Report from Dagstuhl Seminar 18101

Scheduling

Edited by
Magnús M. Halldórsson[1], Nicole Megow[2], and Clifford Stein[3]

1 **Reykjavik University, IS,** `mmh@ru.is`
2 **Universität Bremen, DE,** `nicole.megow@uni-bremen.de`
3 **Columbia University, US,** `cliff@ieor.columbia.edu`

Abstract

This report documents the program and outcomes of the Dagstuhl Seminar 18101 "Scheduling" in March 2018. The seminar brought together algorithmically oriented researchers from two communities with interests in resource management: (i) the scheduling community and (ii) the networking and distributed computing community. The primary objective of the seminar was to expose each community to the important problems and techniques from the other community, and to facilitate dialog and collaboration between researchers.

Seminar March 4–9, 2018 – https://www.dagstuhl.de/18101
2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization, Networks → Network algorithms, Theory of computation → Approximation algorithms analysis, Theory of computation → Scheduling algorithms
Keywords and phrases scheduling, optimization, approximation algorithms
Digital Object Identifier 10.4230/DagRep.8.3.1
Edited in cooperation with Lukas Nölke

1 Executive Summary

Magnús M. Halldórsson
Nicole Megow
Clifford Stein

This fifth meeting in a series of Dagstuhl "Scheduling" seminars brought together part of the community of algorithmic researchers who focus on scheduling, and part of the community of algorithmic researchers who focus on networking in general, and resource management within networks in particular. These communities are far from unknown to each other as they attend the same general academic conferences. But as each community has its own specialized conferences, there is less interaction between these communities than there should be. Further there are differences in the types of algorithmic problems these communities are naturally drawn towards.

The primary objective of the seminar was to expose each community to the important models, problems and techniques from the other community, and to facilitate dialog and collaboration between researchers. The program included 22 invited main talks including an inspiring talk on practical applications at ABB Corporate Research, 8 short spot-light talks, two open problem sessions in the beginning of the week, and ample unstructured time for research and interaction. The overall atmosphere among the 44 participants was very interactive.

A highlight of the seminar was a joint Wednesday-session with the Dagstuhl Seminar 18102 "Dynamic Traffic Models in Transportation Science". It was a fortunate coincidence that both seminars were scheduled in parallel. Indeed, questions related to networks, scheduling and resource sharing arise naturally in traffic control and transportation science. It was an inspiring secondary outcome of the workshop to realize this strong overlap in interests which led to interesting discussions between researchers of the different communities.

2 Table of Contents

3 Overview of Talks

3.1 Improved Online Algorithm for Weighted Flow Time

Yossi Azar (Tel Aviv University, IL)

Joint work of Yossi Azar, Noam Touitou
Main reference Yossi Azar, Noam Touitou: "Improved Online Algorithm for Weighted Flow Time", CoRR, Vol. abs/1712.10273, 2017.
URL http://arxiv.org/abs/1712.10273

We discuss one of the most fundamental scheduling problem of processing jobs on a single machine to minimize the weighted flow time (weighted response time). Our main result is a $O(\log P)$-competitive algorithm, where P is the maximum-to-minimum processing time ratio, improving upon the $O(\log^2 P)$-competitive algorithm of Chekuri, Khanna and Zhu (STOC 2001). We also design a $O(\log D)$-competitive algorithm, where D is the maximum-to-minimum density ratio of jobs. Finally, we show how to combine these results with the result of Bansal and Dhamdhere (SODA 2003) to achieve a $O(\log(\min(P, D, W)))$-competitive algorithm (where W is the maximum-to-minimum weight ratio), without knowing P, D, W in advance. As shown by Bansal and Chan (SODA 2009), no constant-competitive algorithm is achievable for this problem.

3.2 Scheduling Under Uncertainty In Safety-critical Systems

Sanjoy K. Baruah (Washington University, US)

Many safety-critical system designs are subject to validation (in some cases, certification) prior to deployment. Consequently, routing and resource-allocation decisions for such systems may need to be made prior to run-time, with incomplete knowledge of the actual conditions that will be encountered by the system during run-time. I will discuss some open scheduling and routing problems that arise in the analysis of such safety-critical real-time systems as a consequence of needing to make decisions in the presence of this uncertainty.

3.3 On Minimizing the Makespan with Bag Constraints

Syamantak Das (IIIT - New Dehli, IN)

Joint work of Syamantak Das, Andreas Wiese
Main reference Syamantak Das, Andreas Wiese: "On Minimizing the Makespan When Some Jobs Cannot Be Assigned on the Same Machine", in Proc. of the 25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria, LIPIcs, Vol. 87, pp. 31:1–31:14, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.ESA.2017.31

We study the classical scheduling problem of assigning jobs to machines in order to minimize the makespan. It is well-studied and admits an EPTAS on identical machines and a $(2-1/m)$-approximation algorithm on unrelated machines. In this work we study a variation in which the input jobs are partitioned into bags and no two jobs from the same bag are allowed to be assigned on the same machine. Such a constraint can easily arise, e.g., due to system stability and redundancy considerations. Unfortunately, as we demonstrate in this work, the techniques of the above results break down in the presence of these additional constraints. Our first result is a PTAS for the case of identical machines. It enhances the methods from the known (E)PTASs by a finer classification of the input jobs and careful argumentations why a good schedule exists after enumerating over the large jobs. For unrelated machines, we prove that there can be no $(\log n)^{1/4-\epsilon}$-approximation algorithm for the problem for any $\epsilon > 0$, assuming that NP $\subseteq$ ZPTIME$\cdot(2^{(logn)^{O(1)}})$. This holds even in the restricted assignment setting. However, we identify a special case of the latter in which we can do better: If the same set of machines we give an 8-approximation algorithm. It is based on rounding the LP-relaxation of the problem in phases and adjusting the residual fractional solution after each phase to order to respect the bag constraints.

3.4 Fairness, Congestion Control, and Related Open Problems

Jelena Diakonikolas (Boston University, US)

Fairness is a central topic in a variety of areas, ranging from political philosophy, economic theory, and operations research to network congestion control, and more recently, machine learning. The purpose of this talk is three-fold: (i) to give a historical overview of different philosophical approaches to fairness, (ii) to formally introduce fair resource allocation problems and discuss different algorithmic approaches to solving them in the context of (offline) network congestion control, and (iii) to discuss related open problems in online optimization and scheduling.

3.5 Internet Transport Service using Dissemination Graphs, and the Shallow-Light Steiner Network Problem

Michael Dinitz (Johns Hopkins University - Baltimore, US)

Joint work of Amy Babay, Emily Wagner, Yair Amir, Zeyu Zhang
Main reference Amy Babay, Emily Wagner, Michael Dinitz, Yair Amir: "Timely, Reliable, and Cost-Effective Internet Transport Service Using Dissemination Graphs", in Proc. of the 37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5-8, 2017, pp. 1–12, IEEE Computer Society, 2017.
URL http://dx.doi.org/10.1109/ICDCS.2017.63
Main reference Amy Babay, Michael Dinitz, Zeyu Zhang: "Characterizing Demand Graphs for (Fixed-Parameter) Shallow-Light Steiner Network", CoRR, Vol. abs/1802.10566, 2018.
URL http://arxiv.org/abs/1802.10566

Emerging applications such as remote manipulation and remote robotic surgery require communication that is both timely and reliable, but the Internet natively supports only communication that is either completely reliable with no timeliness guarantees (e.g. TCP) or timely with best-effort reliability (e.g. UDP). We present an overlay transport service that can provide highly reliable communication while meeting stringent timeliness guarantees over the Internet. To do this we introduce "dissemination graphs", providing a unified framework for specifying routing schemes ranging from a single path, to multiple disjoint paths, to arbitrary graphs. We develop a timely dissemination-graph-based routing method using these graphs that can add targeted redundancy in problematic areas of the network. This routing method is based on algorithms for the Shallow-Light Steiner Network problem, in which we are given a graph $G = (V, E)$, a collection of pairs of vertices (the demands), and a length bound L, and are asked to find the smallest subgraph in which all demands have distance at most L. Motivated by dissemination graphs, we exactly characterize the classes of demands for which the problem is fixed parameter tractable, and prove that for all other classes of demands the problem is $W[1]$-hard.

3.6 Proximity Results and Faster Algorithms for Integer Programming using the Steinitz Lemma

Fritz Eisenbrand (EPFL - Lausanne, CH)

Joint work of Friedrich Eisenbrand, Robert Weismantel
Main reference Friedrich Eisenbrand, Robert Weismantel: "Proximity results and faster algorithms for Integer Programming using the Steinitz Lemma", in Proc. of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018, pp. 808–816, SIAM, 2018.
URL http://dx.doi.org/10.1137/1.9781611975031.52

We consider integer programming problems in standard form $\max\{c^T x : Ax = b,\, x \geq 0,\, x \in Z^n\}$ where $A \in Z^{m\times n}$, $b \in Z^m$ and $c \in Z^n$. We show that such an integer program can be solved in time $(m\Delta)^{O(m)} \cdot \|b\|_\infty^2$, where Δ is an upper bound on each absolute value of an entry in A. This improves upon the longstanding best bound of Papadimitriou (1981) of $(m \cdot \Delta)^{O(m^2)}$, where in addition, the absolute values of the entries of b also need to be bounded by Δ. Our result relies on a lemma of Steinitz that states that a set of vectors in R^m that is contained in the unit ball of a norm and that sum up to zero can be ordered such that all partial sums are of norm bounded by m. We also use the Steinitz lemma to show that the ℓ_1-distance of an optimal integer and fractional solution, also under the presence of upper bounds on the variables, is bounded by $m \cdot (2\, m \cdot \Delta + 1)^m$. Here Δ is again an upper bound on the absolute values of the entries of A. The novel strength of our bound is that it is independent of n. We provide evidence for the significance of our bound by applying it to general knapsack problems where we obtain structural and algorithmic results that improve upon the recent literature.

3.7 Optimization and Scheduling with Explorable Uncertainty

Thomas Erlebach (University of Leicester, GB)

Explorable uncertainty refers to settings where parts of the input data are initially unknown, but can be obtained at a certain cost using queries. In a typical setting, initially only intervals that contain the exact input values are known, and queries can be made to obtain exact values. An algorithm must make queries one by one until it has obtained sufficient information to solve the given problem. We discuss two lines of work in this area: In the area of query-competitive algorithms, one compares the number of queries made by the algorithm with the best possible number of queries for the given input. In the area of scheduling with explorable uncertainty, queries may correspond to tests that can reduce the running-time of a job by an a priori unknown amount and are executed on the machine that also schedules the jobs, thus contributing directly to the objective value of the resulting schedule.

3.8 On Scheduling Consistent Software-Defined Network Updates

Klaus-Tycho Foerster (Universität Wien, AT)

While Software Defined Networks are controlled centrally by one (logical) controller, the dissemination of updates in the network itself is an inherently asynchronous distributed process. Even though eventual consistency (e.g., no forwarding loops) is easy to guarantee, many useful network properties might be violated during the update process. In this talk, we will concentrate on the problem of scheduling such updates in a way that the consistency properties are not violated. In particular, we focus on the consistency properties of loop freedom and congestion freedom, providing a general overview and pointing out open problems.

3.9 How To Plan Ahead

Seth Gilbert (National University of Singapore, SG)

Joint work of M. Bender, M. Farach-Colton, Sándor P. Fekete, Jeremy T. Fineman, Seth Gilbert, Shunhao Oh
Main reference Michael A. Bender, Martin Farach-Colton, Sándor P. Fekete, Jeremy T. Fineman, Seth Gilbert: "Reallocation Problems in Scheduling", Algorithmica, Vol. 73(2), pp. 389–409, 2015.
URL http://dx.doi.org/10.1007/s00453-014-9930-4
Main reference Shunhao Oh, Seth Gilbert: "A Reallocation Algorithm for Online Split Packing of Circles", CoRR, Vol. abs/1802.05873, 2018.
URL http://arxiv.org/abs/1802.05873

Planning ahead has many benefits: it often leads to better results and less last minute stress. That is not what this talk will be about.

Instead, let us think about scheduling. Imagine you are scheduling appointments at a doctor's office. Dr. Spaceman has a busy schedule, with patients all day. And then a VIP case arrives which has to be scheduled exactly at 10am. The result? All the other patients have to be rescheduled, which makes them exceedingly unhappy. Ideally, there would be some way to plan ahead and avoid this disruption.

The same type of rescheduling problem occurs in many different contexts, ranging from delivery schedules to airline routes to assembly lines on a factory floors. In general, we want to maintain a nearly optimal schedule, while allowing jobs to be inserted and deleted. As the set of jobs in the system changes, we will reschedule existing jobs to maintain an efficient schedule. However, rescheduling jobs has a cost, and our goal is to minimize that cost. Specifically, I will talk about three examples: scheduling with arrival times and deadlines, scheduling to minimize the makespan, and scheduling to minimize the sum-of-completion-times.

In general, we will see that by planning ahead, we can minimize the disruption caused by changes to the schedule.

3.10 Approximation Algorithms for Stochastic Scheduling and Routing

Anupam Gupta (Carnegie Mellon University - Pittsburgh, US)

Joint work of Anupam Gupta, Archit Karandikar, Amit Kumar, Viswanath Nagarajan, Xiangkun Shen
Main reference Anupam Gupta, Archit Karandikar: "Stochastic Unsplittable Flows", in Proc. of the Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA, LIPIcs, Vol. 81, pp. 7:1–7:19, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.APPROX-RANDOM.2017.7
Main reference Anupam Gupta, Amit Kumar, Viswanath Nagarajan, Xiangkun Shen: "Stochastic Load Balancing on Unrelated Machines", in Proc. of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018, pp. 1274–1285, SIAM, 2018.
URL http://dx.doi.org/10.1137/1.9781611975031.83

In this talk we will talk about some recent results for scheduling and routing problems where the input parameters are are not deterministic but random variables with known distributions. The goal now is to optimize some expected measure of goodness. We will show some of the ideas needed to develop algorithms in this setting, and to prove their performance guarantees.

3.11 MapReduce Models and Algorithmics

Sungjin Im (University of California - Merced, US)

MapReduce and its follow-up runners such as Spark have become dominant massively parallel computing platforms. These platforms take care of underlying cumbersome distributed system issues under the hood while offering easy interface to the programmers. They are distinguished from traditional parallel computing platforms in the effective way they bridge local computation and communication across machines. Motivated by the tremendous success of the platforms, there have been attempts in the algorithms community to model their unique computing constraints and power, and consequently, many interesting algorithmic ideas have been discovered. This talk will give a quick overview of the theoretical models and key algorithmic ideas/results developed for the massively parallel computing platforms.

3.12 Bypassing Lower Bounds by Stochastic Input Models: The Temp Secretary Problem and Beyond

Thomas Kesselheim (TU Dortmund, DE)

Joint work of Thomas Kesselheim, Andreas Tönnis
Main reference Thomas Kesselheim, Andreas Tönnis: "Think Eternally: Improved Algorithms for the Temp Secretary Problem and Extensions", in Proc. of the 24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark, LIPIcs, Vol. 57, pp. 54:1–54:17, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.ESA.2016.54

Stochastic input models are a promising way to bypass lower bounds of worst-case analysis. Generally, the input is specified by an adversary only to some degree and the remainder is drawn from a probability distribution. Typical examples are random-order analyses of online algorithms and smoothed analysis.

The Temp Secretary Problem (originally proposed by Fiat et al., ESA 2015) is a variant of online deadline scheduling in such a model. An adversary defines jobs by weights and processing times. The release dates are drawn i.i.d. uniformly from $[0, 1]$. At any release date, the algorithm may choose to accept the job and then has to process it immediately without preemption, or it may reject it. Each machine can process only one job at a time. The goal is to maximize the weight of the accepted jobs. If the input is completely adversarial, one cannot achieve any reasonable guarantee. In the partly-stochastic model, it is possible to be constant competitive (see K. and Tönnis, ESA 2016).

In the domain of scheduling, there are certainly many more examples of problems that are interesting to study in such an input model. Besides bypassing lower bounds on the competitive ratio, probably one can also bypass hardness-of-approximation bounds. An interesting candidate would be Flow Time minimization. I am interested in (further) potential applications and feedback for model refinement as well as collaborations to solve the (still to be defined) open problems.

3.13 Coflow Scheduling and Beyond

Samir Khuller (University of Maryland - College Park, US)

Applications designed for data-parallel computation frame-works such as MapReduce usually alternate between computation and communication stages. Coflow scheduling is a popular networking abstraction introduced to capture such application-level communication patterns in datacenters. In this framework, a datacenter is modeled as a single non-blocking switch with m input ports and m output ports. A coflow is a collection of flow demands that is said to be complete once all of its requisite flows have been scheduled. We consider the offline coflow scheduling problem with and without release times to minimize the total weighted completion time. Coflow scheduling generalizes the well studied concurrent open shop scheduling problem and is thus NP-hard.

We give a survey of recent results on Coflow scheduling and also some recent directions. This will be a short survey talk.

3.14 Sublinear communication for Solving Network Problems

Valerie King (University of Victoria, CA)

I describe a simple algorithm by which each node in a network sends one message of $n^{1/2}$ bits to a central controller which can then determine the connected components of the network. The nodes know only the approximate size of the network and there is no shared randomness.

3.15 Constant Factor Approximation Algorithm for Weighted Flow Time on a Single Machine in Pseudo-polynomial time

Amit Kumar (Indian Inst. of Technology - New Dehli, IN)

Joint work of Jatin Batra, Naveen Garg, Amit Kumar
Main reference Jatin Batra, Naveen Garg, Amit Kumar: "Constant Factor Approximation Algorithm for Weighted Flow Time on a Single Machine in Pseudo-polynomial time", CoRR, Vol. abs/1802.07439, 2018.
URL http://arxiv.org/abs/1802.07439

In the weighted flow-time problem on a single machine, we are given a set of n jobs, where each job has a processing requirement, release date and weight. The goal is to find a preemptive schedule which minimizes the sum of weighted flow-time of jobs, where the flow-time of a job is the difference between its completion time and its released date. We give the first pseudo-polynomial time constant approximation algorithm for this problem. The running time of our algorithm is polynomial in n, the number of jobs, and P, which is the ratio of the largest to the smallest processing requirement of a job. Our algorithm relies on a novel reduction of this problem to a generalization of the multi-cut problem on trees, which we call the Demand Multi-Cut problem. Even though we do not give a constant factor approximation algorithm for the Demand Multi-Cut problem on trees, we show that the specific instances of Demand Multi-Cut obtained by reduction from weighted flow-time problem instances have more structure in them, and we are able to employ techniques based on dynamic programming. Our dynamic programming algorithm relies on showing that there are near optimal solutions which have nice smoothness properties, and we exploit these properties to reduce the size of DP table.

3.16 Distributed Shortest Paths, Exactly

Danupon Nanongkai (KTH Royal Institute of Technology, SE)

Joint work of Chien-Chung Huang, Danupon Nanongkai, Thatchaphol Saranurak
Main reference Chien-Chung Huang, Danupon Nanongkai, Thatchaphol Saranurak: "Distributed Exact Weighted All-Pairs Shortest Paths in $\tilde{O}(n^{5/4})$ Rounds", in Proc. of the 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, pp. 168–179, IEEE Computer Society, 2017.
URL http://dx.doi.org/10.1109/FOCS.2017.24

This talk concerns the problem of quickly computing distances and shortest paths on distributed networks (the CONGEST model). There have been many developments for this problem in the last few year, resulting in tight approximation schemes. This left widely open whether exact algorithms can perform equally well. In this talk, we will discuss some recent progress in answering this question.

3.17 Guest lecture (Seminar 18102): Equilibria in the Fluid Queueing Model

Neil Olver (VU University of Amsterdam, NL)

I will discuss the fluid queueing model, introduced by Vickrey in '69. It is probably the simplest model that plausibly captures the notion of time-varying flows. Although the model is quite simple, our current theoretical understanding of equilibrium behaviour in this model is rather limited, and many fundamental questions remain open. I'll survey a few aspects, such as a structural characterization by Koch and Skutella, and quite general existence and uniqueness results by Cominetti, Correa and Larré. In the second part of the talk I'll discuss a recent result (joint work with Roberto Cominetti and Jose Correa) where we resolve one simple-sounding question: do queue lengths remain bounded in the equilibria under natural necessary conditions?

3.18 Scheduling and Optimization Problems in the Wild

Yvonne-Anne Pignolet (ABB Corporate Research - Baden-Dättwil, CH)

With the ongoing trend of increased automation and digitalization, the scheduling and networking algorithms that cope with the growing amount of data produced by industrial plants and processes rise in importance. From factory automation, operating power systems to mining, many automation systems have a scheduling component and due to the many interdependencies graph-theoretic and networking aspects are abundant as well.

In this talk I will present some of the challenges we have worked on at ABB Corporate research in the past. They include real-time communication in substations, stator winding optimizations and workforce scheduling.

3.19 Deterministic Discrepancy Minimization via the Multiplicative Weight Update Method

Thomas Rothvoss (University of Washington - Seattle, US)

Joint work of Avi Levy, Harishchandra Ramadas, Thomas Rothvoss
Main reference Avi Levy, Harishchandra Ramadas, Thomas Rothvoss: "Deterministic Discrepancy Minimization via the Multiplicative Weight Update Method", in Proc. of the Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10328, pp. 380–391, Springer, 2017.
URL http://dx.doi.org/10.1007/978-3-319-59250-3_31

A well-known theorem of Spencer shows that any set system with n sets over n elements admits a coloring of discrepancy $O(\sqrt{n})$. While the original proof was non-constructive, recent progress brought polynomial time algorithms by Bansal, Lovett and Meka, and Rothvoss. All those algorithms are randomized, even though Bansal's algorithm admitted a complicated derandomization.

We propose an elegant deterministic polynomial time algorithm that is inspired by Lovett-Meka as well as the Multiplicative Weight Update method. The algorithm iteratively updates a fractional coloring while controlling the exponential weights that are assigned to the set constraints.

A conjecture by Meka suggests that Spencer's bound can be generalized to symmetric matrices. We prove that $n \times n$ matrices that are block diagonal with block size q admit a coloring of discrepancy $O(\sqrt{n} \cdot \sqrt{\log(q)})$. Bansal, Dadush and Garg recently gave a randomized algorithm to find a vector x with entries in $\{-1, 1\}$ with $\|Ax\|_\infty \leq O(\sqrt{\log n})$ in polynomial time, where A is any matrix whose columns have length at most 1. We show that our method can be used to deterministically obtain such a vector.

3.20 Clustering with an Oracle

Barna Saha (University of Massachusetts, US)

Joint work of Arya Mazumdar, Barna Saha
Main reference Arya Mazumdar, Barna Saha: "Clustering with an oracle", in Proc. of the 54th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2016, Monticello, IL, USA, September 27-30, 2016, pp. 738–739, IEEE, 2016.
URL http://dx.doi.org/10.1109/ALLERTON.2016.7852305

Suppose, we are given $V = \{1, 2, \ldots, n\} \equiv [n]$, a set of n points, that can be clustered into k parts $V_i, i = 1, \ldots, k; V_i \cap V_j = \emptyset, \forall i \neq j$; the subsets $V_i \subset [n]$ and k are unknown to us. There is an oracle that can answer any pair-wise queries $V \times V \to \{\pm 1\}$, where a query answer of $+1$ for $(u, v) \in V \times V$ indicates u and v belong to the same cluster, and -1 indicates they do not. How many such queries are necessary and/or sufficient to find the clusters exactly?

3.21 Interactive Communication with Multiple Parties, or Scheduling with Noise and Feedback

Jared Saia (University of New Mexico, US)

A group of n users want to run a distributed protocol Π over a network where communication occurs via private point-to-point channels. Unfortunately, an adversary, who knows Π, is able to maliciously flip bits on the channels. Can we efficiently simulate Π in the presence of such an adversary?

We show that this is possible, even when L, the number of bits sent in Π, and T, the number of bits flipped by the adversary are not known in advance. In particular, we show how to create a robust version of Π that 1) fails with probability at most ϵ, for any $\epsilon > 0$; and 2) sends soft-$O(L + T)$ bits, where the soft-O notation hides a $\log(n(L + T)/\epsilon)$ term multiplying L.

We believe that algorithms for this problem can be viewed as "scheduling with noise" in the following sense. In each time step it is possible for a "good" node to send a bit of communication over a channel, and it is also possible for the adversary to jam that channel. Given the worst case actions of the adversary, we would like to schedule all necessary communication as efficiently as possible. A major open problem is to obtain results with a "makespan" of $O(L+T)$ instead of soft-$O(L+T)$.

3.22 Queueing in the Mist: Buffering and Scheduling with Limited Knowledge

Gabriel Scalosub (Ben Gurion University - Beer Sheva, IL)

Joint work of Itamar Cohen, Gabriel Scalosub
Main reference Itamar Cohen, Gabriel Scalosub: "Queueing in the mist: Buffering and scheduling with limited knowledge", in Proc. of the 25th IEEE/ACM International Symposium on Quality of Service, IWQoS 2017, Vilanova i la Geltrú, Spain, June 14-16, 2017, pp. 1–6, IEEE, 2017.
URL http://dx.doi.org/10.1109/IWQoS.2017.7969126

Managing queues and scheduling with bounded buffers are among the most fundamental problems in computer networking. Although traditionally it is often assumed that all the properties of each packet are known immediately upon arrival, various real life scenarios render such assumptions invalid. We study some buffering and scheduling problems where traffic characteristics only become known after some preliminary processing. As opposed to having stochastic assumptions on the underlying traffic, we use an adversarial model, and present algorithms and lower bounds for such settings.

3.23 Routing and Scheduling in Hybrid Networks

Christian Scheideler (Universität Paderborn, DE)

Hybrid networks are networks in which the nodes have different communication modes. Cell phones, for example, can communicate via the cellular infrastructure or via their Wifi interfaces. Communication via Wifi interfaces has the advantage that there is no limit (other than the bandwidth and battery constraints) on the amount of data that can be exchanged while the amount of data that can be transferred at a reasonable rate via long-range links using the cellular infrastructure or satellite is limited (by some data plan) or costly. However, routing and scheduling in Wifi networks is challenging, particularly because the topology of the network might be messy and not under the control of the nodes, while the nodes would be able to set up arbitrary overlay networks on top of the cellular infrastructure, but this comes at a price. In abstract terms, we are given a network of cheap links of arbitrary topology that is not under the control of the nodes and potentially changing, but in addition to that the nodes have the ability to build arbitrary overlay networks of costly links that are under the control of the nodes. How can these overlays be used effectively in order to solve routing and scheduling problems for the cheap network? I will give an overview of recent results on that problem.

3.24 Recent Advances for Online Machine Minimization

Kevin Schewior (University of Chile, CL)

Joint work of Lin Chen, Nicole Megow, Kevin Schewior
Main reference Lin Chen, Nicole Megow, Kevin Schewior: "The Power of Migration in Online Machine Minimization", in Proc. of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2016, Asilomar State Beach/Pacific Grove, CA, USA, July 11-13, 2016, pp. 175–184, ACM, 2016.
URL http://dx.doi.org/10.1145/2935764.2935786
Main reference Lin Chen, Nicole Megow, and Kevin Schewior: "An O(log m)-competitive algorithm for online machine minimization". In Proc. of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms (SODA '16), Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 155-163, 2016.
URL https://dl.acm.org/citation.cfm?id=2884447

Online Machine Minimization (Phillips et al., STOC 1997) is a fundamental online scheduling problem: Jobs arrive over time at their release dates and need to be scheduled preemptively on parallel machines until their deadlines. Here, the performance of a schedule is measured in the number of machines it requires. It has been an open question since the introduction of the problem whether constant-competitive algorithms exist, until the results presented in this talk even if the number m of machines that the offline optimum uses is fixed. We review the recent progress that has been made on this problem. We introduce the new lower-bound technique by Chen, Megow, and Schewior (SODA 2016) and the algorithms that arose from it: We present the $O(\log m)$-competitive algorithm from the same paper, the modification that yields an $O(\log m/\log\log m)$-competitive algorithm (Azar and Cohen, OR Letters 2018), and the additional building block used to obtain an $O(\log\log m)$-competitive algorithm (Im et al., RTSS 2017). We complement these results with improved upper bounds for special cases and an impossibility result for non-migratory algorithms (Chen, Megow, and Schewior, SPAA 2016).

3.25 On Packet Scheduling with Adversarial Jamming and Speedup

Jirí Sgall (Charles University - Prague, CZ)

Joint work of Martin Böhm, Lukasz Jez, Jirí Sgall, Pavel Veselý
Main reference Martin Böhm, Lukasz Jez, Jirí Sgall, Pavel Veselý: "On Packet Scheduling with Adversarial Jamming and Speedup", in Proc. of the Approximation and Online Algorithms - 15th International Workshop, WAOA 2017, Vienna, Austria, September 7-8, 2017, Revised Selected Papers, Lecture Notes in Computer Science, Vol. 10787, pp. 190–206, Springer, 2017.
URL http://dx.doi.org/10.1007/978-3-319-89441-6_15

In Packet Scheduling with Adversarial Jamming packets of arbitrary sizes arrive over time to be transmitted over a channel in which instantaneous jamming errors occur at times chosen by the adversary and not known to the algorithm. The transmission taking place at the time of jamming is corrupt, and the algorithm learns this fact immediately. An online algorithm maximizes the total size of packets it successfully transmits and the goal is to develop an algorithm with the lowest possible asymptotic competitive ratio, where the additive constant may depend on packet sizes.

Our main contribution is a universal algorithm that works for any speedup and packet sizes and, unlike previous algorithms for the problem, it does not need to know these properties in advance. We show that this algorithm guarantees 1-competitiveness with speedup 4, making it the first known algorithm to maintain 1-competitiveness with a moderate speedup in the general setting of arbitrary packet sizes. We also prove a lower bound of $\phi + 1 \approx 2.618$ on the speedup of any 1-competitive deterministic algorithm, showing that our algorithm is close to the optimum.

Additionally, we formulate a general framework for analyzing our algorithm locally and use it to show upper bounds on its competitive ratio for speedups in $[1, 4)$ and for several special cases, recovering some previously known results, each of which had a dedicated proof. In particular, our algorithm is 3-competitive without speedup, matching the algorithm and the lower bound of Jurdzinski et al.

3.26 Generalizing the Kawaguchi-Kyan Bound to Stochastic Parallel Machine Scheduling

Martin Skutella (TU Berlin, DE)

Joint work of Sven Jäger, Martin Skutella
Main reference Sven Jäger, Martin Skutella: "Generalizing the Kawaguchi-Kyan Bound to Stochastic Parallel Machine Scheduling", in Proc. of the 35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France, LIPIcs, Vol. 96, pp. 43:1–43:14, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
URL http://dx.doi.org/10.4230/LIPIcs.STACS.2018.43

Minimizing the sum of weighted completion times on m identical parallel machines is one of the most important and classical scheduling problems. For the stochastic variant where processing times of jobs are random variables, Möhring, Schulz, and Uetz (1999) presented the first and still best known approximation result achieving, for arbitrarily many machines, performance ratio $1 + \frac{1}{2}(1 + \Delta)$, where Δ is an upper bound on the squared coefficient of variation of the processing times. We prove performance ratio $1 + \frac{1}{2}(\sqrt{2} - 1)(1 + \Delta)$ for the same underlying algorithm – the Weighted Shortest Expected Processing Time (WSEPT) rule. For the special case of deterministic scheduling (i.e., $\Delta = 0$), our bound matches the tight performance ratio $\frac{1}{2}(1 + \sqrt{2})$ of this algorithm (WSPT rule), derived by Kawaguchi and Kyan in a 1986 landmark paper.

3.27 Partitioning into Quadruples

Frits C. R. Spieksma (TU Eindhoven, NL)

Joint work of Thomas Erlebach, Annette Ficker, Matúš Mihalak, Frits C. R. Spieksma

We consider a clustering problem where $4k$ given vectors need to be partitioned into k clusters of four vectors each. A cluster of four vectors is called a quad, and the cost of a quad is the sum of the component-wise maxima of the four vectors in the quad. The problem is to partition the given $4k$ vectors into k quads with minimum total cost. We analyze the worst-case behavior of a straightforward matching-based algorithm, and prove that this algorithm is a 3/2-approximation algorithm for this clustering problem. We also analyze the performance of this algorithm on special cases of the problem.

3.28 Locality and Distributed Scheduling

Jukka Suomela (Aalto University, FI)

In this talk I will look at scheduling problems from the perspective of distributed computing. It turns out that there are two commonly used interpretations of the term distributed computing, and at first they seem to take us in the opposite directions in comparison with the classical centralised setting:

(1) "Big data perspective." The key resource is computation, and distributed computing helps us in comparison with centralised algorithms. We have access to the computing power of multiple computers, and we can use them to speed up computation and to solve large instances that do not fit in the memory of one computer. Relevant models of computing include the MapReduce model and the bulk synchronous parallel model (BSP), and there is a close connection to parallel computing (e.g. the PRAM model).

(2) "Network algorithms perspective." The key resource is communication, and the distributed setting is an additional challenge in comparison with centralised algorithms. We have a large number of nodes, each of them is initially aware of its own part of the input, and each node needs to compute its own part of the solution (e.g. when to switch on), ideally without any global coordination. Relevant models of computing include the LOCAL model and the CONGEST model, and there is a close connection to sublinear-time algorithms (e.g. property testing) and communication complexity.

In this talk I will mainly focus on the new challenges that we have when we look at scheduling problems from perspective (2). Here the overarching theme is locality: how far does a node need to see in order to find its own part of the solution? We will discuss ways of defining scheduling problems in this setting, and how they are connected to the main challenges of the field (e.g. symmetry breaking) and current research themes (e.g. classification of so-called locally checkable problems).

However, I will also look at ways of bridging the gap between perspectives (1) and (2); I will discuss unifying models such as the congested clique model, which is closely related to both the BSP model and the CONGEST model.

3.29 A Constant-factor Approximation Algorithm for the Asymmetric Traveling Salesman Problem

Ola Svensson (EPFL - Lausanne, CH)

Joint work of Ola Svensson, Jakub Tarnawski, László A. Végh
Main reference Ola Svensson, Jakub Tarnawski, László A. Végh: "A Constant-Factor Approximation Algorithm for the Asymmetric Traveling Salesman Problem", CoRR, Vol. abs/1708.04215, 2017.
URL http://arxiv.org/abs/1708.04215

We give a constant-factor approximation algorithm for the asymmetric traveling salesman problem. Our approximation guarantee is analyzed with respect to the standard LP relaxation, and thus our result confirms the conjectured constant integrality gap of that relaxation.

Our techniques build upon the constant-factor approximation algorithm for the special case of node-weighted metrics. Specifically, we give a generic reduction to structured instances that resemble but are more general than those arising from node-weighted metrics. For those instances, we then solve Local-Connectivity ATSP, a problem known to be equivalent (in terms of constant-factor approximation) to the asymmetric traveling salesman problem.

3.30 Greed is Good (for Scheduling under Uncertainty)

Marc Uetz (University of Twente, NL)

Joint work of Varun Gupta, Benjamin Moseley, Marc Uetz, Qiaomin Xie
Main reference Varun Gupta, Benjamin Moseley, Marc Uetz, Qiaomin Xie: "Stochastic Online Scheduling on Unrelated Machines", in Proc. of the Integer Programming and Combinatorial Optimization - 19th International Conference, IPCO 2017, Waterloo, ON, Canada, June 26-28, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10328, pp. 228–240, Springer, 2017.
URL http://dx.doi.org/10.1007/978-3-319-59250-3_19

The talk addresses a classical problem in the area of scheduling, namely minimizing the total weighted completion time of non-preemptive jobs on a set of unrelated machines. Uncertainty enters the model by assuming that job processing times are stochastic. In order to obtain constant factor approximation algorithms for that problem, prior work required sophisticated linear or convex programming relaxations for the assignment of jobs to machines. In contrast, we analyze a purely combinatorial, online algorithm. Maybe surprisingly, we show how to derive performance bounds for that algorithm that are of the same order of magnitude as those of earlier work, while our results are the first for an online setting. The analysis is based on dual fitting techniques.

4 Open problems

4.1 Algorithmic Problems Combining Network Design, Routing, and Scheduling

Michael Dinitz (Johns Hopkins University - Baltimore, US)

Some next-generation networks will have the ability to dynamically reconfigure their topology. Specific proposals for such networks have recently appeared in the datacenter architecture literature, and this is already happening in optical wide-area networks. Particularly in the WAN setting, scheduling large flows becomes an important problem. But in a graph where we control the topology dynamically, we get problems which combine three different types of algorithmic questions: Network design, routing, and scheduling. There is already systems work in this area, but the theoretical work typically ignores at least one of these aspects (usually routing or scheduling). Combining all three should lead to some interesting questions.

4.2 Distributed Computation on Speed Scalable Processors

Kirk Pruhs (University of Pittsburgh, US)

Consider a problem P that one wants to solve in some distributed model of computation. For example, computing the minimum spanning tree in the congest model. Assume that there is an algorithm A for this problem that runs in time T. If one assumes that each node in the n node network uses a unit of energy per unit time, then this algorithm would use nT units of energy. Now assume that the nodes are speed scalable, that is they can run at any speed s. Running at speed s means that a node can send s rounds of messages in unit time. But running as speed s costs s^2 unit of energy. Then algorithm A can be converted into an algorithm that runs in T/s time using energy nTs^2. If A is optimal then this is an optimal time/energy trade-off if all the processors run at the same speed all the time. So the question is, for your favorite problem P, like minimum spanning, can one ever beat this time/energy trade-off by allowing processors to run at different speeds at different times. Intuitively this is asking whether solving P inherently involves geographically and temporally local bottlenecks, that can be ameliorated with speed scaling, whether bottlenecks can be evenly spread out temporally and geographically.

4.3 Online Buffers on the Line

Rob van Stee (Universität Siegen, DE)

Main reference Matthias Englert: "The reordering buffer problem on the line revisited", SIGACT News, Vol. 49(1), pp. 67–72, 2018.
URL http://dx.doi.org/10.1145/3197406.3197418
Main reference Iftah Gamzu, Danny Segev: "Improved online algorithms for the sorting buffer problem on line metrics", ACM Trans. Algorithms, Vol. 6(1), pp. 15:1–15:14, 2009.
URL http://dx.doi.org/10.1145/1644015.1644030

A single server needs to serve requests (points) appearing on the line by visiting them. It can store k requests in a buffer, and the goal is to minimize the total distance traveled by the server. There is a relatively simple online algorithm which is also the best algorithm known, online or offline (!). It works by partitioning the line geometrically and is $O(\log n)$-competitive. Improve it. The best known lower bound is less than 3.

Participants

- Susanne Albers
TU München, DE
- Yossi Azar
Tel Aviv University, IL
- Nikhil Bansal
TU Eindhoven, NL
- Sanjoy K. Baruah
University of North Carolina at Chapel Hill, US
- Syamantak Das
IIIT – New Dehli, IN
- Jelena Diakonikolas
Boston University, US
- Michael Dinitz
Johns Hopkins University – Baltimore, US
- Fritz Eisenbrand
EPFL – Lausanne, CH
- Thomas Erlebach
University of Leicester, GB
- Klaus-Tycho Foerster
Universität Wien, AT
- Seth Gilbert
National University of Singapore, SG
- Anupam Gupta
Carnegie Mellon University, US
- Magnús M. Halldórsson
Reykjavik University, IS
- Sungjin Im
University of California – Merced, US
- Thomas Kesselheim
TU Dortmund, DE
- Samir Khuller
University of Maryland – College Park, US
- Valerie King
University of Victoria, CA
- Fabian Daniel Kuhn
Universität Freiburg, DE
- Amit Kumar
Indian Inst. of Technology – New Dehli, IN
- Alberto Marchetti-Spaccamela
Sapienza University of Rome, IT
- Nicole Megow
Universität Bremen, DE
- Danupon Nanongkai
KTH Royal Institute of Technology, SE
- Yvonne-Anne Pignolet
ABB Corporate Research – Baden-Dättwil, CH
- Kirk Pruhs
University of Pittsburgh, US
- Thomas Rothvoss
University of Washington – Seattle, US
- Barna Saha
University of Massachusetts, US
- Jared Saia
University of New Mexico, US
- Gabriel Scalosub
Ben Gurion University – Beer Sheva, IL
- Christian Scheideler
Universität Paderborn, DE
- Kevin Schewior
University of Chile - Santiago, CL
- Jiri Sgall
Charles University – Prague, CZ
- David Shmoys
Cornell University, US
- René Sitters
VU University of Amsterdam, NL
- Martin Skutella
TU Berlin, DE
- Frits C. R. Spieksma
TU Eindhoven, NL
- Clifford Stein
Columbia University, US
- Leen Stougie
VU University of Amsterdam, NL
- Jukka Suomela
Aalto University, FI
- Ola Svensson
EPFL – Lausanne, CH
- Marc Uetz
University of Twente, NL
- Rob van Stee
University of Leicester, GB
- Anke van Zuylen
College of William and Mary – Williamsburg, US
- Jose Verschae
Pontifical Catholic University of Chile - Santiago, CL
- Tjark Vredeveld
Maastricht University, NL