Recent Trends in Graph Decomposition
Dagstuhl 23331, Schedule

Motivation

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

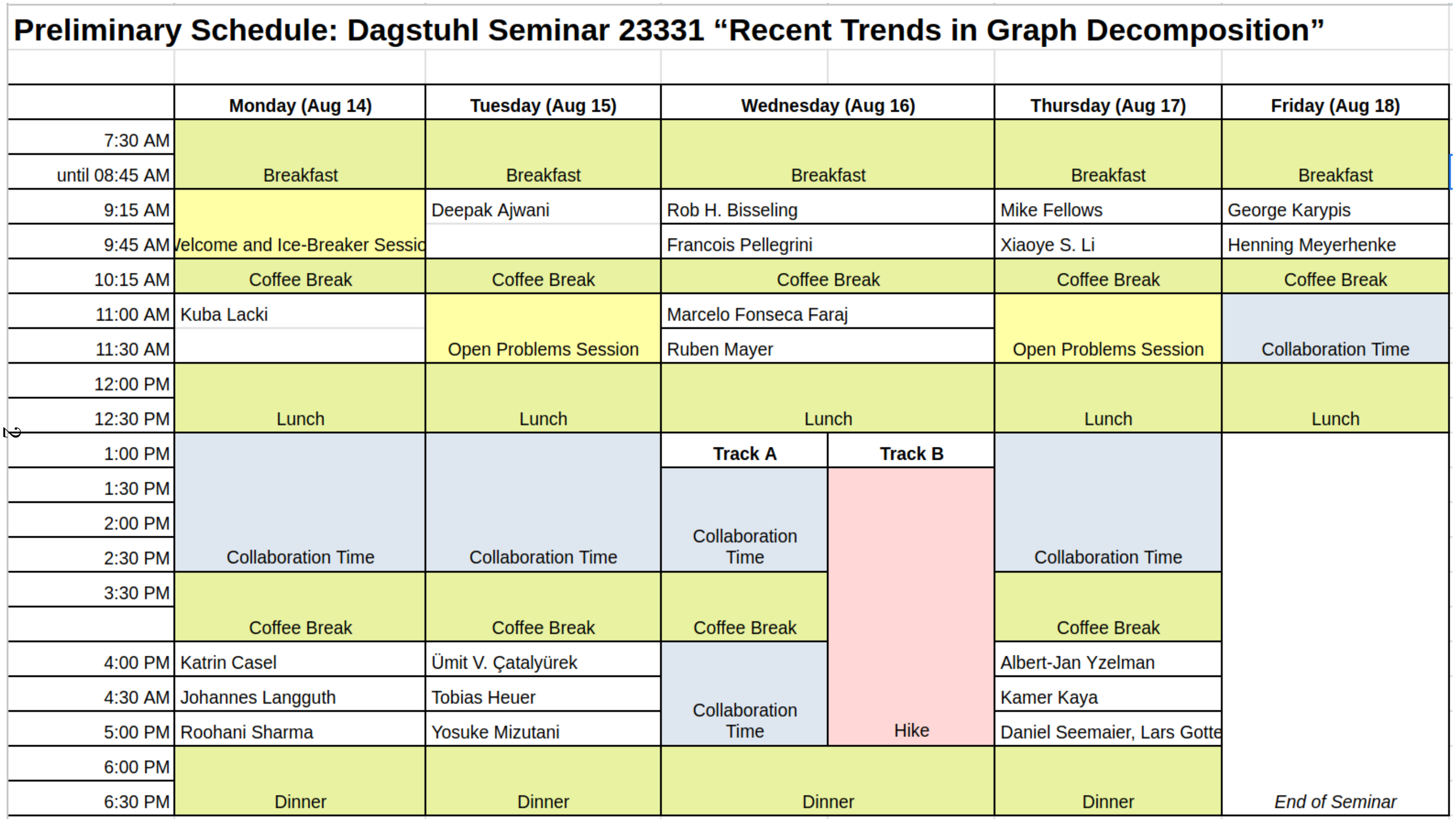
Hardware Design for Dealing with Graphs. Modern processors are optimized for computations that are floating point intensive and have regular memory accesses. Unfortunately, the computations performed by sparse graph algorithms do not benefit from such optimizations. To address this mismatch, new processor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

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

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

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

Schedule



Preliminary Schedule: Dagstuhl Seminar 23331 “Recent Trends in Graph Decomposition”

	Monday (Aug 14)	Tuesday (Aug 15)	Wednesday (Aug 16)		Thursday (Aug 17)	Friday (Aug 18)
7:30 AM	Breakfast	Breakfast	Breakfast		Breakfast	Breakfast
until 08:45 AM						
9:15 AM	Welcome and Ice-Breaker Sessio	Deepak Ajwani	Rob H. Bisseling		Mike Fellows	George Karypis
9:45 AM			Francois Pellegrini		Xiaoye S. Li	Henning Meyerhenke
10:15 AM	Coffee Break	Coffee Break	Coffee Break		Coffee Break	Coffee Break
11:00 AM	Kuba Lacki	Open Problems Session	Marcelo Fonseca Faraj		Open Problems Session	Collaboration Time
11:30 AM			Ruben Mayer			
12:00 PM	Lunch	Lunch	Lunch		Lunch	Lunch
12:30 PM						
1:00 PM	Collaboration Time	Collaboration Time	**Track A**	**Track B**	Collaboration Time	*End of Seminar*
1:30 PM			Collaboration Time	Hike		
2:00 PM						
2:30 PM						
3:30 PM	Coffee Break	Coffee Break	Coffee Break		Coffee Break	
4:00 PM	Katrin Casel	Ümit V. Çatalyürek	Collaboration Time		Albert-Jan Yzelman	
4:30 AM	Johannes Langguth	Tobias Heuer			Kamer Kaya	
5:00 PM	Roohani Sharma	Yosuke Mizutani			Daniel Seemaier, Lars Gotte	
6:00 PM	Dinner	Dinner	Dinner		Dinner	
6:30 PM						

Abstracts

Monday (Aug 14)

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

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

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

Presenter: Roohani Sharma
Title: Parameterized Approximation Schemes for Clustering with General Norm Objectives
Abstract: We consider the well-studied algorithmic regime of designing a $(1 + \epsilon)$-approximation algorithm for a k-clustering problem that runs in time $f(k, \epsilon)\text{poly}(n)$. Our main contribution is a clean and simple EPAS that settles more than ten clustering problems (across multiple well-studied objectives as well as metric spaces) and unifies well-known

EPASes. Our algorithm gives EPASes for a large variety of clustering objectives (for example, k-means, k-center, k-median, priority k-center, ℓ-centrum, ordered k-median, socially fair k-median aka robust k-median, or more generally monotone norm k-clustering) and metric spaces (for example, continuous high-dimensional Euclidean spaces, metrics of bounded doubling dimension, bounded treewidth metrics, and planar metrics). Key to our approach is a new concept that we call bounded ϵ-scatter dimension–an intrinsic complexity measure of a metric space that is a relaxation of the standard notion of bounded doubling dimension.

Tuesday (Aug 15)

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

Presenter: Tobias Heuer
Title: Using Steiner Trees in Hypergraph Partitioning
Abstract: Minimizing wire-lengths is one of the most important objectives in the realization of modern circuits. The design process involves initially placing the logical units (cells) of a circuit onto a physical layout, and subsequently routing the wires to connect the cells. Hypergraph partitioning (HGP) has been long used as a placement strategy in this process. However, it has been replaced by other methods due to limitations of existing objective functions for HGP, which only minimizes wire-lengths implicitly. In this talk, we present a novel HGP formulation that maps a hypergraph H, representing a logical circuit, onto a routing layout represented by a weighted graph G. The objective is to minimize the total length of all wires induced by the hyperedges of H on G. To capture wire-lengths, we compute minimal Steiner trees - a metric commonly used in routing algorithms. For this formulation, we present a direct k-way multilevel algorithm that we integrate into the shared-memory hypergraph partitioner Mt-KaHyPar. Mt-KaHyPar is a highly scalable partitioning algorithm that achieves the same solution quality as the best sequential algorithms, while being an order of magnitude faster with only ten threads. Our

experiments demonstrate that our new algorithm achieves an improvement in the Steiner tree metric by 7% (median) on VLSI instances when compared to the best performing partitioning algorithm that optimizes the mapping in a postprocessing step. Although computing Steiner trees is an NP-hard problem, we achieve this improvement with only a 2–3 times slowdown in partitioning time compared to optimizing the connectivity metric.

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

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

Wednesday (Aug 16)

Presenter Rob H. Bisseling (joint work with Engelina Jenneskens)
Title: Exact k-way sparse matrix partitioning

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

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

Presenter: Marcelo Fonseca Faraj
Title: Recent Advances in Streaming (Hyper)Graph Decomposition
Abstract: There is a gap in (hyper)graph decomposition algorithms. Streaming algorithms, which are adaptable to small machines, partition huge (hyper)graphs quickly, but yield low-quality results. Conversely, in-memory algorithms produce high-quality solutions but require significant memory. Our talk explores recent advances in streaming (hyper)graph decomposition. We begin with streaming graph partitioning, covering from hash-based to buffered approaches. Next, we discuss the state of the art in streaming process mapping. Finally, we present recent advances in streaming hypergraph partitioning. With their recent introduction and potential for further improvement, these families of streaming algorithms present unexplored avenues for improvement, while their recent strong results can provide new insights into solving the respective in-memory versions of these problems.

Presenter: Ruben Mayer
Title: Graph partitioning and distributed graph processing – An end-to-end optimization perspective
Abstract: Graph partitioning is often considered as a necessary preprocessing step for distributed graph processing. In doing so, the partitioning quality in terms of cut size and balancing is crucial to the performance of distributed graph processing jobs. However, yielding high graph partitioning quality is a challenging and compute-intensive problem. Many different graph partitioning algorithms have been proposed, which differ both in their achieved partitioning quality as well as their computational costs. How many resources

and how much time to invest into partitioning depends on various factors such as the graph size, the resource budget of the user, and the complexity and run-time of subsequent graph processing on the partitioned graph. In my talk, I will elaborate on the problem of optimizing the end-to-end graph processing pipeline.

Thursday (Aug 17)

Presenter: Mike Fellows
Title: Parameterized complexity and algorithmics — some horizons — and the universal applied paradigm of diverse solutions
Abstract: The talk will describe in a very accessible way the foundational motivations and brief formal setup of parameterized complexity, a name which has too many syllables, but it is pretty straightforward and can be compared to ”coordinatized geometry”. It has always been nurtured as a theory, with the aspiration to be useful in practice. It goes beyond the one-dimensional P versus NP framework in a quite simple way, much as coordinatized geometry went beyond the Geometry of the Greeks. One natural way to usefully deploy parameterization is to address the working reality of many computing applications domains: that a single mathematically optimal solution is not what is really wanted! In many practical computing applications areas, what is really wanted is a moderate-sized collection of quality-wise pretty good solutions to choose from, often on the basis of side information not included in the strict optimal mathematical model. The talk is based on a recent IJCAI paper with multiple authors that began this direction of research, that fits very neatly with the mathematical algorithm design tools of parameterized complexity.

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

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

the art in combinatorial scientific computing algorithms –such as partitioners or schedulers– for improving the performance of ALP-based programs.

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

Presenter: Daniel Seemaier, Lars Gottesbüren
Title: Recent Advances in Ka (Hyper)Graph Partitioning
Abstract: In the first part of our talk, we will present a brief overview of our two main solvers Mt-KaHyPar and KaMinPar for balanced (hyper)graph partitioning. Until recently, there was a severe gap in terms of solution quality between sequential and parallel solvers. We have since parallelized all of the techniques in sequential solvers that lead to high solution quality, which are now available in Mt-KaHyPar. Moreover, we tackled new problem domains such as efficiently partitioning into a very large number of clusters with the deep multilevel scheme, which is available in KaMinPar. Additionally, KaMinPar was recently extended with a high-performance MPI mode, adapting the deep multilevel scheme to the distributed scenario. In the second part of our talk, we will then present some currently ongoing work. Traditionally, local search algorithms perform only balance-preserving moves. Perhaps unsurprisingly, we can achieve huge quality improvements by permitting large balance violations; performing unconstrained moves with some caution and rebalancing the solution later on. The same quality gap that used to exist between sequential and shared-memory parallel, now exists between shared-memory parallel and distributed-memory partitioners. Therefore, we are working on a distributed FM version in KaMinPar, as well as a distributed version of the recently proposed JET algorithm, both of which show promising results in preliminary experiments.