Report from Dagstuhl Seminar 11091

Packing and Scheduling Algorithms for Information and Communication Services

Edited by
Klaus Jansen[1], Claire Mathieu[2], Hadas Shachnai[3], and Neal E. Young[4]

1 **Universität Kiel, DE,** kj@informatik.uni-kiel.de
2 **Brown University - Providence, US,** claire@cs.brown.edu
3 **Technion - Haifa, IL,** hadas@cs.technion.ac.il
4 **Univ. California - Riverside, US,** neal@cs.ucr.edu

Abstract

From 27.02.2011 to 4.03.2011, the Dagstuhl Seminar 11091 "Packing and Scheduling Algorithms for Information and Communication Services" was held in Schloss Dagstuhl Leibniz Center for Informatics. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Seminar 27.02–4.03.2011 – www.dagstuhl.de/11091
1998 ACM Subject Classification F.2.2 Sequencing and scheduling, G.4 Algorithm Design and Analysis, G.2 Discrete Mathematics, C.2 Computer-Communication Networks
Keywords and phrases Packing, scheduling, information and communication services, combinatorial optimization, mathematical programming, parameterized complexity
Digital Object Identifier 10.4230/DagRep.1.2.67

1 Executive Summary

Klaus Jansen
Claire Mathieu
Hadas Shachnai
Neal E. Young

Packing and scheduling are one area where mathematics meets puzzles. While many of these problems stem from real-life applications, they have also been of fundamental theoretical importance. In a *packing* problem given is a set of items and one or more (multi-dimensional) bins. The objective is to maximize the profit from packing a subset of the items, or to minimize the cost of packing *all* items. In a *scheduling* problem, given are a set of jobs and a set of machines. One needs to schedule the jobs to run on the machines (under some constraints) so as to optimize an objective function that depends on the order of the jobs, on their completion times or on the machines by which they are processed.

Storage allocation in computer networks, cutting stock problems in various industries and production planning are only few of the applications of packing and scheduling. With the growing impact of next generation technologies in information and communication services (some examples are Video-on-Demand systems, web applications and wireless networks), practitioners as well as theoreticians seek fast and efficient solutions for new variants of some classic packing and scheduling problems, which are crucial for optimizing the performance of these systems.

Since many of these problems are NP-hard, it is natural to seek efficient approximate solutions. Traditionally, such approximations are obtained by using fundamental tools from combinatorial optimization and mathematical programming. While for some of the problems there exist algorithms which achieve the best possible approximation ratio, one major effort of this community has been to close the gaps in running times between heuristic solutions, which perform well in practice, and algorithms which are provably efficient in terms of approximation ratio, but impractical in use. The large class of approximation schemes for packing and scheduling problems has been the recent target of this effort.

Parameterized complexity uses refined measures for the approximability of a given problem, by referring, e.g., to approximation with instance parameters, by defining performance functions (instead of performance ratios) and by defining the quality of approximation as parameter. Such measures provide further insight to the studied problems and lead to the design of algorithms that work efficiently if the parameters of the input instance are small (even if the size of the input is large). Efficient parameterization for packing and scheduling problems is a major challenge on the way to obtaining practical algorithms.

During the 5 days of the seminar, 24 talks were given by the participants. Five of these talks were two-hour tutorials and 60-minute survey talks on various topics:Kirk Pruhs gave an exciting tutorial on the challenges faced by designers of algorithms for green computing; Dániel Marx talked about several existing connections between approximation algorithms and fixed-parameter algorithms; Ola Svensson gave an overview of the implications and techniques of two fascinating hardness of approximation results for shops and precedence constraints scheduling; Neal Young talked about using lagrangian-relaxation algorithms to solve packing and covering problems, and Magnús Halldórsson gave an overview of recent analytic work on scheduling wireless links.

The seminar successfully brought together both experts and newcomers from the areas of packing and sequencing, combinatorial optimization, mathematical programming, and parameterized complexity, with many interesting interactions. The talks left plenty of time for discussion in the afternoon. An open problem session was held on Tuesday, and problems raised there were discussed by different groups throughout the seminar and in a research groups session on Friday. A session on current and future trends in scheduling was held on Thursday, and brought up some exciting issues relating to this area.

2 Table of Contents

3 Overview of Talks

3.1 On Packing Resizable Items and Covering by Holes

Sivan Albagli-Kim (Technion, IL)

Joint work of Albagli-Kim, Sivan; Shachnai, Hadas; Tamir, Tami

In next generation video services, movie files can be transmitted to the clients at different encodings, thus enabling the system to degrade Quality- of-Service for bounded time intervals, while increasing the number of serviced clients. Maximizing throughput in such systems can be modeled as the following problem of packing a set of items, whose sizes may change over time. Given is a set I of unit-sized items and a bin of capacity $B > 1$. The items need to be packed in the bin for a fixed time interval. Each item j can be compressed to the size $p_j \in (0, 1)$ for at most a fraction $q_j \in (0, 1]$ of its packing time. The goal is to pack in the bin, for the given time interval, a subset of the items of maximum cardinality. This problem of *packing resizable items (PRI)* is strongly NP-hard already for highly restricted instances.

In this paper we present approximation algorithms for two subclasses of instances of PRI which are of practical interest. For instances with uniform compression ratio, we develop an asymptotic fully polynomial time approximation scheme. For instances with uniform compression time, we give an almost optimal algorithm, which packs at least $OPT(I) - 1$ items, where $OPT(I)$ is the number of items packed by an optimal algorithm. We derive our results by using a non-standard transformation of PRI to the problem of covering a region by sliceable rectangles. The resulting problem, which finds numerous applications in computational geometry, is of independent interest.

3.2 Secretary Problems via Linear Programming

Niv Buchbinder (Open Univ., IL)

Joint work of Buchbinder, Niv; Jain, Kamal; Singh, Mohit

In the classical secretary problem an employer would like to choose the best candidate among n competing candidates that arrive in a random order. This basic concept of n elements arriving in a random order and irrevocable decisions made by an algorithm have been explored extensively over the years, and used for modeling the behavior of many processes. Our main contribution is a new linear programming technique that we introduce as a tool for obtaining and analyzing mechanisms for the secretary problem and its variants. Capturing the set of mechanisms as a linear polytope holds the following immediate advantages.

1. Computing the optimal mechanism reduces to solving a linear program. 2. Proving an upper bound on the performance of any mechanism reduces to finding a feasible solution to the dual program. 3. Exploring variants of the problem is as simple as adding new constraints, or manipulating the objective function of the linear program.

We demonstrate the applicability of these ideas in several settings including online auctions.

3.3 Approximating the Non-Contiguous Multiple Organization Packing Problem

Pierre-Francois Dutot (INRIA Rhôn-Alpes, FR)

We present in this paper a 5/2-approximation algorithm for scheduling rigid jobs on multi-organizations. For a given set of n jobs, the goal is to construct a schedule for N organizations (composed each of m identical processors) minimizing the maximum completion time (makespan). This algorithm runs in $O(n(N + \log(n)) \log(np_{\max}))$, where p max is the maximum processing time of the jobs. It improves the best existing low cost approximation algorithms. Moreover, the proposed analysis can be extended to a more generic approach which suggests different job partitions that could lead to low cost approximation algorithms of ratio better than 5/2.

3.4 Online Clustering with Variable Sized Clusters

Leah Epstein (Univ. of Haifa, IL)

Joint work of Csirik, János; Epstein, Leah; Imreh, Csanád; Levin, Asaf
Main reference J. Csirik, L. Epstein, C. Imreh, A. Levin, "Online Clustering with Variable Sized Clusters ", MFCS'10, pp.282–293, LNCS.
URL http://dx.doi.org/10.1007/978-3-642-15155-2_26

Online clustering problems are problems where the classification of points into sets (called clusters) is done in an online fashion. Points arrive at arbitrary locations, one by one, to be assigned to clusters at the time of arrival. A point can be either assigned to an existing cluster or a new cluster can be opened for it. We study a one dimensional variant on a line. Each cluster is a closed interval, and there is no restriction on the length of a cluster. The cost of a cluster is the sum of a fixed set-up cost and its diameter (or length). The goal is to minimize the sum of costs of the clusters used by the algorithm. We study several variants, all maintaining the essential property that a point which was assigned to a given cluster must remain assigned to this cluster, and clusters cannot be merged. In the strict variant, the diameter and the exact location of the cluster must be fixed when it is initialized. In the flexible variant, the algorithm can shift the cluster or expand it, as long as it contains all points assigned to it. In an intermediate model, the diameter is fixed in advance while the exact location can be modified. We give tight bounds on the competitive ratio of any online algorithm in each of these variants. In addition, for each one of the models, we also consider the semi-online case, where points are presented sorted by their location. The paper is joint work with J. Csirik, Cs. Imreh and A. Levin, and was presented in MFCS2010.

3.5 Competitive Strategies for Routing Flow Over Time

Lisa K. Fleischer (Dartmouth College, US)

Joint work of Bhaskar, Umang; Fleischer, Lisa K.; Anshelevich, Elliot

The study of routing games is motivated by the desire to understand the impact of individual user's decisions on network efficiency. To do this, prior work uses a simplified model of network flow where all flow exists simultaneously, and users route flow to either minimize their maximum delay or their total delay. Both of these measures are surrogates for measuring how long it takes to get all of your traffic through the network over time.

Instead of using these surrogates, we attempt a more direct study of how competition among users effects network efficiency by examining routing games in a flow-over-time model. We show that the network owner can reduce available capacity so that the competitive equilibrium in the reduced network is no worse than a small constant times the optimal solution in the original network using two natural measures of optimum: the time by which all flow reaches the destination, and the average amount of time it takes flow to reach the destination.

3.6 Potential Reduction Schemes in Structured Optimization

Michael D. Grigoriadis (Rutgers Univ., US)

Joint work of Grigoriadis,Michael D.; Khachiyan, Leonid G.; Villavicencio, J. U.

We study the performance of approximately computing a min-max [max- min] solution of a given set of M convex [concave], nonnegative-valued and block-separable coupling inequalities over the product of K convex compact "blocks". The generality of the model allows for a variety of specializations for applications in packing [covering] feasibility LPs, matrix games, block angular LPs, routing in multicommodity flows, and others. Optimization variants run within polylogarithmic factors. All of our FPTAS's include a quadratic term of 1/epsilon. Working within the well-known Lagrangian decomposition framework, we replace the underlying piecewise convex [concave] objective with its exponentially many breakpoints, by a smooth approximation, such as an exponential or logarithmic potential function, which is gradually improved. For implementations using the exponential potential, the original blocks are further restricted by their part of the coupling inequalities, adjustable by a restriction parameter. This helps in controlling the so-called "width". In contrast, logarithmic potential-based implementations are shown to be "width-free" and thus work with the unrestricted (original) blocks. We show that best coordination complexities obtain by using the logarithmic potential with unrestricted blocks for instances with roughly $M < K \log K$, but switching to the exponential potential with restricted blocks when $M > K \log K$. The exponential potential-based scheme solves (n,m)-matrix games A with elements in [-1,+1] to a prescribed relative error in quadratic $log(nm)$ time on an nm-processor EREW PRAM. In addition there is a parallel randomized approximation scheme for solving such games to within a given absolute accuracy, in expected quadratic $log(n+m)$ time on an $(n+m)/log(n+m)$-processor EREW PRAM, thus providing a sublinear support for such games. A roughly quadratic expected speedup is obtained relative to any deterministic approximation scheme.

Computational experiments show that optimal solutions of very large maximum concurrent flow problems are computed routinely to 4-digit accuracy, several orders faster than modern LP codes.

3.7 Wireless Scheduling in the Physical Model

Magnús M. Halldórsson (Reykjavik Univ., IS)

I will survey recent work on analytic work on scheduling wireless links in the SINR model. The first half will be focused on properties of the model, and on capacity (throughput) maximization in the case of uniform power. The second half will look at the problems involving power control, as well as other related issues, such as distributed algorithms.

3.8 A Polynomial Time OPT+1 Algorithm for the Cutting Stock Problem with a Constant Number of Object Lengths

Klaus Jansen (Universität Kiel, DE)

Joint work of Jansen, Klaus; Solis-Oba, Roberto

In the cutting stock problem we are given a set $T = T_1, ..., T_d$ of object types, where objects of type T_i have integer length $p_i > 0$. Given a set O of n objects containing n_i objects of type T_i, for each $i = 1, ..., d$, the problem is to pack O into the minimum number of bins of capacity beta. In this talk we consider the version of the problem in which the number d of different object types is constant and we present a polynomial time algorithm that computes a solution using at most $OPT + 1$ bins, where OPT is the value of an optimum solution.

3.9 The Cutting-Stock Approach to Bin Packing: Theory and Experiments

David S. Johnson(AT&T Research, US)

We report on results of an experimental study of the Gilmore-Gomory cutting-stock heuristic and related LP-based approaches to bin packing, as applied to instances generated according to discrete distributions. We examine the questions of how best to solve the knapsack problems used to generate columns in the Gilmore-Gomory approach, how the various algorithms' running times and solution qualities scale with key instance parameters, and how the algorithms compare to more traditional bin packing heuristics.

No polynomial running time bound is known to hold for the Gilmore-Gomory approach, and high-level empirical operation counts suggest that no straightforward implementation can have average running time $O(m^3)$, where m is the number of distinct item sizes. Our experiments suggest that by using dynamic programming to solve the knapsack problems, one can robustly obtain average running times that are $o(m^4)$ and are feasible for m well in excess of 1,000. This makes a variant on the previously un-implemented asymptotic approximation scheme of Fernandez de la Vega and Lueker practical for arbitrarily large values of m and quite small values of ϵ.

In the process of performing these experiments we discovered two interesting anomalies: (1) running time *decreasing* as the number n of items increases and (2) solution quality improving as running time is reduced and an approximation guarantee is weakened. We provide explanations for these phenomena and characterize the situations in which they occur.

3.10 Disjoint-Path Facility Location: Theory and Practice

Howard Karloff (AT&T Research, US)

Internet service providers hope to provide their customers with superior Internet connectivity, but do they always do so? How can an ISP even know what quality of service it's providing to its customers? To this end, researchers recently proposed a new scheme an ISP could use in order to estimate the packet loss rates experienced by its customers.

To implement the new scheme, one has to approximately solve an interesting NP-Hard optimization problem on the ISP's network. Specifically, one must choose a small set of network nodes such that from each customer node there are arc-disjoint paths to *two* of the selected nodes. I will discuss recent work, mostly at ATT, attacking this problem and its surprisingly good results, in light of the problem's provable inapproximability in the worst case.

3.11 Procrastination Pays: Scheduling Jobs in Batches to Minimize Energy Usage

Samir Khuller (Univ. of Maryland, US)

Joint work of Chang, J.; Gabow, H.; Khuller, S.

We consider an elementary scheduling problem defined as follows. Given a collection of n jobs, where each job J_i has an integer length l_i as well as a set T_i of time intervals in which it can be feasibly scheduled. We are given a parallelism parameter P and can schedule up to P jobs at any time slot in which the machine is "active". The goal is to preemptively schedule all the jobs in the fewest number of active time slots.

The machine consumes a fixed amount of energy per time slot, regardless of the number of jobs scheduled at that slot (as long as the number of jobs is non-zero). In other words, subject to l_i units of each job i being scheduled in its feasible region and at each slot at most P jobs being scheduled, we are interested in minimizing the total time duration when the machine is active.

We present an O(n log n) algorithm for the case where jobs have unit length and T_i forms a single interval. For general T_i (and unit jobs), we show that the problem is NP-complete even for $P = 3$. However when $P = 2$, we show that it can be solved in polynomial time – we also present several extensions: for example when the jobs have non–unit requirements we can still solve this version in polynomial time.

No previous background knowledge on scheduling is expected. In addition, we will survey some recent work on bundling jobs in batches.

3.12 An AFPTAS for Variable Sized bin Packing with General bin Costs

Asaf Levin (Technion, IL)

Joint work of Epstein, Leah; Levin, Asaf

In variable sized bin packing problems, bins of different sizes are to be used for the packing of an input set of items. We consider variable sized bin packing with general costs. Each bin type has a cost associated with it, where the cost of a bin may be smaller or larger than its size, and the costs of different bin sizes are unrelated. For each bin type, this cost is to be paid for each instance which is used for the packing of input items. This generalized setting of the problem has numerous applications in storage and scheduling. We introduce new reduction methods and separation techniques, which allow us to design an AFPTAS for the problem.

3.13 Survey of connections between approximation algorithms and parameterized complexity

Dániel Marx (HU Berlin, DE)

Approximation algorithms and parameterized complexity are two well-studied approaches for attacking hard combinatorial problems. In my talk, I overview the ways approximation can be introduced into the framework of parameterized complexity, survey results in this direction, and show how parameterized hardness theory can be used to give lower bounds on the efficiency of approximation schemes.

3.14 Vertex Cover in Graphs with Locally Few Colors and Precedence Constrained Scheduling with Few Predecessors

Monaldo Mastrolilli (IDSIA - Lugano, CH)

In 1986 Erdös et. al. defined the local chromatic number of a graph as the minimum number of colors that must appear within distance 1 of a vertex. For any fixed $\Delta \geq 2$, they presented graphs with arbitrarily large chromatic number that can be colored so that: (i) no vertex neighborhood contains more than Δ different colors (bounded local colorability), and (*ii*) adjacent vertices from two color classes form an induced subgraph that is complete and bipartite (local completeness).

We investigate the weighted vertex cover problem in graphs when a locally bounded coloring is given as input. This generalizes in a very natural vein the vertex cover problem in bounded degree graphs to a class of graphs with arbitrarily large chromatic number. Assuming the Unique Game Conjecture, we provide a tight characterization. More precisely, we prove that it is UG-hard to improve the approximation ratio of $2 - 2/(\Delta + 1)$ if only the bounded local colorability, but not the local completeness condition holds for the given coloring. A matching upper bound is also provided. Vice versa, when both the above two properties (i) and (ii) hold, we present a randomized approximation algorithm with performance ratio of $2 - \Omega(1)\frac{\ln \ln \Delta}{\ln \Delta}$. This matches (up to the constant factor in the lower order term) known inapproximability results for the special case of bounded degree graphs.

Moreover, we show that when both the above two properties (i) and (ii) hold, the obtained result finds a natural application in a classical scheduling problem, namely the precedence constrained single machine scheduling problem to minimize the weighted sum of completion times. In a series of recent papers it was established that this scheduling problem is a special case of the minimum weighted vertex cover in graphs G of incomparable pairs defined in the dimension theory of partial orders. We show that G satisfies properties (i) and (ii) where $\Delta - 1$ is the maximum number of predecessors (or successors) of each job.

3.15 Min-Max Graph Partitioning and Small Set Expansion

Seffi Naor (Technion, IL)

We study graph partitioning problems from a min-max perspective, in which an input graph on n vertices should be partitioned into k parts, and the objective is to minimize the maximum number of edges leaving a single part. The two main versions we consider are where the k parts need to be of equal-size, and where they must separate a set of k given terminals. We consider a common generalization of these two problems, and design for it an approximation algorithm. This improves over an O(log2 n) approximation for the second version due to Svitkina and Tardos [ST04], and roughly O(k log n) approximation for the first version that follows from other previous work.

3.16 Green Computing Algorithmics

Kirk Pruhs (Univ. of Pittsburgh, US)

We are in the midst of a green computing revolution involving the redesign of information technology hardware and software at all levels of the information technology stack. Such a revolution spawns a multitude of technological challenges, many of which are algorithmic in nature. The most obvious type of algorithmic problem arising from this green computing revolution involves directly managing power, energy or temperature as a resource. Other algorithmic problems arise because the new technology, which was adopted for energy and power considerations, has different physical properties than previous technologies. I will try to give a feel for the current state of green computing algorithmics research, and provide some advice about how to contribute to this research.

3.17 Minimizing Busy Time in Multiple Machine Real-time Scheduling

Baruch Schieber (IBM TJ Watson Research Center, US)

Joint work of Khandekar, Rohit; Schieber, Baruch; Shachnai, Hadas; Tamir, Tami

We consider the following fundamental scheduling problem. The input consists of n jobs to be scheduled on a set of machines of bounded capacities. Each job is associated with a release time, a due date, a processing time and demand for machine capacity. The goal is to schedule all of the jobs non-preemptively in their release-time-deadline windows, subject to machine capacity constraints, such that the total busy time of the machines is minimized. Our problem has important applications in power-aware scheduling, optical network design and customer service systems. Scheduling to minimize busy times is APX-hard already in the special case where all jobs have the same (unit) processing times and can be scheduled in a fixed time interval.

Our main result is a 5-approximation algorithm for general instances. We extend this result to obtain an algorithm with the same approximation ratio for the problem of scheduling moldable jobs, that requires also to determine, for each job, one of several processing-time vs. demand configurations. Better bounds and exact algorithms are derived for several special cases, including proper interval graphs, intervals forming a clique and laminar families of intervals.

3.18 Bin Packing with Fixed Number of Bins Revisited

Ildiko Schlotter (Budapest Univ. of Technology & Economicsn, HU)

Joint work of Jansen, Klaus; Kratsch, Stefan; Marx, Dániel; Schlotter, Ildiko

As Bin Packing is NP-hard already for $k = 2$ bins, it is unlikely to be solvable in polynomial time even if the number of bins is a fixed constant. However, if the sizes of the items are polynomially bounded integers, then the problem can be solved in time $n^{O(k)}$ for an input of length n by dynamic programming. We show, by proving the W[1]-hardness of Unary Bin Packing (where the sizes are given in unary encoding), that this running time cannot be improved to $f(k) \cdot n^{O(1)}$ for any function $f(k)$ (under standard complexity assumptions). On the other hand, we provide an algorithm for Bin Packing that obtains in time $2^{O(k \log^2 k)} + O(n)$ a solution with additive error at most 1, i.e., either finds a packing into $k+1$ bins or decides that k bins do not suffice.

3.19 Balanced Interval Coloring

Alexander Souza (HU Berlin, DE)

Joint work of Souza, Alexander; Antoniadis, Antonios; Hüffner, Falk; Lenzner, Pascal
URL http://arxiv.org/abs/1012.3932

We consider the discrepancy problem of coloring n intervals with k colors such that at each point on the line, the maximal difference between the number of intervals of any two colors is minimal. Somewhat surprisingly, a coloring with maximal difference at most one always exists. Furthermore, we give an algorithm with running time O(n log n + kn log k) for its construction. This is in particular interesting because many known results for discrepancy problems are non-constructive.

This problem naturally models a load balancing scenario, where n tasks with given start- and end-times have to be distributed among k servers.

Our results imply that this can be done ideally balanced.

When generalizing to d dimensional boxes (instead of intervals), a solution with difference at most one is not always possible. We show that for any d > 1 and any k > 1 it is NP-complete to decide if such a solution exists, which implies also NP-hardness of the respective minimization problem.

In an online scenario, where intervals arrive over time and the color has to be decided upon arrival, the maximal difference in the size of color classes can become arbitrarily high for any online algorithm.

3.20 Fast Separation Algorithms for Multidimensional Assignment Problems

Frits C.R. Spieksma (K.U. Leuven, BE)

In polyhedral combinatorics, the polytope corresponding to an integer programming formulation of a combinatorial optimization problem is examined in order to obtain families of valid inequalities. To incorporate such families of inequalities within a cutting plane algorithm requires an additional step: determining whether an inequality of a specific family is violated by a given vector x (the separation problem). The idea put forward in this work is to consider a compact representation of this given vector x, and to measure the complexity of a separation algorithm in terms of this compact representation.

We illustrate this idea on the separation problem of well-known families of inequalities associated to the (multi-index) assignment polytope, and we show that for these families of inequalities, better time-complexities than the current ones are possible.

3.21 Hardness of Shops and Optimality of List Scheduling

Ola Svensson (KTH - Stockholm, SE)

We give an overview of the implications and techniques of the following hardness of approximation results:

- Non-constant inapproximability results for various shop scheduling problems that essentially match the best known approximation algorithm for acyclic job shops and general flow shops.

- A 2 hardness assuming the unique games conjecture for the problem of scheduling jobs with precedence constraints on identical machines so as to minimize the makespan. This matches the classical 2-approximation algorithm by Graham from 66.

3.22 Scheduling with Bully Selfish Jobs

Tami Tamir (The Interdisciplinary Center - Herzliya, IL)

In job scheduling with precedence constraints, $i < j$ means that job j cannot start being processed before job i is completed. In this paper we consider selfish bully jobs who do not let other jobs start their processing if they are around. Formally, we define the selfish precedence–constraint where $i <_s j$ means that j cannot start being processed if i has not started its processing yet. Interestingly, as was detected by a devoted kindergarten teacher whose story is told below, this type of precedence constraints is very different from the traditional one, in a sense that problems that are known to be solvable efficiently become NP-hard and vice-versa. The work of our hero teacher, Ms. Schedule, was initiated due to an arrival of bully jobs to her kindergarten. Bully jobs bypass all other nice jobs, but respect each other. This natural environment corresponds to the case where the selfish precedence-constraints graph is a complete bipartite graph. Ms. Schedule analyzed the minimum makespan and the minimum total flow-time problems for this setting. She then extended her interest to other topologies of the precedence constraints graph and other special instances with uniform length jobs and/or release times. Finally, she defined a generalization of her problem, where the precedence constraints graph is weighted, and $w(i, j)$ specifies the minimal gap between the starting times of i and j. The paper was presented in FUN with Algorithms 2010.

3.23 How to use Lagrangian-Relaxation Algorithms to solve Packing and Covering Problems

Neal E. Young (Univ. of California - Riverside, US)

Following a brief review of the history of Lagrangian-relaxation algorithms, I will summarize recent results in the area in a concrete form that (hopefully) makes it easy to understand how to apply the results.

Given any linear program (LP) that includes some packing constraints and/or some covering constraints, the packing and/or covering constraints can be "dualized", replacing the packing constraints by a carefully chosen linear combination of the packing constraints, and likewise for the covering constraints. This replaces all m packing/covering constraints by just one or two constraints, and gives an LP relaxation LP' of the problem that is combinatorially simpler than the original problem.

Given any algorithm alg' for the simpler problem LP', there is a simple algorithm for the original problem that calls $alg'O(min(m, width) * log(m)/epsilon^2)$ times, then returns an epsilon-approximate solution to the original problem.

I will illustrate the ideas using zero-sum matrix games, the Held-Karp lower bound on TSP, the "configuration LP" for bin packing, and on multi-commodity flow problems.

3.24 A Truthful Constant Approximation for Maximizing the Minimum Load on Related Machines

Rob van Stee (MPI für Informatik - Saarbrücken, DE)

Joint work of Christodoulou, Giorgos; Kovacs, Annamaria; van Stee, Rob

Designing truthful mechanisms for scheduling on related machines is a very important problem in single-parameter mechanism design. We consider the covering objective, that is we are interested in maximizing the minimum completion time of a machine. This problem falls into the class of problems where the optimal allocation can be truthfully implemented. A major open issue for this class is whether truthfulness affects the polynomial-time implementation.

We provide the first constant factor approximation for deterministic truthful mechanisms. In particular we come up with a approximation guarantee of 2+eps, significantly improving on the previous upper bound of $min(m, (2 + eps)s_m/s_1)$.

4 Discussion notes

4.1 Current and Future Trends in Scheduling

Alexander Souza (HU Berlin, DE), souza@informatik.hu-berlin.de

Notes of the discussion on "Current and Future Trends in Scheduling" that took place at the Dagstuhl meeting 11091 on "Packing and Scheduling Algorithms for Information and Communication Services" from 27.2.2011 to 4.3.2011.

Theory and Applications

- *To what extent do scheduling problems in theory and practice relate?*
  1. "No relation", "There should be", "There are"
  2. Example of a project of a Steel company together with TU Berlin: Initiated at an OR conference in Germany; Company approached TU Berlin; Solution was implemented by TU Berlin, but no support was given; Approach was a dynamic programming algorithm combined with heuristics.
- *What does it bring you to be attached to reality?*
  More satisfying research; Algorithm engineering; Per-Instance-Guarantees; Modeling as an issue.

New Theoretical Promises and Challenges

- *Are there new theoretical problems that we need to work on?*
  (a) Inapproximability results: Long standing open questions; tight bounds
  (b) New variants of classical problems: Measures; Models; Green IT; Cloud Computing
  (c) Dynamic Aspects: Practically important; Maybe theoretically nice; Stochastic models; Observed distributions; Technically difficult; Modeling again an issue; Communication with other disciplines required (workshop); Availability of data; Storing solutions for later reuse; Markov chain models (for online scheduling)
  (d) Theoretical vs. practical results: Essentially same outline as with the Paging problem; "Why something does well"

Sustainability Domain

1. Example: Land-lot purchase; Can have scheduling components
2. Currently well funded; For example Carla Gomes

   `http://www.cs.cornell.edu/gomes/`

Per Instance Guarantees

1. Certificates of instance-wise approximation ratio
2. Without LP-bounds? Maybe by MILP solutions or lower bounds

Insights from Game Theory

- *What are the new insights gained from game theoretic approach to scheduling (Does it capture better than classic scheduling contemporary systems)?*

  (a) "Next question"
  (b) Canonical examples from Game Theory are scheduling and network design problems
  (c) Behavioral economics: Model how people behave; Maybe not accurately reflected in scheduling (pain-scheduling at a dentist); Human aspect of scheduling; Interaction of schedules with people; Indirectly done already; Find out criteria and objective function is an issue; "User happiness" is the objective function (in order to have an impact)

Stochastic Scheduling and Robustness

- Is *stochastic scheduling the 'right' direction for future research?* (*Can this direction be fruitful in view of the experience of the 70's?*)

  (a) Contacts with industry: Combining scheduling and routing; Transportation problems
  (b) Robustness: Varying data (small perturbations); Stable schedules

Integer Programming Approach in Scheduling

- *What do you think about integer programming as an approach for solving scheduling problems?*
  (a) CSP's are maybe better because more flexible
  (b) Problem: CPLEX not available; "Black magic"; Free solvers available at TU Berlin; Practical algorithms for large-scale scheduling problems are available
  (c) ILP research mostly in OR, but not so much in CS; More collaboration between OR and CS needed
  (d) Formulations matter

Theoretical Knowledge in Companies

1. Theoretically good algorithms rarely implemented for critical systems; Mostly prototypes
2. Examples for benefit of theoretical knowledge in companies
   a. Algorithm used for something it was not designed for, but it worked
   b. Akamai; Theoretical knowledge went into applications
   c. Start-up companies sometimes initiated by CS PhD's (also theoretical)

Personal Motivation

- *What drives your interest in the area of scheduling?*
  (a) Open fundamental problems; Optimization of resources (also in real life); New problems; Old problems; Get paid; Beauty
  (b) Disconnect between "formal motivation" (the introduction of your paper) and "personal motivation" (why you really do it)
  (c) Playground for new questions; Models; Techniques
  (d Can be explained to people

5 Open Problems

Notes of the "Open problems" session that took place at the Dagstuhl meeting 11091 on "Packing and Scheduling Algorithms for Information and Communication Services" from 27.2.2011 to 4.3.2011.

5.1 Implementing the Sum-of-Squares Bin-Packing Algorithm

David Johnson (AT&T, US)

The Sum-of-Squares bin packing algorithm (SS) is designed for instances in which the bin size B is an integer, as are all the item sizes. It is an online algorithm that chooses the bin into which to place the next item as follows: Let $c[i]$ be the number of bins in the current packing whose gap (B minus the sum of the sizes of the items already packed in the bin) equals i. Initially all the $c[i]$'s are 0. It chooses a bin into which to pack the item so as to minimize

$$\sum_{i=1}^{B-1} c[i]^2,$$

where the choice can be either an existing bin or a new bin with initial gap B.

It is straightforward to implement this algorithm to run in time $O(nB)$, where n is the number of items, whereas the classical Best-Fit algorithm (place each item in a bin with the smallest gap that will contain it) can be implemented to run in time $O(n \log B)$ by maintaining a priority queue for the non-zero values of $c[i]$.

For instances, when the number of item sizes is bounded by some constant J, SS can be implemented in time $O(nJ \log B)$, by maintaining a priority queue for each item size. But what if there is no such bound, or if $J = \Omega(B)$?

Our question: Is there an implementation of SS that, for any fixed B and without restriction on J, runs in time $o(nB)$?

For a detailed discussion of the Sum-of-Squares algorithm and its performance, see [1].

References

1 J. Csirik, D. S. Johnson, C. Kenyon, J. B. Orlin, P. W. Shor and R. R. Weber. On the Sum-of-Squares Algorithm for Bin Packing. *J. ACM* 53, pp. 1–65, 2006).

5.2 Covering by Rectangles: Is Slicing Essential?

Sivan Albagli-Kim (Technion, IL)

The following open problem explores the relation between *packing resizable items (PRI)* and geometric covering. As shown in [1], PRI is equivalent to the problem of *covering with holes (CwH)*, defined as follows. Given is a set $H_I = \{h_1, \cdots, h_n\}$ of n holes; each hole h_j is associated with a length $0 < q_j < X$ and a width $0 < p_j < Y$. We want to determine whether it is possible to cover an $X \times Y$ rectangle by holes in H_I. A cover is a placement of the holes. For each hole h_j, the solution specifies the x-interval $X_j = [x_{1j}, x_{2j}]$ in which h_j is spanned, such that $x_{2j} - x_{1j} = q_j$. An $X \times Y$ rectangle is covered if, for any $0 \le t \le X$, the total width of holes whose x-interval contains t is at least Y. Note that the holes need not be placed as rectangles and can be *sliced* along the y-axis. This type of cover models, e.g., applications in which the x-axis corresponds to time, and the y-axis corresponds to a resource whose allocation is not associated with specific location.

Figure 1 shows a cover of a 1×1 rectangle by 7 holes. Note, for example, that hole h_4 spans along $[0.6, 1]$ and its width is 0.5. Similarly, hole h_2 spans along $[0, 0.6]$ and its width is 0.3. We also note that it is possible to have overlapping holes, as well as holes whose intervals span beyond the covered area.

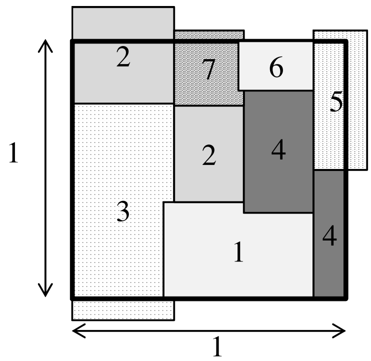

Figure 1 Covering a 1×1-rectangle by 7 holes.

In the *Covering with Rectangles (CwR)* problem, given is a set $R_I = \{r_1, \cdots, r_n\}$ of n rectangles, such that each rectangle r_j is associated with a length $0 < q_j < X$ and a hight $0 < p_j < Y$. We need to determine whether it is possible to cover an $X \times Y$ rectangle with rectangles in R_I. A cover is a placement of the rectangles. For each rectangle r_j, the solution specifies the x-interval $X_j = [x_{1j}, x_{2j}]$ such that $x_{2j} - x_{1j} = q_j$, and the y-interval $Y_j = [y_{1j}, y_{2j}]$ such that $y_{2j} - y_{1j} = p_j$. A solution covers an $X \times Y$ rectangle if, for any $0 \le t \le X$, the total hight of rectangles whose x-interval contains t is at least Y. Note that, unlike the CwH problem, in CwR the rectangles are rigid (and therefore, cannot be sliced).

Let $H_I = R_I$. Clearly, for all X, Y, a positive answer for CwR implies a positive answer for CwH. However, does the reverse hold, namely, does the existence of a solution for CwH imply the existence of a solution for CwR?

This open problem was settled during the seminar. We thank Jiří Sgall for the following example, which implies that the answer to the above is NO. The input consists of 8 holes: $(a)3 \times 2, (b)2 \times 1, (c)1 \times 4, (d)3 \times 2, (e)2 \times 1, (f)1 \times 4, (g)1 \times 3, (h)1 \times 1$. The holes need to cover a 4×7 rectangle. As shown in Figure 2, there exists a solution for CwH (in which g is sliced); however, there is no solution for CwR with this set of rectangles.

References

1 S. Albagli-Kim, H. Shachnai and T. Tamir. Approximation Algorithms for Packing Resizable Items and Covering by Holes. *Submitted.*

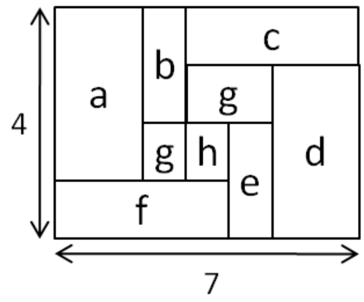

Figure 2 A solution for covering with holes in which slicing is essential.

5.3 Fixed-parameter Tractable Scheduling Problems

Dániel Marx (Humbold-Universität zu Berlin, DE)

Recall that a problem is *fixed-parameter tractable (FPT)* with some parameter k if it can be solved in time $f(k) \cdot n^{O(1)}$, where n is the input size and f is an arbitrary computable function depending only on k. Typically, if a problem is solvable in polynomial time for every fixed value of the parameter k (for example, there is an algorithm with running time $O(n^k)$), then it makes sense to ask if the problem is FPT, i.e., if we can remove k from the exponent of n and make it a multiplicative factor.

Compared to graph algorithms and other applications, there is surprisingly little work on the fixed-parameter tractability of scheduling problems (see [1,2] for a few examples). One problem is that it is not obvious how to choose relevant parameters that lead to interesting positive results. For example, the number k of processors is an obvious choice for the parameter, but a large fraction of the scheduling problems is NP-hard already for constant number of processors (and hence unlikely to be FPT with respect to this parameter).

A parameter which looks much more promising for obtaining fixed-parameter tractability results is the *number of rejected jobs.* Consider *any* scheduling problem that can be solved optimally in polynomial time. Then we can extend the problem by allowing rejections: the input contains an additional integer k, and the solution has to schedule all but k jobs. Assuming that the original problem is polynomial-time solvable, it is clear that the extended problem can be solved in $n^{O(k)}$ time: we first guess which k of the jobs to reject and solve the problem optimally for the remaining jobs. However, it is not obvious if the extended problem is fixed-parameter tractable parameterized by k. This question can be raised for any polynomial-time solvable scheduling problem and could be potentially interesting to explore. The open question is to find concrete scheduling problems, where the extended version with rejected jobs is NP-hard, but fixed-parameter tractable.

References

1 Fellows, M., and McCartin, C. On the parameterized complexity of minimizing tardy tasks. Theoretical Computer Science A 298 (2003), 317-324.
2 Bodlaender, H., and Fellows, M. On the complexity of k-processor scheduling. Operations Research Letters 18 (1995), 93-98.

5.4 Scheduling with Buffering on the Line

Adi Rosén (CNRS, FR)

We consider directed linear communication networks. The linear network consists of n nodes, $\{1, \ldots, n\}$, and $n-1$ directed edges, $(i, i+1)$, for $1 \leq i \leq n-1$. The system is synchronous, and at any time step, each edge can transmit one message. In one version of the problem, each node can store at any time an *infinite* number of messages. We are given a set $\mathcal{M}$, $|\mathcal{M}| = M$, of messages. Each message $m = (s_m, t_m, r_m, d_m) \in \mathcal{M}$ consists of a source node s_m, a target node t_m, a release time r_m, and a deadline d_m. For a message m, we define the slack of m, σ_m, to be $m = (d_m - r_m) - (t_m - s_m)$ (this is the number of steps the message can be idle and still make it to its destination by its deadline.).We define $\Sigma = \max_{m \in \mathcal{M}} \sigma_m$. We want to find a schedule for the messages that maximizes the number of messages that arrive to their destinations by their respective deadlines.

The open problem is whether there exists a polynomial-time algorithm with constant approximation ratio.

The problem is NP-hard [2]. A polynomial-time algorithm with approximation ratio $O(\min\{\log^* n, \log^* \Sigma, \log^* M\})$ is known [3].

References

1 Micah Adler, Sanjeev Khanna, Rajmohan Rajaraman, and Adi Rosén. Time-constrained scheduling of weighted packets on trees and meshes. *Algorithmica*, 36(2), pp. 123–152, 2003.
2 Micah Adler, Arnold L. Rosenberg, Ramesh K. Sitaraman, and Walter Unger. Scheduling time-constrained communication in linear networks. *Theory of Computing Systems* 35(6), pp. 599–623, 2002.
3 H. Räcke, A. Rosén, Approximation Algorithms for Time-Constrained Scheduling on Line Networks. In *Proc. of the 21st ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pp. 337–346, 2009.

5.5 Wireless Scheduling

Magnús Halldórsson (Reykjavik University, IS)

Let $p_1, p_2, \ldots, p_n$ be points on the real line with *capacities* $c_1, \ldots, c_n$. The problem is to partition $P = \{p_i\}$ into fewest sets $P_1, \ldots, P_t$, such that

$$\sum_{p' \in P_i, p' \neq p} |p - p'|^3 \leq c_i, \quad \text{for each } i \text{ and each } p \in P_i.$$

We seek an $O(1)$-approximation.

This problem statement captures the most basic open question in scheduling wireless links under the physical (or, SINR) model. Normally, links are given as sender-receiver pairs, but it is known that when messages are all transmitted with the same uniform power, we can blur the distinction between sender and receiver, by paying a constant factor. The problem is usually specified on the plane, or in a general distance metric, but results for the one-dimensional case can typically be generalized relatively easily. The exponent "3", known as the path-loss constant, is situation dependent, and can be any number between 2 and 6.

An $O(1)$-approximation is known for the throughput problem of finding a single set P_1 of maximum cardinality within which all points satisfy the inequality above [1]. This immediately gives an $O(\log n)$-factor, but no better is known.

References

1 O. Goussevskaia, M. M. Halldorsson, R. Wattenhofer, and E. Welzl. Capacity of Arbitrary Wireless Networks. In *INFOCOM*, pages 1872–1880, April 2009.

5.6 Feedback Arc Set Problems with Near-metric Weights

Monaldo Mastrolilli (IDSIA Lugano, CH)

Introduction

The Minimum Feedback Arc Set problem (MinFAS) is a fundamental and classical combinatorial optimization problem that finds application in many different settings that span from circuit design, constraint satisfaction problems, artificial intelligence, scheduling, etc. (see e.g. Chapter 4 in [19] for a survey). For this reason it has been deeply studied since the late 60's (see, e.g., [17]).

Its input consists of a set of vertices V and nonnegative weights $\{w_{(i,j)}, w_{(j,i)} : \{i,j\} \subseteq V\}$ for every oriented pair of vertices. The goal is to find a permutation π that minimizes $\sum_{\pi(i)<\pi(j)} w_{(i,j)}$, i.e. the weight of pairs of vertices that comply with the permutation[1]. A *partially ordered set* (*poset*) $\mathbf{P} = (V, P)$, consists of a set V and a partial order P on V, i.e., a reflexive, antisymmetric, and transitive binary relation P on V, which indicates that, for certain pairs of elements in the set, one of the elements precedes the other. In the *constrained* MinFAS (see [23]) we are given a partially ordered set $\mathbf{P} = (V, P)$ and we want to find a linear extension of $\mathbf{P}$ of minimum weight.

MinFAS was contained in the famous list of 21 NP-complete problems by Karp [14]. Despite intensive research for almost four decades, the approximability of this problem remains very poorly understood due to the big gap between positive and negative results. It is known to be APX-hard [13], but no constant approximation ratio has been found yet. The best known approximation algorithm achieves a performance ratio $O(\log n \log\log n)$ [21, 10, 9], where n is the number of vertices of the digraph. Closing this approximability gap is a well-known major open problem in the field of approximation algorithms (see e.g. [25], p. 337). Very recently and conditioned on the Unique Games conjecture, it was shown [11] that for every constant $C > 0$, it is NP-hard to find a C-approximation to the MinFAS.

Important ordering problems can be seen as special cases of MinFAS with restrictions on the weighting function. Examples of this kind are provided by ranking problems related to the aggregation of inconsistent information, that have recently received a lot of attention [1, 2, 15, 24]. Several of these problems can be modeled as (constrained) MinFAS with weights satisfying either *triangle inequalities* (i.e., for any triple i, j, k, $w_{(i,j)} + w_{(j,k)} \geq w_{(i,k)}$), or *probability constraints* (i.e., for any pair i, j, $w_{(i,j)} + w_{(j,i)} = 1$). Ailon, Charikar and

[1] Different, but equivalent formulations are often given for the problem.

Newman [2] give the first constant-factor randomized approximation algorithm for the unconstrained MinFAS problem with weights that satisfy the triangle inequalities. When the probability constraints hold, Mathieu and Schudy [15] obtain a PTAS. The currently best known constant approximation algorithms for the (constrained) MinFAS with triangle inequalities on the weights can be found in [1, 24]. Another prominent example is given by a classical problem in scheduling, namely the precedence constrained single machine scheduling problem to minimize the weighted sum of completion times, denoted as $1|prec|\sum w_j C_j$ (see e.g. [16] and [12] for a 2-approximation algorithm). This problem can be seen as a constrained MinFAS where the weight of arc (i,j) is equal to the product of two numbers $p_i \cdot w_j$: p_i is the processing time of job i and w_j is a weight associated to job j (see [3, 4, 7, 8] for recent advances).

The (constrained) MinFAS can be described by the following natural (compact) ILP using linear ordering variables $\delta_{(i,j)}$ (see e.g. [24]): variable $\delta_{(i,j)}$ has value 1 if vertex i precedes vertex j in the corresponding permutation, and 0 otherwise.

$$\text{[FAS]} \qquad \min \sum_{i \neq j} \delta_{(i,j)} w_{(i,j)} \tag{1a}$$

$$\text{s.t.} \qquad \delta_{(i,j)} + \delta_{(j,i)} = 1, \qquad \text{for all distinct } i,j \tag{1b}$$

$$\delta_{(i,j)} + \delta_{(j,k)} + \delta_{(k,i)} \geq 1, \qquad \text{for all distinct } i,j,k \tag{1c}$$

$$\delta_{(i,j)} = 1, \qquad \text{for all } (i,j) \in P \tag{1d}$$

$$\delta_{(i,j)} \in \{0,1\}, \qquad \text{for all distinct } i,j \tag{1e}$$

Constraint (1b) ensures that in any feasible permutation either vertex i is before j or vice versa. The set of Constraints (1c) is used to capture the transitivity of the ordering relations (i.e., if i is ordered before j and j before k, then i is ordered before k, since otherwise by using (1b) we would have $\delta_{(j,i)} + \delta_{(i,k)} + \delta_{(k,j)} = 0$ violating (1c)). Constraints (1d) ensure that the returned permutation complies with the partial order P.

To some extent, one source of difficulty that makes the MinFAS hard to approximate within any constant is provided by Constraint (1b). To see this, consider, for the time being, the unconstrained MinFAS. The following covering relaxation obtained by relaxing Constraint (1b) behaves very differently with respect to approximation.

$$\min \sum_{i \neq j} \delta_{(i,j)} w_{(i,j)} \tag{2a}$$

$$\text{s.t.} \qquad \delta_{(i,j)} + \delta_{(j,i)} \geq 1, \qquad \text{for all distinct } i,j \tag{2b}$$

$$\delta_{(i,j)} + \delta_{(j,k)} + \delta_{(k,i)} \geq 1, \qquad \text{for all distinct } i,j,k \tag{2c}$$

$$\delta_{(i,j)} \in \{0,1\}, \qquad \text{for all distinct } i,j \tag{2d}$$

Problem (2) is a special case of the vertex cover problem in hypergraphs with edges of sizes at most 3. It admits "easy" constant approximate solutions, whereas problem (1) does not seem to have any constant approximation [11]. Moreover, the fractional relaxation of (2), obtained by dropping the integrality requirement, is a positive linear program and therefore fast NC approximation algorithms exists: Luby and Nissan's algorithm [18] computes a feasible $(1+\varepsilon)$-approximate solution in time polynomial in $1/\varepsilon$ and $\log N$, using $O(N)$ processors, where N is the size of the input (fast approximate solution can also be obtained through the methods of [20]). On the other side, the linear program relaxation of (1) is not positive.

In a recent (unpublished, but available upon request) paper we show that the covering relaxation (2) is an "optimal" relaxation, namely, a *proper* formulation, for the unconstrained

MinFAS when the weights satisfy the triangle inequalities. More precisely, we show that any α-approximate solution to (2) can be turned in polynomial time into an α-approximate solution to (1), for any $\alpha \geq 1$ and when the weights satisfy the triangle inequalities. The same claim applies to fractional solutions. (We also observe that the same result does not hold when the weights satisfy the probability constraints.)

Interestingly, a compact covering formulation can be also obtained for the more general setting with precedence constraints. In this case we need to consider the following covering relaxation[2] which generalizes (2) to partially ordered sets $\mathbf{P} = (V, P)$.

$$\min \sum_{i \neq j} \delta_{(i,j)} w_{(i,j)} \tag{3a}$$

$$\text{s.t.}\quad \delta_{(x_1,y_1)} + \delta_{(x_2,y_2)} \geq 1, \qquad (x_2,y_1),(x_1,y_2) \in P \tag{3b}$$

$$\delta_{(x_1,y_1)} + \delta_{(x_2,y_2)} + \delta_{(x_3,y_3)} \geq 1, \qquad (x_2,y_1),(x_3,y_2),(x_1,y_3) \in P \tag{3c}$$

$$\delta_{(i,j)} \in \{0,1\}, \qquad \text{for all distinct } i,j \tag{3d}$$

Open problems

The constrained MinFAS problem admits a natural covering formulation with an exponential number of constraints (see e.g. [5]):

$$\min \sum_{(i,j)} \delta_{(i,j)} w_{(i,j)} \tag{4a}$$

$$\text{s.t.} \sum_{i=1}^{c} \delta_{(x_i,y_i)} \geq 1, \qquad \text{for all } c \geq 2,\ (x_i,y_i)_{i=1}^{c} \text{ s.t. } (x_i,y_{i+1}) \in P \tag{4b}$$

$$\delta_{(i,j)} \in \{0,1\}, \qquad \text{for all distinct } i,j \tag{4c}$$

The condition $(x_i, y_{i+1}) \in P$ in constraint (4b) is to be read cyclically, i.e. $(x_c, y_1) \in P$. The hyperedges in this vertex cover problem are exactly the alternating cycles of poset P (see e.g. [22]).

We know that when the weights satisfy the triangle inequality then we can drop from (4) all the constraints of size strictly larger than three. Generalizing, it would be nice to prove/disprove the following statement that we conjecture to be true.

▶ Hypothesis 1. When the weights satisfy the k-gonal inequalities[3], then there exists a constant $c(k)$, whose value depends on k, such that a proper formulation for the constrained MinFAS problem can be obtained by dropping from (4) all the constraints of size strictly larger than $c(k)$.

Moreover, it would be nice to use the large literature and techniques developed for covering problems to improve the best known ratios for MinFAS with (near-)metric weights. This was actually the case for the scheduling problem $1|prec|\sum w_j C_j$: in [3, 8] it was first shown that the structure of the weights for this problem allows for all the constraints of size strictly larger than two to be ignored, therefore the scheduling problem can be seen as a special case of the vertex cover problem. The established connection proved later to be very valuable both for positive and negative results: studying this graph yielded a framework that

[2] It is a relaxation to constrained MinFAS since if either Constraint (3b) or (3c) was violated then we would have a cycle.

[3] For all $a_1, \ldots, a_k \in V$ the following holds: $w_{(a_1,a_k)} \leq w_{(a_1,a_2)} + \ldots + w_{(a_{k-1},a_k)}$.

unified and improved upon previously best-known approximation algorithms [4, 6]; moreover, it helped to obtain the first inapproximability results for this old problem [7] by revealing more of its structure.

References

1 N. Ailon. Aggregation of partial rankings, -ratings and top- lists. *Algorithmica*, 57(2):284–300, 2010.
2 N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5), 2008.
3 C. Ambühl and M. Mastrolilli. Single machine precedence constrained scheduling is a vertex cover problem. *Algorithmica*, 53(4):488–503, 2009.
4 C. Ambühl, M. Mastrolilli, N. Mutsanas, and O. Svensson. Scheduling with precedence constraints of low fractional dimension. In *Proceedings of IPCO 2007*, volume LNCS 4168, pages 28–39. Springer, 2007.
5 C. Ambühl, M. Mastrolilli, N. Mutsanas, and O. Svensson. Precedence constraint scheduling and connections to dimension theory of partial orders. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 95:45–58, 2008.
6 C. Ambühl, M. Mastrolilli, and O. Svensson. Approximating precedence-constrained single machine scheduling by coloring. In *Proceedings of the APPROX + RANDOM*, volume LNCS 4110, pages 15–26. Springer, 2006.
7 C. Ambühl, M. Mastrolilli, and O. Svensson. Inapproximability results for sparsest cut, optimal linear arrangement, and precedence constraint scheduling. In *Proceedings of FOCS 2007*, pages 329–337, 2007.
8 J. R. Correa and A. S. Schulz. Single machine scheduling with precedence constraints. *Mathematics of Operations Research*, 30(4):1005–1021, 2005.
9 G. Even, J. Naor, S. Rao, and B. Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *J. ACM*, 47(4):585–616, 2000.
10 G. Even, J. Naor, B. Schieber, and M. Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.
11 V. Guruswami, R. Manokaran, and P. Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *FOCS*, pages 573–582, 2008.
12 L. A. Hall, A. S. Schulz, D. B. Shmoys, and J. Wein. Scheduling to minimize average completion time: off-line and on-line algorithms. *Mathematics of Operations Research*, 22:513–544, 1997.
13 V. Kann. *On the Approximability of NP-Complete Optimization Problems*. PhD thesis, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, 1992.
14 R. Karp. *Reducibility Among Combinatorial Problems*, pages 85–103. Plenum Press, NY, 1972.
15 C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *STOC*, pages 95–103, 2007.
16 E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. Sequencing and scheduling: Algorithms and complexity. In S. C. Graves, A. H. G. Rinnooy Kan, and P. Zipkin, editors, *Handbooks in Operations Research and Management Science*, volume 4, pages 445–552. North-Holland, 1993.
17 A. Lempel and j. . I. n. . . v. . . y. . . p. . . I. Cederbaum, title = Minimum feedback arc and vertex sets of a directed graph.
18 M. Luby and N. Nisan. A parallel approximation algorithm for positive linear programming. In *STOC*, pages 448–457, 1993.
19 P. Pardalos and D. Du. *Handbook of Combinatorial Optimization: Supplement*, volume 1. Springer, 1999.

20 A. Plotkin, D. Shmoys, and E. Tardos. Fast Approximation Algorithms for Fractional Packing and Covering Problems. *Mathematics of Operation Research*, 20, 1995.

21 P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.

22 W. T. Trotter. *Combinatorics and Partially Ordered Sets: Dimension Theory.* Johns Hopkins Series in the Mathematical Sciences. The Johns Hopkins University Press, 1992.

23 A. van Zuylen, R. Hegde, K. Jain, and D. P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. In *SODA*, pages 405–414, 2007.

24 A. van Zuylen and D. P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Math. Oper. Res.*, 34(3):594–620, 2009.

25 V. V. Vazirani. *Approximation Algorithms.* Springer, 2001.

Participants

- Sivan Albagli
Technion – Haifa, IL
- Evripidis Bampis
UPMC – Paris, FR
- Niv Buchbinder
Open University – Israel, IL
- Ed G. Coffman Jr.
Columbia University, US
- Pierre-Francois Dutot
INRIA Rhône-Alpes, FR
- Leah Epstein
University of Haifa, IL
- Lisa K. Fleischer
Dartmouth Coll. – Hanover, US
- Michael D. Grigoriadis
Rutgers Univ. – Piscataway, US
- Magnús M. Halldórsson
Reykjavik University, IS
- Klaus Jansen
Universität Kiel, DE
- David S. Johnson
AT&T Res. – Florham Park, US
- Howard Karloff
AT&T Res. – Florham Park, US
- Marek Karpinski
Universität Bonn, DE
- Samir Khuller
University of Maryland, US
- Asaf Levin
Technion – Haifa, IL
- Alejandro Lopez-Ortiz
University of Waterloo, CA
- Dániel Marx
HU Berlin, DE
- Monaldo Mastrolilli
IDSIA – Lugano, CH
- Claire Mathieu
Brown Univ. – Providence, US
- Ernst W. Mayr
TU München, DE
- Nicole Megow
MPI für Informatik – Saarbrücken, DE
- Rolf H. Möhring
TU Berlin, DE
- Seffi Naor
Technion – Haifa, IL
- Kirk Pruhs
University of Pittsburgh, US
- Christina Robenek
Universität Kiel, DE
- Adi Rosén
Univ. Paris-Diderot, CNRS, FR
- Nicolas Schabanel
Univ. Paris-Diderot, CNRS, FR
- Baruch Schieber
IBM TJ Watson Res. Center, US
- Ildiko Schlotter
Budapest Univ. of Technology & Economics, HU
- Ilka Schnoor
Universität Kiel, DE
- Jiri Sgall
Charles University – Prague, CZ
- Hadas Shachnai
Technion – Haifa, IL
- Alexander Souza
HU Berlin, DE
- Frits C.R. Spieksma
K.U. Leuven, BE
- Ola Svensson
KTH – Stockholm, SE
- Tami Tamir
The Interdisciplinary Center – Herzliya, IL
- Rob van Stee
MPI für Informatik - Saarbrücken, DE
- Gerhard Woeginger
TU Eindhoven, NL
- Prudence W. H. Wong
University of Liverpool, GB
- Neal E. Young
University of California – Riverside, US
- Shmuel Zaks
Technion – Haifa, IL