IP1

Finding Large Matchings in Dynamic Graphs via Extremal Graph Theory

Maintaining large matchings in dynamic graphs—where edges are continuously inserted and deleted—is a fundamental algorithmic challenge. In this talk, we examine a new approach for this problem that leverages tools from extremal graph theory, particularly the so-called Ruzsa–Szemerédi graphs. This perspective leads to a novel algorithmic framework that can be viewed in two ways: as a conditional worst-case guarantee, assuming the nonexistence of dense Ruzsa–Szemerédi graphs with certain parameters in general; or as a beyond worst-case strategy, exploiting the absence of such extremal subgraphs in typical inputs. This blend of extremal graph theory and algorithmic reasoning opens up new directions in the design of efficient dynamic graph algorithms.

Sepehr Assadi
Rutgers University, U.S.
sepehr@assadi.info

IP2

ACDA: When the First "A" Comes First

Algorithm design with mathematical and experimental analysis for classic combinatorial-optimization problems can be excellent starting points for applications. But application details in national security, and likely in industry, can profoundly change the algorithmic questions and techniques even for classic problems. In this sense the first "A" in ACDA ("Applied") drives the second ("Algorithms"). In this talk I will describe two real-world applications and the resulting algorithmic results/journey: significantly reducing false negatives when finding patterns in streaming data, and how to make open social-network datasets more "human." I will also describe a few other algorithmic challenges in areas such as sensor placement in municipal water networks.

Cynthia Phillips
Sandia National Laboratories
caphill@sandia.gov

IP3

Adaptive Dynamic Bitvectors

While operations rank and select on static bitvectors can be supported in constant time, lower bounds show that supporting updates raises the cost per operation to Theta(log n / log log n) on bitvectors holding n bits. This is a shame in scenarios where updates are possible but uncommon. I will describe a representation of bitvectors that I called "adaptive dynamic bitvector", which uses the asymptotically optimal n + o(n) bits of space and, if there are q queries per update, supports all the operations in O(log(n/q) / log log n) amortized time. Further, this time can be proved to be optimal in the cell probe model. I will also describe an implementation that outperforms, for sufficiently large q, much better-engineered nonadaptive implementations. I will describe a particular application where we have successfully used this data structure, and mention several other cases where it could be applied.

Gonzalo Navarro
University of Chile
gnavarro@dcc.uchile.cl

IP4

Overcoming Parallelism Challenges in Data Analytics Using Sparse Linear Algebra

The diverse and non-trivial challenges of parallelism in data analytics require computing infrastructures that go beyond the demand of traditional simulation-based sciences. The growing data volume and complexity have outpaced the processing capacity of single-node machines in these areas, making massively parallel systems an indispensable tool. However, programming on high-performance computing (HPC) systems poses significant productivity and scalability challenges. It is important to introduce an abstraction layer that provides programming flexibility and productivity while ensuring high system performance. As we enter the post-Moore's Law era, effective programming of specialized architectures is critical for improved performance in HPC. As large-scale systems become more heterogeneous, their efficient use for new, often irregular and communication-intensive data analysis computation becomes increasingly complex. In this talk, we discuss how sparse linear algebra can be used to achieve performance and scalability on extreme-scale systems while maintaining productivity for emerging data-intensive scientific challenges.

Giulia Guidi
Cornell University
gg434@cornell.edu

SP1

Data Structures for Fast Systems

In this talk, I'll show how algorithms can be used to solve decades-old problems in systems design. I'll present an algorithmic approach to co-designing TLB hardware and the paging mechanism to increase TLB reach without the fragmentation issues incurred by huge pages. Along the way, I'll introduce a new hash-table design that overcomes existing tradeoffs and achieves better performance than state-of-the-art hash tables both in theory and in practice. Key to these results are "tiny pointers," an algorithmic technique for compressing pointers. Going forward, Ill discuss how tiny pointers and related ideas can continue to have impact in practical systems.

Alex Conway
Cornell University
ajc473@cornell.edu

JP1

Joint Plenary with AN25: Setting a Course for Post-Moore Software Performance

Software performance engineering (SPE) is the science and art of making code run fast or otherwise limiting its consumption of resources, such as power, memory footprint, network utilization, file IOs, response time, etc. SPE encompasses algorithms, parallel computing, caching, vectorization, bit tricks, loop unrolling, compiler-switch selection, tailoring code to the architecture, exploiting sparsity, changing data representation, metaprogramming, etc. Historically, gains in performance from miniaturization, codified in Moore's Law, relieved programmers from the burden of making software run fast and learning SPE techniques. I will explain why the end of Moores Law now makes SPE

a critical technical skill. Since SPE is neither extensively researched nor widely taught in the universities, however, it risks devolving into an unstructured collection of ad hoc tricks. Now is the time to establish SPE as a science-based discipline, alongside traditional areas of computer science.

Leiserson Charles
Massachusetts Institute of Technology
cel@mit.edu

CP1

Optimization of Infrastructure Network Maintenance: Complexity, Exact Solutions, and a Greedy Heuristic

Efficient maintenance planning under resource constraints is an important task in infrastructure management. In this context, we consider the optimization problem of finding a set of edges in a network to perform maintenance on in order to maximize the quality of the resulting network (i.e., to minimize the sum of failure scores), while keeping the cost below a budget threshold. Our model is based on two cost factors: (a) individual costs for maintenance on each edge and (b) additional fixed costs for each connected set of edges to model efficiency gains in larger construction sites. By a reduction from the Steiner tree problem, we show that our optimization problem is NP-hard. Since real-world data on infrastructure networks with quality and cost assessments are hardly available to the public, we also provide two generators to construct simulated problem instances that resemble real-world data. To solve the optimization problem optimally on small instances, we build an ILP and employ Gurobi. For instances of non-trivial size, we propose a heuristic that grows a construction site greedily around edges that need maintenance with high priority. Our experiments on 20 datasets from two classes show that our greedy algorithm can solve instances with up to 50K edges in at most 78 minutes. Comparative experiments on small instances yield that the best greedy solutions per instance are on average only 13% away from the optimum.

Lukas Berner, Felix Heitmann
Humboldt-Universität zu Berlin
lukas.berner@hu-berlin.de,
heitmanf@informatik.hu-berlin.de

Henning Meyerhenke
Karlsruhe Institute of Technology
henning.meyerhenke@kit.edu

CP1

Partitioning Trillion Edge Graphs on Edge Devices

Processing large graphs with billions of entities is critical in fields like bioinformatics, high-performance computing, and navigation. Efficient graph partitioning, which divides a graph into subgraphs while minimizing inter-block edges, is essential to graph processing, as it optimizes parallelism and enhances data locality. Traditional in-memory partitioners like METIS and KaHIP offer high-quality partitions but are often infeasible for huge graphs due to large memory overhead. Streaming partitioners reduce memory usage to $\mathcal{O}(n)$, where n is the number of nodes, by loading nodes sequentially and assigning them to blocks on-the-fly. This paper introduces StreamCPI, a novel framework that reduces the memory overhead of streaming partitioners through run-length compression of block assignments. StreamCPI enables partitioning of trillion-edge graphs on edge devices. Within this framework, we propose a modification to the LA-vector bit vector for append support, usable for online run-length compression across streaming applications. Empirical results show that StreamCPI reduces memory usage while maintaining or improving partition quality. Using StreamCPI, the Fennel partitioner partitions a graph with 17 billion nodes and 1.03 trillion edges on a Raspberry Pi, achieving significantly better quality than Hashing, a widely used feasible approach on edge devices. StreamCPI thus advances graph processing by enabling high-quality partitioning on low-cost machines.

Adil Chhabra
Heidelberg University
adil.chhabra@stud.uni-heidelberg.de

Florian Kurpicz
Karlsruhe Institute of Technology
kurpicz@kit.edu

Christian Schulz
Heidelberg University, Germany
christian.schulz@informatik.uni-heidelberg.de

Dominik Schweisgut
Humboldt University of Berlin
dominik.schweisgut@hu-berlin.de

Daniel Seemaier
Karlsruhe Institute of Technology
daniel.seemaier@kit.edu

CP1

Why Is My Route Different Today? An Algorithm for Explaining Route Selection

Users of routing services like Apple Maps, Google Maps, and Waze frequently wonder why a given route is proposed. This question particularly arises when dynamic conditions like traffic and road closures cause unusual routes to be proposed. While many dynamic conditions may exist in a road network at any time, only a small fraction of those conditions are typically relevant to a given users route. In this work, we introduce the concept of an simple valid explanation (SVE), which consists of a small set of traffic-laden road segments that answer the following question: Which traffic conditions cause a particular shortest traffic-aware route to differ from the shortest traffic-free route? We give an efficient algorithm for finding SVEs and show that they theoretically and experimentally lead to small and interpretable answers to the question.

Aaron Schild, Sreenivas Gollapudi
Google Research
aschild@google.com, sgollapu@google.com

Anupam Gupta
New York University
anupamgupta@google.com

Kostas Kollias, Ali Sinop
Google Research
kostaskollias@google.com, asinop@google.com

CP1

Efficient Dual Decomposition for Vehicle Routing

Problems in Military Logistics

Military logistics requires delivering heterogeneous commodities across a complex intermodal network using a fleet of heterogeneous vehicles. In a collaboration with the U.S. Marine Corps and Navy providing decision support in training exercises, we found that standard mixed integer programming (MIP) formulations of this problem becomes computationally intractable as the network size grows large. To address this challenge, we present a fast, memory-efficient algorithm for solving this problem using dual decomposition. By relaxing the inventory balance constraints, we separate the problem into subproblems that can be solved independently across vehicles and locations. Crucially, we show that these subproblems can be solved efficiently: the vehicle subproblems admit a dynamic programming solution, while the location subproblems have analytical solutions. These subproblems are parallelizable, and we further accelerate convergence using the bundle method. While only weak duality holds due to integer variables, we prove that the vehicle subproblem's LP relaxation yields integral solutions, allowing us to match the LP relaxation objective. Moreover, to recover a feasible primal solution from the dual solution, we propose an algorithm that generates paths matching vehicles with supply-demand pairs using network flow. In realistic scenarios, we show our method achieves near-optimality in under a minute, improving scalability compared to traditional MIP solvers.

Samuel Tan
Cornell University
sst76@cornell.edu

Peter I. Frazier
School of Operations Research and Information Engineering
Cornell University
pf98@cornell.edu

Matthew Ford, Huseyin Topaloglu
Cornell University
mtf62@cornell.edu, ht88@cornell.edu

CP1

The Case for External Graph Sketching

Algorithms in the data stream model use $O(\text{polylog}N)$ space to compute some property of an input of size N, and many of these algorithms are implemented and used in practice. However, sketching algorithms in the graph semi-streaming model use $O(V\text{polylog}V)$ space for a V-vertex graph, and the fact that implementations of these algorithms are not used in the academic literature or in industrial applications may be because this space requirement is too large for RAM on today's hardware. In this paper we introduce the external semi-streaming model, which addresses the aspects of the semi-streaming model that limit its practical impact. In this model, the input is in the form of a stream and $O(V\text{polylog}V)$ space is available, but most of that space is accessible only via block I/O operations as in the external memory model. The goal in the external semi-streaming model is to simultaneously achieve small space and low I/O cost. Using this transformation and other techniques, we present external semi-streaming algorithms for connectivity, bipartiteness testing, $(1+\epsilon)$-approximating MST weight, testing k-edge connectivity, $(1+\epsilon)$-approximating the minimum cut of a graph, computing ϵ-cut sparsifiers, and approximating the density of the densest subgraph.For many of these problems, our external semi-streaming algorithms outperform the state of the art algorithms in both the sketching and external-memory models.

David Tench
Rutgers
dtench@pm.me

Hanna Komlos
New York University
hkomlos@gmail.com

Michael A. Bender
Stony Brook University
bender@cs.stonybrook.edu

Martin Farach-Colton
New York University
martin@farach-colton.com

Riko Jacob
IT University Copenhagen
rikj@itu.dk

Evan West
Stony Brook University
etwest@cs.stonybrook.edu

CP2

An Efficient Sparse Kernel Generator for O(3)-Equivariant Deep Networks

Rotation equivariant graph neural networks, i.e., networks designed to guarantee certain geometric relations between their inputs and outputs, yield state-of-the-art performance on spatial deep learning tasks. Key to these models is the Clebsch-Gordon (CG) tensor product, a kernel that contracts two dense feature vectors with a highly structured sparse tensor to produce a dense output vector. The operation, which may be repeated millions of times for typical equivariant models, is a costly and inefficient bottleneck. We introduce a GPU sparse kernel generator for the CG tensor product that provides significant speedup over the best existing open and closed-source implementations. We break the tensor product into a series of kernels with operands that fit entirely into registers, enabling us to emit long arithmetic instruction streams that maximize instruction-level parallelism. By fusing the CG tensor product with a subsequent graph convolution, we reduce both intermediate storage and global memory traffic over naive approaches that duplicate input data. We also provide optimized kernels for the gradient of the CG tensor product and a novel identity for the higher partial derivatives required to predict interatomic forces. In FP64 precision, We offer up to 6.2x inference-time speedup for the MACE chemistry foundation model over the original unoptimized version.

Vivek Bharadwaj
University of California, Berkeley
vivek_bharadwaj@berkeley.edu

Austin Glover
University of California Berkeley
austin_glover@berkeley.edu

Aydin Buluç
Lawrence Berkeley National Laboratory
abuluc@lbl.gov

James W. Demmel
UC Berkeley
Division of Computer Science
demmel@berkeley.edu

CP2

Evaluating Learned Indexes for External Memory Joins

Joins are among the most time-consuming and data-intensive operations in relational query processing. Much research effort has been applied to the optimization of join processing due to their frequent execution. Recent studies have shown that CDF-based learned models can create smaller and faster indexes, accelerating in-memory joins. However, their effectiveness for external-memory joins, which are crucial for large-scale databases, remains underexplored. This paper evaluates the impact of learned indexes on external-memory joins for both sorted and unsorted data. We compare learned index-based joins against traditional join methods such as hash joins, sort joins, and indexed nested-loop joins on real-world and simulated datasets. Additionally, we analyze learned index-based joins across multiple dimensions, including storage device types, data sorting, parallelism, constrained memory environments, and varying model error. Our experiments reveal that learned indexes in external-memory joins can trade off accuracy for space without significantly degrading performance. While learned indexes provide smaller index sizes and faster lookups, they perform similarly to B-trees in external-memory joins since the total amount of I/O remains unchanged. Additionally, the construction times of learned indexes are $\sim 1000\times$ longer, and although they are $24\times$ smaller than the internal nodes of a B-tree, these nodes only represent 0.4%1% of the data size.

Yuvaraj Chesetti, Prashant Pandey
Northeastern University
chesetti.y@northeastern.edu,
p.pandey@northeastern.edu

CP2

Accelerating Reductions Using Graph Neural Networks for the Maximum Weight Independent Set Problem

The Maximum Weight Independent Set problem is a fundamental NP-hard problem in combinatorial optimization with several real-world applications. Given an undirected vertex-weighted graph, the task is to find a subset of the vertices with the highest weight such that no two vertices in the set are adjacent. An important part of solving this problem in both theory and practice is data reduction rules. However, the most complicated rules are often not used in practice since the time needed to check them exhaustively becomes infeasible. In this work, we introduce two main results. First, we introduce several new data reduction rules and evaluate their effectiveness on real-world data. Second, we use a machine learning screening algorithm to speed up the reduction phase, thereby enabling more complicated rules to be applied. Our screening algorithm consults a Graph Neural Network oracle to decide if the probability of successfully reducing the graph is sufficiently large. For this task, we provide a dataset of labeled vertices for use in supervised learning. We also present the first results for this dataset using established Graph Neural Network architectures. Our experimental evaluation shows that with our preprocessing technique, we can reduce instances to within one percent of what is possible using the full set of rules exhaustively, while being four times faster for the reductions that undergo early screening.

Ernestine Großmann
Heidelberg University
e.grossmann@informatik.uni-heidelberg.de

Kenneth Langedal
University of Bergen
kenneth.langedal@hotmail.com

Christian Schulz
Heidelberg University, Germany
christian.schulz@informatik.uni-heidelberg.de

CP2

Computing Experiment-Constrained D-Optimal Designs

In optimal experimental design, the objective is to select a limited set of experiments that maximizes information about unknown model parameters based on factor levels. This work addresses the generalized D-optimal design problem, allowing for nonlinear relationships in factor levels. We develop scalable algorithms suitable for cases where the number of candidate experiments grows exponentially with the factor dimension, focusing on both first- and second-order models under design constraints. Particularly, our approach integrates convex relaxation with pricing-based local search techniques, which can provide upper bounds and performance guarantees. Unlike traditional local search methods, such as the Fedorov exchange and its variants, our method effectively accommodates arbitrary side constraints in the design space. Furthermore, it yields both a feasible solution and an upper bound on the optimal value. Numerical results highlight the efficiency and scalability of our algorithms, demonstrating superior performance compared to the state-of-the-art commercial software, JMP.

Aditya Pillai
Georgia Institute of Technology
apillai32@gatech.edu

Gabriel Ponte
University of Michigan
gabponte@umich.edu

Marcia Fampa
Federal University of Rio de Janeiro
fampa@cos.ufrj.br

Jon Lee
University of Michigan
jonxlee@umich.edu

Mohit Singh
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology
mohitsinghr@gmail.com

Weijun Xie
Georgia Institute of Technology
wxie@gatech.edu

CP2

Approximate Forest Completion and Learning-

Augmented Algorithms for Metric Minimum Spanning Trees

Finding a minimum spanning tree for n points in an arbitrary metric space is a fundamental combinatorial primitive, but this takes $\Omega(n^2)$ time to even approximate. Motivated by massive-scale HPC clustering applications, we present a new framework for metric spanning tree computation that first (1) uses heuristics to find a subset of a spanning tree (the initial forest) then (2) finds a nearly optimal way to connect the forest into a full spanning tree. The algorithm has $o(n^2)$ time and space complexity, is easy to parallelize, and comes with rigorous theoretical guarantees in the form of learning-augmented approximation algorithms.

Nate Veldt
Texas A & M University
nveldt@tamu.edu

Thomas Stanley
Texas A&M University
thomas.stanley@tamu.edu

Benjamin W. Priest, TREVOR Steil, KEITA Iwabuchi, T.S. Jayram
Lawrence Livermore National Laboratory
priest2@llnl.gov, steil1@llnl.gov, iwabuchi@llnl.gov, thathachar1@llnl.gov

Geoffrey D. Sanders
Center for Applied Scientific Computing
Lawrence Livermore National Lab
sanders29@llnl.gov

CP3

Parallelizing the Approximate Minimum Degree Ordering Algorithm: Strategies and Evaluation

The approximate minimum degree algorithm is widely used before numerical factorization to reduce fill-in for sparse matrices. While considerable attention has been given to the numerical factorization process, less focus has been placed on parallelizing the approximate minimum degree algorithm itself. In this paper, we explore different parallelization strategies, and introduce a novel parallel framework that leverages multiple elimination on distance-2 independent sets. Our evaluation shows that parallelism within individual elimination steps is limited due to low computational workload and significant memory contention. In contrast, our proposed framework overcomes these challenges by parallelizing the work across elimination steps. To the best of our knowledge, our implementation is the first scalable shared memory implementation of the approximate minimum degree algorithm. Experimental results show that we achieve up to a 8.30x speedup using 64 threads over the state-of-the-art sequential implementation in SuiteSparse.

Yen-Hsiang Chang
University of California, Berkeley
yenhsiangc@berkeley.edu

Aydin Buluç
Lawrence Berkeley National Laboratory
abuluc@lbl.gov

James W. Demmel
UC Berkeley
Division of Computer Science
demmel@berkeley.edu

CP3

Alternative Bases for New Fast Matrix Multiplication Algorithms

Fast matrix multiplication algorithms are of practical use, provided that they apply to feasible input sizes and have small leading coefficients in their arithmetic and IO complexities. In recent years, many new sub-cubic time matrix multiplication algorithms that are applicable to feasible matrices have been introduced, including the recently discovered algorithms using AlphaTensor (Nature, 2022) and flip graphs (ISSAC, 2023). However, their arithmetic and IO complexities have quite large leading coefficients, making them impractical. We decrease these coefficients (by up to 89%), resulting in algorithms with more practical potential. To this end, we use the alternative basis method and provide a new way to compute the multiplications recursively. We provide an algorithm for finding optimal decompositions for the alternative basis method and use dynamic programming for the recursion reordering. Our new matrix multiplication algorithms retain the improved exponent while decreasing the leading coefficient of the arithmetic and communication costs. These result in the fastest existing algorithms for several base case dimensions. Combined with lower bounds on the arithmetic costs of bilinear algorithms, we conclude that some of our algorithms are optimal.

Olga Holtz, Abraham Hsu
University of California, Berkeley
holtz@math.berkeley.edu, hpghsu@berkeley.edu

Yoav Moran
The Hebrew University
yoav.gross@mail.huji.ac.il

Oded Schwartz
Hebrew University of Jerusalem
odedsc@cs.huji.ac.il

Gal Wiernik
The Hebrew University of Jerusalem
Tel Aviv University
galwiernik@gmail.com

CP3

The Number of the Beast: Reducing Additions in Fast Matrix Multiplication Algorithms for Dimensions Up to 666

Strassen's algorithm multiplies two 2×2 matrices using 7 multiplications and 18 additions, instead of the naive 8 multiplications and 4 additions. Winograd reduced the number of additions to 15. By changing basis, Karstadt and Schwartz lowered the number of additions to 12, which they showed to be optimal within this generalized Karstadt-Schwartz (KS) framework. We present improved methods for classical optimization of the number of additions for larger Strassen-type matrix multiplication schemes, without any change of basis. For some schemes our methods beat the results found within the KS framework, including Laderman's algorithm for multiplying 3×3 matrices, which we reduce from the naive 98 additions to 62, compared to 74 in the KS framework. We indicate that our algorithm's performance improves with the dimension

compared to previous work within the KS framework. We also apply our algorithms to a set of algorithms due to Moosbauer and Kauers for which we have no reference results from previous work. When multiplying an $(n \times m)$ matrix with an $(m \times k)$ matrix, the schoolbook algorithm uses $nk(m-1)$ additions. Our algorithm optimizes the number of additions to roughly $cnk(m-1)$, where c is a small dimension-independent constant. Our implementation is very efficient for dimensions including (and surpassing) the 666 limit, i.e. $(n, m, k) = (6, 6, 6)$, in our reference set of fast multiplication algorithms.

Erik Mårtensson, Paul Stankovski Wagner
Lund University
erik.martensson@eit.lth.se,
paul.stankovski_wagner@eit.lth.se

CP3

Minimizing the Division Factor of Toom-Cook Algorithms

Long integer multiplication is a critical kernel in many cryptography and quantum applications. Toom-Cook-k fast integer multiplication algorithms are often favored in large-scale computations due to their superior asymptotical runtime, but often include nontrivial divisions (divisions by integers that are not powers of 2), which are incredibly costly for applications like quantum computing and cryptography. For this reason, Gu and Li (2018) proposed the 'division-free' Toom-Cook, which combines all nontrivial division operations into a single large division at the end of the algorithm, called the division factor, which determines the overhead costs. We extend their work and reduce the division factors of Toom-Cook-4, Toom-Cook-5, and Toom-Cook-8, achieving a significant $\sim 30\%$ reduction in bit count. Evaluation points are traditionally chosen as powers of 2 due to their presumed performance benefits. However, our findings suggest that using prime-based evaluation points results in considerably better division factors and reduced costs. We establish a universal lower bound for the division factor of Toom-Cook-k, regardless of evaluation points, and provided a tighter lower bound for cases where evaluation points are powers of 2. These bounds align with our results for Toom-Cook-3, Toom-Cook-4, and Toom-Cook-5, demonstrating that our algorithms are optimal.

Roy Nissim
The Hebrew University of Jerusalem
roynis1@gmail.com

CP3

A Static Pivoting Strategy for Direct Solution of Large and Sparse Symmetric Linear Systems

Direct solution of large and sparse linear systems on distributed-memory systems often favor static pivoting over dynamic pivoting as the row exchanges required by the latter can be prohibitively expensive to perform in parallel. Although static pivoting for non-symmetric systems is well-studied, such pivoting is less explored for symmetric systems as it is more challenging due to the additional constraint of preserving symmetricity in permuting the matrix. In this work we develop a graph model that enables us to use any weighted matching algorithm to get heavy 2x2 blocks and exploit them in the parallel sparse direct solvers. This graph model represents the symmetric model with an enhanced standard graph representation where there is an additional edge and vertex for each row/column of matrix with the weight of the edge equal to the magnitude of the diagonal. We utilize a fast and parallel weighted matching heuristic called Suitor on the graph model to produce heavy blocks. The existence of 2x2 blocks requires processing of the rows in such blocks together and hence they necessitate extensive changes in the solve such as supernode structure, fill-in reduction, etc. We integrate our approach into SuperLU_DIST, a distributed sparse direct solver and evaluate our approach on symmetric indefinite systems.

Oguz Selvitopi
Lawrence Berkeley Lab
roselvitopi@lbl.gov

Tianyi Shi, Yang Liu, Aydin Buluc, Sherry Li
Lawrence Berkeley National Laboratory
tianyishi@lbl.gov, liuyangzhuan@lbl.gov, abuluc@lbl.gov, xsli@lbl.gov

CP4

DataFlow Embedding Via Recursively-Tiled Mixed-Integer Programs

Spatial architectures such as Cerebras can offer great on-chip bandwidth, but they lack a mature programming model. This requires compiler stacks that embed a control dataflow graph (CDFG) onto a physical architecture. This embedding problem is a challenging variant of place & route, in which the processing elements (PEs) can host a variety of different types of jobs. Naive applications of mixed-integer programming (MIP) have proven intractable in our experience, even on problems with only 100 CDFG nodes and PEs. Even finding an incumbent solution has taken weeks with Gurobi, a state-of-the-art solver, on capable machines. Thus, the literature moved to reinforcement learning and metaheuristics like simulated annealing. We introduce a more scalable MIP approach. We define a single MIP that both produces embeddings of kernel computations ("tiles") and instantiates previously embedded tiles represented as single jobs in the CDFG. We demonstrate our process on canonical linear algebra primitives with loop unrolling. We show results, evaluated for correctness in the Structural Simulation Toolkit. One primary impact of our work is that our MIP and template library could generate extensive sets of embedding designs of many computations. These could provide training data for machine-learning-based embedding methods. Such approaches have been options only for those with troves of data, such as vendors with decades of design history.

Jonathan Berry
Sandia National Laboratories, U.S.
jberry@sandia.gov

Clay Hughes, Cynthia Phillips
Sandia National Laboratories
chughes@sandia.gov, caphill@sandia.gov

CP4

Optimizing Districting Plans to Maximize Majority-Minority Districts Via IPs and Local Search

In redistricting litigation, effective enforcement of the Voting Rights Act has often involved providing the court with districting plans that display a larger number of majority-minority districts than the current proposal. Recent work by Cannon et al. proposed a heuristic algorithm for generating plans to optimize majority-minority districts, which

they called *short bursts*; that algorithm relies on a sophisticated random walk over the space of all plans, transitioning in bursts, where the initial plan for each burst is the most successful plan from the previous burst. We propose a method based on integer programming, where we build upon another previous work, the stochastic hierarchical partitioning algorithm, which heuristically generates a robust set of potential districts (viewed as columns in a standard set partitioning formulation); that approach was designed to optimize a different notion of fairness across a statewide plan. We design a new column generation algorithm to find plans via integer programming that outperforms short bursts on multiple data sets in generating statewide plans with significantly more majority-minority districts. These results also rely on a new local re-optimization algorithm to iteratively improve on any baseline solution, as well as an algorithm to increase the compactness of districts in plans generated (without impacting the number of majority-minority districts).

Philip D. Brous, David B. Shmoys
Cornell University
pdb62@cornell.edu, david.shmoys@cornell.edu

CP4

A Space-Efficient Algebraic Approach to Robotic Motion Planning

We investigate whether algorithms based on arithmetic circuits are a viable alternative to existing solvers for Graph Inspection, a problem with direct application in robotic motion planning. Specifically, we seek to address the high memory usage of existing solvers. Aided by novel theoretical results enabling fast solution recovery, we implement a circuit-based solver for Graph Inspection which uses only polynomial space and test it on several realistic robotic motion planning datasets. In particular, we provide a comprehensive experimental evaluation of a suite of engineered algorithms for three key subroutines. This evaluation demonstrates that while circuit-based methods are not yet practically competitive in running times for our robotics application, they can be useful in the setting where the memory resource is limited. It also provides insights which may guide future efforts to bring circuit-based algorithms from theory to practice.

Matthias Bentert
University of Bergen
matthias.bentert@uib.no

Daniel Coimbra Salomao, Alex Crane, Yosuke Mizutani
University of Utah
u1397516@utah.edu, alex.crane@utah.edu,
yos@cs.utah.edu

Felix Reidl
Birkbeck, University of London
f.reidl@bbk.ac.uk

Blair D. Sullivan
University of Utah
sullivan@cs.utah.edu

CP4

Smart Charging and Optimization of Personalized Flexibility Services for Electric Vehicles's Users

The growing adoption of Electric Vehicles (EVs) presents new challenges for Charging Point Operators (CPOs) due to the increasing charging demand. At the same time, it creates opportunities to influence the flexibility of EV users. In this work, we consider a CPO that offers a price menu to EV users who are flexible with their charging completion time but have fixed energy demand. The price menu corresponds to multiple charging completion time options, each associated with a different price. The optimal price menu design problem is formulated as a bilevel optimization problem. At the upper level, the CPO determines the charging prices of the menu and the optimal power allocation among EVs to maximize its profit. At the lower level, each EV selects the option that maximizes its utility. This bilevel optimization problem is reformulated into a single-level optimization problem using various approaches, such as the classical Karush-Kuhn-Tucker (KKT) transformation or the optimal value transformation. Numerical results evaluate the performance of the proposed price menu.

Rita Safi
EDF R&D SYSTEME
rita.safi@edf.fr

Raphaël Payen
EDF R&D OSIRIS
raphael.payen@edf.fr

Yezekael Hayel
LIA - Avignon University
yezekael.hayel@univ-avignon.fr

Alix Dupont
EDF R&D SYSTEME
alix.dupont@edf.fr

Tania Jiménez
LIA - Avignon University
tania.jimenez@univ-avignon.fr

CP4

Speeding Up Stencil Computation Using Gaussian Approximations

Stencils are widely used in scientific and industrial computing for the simulation of physical systems. Given a multidimensional spatial grid containing initial data, these stencil patterns are applied uniformly to all cells of the grid over multiple timesteps to obtain the final data. All known algorithms for general stencil computations performed $\Omega(NT)$ work, where N is the number of cells of the grid and T is the number of timesteps. In 2021, Ahmad et al. (SPAA 2021) reduced the computational complexity of performing linear stencil computations on periodic and aperiodic grids to $o(NT)$ using the Fast Fourier Transform. In this paper, we present an approximation algorithm for linear stencil computations on free-space grids which performs work almost linear in the size of the grid and thus significantly improves over the computational complexities of all known algorithms for solving the problem exactly. Our paper is built on several interesting ideas: (i) linear stencils can be modeled as biased random walks, (ii) the influence of a linear stencil over multiple timesteps follows a Gaussian distribution, and (iii) linear stencil computation can be modeled as an n-body problem taking the Gaussian as the force function and hence linear stencil computations can be performed efficiently using the fast Gauss transform. Our algorithm produces approximate results that are controlled

by a tunable error parameter.

Rezaul Chowdhury
Stony Brook University, U.S.
shaikat@cs.utexas.edu

Zafar Ahmad
Stony Brook University
zafahmad@cs.stonybrook.edu

Rathish Das
University of Houston
rathish@uh.edu

Pramod Ganapathi, Aaron Gregory, Yimin Zhu
Stony Brook University
pramod.ganapathi@cs.stonybrook.edu, afgregory@cs.stonybrook.edu, yimzhu@cs.stonybrook.edu

CP5

New Algorithms for Incremental Minimum Spanning Trees and Temporal Graph Applications

Processing graphs with temporal information (the temporal graphs) has become increasingly important in the real world. In this paper, we study efficient solutions to temporal graph applications using new algorithms for Incremental Minimum Spanning Trees (MST). The first contribution of this work is to formally discuss how a broad set of setting-problem combinations of temporal graph processing can be solved using incremental MST, along with their theoretical guarantees. However, to give efficient solutions for incremental MST, we observe a gap between theory and practice. While many classic data structures, such as the link-cut tree, provide strong bounds for incremental MST, their performance is limited in practice. Meanwhile, existing practical solutions used in applications do not have any non-trivial theoretical guarantees. Our second and main contribution includes new algorithms for incremental MST that are efficient both in theory and in practice. Our new data structure, the AM-tree, achieves the same theoretical bound as the link-cut tree for temporal graph processing and shows strong performance in practice. In our experiments, the AM-tree has competitive or better performance than existing practical solutions due to theoretical guarantee, and can be significantly faster than the link-cut tree (7.811x in update and 7.713.7x in query).

Xiangyun Ding, Yan Gu
University of California, Riverside
xding047@ucr.edu, ygu@cs.ucr.edu

Yihan Sun
UC Riverside
yihans@cs.ucr.edu

CP5

Zip-Tries: Simple Dynamic Data Structures for Strings

In this paper, we introduce zip-tries, which are simple, dynamic, memory-efficient data structures for strings. Zip-tries support search and update operations for k-length strings in $O(k+\log n)$ time in the standard RAM model or in $O(k/\alpha+\log n)$ time in the word RAM model, where α is the length of the longest string that can fit in a memory word, and n is the number of strings in the trie. Importantly, we show how zip-tries can achieve this while only requiring $O(\log\log n+\log\log\frac{k}{\alpha})$ bits of metadata per node w.h.p., which is an exponential improvement over previous results for long strings. Despite being considerably simpler and more memory efficient, we show how zip-tries perform competitively with state-of-the-art data structures on large datasets of long strings. Furthermore, we provide a simple, general framework for parallelizing string comparison operations in linked data structures, which we apply to zip-tries to obtain parallel zip-tries. Parallel zip-tries are able to achieve good search and update performance in parallel, performing them in $O(\log n)$ span. We apply our techniques to an existing external-memory string data structure, the string B-tree, obtaining a parallel string B-tree which performs search operations in only $O(\log n/\log\log n)$ span under the practical PRAM model. We provide LCP-aware variants of all our algorithms that are efficient in practice, which we justify empirically.

David Eppstein, Ofek Gila, Michael Goodrich, Ryuto Kitagawa
University of California, Irvine
david.eppstein@gmail.com, ofek.gila@gmail.com, goodrich@acm.org, ryutok@uci.edu

CP5

Online Distributed Queue Length Estimation

Queue length monitoring is a commonly arising problem in numerous applications such as queue management systems, scheduling, and traffic monitoring. Motivated by such applications, we formulate a queue monitoring problem, where there is a FIFO queue with arbitrary arrivals and departures, and a server needs to monitor the length of a queue by using *decentralized* pings from packets in the queue. Packets can send pings informing the server about the number of packets *ahead* of them in the queue. Via novel online policies and lower bounds, we tightly characterize the trade-off between the number of pings sent and the accuracy of the server's real time estimates. Our work studies the trade-off under various arrival and departure processes, including constant-rate, Poisson, and adversarial processes.

Aditya Bhaskara
Google Research NYC
bhaskaraaditya@gmail.com

Sreenivas Gollapudi
Google Research
sgollapu@google.com

Sungjin Im
UC Santa Cruz
sim9@ucsc.edu

Kostas Kollias
Google Research
kostaskollias@google.com

Kamesh Munagala
Duke University
munagala@duke.edu

CP5

Parallel kd-tree with Batch Updates

The kd-tree is one of the most widely used data structures to manage multi-dimensional data. With increas-

ing data volume, it is imperative to consider parallelism in kd-trees. However, we observed challenges in existing parallel kd-tree implementations, for both constructions and updates. The goal of this paper is to develop efficient in-memory kd-trees by supporting high parallelism and cache-efficiency. We propose the Pkd-tree, a parallel kd-tree that is efficient both in theory and in practice. The Pkd-tree supports parallel tree construction, batch update (insertion and deletion), and various queries including k-nearest neighbor search, range query, and range count. We proved that our algorithms have strong theoretical bounds in work (sequential time complexity), span (parallelism), and cache complexity. Our key techniques include 1) an efficient construction algorithm that optimizes work, span, and cache complexity simultaneously, and 2) reconstruction-based update algorithms that guarantee the tree to be weight-balanced. With the new algorithmic insights and careful engineering effort, we achieved a highly optimized implementation of the Pkd-tree. We tested Pkd-tree with various synthetic and real-world datasets. We compare the Pkd-tree with state-of-the-art parallel kd-tree implementations. In all tests, with better or competitive query performance, Pkd-tree is much faster in construction and updates consistently than all baselines.

Ziyang Men, Zheqi Shen, Yan Gu
University of California, Riverside
zmen002@ucr.edu, zshen055@ucr.edu, ygu@cs.ucr.edu

Yihan Sun
UC Riverside
yihans@cs.ucr.edu

CP6

PECANN: Parallel Efficient Clustering with Graph-Based Approximate Nearest Neighbor Search

In this paper, we study variants of density peaks clustering, a popular type of density-based clustering algorithm for points that has been shown to work well in practice. Our goal is to cluster *large high-dimensional* datasets, which are prevalent in practice. Prior solutions are either sequential and cannot scale to large data, or are specialized for low-dimensional data. This paper unifies the different variants of density peaks clustering into a single framework, PECANN (**P**arallel **E**fficient **C**lustering with **A**pproximate **N**earest **N**eighbors), by abstracting out several key steps common to this class of algorithms. One such key step is to find nearest neighbors that satisfy a predicate function, and one of the main contributions of this paper is an efficient way to do this predicate search using graph-based approximate nearest neighbor search (ANNS). To provide ample parallelism, we propose a doubling search technique that enables points to find an approximate nearest neighbor satisfying the predicate in a small number of rounds. Our technique can be applied to many existing graph-based ANNS algorithms, which can all be plugged into PECANN.

Shangdi Yu, Joshua Engels, Yihao Huang, Julian Shun
MIT
shangdiy@mit.edu, jengels@mit.edu, yh_huang@mit.edu, jshun@mit.edu

CP6

Parallel k-Core Decomposition: Theory and Practice

We study parallel k-core decomposition and propose a simple, work-efficient framework adaptable to various optimizations. To enhance parallelism, we introduce a sampling scheme to reduce contention and vertical granularity control (VGC) to mitigate scheduling overhead. We also design a hierarchical bucket structure to serve as a more adaptive bucketing structure for large-coreness graphs. Our approach is evaluated on diverse graphs, demonstrating efficiency and scalability.

Youzhe Liu
UC Riverside
yliu908@ucr.edu

Xiaojun Dong, Yan Gu
University of California, Riverside
xdong038@ucr.edu, ygu@cs.ucr.edu

Yihan Sun
UC Riverside
yihans@cs.ucr.edu

CP6

Deterministic Parallel High-Quality Hypergraph Partitioning

We present a deterministic parallel multilevel algorithm for balanced hypergraph partitioning that matches the state of the art for non-deterministic algorithms. Deterministic parallel algorithms produce the same result in each invocation, which is crucial for reproducibility. Moreover, determinism is highly desirable in application areas such as VLSI design. While there has been tremendous progress in parallel hypergraph partitioning algorithms recently, deterministic counterparts for high-quality local search techniques are missing. Consequently, solution quality is severely lacking in comparison to the non-deterministic algorithms. In this work we close this gap. First, we present a generalization of the recently proposed Jet refinement algorithm. While Jet is naturally amenable to determinism, significant changes are necessary to achieve competitive performance on hypergraphs. We also propose an improved deterministic rebalancing algorithm for Jet. Moreover, we consider the powerful but slower flow-based refinement and introduce a scheme that enables deterministic results while building upon a non-deterministic maximum flow algorithm. As demonstrated in our thorough experimental evaluation, this results in the first deterministic parallel partitioner that is competitive to the highest quality solvers. With Jet refinement, we match or exceed the quality of Mt-KaHyPar's non-deterministic default configuration while being only 15% slower on average.

Robert Krause
Karlsruhe Institute of Technology
robert.krause@student.kit.edu

Lars Gottesbüren
Google Research
gottesbueren@google.com

Nikolai Maas
Karlsruhe Institute of Technology (KIT)
nikolai.maas@kit.edu

CP6

Towards Efficient Methods for Approximate

Weighted Matching Applications

As a classical graph problem prevalent across a variety of fields, matching, or the linear assignment problem, seeks to map one set of discrete entities to another. This mapping is generated within a wide range of contexts, whether that be instances of residents to hospitals, computer vision pattern recognition, or numerical linear algebra applications, among countless others. Specifically, we consider maximum weighted matching (MWM) instances. In modern applications as input graph sizes become increasingly large, demands for efficient, HPC-enabled approximate matching solutions are high. These solutions often face inefficiencies through irregularity in real-world graph data, high data movement costs and memory restrictions on device, among others. Despite this, there has been significant work using CPU parallel platforms, GPU hardware, or some combination of the two. Analyzing these instances, we take notice of an approaching 'performance wall' in traditional $\frac{1}{2}$-approximate MWM approaches. Thus, we aim to further advancements through alternative platforms and methods for efficiency in application. Our goals here are two-fold: (1) improve communication costs of HPC-enabled approximate matching implementations through a block-partition-based 2D sparse-communication grid; (2) explore efficient incorporations of such approximate methods in application, namely in data joining techniques. This work highlights each contribution towards these goals.

Michael Mandulak
RPI
mandum@rpi.edu

Sayan Ghosh, S M Ferdous, Mahantesh Halappanavar
Pacific Northwest National Laboratory
sayan.ghosh@pnnl.gov, sm.ferdous@pnnl.gov,
hala@pnnl.gov

George Slota
Rensselaer Polytechnic Institute
slotag@rpi.edu

CP6
Shared-Memory Hierarchical Process Mapping

Modern large-scale scientific applications consist of thousands to millions of individual tasks. These tasks involve not only computation but also communication with one another. Typically, the communication pattern between tasks is sparse and can be determined in advance. Such applications are executed on supercomputers, which are often organized in a hierarchical hardware topology, consisting of islands, racks, nodes, and processors, where processing elements reside. To ensure efficient workload distribution, tasks must be allocated to processing elements in a way that ensures balanced utilization. However, this approach optimizes only the workload, not the communication cost of the application. It is straightforward to see that placing groups of tasks that frequently exchange large amounts of data on processing elements located near each other is beneficial. The problem of mapping tasks to processing elements considering optimization goals is called process mapping. In this work, we focus on minimizing communication cost while evenly distributing work. We present the first shared-memory algorithm that utilizes hierarchical multisection to partition the communication model across processing elements. Our parallel approach achieves the best solution on 95 percent of instances while also being marginally faster than the next best algorithm. Even in a serial setting, it delivers the best solution quality while also outperforming previous serial algorithms in speed.

Christian Schulz
Heidelberg University, Germany
christian.schulz@informatik.uni-heidelberg.de

Henning Woydt
Heidelberg University
henning.woydt@informatik.uni-heidelberg.de

CP7
Algorithmic Differentiation and Vertex Elimination on Computational Hypergraphs

Algorithmic differentiation is a method for computing derivatives of functions defined by computer codes. It systematically applies the chain rule to the operations executed during a program evaluation, typically represented by a directed acyclic graph. The vertices of this graph correspond to intermediate scalar values computed and stored during the program execution, and edges represent the data dependencies between these values. Accumulating derivatives involves successively eliminating vertices in a specific order within this graph. While the graph model captures the situation where the program executes scalar-valued operations with one or two input scalars, it does not immediately allow for raising the abstraction to a coarser level. The contribution of this manuscript is a novel combinatorial model that is capable of representing algorithmic differentiation for more general operations. Specifically, the model accommodates operations with multiple-input and multiple-output relations as well as intermediate values that are non-scalar. The combinatorial model is based on computational hypergraphs, a novel class of directed hypergraphs where vertices represent non-scalar intermediate values and hyperedges correspond to operations involving multiple inputs and outputs. Accumulating derivatives then corresponds to vertex elimination within these computational hypergraphs. In contrast to the graph model, the new elimination rules in the hypergraph model differ significantly.

H. Martin Bücker, Torsten F. Bosse
Friedrich-Schiller-Universität Jena
matrin.buecker.uni-jena.de, torsten.bosse@uni-jena.de

CP7
Algorithmic Differentiation: Vertex Elimination in Dags

We consider graph-theoretic formulations of two problems with wide application in algorithmic differentiation: accumulating a Jacobian matrix while minimizing multiplications (Minimum Cost), and obtaining a minimum-size matrix-free Jacobian representation (Minimum Representation). Both problems ask for a sequence of vertex eliminations in a directed acyclic graph; though a variety of heuristics used in practice are based on these formulations, relatively little is known about their complexity. We prove that both problems are NP-complete, and give an $O^*(2^n)$ algorithm for Minimum Cost. Further, we prove that this running time is essentially the best possible under the Exponential Time Hypothesis. Our results are facilitated by several illustrative structural insights about sequences of vertex eliminations, which also lead to a novel SAT-encoding for Minimum Cost. We conclude with a discussion of open problems related to parameterized algorithms, approximations, and problem variants allowing edge eliminations. In addition to sharing our results, our intent is

to spark symbiotic conversations between the algorithmic differentiation and graph algorithms communities.

Alex Crane
University of Utah
alex.crane@utah.edu

Matthias Bentert, Pal Drange Drange
University of Bergen
matthias.bentert@uib.no, pal.drange@uib.no

Yosuke Mizutani, Blair D. Sullivan
University of Utah
yos@cs.utah.edu, sullivan@cs.utah.edu

CP7

Combining Bayesian Probing and Bloom Filters to Determine Jacobian and Hessian Sparsity Patterns

Many techniques for the efficient computation of sparse derivative matrices (Jacobians and Hessians) require knowing the sparsity pattern of the matrix. One of the two main methods for determining the sparsity pattern relies on propagating bit vectors through a computation. In the naive version of bit vector probing, each bit represents one independent variable (column of the derivative matrix). However, Griewank and Mitev showed that one can determine the sparsity pattern with fewer bit probes by using carefully selected probes and Bayes theorem. We previously demonstrated that one can also reduce the number of bit probes by using randomized probes based on Bloom filters. In this work, we combine Bayesian probing and Bloom filter probing to overcome the shortcomings of each method in isolation. We also examine how to use symmetry in determining the sparsity pattern of Hessian matrices and how to estimate the number of nonzeros per row.

Paul D. Hovland
Argonne National Laboratory
MCS Division
hovland@anl.gov

CP7

Scheduled Jacobian Chaining

We address the efficient computation of Jacobian matrices for programs composed of sequential differentiable subprograms. By representing the overall Jacobian as a chain product of the Jacobians of these subprograms, we reduce the problem to optimizing the sequence of matrix multiplications, known as the Jacobian Matrix Chain Product problem. Solutions to this problem yield "optimal bracketings", which induce a precedence-constraint scheduling problem. We investigate the inherent parallelism in the solutions and develop a new dynamic programming algorithm as a heuristic that incorporates scheduling. To assess its performance, we benchmark it against the global optimum, which is computed via a branch-and-bound algorithm.

Simon Märtens, Uwe Naumann
RWTH Aachen University, Germany
maertens@stce.rwth-aachen.de, naumann@stce.rwth-aachen.de

CP7

Heuristic Graph Colouring Using Hypergraph Representations

We revisit the classic Graph Colouring problem in the context of column partitioning of matrices. In this context, we are given a hypergraph and the aim is to colour the vertices of the hypergraph by a minimum number of colours such that in each hyperedge every colour occurs at most once. We provide a comprehensive implementation and experimental evaluation of combinations of known heuristical techniques such as vertex orderings for greedy colouring, vertex reordering after an initial colouring, and recolouring of vertices to save colours. In addition, we provide a new stronger recolouring method that is based on the hypergraph view of the problem. Our implementation is specifically tailored towards efficiency on hypergraph instances. We compare our implementation with the *ColPack* library and find that several heuristics not contained in *ColPack* yield solutions with fewer colours at the cost of higher but still affordable running times. We complement these results by a theoretical analysis of the problem complexity on hypergraphs that can be covered by few hyperedges.

Jurek Rostalsky, Hanns Martin Bücker, Christian Komusiewicz
Friedrich-Schiller-University Jena
jurek.rostalsky@uni-jena.de, buecker@acm.org, c.komusiewicz@uni-jena.de

CP8

Theoretically and Practically Efficient Parallel Algorithms for Semisort and Integer Sort

This paper studies parallel algorithms for semisorting and integer sorting from both theoretical and practical perspectives. In theory, we prove tighter bounds for a class of existing practical integer sorting algorithms. In practice, we design new semisorting and integer sorting algorithms that outperform SOTA baselines and achieve strong self-relative speedups.

Xiaojun Dong
University of California, Riverside
xdong038@ucr.edu

Laxman Dhulipala
University of Maryland, College Park, MD
laxman@umd.edu

Yan Gu
University of California, Riverside
ygu@cs.ucr.edu

Yihan Sun
UC Riverside
yihans@cs.ucr.edu

CP8

Subset Selection Using Nuclear Scores: Algorithms and Theory for Nystrom Approximation, Cur Decomposition, and Graph Laplacian Reduction

Column selection is an essential tool for structure-preserving low-rank approximation, with wide-ranging applications across many fields, such as data science, machine learning, and theoretical chemistry. In this work, we develop unified methodologies for fast, efficient, and theoretically guaranteed column selection. First we derive and implement a sparsity-exploiting deterministic algorithm applicable to tasks including kernel approximation and CUR

decomposition. Next, we develop a matrix-free formalism relying on a randomization scheme satisfying guaranteed concentration bounds, applying this construction both to CUR decomposition and to the approximation of matrix functions of graph Laplacians. Importantly, the randomization is only relevant for the computation of the scores that we use for column selection, not the selection itself given these scores. For both deterministic and matrix-free algorithms, we bound the performance favorably relative to the expected performance of determinantal point process (DPP) sampling and, in select scenarios, that of exactly optimal subset selection. The general case requires new analysis of the DPP expectation. Finally, we demonstrate strong real-world performance of our algorithms on a diverse set of example approximation tasks including tensor interpolative decomposition and Markov chain reduced order modeling.

Mark Fornace
Lawrence Berkeley National Laboratory
mefornace@lbl.gov

Michael Lindsey
University of California, Berkeley
lindsey@berkeley.edu

CP8

Covering Maximal Cliques in Real-World Graphs with Dense Subgraphs

Listing dense subgraphs of a given graph is a widely studied problem with practical applications in social network analysis, bioinformatics, topological graph analysis, fraud detection and others. A fundamental challenge is the fact that the number of dense subgraphs – and even the number of maximal cliques – of a graph can be exponential in the number of vertices. In order to address this difficulty, we reformulate the problem as follows: can a small set of dense subgraphs capture the maximal cliques of a graph? We call a collection $\mathcal{C} = \{C_i\}$ of subgraphs a ρ-dense clique container of the graph if (a) every maximal clique is contained in some C_i and (b) the edge density of each C_i is $\geq \rho$. We prove that for a constant ρ, given a graph with n vertices and polylogarithmic degeneracy, there exists a ρ-dense clique container of size $n^{1+o(1)}$. Further, we consider the class of weakly c-closed graphs, which model the real-world phenomenon of triadic closure, and show they also have ρ-dense clique containers whose size is exponentially smaller than the number of maximal cliques (which itself is polynomial in n but exponential in c). We complement these results with simple algorithms that construct such clique containers and experimentally demonstrate that their sizes are surprisingly small in the case of real-world graphs.

Shweta Jain
University of Utah
shweta.jain@utah.edu

Sabyasachi Basu
University of California, Santa Cruz
sbasu3@ucsc.edu

Haim Kaplan
Tel Aviv University, Google Research
haimk@tauex.tau.ac.il

Jakub Lacki
Google Research
jlacki@google.com

Blair D. Sullivan
University of Utah
blair.sullivan@utah.edu

CP8

Spectral Sparsification for Protein Family Identification

Identifying novel protein families from metagenomic datasets is a critical endeavor in biology and bioinformatics due to its potential to expand our understanding of microbial diversity, uncover new biological functions, and drive innovation in biotechnology. Many newly identified protein families are associated with unique functions tailored to specific environments, such as extreme habitats. Novel proteins can serve as templates for developing industrial enzymes, biofuels, or bioremediation tools. Metagenomics has already yielded enzymes with unique properties that are valuable for biotechnology. In the pipeline for solving protein family identification problem, the SSN produced is often very large: prior work studies graphs with half a billion vertices and more than 5 billion edges. The application would benefit from generating a denser version of this graph, but we are limited by the resource requirements of HipMCL. Specifically, the worst-case time complexity of MCL clustering grows quadratically in the number of edges in the graph, so denser graphs take longer to cluster. This leads us to our goal: a practical, highly scalable method for producing a sparsifier of the SSN which is designed to preserve Markov clustering quality. We could then use HipMCL to cluster this sparsifier, increasing the scale at which protein family identification can be performed.

Tianyu Liang
UC Berkeley
tianyul@berkeley.edu

David Tench, Yotam Yaniv
Lawrence Berkeley National Lab
dtench@lbl.gov, yotamy@lbl.gov

Aydin Buluc
Lawrence Berkeley National Laboratory
abuluc@lbl.gov

Xiaoye S. Li
Computational Research Division
Lawrence Berkeley National Laboratory
xsli@lbl.gov

CP8

Scaling Graph Connectivity to Hundreds of Billions of Edges on Hundreds of GPUs

Graph connectivity is a basic but important graph problem, used as a graph analytic, a preprocessing step for more complex algorithms, and within related fields such as linear algebra. This broad applicability has made graph connectivity one of the most studied problems in the discrete algorithms field. As such, there have been a plethora of parallel algorithms, variants, and optimizations proposed in the literature. We designate these algorithms as falling into four broad classes, and we consider for implementation a representative algorithm from each class within a distributed multi-GPU graph processing framework. We design efficient communication and workload processing methods for these algorithms, and we introduce novel optimizations. Our approach strong scales all the way to

256 GPUs on graphs ranging from hundreds of millions to hundreds of billions of edges in size. We present the fastest time-to-solution for graph connectivity in the literature for the largest publicly available dataset - the 128 billion-edge Web Data Commons 2012 (WDC12) crawl. We also show performant weak scaling on graphs up to 4x larger than WDC12.

George Slota
Rensselaer Polytechnic Institute
gmslota@gmail.com

Michael Mandulak
RPI
mandum@rpi.edu

Ujwal Pandey
Rensselaer Polytechnic Institute
ujpan314@gmail.com

MT1
Scalable and Practical Graph Clustering

We present recent results on scalable graph clustering algorithms in the parallel, distributed and dynamic settings. We focus on several prominent applications of graph clustering: near-duplicate detection, anomaly detection and data partitioning, and demonstrate algorithmic advancements that enable scalable algorithms for each of them. A particular success story that we highlight is scaling hierarchical agglomerative clustering to handle trillion edge graphs. The theoretical and practical insights that we present are drawn from the experience of the Graph Mining team, which has built a graph clustering library powering over a hundred applications across Google.

Jakub Lacki
Google Research
jlacki@google.com

MT2
Learning-Augmented Algorithms

To Come

Quanquan C. Liu
Yale University
quanquan.liu@yale.edu

Vaidehi Srinivas
Northwestern University
vaidehi@u.northwestern.edu