of buffered writing, where the contents of the operation are committed to the file when the computation m finishes.

Let us define an auxiliary function that writes a string to the stdout file.

$$\begin{array}{l}
\mathsf{echo} : \mathsf{String} \to 1 ! \mathsf{BIO} \\
\mathsf{echo}\ cs := \mathbf{do}\ \mathsf{Write}\ \langle \mathsf{stdout}; cs \rangle
\end{array}$$

The function echo is a simple wrapper around an invocation of Write. We can now write some contents to the file and observe the effects.

```
    basicIO (λ⟨⟩.echo "Hello"; echo "World")
⇝+ ⟨⟨⟩; "HelloWorld"⟩ : ⟨1; File⟩
```

3.8.3 Exceptions: non-local exits

A process may terminate successfully by running to completion, or it may terminate with success or failure in the middle of some computation by performing an *exit* system call. The exit system call is typically parameterised by an integer value intended to indicate whether the exit was due to success or failure. By convention, UNIX interprets the integer zero as success and any nonzero integer as failure, where the specific value is supposed to correspond to some known error code.

We can model the exit system call by way of a single operation Exit.

```
Status := {Exit : Int ↠ 0}
```

The operation is parameterised by an integer value, however, an invocation of Exit can never return, because the type 0 is uninhabited. Thus Exit acts like an exception. It is convenient to abstract invocations of Exit to make it possible to invoke the operation in any context.

```
exit : Int → α!Status
exit n := absurd (do Exit n)
```

The **absurd** computation term is used to coerce the return type 0 of Exit into α. This coercion is safe, because 0 is an uninhabited type. An interpretation of Exit amounts to implementing an exception handler.

```
status : (1 → α!Status) → Int
status m :=  handle m ⟨⟩ with
               return _ ↦ 0
               ⟨Exit n⟩  ↦ n
```

Following the UNIX convention, the **return**-case interprets a successful completion of m as the integer 0. The operation case returns whatever payload the Exit operation was carrying. As a consequence, outside of status, an invocation of Exit 0 in m is indistinguishable from m returning normally, e.g. status (λ⟨⟩.exit 0) = status (λ⟨⟩.⟨⟩).

To illustrate status and exit in action consider the following example, where the computation gets terminated mid-way.

```
    basicIO (λ⟨⟩.status (λ⟨⟩.echo "dead"; exit 1; echo "code"))
⇝+ ⟨1; "dead"⟩ : ⟨Int; File⟩
```

The (delimited) continuation of exit 1 is effectively dead code. Here, we have a choice as to how we compose the handlers. Swapping the order of handlers would cause the whole computation to return just 1 : Int, because the status handler discards the return value of its computation. Thus with the alternative layering of handlers the system would throw away the file state after the computation finishes. However, in this particular instance the semantics the (local) behaviour of the operations Write and Exit would be unaffected if the handlers were swapped. In general the behaviour of operations may be affected by the order of handlers. The canonical example of this phenomenon is the composition of nondeterminism and state, which we will discuss in Section 3.8.2.

3.8.4 Dynamic binding: user-specific environments

When a process is run in UNIX, the operating system makes available to the process a collection of name-value pairs called the *environment*. The name of a name-value pair is known as an *environment variable*. During execution the process may perform a system call to ask the operating system for the value of some environment variable. The value of environment variables may change throughout process execution, moreover, the value of some environment variables may vary according to which user asks the environment. For example, an environment may contain the environment variable `USER` that is bound to the name of the enquiring user.

An environment variable can be viewed as an instance of dynamic binding. It is well-known that dynamic binding can be encoded as a computational effect by using delimited control [6]. Unsurprisingly, we will use this insight to simulate user-specific environments using effect handlers.

For simplicity we fix the users of the operating system to be root, Alice, and Bob.

$$\mathsf{User} := [\mathsf{Alice}; \mathsf{Bob}; \mathsf{Root}]$$

Our environment will only support a single environment variable intended to store the name of the current user. The value of this variable can be accessed via an operation $\mathsf{Ask} : 1 \twoheadrightarrow \mathsf{String}$. Using this operation we can readily implement the *whoami* utility from the GNU coreutils [14, Section 20.3], which returns the name of the current user.

$$\begin{array}{l}\mathsf{whoami} : 1 \to \mathsf{String}!\{\mathsf{Ask} : 1 \twoheadrightarrow \mathsf{String}\} \\ \mathsf{whoami}\ \langle\rangle := \mathbf{do}\ \mathsf{Ask}\ \langle\rangle\end{array}$$

The following handler implements the environment.

$$\begin{array}{l}\mathsf{env} : \langle \mathsf{User}; 1 \to \alpha!\{\mathsf{Ask} : 1 \twoheadrightarrow \mathsf{String}\}\rangle \to \alpha \\ \mathsf{env}\ \langle \mathit{user}; m\rangle := \ \mathbf{handle}\ m\ \langle\rangle\ \mathbf{with} \\ \qquad\qquad \mathbf{return}\ \mathit{res} \qquad\quad \mapsto \mathit{res} \\ \qquad\qquad \langle \mathsf{Ask}\ \langle\rangle \twoheadrightarrow \mathit{resume}\rangle \mapsto \mathbf{case}\ \mathit{user}\ \{\mathsf{Alice} \mapsto \mathit{resume}\ \texttt{"alice"} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathsf{Bob}\ \mapsto \mathit{resume}\ \texttt{"bob"} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathsf{Root} \mapsto \mathit{resume}\ \texttt{"root"}\}\end{array}$$

The handler takes as input the current *user* and a computation that may perform the Ask operation. When an invocation of Ask occurs the handler pattern matches on the *user* parameter and resumes with a string representation of the user. With this implementation we can interpret an application of whoami.

$$\mathsf{env}\ \langle \mathsf{Root}; \mathsf{whoami}\rangle \rightsquigarrow^{+} \texttt{"root"} : \mathsf{String}$$

It is not difficult to extend this basic environment model to support an arbitrary number of variables. This can be done by parameterising the Ask operation by some name representation (e.g. a string), which the environment handler can use to index into a list of string values. In case the name is unbound the environment, the handler can embrace the laissez-faire attitude of UNIX and resume with the empty string.

3.8.4.1 User session management

It is somewhat pointless to have multiple user-specific environments, if the system does not support some mechanism for user session handling, such as signing in as a different user. In UNIX the command *substitute user* (su) enables the invoker to impersonate another

user account, provided the invoker has sufficient privileges. We will implement su as an operation $\mathsf{Su} : \mathsf{User} \twoheadrightarrow 1$ which is parameterised by the user to be impersonated. To model the security aspects of su, we will use the weakest possible security model: unconditional trust. Put differently, we will not bother with security at all to keep things relatively simple. Consequently, anyone can impersonate anyone else.

The session signature consists of two operations, Ask, which we used above, and Su, for switching user.

$$\mathsf{Session} := \{\mathsf{Ask} : 1 \twoheadrightarrow \mathsf{String}; \mathsf{Su} : \mathsf{User} \twoheadrightarrow 1\}$$

As usual, we define a small wrapper around invocations of Su.

$$\begin{array}{l}\mathsf{su} : \mathsf{User} \to 1!\{\mathsf{Su} : \mathsf{User} \twoheadrightarrow 1\}\\ \mathsf{su}\ \mathit{user} := \mathbf{do}\ \mathsf{Su}\ \mathit{user}\end{array}$$

The intended operational behaviour of an invocation of Su *user* is to load the environment belonging to *user* and continue the continuation under this environment. We can achieve this behaviour by defining a handler for Su that invokes the provided resumption under a fresh instance of the env handler.

$$\begin{array}{l}\mathsf{sessionmgr} : \langle \mathsf{User}; 1 \to \alpha!\mathsf{Session}\rangle \to \alpha\\ \mathsf{sessionmgr}\ \langle \mathit{user}; m\rangle := \mathsf{env}\langle \mathit{user}; (\lambda\langle\rangle.\mathbf{handle}\ m\ \langle\rangle\ \mathbf{with}\\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathbf{return}\ \mathit{res} \mapsto \mathit{res}\\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \langle \mathsf{Su}\ \mathit{user}' \twoheadrightarrow \mathit{resume}\rangle \mapsto \mathsf{env}\langle \mathit{user}'; \mathit{resume}\rangle)\rangle\end{array}$$

The function sessionmgr manages a user session. It takes two arguments: the initial user (*user*) and the computation (m) to run in the current session. An initial instance of env is installed with *user* as argument. The computation argument is a handler for Su enclosing the computation m. The Su-case installs a new instance of env, which is the environment belonging to user', and runs the resumption *resume* under this instance. The new instance of env shadows the initial instance, and therefore it will intercept and handle any subsequent invocations of Ask arising from running the resumption. A subsequent invocation of Su will install another environment instance, which will shadow both the previously installed instance and the initial instance.

To make this concrete, let us plug together the all components of our system we have defined thus far.

$$\begin{array}{l}\quad \mathsf{basicIO}\,(\lambda\langle\rangle.\\ \qquad \mathsf{sessionmgr}\,\langle \mathsf{Root}; \lambda\langle\rangle.\\ \qquad\quad \mathsf{status}\,(\lambda\langle\rangle.\mathsf{su}\ \mathsf{Alice};\ \mathsf{echo}\,(\mathsf{whoami}\,\langle\rangle);\ \mathsf{echo}\ \texttt{" "};\\ \qquad\qquad\qquad\quad \mathsf{su}\ \mathsf{Bob};\ \ \mathsf{echo}\,(\mathsf{whoami}\,\langle\rangle);\ \mathsf{echo}\ \texttt{" "};\\ \qquad\qquad\qquad\quad \mathsf{su}\ \mathsf{Root};\ \mathsf{echo}\,(\mathsf{whoami}\,\langle\rangle)))\rangle)\\ \rightsquigarrow^{+} \langle 0; \texttt{"alice bob root"}\rangle : \langle \mathsf{Int}; \mathsf{File}\rangle\end{array}$$

The session manager (sessionmgr) is installed in between the basic IO handler (basicIO) and the process status handler (status). The initial user is Root, and thus the initial environment is the environment that belongs to the root user. Main computation signs in as Alice and writes the result of the system call whoami to the global file, and then repeats these steps for Bob and Root. Ultimately, the computation terminates successfully (as indicated by 0 in the first component of the result) with global file containing the three user names.

The above example demonstrates that we now have the basic building blocks to build a multi-user system.

3.8.5 Nondeterminism: time sharing

Time sharing is a mechanism that enables multiple processes to run concurrently, and hence, multiple users to work concurrently. Thus far in our system there is exactly one process. In UNIX there exists only a single process whilst the system is bootstrapping itself into operation. After bootstrapping is complete the system duplicates the initial process to start running user managed processes, which may duplicate themselves to create further processes. The process duplication primitive in UNIX is called *fork* [2]. The fork-invoking process is typically referred to as the parent process, whilst its clone is referred to as the child process. Following an invocation of fork, the parent process is provided with a nonzero identifier for the child process and the child process is provided with the zero identifier. This enables processes to determine their respective role in the parent-child relationship, e.g.

$$\begin{array}{l}\textbf{let } i \leftarrow \mathit{fork}\ \langle\rangle \textbf{ in}\\ \textbf{if } i = 0 \textbf{ then } \ \mathit{child's\ code}\\ \textbf{else } \ \mathit{parent's\ code}\end{array}$$

In our system, we can model fork as an effectful operation, that returns a boolean to indicate the process role; by convention we will interpret the return value true to mean that the process assumes the role of parent.

$$\begin{array}{l}\mathsf{fork} : 1 \to \mathsf{Bool}!\{\mathsf{Fork} : 1 \twoheadrightarrow \mathsf{Bool}\}\\ \mathsf{fork}\ \langle\rangle := \textbf{do}\ \mathsf{Fork}\ \langle\rangle\end{array}$$

In UNIX the parent process *continues* execution after the fork point, and the child process *begins* its execution after the fork point. Thus, operationally, we may understand fork as returning twice to its invocation site. We can implement this behaviour by invoking the resumption arising from an invocation of Fork twice: first with true to continue the parent process, and subsequently with false to start the child process (or the other way around if we feel inclined). The following handler implements this behaviour.

$$\begin{array}{l}\mathsf{nondet} : (1 \to \alpha!\{\mathsf{Fork} : 1 \twoheadrightarrow \mathsf{Bool}\}) \to \mathsf{List}\ \alpha\\ \mathsf{nondet}\ m := \textbf{handle}\ m\ \langle\rangle\ \textbf{with}\\ \qquad\qquad \textbf{return}\ \mathit{res} \qquad\qquad \mapsto [\mathit{res}]\\ \qquad\qquad \langle \mathsf{Fork}\ \langle\rangle \twoheadrightarrow \mathit{resume}\rangle \mapsto \mathit{resume}\ \mathsf{true} +\!\!+ \mathit{resume}\ \mathsf{false}\end{array}$$

The **return**-case returns a singleton list containing a result of running m. The Fork-case invokes the provided resumption resume twice. Each invocation of resume effectively copies m and runs each copy to completion. Each copy returns through the **return**-case, hence each invocation of resume returns a list of the possible results obtained by interpreting Fork first as true and subsequently as false. The results are joined by list concatenation ($+\!\!+$). Thus the handler returns a list of all the possible results of m. (Remark: this handler is an instance of the standard backtracking nondeterminism handler from the literature, which has been used in related work to show that effect handlers endow their host language with additional asymptotic computational efficiency [13].)

Let us consider nondet together with the previously defined handlers. But first, let us define two computations.

$$\mathsf{ritchie},\ \mathsf{hamlet} : 1 \to 1!\{\mathsf{Write} : \langle \mathsf{FileDescr}; \mathsf{String}\rangle \twoheadrightarrow 1\}$$

```
ritchie ⟨⟩ := echo "UNIX is basically ";
              echo "a simple operating system, ";
              echo "but ";
              echo "you have to be a genius
                    to understand the simplicity.\n"
hamlet ⟨⟩ := echo "To be, or not to be, ";
             echo "that is the question:\n";
             echo "Whether 'tis nobler in the mind to suffer\n"
```

The computation ritchie writes a quote by Dennis Ritchie to the file, whilst the computation hamlet writes a few lines of William Shakespeare's *The Tragedy of Hamlet, Prince of Denmark*, Act III, Scene I [1] to the file. Using nondet and fork together with the previously defined infrastructure, we can fork the initial process such that both of the above computations are run concurrently.

```
basicIO (λ⟨⟩.
    nondet (λ⟨⟩.
        sessionmgr ⟨Root; λ⟨⟩.
            status (λ⟨⟩.if fork ⟨⟩ then su Alice; ritchie ⟨⟩
                       else su Bob; hamlet ⟨⟩))))
⇝+ ⟨[0, 0]; "UNIX is basically a simple operating system, but
            you have to be a genius to understand the simplicity.\n
            To be, or not to be, that is the question:\n
            Whether 'tis nobler in the mind to suffer\n"⟩ : ⟨List Int; File⟩
```

The computation running under the status handler immediately performs an invocation of fork, causing nondet to explore both the **then**-branch and the **else**-branch. In the former, Alice signs in and quotes Ritchie, whilst in the latter Bob signs in and quotes a Hamlet. Looking at the output there is supposedly no interleaving of computation, since the individual writes have not been interleaved. From the stack of handlers, we *know* that there has been no interleaving of computation, because no handler in the stack handles interleaving. Thus, our system only supports time sharing in the extreme sense: we know from the nondet handler that every effect of the parent process will be performed and handled before the child process gets to run. In order to be able to share time properly amongst processes, we must be able to interrupt them.

3.8.5.1 Interleaving computation

We need an operation for interruptions and corresponding handler to handle interrupts in order for the system to support interleaving of processes.

```
interrupt : 1 → 1!{Interrupt : 1 ↠ 1}
interrupt ⟨⟩ := do Interrupt ⟨⟩
```

The intended behaviour of an invocation of Interrupt is to suspend the invoking computation in order to yield time for another computation to run. We can achieve this behaviour by reifying the process state. For the purpose of interleaving processes via interruptions it suffices to view a process as being in either of two states: 1) it is done, that is it has run to completion, or 2) it is paused, meaning it has yielded to provide room for another process to run. We can model the state using a recursive variant type parameterised by some return value α and a set of effects ε that the process may perform.

```
Pstate α ε θ := [Done : α;
                 Paused : 1 → Pstate α ε θ!{Interrupt : θ; ε}]
```

This data type definition is an instance of the *resumption monad* [3]. The Done-tag simply carries the return value of type α. The Paused-tag carries a suspended computation, which returns another instance of Pstate, and may or may not perform any further invocations of Interrupt. Payload type of Paused is precisely the type of a resumption originating from a handler that handles only the operation Interrupt such as the following handler.

```
reifyProcess : (1 → α!{Interrupt : 1 ↠ 1; ε}) → Pstate α ε
reifyProcess m := handle m ⟨⟩ with
                    return res                  ↦ Done res
                    ⟨Interrupt ⟨⟩ ↠ resume⟩ ↦ Paused resume
```

This handler tags and returns values with Done. It also tags and returns the resumption provided by the Interrupt-case with Paused. This particular implementation is amounts to a handler-based variation of Harrison's [5] non-reactive resumption monad. If we compose this handler with the nondeterminism handler, then we obtain a term with the following type.

$$\mathsf{nondet}\,(\lambda\langle\rangle.\mathsf{reifyProcess}\ m) : \mathsf{List}\ (\mathsf{Pstate}\ \alpha\ \{\mathsf{Fork} : 1 \twoheadrightarrow \mathsf{Bool}; \varepsilon\})$$

for some $m : 1 \to \{\mathsf{Proc}; \varepsilon\}$ where $\mathsf{Proc} := \{\mathsf{Fork} : 1 \twoheadrightarrow \mathsf{Bool}; \mathsf{Interrupt} : 1 \twoheadrightarrow 1\}$. The composition yields a list of process states, some of which may be in suspended state. In particular, the suspended computations may have unhandled instances of Fork as signified by it being present in the effect row. The reason for this is that in the above composition when reifyProcess produces a Paused-tagged resumption, it immediately returns through the **return**-case of nondet, meaning that the resumption escapes the nondet. Recall that a resumption is a delimited continuation that captures the extent from the operation invocation up to and including the nearest enclosing suitable handler. In this particular instance, it means that the nondet handler is part of the extent. We ultimately want to return just a list of αs to ensure every process has run to completion. To achieve this, we need a function that keeps track of the state of every process, and in particular it must run each Paused-tagged computation under the nondet handler to produce another list of process state, which must be handled recursively.

$$\begin{array}{l}
\mathsf{schedule} : \mathsf{List}\ (\mathsf{Pstate}\ \alpha\ \{\mathsf{Fork} : \mathsf{Bool}; \varepsilon\}\ \theta) \to \mathsf{List}\ \alpha!\varepsilon \\
\mathsf{schedule}\ ps := \mathbf{let}\ run \leftarrow \mathbf{rec}\ sched\,\langle ps; done\rangle. \\
\qquad\qquad \mathbf{case}\ ps\ \{\qquad\qquad [] \mapsto done \\
\qquad\qquad\qquad (\mathsf{Done}\ res) :: ps' \mapsto sched\,\langle ps'; res :: done\rangle \\
\qquad\qquad\qquad (\mathsf{Paused}\ m) :: ps' \mapsto sched\,\langle ps' +\!\!+ (\mathsf{nondet}\ m); done\rangle\} \\
\qquad \mathbf{in}\ run\,\langle ps; []\rangle
\end{array}$$

The function schedule implements a process scheduler. It takes as input a list of process states, where Paused-tagged computations may perform the Fork operation. Locally it defines a recursive function *sched* which carries a list of active processes *ps* and the results of completed processes *done*. The function inspects the process list *ps* to test whether it is empty or nonempty. If it is empty it returns the list of results *done*. Otherwise, if the head is Done-tagged value, then the function is recursively invoked with tail of processes ps' and the list *done* augmented with the value *res*. If the head is a Paused-tagged computation m, then *sched* is recursively invoked with the process list ps' concatenated with the result of running m under the nondet handler.

Using the above machinery, we can define a function which adds time-sharing capabilities to the system.

$$\begin{array}{l}
\mathsf{timeshare} : (1 \to \alpha!\mathsf{Proc}) \to \mathsf{List}\ \alpha \\
\mathsf{timeshare}\ m := \mathsf{schedule}\,[\mathsf{Paused}\,(\lambda\langle\rangle.\mathsf{reifyProcess}\ m)]
\end{array}$$

The function timeshare handles the invocations of Fork and Interrupt in some computation m by starting it in suspended state under the reifyProcess handler. The schedule actually starts the computation, when it runs the computation under the nondet handler.

The question remains how to inject invocations of Interrupt such that computation gets interleaved. The interested reader may consult my dissertation for a discussion of different ways to inject interrupts, and for a more complete development of Tiny UNIX with file I/O, process synchronisation, programmable shell environment via shallow handlers, and more, as well as a discussion of ways to realise effect handlers, and hence, the operating system using canonical implementation techniques [8, 9, 11, 12, 15].

References

1 William Shakespeare. *The Tragedy of Hamlet, Prince of Denmark*. 1564-1616
2 Dennis Ritchie and Ken Thompson. *The UNIX Time-Sharing System*. Commun. ACM, 17, 1974
3 Nikolaos S. Papspyrou. *A resumption monad transformer and its applications in the semantics of concurrency* Proceedings of the 3rd Panhellenic Logic Symposium, Anogia, Greece, 2001
4 Eric Steven Raymond. *The Art of UNIX Programming*. ISBN 0131429019. Pearson Education, 2003
5 William L. Harrison. *The Essence of Multitasking*. AMAST, LNCS, 2006
6 Oleg Kiselyov, Chung-chieh Shan, and Amr Sabry. *Delimited dynamic binding*. ICFP, Portland, Oregon, USA, 2006
7 Ohad Kammar, Sam Lindley, and Nicolas Oury. *Handlers in action*. ICFP, Boston, Massachusetts, USA, 2013
8 Daniel Hillerström and Sam Lindley. *Liberating Effects with Rows and Handlers*. TyDe@ICFP, Nara, Japan, 2016
9 Daniel Hillerström, Sam Lindley, Robert Atkey, and KC Sivaramakrishnan. *Continuation Passing Style for Effect Handlers*. FSCD, Oxford, UK, 2017
10 Daan Leijen. *Implementing Algebraic Effects in C – "Monads for Free in C"*. APLAS, Suzhou, China, 2017
11 Daniel Hillerström and Sam Lindley. *Shallow Effect Handlers*. APLAS, New Zealand, 2018
12 Daniel Hillerström, Sam Lindley, and Robert Atkey. *Effect Handlers via Generalised Continuations*. JFP (special issue on algebraic effects and handlers) 30:e5, 2020
13 Daniel Hillerström, Sam Lindley, and John Longley. *Effects for Efficiency: Asymptotic Speedup with First-Class Control*. ICFP, New Jersey, USA, 2020
14 David MacKenzie and others. *GNU Coreutils (for version 8.32)*. Free Software Foundation, 2020
15 Daniel Hillerström. *Foundations for Programming and Implementing Effect Handlers*. PhD thesis, The University of Edinburgh, UK, 2021

3.9 ParaFuzz: Fuzzing Multicore OCaml Programs

Sivaramakrishnan Krishnamoorthy Chandrasekaran (Indian Institute of Techology, IN)

Joint work of Sivaramakrishnan Krishnamoorthy Chandrasekaran, Sumit Padhiyar, Adharsh Kamath

Parallel programs are notoriously hard to test due to the particular combination of input and scheduling non-determinsm. Techniques such as property-based testing and fuzz testing are extremely effective for handling input non-determinsm. Crowbar is a tool for OCaml which combines property-based testing and fuzz testing for OCaml programs. Can we extend this to capture scheduling non-determinism? The answer is yes, and ParaFuzz shows how. A key challenge is getting control over the thread scheduling decisions. We should how effect handlers can help with this.

3.10 Retrofitting Effect Handlers onto OCaml

Sivaramakrishnan Krishnamoorthy Chandrasekaran (Indian Institute of Techology, IN)

Joint work of Krishnamoorthy Chandrasekaran Sivaramakrishnan, Stephen Dolan, Leo White, Tom Kelly, Sadiq Jaffer, Anil Madhavapeddy
Main reference K. C. Sivaramakrishnan, Stephen Dolan, Leo White, Tom Kelly, Sadiq Jaffer, Anil Madhavapeddy: "Retrofitting effect handlers onto OCaml", in Proc. of the PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual Event, Canada, June 20-25, 20211, pp. 206–221, ACM, 2021.
URL https://doi.org/10.1145/3453483.3454039

Multicore OCaml extends OCaml with support for effect handlers in order to express concurrency natively in direct-style. Given that we're extending an industrial-strength language with millions of lines of existing code, none of which is written with non-local control-flow in mind, our primary concern is backwards compatibility. Specifically, (a) not breaking legacy code, (b) retaining the performance profile of legacy code, and (c) debugging and profiling tool compatibility. In this talk, I shall discuss the backwards compatibility challenges and our solutions in Multicore OCaml.

3.11 Koka update: Compilation to C via generalized evidence passing and Perceus reference counting.

Daan Leijen (Microsoft Research – Redmond, US)

Koka can now compile effect handlers to standard C code; it uses a generalized evidence passing in combination with a multi-prompt delimited control monad to compile effect handlers (ICFP21). Moreover, it uses compile-time optimized reference counting (PLDI21) to manage memory without needing a GC or runtime system. I will show some of the new Koka language features, highlight the interesting parts of the compilation phases, and show various benchmarks.

3.12 Handler Calculus

Sam Lindley (University of Edinburgh, GB)

We present handler calculus, a core calculus of effect handlers. Inspired by the Frank programming language, handler calculus does not have primitive functions, just handlers. Functions, products, sums, and inductive types, are all encodable in handler calculus. We extend handler calculus with recursive effects, which we use to encode recursive data types. We extend handler calculus with parametric operations, which we use to encode existential data types. We then briefly outline how one can encode universal data types by composing a CPS translation for parametric handler calculus into System F with Fujita's CPS translation of System F into minimal existential logic.

3.13 Efficient Compilation of Algebraic Effect Handlers

Matija Pretnar (University of Ljubljana, SI)

Joint work of Georgios Karachalias, Filip Koprivec, Matija Pretnar, Tom Schrijvers
Main reference Georgios Karachalias, Filip Koprivec, Matija Pretnar, Tom Schrijvers: "Efficient Compilation of Algebraic Effect Handlers", Proc. ACM Program. Lang., Vol. 5(OOPSLA), Association for Computing Machinery, 2021.
URL https://doi.org/10.1145/3485479

The popularity of algebraic effect handlers as a programming language feature for user-defined computational effects is steadily growing. Yet, even though efficient runtime representations have already been studied, most handler-based programs are still much slower than hand-written code.

In the talk, I have presented our OOPSLA submission, which shows that the performance gap can be drastically narrowed (in some cases even closed) by means of type-and-effect directed optimising compilation. Our approach consists of source-to-source transformations in two phases of the compilation pipeline. Firstly, elementary rewrites, aided by judicious function specialisation, exploit the explicit type and effect information of the compiler's core language to aggressively reduce handler applications. Secondly, after erasing the effect information further rewrites in the backend of the compiler emit tight code.

This work comes with a practical implementation: an optimising compiler from Eff, an ML style language with algebraic effect handlers, to OCaml. Experimental evaluation with this implementation demonstrates that in a number of benchmarks, our approach eliminates much of the overhead of handlers, outperforms capability-passing style compilation and yields competitive performance compared to hand-written OCaml code as well Multicore OCaml's dedicated runtime support.

3.14 Programming and Proving with Indexed effects in F*

Aseem Rastogi (Microsoft Research India – Bangalore, IN) and Nikhil Swamy (Microsoft Research – Redmond, US)

Joint work of Guido Martínez, Aymeric Fromherz, Tahina Ramananandro
Main reference Aseem Rastogi, Guido Martínez, Aymeric Fromherz, Tahina Ramananandro, Nikhil Swamy: "Programming and Proving with Indexed Effects"
URL https://www.fstar-lang.org/papers/indexedeffects/

F* now supports a feature that allows programmers to define monadic effects with an arbitrary indexing structure. We have been using this to program and prove a variety of systems, using custom effect-typing disciplines, combining various prior approaches in novel ways. For example, we've been developing graded parameterized monads, or parameterized Dijkstra monads, graded Dijkstra monads, and parameterized-monad-indexed monads, and other seemingly exotic but very useful constructions. We've applied them to settings ranging from information flow control, to parsers, to separation logic, and to algebraic effects. The talk is intended to tell people about these structures, point out how one can program with them in F*, get feedback about it, and hopefully interest folks to develop new such structures, to use them in practice, and to study their semantics.

3.15 Low-level effect handlers for Wasm

Andreas Rossberg (Dfinity – Zürich, CH)

Joint work of Andreas Rossberg, Daniel Hillerström, Sam Lindley, KC Sivaramakrishnan, Matija Pretnar, Daan Leijen
URL https://github.com/effect-handlers/wasm-spec

I presented the ongoing work on a proposal for adding low-level effect handlers to Wasm.

3.16 Back to Direct Style 3

Philipp Schuster (Universität Tübingen, DE)

Joint work of Philipp Schuster, Jonathan Immanuel Brachthäuser, Marius Müller, Klaus Ostermann

Programs in continuation-passing style are good to optimize but bad to run. We present a program transformation that goes from continuation-passing style back to direct style. It is a continuation of the "Back to Direct Style" line of work by Danvy and Lawall. We present a language with a type-and-effect system where it is possible to have multiple levels of control. Just like we can iterate the CPS transformation to make more and more levels of control explicit, we can iterate the direct-style transformation, to make more and more levels of control implicit. What we present is "work in progress" and we would like to discuss the approach in general, possible applications, and a logical interpretation with the audience.

3.17 CPS Transformation with Affine Types for Call-By-Value Implicit Polymorphism

Taro Sekiyama (National Institute of Informatics – Tokyo, JP)

Joint work of Taro Sekiyama, Tsukada, Takeshi
Main reference Taro Sekiyama, Takeshi Tsukada: "CPS transformation with affine types for call-by-value implicit polymorphism", Proc. ACM Program. Lang., Vol. 5(ICFP), pp. 1–30, 2021.
URL https://doi.org/10.1145/3473600

Transformation of programs into continuation-passing style (CPS) reveals the notion of continuations, enabling many applications such as control operators and intermediate representations in compilers. Although type preservation makes CPS transformation more beneficial, achieving type-preserving CPS transformation for implicit polymorphism with call-by-value (CBV) semantics is known to be challenging. We identify the difficulty in the problem that we call scope intrusion. To address this problem, we propose a new CPS target language $\wedge^{open}$ that supports two additional constructs for polymorphism: one only binds and the other only generalizes type variables. Unfortunately, their unrestricted use makes $\wedge^{open}$ unsafe due to undesired generalization of type variables. We thus equip $\wedge^{open}$ with affine types to allow only the type-safe generalization. We then define a CPS transformation from Curry-style CBV System F to type-safe $\wedge^{open}$ and prove that the transformation is meaning and type preserving. We also study parametricity of $\wedge^{open}$ as it is a fundamental

property of polymorphic languages and plays a key role in applications of CPS transformation. To establish parametricity, we construct a parametric, step-indexed Kripke logical relation for $\wedge^{open}$ and prove that it satisfies the Fundamental Property as well as soundness with respect to contextual equivalence.

3.18 Effects with Shifted Names in OCaml

Antal Spector-Zabusky (Jane Street – London, GB)

Joint work of Antal Spector-Zabusky, Stephan Dolan, Leo White

We are currently designing an effect system for OCaml consisting of algebraic effects with a fused "resume a continuation inside a handler" operation. We use shifted names to name effects, allowing operations that abstract over names to avoid shadowing names in the surrounding environment via renaming. This talk presents the design of the runtime semantics of this language as they currently stand.

3.19 Effects, Interface Types and async APIs

Luke Wagner (Fastly – San Francisco, US)

One focus of WASI right now is on HTTP APIs and supporting efficient request chaining via simple module linking/composition. Due to the streaming async nature of HTTP request handling, effects/coroutines are a natural fit. Expressing async APIs in a cross-language-compositional manner is challenging, though, when most of the constituent languages don't directly support algebraic effects. This talk discusses an idea we're working on for how to reconcile these constraints by building in a fixed 'async' effect to Interface Types that can be thought of as a specialized use of algebraic effects. When bound to JavaScript, Interface-Typed async functions would naturally bind to JavaScript async (i.e., Promise-returning) functions in a manner similar to the current wasm stack-switching JS API proposal.

4 Working groups

4.1 Control Operators Breakout Session

Jonathan Immanuel Brachthäuser (EPFL – Lausanne, CH), Youyou Cong (Tokyo Institute of Technology, JP), Sam Lindley (University of Edinburgh, GB), and Taro Sekiyama (National Institute of Informatics – Tokyo, JP)

In this breakout session, we explored the correspondence between effect handlers and delimited control operators from different perspectives. One question we discussed is what is the effect-handler-counterpart of shift/reset and control/prompt. These control operators keep the

surrounding delimiter upon capture of a continuation, while effect handlers remove the surrounding handler upon a call of an operation. Sam suggested that such effect handlers may be useful for implementing the fork and yield operations, where we need a form of recursion, but we found that the control operators do not have enough expressive power.

Inspired by Sam's idea, we had a discussion on the correspondence between effect handlers and recursion schemes. Jonathan drafted a version of effect handlers that could correspond to histo-morphisms. In this sketch it appears histo-morphic effect handlers support a combination of the usual (deep) resumptions and shallow resumptions at each effect call.

We also talked about what is the control-operator counterpart of multi-handlers. Multi-handlers can handle multiple computations at once, which is difficult to express using shift/reset-style control operators. Daniel suggested that fcontrol/run may be easier to work with, and Youyou successfully implemented a "bi-handler" (handlers that can handle two computations) using these operators.

4.2 UX of Effect Systems Breakout Session

Jonathan Immanuel Brachthäuser (EPFL – Lausanne, CH), Youyou Cong (Tokyo Institute of Technology, JP), Paulo Emílio de Vilhena (INRIA – Paris, FR), and Filip Koprivec (University of Ljubljana, SI)

In this breakout session, we had a discussion on teaching effect systems. As a scenario where effect systems can be useful, Conor suggested building an OS, and April suggested developing GUIs (especially Web applications). As a tool for helping students understand effects, Youyou introduced an algebraic stepper developed at Ochanomizu University, and Matija introduced a similar tool supported in the aeff language. After the seminar, Nick, Jonathan, and Youyou had a meeting on the curriculum design of an effect handler course. There is also a plan to write a textbook called "How to Design Effectful Programs", which defines a series of design recipes for effect constructs.

4.3 Effect Handlers Benchmark Suite

Daniel Hillerström (University of Edinburgh, GB)

URL https://github.com/effect-handlers/effect-handlers-bench

At the moment, a lot of work is about efficient runtime systems, or compilation, for effect handlers. However, as identified by this working group there is no standard benchmark suite for effect handler oriented programs. The literature makes use of a varying collection of ad-hoc benchmarks. This working group has begun the effort to create a community-maintained standardised benchmark suite for effect handler oriented programs. By standardised, we mean that the suite will contain a set of benchmarks intended to measure different aspects of effect handlers, e.g. single-shot, multi-shot, tail-resumptive handlers, etc, and each benchmark will have a description of its objective, how it should be realised (e.g. common implementation), and its parameters.

The benchmark suite is being actively developed on GitHub on the following repository.
`https://github.com/effect-handlers/effect-handlers-bench`

4.4 Wasm breakout session

Andreas Rossberg (Dfinity – Zürich, CH), Sam Lindley (University of Edinburgh, GB), and Luke Wagner (Fastly – San Francisco, US)

4.4.1 Introduction

The main purpose for adding effect handlers to WebAssembly is to provide a well-behaved mechanism for "stack switching". The proposal uses the asymmetric suspend/resume pair of primitives that is characteristic of handlers. This has been criticised for lacking a symmetric way of switching to another continuation directly, without going through a handler, and there is some concern that the double hop through a handler might involve unnecessary overhead for use cases like lightweight threading.

We discussed an idea, originally brought up by Luke Wagner, for extending the proposal with a more symmetric `switch_to` primitive. In fact, this can be broken down into two independent mechanisms:

1. Naming individual handlers, as a way of targeting them directly with a suspend, and thereby avoiding the linear search for a handler (somewhat similar to multi-prompt continuations).
2. A special built-in effect that switches to another continuation and is implicitly handled by every handler (or can be declared to be).

In addition, we discussed the possibility of first-class effect tags.

4.4.2 Named handlers

The idea here is to introduce a new reference type `(handler t*)`, which essentially is a unique prompt created by executing a variant of the resume instruction and is passed to the continuation:

```
cont.resume_from (event $tag $handler)* : [ t1* (cont $ft) ] -> [ t2* ]
where:
 -- $ft = [ (handler t2*) t1* ] -> [ t2* ]
```

The handler reference is similar to a prompt in a system of multi-prompt continuations. However, since its created fresh for each handler, multiple activations of the same prompt cannot exist by construction.

This instruction is complemented by an instruction for suspending to a specific handler:

```
cont.suspend_to $tag : [ t1* (handler t3*) ] -> [ t2* ]
where:
-- $tag : [ t1* ] -> [ t2* ]
```

If the handler is not currently active, e.g., because an outer handler has been suspended, then this instruction would trap.

We briefly pondered over the possibility of also an additional instruction to terminate a handler:

```
cont.return_to : [ t3* t1* (handler t1*) ] -> [ t2* ]
```

However, this would be like a throw, but without the ability to catch it. IT would therefore introduce yet another form of control flow transfer, whose interaction with other control operators (e.g., finally) would have to be considered. We concluded that it is preferable for the time being not to go there.

4.4.3 Direct switching

Given named handlers, it is possible to introduce a slightly more magic instruction for switching directly to another continuation:

```
cont.switch_to : [ t1* (cont $ft1) (handler t3*) ] -> [ t2* ]
where:
 -- $ft1 = [ (handler t3*) (cont $ft2) t1* ] -> [ t3* ]
 -- $ft2 = [ t2* ] -> [ t3* ]
```

This behaves as if there was a built-in tag

```
(tag Switch (param t1* (cont $ft1)) (result t3*))
```

with which the computation `suspends_to` the handler, and the handler implicitly handles this by `resuming_to` the continuation argument, thereby effectively switching to it in one step. Like `suspend_to`, this would trap if the handler wasn't currently active.

The fact that the handler implicitly resumes_to, passing itself as a handler to the target continuation, makes this construct behave like a deep handler, which is slightly add odds with the rest of the proposal.

In addition to the handler, `switch_to` also passed the new continuation to the target, which would allow the target to switch_to back to it in a symmetric fashion. Notably, in such a use case, `$ft1` and `$ft2` would be the same type (and hence recursive).

One observation we made is that symmetric switching is not necessarily tied to named handlers, since there could also be an indirect version with dynamic handler lookup:

```
cont.switch : [ t1* (cont $ft1) ] -> [ t2* ]
where:
-- $ft1 = [ (cont $ft2) t1* ] -> [ t3* ]
-- $ft2 = [ t2* ] -> [ t3* ]
```

Finally, it seems undesirable that every handler implicitly handles the built-in Switch tag, so this should be opt-in by a mode flag on the resume instruction(s).

4.4.4 First-class effect tags

We also discussed the possibility of having first-class effect tags. This would address a different but overlapping set of use cases compared to named handlers.

It would take the introduction of a new form of structured type, a tag type, and an instruction to generate fresh tags of such a type:

```
(type $tagtype (tag ...))

tag.new $tagtype : [] -> [(ref $tagtype)]
```

To be useful, though, this would require a new variant of resume instruction, whose handler table is created dynamically from its tag operands:

```
cont.resume (event $handler)* : [ (ref $tt)* t1* (cont $ft) ] -> [ t2* ]
where:
 -- ($tt = tag ...)*
 -- $ft = [ t1* ] -> [ t2* ]
```

Since the dispatch table has to be created dynamically at each execution of this instruction, it might be quite expensive in practice, especially since the handlers in the proposal behave like shallow handlers, i.e., must be recreated for every resumption. Also, this cannot easily be circumvented by adding first-class handlers, since the latter are made difficult because of the local nature of the branch labels handlers depend on. More investigation is needed.

4.5 Dependent types breakout session

Wouter Swierstra (Utrecht University, NL) and Robert Atkey (University of Strathclyde – Glasgow, GB)

In this session we discussed the various approaches to modelling effects and handlers accurately using rich types, typically involving some variation of monads such as parametrised monads, indexed monads, and graded monads. These can often be embedded in an existing programming language – but languages such as F^* add native support for collecting and resolving the proof obligations associated with certain effectful computations.

5 Open problems

5.1 Efficient stack layout for multishot handlers

Filip Koprivec (University of Ljubljana, SI)

Joint work of Filip Koprivec, Matija Pretnar

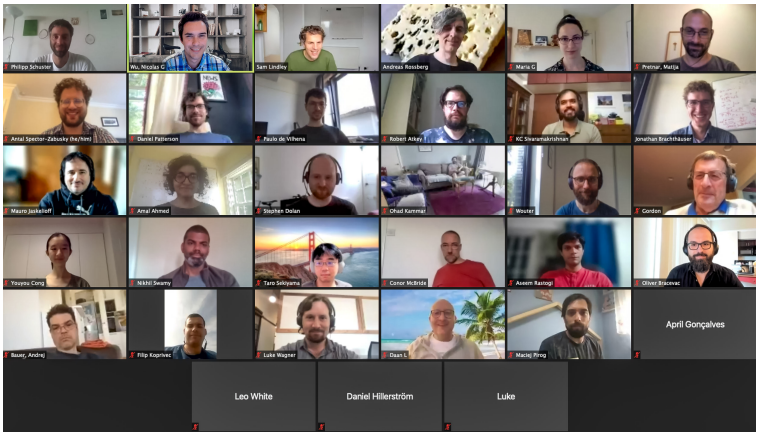
Much has been done on optimizing the performance of effect handlers, from an optimized runtime and evidence translation in Koka to a specialized stack structure in Multicore OCaml. We proposed an efficient stack management technique based on heap-allocated fibres by Sivaramakrishnan and others. We present a "work in progress" idea for a stack structure specialized for multiple resumptions.

The program stack is stored as a sequence of fibres corresponding either to computation or a handled effect. Once the computation is handled by an enclosing handler, the whole part of the stack corresponding to that computation is "frozen" and specifically marked for a copy on reuse when invoking the continuation. When the continuation is resumed before the frozen computation gets popped from the stack, there is no need to allocate any heap storage for the environment of continued computation as frozen fibre gets copied from the stack to the top of the stack directly.

This decreases the pressure on both allocator and garbage collector while reusing already allocated stack memory. Reuse of stack saved computations is faster than allocation on the heap and this especially improves performance when reusing the same continuation multiple times. The improvement an optimized stack structure offers is heavily dependent on handler usage and memory allocator performance.

Participants

- Danel Ahman
University of Ljubljana, SI
- Amal Ahmed
Northeastern University – Boston, US
- Robert Atkey
University of Strathclyde – Glasgow, GB
- Andrej Bauer
University of Ljubljana, SI
- Oliver Bracevac
Purdue University – West Lafayette, US
- Jonathan Immanuel Brachthäuser
EPFL – Lausanne, CH
- Youyou Cong
Tokyo Institute of Technology, JP
- Paulo Emílio de Vilhena
INRIA – Paris, FR
- Stephen Dolan
Jane Street – London, GB
- Ronald Garcia
University of British Columbia – Vancouver, CA
- April Gonçalves
Heliax – Glasgow, GB
- Maria Gorinova
University of Edinburgh, GB
- Daniel Hillerström
University of Edinburgh, GB
- Mauro Jaskelioff
National University of Rosario, AR
- Ohad Kammar
University of Edinburgh, GB
- Oleg Kiselyov
Tohoku University – Sendai, JP
- Filip Koprivec
University of Ljubljana, SI
- Sivaramakrishnan Krishnamoorthy Chandrasekaran
Indian Institute of Techology, IN
- Daan Leijen
Microsoft Research – Redmond, US
- Sam Lindley
University of Edinburgh, GB
- Conor McBride
University of Strathclyde – Glasgow, GB
- Daniel Patterson
Northeastern University – Boston, US
- Maciej Piróg
University of Wroclaw, PL
- Gordon Plotkin
Google – Mountain View, US
- Matija Pretnar
University of Ljubljana, SI
- Aseem Rastogi
Microsoft Research India – Bangalore, IN
- Andreas Rossberg
Dfinity – Zürich, CH
- Philipp Schuster
Universität Tübingen, DE
- Taro Sekiyama
National Institute of Informatics – Tokyo, JP
- Antal Spector-Zabusky
Jane Street – London, GB
- Nikhil Swamy
Microsoft Research – Redmond, US
- Wouter Swierstra
Utrecht University, NL
- Luke Wagner
Fastly – San Francisco, US
- Leo White
Jane Street – London, GB
- Nicolas Wu
Imperial College London, GB



Report from Dagstuhl Seminar 21293

Parameterized Complexity in Graph Drawing

Edited by

Robert Ganian[1], Fabrizio Montecchiani[2], Martin Nöllenburg[3], and Meirav Zehavi[4]

1 **TU Wien, AT,** rganian@gmail.com
2 **University of Perugia, IT,** fabrizio.montecchiani@unipg.it
3 **TU Wien, AT,** noellenburg@ac.tuwien.ac.at
4 **Ben-Gurion University, IL,** zehavimeirav@gmail.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 21293 "Parameterized Complexity in Graph Drawing". The seminar was held mostly in-person from July 18 to July 23, 2021. It brought together 28 researchers from the Graph Drawing and the Parameterized Complexity research communities with the aim to discuss and explore open research questions on the interface between the two fields. The report collects the abstracts of talks and open problems presented in the seminar, as well as brief progress reports from the working groups.

Seminar July 18–23, 2021 – http://www.dagstuhl.de/21293
2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Graph algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms
Keywords and phrases exact computation, graph algorithms, graph drawing, parameterized complexity
Digital Object Identifier 10.4230/DagRep.11.6.82
Edited in cooperation with Jules Wulms

1 Executive Summary

Robert Ganian (TU Wien, AT)
Fabrizio Montecchiani (University of Perugia, IT)
Martin Nöllenburg (TU Wien, AT)
Meirav Zehavi (Ben-Gurion University, IL)

Graph Drawing. Graph-based models are pervasive in many fields of science and technology. Very often scientists and users analyze these models and communicate their findings by means of graphical representations. This motivated the birth and evolution of *graph drawing*, a self-standing discipline that has evolved tremendously over the past 50 years. Today graph drawing is a mature area of computer science [5, 13, 17, 18] with its own annual conference, the International Symposium on Graph Drawing and Network Visualization (GD)[1]. The focus of the research area today is on combinatorial and algorithmic aspects of drawing graphs as well as on the design of network visualization systems and interfaces. Graph

[1] see www.graphdrawing.org

drawing is motivated by applications where it is crucial to visually analyze and interact with relational datasets. Examples of such application areas include data science, social sciences, web computing, information systems, biology, geography, business intelligence, information security and software engineering.

Roughly speaking, graph drawing deals with the construction and analysis of geometric representations of graphs and networks subject to specific layout conventions, such as different notions of planarity or more general crossing constraints, grid layouts, orthogonal drawings etc. Many classic graph drawing problems are NP-hard and thus a variety of theoretical and practical algorithmic techniques for dealing with hard problems are required in graph drawing.

Parameterized Complexity. Numerous computational problems of wide interest are known to be NP-hard in general. Yet, it is often possible to utilize the structure implicitly underlying many real-world instances to find exact solutions efficiently. There is long-standing systematic research of tractability results for various problems on specific classes of instances, and research in this direction constitutes one of the fundamental areas of computer science. However, in many real-world situations it is not possible to define a clear-cut class of instances that we wish to solve; instead of being black and white (belonging to a specific class or not), instances often come in various shades of grey (having certain degrees of internal structure).

The relatively young *parameterized complexity* paradigm [6, 4, 8, 16] offers the perfect tools to deal with this situation. In the parameterized setting, we associate each instance with a numerical *parameter*, which captures how "structured" the instance is. This then allows the development of algorithms whose performance strongly depends on the parameter – instead of the classical setting, where we often associate tractability with polynomial running times and intractability with superpolynomial ones, parameterized algorithms naturally "scale" with the amount of structure contained in the instance. The central notion of tractability in the parameterized setting is *fixed-parameter tractable* (FPT in short), which means that the given problem can be solved by an algorithm with runtime of the form $f(k) \cdot n^{\mathcal{O}(1)}$ (where f is an arbitrary computable function, k is the value of the parameter, and n is the input size). Aside from fixed-parameter tractability, the parameterized complexity landscape consists of a variety of companion notions such as XP-*tractability*, *kernelization* and W-*hardness*.

Parameterized Complexity in Graph Drawing. Research at the intersection of graph drawing and parameterized complexity (and parameterized algorithms in particular) is in its infancy. Most of the early efforts have been directed at variants of the classic Crossing Minimization problem, introduced by Turán in 1940 [19], parameterized by the number of crossings. Here, the objective is to draw a given graph in the plane so as to induce minimum number of crossings. Already in 2001, it was shown to be FPT [9]. A few subsequent works followed [14, 11], including the best paper of GD 2019 [12], but also concerning restricted layouts such as two-layered embeddings [7] and two-sided circular graph layouts [15]. On a related note, given a graph drawn in the plane, some preliminary works considered the detection of a subgraph having a particular structure with minimum number of crossings [1, 10]. Recently, parameterized analysis of specific embeddings such as book embeddings [3, 2], was also brought into life. Overall, the intersection of graph drawing and parameterized complexity still remains mostly unexplored, yet we see many interesting challenges and opportunities for taking a parameterized perspective on graph drawing problems and investigating the applicability of advanced parameterized techniques.

Seminar Goals

The main goal of the seminar was to chart new paths towards research combining the latest findings and techniques in parameterized complexity and graph drawing. In particular, the seminar focused on several prominent topics in graph drawing as well as state-of-the-art tools in parameterized complexity. The discussions addressed both concrete open problems as well as general directions for future research. An integral part of these discussions was the identification and formulation of major challenges as well as novel parameterizations of graph drawing problems relevant to parameterized analysis. The discussions also addressed the applicability of classic as well as cutting-edge tools in parameterized complexity to graph drawing.

In view of the above, it is safe to say that the selection of suitable problems to target was of great importance for the success of the seminar. Our main aim was to offer the participants the opportunity to propose problems to work on, and so the final selection of problems targeted by working groups was carried out during the seminar itself. That being said, we have also prepared a list of candidate problems that we believe would be prime candidates for further investigation through the lens of parameterized complexity.

Seminar Program

1. On the first day of the seminar we enjoyed short introductions of all participants, and four invited overview lectures on different research domains within Graph Drawing. The topics and speakers were chosen as to create a joint understanding of the state of the art of problems in Graph Drawing suitable for parameterized anaylsis. Thekla Hamm presented the topic of graph drawing extension problems, Petr Hliněný presented the topic of planar insertion problems, Michael Kaufmann presented the topic of graph drawing beyond planarity and parametrized complexity, and Ignaz Rutter presented the topic of constrained embedding problems. More information on each lecture can be found in Section 3. Overall, this day prepared the ground for the open problem session on the second day.
2. The open problem session took place in the morning of the second day of the seminar. In this session, we collected a list of open research problems that were contributed by the seminar participants. In a preference voting we determined the five topics that raised the most interest among the participants and formed small working groups around them. Each group contained experts in both Graph Drawing and Parameterized Complexity. During the following days the groups worked by themselves, except for a few plenary reporting sessions, formalizing and solving their respective challenges. Below is a list of the working group topics; more detailed group reports are found in Section 5.
   a. **Upward/level planarity:** This group studied two previously established restrictions of drawing planar graphs: vertices are either assigned a "horizontal level" that they must be placed on, or there are directed arcs and the drawing must have all edges facing upwards. The group aimed at the development of new parameterized algorithms for both of these NP-hard problems.
   b. **Two-page embeddings of upward planar graphs:** The group studied the complexity of recognizing whether *st*-planar graphs admit an upward two-page book embedding.
   c. **Orthogonal drawings:** The group focused on the Compaction problem (computing a minimum-area drawing for an orthogonal graph), parameterized primarily by the number of kitty corners, that is, pairs of reflex vertices that point to each other.

   d. **Almost Separated Fixed Order Stack Layouts:** In a fixed order stack layout the vertices of a graph are given with a fixed order and one has to assign the edges to pages so that no two edges on any page cross. This group studied a variant of this well-known NP-complete problem, where the graph is bipartite and the vertices form k consecutive blocks from either part.
   e. **Graph product structure theorem:** The group considered strong products of graphs that yield supergraphs of k-planar graphs, i.e. of graphs that admit a drawing in the plane in which each edge is crossed at most k times. The objective is to exploit these products to derive new upper bounds on the queue number of k-planar graphs.
   f. **Decision trees:** Decision Trees are well known tools used to describe, classify, and generalize data. Besides their simplicity, decision trees are particularly attractive for providing interpretable models of the underlying data. The group studied the complexity of learning decision trees of minimum size under several different parameterizations.
3. After the open problem session, Robert Ganian gave a tutorial in the second day of the seminar that showcased how some of the tools in Parameterized Complexity can be applied to difficult problems, with a special focus on problems that are relevant to graph drawing. The tutorial was prepared in a way so as to make it accessible to the graph drawing community, acting as catalysis for progress on the five selected topics.
4. In the rest of the second day and the other days of the seminar, we had a flexible working schedule with a short plenary session every morning to accommodate group reports and impromptu presentations by participants.

Future Plans

The seminar was designed to foster new research collaborations between researchers in the graph drawing and parameterized complexity communities, whose paths rarely cross in the traditional conferences. These collaborations are very likely to result in new breakthroughs and results, and we expect that the seminar will lead to tangible progress in our understanding of problems of interest. In this sense, the primary outcome from the seminar will be research papers published at the core conferences and journals for the graph drawing and parameterized complexity communities, such as:

- The International Symposium on Computational Geometry (**SoCG**),
- The International Symposium on Graph Drawing and Network Visualization (**GD**),
- The ACM-SIAM Symposium on Discrete Algorithms (**SODA**), and
- The International Symposium on Theoretical Aspects of Computer Science (**STACS**).

In the mid- and long-term horizon, the seminar will also help build a bridge between the two communities and identify other interesting graph drawing problems which would benefit from a rigorous investigation using tools from parameterized complexity. It can also lead to the development of new parameterized tools and techniques that are designed to deal with the specific obstacles that arise when trying to apply parameterized approaches in the graph drawing setting. Last but not least, the seminar will raise the awareness for the typical research problems and the latest techniques in each others community and thus enrich the knowledge and toolbox of individual participants.

Dagstuhl seminar in 2022/2023 on Graph Drawing in Parameterized Complexity. This Dagstuhl seminar has revealed, for the first time in a systematic way, the astounding wealth of problems in Graph Drawing that are naturally multivariate and hence suitable

for parameterized analysis; thus a follow-up Dagstuhl seminar will be proposed to further discuss and deepen our understanding of this topic whose full potential is yet to be unlocked, once again bringing together researchers in Graph Drawing and Parameterized Complexity.

Evaluation

According to the Dagstuhl survey conducted after the seminar, as well as informal feedback to the organizers, the seminar was highly appreciated. Particularly the small group size, group composition, and the seminar structure focusing on hands-on working groups was very well received. The seminar's goals to identify new research directions and initiate collaborations at the intersection of the two different fields of Graph Drawing and Parameterized Complexity was very successful (also in comparison to other Dagstuhl seminars). Indeed, the participants rated the seminar highly for the mixture of these two fields and its productive interdisciplinary atmosphere, yielding new research perspectives, which have also resulted in new collaborations, joint projects and publications. We are looking forward to seeing the first scientific outcomes of the seminar in the near future and to continuing the efforts to support the growth of interest in parameterized analysis of problems in Graph Drawing.

The seminar had more participants from the Graph Drawing community than from the Parameterized Complexity community due to critical uncertainties caused by the COVID-19 pandemic. We hope that the current trend of improvement in the situation will help in composing a more balanced list of participants in future seminars on this topic.

Acknowledgments

Schloss Dagstuhl was the perfect place for hosting a seminar like this. The unique scientific atmosphere and the historic building provided not only all the room we needed for our program and the working groups, but also plenty of opportunities for continued discussions and socializing outside the official program, especially in these difficult times during the COVID-19 pandemic with all participants being eager to meet and do research together in real life. On behalf of all participants, the organizers want to express their deep gratitude to the entire Dagstuhl staff for their outstanding support and service accompanying this seminar. We further thank Jules Wulms for helping us collect the contributions and prepare this report.

References

1 Akanksha Agrawal, Grzegorz Guspiel, Jayakrishnan Madathil, Saket Saurabh, and Meirav Zehavi. Connecting the Dots (with Minimum Crossings). In Gill Barequet and Yusu Wang, editors, *Symposium on Computational Geometry (SoCG 2019)*, volume 129 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:17, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

2 Michael J. Bannister, David Eppstein, and Joseph A. Simons. Fixed parameter tractability of crossing minimization of almost-trees. In *Graph Drawing (GD 2013)*, volume 8242 of *Lecture Notes in Computer Science*, pages 340–351. Springer, 2013.

3 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for book embedding problems. In *Graph Drawing and Network Visualization (GD 2019)*, Lecture Notes in Computer Science. Springer, 2019. To appear.

4 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

5 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

6 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer Verlag, 2013.

7 Vida Dujmovic, Michael R. Fellows, Matthew Kitching, Giuseppe Liotta, Catherine McCartin, Naomi Nishimura, Prabhakar Ragde, Frances A. Rosamond, Sue Whitesides, and David R. Wood. On the parameterized complexity of layered graph drawing. *Algorithmica*, 52(2):267–292, 2008.

8 F.V. Fomin, D. Lokshtanov, S. Saurabh, and M. Zehavi. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2018.

9 Martin Grohe. Computing crossing numbers in quadratic time. *J. Comput. Syst. Sci.*, 68(2):285–302, 2004.

10 Magnús M. Halldórsson, Christian Knauer, Andreas Spillner, and Takeshi Tokuyama. Fixed-parameter tractability for non-crossing spanning trees. In *Algorithms and Data Structures (WADS 2007)*, volume 4619 of *Lecture Notes in Computer Science*, pages 410–421. Springer, 2007.

11 Petr Hlinený and Marek Dernár. Crossing number is hard for kernelization. In *Symposium on Computational Geometry (SoCG 2016)*, volume 51 of *LIPIcs*, pages 42:1–42:10. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.

12 Petr Hlinený and Abhisekh Sankaran. Exact crossing number parameterized by vertex cover. In *Graph Drawing and Network Visualization (GD 2019)*, Lecture Notes in Computer Science. Springer, 2019. To appear.

13 Michael Jünger and Petra Mutzel, editors. *Graph Drawing Software*. Springer, 2004.

14 Ken-ichi Kawarabayashi and Bruce A. Reed. Computing crossing number in linear time. In *Symposium on Theory of Computing (STOC 2007)*, pages 382–390. ACM, 2007.

15 Fabian Klute and Martin Nöllenburg. Minimizing crossings in constrained two-sided circular graph layouts. In *Symposium on Computational Geometry (SoCG 2018)*, volume 99 of *LIPIcs*, pages 53:1–53:14. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2018.

16 Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications. OUP Oxford, 2006.

17 Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing*, volume 12 of *Lecture Notes Series on Computing*. World Scientific, 2004.

18 Roberto Tamassia, editor. *Handbook on Graph Drawing and Visualization*. Chapman and Hall/CRC, 2013.

19 Paul Turán. A note of welcome. *Journal of Graph Theory*, 1(1):7–9, 1977.

2 Table of Contents

Executive Summary
Robert Ganian, Fabrizio Montecchiani, Martin Nöllenburg, and Meirav Zehavi . . . 82

Overview of Talks

Graph Drawing Extension Problems
Thekla Hamm . . . 90

Planar insertion problems
Petr Hlinený . . . 90

Constrained Embedding Problems
Ignaz Rutter . . . 91

Graph Drawing beyond planarity and Parametrized Complexity
Michael Kaufmann . . . 91

Open problems

Bundled Crossings
Steven Chaplick . . . 92

Algorithmic and Combinatorial Applications of the Product Structure Theorems
Giordano Da Lozzo . . . 93

Three open problems about orthogonal and upward drawings
Emilio Di Giacomo, Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani . 94

Parameterized Complexity of Computing Stack and Queue Numbers
Robert Ganian . . . 97

Almost Separated Fixed Order Stack Layouts
Martin Gronemann . . . 97

Embedding Upward Planar Graphs in two Pages
Martin Gronemann . . . 98

Fine-grained complexity of the crossing number of almost planar graphs
Petr Hlinený . . . 99

Labeling Curve Arrangements
Maarten Löffler . . . 100

Bend Minimization in Orthogonal Drawings
Ignaz Rutter and Meirav Zehavi . . . 103

Is Extending Partial Drawings of Level Planar Graphs FPT?
Ignaz Rutter . . . 104

The Parameterized Complexity of Learning Small Decision Trees in Low-Dimensional Space
Manuel Sorge . . . 105

Two open problems on drawings of complete graphs
Birgit Vogtenhuber . . . 106

Working groups

Progress on Upward Planarity Testing
Robert Ganian, Steven Chaplick, Emilio Di Giacomo, Fabrizio Frati, Chrysanthi Raftopoulou, and Kirill Simonov 108

Progress on Embedding Upward Planar Graphs in two Pages
Michael A. Bekos, Giordano Da Lozzo, Fabrizio Frati, Martin Gronemann, and Chrysanthi Raftopoulou 111

Progress on A Parameterized Approach to Orthogonal Compaction
Philipp Kindermann, Walter Didimo, Siddharth Gupta, Giuseppe Liotta, Alexander Wolff, and Meirav Zehavi 113

Progress on Almost Separated Fixed Order Stack Layouts
Johannes Zink, Martin Gronemann, Thekla Hamm, Boris Klemz, Martin Nöllenburg, and Birgit Vogtenhuber 116

Progress on Applications of the Product Structure Theorems
Giordano Da Lozzo, Michael A. Bekos, Petr Hlinený, and Michael Kaufmann . . . 118

Progress on the Parameterized Complexity of Small Decision Tree Learning
Stephen G. Kobourov, Maarten Löffler, Fabrizio Montecchiani, Raimund Seidel, Ignaz Rutter, Manuel Sorge, and Jules Wulms 120

Participants 123

Remote Participants 123

3 Overview of Talks

3.1 Graph Drawing Extension Problems

Thekla Hamm (TU Wien, AT)

Joint work of Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, Martin Nöllenburg, Irene Parada, Birgit Vogtenhuber
Main reference Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, Martin Nöllenburg: "Extending Partial 1-Planar Drawings", in ICALP 2020, LIPIcs, Vol. 168, pp. 43:1–43:19, 2020.
URL https://doi.org/10.4230/LIPIcs.ICALP.2020.43
Main reference Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, Birgit Vogtenhuber: "Crossing-Optimal Extension of Simple Drawings", in ICALP 2021, LIPIcs, Vol. 198, pp. 72:1–72:17, 2021.
URL https://doi.org/10.4230/LIPIcs.ICALP.2021.72

The investigation of problems that ask for drawings of graphs with desirable properties (most commonly restricing crossings of edge drawings) while also fixing the drawing of a given subgraph is an increasingly popular direction in the field of graph drawing. These problems are also called *drawing extension problems*.

While the planar drawing extension problem can be solved in polynomial time, for many other important drawing styles, such as 1-planar, k-planar, IC-planar, straight-line planar and level planar, drawing extension is NP-hard. In this talk we explore the possibility of circumventing these hardness results when a large part of the graph is predrawn using the framework of parameterised complexity theory. In particular we review a general technique which can be used to show FPT results for a number of beyond-planar drawing styles and outline a variety of related open questions.

3.2 Planar insertion problems

Petr Hlinený (Masaryk University – Brno, CZ)

Joint work of Petr Hlinený, Markus Chimani, Gelasio Salazar
Main reference Markus Chimani, Petr Hlinený: "A tighter insertion-based approximation of the crossing number", J. Comb. Optim., Vol. 33(4), pp. 1183–1225, 2017.
URL https://doi.org/10.1007/s10878-016-0030-z

A *planar insertion* problem is defined as follows: Given graphs G (planar) and H, the task of insertion of H into G is to find a crossing-minimal drawing of $G \cup H$ such that G itself is planar in the drawing. This problem is intermediate between ordinary crossing minimization and drawing extension problems, in the following sense. While in ordinary crossing minimization any drawing of the target graph is allowed, in planar insertion certain part of it (here G) must be planarly drawn. On the other hand, unlike in drawing extension problems, the planar part G may choose between its planar embeddings.

We survey past achievements in solving planar insertion problem variants. Firstly, we outline the linear-time algorithm for a single edge insertion by Gutwenger, Mutzel and Weiskircher from 2005, and show how this approximates the crossing number of a planar graph plus one edge (up to a multiplicative factor depending on the maximum degree). Note that determining the exact crossing number of a planar graph plus one edge is NP-hard by a result of Cabello and Mohar from 2011.

We then show how the multiple edge insertion problem can be in polynomial time approximated up to an additive error depending on the number of inserted edges and the maximum degree. Again, the general question is NP-hard. From another perspective, we show that the multiple edge insertion problem can be solved exactly in FPT time when the parameter is the number of inserted edges.

3.3 Constrained Embedding Problems

Ignaz Rutter (Universität Passau, DE)

Determining a planar embedding of a graph is a classical problem. In many applications, one is interested in finding a planar embedding that satisfies additional constraints. In this talk, we survey several techniques and demonstrate their application on a number such problems. For local constraints that mostly concern rotations, i.e., the circular orders of edges around vertices, PQ-trees and their circular variants known as PC-trees serve as a powerful tool. If a more global view of the possible embeddings is necessary, often the SPQR-tree is useful, as it breaks up the complicated choice of a planar embedding into several simple and independent choices. Lastly, the ability to synchronize the rotations of different vertices is a powerful method, whose solution requires a combination of both of the above techniques.

3.4 Graph Drawing beyond planarity and Parametrized Complexity

Michael Kaufmann (Universität Tübingen, DE)

In this talk, we gave an overview on different aspects on graph drawing beyond planarity, i.e. drawings where some crossing configurations for the edges are forbidden. Notable criteria are density of the graphs, recognition, class hierarchies, constraints,

We discussed several results from the literature related to aspects of parametrized complexity, in particular kernel-based methods, separators, path – and treewidth- related questions. We reviewed the most important results from the seminal paper on the parametrized complexity of 1-planarity by Bannister, Cabello and Eppstein [1]. Furthermore we highlighted some of the methods developed on track-layout of fan-planar graphs by Biedl et al. [3].

We extracted and discuss possible open directions related to k-planarity, fan-planarity and other classes of beyond-planar graphs that could be attacked during and after the workshop.

A notable paper which we did not included is the work by Bhore et al. [2], which extends the recent research direction on linear layouts towards parametrized complexity.

References

1 Michael J. Bannister, Sergio Cabello, and David Eppstein. Parameterized complexity of 1-planarity. In *Algorithms and Data Structures – 13th International Symposium, WADS 2013, London, ON, Canada, August 12-14, 2013. Proceedings*, volume 8037 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2013.

2 Sujoy Bhore, Robert Ganian, Fabrizio Montecchiani, and Martin Nöllenburg. Parameterized algorithms for book embedding problems. *J. Graph Algorithms Appl.*, 24(4):603–620, 2020.

3 Therese C. Biedl, Steven Chaplick, Michael Kaufmann, Fabrizio Montecchiani, Martin Nöllenburg, and Chrysanthi N. Raftopoulou. Layered fan-planar graph drawings. In *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 14:1–14:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

4 Open problems

4.1 Bundled Crossings

Steven Chaplick (Maastricht University, NL)

An effective way to reduce clutter in a graph drawing that has (many) crossings is to group edges that travel in parallel into *bundles*. This concept was introduced by Holten [4].

Each edge can participate in many such bundles. Any crossing in this bundled drawing occurs between two bundles (possibly one such bundle will consist of a single edge) and these crossings are referred to as*bundled crossing*. We consider the problem of bundled crossing minimization: A graph is given and the goal is to find a bundled drawing with at most k bundled crossings. This problem is known to be NP-complete when in both the case when a simple drawing is required and when the drawing is allowed to be non-simple. The latter (non-simple) case turns out to be equivalent to the computing the graph genus [1], and as such has a long history including efficient FPT algorithms, see, e.g., whereas for the simple case it is open whether the problem is FPT [5]. In the case of simple drawings the problem is known to be FPT when one further insists on a *circular layout* where vertices are placed in convex position and all edges are required to be drawn within the convex hull of the vertices [2].

Finally, we note that even when given a graph drawn in the plane (with crossings) and parameter k, and one desires to bundle this drawing to have at most k crossings, the problem is also NP-complete [3]. In other words, trying to find an optimal bundling of a given drawing is also an interesting problem where, as far as we are aware, the question of fixed-parameter tractability remains open as well.

References

1 Md. Jawaherul Alam, Martin Fink, and Sergey Pupyrev. The bundled crossing number. In Yifan Hu and Martin Nöllenburg, editors, *GD*, volume 9801 of *LNCS*, pages 399–412. Springer, 2016.
2 Steven Chaplick, Thomas C. van Dijk, Myroslav Kryven, Ji-won Park, Alexander Ravsky, and Alexander Wolff. Bundled crossings revisited. *J. Graph Algorithms Appl.*, 24(4):621–655, 2020.
3 Martin Fink, John Hershberger, Subhash Suri, and Kevin Verbeek. Bundled crossings in embedded graphs. In Evangelos Kranakis, Gonzalo Navarro, and Edgar Chávez, editors, *LATIN*, volume 9644 of *LNCS*, pages 454–468. Springer, 2016.
4 Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. Vis. Comput. Graphics*, 12(5):741–748, 2006.
5 Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *FOCS*, pages 771–780. IEEE, 2008.

4.2 Algorithmic and Combinatorial Applications of the Product Structure Theorems

Giordano Da Lozzo (University of Rome III, IT)

Main reference Vida Dujmovic, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, David R. Wood: "Planar Graphs Have Bounded Queue-Number", J. ACM, Vol. 67(4), pp. 22:1–22:38, 2020.
URL https://dl.acm.org/doi/10.1145/3385731

Consider two graphs A and B. The *strong product* of A and B, denoted by $A \boxtimes B$, is the graph such that: (i) $V(A \boxtimes B) = V(A) \times V(B)$ and (ii) there exists an edge between the vertices $(a_1, b_1), (a_2, b_2) \in V(A \boxtimes B)$ if and only if one of the following occurs: (a) $a_1 = a_2$ and $b_1 b_2 \in E(B)$, (b) $b_1 = b_2$ and $a_1 a_2 \in E(A)$, or $a_1 a_2 \in E(A)$ and $b_1 b_2 \in E(B)$. In a breakthrough result, Dujmović et al. [1] have shown that every planar graph is a subgraph of the strong product of a graph of treewidth 8 and a path. In the same paper, such a result has also been generalized to graphs of bounded Euler genus and to proper minor-closed classes of graphs. In a recent preprint [4], Dujmović, Morin, and Wood have extended this result to some non-minor-closed graph classes. In particular, they proved that that every k-planar graph is a subgraph of the strong product of a graph of treewidth $O(k^5)$ and a path. These results, commonly referred to as the Product Structure Theorems (PSTs), have proved essential to solve several *combinatorial* long-standing open questions for the above mentioned graph classes. For instance, the PSTs allowed to prove that planar graphs have bounded queue number and bounded non-repetitive chromatic number [1], to improve the best known bounds for p-centered colorings of planar graphs and graphs excluding any fixed graph as a subdivision [2], to find shorter adjacency labelings of planar graphs [5], and to find asymptotically optimal adjacency labelings of planar graphs [3].

First, we suggest to keep exploring the above line of research by studying the following problem.

OP1: Can the PST be improved for k-planar graphs, with $k \in \{1, 2\}$?

Furthermore, it is interesting to consider a new line of research aimed at investigating the *algorithmic applications* of the PSTs. We believe that these theorems could support new results in fixed-parameter tractability, approximations, and bidimensionality theory. In particular, we propose the following problem.

OP2: Are there notable applications of the PST for topological k-planar graphs to obtain FPT algorithms parameterized by k?

References

1 V. Dujmović, G. Joret, P. Micek, P. Morin, T. Ueckerdt, and D. Wood. Planar Graphs Have Bounded Queue-Number. *J. ACM*,67(4): 22:1-22:38 (2020).
2 M. Debski, S. Felsner, P. Micek, and F. Schröder. Improved bounds for centered colorings. *SODA 2020*: 2212-2226 (2020).
3 V. Dujmović, L. Esperet, C. Gavoille, G. Joret, P. Micek, and P. Morin. Adjacency Labelling for Planar Graphs (and Beyond). *FOCS 2020*: 577-588 (2020).
4 V. Dujmović, P. Morin, and D. Wood. Graph product structure for non-minor-closed classes. *CoRR* abs/1907.05168 (2020).
5 M. Bonamy, C. Gavoille, and M. Pilipczuk. Shorter Labeling Schemes for Planar Graphs. *SODA 2020*: 446-462 (2020).

4.3 Three open problems about orthogonal and upward drawings

Emilio Di Giacomo (University of Perugia, IT), Walter Didimo (University of Perugia, IT), Giuseppe Liotta (University of Perugia, IT), and Fabrizio Montecchiani (University of Perugia, IT)

Problem 1: Rectlinear planarity testing

A graph is *planar* if it admits a drawing in the plane such that edges intersect only at common endpoints. Testing graph planarity is a fundamental problem in graph algorithms that have been studied in several variants and restrictions, such as upward planarity, clustered planarity, and constrained planarity. A classical planarity variant is the *rectilinear planarity*, which asks whether a planar graph with maximum vertex degree four admits a *rectilinear drawing*, i.e., a planar drawing where each edge is either a horizontal or a vertical segment.

Rectilinear drawings are a special case of *orthogonal drawings*, where edges are represented as chains of horizontal and vertical segments. Orthogonal drawings are among the most investigated research subjects in graph drawing (see, e.g., [6, 12]). A natural measure of the complexity of an orthogonal drawing is the number of bends along the edges, which should be minimized. In this sense, a rectilinear drawing is optimal, since it has no bends.

Garg and Tamassia [13] proved that rectilinear planarity testing is NP-complete. In fact, it is even NP-hard to approximate the minimum number of bends in an orthogonal drawing with an $O(n^{1-\varepsilon})$ error for any $\varepsilon > 0$ [13]. On the other hand if the input graph is *plane*, i.e., it has a fixed embedding in the plane, Tamassia [19] showed that rectilinear planarity testing can be decided in polynomial time. When a planar embedding is not given as part of the input, polynomial-time algorithms exist for some restricted cases, such as subcubic planar graphs and series-parallel graphs [5, 7, 11, 18, 20]).

Given the hardness results for rectilinear planarity testing, it is natural to study its parameterized complexity. Few results are known in this direction: Didimo and Liotta [10] described an algorithm for biconnected planar graphs that runs in $O(6^r n^4 \log n)$ time, where r is the number of degree-4 vertices. More recently Di Giacomo, Liotta, and Montecchiani [8] proved that the problem belongs to the XP class when parameterized by the treewidth and to the FPT class when parameterized by the treewidth plus the number of vertices of degree at most 2.

In the light of these last results it is natural to ask if the problem is in FPT when parametreized by only one of the two parameters.

▶ Problem 1. Is rectilinear planarity testing in FPT when parameterized by the treewidth? Is rectilinear planarity testing in FPT when parameterized by the number of degree-2 vertices?

Problem 2: Orthognal compaction

As mentioned above, if the planar embedding is fixed, an orthogonal drawing with the minimum number of bends can be computed in polynomial time. Thus, one of the most used algorithmic frameworks to compute orthogonal drawings is the one proposed by Tamassia [19], usually referred to as the *Topology-Shape-Metrics approach*. This approach works in three steps. The first step, called *Planarization*, fixes the *topology* of the input graph G, that is, it computes a planar embedding of G; if G is not planar a planarization of G is constructed, i.e., a planar graph obtained by replacing crossings with dummy vertices; the optimization goal of this step is to reduce the number of crossings and therefore of dummy vertices. The second

step, called *Orthogonalization*, decides the *shape* of the drawing, that is, it computes what is called an *orthogonal representation* of G. An orthogonal representation is a description of the shape of an orthogonal drawing in terms of the angles at the vertices and the number of bends along the edges. In this step the optimization goal is to minimize the number of bends, which, as said above, can be done in polynomial time once the planar embedding is fixed. In the third step the actual coordinates of the vertices and bends are decided thus fixing the *metrics* of the drawing. This step is called *Compaction* step because the coordinates are assigned with the goal of minimizing the area of the drawing (or the total length of the edges).

The compaction step hence, solves the following problem, called the *othogonal compaction* problem: Given an orthogonal representation, compute vertex and bend coordinates in such a way that the area is minimized. This problem is known to be NP-complete [17] but it is polynomially-time solvable for *turn-regular* orthogonal representations [3]. An orthogonal representation is turn-regular if it does not contain any pairs of *kitty corners*. A pair of kitty corners is a pair of vertices u and v such that: (i) both u and v form a $\frac{3\pi}{2}$ angle inside a face f; and (ii) walking clockwise along the boundary of f from u (included) to v (excluded) or vice versa the number of encountered vertices that form an angle of $\frac{\pi}{2}$ minus the number of encountered vertices that form an angle of $\frac{3\pi}{2}$ is 2. The two mentioned results suggest the following problem.

▶ Problem 2. Is orthogonal compaction problem in FPT when parameterized by the number of kitty corners?

Problem 3: Upward planarity testing

Upward planarity is another variant of planarity that has been widely investigated in the literature. An *upward planar drawing* of a directed acyclic graph is a planar drawing such that all edges are drawn as curves monotonically increasing in the upward direction. Similar to the case of rectilinear and orthogonal planarity, the problem is polinomially time solvable if a planar embedding of the input graph is fixed [1] and it is NP-complete if the planar embedding can be changed [13]. In the variable embedding setting polynomial-time algorithms exists for special cases, such as outerplanar DAGs [16] or series-parallel DAGs [9]. In particular, the problem can be solved in polynomial tile when the input DAG has a single source, i.e., a single vertex without incoming edges [2, 15]. These results naturally motivates the following problem.

▶ Problem 3. Is upward planarity testing in FPT when parameterized by the number of sources?

It is worth mentioning that FPT algorithms exists for the upward planarity testing when parameterized by the number of cut-vertices and the number of triconnected components [4], only by the number of triconnected components [14], by the difference between the number of edges and the number of vertices [14] and by the number of triconncted components and the diameter of any split component [9].

References

1 P. Bertolazzi, G. D. Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994. doi:10.1007/BF01188716.

2 P. Bertolazzi, G. D. Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM J. Comput.*, 27(1):132–169, 1998. doi:10.1137/S0097539794279626.

3 S. S. Bridgeman, G. D. Battista, W. Didimo, G. Liotta, R. Tamassia, and L. Vismara. Turn-regularity and optimal area drawings of orthogonal representations. *Comput. Geom.*, 16(1):53–93, 2000. doi:10.1016/S0925-7721(99)00054-1.

4 H. Y. Chan. A parameterized algorithm for upward planarity testing. In S. Albers and T. Radzik, editors, *Algorithms – ESA 2004, 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3221 of *Lecture Notes in Computer Science*, pages 157–168. Springer, 2004. doi:10.1007/978-3-540-30140-0_16.

5 Y. Chang and H. Yen. On bend-minimized orthogonal drawings of planar 3-graphs. In *SOCG 2017*, volume 77 of *LIPIcs*, pages 29:1–29:15. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.

6 G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice-Hall, 1999.

7 G. Di Battista, G. Liotta, and F. Vargiu. Spirality and optimal orthogonal drawings. *SIAM J. Comput.*, 27(6):1764–1811, 1998.

8 E. Di Giacomo, G. Liotta, and F. Montecchiani. Sketched representations and orthogonal planarity of bounded treewidth graphs. In D. Archambault and C. D. Tóth, editors, *Graph Drawing and Network Visualization – 27th International Symposium, GD 2019, Prague, Czech Republic, September 17-20, 2019, Proceedings*, volume 11904 of *Lecture Notes in Computer Science*, pages 379–392. Springer, 2019. doi:10.1007/978-3-030-35802-0_29.

9 W. Didimo, F. Giordano, and G. Liotta. Upward spirality and upward planarity testing. *SIAM J. Discret. Math.*, 23(4):1842–1899, 2009. doi:10.1137/070696854.

10 W. Didimo and G. Liotta. Computing orthogonal drawings in a variable embedding setting. In *ISAAC 1998*, volume 1533 of *LNCS*, pages 79–88. Springer, 1998.

11 W. Didimo, G. Liotta, and M. Patrignani. Bend-minimum orthogonal drawings in quadratic time. In *GD 2018*, volume 11282 of *LNCS*, pages 481–494. Springer, 2018.

12 C. A. Duncan and M. T. Goodrich. Planar orthogonal and polyline drawing algorithms. In *Handbook of Graph Drawing and Visualization*, pages 223–246. Chapman and Hall/CRC, 2013.

13 A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001.

14 P. Healy and K. Lynch. Two fixed-parameter tractable algorithms for testing upward planarity. *Int. J. Found. Comput. Sci.*, 17(5):1095–1114, 2006. doi:10.1142/S0129054106004285.

15 M. D. Hutton and A. Lubiw. Upward planning of single-source acyclic digraphs. *SIAM J. Comput.*, 25(2):291–311, 1996. doi:10.1137/S0097539792235906.

16 A. Papakostas. Upward planarity testing of outerplanar dags. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing, DIMACS International Workshop, GD '94, Princeton, New Jersey, USA, October 10-12, 1994, Proceedings*, volume 894 of *Lecture Notes in Computer Science*, pages 298–306. Springer, 1994. doi:10.1007/3-540-58950-3_385.

17 M. Patrignani. On the complexity of orthogonal compaction. *Comput. Geom.*, 19(1):47–67, 2001. doi:10.1016/S0925-7721(01)00010-4.

18 M. S. Rahman, N. Egi, and T. Nishizeki. No-bend orthogonal drawings of subdivisions of planar triconnected cubic graphs. *IEICE Transactions*, 88-D(1):23–30, 2005.

19 R. Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comp.*, 16(3):421–444, 1987.

20 X. Zhou and T. Nishizeki. Orthogonal drawings of series-parallel graphs with minimum bends. *SIAM J. Discrete Math.*, 22(4):1570–1604, 2008.

4.4 Parameterized Complexity of Computing Stack and Queue Numbers

Robert Ganian (TU Wien, AT)

The problems of computing a queue or stack layout with the minimum number of pages (the so-called Queue Number and Stack Number problems) are well-studied and known to be NP-complete, but we still do not understand the conditions under which these problems become tractable. In particular, while recent works have shown that both problems are fixed-parameter tractable when parameterized by the vertex cover number [1, 2], we do not know anything about their parameterized complexity when parameterized by clique-width, treewidth, pathwidth, treedepth, feedback vertex number, and even feedback edge number. In fact, we do not even know whether the problem is fixed-parameter tractable, W-hard, or paraNP-hard when parameterized by a parameter as simple as the edge deletion distance to a collection of paths.

References

1 Sujoy Bhore and Robert Ganian and Fabrizio Montecchiani and Martin Nöllenburg. *Parameterized Algorithms for Book Embedding Problems*. J. Graph Algorithms Appl., vol. 24, issue 4, 2020.
2 Sujoy Bhore and Robert Ganian and Fabrizio Montecchiani and Martin Nöllenburg. *Parameterized Algorithms for Queue Layouts*. Graph Drawing and Network Visualization – 28th International Symposium, GD 2020, Vancouver, BC, Canada, September 16-18, 2020, Revised Selected Papers.

4.5 Almost Separated Fixed Order Stack Layouts

Martin Gronemann (Universität Osnabrück, DE)

Book embeddings have a long history in graph theory. Today, book embeddings are often referred to as stack layouts. Formally, a stack layout consists of a linear ordering of the vertices σ drawn on a line and a partitioning of the edges into *pages* such that no two edges on the same page cross when drawn in the same half-plane defined by the line. Given a graph G, the minimum number of pages required in any stack layout of G is referred to as stack number $\mathrm{sn}(G)$. Determining the stack number of a graph is inherently difficult. While this problem is linear time solvable for $\mathrm{sn}(G) = 1$ by testing if the input graph is outerplanar, testing if two stacks are sufficient is already NP-complete [2].

Therefore, it makes sense to consider a more restricted variant of this problem by assuming that the vertex order σ is given as part of the input. Hence, it remains to assign the edges to pages by using as few pages as possible. This problem is sometimes referred to as the fixed-order book thickness problem. Unfortunately, also this problem is known to be NP-complete for four or more pages [1]. However, for some vertex orderings, the problem becomes easier. Consider a bipartite graph with partitions A and B. If in the vertex order A and B are *separated*, that is, all vertices of A precede those of B, the stack number equals the number of pairwise crossing edges. One may now generalize this concept of being separated by assuming that for a bipartite graph $G = (A \cup B, E)$ the vertices of A and B form k

consecutive blocks in the fixed vertex order. More, specifically, in σ there are exactly $\frac{k}{2}$ consecutive blocks containing vertices of A and $\frac{k}{2}$ consecutive blocks containing solely vertices of B. We refer to such a layout as *fixed k-separated layout*.

Open Problem. Is the FIXED-ORDER BOOK THICKNESS problem for fixed k-separated layouts fixed-parameter tractable in k?

References

1 Walter Unger. On the k-colouring of circle-graphs. In Robert Cori and Martin Wirsing, editors, *STACS 88*, pages 61–72, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.

2 Avi Wigderson. The complexity of the Hamiltonian circuit problem for maximal planar graphs. Technical Report TR-298, EECS Department, Princeton University, 1982. `arXiv: https://www.math.ias.edu/avi/node/820`.

4.6 Embedding Upward Planar Graphs in two Pages

Martin Gronemann (Universität Osnabrück, DE)

Book embeddings of graphs are a classic topic in graph theory and graph drawing [1, 2, 4, 3, 6, 7, 14, 15]. Formally, in a *book embedding*, the vertices of a given graph must be ordered along a line, called *spine*, and its edges must be drawn in different half-planes bounded by the spine, called *pages* of the book, such that no two edges of the same page cross. For a 2-page book embedding, where only two pages are available, one can use the two half-planes defined by the spine for drawing the edges of the graph [14]. As a consequence such a drawing is planar.

For directed graphs (digraphs), Heath, Pemmaraju, and Trenk introduced a variant of book embeddings, called *upward*, in which all the edges are oriented in the upward direction, i.e., such that for every directed edge uv of the given graph u precedes v along the spine [10]. Clearly, this immediately implies that the input graph is acyclic. In the case, in which only two pages are available, one obtains a special form of an upward planar drawing. An *upward planar* drawing of a directed acyclic graph is a planar drawing in which each edge uv is drawn as a y-monotone curve from u to v. A planar directed acyclic graph that admits such a drawing is called *upward planar*. However, deciding whether a planar directed acyclic graph is upward planar is known to be NP-complete [8]. An important family of graphs in this context are the *st-planar graphs*, i.e., planar directed acyclic graphs having only one source s and one sink t. It is known that all st-planar graphs are upward planar and that every upward planar graph is a subgraph of an st-planar graph. An interesting question which attracted attention in the literature is the 2-page embeddability of st-planar graphs [9, 10]. For specific families of st-planar graphs or graphs where certain conditions are met, the existence of an upward 2-page book embedding can be efficiently decided [5, 12, 11]. However, the general question remains unanswered [13].

Open Problem. What is the complexity of deciding whether a given embedded st-planar graph admits a (planar) upward 2-page book embedding?

References

1 M. A. Bekos, T. Bruckdorfer, M. Kaufmann, and C. N. Raftopoulou. The book thickness of 1-planar graphs is constant. *Algorithmica*, 79(2):444–465, 2017.

2 M. A. Bekos, G. Da Lozzo, S. Griesbach, M. Gronemann, F. Montecchiani, and C. N. Raftopoulou. Book embeddings of nonplanar graphs with small faces in few pages. In S. Cabello and D. Z. Chen, editors, *SoCG 2020*, volume 164 of *LIPIcs*, pages 16:1–16:17. Schloss Dagstuhl, 2020.

3 M. A. Bekos, M. Gronemann, and C. N. Raftopoulou. Two-page book embeddings of 4-planar graphs. *Algorithmica*, 75(1):158–185, 2016.

4 F. Bernhart and P. C. Kainen. The book thickness of a graph. *Journal of Combinatorial Theory, Series B*, 27(3):320 – 331, 1979.

5 C. Binucci, G. Da Lozzo, E. D. Giacomo, W. Didimo, T. Mchedlidze, and M. Patrignani. Upward book embeddings of st-graphs. In G. Barequet and Y. Wang, editors, *SoCG 2019*, volume 129 of *LIPIcs*, pages 13:1–13:22. Schloss Dagstuhl, 2019.

6 F. R. K. Chung, F. T. Leighton, and A. L. Rosenberg. Embedding graphs in books: A layout problem with applications to VLSI design. *SIAM Journal on Algebraic Discrete Methods*, 8(1):33–58, 1987.

7 H. Enomoto, T. Nakamigawa, and K. Ota. On the pagenumber of complete bipartite graphs. *Journal of Combinatorial Theory, Series B*, 71(1):111–120, 1997.

8 A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001.

9 L. S. Heath and S. V. Pemmaraju. Stack and queue layouts of posets. *SIAM Journal on Discrete Mathematics*, 10(4):599–625, 1997.

10 L. S. Heath, S. V. Pemmaraju, and A. N. Trenk. Stack and queue layouts of directed acyclic graphs: Part I. *SIAM Journal on Computing*, 28(4):1510–1539, 1999.

11 T. Mchedlidze and A. Symvonis. Crossing-free acyclic hamiltonian path completion for planar *st*-digraphs. In Y. Dong, D. Du, and O. H. Ibarra, editors, *ISAAC 2009*, volume 5878 of *LNCS*, pages 882–891. Springer, 2009.

12 T. Mchedlidze and A. Symvonis. Crossing-optimal acyclic HP-completion for outerplanar st-digraphs. *JGAA*, 15(3):373–415, 2011.

13 R. Nowakowski and A. Parker. Ordered sets, pagenumbers and planarity. *Order*, 6(3):209–218, 1989.

14 A. Wigderson. The complexity of the Hamiltonian circuit problem for maximal planar graphs. Technical report, 298, EECS Department, Princeton University, 1982.

15 M. Yannakakis. Embedding planar graphs in four pages. *Journal of Computer and System Sciences*, 38(1):36–67, 1989.

4.7 Fine-grained complexity of the crossing number of almost planar graphs

Petr Hlinený (Masaryk University – Brno, CZ)

A graph is *almost planar* (or near-planar) if it becomes planar after deleting a suitable one edge. Cabello and Mohar in 2010 [1] proved that, surprisingly, computing the exact crossing number of almost planar graphs is NP-hard. At the same time this problem can be efficiently approximated, up to the factor of maximum degree, by a planar edge insertion solution. Specially, for cubic almost planar graphs, the mentioned edge insertion solves the crossing number exactly. We hence suggest to investigate the possibility of having an FPT algorithm for the exact crossing number of almost planar graphs parameterized by the maximum degree.

Furthermore, one can modify the hardness reduction of Cabello and Mohar in a way that it uses only 16 vertices of degree greater than 3 (this is not published, but it follows from a 2015 paper by Hliněný and Salazar [2] on hardness of joint crossing number). Therefore, it would be interesting to determine the smallest $h > 0$ such that computing the exact crossing number of almost planar graphs with only h vertices of degree greater than 3 is NP-hard (as we know that $h \leq 16$).

References

1 S. Cabello and B. Mohar. Adding one edge to planar graphs makes crossing number hard. In *SoCG*, pages 68–76. ACM, 2010.
2 P. Hliněný and G. Salazar. On Hardness of the Joint Crossing Number. In *ISAAC*, volume 9472 of *LNCS*, pages 603–613. Springer, 2015.

4.8 Labeling Curve Arrangements

Maarten Löffler (Utrecht University, NL)

Introduction. Consider the following problem. Given is a region in the plane (say, a polygon, or a collection of polygons), together with a set of curves that lie in the interior of the region, and which start and end on the boundary of the region. Refer to Figure 1a. Now suppose we wish to annotate these curves with some text describing the meaning of the curves. One option is to write the text along the curve itself (*interior labeling*), but in some applications this is undesirable, as the text might obfuscate other important information. In this case, we may choose to instead extend the curves outside the region, and label (one or both sides of) the curve there. Refer to Figure 1b. When doing this, we have a choise: we can extend each curve on either side. Depending on these choises, we may reach a conclicting labeling (where several labels overlap each other) or not. Refer to Figure 1c. Furthermore, in order to avoid conflicts, we might extend a curve on both sides (and label each side of the curve on another end), or we might extend a curve even farther to move the text away from the region.

This problem was recently studied in the context of *nonogram* generation [1]. A *curved nonogram* is a variation on the classic logic puzzle in which the objective is to colour several cells in an arrangement of curves based on a sequence of *clues*, which are placed outside the diagram [2]. When placing these labels naïvely, conflicts may occur. Refer to Figure 2. Löffler and Nöllenburg show that in general, the problem of finding a non-conflicting labeling of a curve arrangement is NP-hard, but they provide polynomial-time solutions for several restricted settings.

Open Problem. The results from [1] suggest that, while hard in general, the problem may be easy when certain *parameters* are small. Depending on the application, several natural parameters come to mind.

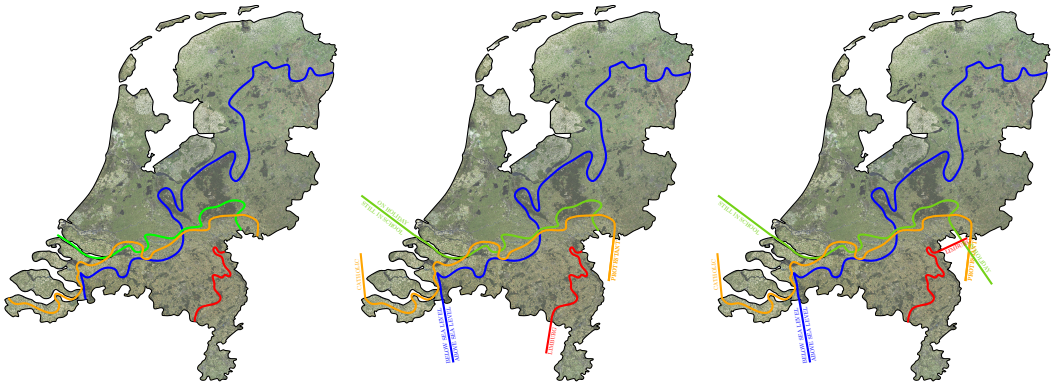
Formally, we may define the input to the curve arrangement labeling problem as:

- a polygon P;
- n pairs of ports on P;
- up to $2n$ label sizes.

The output is then:

- a location of each label;
- a curve from each label to one of the ports.

Refer to Figure 3.



(a) (b) (c)

Figure 1 (a) An arrangement of curves in a polygon. (b) A labeling of some of the curves. Some curves may be labeled only on one side. For curves which are labeled on both sides, we can either place both labels on the same end, or on opposite ends. (c) Some possible locations for curve labels may conflict each other.

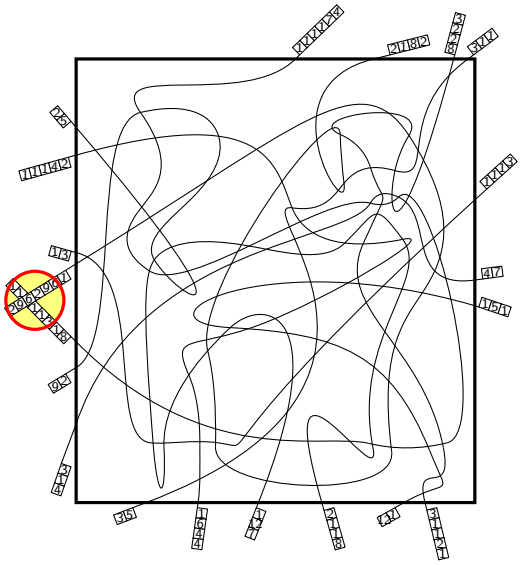


Figure 2 One application of the curve arrangement labeling problem is in automatic puzzle generation.

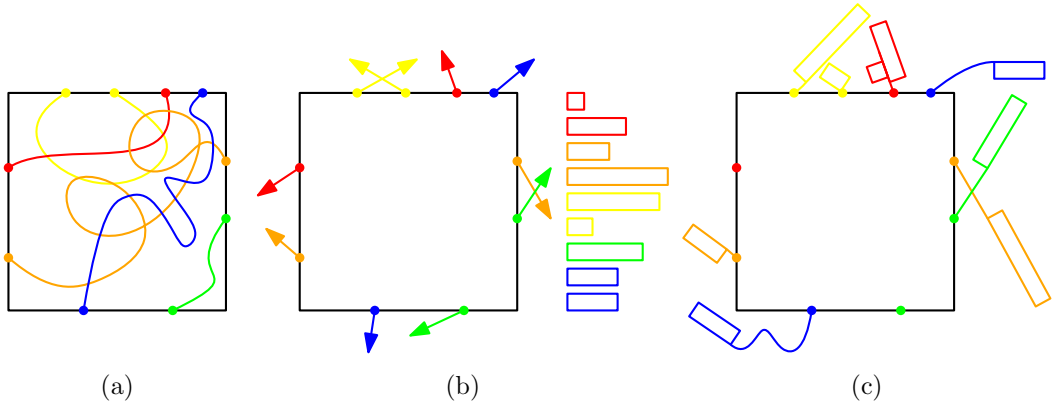


Figure 3 (a) An arrangement of curves in a polygon. (b) The resulting labeling problem. The interior is irrelevant; only the tangent vectors of the curves at the ports are retained. (c) A possible solution.

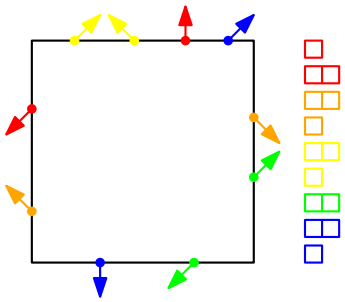


Figure 4 We may restrict the problem in several ways.

The open problem we propose is to investigate the parameterized complexity of the curve arrangement labeling problem. We suggest several possible parameters, which we may classify into *input parameters* (which quantify certain aspects about the problem input) and *output parameters* (which restrict the set of labelings considered).

Possible input parameters include:

- the number of port orientations;
- the maximum label length;
- the complexity of polygon.

Refer to Figure 4.

Possible output parameters include:

- the number of unplaced labels;
- the number of extended labels;
- the maximum extension length;
- the complexity of the extensions;
- the number of outside crossings;
- the number of split labels;
- the size of the bounding box.

References

1 Maarten Löffler and Martin Nöllenburg. Labeling nonograms. In *Proc. 36st European Workshop on Computational Geometry*, pages 53:1–8, 2020.

2 Mees van de Kerkhof, Tim de Jong, Raphael Parment, Maarten Löffler, Amir Vaxman, and Marc van Kreveld. Design and automated generation of japanese picture puzzles. In *Proc. 40th Annual Conference of the European Association for Computer Graphics*, 2019.

4.9 Bend Minimization in Orthogonal Drawings

Ignaz Rutter (Universität Passau, DE) and Meirav Zehavi (Ben-Gurion University, IL)

Let $G = (V, E)$ be a graph with maximum degree 4. In an planar orthogonal drawing of G the vertices are mapped to grid points and the edges are mapped to pairwise non-crossing chains of horizontal and vertical segments that connect the endpoints of each edge. A bend is an interior point of an edge, where a horizontal and a vertical segment meet. Minimizing the number of bends in planar orthogonal drawings is a classical problem. If the graph comes with a fixed combinatorial embedding, then the number of bends can be minimized efficiently [5], whereas without a fixed combinatorial embedding, it is even NP-complete to decide whether there exists a bend-free planar orthogonal drawing [4].

In an attempt to work around the NP-hardness and to gain more control of the drawing, Bläsius et al. introduced two variants of the problem. In FlexDraw the input graph $G = (V, E)$ comes together with a flexibility function $f\colon E \to \mathbb{N}_0 \cup \{\infty\}$, which assigns to each edge e a *flexibility*. The question is whether there exists an orthogonal planar drawing such that each edge e has at most $f(e)$ bends. The problem can be solved in polynomial time if $f(e) > 0$ holds for all $e \in E$[1] and it is FPT with respect to the number of edges with $f(e) = 0$ [2].

The disadvantage of these approaches is that, in the negative case, the algorithm does not output any drawing. To remedy this, Bläsius et al. [3] introduce Optimal Flex Draw, whose input consists of a graph $G = (V, E)$ and for each edge $e \in E$ a cost function $c_e\colon \mathbb{N}_0 \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ that specifies for each edge a cost function. The cost of a drawing is then $\sum -e \in E c_e(b_e)$, where b_e denotes the number of bends in the drawing. They show that an optimal drawing can be found efficiently, if (i) all cost functions are convex and (ii) $c_e(1) = 0$ for all $e \in E$, i.e., the first bend on each edge is free.

Our question is whether Optimal Flex Draw is FPT w.r.t. k if (i) all cost functions are convex and (ii) all but k edges $e \in E$ satisfes $c_e(1) = 0$.

As a response to the above open problem, Meirav Zehavi posed the question whether a similar model could work for finding a drawing that optimizes the number of crossings on graphs that are not necessarily planar, or when no planar drawing is given. Since both problems are in the same spirit, this new problem was called Optimal Flex Crossing.

References

1 Thomas Bläsius, Marcus Krug, Ignaz Rutter, and Dorothea Wagner. Orthogonal graph drawing with flexibility constraints. *Algorithmica*, 68(4):859–885, 2014.
2 Thomas Bläsius, Sebastian Lehmann, and Ignaz Rutter. Orthogonal graph drawing with inflexible edges. *Comput. Geom.*, 55:26–40, 2016.
3 Thomas Bläsius, Ignaz Rutter, and Dorothea Wagner. Optimal orthogonal graph drawing with convex bend costs. *ACM Trans. Algorithms*, 12(3):33:1–33:32, 2016.
4 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001.
5 Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987.

4.10 Is Extending Partial Drawings of Level Planar Graphs FPT?

Ignaz Rutter (Universität Passau, DE)

A *(k-)level graph* is a directed graph $G = (V, E)$ together with a leveling $\ell\colon V \to \{1, \dots, k\}$ such that each directed edge (u, v) satisfies $\ell(u) < \ell(v)$. A *level drawing of* G is a drawing of G where each edge is drawn as a y-monotone curve and each vertex v in V lies on the horizontal line with $y = \ell(v)$. Such a drawing is *level planar*, if its edges do not cross, except at common endpoints. A level graph is *level planar* if it admits a level planar drawing.

Level-planarity has been an active topic of research and it is well-known that level-planar graphs can be recognized in polynomial time. In fact, there are several algorithms that run in quadratic time [6, 3, 2], and even a linear-time algorithm is known [5, 4]. Brückner and Rutter [1] study the variant of the problem where the input comes with a fixed drawing of a subgraph and the question is whether the given drawing can be extended to a level-planar drawing of the whole graph without modifying the predrawn part. Their main result is that the problem can be solved in polynomial time if the input graph has a single source, and otherwise it is NP-complete. The hardness result holds under fairly strong restrictions, which include, e.g., a fixed embedding as well as bounded degree.

On the other hand, if we use the number s of sources in the input graph as our parameter, it is readily seen that there exists an XP-algorithm: Any level-planar drawing can be augmented to a level-planar drawing of a single-source graph by adding $s - 1$ edges. So we can simply guess beforehand $s - 1$ edges that we shall add to remove $s - 1$ sinks and then run the polynomial-time algorithm for single-source graphs. Our open question hence is, is the problem FPT with respect to the number of sources in the input graph?

References

1 Guido Brückner and Ignaz Rutter. Partial and constrained level planarity. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2000–2011. SIAM, 2017.

2 Guido Brückner, Ignaz Rutter, and Peter Stumpf. Level planarity: Transitivity vs. even crossings. In Therese C. Biedl and Andreas Kerren, editors, *Proceedings of the 26th International Symposium on Graph Drawing and Network Visualization (GD'18)*, volume 11282 of *Lecture Notes in Computer Science*, pages 39–52. Springer, 2018.

3 Radoslav Fulek, Michael J. Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Hanani-Tutte, Monotone Drawings, and Level-Planarity. In János Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 263–287. Springer New York, 2013.

4 Michael Jünger and Sebastian Leipert. Level planar embedding in linear time. *J. Graph Algorithms Appl.*, 6(1):67–113, 2002.

5 Michael Jünger, Sebastian Leipert, and Petra Mutzel. Level planarity testing in linear time. In Sue Whitesides, editor, *Graph Drawing, 6th International Symposium, GD'98, Montréal, Canada, August 1998, Proceedings*, volume 1547 of *Lecture Notes in Computer Science*, pages 224–237. Springer, 1998.

6 Bert Randerath, Ewald Speckenmeyer, Endre Boros, Peter L. Hammer, Alexander Kogan, Kazuhisa Makino, Bruno Simeone, and Ondrej Cepek. A satisfiability formulation of problems on level graphs. *Electron. Notes Discret. Math.*, 9:269–277, 2001.

4.11 The Parameterized Complexity of Learning Small Decision Trees in Low-Dimensional Space

Manuel Sorge (TU Wien, AT)

A basic machine-learning and data-analysis task is to classify a given set of examples together with their labels. That is, an *example set* is a set $E \subseteq \mathbb{R}^d$ together with a function $\lambda\colon E \to L$ for a label set L. A decision tree is a fundamental tool to classify example sets and can lead to particularly accessible and visual classifications. In a simple form, we have an ordered rooted tree, for each inner node a dimension from $\{1, 2, \ldots, d\}$ and a threshold in $\mathbb{R}$, and each leaf of the tree has a label from L. To classify a given example e, we move through the tree as follows, starting from the root. At each node t we ask whether e's entry in t's dimension is less than or equal to the threshold at t. If so we move to the left child and otherwise to the right child. The class of the example e is then the label of the leaf of the tree at which we arrive in this manner. We say that the tree *decides* E if the class assigned by the tree to each example e in E agrees with $\lambda(e)$.

Heuristics for computing decision trees have been studied since at least the 1970s [1, 2] and many machine-learning libraries implement one of them. Apart from optimizing other parameters, often these heuristics minimize the size of the obtained decision tree. Hence the computational complexity of the following problem is interesting to know: In Decision Tree Size we are given an example set E and want to compute the minimum size of a decision tree for E. Decision Tree Size was known to be NP-complete since the 1970s [2]. However, to my knowledge, more fine-grained investigation into the complexity of Decision Tree Size in form introduced above started only recently with [3]. In particular, the problem remains W[2]-hard with respect to the size of the tree and hence it is interesting to study the parameterized complexity of Decision Tree Size with respect to other small parameters.

A mainstay in data analysis is performing dimensionality-reduction techniques, e.g. based on principal-component analysis, prior to using classification methods. The case where the number d of dimensions of the example space is small is thus an interesting special case. Marcin Pilipczuk pointed out to me that there is a simple dynamic programming algorithm that solves Decision Tree Size in $n^{O(d)}$ time, where n is the number of input examples. In the seminar I asked: *Is Decision Tree Size fixed-parameter tractable with respect to the number d of dimensions?*

References

1 L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Chapman & Hall/CRC, 1984. ISBN 0-534-98053-8.

2 L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976. doi: 10.1016/0020-0190(76)90095-8.

3 S. Ordyniak and S. Szeider. Parameterized complexity of small decision tree learning. In *Proceedings of the 35th AAAI Converence on Artificial Intelligence (AAAI '21)*, pages 6454–6462. AAAI Press, 2021.

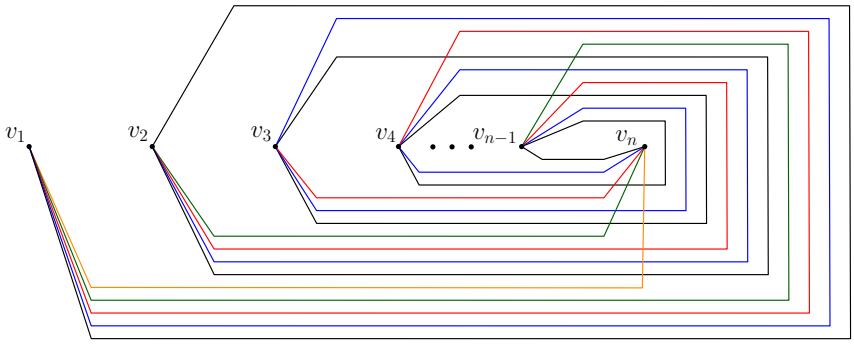


Figure 5 Construction taken from [3]: A simple drawing of K_n that does not contain a triangulation. (All empty triangles have either the edge v_1v_2 or the edge $v_{n-1}v_n$ on it's boundary [3]. This drawing is also called the twisted drawing or Harborth's drawing.) The edges in this figure are colored for easier visibility; otherwise, the colors do not have any significance.

4.12 Two open problems on drawings of complete graphs

Birgit Vogtenhuber (TU Graz, AT)

Problem 1: Triangulations in simple drawings of the complete graph

A *simple drawing* of a graph $G = (V, E)$ in the plane is a drawing where vertices are distinct points, edges are Jordan arcs connecting their endpoints, and any pair of edges intersects at most once (either in a common endpoint or at a proper crossing in the relative interior of both edges). In a simple drawing D of a graph $G = (V, E)$, the edges of any three pairwise connected vertices form a Jordan curve that we call *triangle*. Any such triangle divides the plane into two regions. A triangle with vertices v_1, v_2, v_3 is called *empty* if the one of those regions does not contain any of the vertices $V \setminus \{v_1, v_2, v_3\}$.

A *triangulation* of a simple drawing D of the complete graph is a connected plane subdrawing of D in which every (bounded) face is an empty triangle (one might require the unbounded face of the subdrawing to be an empty triangle as well, or allow it to be a Jordan curve consisting of more than three edges of D). In simple drawings of the complete graph, any three vertices induce a triangle. However, not all simple drawings of the complete graph contain triangulations; see the below Figure for an example. This prompts the following open problem, which originally has been asked in [2].

Open Problem. What is the complexity of deciding whether a simple drawing of the complete graph contains a triangulation?

Related Results. Given a simple drawing D of the complete graph, and a cardinality k, it is NP-complete to decide whether there is a plane subdrawing of D that has at least k edges [7]. (This result has been proven in [7] via a reduction from the independent set problem: Given a set of segments in the plane that pairwise either are disjoint or intersect in a proper crossing, and an integer $k > 0$, it is NP-complete to decide whether there is a subset of k disjoint segments [4].) For straight-line drawings of the complete graph, it is easy to see that there always are triangulations. However, given a straight-line drawing of a non-complete graph, it is again NP-hard to decide whether it contains a triangulation [5].

Problem 2: The 2-colored crossing number for straight-line drawings of complete graphs

A *straight-line drawing* of G is a drawing D of G in the plane in which the vertices are drawn as points in general position, that is, no three points on a line, and the edges are drawn as straight line segments. A 2-*edge-coloring* of D of a graph is an assignment of one of k possible colors to every edge of D. The 2-*colored crossing number* of D is the minimum number of monochromatic crossings (pairs of edges of the same color that cross) in any 2-edge-coloring of D.

Open Problem. What is the complexity of deciding whether the 2-colored crossing number of a straight-line drawing D of the complete graph K_n is at most k?

Remarks. Bounds on ratio between the 2-colored crossing number of a drawing D of K_n in relation to the total number of crossings in D have been studied in [1]. For (straight-line drawings of) general graphs, this problem is known to be NP-complete, even if the underlying point set in convex position [6]. The problem corresponds to finding a maximum cut in the segment intersection graph that is induced by the edges of the drawing.

References

1 Oswin Aichholzer, Ruy Fabila-Monroy, Adrian Fuchs, Carlos Hidalgo-Toscano, Irene Parada, Birgit Vogtenhuber, and Francisco Zaragoza. On the 2-colored crossing number. In Daniel Archambault and Csaba D. Tóth, editors, *Graph Drawing and Network Visualization*, pages 87–100, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-35802-0_7.

2 Oswin Aichholzer, Thomas Hackl, Alexander Pilz, Pedro Ramos, Vera Sacristán, and Birgit Vogtenhuber. Empty triangles in good drawings of the complete graph. *Graphs and Combinatorics*, 31(2):335–345, 2015. doi:10.1007/s00373-015-1550-5.

3 Heiko Harborth. Empty triangles in drawings of the complete graph. *Discrete Mathematics*, 191(1–3):109–111, 1998. doi:10.1016/S0012-365X(98)00098-3.

4 Jan Kratochví and Jaroslav Nešetřil. Independent set and clique problems in intersection-defined classes of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 31(1):85–93, 1990. URL: http://eudml.org/doc/17810.

5 Errol L. Lloyd. On triangulations of a set of points in the plane. In *Proc. 18th Annu. Symposium on Foundations of Computer Science*, pages 228–240, 1977. URL: https://ieeexplore.ieee.org/document/4567947.

6 S. Masuda, K. Nakajima, T. Kashiwabara, and T. Fujisawa. Crossing minimization in linear embeddings of graphs. *IEEE Transactions on Computers*, 39(1):124–127, 1990. doi:10.1109/12.46286.

7 Alfredo García Olaverri, Alexander Pilz, and Javier Tejel Altarriba. On plane subgraphs of complete topological drawings. *ARS MATHEMATICA CONTEMPORANEA*, 20(1):69–87, 2021. doi:10.26493/1855-3974.2226.e93.

5 Working groups

5.1 Progress on Upward Planarity Testing

Robert Ganian (TU Wien, AT), Steven Chaplick (Maastricht University, NL), Emilio Di Giacomo (University of Perugia, IT), Fabrizio Frati (University of Rome III, IT), Chrysanthi Raftopoulou (National Technical University of Athens, GR), and Kirill Simonov (TU Wien, AT)

The first problem considered by this working group is Upward Planarity Testing (UP). In UP, the input is a directed acyclic graph (DAG) G and the question is whether there exists a planar drawing of G such that all edges are drawn upward, i.e., all edges monotonically increase in the vertical direction.

UP has been extensively studied in the literature, and arises naturally in a number of situations where the aim is to obtain easy-to-parse planar representations of DAGs. The problem has been shown to be NP-complete already 25 years ago [8, 9], but the first polynomial-time algorithms for restricted variants of UP have been published already in the early nineties [13]. Among others, the problem is known to be polynomial-time tractable when G is provided with a plane embedding [1] (which also implies polynomial-time tractability for triconnected DAGs, since these admit a single plane embedding), or restricted to class of single-source DAGs [2], the class of outerplanar DAGs [16] or the class of orientations of series-parallel graphs [6].

In spite of the broad range of results for UP on specific subclasses of instances, the problem is considerably less explored from the parameterized complexity perspective. It was shown that UP is fixed-parameter tractable when parameterized by the cyclomatic number of the input DAG (or, equivalently, the feedback edge number of the underlying undirected graph of G) [5], and also when parameterized by the number of triconnected components and cut vertices [11].

We began our investigation by considering structural parameters for UP. Using standard reduction arguments, we could show that UP admits a polynomial kernel when parameterized by the vertex cover number of (the underlying undirected graph of) G. Moreover, we could strengthen these arguments to show that UP is fixed-parameter tractable when parameterized by the treedepth of (the underlying undirected graph of) G.

On a high level, the idea behind this result can be summarized as follows. We start by employing known results to compute a treedepth decomposition T for G [15]. T is a rooted tree over the vertices of G with the property that the endpoints of every arc in G have an ancestor-descendant relationship in T, with the property that the height of the tree (i.e., the maximum distance between a leaf and the root r) is at most the parameter value k. Let the level of a node v in T be its distance from the root r in T. Consider a node v on level i in T with the property that v has at least $f(k,i)$-many children in T, for some well-defined and computable function f. We can then identify, in fixed-parameter time, a child w of v such that deleting the subtree of T rooted at w results in a subgraph which admits an upward planar drawing if and only if G admits an upward planar drawing. In other words, in this case we can reduce the size of the instance and restart the algorithm on a strictly smaller instance. On the other hand, if every node v on level i in T has at most $f(k,i)$-many children, then G has size bounded by a function of k, and in particular it can be solved by a brute-force algorithm with runtime depending exclusively on k.

The general high-level approach outlined above is not entirely new, as it has been applied to obtain fixed-parameter algorithms for a handful of other problems parameterized by treedepth [7, 3]. However, the main technical challenge lies in the subtask of identifying a suitable child w that could be pruned from G if v had sufficiently many children, and in arguing that this operation is safe – i.e., that the resulting DAG is equivalent to G. To resolve this, we had to obtain a sufficient level of geometric insight into the problem's behavior and apply a "swapping" argument whose details go beyond the scope of this brief report.

We then turned our attention to a different parameterization for the problem: the number of sources (s) and/or the number of sinks (t) of the input DAG. This was motivated by the fact that UP was shown to admit a non-trivial polynomial-time algorithm for the case when G has a single source [2]. Moreover, it is not difficult to observe that UP parameterized by s and/or by t can be reduced to the single-source case via branching that can be carried out in time $O(n^{\min(s,t)})$, which places the problem in XP for these parameterizations.

Our aim here was to determine whether this result could be strengthened to a fixed-parameter algorithm. While we made considerable progress towards this goal and are now convinced that this should be possible, some technical issues remain that we plan to address in follow-up virtual meetings. On the other hand, during our work we have already developed all the ingredients required to show that UP is fixed-parameter tractable when parameterized by $s + t$. The algorithm showing this is non-trivial, and a detailed summary exceeds the scope of this report: on a high level, we perform dynamic programming along the SPQR tree decomposition of G [12, 10], whereas we can show that at each node (which may be rigid, parallel or series, all of which must be handled separately) the number of decisions that need to be made and have an impact on whether the resulting drawing is upward planar or not can be upper-bounded to a function of k alone.

Last but not least, we briefly also considered the related problem of Partial Level Planarity Testing (PLP). There, we are given an undirected graph G where each vertex is assigned to a level (i.e., an integer), and some subset H of the vertices are already drawn on the plane. The question is whether there exists a drawing of G which extends H and places each vertex of G in a way which matches the vertical levels prescribed on the input – in particular, two vertices must have the same y-coordinate if and only if they have the same level, vertex a has a higher y-coordinate than vertex b if and only if a has a higher level than b, and all edges monotonically increase in the vertical direction. Unlike UP, Level Planarity Testing (i.e., PLP when $H = \emptyset$) is polynomial-time tractable [14], but PLP is NP-hard in general [4]. As the final result for this report, we mention that we have made considerable progress towards showing that PLP is fixed-parameter tractable when parameterized by $|H|$; only a single technical hurdle remains, and we are optimistic that it will be resolved during the next follow-up meeting or two.

References

1. P. Bertolazzi, G. D. Battista, G. Liotta, and C. Mannino. Upward drawings of triconnected digraphs. *Algorithmica*, 12(6):476–497, 1994.
2. P. Bertolazzi, G. D. Battista, C. Mannino, and R. Tamassia. Optimal upward planarity testing of single-source digraphs. *SIAM J. Comput.*, 27(1):132–169, 1998.
3. S. Bhore, R. Ganian, F. Montecchiani, and M. Nöllenburg. Parameterized algorithms for queue layouts. In D. Auber and P. Valtr, editors, *Graph Drawing and Network Visualization – 28th International Symposium, GD 2020, Vancouver, BC, Canada, September 16-18, 2020, Revised Selected Papers*, volume 12590 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2020.

4 G. Brückner and I. Rutter. Partial and constrained level planarity. In P. N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2000–2011. SIAM, 2017.

5 H. Y. Chan. A parameterized algorithm for upward planarity testing. In S. Albers and T. Radzik, editors, *Algorithms – ESA 2004, 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004, Proceedings*, volume 3221 of *Lecture Notes in Computer Science*, pages 157–168. Springer, 2004.

6 W. Didimo, F. Giordano, and G. Liotta. Upward spirality and upward planarity testing. *SIAM J. Discret. Math.*, 23(4):1842–1899, 2009.

7 R. Ganian and S. Ordyniak. The complexity landscape of decompositional parameters for ILP. *Artif. Intell.*, 257:61–71, 2018.

8 A. Garg and R. Tamassia. On the compuational complexity of upward and rectilinear planarity testing. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing, DIMACS International Workshop, GD '94, Princeton, New Jersey, USA, October 10-12, 1994, Proceedings*, volume 894 of *Lecture Notes in Computer Science*, pages 286–297. Springer, 1994.

9 A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001.

10 C. Gutwenger and P. Mutzel. A linear time implementation of spqr-trees. In J. Marks, editor, *Graph Drawing, 8th International Symposium, GD 2000, Colonial Williamsburg, VA, USA, September 20-23, 2000, Proceedings*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2000.

11 P. Healy and K. Lynch. Two fixed-parameter tractable algorithms for testing upward planarity. *Int. J. Found. Comput. Sci.*, 17(5):1095–1114, 2006.

12 J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973.

13 M. D. Hutton and A. Lubiw. Upward planar drawing of single source acyclic digraphs. In A. Aggarwal, editor, *Proceedings of the Second Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 28-30 January 1991, San Francisco, California, USA*, pages 203–211. ACM/SIAM, 1991.

14 M. Jünger, S. Leipert, and P. Mutzel. Level planarity testing in linear time. In S. Whitesides, editor, *Graph Drawing, 6th International Symposium, GD'98, Montréal, Canada, August 1998, Proceedings*, volume 1547 of *Lecture Notes in Computer Science*, pages 224–237. Springer, 1998.

15 J. Nesetril and P. O. de Mendez. *Sparsity – Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012.

16 A. Papakostas. Upward planarity testing of outerplanar dags. In R. Tamassia and I. G. Tollis, editors, *Graph Drawing, DIMACS International Workshop, GD '94, Princeton, New Jersey, USA, October 10-12, 1994, Proceedings*, volume 894 of *Lecture Notes in Computer Science*, pages 298–306. Springer, 1994.

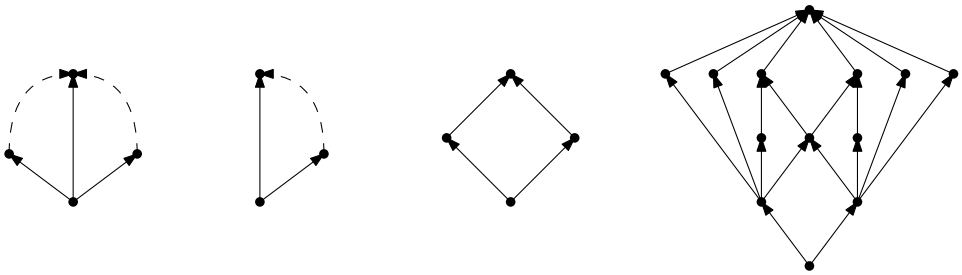


Figure 6 (a) The forbidden configuration, (b) a generalized triangle, (c) a rhombus, and (d) a graph that is not upward 2-page book embeddable.

5.2 Progress on Embedding Upward Planar Graphs in two Pages

Michael A. Bekos (Universität Tübingen, DE), Giordano Da Lozzo (University of Rome III, IT), Fabrizio Frati (University of Rome III, IT), Martin Gronemann (Universität Osnabrück, DE), and Chrysanthi Raftopoulou (National Technical University of Athens, GR)

The general problem of whether an *st*-planar graph can be embedded into two pages or not has remained unanswered so far. However, for some special cases there exist efficient recognition algorithms. Mchedlidze and Symvonis [3] consider the case in which the input is triangulated. They show that if no transitive edge bounds two faces, the graph can be embedded in two pages. Furthermore, they prove that this condition is also sufficient for triangulations. Afterwards, Binucci et al. [1] extend this result to larger faces. Crucial in their work is the notion of forbidden configuration (see Fig. 6a, which consists of a transitive edge that bounds two internal faces (that are not necessarily triangles). Note that this is a generalization of the configuration used by Mchedlidze and Symvonis [3]. Clearly, the absence of forbidden configurations is a necessary condition for the existence of an upward 2-page book embedding. However, there exist examples that do not contain such a forbidden configuration and still do not admit an upward 2-page book embedding [4]; see Fig. 6d. Hence, the condition is not sufficient.

Based on their forbidden configuration Binucci et al. present in their work a linear-time recognition algorithm for a special class of *st*-planar graphs. Crucial in their paper is the notion of *generalized triangle*, i.e. an internal face bounded by a transitive edge; see Fig. 6b. Furthermore, the define a *rhombus* to be a face of size four that is not bounded by a transitive edge, i.e., the left and right path are both of length two; see Fig. 6c. They show that for an *st*-planar graph that solely consists of faces that are either a rhombus or a generalized triangle, one can decide in linear-time whether the graph admits an upward 2-page book embedding or not.

We tackle the problem from a different perspective and extend the concept of forbidden configurations. Central in our approach is the so-called bitonic *st*-ordering, which was introduced by Gronemann [2] to obtain upward planar polyline drawings of small size. Intuitively, a bitonic *st*-ordering forms a special type of *st*-ordering that takes the underlying embedding into account. The idea is that for a given embedding and an *st*-ordering, one considers the order of the successors of a vertex as they appear in the embedding.

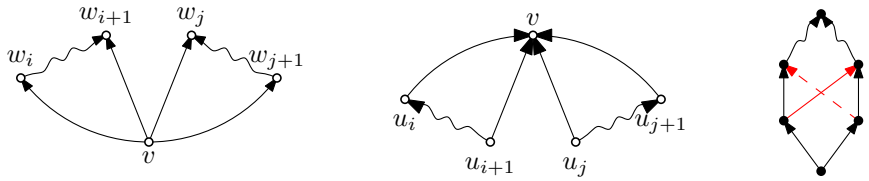


Figure 7 (a)-(b) The butterfly configuration, (c) splitting a large face.

Gronemann [2] showed that if these form an increasing and then decreasing sequence in the *st*-ordering, one may obtain an upward planar straight-line drawing of the underlying *st*-planar graph. We take this idea and adapt it to upward 2-page book embeddings.

Consider an *st*-planar graph $G = (V, E)$ and an upward 2-page book embedding of it. Clearly, the ordering of the vertices on the spine is a feasible *st*-ordering π for G. Moreover, the book embedding is a planar embedding of G. When now considering a vertex v and its successors $S(v) = \{w_1, \ldots, w_s\}$ with $vw_i \in E$ and $1 \leq i \leq s$ ordered as the appear in the embedding, then one can observe that $\pi(w_1) > \ldots > \pi(w_k) < \ldots < \pi(w_s)$ holds for some $1 \leq k \leq s$. Symmetrically, we can make the same observations for all predecessors $P(v) = \{u_1, \ldots u_p\}$. That is $\pi(u_1) < \ldots < \pi(u_l) > \ldots > \pi(u_p)$ holds for some $1 \leq l \leq p$. In other words, the successors form a bitonic decreasing sequence w.r.t. to the spine ordering, while the predecessors from a bitonic increasing sequence. Gronemann [2] identified forbidden configurations in the graph that prevent the existence of *st*-orderings with such properties. We adapt these configurations to both the successors and predecessors. Figure 7 shows the forbidden configuration for both. We refer to these configurations as *butterfly*. Without proof, we observe:

▶ **Lemma 1.** *Let G be an embedded st-planar graph. If G contains a butterfly configuration, then G does not admit an upward 2-page book embedding.*

Note that a butterfly is a generalization of the forbidden configuration of Binucci et al. [1]. Our idea is to assume the absence of butterflies and augment the graph by adding edges. One promising strategy is to split large faces that are not generalized triangles to reduce the size of the largest face. In particular, we split such a large face into a smaller face and a rhombus; refer to Fig. 7c. The overall challenge with this approach is that one has two choices to perform such a split. While the augmentation with one edge is always possible, inserting the second edge may create too many restrictions.

References

1 C. Binucci, G. Da Lozzo, E. D. Giacomo, W. Didimo, T. Mchedlidze, and M. Patrignani. Upward book embeddings of st-graphs. In G. Barequet and Y. Wang, editors, *SoCG 2019*, volume 129 of *LIPIcs*, pages 13:1–13:22. Schloss Dagstuhl, 2019.

2 M. Gronemann. Bitonic st-orderings for upward planar graphs. In Y. Hu and M. Nöllenburg, editors, *Graph Drawing and Network Visualization*, volume 9801 of *LNCS*, pages 222–235. Springer, 2016.

3 T. Mchedlidze and A. Symvonis. Crossing-optimal acyclic HP-completion for outerplanar st-digraphs. *JGAA*, 15(3):373–415, 2011.

4 R. Nowakowski and A. Parker. Ordered sets, pagenumbers and planarity. *Order*, 6(3):209–218, 1989.

5.3 Progress on A Parameterized Approach to Orthogonal Compaction

Philipp Kindermann (Universität Trier, DE), Walter Didimo (University of Perugia, IT), Siddharth Gupta (Ben-Gurion University, IL), Giuseppe Liotta (University of Perugia, IT), Alexander Wolff (Universität Würzburg, DE), and Meirav Zehavi (Ben-Gurion University, IL)

Background

Orthogonal Drawings and Representations. Let $G = (V, E)$ be a connected planar graph of vertex-degree at most four. A *planar orthogonal drawing* Γ of G maps each vertex $v \in V$ to a point p_v of the plane and each edge $e = (u, v) \in E$ to an alternating sequence of horizontal and vertical segments connecting p_u and p_v. A point of an edge of Γ in which a horizontal segment and a vertical segment meet is called a *bend*. We assume that in Γ all the vertices and bends have integer coordinates, i.e., we assume that Γ is an integer-coordinate grid drawing. Two planar orthogonal drawings Γ_1 and Γ_2 of G are *shape-equivalent* if: (i) Γ_1 and Γ_2 have the same planar embedding; (ii) for each vertex $v \in V$, the geometric angles at v (formed by any two consecutive edges incident on v) are the same in Γ_1 and Γ_2; (iii) for each edge $e = (u, v) \in E$ the sequence of left and right bends along e while moving from u to v is the same in Γ_1 and Γ_2. An *orthogonal representation* (also called *orthogonal shape*) H of G is a class of shape-equivalent planar orthogonal drawings of G. It follows that an orthogonal representation H is completely described by a planar embedding of G, by the values of the angles around each vertex (each angle being a value in the set $\{90°, 180°, 270°, 360°\}$), and by the ordered sequence of left and right bends along each edge (u, v), moving from u to v; if we move from v to u this sequence and the direction (left/right) of each bend are reversed. If Γ is a planar orthogonal drawing in the class H, we also say that H is the orthogonal representation of Γ or that Γ *preserves* H. Without loss of generality, from now on we also assume that an orthogonal representation H comes with a given orientation, i.e., for each edge segment $\overline{pq}$ of H (where p and q correspond to vertices or bends), we fix whether p lies to the left, to the right, above, or below q.

The Orthogonal Compaction Problem. Let H be an orthogonal representation of a connected planar graph G. The compaction problem for H asks to compute a minimum-area planar orthogonal drawing that preserves H. In other words, it asks to assign integer coordinates to the vertices and to the bends of H such that the area of the resulting planar orthogonal drawing is minimum among all planar orthogonal drawings that preserve H. In the following, we will refer to this problem as the Orthogonal Compaction (OC) problem.

Previous Work. Patrignani proved that OC is NP-hard in the general case [5]; the problem remains NP-hard even for orthogonal representations of cycles [4]. Nevertheless, Bridgeman et al. [2] showed that OC can be solved in linear time for a subclass of orthogonal representations called *turn-regular*. Informally speaking, a face of a planar orthogonal representation H is turn-regular if it does not contain a pair of reflex corners (i.e., turns of 270°) that point to each other; H is turn-regular if all its faces are turn-regular.

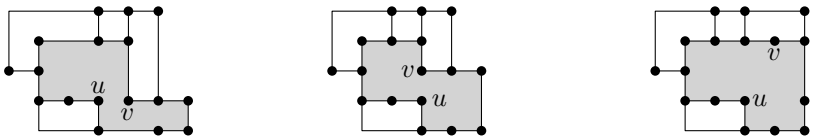


Figure 8 (Left) Drawing of a non-turn-regular orthogonal representation H; vertices u and v point to each other in the gray face, thus they represent a pair of kitty corners. (Middle) Another drawing of H with minimum area. (Right) Drawing of a turn-regular orthogonal representation of the same graph.

More formally, let f be a face of H and assume that the boundary of f is always traversed counterclockwise (resp. clockwise) if f is internal (resp. external). Let c and c' be two reflex corners of f (corresponding to either vertices or bends)[2]. Let $\mathsf{rot}(c, c')$ be the number of convex corners minus the number of reflex corners encountered while traversing the boundary of f from c (included) to c' (excluded); a reflex corner corresponding to a vertex of degree one must be counted like two reflex corners. We say that c and c' is a pair of *kitty corners* of f if $\mathsf{rot}(c, c') = 2$ or $\mathsf{rot}(c', c) = 2$. A vertex is a kitty corner if it is part of a pair of kitty corners. Notice that the following property holds:

▶ Property 1. Let c and c' be two reflex corners of a face f. If f is internal, then $\mathsf{rot}(c, c') = 2$ if and only if $\mathsf{rot}(c', c) = 2$. If f is external, then $\mathsf{rot}(c, c') = 2$ if and only if $\mathsf{rot}(c', c) = -6$.

A face f of H is *turn-regular* if it does not contain a pair of kitty corners. The orthogonal representation H is *turn-regular* if all faces are turn-regular. Figure 8 shows two different drawings of the same orthogonal representation H that is not turn-regular, and a drawing of a turn-regular orthogonal representation of the same graph.

If H is turn-regular, then the compaction problem for H can be solved in linear time by independently solving two one-dimensional compaction problems for H; one in the x-direction and the other in the y-direction [2]. Namely, for the x-direction, the one-dimensional compaction is solved as follows: (i) Construct a planar DAG D_x whose nodes are the maximal vertical chains of H and such that two nodes are connected by an arc (oriented from left to right) if the corresponding vertical chains are connected by a horizontal segment in H; (ii) augment D_x to become a planar st-graph; (iii) apply an optimal topological numbering X to D_x (see [3], p. 89); the number $X(u)$ that is assigned to a node u determines the x-coordinate of all vertices of H in the vertical chain corresponding to u. The one-dimensional compaction in the y-direction is solved symmetrically.

Unfortunately, if H is not turn-regular, then the aforementioned approach fails; solving independently the one-dimensional compaction problem in the x- and in the y-directions may lead to non-planar drawings. This is due to the fact that, if c and c' form a pair of kitty corners, a directed path connecting the two (horizontal or vertical) maximal chains that include c and c' does not exist, neither in D_x nor in D_y.

Parameterized Analysis of the Compaction Problem

We initiate the study of the parameterized complexity of the OC problem. We study the complexity of the problem parameterized by the following parameters.

[2] For simplicity, one can assume that each bend of an orthogonal representation is replaced with a dummy vertex, so that all corners in a face correspond to vertices.

Kitty corners. Since the absence of kitty corners in an orthogonal representation suffices to solve OC efficiently, the most natural parameter to be considered is the number of kitty corners.

Number of Faces. Since OC remains NP-hard for orthogonal representations of cycles [4], we cannot expect an FPT (or even XP) algorithm parameterized by the number of faces of the embedded graph alone.

Face-Degree. The degree of a face is the number of vertices incident to the face, and the *face-degree* of an embedded graph is the maximum degree among all of its faces. Since both the NP-hardness reduction by Patrignani [5] and Evans et al. [4] require faces of linear size, it is interesting to know whether constant-size faces make the problem tractable.

Treewidth and Pathwidth. A *tree decomposition* of a graph $G = (V, E)$ is a tree $T = (V_T, E_T)$ and a function $\beta \colon V_T \to 2^V$ that assigns, to each node in T, a subset of vertices of G such that: (i) for each edge $(u, v) \in E$, the vertices u and v appear in a common subset; and (ii) for each vertex $v \in V$, the subsets in which v appears form a nonempty subtree (rather than just a subforest) in T. The *width* of T is one less than the size of the largest subset assigned by β, and the *treewidth* of G is the minimum width among all possible tree decompositions of G. Similarly, a *path decomposition* is a tree decomposition where T is a path, and the *pathwidth* of G is defined analogously to the treewidth of G.

Treewidth and pathwidth are among the most frequently used parameters in parameterized complexity. However, since cycles have constant pathwidth and treewidth (that is equal to 2), we also cannot expect an FPT (or even XP) algorithm parameterized by either of these two parameters alone.

Height. The *height* of a graph G is the minimum number of distinct y-coordinates required to draw the graph. In the case of orthogonal drawings, this is the same as the number of rows required. Since a $W \times H$ grid has pathwidth at most H, graphs with bounded height have bounded pathwidth, but the converse is generally not the case [1].

Our Results

We develop an XP algorithm, and then an FPT algorithm, parameterized by the number of kitty corners. For the XP algorithm, the idea is to "guess", for each pair of kitty corners $\{u, v\}$, the relative positions of u and v, i.e., $x(u) \lesseqgtr x(v)$ and $y(u) \lesseqgtr y(v)$. For the FPT algorithm, more involved arguments are required to reduce the number of pairs of kitty corners for which the relative position has to be guessed. The rough idea is to explore a suitable set of planar edge augmentations of the two DAGs D_x and D_y resulting from the orthogonal representation H; an augmenting edge connects a pair of nodes (i.e., vertical/horizontal chains of H) that involve a pair of kitty corners. For each combination of planar augmentations of D_x and D_y, one can further augment each of the two DAGs with edges that make it an st-planar graph, and then compute a pair of optimal topological numberings to determine the x- and the y-coordinates of a minimum-area drawing of H, within the given relative positions for the kitty corners.

▶ **Theorem 1.** *OC admits an* XP *algorithm parameterized by the number of kitty corners.*

▶ **Theorem 2.** *OC admits an* FPT *algorithm parameterized by the number of kitty corners.*

Note that the second theorem subsumes the first theorem. Unfortunately, our FPT algorithm does not imply a polynomial kernel for the problem. If we, however, take the sum of the number of kitty corners and the number of faces as parameter, then we can obtain a polynomial kernel.

▶ **Theorem 3.** *OC admits a polynomial kernel if parameterized by the sum of the number of kitty corners and the number of faces.*

By adjusting Patrignani's NP-hardness proof for OC [5] accordingly, we show that the problem remains NP-hard even if all faces have constant degree.

▶ **Theorem 4.** *OC is* NP*-complete even for graphs of constant face-degree.*

By "guessing" for every column of the drawing which vertex or edge lies in each of the grid points, we obtain an XP algorithm parameterized by the height of the graph.

▶ **Theorem 5.** *OC admits an* XP *algorithm parameterized by the height of the graph.*

So, while OC is unlikely to admit an XP algorithm with respect to pathwidth, it admits an XP algorithm with respect to height. This motivates us to define a parameterization that lies "in-between" pathwidth and height: that can be arbitrarily smaller than height, yet yield an XP algorithm. In addition to OC, we prove that our new parameterization is of independent interest, being relevant to several other problems in Graph Drawing.

Future Work

The following questions remain open, and will be part of our future research.

- Can we find a polynomial kernel for OC with respect to only the number of kitty corners?
- Does OC admit an FPT algorithm parameterized by the height of the graph?
- Is OC solvable in $2^{O(\sqrt{n})}$ time? This bound is tight assuming that the Exponential Time Hypothesis is true.

References

1 Therese C. Biedl. Small drawings of outerplanar graphs, series-parallel graphs, and other planar graphs. *Discret. Comput. Geom.*, 45(1):141–160, 2011. Doi: 10.1007/s00454-010-9310-z

2 Stina S. Bridgeman, Giuseppe Di Battista, Walter Didimo, Giuseppe Liotta, Roberto Tamassia, and Luca Vismara. Turn-regularity and optimal area drawings of orthogonal representations. *Comput. Geom.*, 16(1):53–93, 2000. Doi: 10.1016/S0925-7721(99)00054-1

3 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs.* Prentice-Hall, Upper Saddle River, 1999.

4 William S. Evans, Krzysztof Fleszar, Philipp Kindermann, Noushin Saeedi, Chan-Su Shin, and Alexander Wolff. Minimum rectilinear polygons for given angle sequences. *Comput. Geom.*, 100(101820):1–39, 2022. Doi: 10.1016/j.comgeo.2021.101820

5 Maurizio Patrignani. On the complexity of orthogonal compaction. *Comput. Geom.*, 19(1):47–67, 2001. Doi: 10.1016/S0925-7721(01)00010-4

5.4 Progress on Almost Separated Fixed Order Stack Layouts

Johannes Zink (Universität Würzburg, DE), Martin Gronemann (Universität Osnabrück, DE), Thekla Hamm (TU Wien, AT), Boris Klemz (Universität Würzburg, DE), Martin Nöllenburg (TU Wien, AT), and Birgit Vogtenhuber (TU Graz, AT)

In the course of the workshop, we jointly developed the following preliminary results on the open problem on *Almost Separated Fixed Order Stack Layouts*. We briefly recall the setting and fix some terminology: As instance we consider a graph G together with an ordering σ

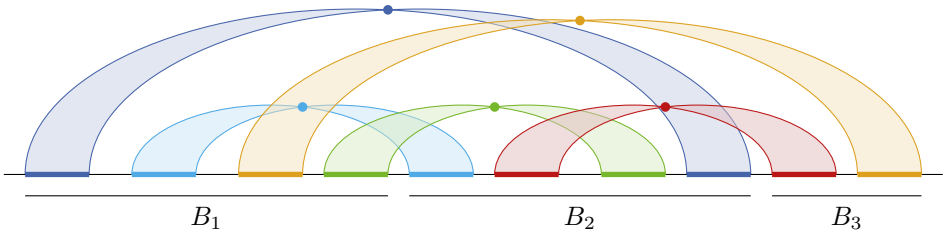


Figure 9 Instance with $k = 3$ and largest twist size t that needs at least $3t/2$ stacks.

of its vertices and $s \in \mathbb{N}$. Additionally we define the *number of blocks* of σ as the smallest number k such that $V(G)$ can be partitioned into k sets $B_1, \ldots, B_k$ of vertices which we call *blocks* that are consecutive in σ such that for all i, and two vertices $v, v' \in B_i$, v and v' are not connected by an edge in G. It is straightforward to observe that G must be p-colorable for some $p \leq k$. Our task is to decide whether the fixed-order stack number of G with respect to σ, denoted by $\mathrm{fosn}(G, n)$, is at most s, using k as a parameter.

A natural lower bound for the fixed-order stack number is the size t of a largest *twist* of a graph, i.e., a maximum size set of mutually intersecting edges (note that the order of the vertices determines which edges intersect when drawn on the same page). An interesting structural observation is that, even for constant k, t plus an additive constant does not upper-bound the fixed-order stack number s as we show by the following simple observation.

▶ Proposition 1. For $k = 3$, there are bipartite graphs and vertex orderings requiring $3t/2$ stacks, where t is the size of the largest twist.

Proof sketch. Each colored bundle in Fig. 9 represents a twist of size $t/2$ and one can easily verify that the largest twist of the entire instance has size t. Since the red and green bundles intersect, they need their separate sets of $t/2$ pages each. The dark blue bundle can be added to pages with green edges, while the light blue bundle can be added to pages with red edges (otherwise they would already increase the fixed-order stack number). Since the orange bundle, however, intersects both blue bundles, it has to use a new set of $t/2$ pages, resulting in a fixed-order stack number of $3t/2$. ◀

On the positive side, we provide the following XP algorithm and approximation algorithm.

▶ **Theorem 1.** *Deciding whether a graph $G = (V, E)$ with respect to a given ordering σ of V with k blocks has fixed-order stack number $\leq s$, is in* XP *in $s + k$. More precisely, there is an algorithm deciding whether $\mathrm{fosn}(G, \sigma) \leq s$ in time $O(m^{4sk+5/2} 4^{sk} sk)$, where $m = |E|$.*

Proof sketch. On each page of a fixed-order stack layout with respect to σ, the set of edges between any two blocks must pairwise nest, i.e. no two edges have endpoints that alternate in σ. We call them a *rainbow*. Our goal is to find rainbows of different block pairs that we can place onto the same page. Here, a key observation is that we will only need to know the top and the bottom edge of the rainbow to decide whether rainbows cross.

We branch on the set of all top and bottom edges of all rainbows for all block pairs and all pages in a hypothetical solution. The number of rainbows on any page of a hypothetical fixed-order stack layout is at most $2k$; this is because for any page contracting each rainbow on the page into an edge and then contracting each block into a vertex yields an outerplanar graph on at most k vertices and hence with $\leq 2k - -3$ edges. Hence overall we have $O(m^{4ks})$ branches. Note that we consider also all hypothetical solutions that use less than $(2k - -3)s$ rainbows since we may select edges multiple times, which corresponds to selecting a smaller set of top or bottom edges.

Our next step is to associate pairs of top and bottom edges for each rainbow and nest the remaining edges into appropriate rainbows. In each branch, we construct the *nesting digraph* N, whose vertices correspond to the edges of G and there is an arc between vertices corresponding to edges e and e' if e' nests inside e. We remove all incoming arcs from vertices that represent top edges and all outgoing arcs from vertices that represent bottom edges. By definition, N is acyclic which allows us to find a minimal path cover in time $O(|V(N)|^{5/2}) = O(m^{5/2})$. Each path in the resulting path cover will correspond to a rainbow. For correctness it is important to note that in this path cover, paths may connect vertices associated to top and bottom edges of different rainbows in a fixed hypothetical solution associated to the current branch. However, we can argue that such a hypothetical solution can be transformed into one that has the same pairs of top and bottom rainbow edges that are connected via our path cover, by a careful exchange argument.

Finally, we assign the rainbows to pages in a way in which they do not cross. For this we consider the *conflict graph* where the rainbows obtained in the previous step correspond to vertices, and edges connect vertices corresponding to rainbows that are of the same block pair or cross. These are precisely the rainbows that may not be placed onto the same page. A proper vertex coloring of the conflict graph with at most s colors immediately corresponds to a page assignment of the rainbows and hence yields a fixed-order stack layout on at most s pages for the computed set of rainbows. Since the conflict graph has $2sk$ vertices, we can find an s-coloring in time $O(2^{2sk}sk)$, or decide that it does not exist.

Correctness follows from our earlier 'key observation' and the fact that if there is a solution at some point we can assume to consider the same pairs of top and bottom edges for all rainbows. ◀

▶ **Theorem 2.** *There is an $O(m^{5/2})$-time $(k--1)$-approximation algorithm for determining the fixed-order stack number of a graph $G = (V, E)$ with respect to a given ordering σ of V, where $m = |E|$ and k is the number of blocks of σ.*

Proof sketch. Construct the (directed acyclic) compatibility graph C as follows. Add a vertex for each edge of G and add an arc if two edges can be placed onto the same page. Formally, for $\sigma = (v_1, v_2, \ldots, v_n)$, $e_i = v_{i_1}v_{i_2}$ and $e_j = v_{j_1}v_{j_2}$, there is an arc from e_i to e_j if $i_1 \leq j_1 < j_2 \leq i_2$ (i.e., e_j nests inside e_i) or if $i_1 < i_2 \leq j_1 < j_2$ (i.e., e_i and e_j form a *necklace*). Find a minimum path cover in C in $O(m^{5/2})$ time. Wherever a path uses a necklace arc, split the path into two. For each resulting path, use its own page.

Clearly, each resulting path is a rainbow and, thus, its edges can be placed onto the same page. Any path in C uses $\leq k--1$ necklace arcs. Since our initial path cover used $\leq \operatorname{fosn}(G, \sigma)$ paths (this follows from the fact that an optimal solution can be represented by $\operatorname{fosn}(G, \sigma)$ paths in C), we have $\leq (k--1)\operatorname{fosn}(G, \sigma)$ paths/rainbows after splitting. ◀

5.5 Progress on Applications of the Product Structure Theorems

Giordano Da Lozzo (University of Rome III, IT), Michael A. Bekos (Universität Tübingen, DE), Petr Hlinený (Masaryk University – Brno, CZ), and Michael Kaufmann (Universität Tübingen, DE)

Consider two graphs A and B. The *strong product* of A and B, denoted by $A \boxtimes B$, is the graph such that: (i) $V(A \boxtimes B) = V(A) \times V(B)$ and (ii) there exists an edge between the vertices $(a_1, b_1), (a_2, b_2) \in V(A \boxtimes B)$ if and only if one of the following occurs: (a) $a_1 = a_2$

and $b_1b_2 \in E(B)$, (b) $b_1 = b_2$ and $a_1a_2 \in E(A)$, or $a_1a_2 \in E(A)$ and $b_1b_2 \in E(B)$. Our research group considered strong products that yield supergraphs of k-planar graphs, which are defined as follows. A graph is *k-planar* if it admits a *k-planar drawing*, i.e., a drawing in the plane in which each edge is crossed at most k times. We exploit these products to derive new upper bounds on the queue number of k-planar graphs. Recall that, the *queue number* $\mathsf{qn}(G)$ of a graph G corresponds to the minimum size of the largest rainbow over all linear orderings of the vertices of G, where a *rainbow* is a set of independent nested edges.

In a recent preprint [3], Dujmović, Morin, and Wood have proved the following theorem on the structure of k-planar graphs.

▶ **Theorem 1.** *Every k-planar graph is a subgraph of the strong product of three graphs $H \boxtimes P \boxtimes K_{18k^2+48k+30}$, where H is a planar graph of treewidth at most $\binom{k+4}{3} - 1$ and P is a path.*

Furthermore, for the special case in which $k = 1$, the same authors proved the following stronger statement.

▶ **Theorem 2.** *Every 1-planar graph is a subgraph of the strong product of three graphs $H \boxtimes P \boxtimes K_{30}$, where H is a planar graph of treewidth at most 3 and P is a path.*

In [2], Dujmović et al. have proved the following useful lemma concerning the queue number of graphs that can be expressed as (subgraphs of) the strong product of three graphs exhibiting the properties of Theorems 1 and 2.

▶ **Lemma 3.** *If $G \subseteq H \boxtimes P \boxtimes K_\ell$ then $\mathsf{qn}(G) \leq 3\ell\, \mathsf{qn}(H) + \lfloor \frac{3}{2}\ell \rfloor$.*

Combining Lemma 3 and Theorem 1, Dujmović, Morin, and Wood showed the first constant upper bound on the queue number of k-planar graphs [3], thus resolving a long-standing open question. Furthermore, applying Lemma 3 and Theorem 2, they improved this bound to 495 for 1-planar graphs. We observe, in particular, that every improvement to the 'K_{30}' term of Theorem 2, would immediately improve the bound on the queue number of 1-planar graphs, as well as other related results.

In this working group, we researched in the direction of improving Theorem 2 for 1-planar graphs. We also investigated a possible generalization of these ideas to k-planar graphs for $k \geq 2$. In this regard, we considered the family of optimal 2-planar graphs, and more in general the larger graph family of h-framed graphs: An *h-framed* graph is a connected graph that admits a drawing in the plane such that the removal of all its crossed edges yields a bi-connected plane graph with faces of size at most h. This graph family was first introduced by Bekos et al. [1], in the context of the counterpart of queue layouts, called *stack layouts*. We remark that the family of h-framed graphs strictly contains the ones of triconnected 1-planar and optimal 2-planar graphs, as the graphs in these families can be augmented to 4-framed and 5-framed graphs, respectively [1].

We state below, without proving, our main claims.

- By carefully redesigning some of the arguments from [3], namely the choices that determine the value ℓ of the graph K_ℓ in Lemma 3, we believe it is possible to improve Theorem 2 to the product $H \boxtimes P \boxtimes K_7$. By Lemma 3, this would immediately improve the upper bound on the queue number of 1-planar graphs to 115.
- Essentially the same improved approach leads to a result that any h-framed graph is a subgraph of the strong product of three graphs $H \boxtimes P \boxtimes K_{O(h)}$. In particular, an upper bound on the size of the complete graph involved in the product seems to be $\lfloor \frac{5h}{2} \rfloor - 3$;

although, several details have to be worked out. Therefore, for h-framed graphs, we obtain the first non-trivial upper bound on the queue number of the graphs in this family that only depends on h.
- Further investigations led to new ideas which could probably improve Theorem 1 for 2-planar graphs to a product $H \boxtimes P \boxtimes K_{33}$. However, this is the subject of ongoing research.
- We provide a new strong product theorem for 1-planar graphs of the form $H \boxtimes P^2 \boxtimes K_3$, where H is a planar graph of treewidth at most 3 and P^2 is the square of a path P. This, in turn, leads to a further improvement of the queue number of 1-planar graphs.

All these informal claims are yet to be written down with proper proofs and verified (at the time of writing up this report).

References

1 M. A. Bekos, G. Da Lozzo, S. Griesbach, M. Gronemann, F. Montecchiani, C. N. Raftopoulou: Book Embeddings of Nonplanar Graphs with Small Faces in Few Pages. *SoCG 2020*: 16:1-16:17
2 V. Dujmović, G. Joret, P. Micek, P. Morin: Planar Graphs Have Bounded Queue-Number. *J. ACM*,67(4): 22:1-22:38 (2020).
3 V. Dujmović, P. Morin, and D. Wood: Graph product structure for non-minor-closed classes. *CoRR* abs/1907.05168 (2020).

5.6 Progress on the Parameterized Complexity of Small Decision Tree Learning

Stephen G. Kobourov (University of Arizona – Tucson, US), Maarten Löffler (Utrecht University, NL), Fabrizio Montecchiani (University of Perugia, IT), Raimund Seidel, Ignaz Rutter (Universität Passau, DE), Manuel Sorge (TU Wien, AT), and Jules Wulms (TU Wien, AT)

Decision Trees are well known tools used to describe, classify, and generalize data. Besides their simplicity, decision trees are particularly attractive for providing interpretable models of the underlying data.

The setting is as follows (see Figure 10). The task is to classify an *example set*, a set $E \subseteq \mathbb{R}^d$ together with a function $\lambda\colon E \to \{\oplus, \ominus\}$ labeling each example with a *class*. For this task, a *decision tree* is a rooted binary tree T together with two functions $\mathsf{dim}\colon V(T) \to [d]$ and $\mathsf{thr}\colon V(T) \to \mathbb{R}$ that label each inner node $t \in V(T)$ by a *dimension* $\mathsf{dim}(t) \in [d]$ and a *threshold* $\mathsf{thr}(t) \in \mathbb{R}$. For each inner node t of T the edges to the two children of t are labeled by yes and no. Each node $t \in V(T)$, including the leaves, defines a subset $E[T,t] \subseteq E$ as follows: Consider the path P from the root of T to t. An example $e \in E$ is in $E[T,t]$ if and only if, for each node v on P, it holds that $e[\mathsf{dim}(v)] \leq \mathsf{thr}(v)$ if the edge following v on P is yes and it holds $e[\mathsf{dim}(v)] > \mathsf{thr}(v)$ if the edge following v on P is no. If the tree T is clear from the context, we simplify $E[T,t]$ to $E[t]$.

A decision tree T is a *decision tree for* (E,λ) if for each leaf ℓ of T we have that all examples in $E[\ell]$ have the same label under λ. The *size* of a decision tree is the number of its inner nodes.

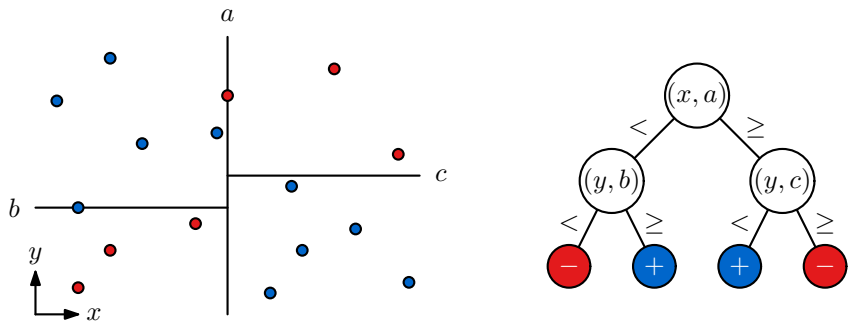


Figure 10 An instance (E, λ) of DTS with a minimum decision tree T for (E, λ). Examples labeled $\oplus$ and $\ominus$ by λ are blue and red respectively. Each internal node $t \in T$ is labeled by $(\mathsf{dim}(t), \mathsf{thr}(t))$.

In this working group we studied the complexity of the problem Minimum Decision Tree Size (DTS). The input consists of an integer s and an example set (E, λ). The task is to decide whether there exists a decision tree for (E, λ) that has size at most s. Our aim was to solve the open question about the parameterized complexity of DTS with respect to the number d of dimensions of the example space, posed by Manuel Sorge in the same seminar.

We did not feel comfortable to immediately and directly attack the open question and first explored the behavior of the problem in particular with respect to other well-motivated parameters. Besides the number d of features (or dimensions) and the natural size-parameter s of the decision tree, additional interesting parameters are the maximum number of features (or dimensions) on which any two examples differ δ_{max}, and the maximum number of different values a feature ranges over (the domain size) D_{max}. Ordyniak and Szeider [1] proved that DTS parameterized by s is W[2]-hard already when each feature is binary, and hence DTS is W[2]-hard also when parameterized by $s + D_{max}$. Moreover, the same reduction shows that the problem is paraNP-hard when parameterized by $\delta_{max} + D_{max}$. On the positive side, DTS parameterized by s lies in XP. The main positive result in Ref. [1] establishes an FPT algorithm for DTS parameterized by $s + \delta_{max} + D_{max}$. It is open whether the problem is FPT parameterized by $s + \delta_{max}$.

We first observed that some small adaptions of an algorithm of Ordyniak and Szeider [1] shows that DTS is fixed-parameter tractable when parameterized by $s+d$. On the other hand, the hardness result in Ref. [1] shows that the DTS is paraNP-hard also when parameterized by the minimum number of examples in one of the two classes (called r). Indeed, the reduction is such that one of the two classes contains only one example. However, the number d of dimensions in the reduction is unbounded. A natural question is therefore whether DTS parameterized by $d+r$ is FPT. We answered this question in the positive, and generalized the result to a more general parameterization, namely $d + R$, where R is the minimum number of leaves of the same class.

We explored various directions regarding the relation of r and δ_{max}, the development of efficient data-reduction rules, and the structure of the hypergraph defined by sets of dimensions in which pairs of examples differ. These directions did not bear tangible results, except for counterexamples for intuitive statements such as "if an example is directly between examples of the same class in all dimensions, then it can be safely removed". Exploring these directions did however provide us with much better intuition about the behavior of the problem. Ultimately, this lead to a construction that with high confidence can be used

to show that DTS is W[1]-hard with respect to the number d of dimensions. This solves the open question and complements the fact that DTS can be solved in $O(D_{max}^{2d+1}d)$ time (personal communication with Marcin Pilipczuk).

References

1 S. Ordyniak and S. Szeider. Parameterized complexity of small decision tree learning. In *Proceedings of the 35th AAAI Converence on Artificial Intelligence (AAAI '21)*, pages 6454–6462. AAAI Press, 2021.

Participants

- Michael A. Bekos
Universität Tübingen, DE
- Steven Chaplick
Maastricht University, NL
- Giordano Da Lozzo
University of Rome III, IT
- Emilio Di Giacomo
University of Perugia, IT
- Walter Didimo
University of Perugia, IT
- Fabrizio Frati
University of Rome III, IT
- Robert Ganian
TU Wien, AT
- Martin Gronemann
Universität Osnabrück, DE
- Thekla Hamm
TU Wien, AT
- Petr Hlinený
Masaryk University – Brno, CZ
- Michael Kaufmann
Universität Tübingen, DE
- Philipp Kindermann
Universität Trier, DE
- Boris Klemz
Universität Würzburg, DE
- Stephen G. Kobourov
University of Arizona – Tucson, US
- Giuseppe Liotta
University of Perugia, IT
- Maarten Löffler
Utrecht University, NL
- Fabrizio Montecchiani
University of Perugia, IT
- Martin Nöllenburg
TU Wien, AT
- Chrysanthi Raftopoulou
National Technical University of Athens, GR
- Ignaz Rutter
Universität Passau, DE
- Kirill Simonov
TU Wien, AT
- Manuel Sorge
TU Wien, AT
- Birgit Vogtenhuber
TU Graz, AT
- Alexander Wolff
Universität Würzburg, DE
- Jules Wulms
TU Wien, AT
- Johannes Zink
Universität Würzburg, DE

Remote Participants

- Meirav Zehavi
Ben-Gurion University – Beer Sheva, IL
- Siddharth Gupta
Ben-Gurion University – Beer Sheva, IL

Report from Dagstuhl Seminar 21301

Matching Under Preferences: Theory and Practice

Edited by

Haris Aziz[1], Péter Biró[2], Tamás Fleiner[3], and Bettina Klaus[4]

1 **UNSW – Sydney, AU,** haris.aziz@unsw.edu.au
2 **Hungarian Academy of Sciences – Budapest, HU,** biro.peter@krtk.mta.hu
3 **Budapest University of Technology & Economics, HU,** fleiner@cs.elte.hu
4 **University of Lausanne, CH,** bettina.klaus@unil.ch

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 21301 "Matching Under Preferences: Theory and Practice". The seminar featured a mixture of technical scientific talks, survey talks, open problem presentations, working group sessions, five-minute contributions ("rump session"), and a panel discussion. This was the first Dagstuhl seminar that was dedicated to matching under preferences.

Seminar July 25–30, 2021 – http://www.dagstuhl.de/21301
2012 ACM Subject Classification Theory of computation → Algorithmic game theory and mechanism design; Applied computing → Economics; Mathematics of computing → Graph theory
Keywords and phrases market design, matching under preferences, matching with distributional constraints, organ exchange, stable matching
Digital Object Identifier 10.4230/DagRep.11.7.124
Edited in cooperation with Özbilen, Seçkin

1 Executive Summary

Haris Aziz
Péter Biró
Tamás Fleiner
Bettina Klaus

Matching under preferences is a general field spanning computer science, economics, and mathematics. The seminal paper in the field is one by Gale and Shapley (1962) that launched an algorithmic approach to matching agents with preferences. The central problems in the field involve matching agents to each other and to resources in a stable and efficient manner. Matching market algorithms based on the preferences of the agents have several applications such as in school admissions, placement of hospital residents, and centralized kidney markets. Topics in the field include two-sided matchings involving agents on both sides (e.g., job markets, school choice, etc.); two-sided matchings involving agents and items (e.g. course allocation, project allocation, assigning papers to reviewers etc.); one-sided matchings (roommates problem, kidney exchanges, etc.); and matching with payments (assignment game, auctions, etc.).

Matching Under Preferences: Theory and Practice, *Dagstuhl Reports*, Vol. 11, Issue 06, pp. 124–146
Editors: Haris Aziz, Péter Biró, Tamás Fleiner, and Bettina Klaus

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

The topic of matching under preferences not only has tremendous applications but is based on a deep mathematical theory that has been developed by multiple research communities including theoretical computer science, artificial intelligence, discrete mathematics, game theory, and microeconomics. One of the main purposes of the seminar was to bring together leading researchers from various communities working on the topic and facilitate collaboration. The participant list was a mixture of researchers from computer science, mathematics, and economics. The seminar provided a platform to discuss state of the art in matching under preferences; identify new and exciting applications of developing research; and understand the mathematical and algorithmic requirements of new and upcoming problems in the field.

The seminar was conducted in a hybrid manner, with 15 participants attending the seminar physically from the Dagstuhl center and 34 participants attending online. The hybrid nature of the work required the need for careful planning to keep participants engaged and to facilitate collaboration between off-site and on-site participants. The online participation was managed via zoom and gather-town softwares.

The four main focus topics of the workshop were the following ones.

1. Matching markets with distributional constraints,
2. Probabilistic and Fractional Matching,
3. Matching in online and dynamic settings, and
4. Matching Markets and machine learning.

All of the four focus areas are important directions for the field. As new applications arise, it is clear that many real-life matching markets have additional feasibility and distributional constraints. Secondly, most of the theoretical developments in the field concern deterministic outcomes, so one of the goals was to make further progress on probabilistic mechanisms. During the seminar, current and new research on probabilistic approaches was discussed. Thirdly, many practical matching markets have online and dynamic aspects. There were several discussions on how to model and solve online matching problems. Fourthly, with the increased importance of machine learning in building computer systems, the seminar provided an opportunity to discuss how learning approaches help solve market design problems.

On each of the first four days, there was a one-hour survey talk given by an expert on the above topics. On the first day, Yash Kanoria presented a survey on “online matching markets”. On the second day, John Dickerson gave a survey talk on machine learning and matching markets. On day three, Makoto Yokoo presented an overview of “matching under constraints.” On day four, Jay Sethuraman surveyed “probabilistic matchings.”

On each of the days there were several shorter scientific presentations. During the workshop, two rump sessions were organized to facilitate different time zones. The rump sessions gave an opportunity to the participants to give brief updates or share open problems.

During the week, there were several time slots dedicated to flexible discussion and collaboration as well as dedicated working groups working on particular research topics. Apart from collaborations in smaller groups, the workshop witnessed major collaboration or discussion around several topics. Robert Bredereck brought up the issue of unifying and streamlining terminology and discussed the issue of using gender-neutral terms. There was a large working group led by Sushmita Gupta and Pallavi Jain that examined computational problems that combine the goals of stability and popularity. There was a group led by Bettina Klaus on lexicographic preferences in matching and market design. Finally, Florian Brandl led a group on the intersection between matching and fair division.

On the last day, there was a panel discussion that was moderated by Haris Aziz and Bettina Klaus. The discussants in the panel were Péter Biró, Ágnes Cseh, Lars Ehlers, Alex Teytelboym, and Utku Ünver. The main topics discussed included ways to build synergies between research communities and having an impact on the practice of matching markets.

The organisers thank all the Dagstuhl staff members for their professional support, the participants for enriching the seminar, Somouaoga Bonkoungou and Alex Lam for providing video conferencing support, and Seçkin Özbilen for his support in putting together the abstracts that compose this report.

2 Table of Contents

Executive Summary
Haris Aziz, Péter Biró, Tamás Fleiner, and Bettina Klaus . . . 124

Overview of Talks

Decentralized Matching in a Probabilistic Environment
Irene Lo . . . 129

The Vigilant Eating Rule: A General Approach for Probabilistic Economic Design with Constraints
Haris Aziz and Florian Brandl . . . 129

Cutoff stability under distributional constraints with an application to summer internship matching
Péter Biró, Anton Baychkov, and Haris Aziz . . . 130

Stable roommates with narcissistic, single-peaked, and single-crossing preferences
Robert Bredereck and Jiehua Chen . . . 130

Fractional Matchings under Preferences: Stability and Optimality
Jiehua Chen . . . 131

Group Fairness in Social Service Allocation
Sanmay Das . . . 131

Machine Learning for Mechanism Design:A Short Intro to Differentiable Economics
John Dickerson . . . 132

Robust Minimal Instability of the Top Trading Cycles Mechanism
Lars Ehlers and Battal Dogan . . . 132

A Parameterized Complexity Analysis of Incremental Stable Matching
Klaus Heeger . . . 133

Accomplice Manipulation of the Deferred Acceptance Algorithm
Hadi Hosseini . . . 133

Stable partitions for proportional generalized claims problems
Bettina Klaus . . . 134

Strict Core and Strategy-Proofness for Hedonic Games with Friend-Oriented Preferences
Bettina Klaus, Flip Klijn, and Seçkin Özbilen . . . 134

Minimal-Access Rights in School Choice and the Deferred Acceptance Mechanism
Flip Klijn and Bettina Klaus . . . 135

Core-Stability in Assignment Markets with Financially Constrained Buyers
Martin Bichler . . . 135

Behavioral Stable Marriage Problems
Nicholas Mattei . . . 136

Quick presentation of the French college admission procedure
Simon Mauras . . . 136

Almost Stable Marriage
Sushmita Gupta and Pallavi Jain . . . 137

Allocation with Weak Priorities and General Constraints
Thanh Nguyen 137

Survey on online matching markets
Peng Shi 138

Reallocation with Priorities
Jan Christoph Schlegel 138

Is it worth sprucing up your home?
Ildikó Schlotter, Péter Biró, and Tamás Fleiner 138

Fractional and Probabilistic Matching: a brief overview
Jay Sethuraman 139

Matching and Prices
Alexander Teytelboym and Ravi Jagadeesan 139

Blood Allocation with Replacement Donors: A Theory of Multi-unit Exchange with Compatibility-based Preferences
Utku Ünver 140

Stability in Large Markets
Karolina Lena Johanna Vocke 140

Mechanisms for Facility Location with Capacity Limits
Toby Walsh 141

Approximability vs. Strategy-proofness in Stable Matching Problems with Ties
Yu Yokoi and Shuichi Miyazaki 141

Survey on Constrained Matching
Makoto Yokoo 142

Absolutely and simply popular rankings
Ágnes Cseh 142

Kidney Exchange progress in Germany
Ágnes Cseh 143

Working groups

Gender Terminology in Bipartite Stable Matching
Robert Bredereck 143

Popular Matching with few blocking pairs
Sushmita Gupta, Ágnes Cseh, Pallavi Jain, Baharak Rastegari, Ildikó Schlotter, and Kavitha Telikepalli 144

Lexicographic preferences in matching and market design
Bettina Klaus 144

Participants 145

Remote Participants 145

3 Overview of Talks

3.1 Decentralized Matching in a Probabilistic Environment

Irene Lo (Stanford University, US)

Joint work of Amin Saberi, Irene Lo, Mobin Y. Jeloudar, Tristan Pollner
Main reference Mobin Y. Jeloudar, Irene Lo, Tristan Pollner, Amin Saberi: "Decentralized Matching in a Probabilistic Environment", CoRR, Vol. abs/2106.06706, 2021.
URL https://arxiv.org/abs/2106.06706

We consider a model for repeated stochastic matching where compatibility is probabilistic, is realized the first time agents are matched, and persists in the future. Such a model has applications in the gig economy, kidney exchange, and mentorship matching.

We ask whether a *decentralized* matching process can approximate the optimal online algorithm. In particular, we consider a decentralized *stable matching* process where agents match with the most compatible partner who does not prefer matching with someone else, and known compatible pairs continue matching in all future rounds. We demonstrate that the above process provides a 0.316-approximation to the optimal online algorithm for matching on general graphs. We also provide a $1/7$-approximation for many-to-one bipartite matching, a $1/11$-approximation for capacitated matching on general graphs, and a $1/2k$-approximation for forming teams of up to k agents. Our results rely on a novel coupling argument that decomposes the successful edges of the optimal online algorithm in terms of their round-by-round comparison with stable matching.

3.2 The Vigilant Eating Rule: A General Approach for Probabilistic Economic Design with Constraints

Haris Aziz (UNSW – Sydney, AU) and Florian Brandl (Universität Bonn, DE)

Main reference Haris Aziz, Florian Brandl: "The Vigilant Eating Rule: A General Approach for Probabilistic Economic Design with Constraints", CoRR, Vol. abs/2008.08991, 2020.
URL https://arxiv.org/abs/2008.08991

We consider the problem of probabilistic allocation of objects under ordinal preferences. We devise an allocation mechanism, called the vigilant eating rule (VER), that applies to nearly arbitrary feasibility constraints. It is constrained ordinally efficient, can be computed efficiently for a large class of constraints, and treats agents equally if they have the same preferences and are subject to the same constraints. When the set of feasible allocations is convex, we also present a characterization of our rule based on ordinal egalitarianism. Our results about VER do not just apply to allocation problems but to all collective choice problems in which agents have ordinal preferences over discrete outcomes. As a case study, we assume objects have priorities for agents and apply VER to sets of probabilistic allocations that are constrained by stability. VER coincides with the (extended) probabilistic serial rule when priorities are flat and the agent proposing deterministic deferred acceptance algorithm when preferences and priorities are strict. While VER always returns a stable and constrained efficient allocation, it fails to be strategyproof, unconstrained efficient, and envy-free. We show, however, that each of these three properties is incompatible with stability and constrained efficiency.

3.3 Cutoff stability under distributional constraints with an application to summer internship matching

Péter Biró (Hungarian Academy of Sciences – Budapest, HU), Anton Baychkov, Haris Aziz (UNSW – Sydney, AU)

Main reference Haris Aziz, Anton Baychkov, Péter Biró: "Cutoff stability under distributional constraints with an application to summer internship matching", CoRR, Vol. abs/2102.02931, 2021.
URL https://arxiv.org/abs/2102.02931

We introduce a new two-sided stable matching problem that describes the summer internship matching practice of an Australian university. The model is a case between two models of Kamada and Kojima on matchings with distributional constraints. We study three solution concepts, the strong and weak stability concepts proposed by Kamada and Kojima, and a new one in between the two, called cutoff stability. Kamada and Kojima showed that a strongly stable matching may not exist in their most restricted model with disjoint regional quotas. Our first result is that checking its existence is NP-hard. We then show that a cutoff stable matching exists not just for the summer internship problem but also for the general matching model with arbitrary heredity constraints. We present an algorithm to compute a cutoff stable matching and show that it runs in polynomial time in our special case of summer internship model. However, we also show that finding a maximum size cutoff stable matching is NP-hard, but we provide a Mixed Integer Linear Program formulation for this optimisation problem.

3.4 Stable roommates with narcissistic, single-peaked, and single-crossing preferences

Robert Bredereck (HU Berlin, DE), Jiehua Chen (TU Wien, AT)

Joint work of Robert Bredereck, Jiehua Chen, Ugo Paavo Finnendahl, Rolf Niedermeier
Main reference Robert Bredereck, Jiehua Chen, Ugo Paavo Finnendahl, Rolf Niedermeier: "Stable roommates with narcissistic, single-peaked, and single-crossing preferences", Auton. Agents Multi Agent Syst., Vol. 34(2), p. 53, 2020.
URL https://doi.org/10.1007/s10458-020-09470-x

The classical Stable Roommates problem is to decide whether there exists a matching of an even number of agents such that no two agents which are not matched to each other would prefer to be with each other rather than with their respectively assigned partners. We investigate Stable Roommates with complete (i.e., every agent can be matched with any other agent) or incomplete preferences, with ties (i.e., two agents are considered of equal value to some agent) or without ties. It is known that in general allowing ties makes the problem NP-complete. We provide algorithms for Stable Roommates that are, compared to those in the literature, more efficient when the input preferences are complete and have some structural property, such as being narcissistic, single-peaked, and single-crossing. However, when the preferences are incomplete and have ties, we show that being single-peaked and single-crossing does not reduce the computational complexity – Stable Roommates remains NP-complete.

References

1 Bredereck, Robert; Chen, Jiehua; Finnendahl, Ugo Paavo; and Niedermeier, Rolf. Stable roommate with narcissistic, single-peaked, and single-crossing preferences. In *Proc. of ADT '17*, volume 10576 of *LNCS*, pages 315–330. Springer, 2017.

2 Bredereck, Robert; Chen, Jiehua; Finnendahl, Ugo Paavo; and Niedermeier, Rolf. Stable roommates with narcissistic, single-peaked, and single-crossing preferences. *Auton. Agents Multi Agent Syst.*, 34(2):53, 2020.

3.5 Fractional Matchings under Preferences: Stability and Optimality

Jiehua Chen (TU Wien, AT)

Joint work of Jieua Chen, Sanjukta Roy, Manuel Sorge

We study generalizations of stable matching in which agents may be matched fractionally; this models time-sharing assignments. We focus on the so-called ordinal stability and cardinal stability, and investigate the computational complexity of finding an ordinally stable or cardinally stable fractional matching which either maximizes the social welfare (i.e., the overall utilities of the agents) or the number of fully matched agents (i.e., agents whose matching values sum up to one). We complete the complexity classification of both optimization problems for both ordinal stability and cardinal stability, distinguishing between the marriage (bipartite) and roommates (non-bipartite) cases and the presence or absence of ties in the preferences. In particular, we prove a surprising result that finding a cardinally stable fractional matching with maximum social welfare is NP-hard even for the marriage case without ties. This answers an open question and exemplifies a rare variant of stable marriage that remains hard for preferences without ties. We also complete the picture of the relations of the stability notions and derive structural properties.

3.6 Group Fairness in Social Service Allocation

Sanmay Das (George Mason University – Fairfax, US)

Joint work of Sanmay Das, Tasfia Mashiat, Xavier Gitiaux, Patrick J. Fowler, Huzefa Rangwala

Motivated by allocation of different types of housing resources to those experiencing homelessness, we consider how to measure the fairness of different allocation rules for different subpopulations. We note how distributional differences in utilities/costs across subpopulations as well as different ways of measuring fairness may affect perceptions of allocation fairness.

3.7 Machine Learning for Mechanism Design:A Short Intro to Differentiable Economics

John Dickerson (University of Maryland – College Park, US)

Joint work of John Dickerson, Michael Curry, Samuel Dooley, Ping-yeh Chiang, Uro Lyi, Neehar Peri, Anthony Ostuni, Elizabeth Horishny, Tom Goldstein
Main reference Neehar Peri, Michael J. Curry, Samuel Dooley, John P. Dickerson: "PreferenceNet: Encoding Human Preferences in Auction Design with Deep Learning", CoRR, Vol. abs/2106.03215, 2021.
URL https://arxiv.org/abs/2106.03215

The design of revenue-maximizing auctions with strong incentive guarantees is a core concern of economic theory. Computational auctions enable online advertising, sourcing, spectrum allocation, and myriad financial markets. Analytic progress in this space is notoriously difficult; since Myerson's 1981 work characterizing single-item optimal auctions, there has been limited progress outside of restricted settings. A recent paper by Dütting et al. circumvents analytic difficulties by applying deep learning techniques to, instead, approximate optimal auctions. Their RegretNet architecture can represent auctions with arbitrary numbers of items and participants; it is trained to be empirically strategyproof, but the property is never exactly verified leaving potential loopholes for market participants to exploit. In parallel, new research from Ilvento et al. and other groups has developed notions of fairness in the context of auction design. Inspired by these advances, in this talk, we discuss extensions of these techniques for approximating auctions using deep learning to address concerns of

- fairness while maintaining high revenue and strong incentive guarantees;
- certified robustness, that is, verification of claimed strategyproofness of deep learned auctions; and
- expressiveness via different demand functions and other constraints.

To enable that last point, we propose a new architecture to learn incentive compatible, revenue-maximizing auctions from sampled valuations, which uses the Sinkhorn algorithm to perform a differentiable bipartite matching. Our new framework allows the network to learn strategyproof revenue-maximizing mechanisms in settings not learnable by the previous RegretNet architecture. This talk connects work in the deep learning for auction design space into the deep learning for **matching market** design space, and provides concrete steps forward regarding differentiable economics and matching market design.

This talk covers hot-off-the-presses work led by: PhD students Michael Curry, Ping-yeh Chiang, and Samuel Dooley; and undergraduate students Elizabeth Horishny, Kevin Kuo, Uro Lyi, Anthony Ostuni, Neehar Peri; and Tom Goldstein at UMD. Papers have appeared at AI/ML conferences or are currently under review; please check arXiv or get in touch for drafts.

3.8 Robust Minimal Instability of the Top Trading Cycles Mechanism

Lars Ehlers (University of Montreal, CA) and Battal Dogan

URL www.battaldogan.com

In the context of priority-based allocation of objects, we formulate methods to compare assignments in terms of their stability. We introduce three basic properties that a reasonable stability comparison should satisfy. We show that, for any stability comparison satisfying the

three properties, the top trading cycles mechanism is minimally unstable among efficient and strategy-proof mechanisms when objects have unit capacities. Our unifying approach covers basically all natural stability comparisons and establishes the robustness of a recent result by Abdulkadiroglu et al. (2020). When objects have non-unit capacities, we characterize the capacity-priority structures for which our result is preserved.

3.9 A Parameterized Complexity Analysis of Incremental Stable Matching

Klaus Heeger (TU Berlin, DE)

Joint work of Niclas Boehmer, Klaus Heeger, Rolf Niedermeier

When computing stable matchings, it is usually assumed that the set of participants in the matching market as well as their preferences is fixed. However, in reality, these may change over time (e.g. when considering the assignment of children to schools, children may leave the market because their family moves to another city). Consequently, an initially stable matching may become unstable over time. Then, a natural goal is to find a stable matching which is as close as possible to the initial one. This problem was introduced as Incremental Stable Matching by Bredereck, Chen, Knop, Luo, and Niedermeier [1]. As they showed that this problem is NP-complete in the roommates setting, we consider its parameterized complexity in ongoing work. Among our results we answer two open questions from Bredereck et al. [1], showing that Incremental Stable Roommates is W[1]-hard parameterized by the number of changes in the preferences (but admits an XP-algorithm with respect to this parameter) and that Incremental Weakly Stable Marriage with Ties is W[1]-hard parameterized by the number of ties. Furthermore, we give FPT-algorithms for two parameters measuring the similarity of the agent's preferences to each other.

References

1 Robert Bredereck, Jiehua Chen, Dušan Knop, Junjie Luo, and Rolf Niedermeier. *Adapting Stable Matchings to Evolving Preferences*. In: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, pp. 1830–1837, 2020.

3.10 Accomplice Manipulation of the Deferred Acceptance Algorithm

Hadi Hosseini (Pennsylvania State University, US)

Joint work of Hadi Hosseini, Fatima Umar, Rohit Vaish
Main reference Hadi Hosseini, Fatima Umar, Rohit Vaish: "Accomplice Manipulation of the Deferred Acceptance Algorithm", in Proc. of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021, pp. 231–237, ijcai.org, 2021.
URL https://doi.org/10.24963/ijcai.2021/33

The deferred acceptance algorithm is an elegant solution to the stable matching problem that guarantees optimality and truthfulness for one side of the market. Despite these desirable guarantees, it is susceptible to strategic misreporting of preferences by the agents on the other side. We study a novel model of strategic behavior under the deferred acceptance algorithm:

manipulation through an accomplice. Here, an agent on the proposed-to side (say, a woman) partners with an agent on the proposing side – an accomplice – to manipulate on her behalf (possibly at the expense of worsening his match). We show that the optimal manipulation strategy for an accomplice comprises of promoting exactly one woman in his true list (i.e., an inconspicuous manipulation). This structural result immediately gives a polynomial-time algorithm for computing an optimal accomplice manipulation. We also study the conditions under which the manipulated matching is stable with respect to the true preferences. Our experimental results show that accomplice manipulation outperforms self manipulation both in terms of the frequency of occurrence as well as the quality of matched partners.

3.11 Stable partitions for proportional generalized claims problems

Bettina Klaus (University of Lausanne, CH)

Joint work of Bettina Klaus, Oihane Gall

We consider a set of agents, e.g., a group of researchers, who have claims on an endowment, e.g., a research budget from a national science foundation. The research budget is not large enough to cover all claims. Agents can form coalitions and coalitional funding is proportional to the sum of the claims of its members, except for singleton coalitions which do not receive any funding. We analyze the structure of stable partitions when coalition members use well-behaved rules to allocate coalitional endowments, e.g., the well-known constrained equal awards rule (CEA) or the constrained equal losses rule (CEL).

For continuous, (strictly) resource monotonic, and consistent rules, stable partitions with (mostly) pairwise coalitions emerge. For CEA and CEL we provide algorithms to construct such a stable pairwise partition. While for CEL the resulting stable pairwise partition is assortative and sequentially matches lowest claims pairs, for CEA the resulting stable pairwise partition is obtained sequentially by matching in each step either a highest claims pair or a highest-lowest claims pair.

More generally, we can also assume that the minimal coalition size to have a positive endowment is larger or equal to two. We then show how all results described above are extended to this general case.

3.12 Strict Core and Strategy-Proofness for Hedonic Games with Friend-Oriented Preferences

Bettina Klaus (University of Lausanne, CH), Flip Klijn (CSIC – Barcelona, ES), and Seçkin Özbilen (University of Lausanne, CH)

We consider hedonic coalition formation problems with friend-oriented preferences; that is, each agent has preferences over coalitions she is part of based on a partition of the set of other agents into friends and enemies. We assume that for each of her coalitions, (1) adding an enemy makes her strictly worse off, (2) adding a friend together with a set of enemies makes her strictly better off, and (3) adding a friend makes her strictly better off than

losing a set of enemies. We show that the partition associated with the strongly connected components (SCC) of the so-called friend-oriented preference graph is in the strict core. The SCC mechanism, which assigns the SCC partition to each hedonic coalition formation problem with friend-oriented preferences, is group strategy-proof. Furthermore, the SCC mechanism is the only mechanism that satisfies strategy-proofness and strict core stability.

3.13 Minimal-Access Rights in School Choice and the Deferred Acceptance Mechanism

Flip Klijn (CSIC – Barcelona, ES) and Bettina Klaus (University of Lausanne, CH)

Main reference Bettina Klaus, Flip Klijn: "Minimal-Access Rights in School Choice and the Deferred Acceptance Mechanism" BSE Working Paper: 1264, June 2021
URL https://www.barcelonagse.eu/research/working-papers/minimal-access-rights-school-choice-and-deferred-acceptance-mechanism

A classical school choice problem consists of a set of schools with priorities over students and a set of students with preferences over schools. Schools' priorities are often based on multiple criteria, e.g., merit-based test scores as well as minimal-access rights (siblings attending the school, students' proximity to the school, etc.). Traditionally, minimal-access rights are incorporated into priorities by always giving minimal-access students higher priority over non-minimal-access students. However, stability based on such adjusted priorities can be considered unfair because a minimal-access student may be admitted to a popular school while another student with higher merit-score but without minimal-access right is rejected, even though the former minimal-access student could easily attend another of her minimal-access schools.

We therefore weaken stability to minimal-access stability: minimal-access rights only promote access to at most one minimal-access school. Apart from minimal-access stability, we also would want a school choice mechanism to satisfy strategy-proofness and minimal-access monotonicity, i.e., additional minimal-access rights for a student do not harm her. Our main result is that the student-proposing deferred acceptance mechanism is the only mechanism that satisfies minimal-access stability, strategy-proofness, and minimal-access monotonicity. Since this mechanism is in fact stable, our result can be interpreted as an impossibility result: fairer outcomes that are made possible by the weaker property of minimal-access stability are incompatible with strategy-proofness and minimal-access monotonicity.

3.14 Core-Stability in Assignment Markets with Financially Constrained Buyers

Martin Bichler (TU München, DE)

Joint work of Eleni Batziou, Martin Bichler, Maximilian Fichtl

We consider auctions of indivisible items to unit-demand bidders with budgets. Without financial constraints and pure quasilinear bidders, this assignment model allows for a simple ascending auction format that maximizes welfare and is incentive-compatible and core-stable.

Introducing budget constraints, the ascending auction requires strong additional conditions on the unit-demand preferences to maintain its properties. We show that without these conditions, there does not exist an incentive-compatible and core-stable mechanism. Even if bidders reveal their valuations and budgets truthfully, the allocation and pricing problem becomes an NP-hard optimization problem. The analysis complements complexity results for more complex valuations and raises doubts on the efficiency of simple auction designs in the presence of financially constrained buyers.

3.15 Behavioral Stable Marriage Problems

Nicholas Mattei (Tulane University – New Orleans, US)

Joint work of Nicholas Mattei, Andrea Martin, Brent Venable
Main reference Andrea Martin, Nicholas Mattei, Brent Venable: "Behavioral Stable Marriage Problems", Presented at the 3rd Games, Agents, and Incentives Workshop @ AAMAS 2021.
URL https://preflib.github.io/gaiw2021/papers/GAIW_2021_paper_33.pdf

The stable marriage problem (SMP) is a mathematical abstraction of two-sided matching markets with many practical applications. Several preference models have been considered in the context of SMPs including orders with ties, incompleteness, and uncertainty, but none have yet captured behavioral aspects of human decision making such as contextual effects. We introduce Behavioral Stable Marriage Problems (BSMPs), bringing together the formalism of matching with cognitive models of decision making to account for multi-attribute, non-deterministic preferences and to study the impact of well known behavioral deviations from rationality on two core notions of SMPs: stability and fairness. We analyze the computational complexity of several related problems, show that proposal-based approaches are affected by contextual effects and propose and evaluate novel ILP and local-search-based methods to efficiently find optimally stable and fair matchings.

3.16 Quick presentation of the French college admission procedure

Simon Mauras (University Paris Diderot, FR)

Each year in France, around 800 000 high-school students apply to the centralized college admission procedure. In 2018, the new platform, called Parcoursup, was launched. The main novelty of procedure is that students do not have to order their applications. Instead, the platform run the school proposing deferred acceptance mechanism, where students answer queries online and have a few days to chose which application they keep each time they receive multiple offers.

The goal of this informal talk was to present the upsides and downsides of this new mechanism. On the positive side, seats vacated by students leaving the market can be filled quickly by the online procedure; and the fact that students do not have to order applications can decrease self-censorship. On the negative side, the speed of convergence of the procedure becomes of paramount importance, and can be the cause of strategic and non-truthful behaviors from colleges and students.

3.17 Almost Stable Marriage

Sushmita Gupta (The Institute of Mathematical Sciences – Chennai, IN), Pallavi Jain (Indian Institute of Techology, IN)

Joint work of Meirav Zehavi, Pallavi Jain, Saket Saurabh, Sanjukta Roy, and Sushmita Gupta
Main reference Sushmita Gupta, Pallavi Jain, Sanjukta Roy, Saket Saurabh, Meirav Zehavi: "On the (Parameterized) Complexity of Almost Stable Marriage", in Proc. of the 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2020, December 14-18, 2020, BITS Pilani, K K Birla Goa Campus, Goa, India (Virtual Conference), LIPIcs, Vol. 182, pp. 24:1–24:17, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.
URL https://doi.org/10.4230/LIPIcs.FSTTCS.2020.24

In the Stable Marriage problem, when the preference lists are complete, all agents of the smaller side can be matched. However, this need not be true when preference lists are incomplete. In most real-life situations, where agents participate in the matching market voluntarily and submit their preferences, it is natural to assume that each agent wants to be matched to someone in his/her preference list as opposed to being unmatched. In light of the Rural Hospital Theorem, we have to relax the "no blocking pair" condition for stable matchings in order to match more agents. In this paper, we study the question of matching more agents with fewest possible blocking edges. In particular, the goal is to find a matching whose size exceeds that of a stable matching in the graph by at least t and has at most k blocking edges. We study this question in the realm of parameterized complexity with respect to several natural parameters, k, t, d, where d is the maximum length of a preference list. Unfortunately, the problem remains intractable even for the combined parameter k + t + d. Thus, we extend our study to the local search variant of this problem, in which we search for a matching that not only fulfills each of the above conditions but is "closest", in terms of its symmetric difference to the given stable matching, and obtain an FPT algorithm.

3.18 Allocation with Weak Priorities and General Constraints

Thanh Nguyen (Purdue University – West Lafayette, US)

Joint work of Thanh Nguyen, Young-san Lin, Hai Nguyen, Kemal Altinkemer
Main reference Young-San Lin, Hai Nguyen, Thành Nguyen, Kemal Altinkemer: "Allocation with Weak Priorities and General Constraints", in Proc. of the EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021, pp. 690–691, ACM, 2021.
URL https://doi.org/10.1145/3465456.3467581

We consider a resource allocation problem that combines three general features: complex resource constraints, weak priority rankings over the agents, and ordinal preferences over bundles of resources. We develop a mechanism based on a new concept called *Competitive Stable Equilibrium*. It has several attractive properties, unifies two different frameworks of one-sided and two-sided markets, and extends existing methods to richer environments. Our framework also allows for an alternative and more flexible tie-breaking rule by giving agents different budgets. We empirically apply our mechanism to reassign season tickets to families in the presence of social distancing. Our simulation results show that our method outperforms existing ones in both efficiency and fairness measures.

3.19 Survey on online matching markets

Peng Shi

Main reference Peng Shi: "Optimal Matchmaking Strategy in Two-sided Marketplaces" (June 4, 2021). USC Marshall School of Business Research Paper
URL https://doi.org/10.2139/ssrn.3536086

I provided an idiosyncratic survey of modern online matching markets, such as those for dating, lodging, labor, school and college admissions, transportation, etc., and research inspired by such marketplaces. A key issue in many of these markets is congestion, i.e., difficulty clearing the market. An interdisciplinary perspective combining operations research, economics, engineering and computer science has been fruitful in understanding congestion and identifying ways to mitigate it. We discussed some recent research in the area and open directions.

3.20 Reallocation with Priorities

Jan Christoph Schlegel (City – University of London, GB)

Joint work of Jan Christoph Schlegel, Julien Combe
URL https://crest.science/wp-content/uploads/2021/06/2021-09.pdf

We consider a reallocation problem with priorities where each agent is initially endowed with a house and is willing to exchange it, but where each house has a priority ordering over the agents of the market. In this setting, it is well known that there is no individually rational and stable mechanism so that the literature has introduced a modified stability notion called μ_0-stability. Contrary to college admission problems, where priorities are present but there is no initial endowment, we show that the modified Deferred Acceptance mechanism identified in the literature is not the only Individually Rational, Strategy-Proof and μ_0-stable mechanism. Introducing a new axiom called Independence of Irrelevant Agents and using the standard axiom of unanimity, we show that modified Deferred Acceptance mechanism is the unique mechanism that is individually rational, strategy-proof, μ_0-stable, unanimous and independent of irrelevant agents.

3.21 Is it worth sprucing up your home?

Ildikó Schlotter (Hungarian Academy of Sciences – Budapest, HU), Péter Biró (Hungarian Academy of Sciences – Budapest, HU), and Tamás Fleiner (Budapest University of Technology & Economics, HU)

We study housing markets as introduced by Shapley and Scarf in 1974. We investigate the computational complexity of various questions regarding the situation of an agent p in a housing market H: we show that it is NP-hard to find an allocation in the core of p where (i) p receives a certain house, (ii) p does not receive a certain house, or (iii) p receives a house

other than her own. We prove that the core of housing markets respects improvement in the following sense: given an allocation in the core of H where agent p receives a house h, if the value of the house owned by p increases, then the resulting housing market admits an allocation where p receives either h, or a house that she prefers to h; moreover, such an allocation can be found efficiently. We further show an analogous result in the Stable Roommates setting by proving that stable matchings in a one-sided market also respect improvement.

3.22 Fractional and Probabilistic Matching: a brief overview

Jay Sethuraman (Columbia University – New York, US)

I will review selected results on probabilistic and fractional matchings, focusing on classical results and highlighting some recent developments.

3.23 Matching and Prices

Alexander Teytelboym (University of Oxford, GB)

Joint work of Alexander Teytelboym, Ravi Jagadeesan
Main reference Ravi Jagadeesan, Alexander Teytelboym: "Matching and Money", in Proc. of the EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021, p. 634, ACM, 2021.
URL https://doi.org/10.1145/3465456.3467587

Indivisibilities and budget constraints are pervasive features of many matching markets. But gross substitutability – a standard condition on preferences in matching models – typically fails in such markets. To accommodate budget constraints and other income effects, we instead assume that agents' preferences satisfy net substitutability. Although competitive equilibria do not generally exist in our setting, we show that stable outcomes always exist and are efficient. We illustrate how the flexibility of prices is critical for our results. We also discuss how budget constraints and other income effects affect the properties of standard auction and matching procedures, as well as of the set of stable outcomes.

3.24 Blood Allocation with Replacement Donors: A Theory of Multi-unit Exchange with Compatibility-based Preferences

Utku Ünver (Boston College, US)

Joint work of Utku Ünver. (Boston College, US), Xiang Han, and Onur Kesten
Main reference Xiang Han, Onur Kesten, M. Utku Ünver: "Blood Allocation with Replacement Donors: A Theory of Multi-unit Exchange with Compatibility-based Preferences", in Proc. of the EC '21: The 22nd ACM Conference on Economics and Computation, Budapest, Hungary, July 18-23, 2021, pp. 585–586, ACM, 2021.
URL https://doi.org/10.1145/3465456.3467565

In 56 developing and developed countries, blood component donations by volun-teer non-remunerated donors can only meet less than 50% of the demand. In these countries, blood banks rely on replacement donor programs that provide blood to patients in return for donations made by their relatives or friends. These programs appear to be disorganized, non-transparent, and inefficient. We introduce the design of replacement donor programs and blood allocation schemes as a new application of market design. We introduce optimal blood allocation mechanisms that accommodate fairness, efficiency, and other allocation objectives, together with endogenous exchange rates between received and donated blood units beyond the classical one-for-one exchange. Additionally, the mechanisms provide correct incentives for the patients to bring forward as many replacement donors as possi- ble. This framework and the mechanism class also apply to general applications of multi-unit exchange of indivisible goods with compatibility-based preferences beyond blood allocation with different information problems.

3.25 Stability in Large Markets

Karolina Lena Johanna Vocke (Universität Innsbruck, AT)

Joint work of Karolina Lena Johanna Vocke, Ravi Jagadeesan

In matching models, pairwise stable outcomes do not generally existwithout substantial restrictions on both preferences and the topology of the networkof contracts. We address the foundations of matching markets by developing amatching model with a continuum of agents that allows for complex preferencesand network structures. We argue thattree stability—a refinement of pairwisestability introduced by Ostrovsky (2008)—is the natural solution concept for thissetting. Our main results show that tree-stable outcomes are guaranteed to existin large markets for arbitrary preferences and network topologies (unlike for otherstability concepts), and provide a noncooperative microfoundation for tree stability.Our framework can flexibly capture the degree to which agents can coordinate byallowing subnetworks of contracts to be made contingent on each other by beingbundled together.

3.26 Mechanisms for Facility Location with Capacity Limits

Toby Walsh (UNSW – Sydney, AU)

Joint work of Toby Walsh, Haris Aziz, Hau Chan, Barton E. Lee, Bo Li
Main reference Haris Aziz, Hau Chan, Barton Lee, Bo Li, Toby Walsh: "Facility Location Problem with Capacity Constraints: Algorithmic and Mechanism Design Perspectives", Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34(02), pp. 1806–1813, 2020.
URL https://doi.org/10.1609/aaai.v34i02.5547

I consider the facility location problem in the one dimensional setting where each facility can serve a limited number of agents from the algorithmic and mechanism design perspectives. From the algorithmic perspective, the optimization problem, where the goal is to locate facilities to minimize either the total cost to all agents or the maximum cost of any agent is NP-hard. However, the problem is fixed-parameter tractable, and the optimal solution can be computed in polynomial time whenever the number of facilities is bounded, or when all facilities have identical capacities. I then consider the problem from a mechanism design perspective where the agents are strategic and need not reveal their true locations. Several natural mechanisms studied in the uncapacitated setting either lose strategy-proofness or a bound on the solution quality for the total or maximum cost objective.

3.27 Approximability vs. Strategy-proofness in Stable Matching Problems with Ties

Yu Yokoi (National Institute of Informatics – Tokyo, JP) and Shuichi Miyazaki (Kyoto University, JP)

Joint work of Hiromichi Goko, Kazuhisa Makino, Shuichi Miyazaki, Yu Yokoi
Main reference Hiromichi Goko, Kazuhisa Makino, Shuichi Miyazaki, Yu Yokoi: "Maximally Satisfying Lower Quotas in the Hospitals/Residents Problem with Ties", CoRR, Vol. abs/2105.03093, 2021.
URL https://arxiv.org/abs/2105.03093

We consider a two-sided stable matching model. When ties are introduced, the rural hospitals theorem fails to hold, which makes some optimization problems nontrivial, such as maximizing the matching size or satisfaction ratio of lower quotas. For these problems, strategy-proof algorithms fail to find an optimal solution, irrespective of the computational complexity. This talk reviews recent three papers on strategy-proof approximation algorithms for those problems. The first two deal with cardinality-maximization of a stable matching and show the best approximation ratios attained by deterministic strategy-proof algorithms. The third one, which is the main part of this talk, investigates the problem of finding a stable matching that maximally satisfies lower quotas. For this new problem, we provide a strategy-proof approximation algorithm and several inapproximability results.

3.28 Survey on Constrained Matching

Makoto Yokoo (Kyushu University – Fukuoka, JP)

Two-sided matching deals with finding a desirable combination of two parties, e.g., students and colleges, workers and companies, and medical residents to hospitals. Beautiful theoretical results on two-sided matching have been obtained, i.e., the celebrated Deferred Acceptance mechanism is strategyproof for students, and obtains the student optimal matching among all stable matchings. However, these results are applicable only for the standard model, where only distributional constraints are the maximum quota (capacity limit) of each college. In many real application domains, various distributional constraints are imposed due to social requirements. For example, a college needs a certain number of students to operate, or some medical residents must be assigned to a rural hospital.

In this talk, I represent a simple and general abstract model, and introduce a few representative constraints that can be formalized using this model. In this model, distributional constraints are defined over a set of allocation vectors, each of which describes the number of students allocated to each college. Then, I present two general mechanisms. One is the generalized DA, which works when distributional constraints satisfy two conditions: hereditary and an M-natural-convex set [1]. More specifically, the generalized DA is strategyproof, and finds the student optimal matching among all matchings that satisfy some stability requirement. The other is the adaptive DA [2], which works when distributional constraints satisfy hereditary condition. It is strategyproof and nonwasteful.

References

1 Kojima, F., Tamura, A., Yokoo, M.: Designing matching mechanisms under constraints: An approach from discrete convex analysis, Journal of Economic Theory, 176 (2018)
2 Goto, M., Kurata, R., Kojima, F., Kurata, R., Tamura, A, Yokoo, M.: Designing Matching Mechanisms under General Distributional Constraints, American Economic Journal: Microeconomics, 9 (2):226-62, (2017).

3.29 Absolutely and simply popular rankings

Ágnes Cseh (Hasso-Plattner-Institut, Universität Potsdam, DE)

Joint work of Ágnes Cseh, Sonja Kraiczy, David Manlove
Main reference Sonja Kraiczy, Ágnes Cseh, David F. Manlove: "On absolutely and simply popular rankings", CoRR, Vol. abs/2102.01361, 2021.
URL https://arxiv.org/abs/2102.01361

Van Zuylen et al. introduced the notion of a popular ranking in a voting context, where each voter submits a strictly-ordered list of all candidates. A popular ranking π of the candidates is at least as good as any other ranking σ in the following sense: if we compare π to σ, at least half of all voters will always weakly prefer π. Whether a voter prefers one ranking to another is calculated based on the Kendall distance.

A more traditional definition of popularity – as applied to popular matchings, a well-established topic in computational social choice – is stricter, because it requires at least half of the voters *who are not indifferent between π and σ* to prefer π. In this paper, we derive

structural and algorithmic results in both settings, also improving upon the results by van Zuylen et al. We also point out strong connections to the famous open problem of finding a Kemeny consensus with 3 voters.

3.30 Kidney Exchange progress in Germany

Ágnes Cseh

URL https://crossover-nierenspenderliste.de/

A short report on the current status quo of kidney exchange in Germany.

4 Working groups

4.1 Gender Terminology in Bipartite Stable Matching

Robert Bredereck (HU Berlin, DE)

Bipartite Stable Matching is classically presented as "Stable Marriage" with one site being men and the other site being women. Meant as illustration and not as proposal for real marriage, the many successful applications of the model are all in completely different domains. The classical terminology, however, can be easily misunderstood and becomes questionable at latest when

- one site behaves always passive while the other behaves always active,
- one site manipulates while the other is honest,
- there is external manipulation, or
- some couples are forced or forbidden.

Participants of the seminar discussed the seriousness of these issues in particular in situations where people from outside the community are involved (teaching, grant proposals, etc.). To avoid misunderstanding many participants are using alternative terminologies:

- sportsmen ↔ sportswomen (mixed teams such as tennis)
- leaders ↔ followers (dancing)
- doctors ↔ hospitals
- student ↔ colleges
- workers ↔ companies
- workers ↔ apprentices
- mentors ↔ mentees

While some of the alternatives even allow to keep using different grammatical gender for the two sites (and so allow to write easily comprehensible texts), other alternatives fit better with the manipulation setting. Some of these alternative terminologies are already established in more specialized or generalized settings of Stable Matching, but may still qualify for the illustration of Bipartite Stable Matching. Another possibility in use is to keep the marriage market terminology while clearly putting it into a historical context.

4.2 Popular Matching with few blocking pairs

Sushmita Gupta (The Institute of Mathematical Sciences – Chennai, IN), Ágnes Cseh (Hasso-Plattner-Institut, Universität Potsdam, DE), Pallavi Jain (Indian Institute of Techology, IN), Baharak Rastegari (University of Southampton, GB), Ildikó Schlotter (Hungarian Academy of Sciences – Budapest, HU), and Kavitha Telikepalli (TIFR Mumbai, IN)

We work in the classic 2-sided matching market model with strict preferences on both sides. In the context of popular matchings, we study two scenarios.

1. We aim to find a popular matching that is blocked by a given edge set. We have a fixed set of edges along which agents won't deviate from the matching. This is very much like free edges / socially stable matchings, with a different optimality principle. We look for a stable matching M with free edges that indeed block M, plus we want M to be popular on the top of it.
2. We aim to find a popular matching that is blocked by exactly / at most k edges. We have limited resources to compensate agents who could be better off by switching to their blocking partner. The regret of a blocking agent is calculated by counting the number of her blocking edges.

Our goal is to find out whether a popular matching in the above two scenarios exist. Preliminary results indicate hardness even in restricted cases.

4.3 Lexicographic preferences in matching and market design

Bettina Klaus (University of Lausanne, CH)

In some recent research projects (own and other's research), lexicographic preference domains have nice interpretations and allow for new positive results. One example are Shapley-Scarf housing markets

- with limited externalities or
- with multiple types.

Another example is the joint coalition formation paper Seçkin Özbilen presented in this workshop.

The working group studied other matching and related models for which lexicographic preferences can be defined in a meaningful way and with the potential for new positive results. In particular, many-to-many two-sided matching markets were considered. The main outcome was an update to all working group participants on the current literature and a better understanding of specific open questions in relation to lexicographic preferences in many-to-many two-sided matching markets.

Participants

- Péter Biró
Hungarian Academy of Sciences – Budapest, HU
- Somouaoga Bonkoungou
University of Lausanne, CH
- Florian Brandl
Universität Bonn, DE
- Jiehua Chen
TU Wien, AT
- Ágnes Cseh
Hasso-Plattner-Institut, Universität Potsdam, DE
- Lars Ehlers
University of Montreal, CA
- Tamás Fleiner
Budapest University of Technology & Economics, HU
- Martin Hoefer
Goethe-Universität – Frankfurt am Main, DE
- Zsuzsanna Jankó
Corvinus University – Budapest, HU
- Bettina Klaus
University of Lausanne, CH
- Simon Mauras
University Paris Diderot, FR
- Seckin Özbilen
University of Lausanne, CH
- Katarzyna Paluch
University of Wroclaw, PL
- Jan Christoph Schlegel
City – University of London, GB
- Karolina Lena Johanna Vocke
Universität Innsbruck, AT

Remote Participants

- Haris Aziz
UNSW – Sydney, AU
- Martin Bichler
TU München, DE
- Felix Brandt
TU München, DE
- Robert Bredereck
HU Berlin, DE
- Christine Cheng
University of Wisconsin – Milwaukee, US
- Sanmay Das
George Mason University – Fairfax, US
- John Dickerson
University of Maryland – College Park, US
- Di Feng
University of Lausanne, CH
- Sushmita Gupta
The Institute of Mathematical Sciences – Chennai, IN
- Klaus Heeger
TU Berlin, DE
- Hadi Hosseini
Pennsylvania State University – University Park, US
- Pallavi Jain
Indian Institute of Techology – Jodhpur, IN
- Yash Kanoria
Columbia University – New York, US
- Flip Klijn
CSIC – Barcelona, ES
- Fuhito Kojima
University of Tokyo, JP
- Alexander Lam
UNSW – Sydney, AU
- Irene Y. Lo
Stanford University, US
- Nicholas Mattei
Tulane University – New Orleans, US
- Michael McKay
University of Glasgow, GB
- Shuichi Miyazaki
Kyoto University, JP
- Thayer Morrill
North Carolina State University – Raleigh, US
- Thanh Nguyen
Purdue University – West Lafayette, US
- Sofiat Olaosebikan
University of Glasgow, GB
- Daniel Paulusma
Durham University, GB
- Baharak Rastegari
University of Southampton, GB
- Ildikó Schlotter
Hungarian Academy of Sciences – Budapest, HU
- Jay Sethuraman
Columbia University – New York, US
- Kavitha Telikepalli
TIFR Mumbai, IN
- Alexander Teytelboym
University of Oxford, GB
- Utku Unver
Boston College, US
- Toby Walsh
UNSW – Sydney, AU
- Mobin YahyazadehJeloudar
Stanford University, US
- Yu Yokoi
National Institute of Informatics – Tokyo, JP
- Makoto Yokoo
Kyushu University – Fukuoka, JP

Report from Dagstuhl Seminar 21302

Approximate Systems

Edited by

Eva Darulova[1], Babak Falsafi[2], Andreas Gerstlauer[3], and Phillip Stanley-Marbell[4]

1 **MPI-SWS – Kaiserslautern, DE,** eva@mpi-sws.org
2 **EPFL – Lausanne, CH,** babak.falsafi@epfl.ch
3 **University of Texas at Austin, US,** gerstl@ece.utexas.edu
4 **University of Cambridge, GB,** phillip.stanleymarbell@gmail.com

Abstract

This report summarizes the presentations and discussion sessions at the Dagstuhl Seminar 21302 "Approximate Systems" that took place during July 25 – 30, 2021. Due to COVID, the seminar was held in a hybrid fashion, with around 1/3 of the attendees on-site and the remaining ones online. The seminar discussed advances and open challenges in applying approximate computing techniques across the stack and across different application domains, and we hope that this report can provide a useful resource also for other researchers.

Seminar July 25–30, 2021 – http://www.dagstuhl.de/21302
2012 ACM Subject Classification Hardware → Analysis and design of emerging devices and systems; Computer systems organization → Architectures; Computer systems organization → Embedded and cyber-physical systems; Software and its engineering → Software notations and tools
Keywords and phrases approximate computing, energy-efficient computing, pareto optimization
Digital Object Identifier 10.4230/DagRep.11.6.147

1 Executive Summary

Eva Darulova
Babak Falsafi
Andreas Gerstlauer
Phillip Stanley-Marbell

Resource efficiency is becoming an increasingly important challenge, especially due to the pervasiveness of computing systems and the diminishing returns from performance improvements of process technology scaling. At the same time, many important applications have nondeterministic specifications or are robust to noise in their execution. They thus do not necessarily require fully reliable computing systems and their resource consumption can be reduced by introducing or exposing approximations.

While trading correctness for efficiency has been part of computing systems since the early days, it has seen renewed interest in the past decade. Different techniques have been since developed for applying and controlling approximations and the errors they introduce at different levels of the compute stack. Unfortunately, most of these techniques have been

applied in isolation, making simplified assumptions about the other levels. It is thus unclear how all the different techniques interact, combine and complement or negate each other to provide end-to-end application benefits.

The aim of this seminar was to bring together researchers from different domains working on approximate computing, algorithms, programming languages, compilers, architecture and circuits, in order to explore open challenges and opportunities and to define cross-area research directions and collaborations relating to an end-to-end application of approximate computing principles across the compute stack.

The seminar consisted of brief presentations by a subset of the participants that covered the entire computing stack from hardware to applications, and that focused on the current challenges. The talks were followed by discussions in breakout groups that first focused on the different application areas of high-performance computing, embedded systems and deep learning, followed by group discussions on particular fundamental and cross-cutting challenges that were identified during the first breakout session. This report includes the abstracts of the participant's presentations as well as summaries of the breakout group discussions.

2 Table of Contents

Executive Summary

Eva Darulova, Babak Falsafi, Andreas Gerstlauer, and Phillip Stanley-Marbell . . . 147

Overview of Talks

Approximate Computing to Fight Temperature Effects in NPUs
Hussam Amrouch . . . 151

On the Curse and the Beauty of Randomness for Guaranteeing Reliable Quality with Unreliable Silicon
Andreas Burg . . . 151

Self-Adaptive FPGA-Based Image Processing Using Approximate Arithmetics
Jürgen Teich . . . 152

Opportunities and Challenges for Approximation in DNA storage
Djordje Jevdjic . . . 152

Calyx: Your DSL-to-Hardware Compiler Construction Kit
Adrian Sampson . . . 152

System-aware Distributed Machine Learning
Gauri Joshi . . . 153

Approximate AI on the Edge
David Atienza Alonso . . . 153

Numerical Encoding for DNN Training
Babak Falsafi . . . 153

Approximating Numerical Kernels and Beyond
Eva Darulova . . . 154

Context-Aware Coding for Computer Memories
Lara Dolecek . . . 154

An Optimization Playground for Precision and Number Representation Tuning
Olivier Sentieys . . . 155

A Review and Characterization of Approximate Arithmetic Circuits for Approximate Computing
Jie Han . . . 155

How do Approximations Impact Analysis, Compiling, and Testing
Sasa Misailovic . . . 156

An Adaptive Application Framework with Customizable Quality Metrics
Ulrich Kremer . . . 156

How to Reduce Numerical Precision in Weather and Climate Simulations
Peter Dueben . . . 157

Some Mathematical Challenges in Inexact Computing
Laura Monroe . . . 157

Working groups

Approximate Computing Challenges for HPC Applications
Eva Darulova 158

Approximate Computing Challenges for Embedded Systems
Phillip Stanley-Marbell 158

Approximate Computing Challenges for Deep Learning
Babak Falsafi 159

Design Patterns for Approximation Across the Stack
Damien Zufferey 159

Intermediate Representations and Tool Flows for Approximate Computing
Andreas Gerstlauer 160

Differentiation of Error Models
Andreas Burg 161

Challenges for Approximate Hardware
Georgios Zervakis 162

Participants 163

Remote Participants 163

3 Overview of Talks

3.1 Approximate Computing to Fight Temperature Effects in NPUs

Hussam Amrouch (Universität Stuttgart, DE)

Main reference Hussam Amrouch, Georgios Zervakis, Sami Salamin, Hammam Kattan, Iraklis Anagnostopoulos, Jörg Henkel: "NPU Thermal Management", IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., Vol. 39(11), pp. 3842–3855, 2020.
URL https://doi.org/10.1109/TCAD.2020.3012753

Neural processing units (NPUs) are becoming an integral part in all modern computing systems due to their substantial role in accelerating neural networks. In this talk, I will discuss the thermal challenges that NPUs bring, demonstrating how multiply-accumulate (MAC) arrays, which form the heart of any NPU, impose serious thermal bottlenecks to any on-chip systems due to their excessive power densities. Some of the questions that we will discuss are 1) the effectiveness of precision scaling and frequency scaling (FS) in temperature reductions for NPUs and 2) how advanced on-chip cooling using superlattice thin-film thermoelectric (TE) open doors for new tradeoffs between temperature, throughput, cooling cost, and inference accuracy in NPU chips.

3.2 On the Curse and the Beauty of Randomness for Guaranteeing Reliable Quality with Unreliable Silicon

Andreas Burg (EPFL – Lausanne, CH)

Joint work of Andreas Burg, Reza Ghanaatian
Main reference Reza Ghanaatian, Marco Widmer, Andreas Burg: "Design for Test with Unreliable Memories by Restoring the Beauty of Randomness", IEEE Design Test, pp. 1–1, 2021.
URL https://doi.org/10.1109/MDAT.2021.3081687

Process variations lead to reliability issues in advanced process nodes. Unfortunately, the associated reliability issues lead to a huge quality spread between manufactured ASICs even for the most fault tolerant applications. This quality spread has burdened "approximate computing" since today's production-test strategies fail to separate dies with sufficient quality from dies with insufficient quality. We analyse this issue, which is ignored in many publications that only report an average quality metric, and provide a surprising and counter-intuitive solution to the testability issue. The key idea is thereby to restore an environment in which frozen (e.g., stuck-at) faults in the hardware no longer have a deterministic effect on computation results. This measure leads to an ergodic fault model that restores the beauty of randomness in a sense that it enables meaningful stochastic quality metrics and allows for simple error mitigation strategies such as averaging which are invalid without randomization.

3.3 Self-Adaptive FPGA-Based Image Processing Using Approximate Arithmetics

Jürgen Teich (Universität Erlangen-Nürnberg, DE)

Joint work of Jutta Pirkl, Andreas Becher, Jorge Echavarria, Jürgen Teich, Stefan Wildermann
Main reference Jutta Pirkl, Andreas Becher, Jorge Echavarria, Jürgen Teich, Stefan Wildermann: "Self-Adaptive FPGA-Based Image Processing Filters Using Approximate Arithmetics", in Proc. of the 20th International Workshop on Software and Compilers for Embedded Systems, SCOPES 2017, Sankt Goar, Germany, June 12-13, 2017, pp. 89–92, ACM, 2017.
URL https://doi.org/10.1145/3078659.3078669

In this talk, we propose a concept of self-adaptive image processing that is able to autonomously adapt 2D-convolution filter operators of different accuracy degrees by means of partial reconfiguration on Field-Programmable-Gate-Arrays (FPGAs). Experimental evaluation shows that the dynamic system is able to better exploit a given error tolerance than any static approximation technique due to its responsiveness to changes in input data. Additionally, it provides a user control knob to select the desired output quality via the metric threshold at runtime.

3.4 Opportunities and Challenges for Approximation in DNA storage

Djordje Jevdjic (National University of Singapore, SG)

DNA has emerged as a chemical medium for both data storage and computation, offering a number of important and unique advantages and promising to close the widening gap between the demand and supply for data storage. However, due to the high error rates and the complex nature of errors in DNA significant amounts of redundant resources must be invested to allow for full recovery of binary data from DNA molecules. The stochastic nature of the chemical processes involved and the approximate nature of data recovery algorithms presents a number of opportunities for approximations across this unique stack. This talk will cover opportunities and challenges in building an error-efficient DNA-based data storage system.

3.5 Calyx: Your DSL-to-Hardware Compiler Construction Kit

Adrian Sampson (Cornell University – Ithaca, US)

Main reference Rachit Nigam, Samuel Thomas, Zhijing Li, Adrian Sampson: "A compiler infrastructure for accelerator generators", in Proc. of the ASPLOS '21: 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Virtual Event, USA, April 19-23, 2021, pp. 804–817, ACM, 2021.
URL https://doi.org/10.1145/3445814.3446712

Calyx is an open-source infrastructure for building DSL-to-hardware compilers. It's centered around a new representation for programs that blends structure (hardware-like components and their connections) and control (temporal ordering). The infrastructure enables optimization and lowering passes that translate high-level DSL semantics into RTL implementations.

3.6 System-aware Distributed Machine Learning

Gauri Joshi (Carnegie Mellon University – Pittsburgh, US)

Large-scale machine learning training, in particular, distributed stochastic gradient descent (SGD), needs to be robust to inherent system variabilities such as unpredictable computation and communication delays. These scalability hurdles are amplified in the emerging framework of federated learning where machine learning models are training on resource-limited edge devices. In this talk, I will discuss open problems in distributed and federated learning.

3.7 Approximate AI on the Edge

David Atienza Alonso (EPFL – Lausanne, CH)

Wearable devices are poised as the next frontier of innovation in the context of Internet-of-Things (IoT) that can benefit from approximate computing to be able to provide personalized healthcare at minimum energy, which can improve our lives and transform the medical industry. This new family of smart wearable medical devices provides a great opportunity for the integration of the nex-generation of artificial intelligence (AI) based technologies in combination with approximate computing in medical devices. However, major key challenges remain in achieving this potential due to the inherent resource-constrained nature of wearable systems, coupled with the uncertainty of the output of the final system when approximation is used at different levels of the system design. In this talk, the current approaches to deliver approximate computing in edge AI to create the next-generation of heterogeneous smart wearables architectures are discussed. The critical architectural enabler is the combination of multiple processors, with a coarse-grained reconfigurable AI accelerator and in-memory computing) as a scalable way to fully deliver the concept of personalized medicine at minimal power. Then, the key challenges to propose an iterative design and optimization flow to bring AI (particularly convolutional neural networks – CNNs) to resource-constrained embedded platforms through selectively applying approximation at different levels will be presented.

3.8 Numerical Encoding for DNN Training

Babak Falsafi (EPFL – Lausanne, CH)

Main reference Mario Drumond, Tao Lin, Martin Jaggi, Babak Falsafi: "Training DNNs with Hybrid Block Floating Point", in Proc. of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pp. 451–461, 2018.
URL https://proceedings.neurips.cc/paper/2018/hash/6a9aeddfc689c1d0e3b9ccc3ab651bc5-Abstract.html

The wide adoption of DNNs has given birth to unrelenting computing requirements, forcing datacenter operators to adopt domain-specific accelerators to train them. These accelerators typically employ densely packed full-precision floating-point arithmetic to maximize performance per area. Ongoing research efforts seek to further increase that performance density by

replacing floating-point with fixed- point arithmetic. However, a significant roadblock for these attempts has been fixed point's narrow dynamic range, which is insufficient for DNN training convergence. We identify block floating point (BFP) as a promising alternative representation since it exhibits wide dynamic range and enables the majority of DNN operations to be performed with fixed-point logic. Unfortunately, BFP alone introduces several limitations that preclude its direct applicability. In this work, we introduce HBFP, a hybrid BFP-FP approach, which performs all dot products in BFP and other operations in floating point. HBFP delivers the best of both worlds: the high accuracy of floating point at the superior hardware density of fixed point. For a wide variety of models, we show that HBFP matches floating point's accuracy while enabling hardware implementations that deliver up to 8.5× higher throughput.

3.9 Approximating Numerical Kernels and Beyond

Eva Darulova (MPI-SWS – Kaiserslautern, DE)

Joint work of Anastasia Volkova, Anastasiia Izycheva, Helmut Seidl, Heiko Becker, Magnus Myreen, Zachary Tatlock, Debasmita Lohar, Sylvie Putot, Eric Goubault

Computing resources are fundamentally limited and sometimes an exact solution may not even exist. Thus, when implementing real-world systems, approximations are inevitable, as are the errors they introduce. The magnitude of errors is problem-dependent but higher accuracy generally comes at a cost in terms of memory, energy or runtime, effectively creating an accuracy-efficiency tradeoff. To take advantage of this tradeoff, we need to ensure that the computed results are sufficiently accurate, otherwise we risk disastrously incorrect results or system failures. Unfortunately, the current way of programming with approximations is mostly manual, and consequently costly, error prone and often produces suboptimal results. I will show how we can already approximate straight-line numerical kernels fully automatically, while guaranteeing a user-provided error bound, and discuss our work towards supporting programs beyond kernels that feature conditional statements and loops. Finally, I will sketch what the outstanding challenges are.

3.10 Context-Aware Coding for Computer Memories

Lara Dolecek (University of California at Los Angeles, US)

Joint work of Lara Dolecek, Clayton Schoeny, Mark Gottscho, Puneet Gupta

Error-control coding (ECC) is routinely used to overcome errors in computer memories. In this talk, we demonstrate how intrinsic system knowledge can be used to offer error recovery beyond the baseline ECC guarantees. This system knowledge is used as context for error recovery, and comes in a variety of ways, including data type, instruction structure, and frequency of instructions, among others. We present a heuristic error recovery approach for a known ECC method and a new code design strategy that can take advantage of the underlying system properties. The proposed approach can have benefits in a variety of applications that have intrinsic structure or redundancy, and we envision this idea to be applicable beyond computer memories.

3.11 An Optimization Playground for Precision and Number Representation Tuning

Olivier Sentieys (University & INRIA – Rennes, FR)

Joint work of Van-Phu Ha, Tomofumi Yuki, Daniel Ménard, Olivier Sentieys
Main reference Van-Phu Ha, Olivier Sentieys: "Leveraging Bayesian Optimization to Speed Up Automatic Precision Tuning", in Proc. of the Design, Automation & Test in Europe Conference & Exhibition, DATE 2021, Grenoble, France, February 1-5, 2021, pp. 1542–1547, IEEE, 2021.
URL https://doi.org/10.23919/DATE51398.2021.9474209

Energy, delay, and area vary a lot between number representations (e.g., float, fixed-point) and word-length (i.e., bit-width of data and computation). Automatic precision tuning is an optimization process that determines the number of bits for each data, minimizing a cost/energy function, constrained by (application) quality degradation (e.g., noise power, SSIM, abs. error). This talk first presents some of the latest results in this field before to move to the problem of jointly exploring number representation during optimization. Results include the development of a custom float operator library and their use in applications such as the training process of deep neural networks with ultra-low precision. This talk concludes with related new problems that may be of interest to build approximate systems.

3.12 A Review and Characterization of Approximate Arithmetic Circuits for Approximate Computing

Jie Han (University of Alberta – Edmonton, CA)

Joint work of Honglan Jiang, Francisco J. H. Santiago, Hai Mo, Leibo Liu, Fabrizio Lombardi, Jie Han
Main reference Honglan Jiang, Francisco Javier Hernandez Santiago, Hai Mo, Leibo Liu, Jie Han: "Approximate Arithmetic Circuits: A Survey, Characterization, and Recent Applications", Proc. IEEE, Vol. 108(12), pp. 2108–2135, 2020.
URL https://doi.org/10.1109/JPROC.2020.3006451
Main reference Honglan Jiang, Cong Liu, Leibo Liu, Fabrizio Lombardi, Jie Han: "A Review, Classification, and Comparative Evaluation of Approximate Arithmetic Circuits", ACM J. Emerg. Technol. Comput. Syst., Vol. 13(4), pp. 60:1–60:34, 2017.
URL https://doi.org/10.1145/3094124

Approximate computing is emerging as a new paradigm for high-performance and energy-efficient design of circuits and systems. This talk aims to provide a brief review and characterization of recently proposed approximate arithmetic circuits under different design constraints. Specifically, approximate adders, multipliers and dividers are characterized via synthesis under optimizations for performance and area, respectively. The error and circuit characteristics are then generalized for different classes of designs. The applications of these circuits in image processing and deep neural networks indicate that such computations are more sensitive to errors in addition than those in multiplication, so a larger approximation can be tolerated in multipliers than in adders. The use of approximate arithmetic circuits can improve the quality of image processing and deep learning in addition to the benefits in performance and power consumption for these applications.

3.13 How do Approximations Impact Analysis, Compiling, and Testing

Sasa Misailovic (University of Illinois – Urbana-Champaign, US)

Tradeoffs between accuracy, performance and energy exist in many resource-intensive applications pervasive in machine learning and robotics. Manually optimizing these tradeoffs with flexible accuracy or precision requirements is extremely difficult. I will highlight our work on programming systems (including languages, compilers, and runtime systems) for accuracy aware optimization of programs.

I will discuss several challenges and lessons learned on how to 1) cope with randomness when testing systems, 2) conquer approximation in heterogeneous systems by novel compilers, and 3) support concurrent and distributed computations in program analysis. I will conclude with a discussion about how we can make future approximation-aware systems more usable.

3.14 An Adaptive Application Framework with Customizable Quality Metrics

Ulrich Kremer (Rutgers University – Piscataway, US)

Joint work of Ulrich Kremer, Liu Liu, Sibren Isaacman
Main reference L. Liu, S. Isaacman, and U.Kremer: An Adaptive Application Framework with Customizable Quality Metrics. ACM Transactions on Design Automation of Electronic Systems (TODAES), Special Issue on Approximate Systems, October 2021, to be pulished.
URL https://doi.org/10.1145/3477428

Many embedded environments require applications to produce outcomes under different, potentially changing, resource constraints. Relaxing application semantics through approximations enables trading off resource usage for outcome quality. Although quality is a highly subjective notion, previous work assumes given, fixed low-level quality metrics that often lack a strong correlation to a user's higher-level quality experience. Users may also change their minds with respect to their quality expectations depending on the resource budgets they are willing to dedicate to an execution. This motivates the need for an adaptive application framework where users provide execution budgets and a customized quality notion. The paper presents a novel adaptive program graph representation that enables user-level, customizable quality based on basic quality aspects defined by application developers. Developers also define application configuration spaces, with possible customization to eliminate undesirable configurations. At runtime, the graph enables the dynamic selection of the configuration with maximal customized quality within the user provided resource budget.

An adaptive application framework based on our novel graph representation has been implemented on Android and Linux platforms, and evaluated on eight benchmark programs, four with fully customizable quality. Using custom quality instead of the default quality, users may improve their subjective quality experience value by up to 3.59×, with 1.76× on average under different resource constraints. Developers are able to exploit their application structure knowledge to define configuration spaces that are on average 68.7compared to existing, structure oblivious approaches. The overhead of dynamic reconfiguration averages less than 1.84% of the overall application execution time.

3.15 How to Reduce Numerical Precision in Weather and Climate Simulations

Peter Dueben (ECMWF – Reading, GB)

This talk will give an overview on ongoing efforts to explore the reduction of numerical precision in weather and climate models. While the European Centre for Medium-Range Weather Forecasts (ECMWF) has recently switched from double to single precision in operational predictions, we also investigate the use of lower precision levels, such as half precision, for our models. The precision reduction is non-trivial as it is difficult to diagnose a precision level that is still "good enough" when simulating a chaotic system – such as atmosphere or ocean. On the other hand, we have good knowledge about forecast uncertainties which can be used to optimise precision within the simulations.

3.16 Some Mathematical Challenges in Inexact Computing

Laura Monroe (Los Alamos National Laboratory, US)

This talk is an overview of the relationship between mathematics and inexact systems, with an emphasis on errors and error-correction. In particular, we emphasize hardware/software codesign and the interplay between mathematics, the base physics of the system, and the algorithms and software that brings them together.

Challenges up and down the stack are discussed, and several examples are given, including software-defined error correction (Gottscho et al.); the interplay between Hamming and application-based distances, with applications ranging from computational fluid dynamics to basic integer calculations; and natural application resilience. The fault model is discussed, with its derivation from the physical device and modes of addressing device-dependent faults.

Finally, we propose development of a catalog of design patterns, inspired by those in the object-oriented design community [1] and the resilience community [2], as a tool describing solutions to problems commonly seen in inexact systems and software and representing best practices used by experienced practitioners in the field.

References

1 Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston, Mass. : Addison-Wesley, 2016

2 Saurabh Hukerikar and Christian Engelmann. *Resilience Design Patterns: A Structured Approach to Resilience at Extreme Scale*. Journal of Supercomputing Frontiers and Innovations (JSFI), volume 4, number 3, pages 4-42, October 1, 2017

4 Working groups

4.1 Approximate Computing Challenges for HPC Applications

Eva Darulova (MPI-SWS – Kaiserslautern, DE)

High-performance computing, such as used for example in weather simulations, has traditionally not actively applied (additional) approximations beyond those necessary by the domain. Computations were done with 64 bit floating-point arithmetic and assumed to be correct enough; hardware is supposed to be correct, too. The picture is starting to change as hardware is not becoming automatically faster and more efficient if one just waits, and is also becoming more heterogeneous. Hence, in principle, approximate computing is of interest to the HPC community. One particular case of existing deliberate approximations are recent efforts to move computations from 64 bit floats to lower precisions. These efforts are motivated by available low-precision hardware for machine learning. The conversion is currently done manually and takes a very long time. Hence, tool support in form of debugging tools would be much appreciated. Static analysis tools or fully automated tools are likely to not be very meaningful, since an exact baseline is often not available, i.e. the approximated code needs to match the existing implementation's behavior. The community seems to be slowly moving away from Fortran to Python or DSLs, making it more feasible to develop debugging tools. In addition to finite-precision, we have identified further approximations at different levels of the stack that may be beneficial to HPC. One are techniques from federated machine learning that may be helpful for the often massively parallel HPCs applications (e.g. a weather simulation may discretize the earth and distribute the simulation for each part on 1000 nodes). Further, error correction techniques may be of interest as the probability of random bitflips increases, one hand hand due to the sheer size of the computations, and on the other hand due to the use of approximate hardware. Accelerators have the potential to provide speed-ups important to HPC, today mostly GPUs are used. FPGAs have been used to run only small models, as programming them is manual and painful. In summary, there seems to be potential to apply approximations in HPC across the stack. In order to develop the corresponding techniques and tools, representative benchmarks or example codes are needed.

4.2 Approximate Computing Challenges for Embedded Systems

Phillip Stanley-Marbell (University of Cambridge, GB)

The discussion in the embedded systems theme centered on the idea that in embedded computing systems, which by definition interface with the physical world, it is essential to know when the result of an approximation technique has led to an erroneous data value or erroneous control flow behavior. One way in which erroneous behavior could be detected is by checking for violation of some invariant property, either on a single value of machine state or across multiple items of program state (e.g., an invariant across entries in a matrix). A general concept discussed was the idea of exploiting the fact that the signals in embedded systems are usually physical signals which need to obey the laws of physics.

The discussions observed that erroneous behavior of interest will typically be input-dependent, since erroneous behaviors that are not input-dependent could in principle be found by static analysis techniques. Dynamic detection of erroneous behavior resulting from approximation is, at the moment, not a well-explored topic and could be fertile ground for future research.

Once erroneous behavior is detected, there is the natural question of how to make this detected state available to a system to act upon it. Erroneous behavior could be detected inside, e.g., a microprocessor or compute accelerator, in which case one natural way to notify the system of the detection is by raising an interrupt. In hardware systems in general, the detection could be used to set a hardware signal, while in software systems an exception could be raised (to be handled by an appropriate exception handler).

Finally, when erroneous behavior has been detected and the system notified, it will still be a challenge to develop new methods to adapt to the result of erroneous behavior resulting from approximation. This was again identified in the embedded theme as a fertile ground for future research.

4.3 Approximate Computing Challenges for Deep Learning

Babak Falsafi (EPFL – Lausanne, CH)

Machine learning has emerged as a killer application with wide applicability across a number of domains. There are two trends that are at inflection point in ML: (1) the imminent end of Moore's Law and the need for post-Moore platforms, and (2) the continued exponential growth in ML at 20% per year (as forecasted by a number of think-tanks including IDC) and the need for scaling platforms. There are a number of fundamental challenges in ML platform design. One is the lack explainability and ad hoc methods to improve algorithms. Another is the divergence in platforms between inference and training. A third fundamental challenge is search for models that would allow for iso-accuracy in prediction while reducing the required computational resources. Fortunately, ML inherently lends itself well to cross-layer optimization in platforms and co-design. One great area to explore is convergence of inference/training through common numerical encodings that would enable algorithm/hardware co-design. Explainability of optimal numerical encoding would require hand-in-hand collaboration of computer system designers and numerical analysts. Another area would be hardware mechanisms that would facilitate parameter and model search. A third area of research would be how the choice of ML application would impact algorithm/hardware co-design.

4.4 Design Patterns for Approximation Across the Stack

Damien Zufferey (MPI-SWS – Kaiserslautern, DE)

Specific approximation techniques, e.g., numerical precision, can be applied on their own. However, to get more benefit one can apply approximation consistently across the entire software and hardware stack. Optimization can span the computation, communication,

memory, storage, and time. Cyber-physical systems is an application domain that can benefit from approximations that touches all theses domains. For instance, reduced precision and perforation can reduce the energy needed by a control loop but the trade-off is affecting the system's stability. Sensors are an ideal place to integrate very fast and efficient analog processing. In the case of distributed control, reducing the amount and frequency of communication is what reducing numerical precision and control loop frequency to local controllers. Optimizations across the stack are challenging to evaluate and tune especially when the ground truth is not know, e.g., SLAM. To help programmers integrate approximations into their system, we propose to write a book gathering design patterns for approximate computing. The book will contain guidelines about when to use approximation, how to use it, what the caveats are. Beyond performances, the book will also discuss the impact of approximation on aspects such as robustness and security.

4.5 Intermediate Representations and Tool Flows for Approximate Computing

Andreas Gerstlauer (University of Texas at Austin, US)

With a large number of design knobs when applying approximations across the compute stack, approximation-aware design automation solutions and design tools will be indispensable in navigating associated design spaces. This will require combining approximations across multiple abstraction levels into integrated cross-layer tool flows. Many approximation techniques at higher levels of abstraction are inherently application-specific. Furthermore, tool flows that are generic to span across application areas have proven to be too complex and infeasible to develop. Instead, domain-specific tool flows have been successfully applied in many key application areas, such as TensorFlow- or PyTorch-based flows in machine learning. Such flows are built around domain-specific languages (DSLs) and domain-specific intermediate representation (IRs), which often take the form of more functional-oriented programming models. On top of such high-level domains-specific IRs, various source-level optimizations, including domain-specific approximations such as neural network pruning are then applied. Domain-specific approximations will also need to account for the inherent dependency of approximations on application-specific inputs and input distributions. At the same time, there are a range of implementation-dependent approximation techniques, such precision scaling or code transformations that are target-specific but general across domains. It is desirable to implement such target-aware optimizations in a common implementation back-end that can be shared across different domain-specific tool flows. We envision tool flows that combine various domain-specific front-end IRs feeding into a common back-end IR for compilation, synthesis and implementation on different software and hardware targets (Figure 1). Such back-end IRs will likely take a more target-specific imperative form. They will need to support back-end approximations using appropriate domain-specific quality models coming from the top as well as target-specific cost models coming from the bottom. Various compiler and high-level synthesis IRs exist, but they are predominantly based on sequential software models that are a poor fit for custom hardware targets and associated

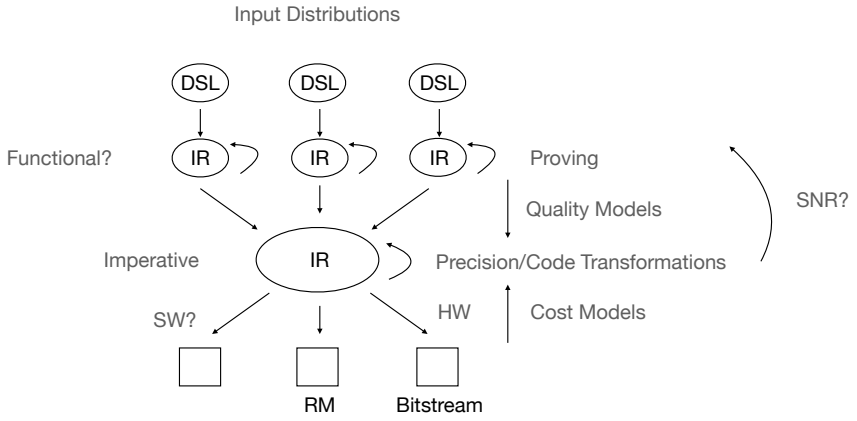


Figure 1 Approximation-aware cross-layer tool flows.

hardware approximations. Some efforts, such as the Calyx or HPVM projects at Cornell and UIUC are underway to develop new hardware- and approximation-aware IRs. However, complete cross-layer tool flows that combine domain-specific front-end with target-specific back-end approximations are still lacking and require further research.

4.6 Differentiation of Error Models

Andreas Burg (EPFL – Lausanne, CH)

The working group discussed the need for a careful and rigorous differentiation between different types of errors that are considered under the approximate computing paradigm. In fact, erroneous or approximate behaviour of a circuit or implementation can originate from different causes or origins at different stages of the life-cycle of a design (from the design stage, to the manufacturing, to the operation in the field). Such different origins require fundamentally different error models that must not be confused neither when assessing the modelling and impact of errors, nor or when considering suitable mitigation measures on circuit, architecture or algorithm levels, at design time or during operation. On the one hand, approximate computing paradigms often introduce errors intentionally at design time, for example at the algorithm level or through approximate arithmetic components. Such errors are deterministic in nature and perfectly known. Hence purely stochastic assessment of their impact and forward error correction for mitigation are not immediately applicable. However, the frequently used and technically inaccurate modelling as stochastic errors may still be useful to provide "compact" quality metrics (avrg, variance, ...) across large random data sets. Such deterministic errors (e.g., introduced at design time) may also be seen as related to source coding, while coding efficiency is not necessarily measured in a reduction in number of bits. Such a view may provide new insights, but need further consideration. On the other hand, errors or parametric variations introduced for example during manufacturing are not known at design time, but still deterministic after manufacturing. Hence, stochastic modelling must be done with care since every realization of the design is ultimately different in its (deterministic) behaviour (non-ergodic behaviour), which requires a yield analysis

and sophisticated test methodologies to identify dies with degraded quality. Finally, some sources of error such as single-event upsets and external noise (e.g., on the power supply) are both unknown and stochastic. Hence, assessment of their impact with deterministic models is not feasible, but in turn stochastic models for impact analysis and error mitigation measures such as error correction apply. Finally, joint source-channel coding ideas could address deterministic approximations and random errors jointly, but further elaboration is also here necessary.

In the second part of the discussion, the group discussed error models for DNA storage and potential options for error control coding. The main issue with DNA storage is capturing the effect of erasures and insertions which render conventional codes not immediately applicable. Work-arounds exist in the literature, but research is still in its infancy.

4.7 Challenges for Approximate Hardware

Georgios Zervakis (KIT – Karlsruhe, DE)

Approximate hardware design forms a very promising solution to boost the efficiency of Domain Specific Accelerators. For example, Samsung already uses approximate multipliers in some of its DSPs while the conventional today 8-bit fixed-point inference accelerators can be viewed as an approximation of the traditionally used single floating-point representations. Nevertheless, embracing approximate circuits for general purpose computing appears less promising or not mature yet. To design approximate circuits, algorithmic approximations seem to deliver better solutions. One of the main reasons is that they can be better supported by the EDA tools. The major deficiency identified in the design of approximate accelerators is to understand how errors, with respect to both inputs (sensing) or computation (approximate units), propagate throughout the application. In addition, errors are input dependent but existing error compensation/balance techniques are mainly based on statistics and cannot always guarantee better accuracy. Mixed approximation or reconfigurable approximation kernels are required along with approximation techniques that force error cancellation throughout the different approximated computations. Moreover, approximate circuit verification and large system simulation when using approximate accelerators still remain open issues. Significant research has focused on arithmetic units and small accelerators. However, the overall system performance or gains is unclear and a systematic methodology to translate the gains and error or the approximate circuit to system gains and quality is required. Finally, two propositions were made: i) examine approximate design for gain in other metrics such as security and fabrication cost and ii) use analog computations for near sensing application and combine approximation in the analog domain with approximation in the digital domain.

Participants

- Hussam Amrouch
Universität Stuttgart, DE
- David Atienza Alonso
EPFL – Lausanne, CH
- Eric Atkinson
MIT – Cambridge, US
- Andreas Burg
EPFL – Lausanne, CH
- Eva Darulova
MPI-SWS – Kaiserslautern, DE
- Lara Dolecek
University of California at Los Angeles, US
- Babak Falsafi
EPFL – Lausanne, CH
- Djordje Jevdjic
National University of Singapore, SG
- Debasmita Lohar
MPI-SWS – Saarbrücken, DE
- Jürgen Teich
Universität Erlangen-Nürnberg, DE
- Damien Zufferey
MPI-SWS – Kaiserslautern, DE

Remote Participants

- Sara Achour
Stanford University, US
- R.Iris Bahar
Brown University – Providence, US
- Swarnendu Biswas
Indian Institute of Technology Kanpur, IN
- Peter Dueben
ECMWF – Reading, GB
- Andreas Gerstlauer
University of Texas at Austin, US
- Ghayoor Gillani
University of Twente, NL
- Jie Han
University of Alberta – Edmonton, CA
- Anastasiia Izycheva
TU München, DE
- Vijay Janapa Reddi
Harvard University – Cambridge, US
- Gauri Joshi
Carnegie Mellon University – Pittsburgh, US
- Ulrich Kremer
Rutgers University – Piscataway, US
- Sasa Misailovic
University of Illinois – Urbana-Champaign, US
- Laura Monroe
Los Alamos National Laboratory, US
- Sri Parameswaran
UNSW – Sydney, AU
- Adrian Sampson
Cornell University – Ithaca, US
- Olivier Sentieys
University & INRIA – Rennes, FR
- Phillip Stanley-Marbell
University of Cambridge, GB
- Radha Venkatagiri
Oregon State University, US
- Norbert Wehn
TU Kaiserslautern, DE
- Georgios Zervakis
KIT – Karlsruher Institut für Technologie, DE