Report from Dagstuhl Seminar 13111

Scheduling

Edited by

Susanne Albers[1], Onno J. Boxma[2], and Kirk Pruhs[3]

1 **HU Berlin, DE,** albers@informatik.hu-berlin.de
2 **TU Eindhoven, NL,** o.j.boxma@tue.nl
3 **University of Pittsburgh, US,** kirk@cs.pitt.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 13111 "Scheduling". The primary objective of the seminar is to facilitate dialog and collaboration between researchers in two different mathematically-oriented scheduling research communities, the stochastic scheduling and queuing community, and the worst-case approximation scheduling community.

Seminar 10.–15. March, 2013 – www.dagstuhl.de/13111
1998 ACM Subject Classification C.4 Performance of Systems
Keywords and phrases Scheduling, Algorithms, Stochastic, Optimization, Queuing
Digital Object Identifier 10.4230/DagRep.3.3.24

1 Executive Summary

Susanne Albers
Onno J. Boxma
Kirk Pruhs

The primary objective of the seminar is to facilitate dialog and collaboration between researchers in two different mathematically-oriented scheduling research communities, the stochastic scheduling and queuing community, and the worst-case approximation scheduling community. To a large extent, the applications considered by the two communities are the same. The stochastic community considers questions related to determining stochastic information (like the expectation or tail bounds) about the performance of algorithms and systems from stochastic information about the input. The worst-case community considers questions related to determining the worst-case performance of algorithms and systems assuming no stochastic information about the input. Each community has developed its own set of mathematical techniques that are best suited to answer these different sorts of questions. While addressing similar problems, these communities tend to attend different conferences (e.g. SIGMETRICS vs. SODA/IPCO), and publish in different journals. Thus the organizers believed that each community would benefit from greater interaction with the other community, and this seminar was an opportunity to further such interaction. The seminar was attended by about 15 researchers from the stochastic community and 40 researchers from the worst-case community.

Scheduling, *Dagstuhl Reports*, Vol. 3, Issue 3, pp. 24–50
Editors: Susanne Albers, Onno J. Boxma, and Kirk Pruhs
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

2 Table of Contents

Executive Summary
Susanne Albers, Onno J. Boxma, and Kirk Pruhs 24

Overview of Talks

Optimal Scheduling Problem for Scalable Queues
Samuli Aalto 27

Online Myopic Network Covering
Konstantin Avrachenkov 28

Scheduling with time-varying capacities
Urtzi Ayesta 29

Online Scheduling in the Cloud
Yossi Azar 29

Worst Case and Stochastic Analysis in Scheduling: Similarities, Differences, and Bridges
Nikhil Bansal, Urtzi Ayesta, and Onno Boxma 29

Clustered Scheduling of Real-Time Tasks
Vincenzo Bonifaci 30

Wireless Random-Access Algorithms: Fluid Limits and Delay Issues
Sem Borst 30

Dominance rules for $1||\sum w_j C_j^\beta$
Christoph Dürr 31

Stochastic Knapsacks and Matchings
Anupam Gupta 31

When Does Stochasticity/Discreteness Matter? Fluid Models and Scheduling Queueing Networks
John Hasenbein 32

Complexity of generalized min-sum scheduling
Wiebke Höhn 32

The generalization of scheduling with machine cost
Csanád Imreh 33

Online Scheduling with General Cost Functions
Sungjin Im 34

Scheduling of Users with Time-Varying Service Rates
Peter Jacko 35

Employee scheduling and rescheduling in call centers
Ger Koole 36

Online scheduling algorithms analyzed by Dual Fitting
Amit Kumar 36

Approximation in Dynamic Stochastic Scheduling
Nicole Megow 36

MapReduce and Distributed Scheduling
Benjamin Moseley 37

Weakly Coupled Stochastic Decision Systems
Kamesh Munagala . . . 37

Stochastic k-TSP
Viswanath Nagarajan . . . 37

Job Scheduling Mechanisms for Large Computing Clusters
Seffi Naor . . . 38

Scheduling in queues with customer grouping
Sindo Nunez Queija . . . 38

Stochastic Comparison of Multicast Pull and Push
Kirk Pruhs . . . 39

Scheduling with time-varying cost: Deterministic and stochastic models
Roman Rischke . . . 39

Online scheduling of jobs with fixed start times on related machines
Jiri Sgall . . . 40

Scheduling Advertising Campaigns
Hadas Shachnai . . . 41

Optimal queue-size scaling in switched networks
Devavrat Shah . . . 41

On Carry Over
Frits C. R. Spieksma . . . 41

Stochastic Optimal Control for a Class of Dynamic Resource Allocation Problems
Mark S. Squillante . . . 42

Truthful Scheduling
Rob van Stee . . . 43

Online Stochastic Matching
Clifford Stein . . . 44

Split scheduling with uniform setup times
Suzanne van der Ster . . . 44

An infinite server system with customer-to-server packing constraints
Alexander Stolyar . . . 45

Analysis of Smith's Rule in Stochastic Scheduling
Marc Uetz . . . 46

Appointment Scheduling with Slot Blocking
Peter van de Ven . . . 46

Learning in Stochastic Scheduling
Tjark Vredeveld . . . 47

FCFS infinite matching, queues with skill based routing, and organ transplants
Gideon Weiss . . . 47

Working Groups . . . 48

Open Problems . . . 49

Participants . . . 50

3 Overview of Talks

The opening talk on Monday morning was presented by Nikhil Bansal, Urtzi Ayesta, Onno Boxma (with materials contributed by Adam Wiermam). The purpose of this talk was to introduce the research approaches of each community to the other community, to a large extent using the two community's research on scheduling speed scalable processors as a running example. Each of the remaining seven half-days contained one longer, 30-40 minute, talk from each community from a speaker invited by the organizers. The talks from the worst-case community were given by Amit Kumar, Anupam Gupta, Cliff Stein, Kamesh Munagala, Yossi Azar, Nicole Megow, and Ben Moseley. The talks from the stochastic community were given by Mark Squillante, Devavrat Shah, Gideon Weiss, John Hasenbein, Alexander Stoylar, Sem Borst, and Ger Koole. Again the goal was to try to introduce each community to important tools and approaches from the other community. The remaining meeting times were filled in by short, 7 minute, talks by the remaining participants. Speakers were encouraged to be forward leaning, and present ongoing or future work, or open problems. Many talks provoked lively discussion, which was allowed to continue to its natural conclusion. The time between lunch and the afternoon coffee break was left open for individual discussions and collaborations. Abstracts of the talks can be found below.

3.1 Optimal Scheduling Problem for Scalable Queues

Samuli Aalto (Aalto University, FI)

Modern wireless cellular systems are able to utilize the opportunistic scheduling gain originating from the variability in users' channel conditions. By favoring the users with good instantaneous channel conditions, the service capacity of the system can be increased with the number of users. On the other hand, for service systems with fixed service capacity, the system performance can be optimized by utilizing the job size information. Combining the advantages of size-based scheduling with the opportunistic scheduling gain has proven a challenging task.

Inspired by the opportunistic scheduling gain, Sadiq and de Veciana (2010) defined a new queueing model, which they called an $M/GI/\mathcal{C}$ queue. Briefly said, it is a service system where the service capacity scales with the number of jobs in the system. Thus, we called it a *scalable queue* in Aalto et. al (2011, 2012). More precisely said, the service capacity is defined via *capacity regions* $\mathcal{C}_n$, where n denotes to the number of jobs in the system.

Sadiq and de Veciana (2010) considered the optimal scheduling problem for a specific class of capacity regions, nested polymatroids, and found the optimal policy that minimizes the total sojourn time in the transient setting (without any arrivals). Aalto et al. (2011) generalized the result to compact and symmetric capacity regions, however, utilizing an additional implicit assumption. Aalto et al. (2012) demonstrated that the additional assumption, indeed, is satisfied by a class of wireless system models stemming from a time-scale separation assumption.

The optimal scheduling problem for scalable queues in the dynamic setting (with arrivals) is still completely open, as well as the problems with other types of objective functions.

References

1 Bilal Sadiq, Gustavo de Veciana: Balancing SRPT prioritization vs opportunistic gain in wireless systems with flow dynamics. Proceedings of ITC-22, 2010
2 Samuli Aalto, Aleksi Penttinen, Pasi Lassila, Prajwal Osti: On the optimal trade-off between SRPT and opportunistic scheduling. Proceedings of ACM SIGMETRICS, 2011
3 Samuli Aalto, Aleksi Penttinen, Pasi Lassila, Prajwal Osti: Optimal size-based opportunistic scheduler for wireless systems. Queueing Systems 72:5–30, 2012

3.2 Online Myopic Network Covering

Konstantin Avrachenkov (INRIA Sophia Antipolis – Méditerranée, FR)

Efficient marketing or awareness-raising campaigns seek to recruit n influential individuals – where n is the campaign budget – that are able to cover a large target audience through their social connections. So far most of the related literature on maximizing this network cover assumes that the social network topology is known. Even in such a case the optimal solution is NP-hard. In practice, however, the network topology is generally unknown and needs to be discovered on-the-fly. In this work we consider an unknown topology where recruited individuals disclose their social connections (a feature known as one-hop lookahead). The goal of this work is to provide efficient online algorithms that recruit individuals as to maximize the size of target audience covered by the campaign.

We analyze the performance of a variety of online algorithms currently used to sample and search large networks. We also propose a new greedy online algorithm, Maximum Expected d-Excess Degree (MEED), and provide, to the best of our knowledge, the first detailed theoretical analysis of the cover size of a variety of well known network sampling algorithms on finite networks. Our proposed algorithm greedily maximizes the expected size of the cover. For a class of random power law networks we show that MEED simplifies into a straightforward procedure, which we denote MOD (Maximum Observed Degree). Thus our problem gives an interesting linkage between deterministic and stochastic scheduling.

We note that performance may be further significantly improved if the node degree distribution is known or can be estimated online during the campaign. Thus, one open question is which algorithm would achieve an optimal trade off between exploration and exploitation of the network structure? Another open question: what is the approximation ratio of the greedy online algorithm?

Preliminary results on the topic can be found at:

References

1 K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro and D. Towsley, “Online Myopic Network Covering”, http://arxiv.org/abs/1212.5035

3.3 Scheduling with time-varying capacities

Urtzi Ayesta (LAAS – Toulouse, FR)

Classical results in size-based scheduling in a single-server queue show that giving preference to short flows is optimal in a wide variety of settings. However all these results typically assume that the speed of the server is constant over time and independent of the state of the queue. In this short talk we will show that when the capacity of the system is time-varying (either as a function of the state or as an exogenous process) giving preference to short flows is no longer necessarily optimal, which opens several interesting questions.

3.4 Online Scheduling in the Cloud

Yossi Azar (Tel Aviv University, IL)

Online task scheduling of jobs on cloud computing infrastructures poses new challenges theoretically and practically. In particular the number of machines or virtual machines(VM) is not fixed any more and the goal is to minimize the cost of the computation as well as minimize the delay or the load. We will discuss various models and questions in this area concerned with identical vs heterogeneous machines, fixed setup time vs arbitrary setup cost and single vs multi dimension job requirements.

The talk is based on three papers:
1. one will appear in SODA – unrelated machine scheduling with startup cost (paper is called Online mixed packing and covering)
2. second submitted to STOC – Online Vector Bin Packing
3. third (not submitted) is Cloud Scheduling with Setup Cost.

3.5 Worst Case and Stochastic Analysis in Scheduling: Similarities, Differences, and Bridges

Nikhil Bansal (TU Eindhoven, NL), Urtzi Ayesta (LAAS – Toulouse, FR), Onno J. Boxma (TU Eindhoven, NL)

Joint work of Nikhil Bansal, Adam Wierman, Urtzi Ayesta, and Onno Boxma

In order to provide background for the workshop participants, this talk will give a quick introduction to the two areas of worst case analysis and stochastic analysis of scheduling polices. We will introduce the basic notions in each of these areas, and then provide some examples of topics that have been studied by both of the communities. The goal of the talk will be to highlight the differences and similarities in the approaches for addressing these topics. We will also describe some examples of topics where the techniques from one community have proven useful in the other.

3.6 Clustered Scheduling of Real-Time Tasks

Vincenzo Bonifaci (National Research Council – Rome, IT)

We consider the scheduling of hard real-time tasks on multiple identical processors. Each such task is described by a worst-case execution time, a relative deadline, and a minimum interarrival time, and recurrently generates jobs with the same features. In the global approach, jobs can be arbitrarily preempted and migrated among the processors. In the partitioned approach, tasks are statically assigned to processors and then each processor is scheduled according to some fixed online policy, such as, say, Earliest Deadline First. Preemption within one processor is always allowed.

Deciding whether the system can be feasibly scheduled is NP-hard even in very simple cases. Therefore, we typically look for assignment and scheduling algorithms that guarantee at least the same service that can be provided by an optimal assignment and scheduling to slightly slower, or fewer, processors.

More recently, a "clustered" approach is being suggested. In the clustered approach, each task is assigned to a cluster of machines (e.g., a set of processor cores sharing a cache) and then globally scheduled within the cluster. This allows for more flexibility than in partitioned scheduling, while avoiding the disadvantages of a fully global approach where the overheads due to task migrations can be large. However, a quantitative study of the trade-offs achievable with the clustered approach is yet to be undertaken.

Pointers into the literature:

References

1 Sanjoy K. Baruah, Nathan Fisher: The Partitioned Multiprocessor Scheduling of Deadline-Constrained Sporadic Task Systems. IEEE Trans. Computers 55(7): 918–923 (2006)
2 Jian-Jia Chen, Samarjit Chakraborty: Resource Augmentation Bounds for Approximate Demand Bound Functions. RTSS 2011: 272–281
3 Andrea Bastoni, Björn B. Brandenburg, James H. Anderson: An Empirical Comparison of Global, Partitioned, and Clustered Multiprocessor EDF Schedulers. RTSS 2010:14–24

3.7 Wireless Random-Access Algorithms: Fluid Limits and Delay Issues

Sem C. Borst (TU Eindhoven, NL)

Queue-based wireless random-access algorithms are relatively simple and inherently distributed, yet provide a striking capability to match the optimal throughput performance of centralized scheduling mechanisms in a wide range of scenarios. Unfortunately, the specific type of activation rules for which throughput optimality has been established, may result in extremely long queues and delays. The use of more aggressive/persistent access schemes can improve the delay performance, but does not provide any universal maximum-stability guarantees.

In order to gain qualitative insights and examine stability properties, we investigate fluid limits where the system dynamics are scaled in space and time. Several distinct types of fluid limits can arise, ranging from ones with smooth deterministic features, to others which exhibit random oscillatory characteristics, depending on the topology of the network, in conjunction with the form of the activation rules. As we will show, these qualitatively different regimes are strongly related to short-term fairness measures and mixing times for random-access mechanisms with fixed activation rates, and carry significant implications for stability properties.

Note: based on joint work with Niek Bouman (TU/e), Javad Ghaderi (UIUC), Johan van Leeuwaarden (TU/e), Alexandre Proutiere (KTH), Peter van de Ven (IBM), Phil Whiting (Alcatel-Lucent Bell Labs), Alessandro Zocca (TU/e)

3.8 Dominance rules for $1||\sum w_j C_j^\beta$

Christoph Dürr (UPMC, Lab. LIP6 – Paris, FR)

We consider a simple scheduling problem on a single machine. Each job j has some given processing time p_j and a priority w_j. The goal is to find an ordering of the given jobs, that minimizes $\sum w_j C_j^\beta$, where β is a fixed positive constant.

Consider two jobs i, j. We say that we have the *local ordering* $i \prec_\ell j$, if for any instance containing i, j, every optimal schedule in which i, j are adjacent, the job i precedes j. Similarly we say that we have the *global ordering* $i \prec_g j$ if for any instance containing i, j every optimal schedule in which i, j are adjacent, the job i is scheduled before job j.

These ordering conditions are useful for any algorithm solving this problem. What are the conditions that imply $i \prec_\ell j$ or even $i \prec_g j$? This short talk gives an overview of what is known, and asks the question whether $i \prec_\ell j$ always implies $i \prec_g j$.

Pointers into the literature:

References

1 Höhn, W., Jacobs, T. An experimental and analytical study of order constraints for single machine scheduling with quadratic cost. In Proc. of the 14th Workshop on Algorithm Engineering and Experiments (ALENEX 2012).

3.9 Stochastic Knapsacks and Matchings

Anupam Gupta (Carnegie Mellon University – Pittsburgh, US)

I will survey some work on two stochastic packing problems: (a) stochastic knapsack where jobs of uncertain sizes and/or rewards are packed into a knapsack, and (b) stochastic matchings where uncertain edges are packed given a set of constraints. I'll focus on the techniques used for these problems (and how they give solutions for extensions and generalizations), and the many open questions.

3.10 When Does Stochasticity/Discreteness Matter? Fluid Models and Scheduling Queueing Networks

John Hasenbein (University of Texas – Austin, US)

We discuss the connection between scheduling multiclass stochastic and fluid (deterministic) networks. First we overview the relationship via a few of classic results and examples. Next, we present more recent research on a stochastic combinatorial scheduling problem in which the "macro" stochasticity must be taken into account, but the "micro" stochasticity is less important.

3.11 Complexity of generalized min-sum scheduling

Wiebke Höhn (TU Berlin, DE)

We consider a single-machine scheduling problem with generalized min-sum objective. Given a set of jobs $j = 1, \ldots, n$ with individual weights $w_j \geq 0$ and processing times $p_j \geq 0$, the goal is to assign the jobs to non-overlapping time intervals on the machine to minimize $\sum_{j=1}^{n} w_j g(C_j)$, where C_j denotes the completion time of job j in the schedule and where g is some non-decreasing cost function. Note that the question of allowing preemption does not play a role here, because the jobs do not have release times and so the possibility of preemption never leads to a cheaper optimal schedule.

Alternatively, we can interpret this problem as the scenario of linear cost but non-uniform processor speed. Assume that the processor speed is given by a nonnegative function s. Then, the total workload processed until time t is $S(t) := \int_0^t s(x)dx$. Conversely, if the total workload of job j and all jobs processed before it is P, then the completion time of j in the schedule is $S^{-1}(P)$. Hence, according to the above model, S^{-1} can be seen as cost function. Note that S^{-1} is always monotone, and it is continuous even if s is not. Moreover if s is increasing or decreasing then S^{-1} is concave or convex, respectively.

Finally, we would like to point out that this general scheduling problem also covers the *Airplane Refueling Problem* (ARP) whose complexity was proposed as an open problem by Gerhard Woeginger on the last Dagstuhl Schdeduling Workshop. The ARP translates one-to-one into the maximization variant of our problem with cost function $1/C_j$, and further into the minimization variant with concave cost function $C - 1/C_j$ for some sufficiently large constant C. Of course, this transformation does not preserve approximation guarantees. However, it does preserve classic complexity results.

Despite many recent approximation results, there are still many open problems concerning the complexity of the problem. For linear and exponential cost functions, the problem is in P, and it is known to be weakly NP-hard for tardiness cost $C_j - d$ with common due date d, i.e., for convex functions. Moreover, the problem is strongly NP-hard for piece-wise linear cost functions alternating between two speeds. On the positive side, Megow and Verschae designed a PTAS for general cost functions and an FPTAS for piece-wise linear functions with a constant number of linear segments. For general convex cost functions, it is open whether the problem is strongly NP-hard, and for concave function—and in particular (concave and convex) monomials C_j^k, $k > 0$—the complexity is completely open.

Pointers into the literature:

References

1 Wiebke Höhn, Tobias Jacobs: On the performance of Smith's rule in single-machine scheduling with nonlinear cost. LATIN 2012:482–493.

2 Nicole Megow, José Verschae: Dual techniques for scheduling on a machine with varying speed. ICALP 2013.
CoRR abs/1211.6216, 2012. http://arxiv.org/abs/1211.6216

3 Michael H. Rothkopf: Scheduling Independent Tasks on Parallel Processors. Management Science 12:437–447, 1966.

4 W. E. Smith: Various optimizers for single-stage production. Naval Research Logistics Quarterly 3:59–66, 1956.

5 Gerhard J. Woeginger: The airplane refueling problem. Open Problems – Scheduling, 2010. http://drops.dagstuhl.de/opus/volltexte/2010/2536

6 J. Yuan: The NP-hardness of the single machine common due date weighted tardiness problem. System Science and Mathematical Sciences 5:328–333, 1992.

3.12 The generalization of scheduling with machine cost

Csanád Imreh (University of Szeged, HU)

In resource allocation problems one has to perform some tasks during the shortest possible time or by a minimal amount of resource. The tasks are usually modeled by rectangles where the sides belong to the processing time and the required amount of resources. If we have fixed amount of resources or we have fixed time to perform the tasks then the problem can be modeled by strip packing where the items has to be packed without rotation and without overlapping into a unit width strip with minimal height. Here we investigate the problem where neither the amount of resources nor the time is fixed. Then the rectangles can be packed into an arbitrary container rectangle and the goal is to minimize the sum of γ times the used resources, denoted by H, and the time, denoted by W, thus our objective is $\gamma H + W$, where $\gamma > 0$ is a fixed parameter. If $\gamma = 1$ then the cost equals to the half of the perimeter of the container rectangle. We consider the online version of this resource allocation problem where the rectangles arrive one by one according to a list L, and we have to pack each of them into the container without any information about the further rectangles.

This model can be considered as the extension of online scheduling with machine cost. In scheduling with machine cost the number of the machines is not given as part of the input but the algorithm has to purchase each machine for a fixed unit cost. The objective function is to minimize the sum of the number of machines and the makespan which belong to the sides of an including rectangle of the schedule (thus $\gamma = 1$ holds in this model). The incoming jobs can be assigned to any existing (already bought) machine, it means that $W = m$, the number of the purchased machines, and the width of each incoming rectangle is just 1. An another closely related problem where one of the sides of the container is fixed is the online strip packing problem.

Suppose (as above) that the vertical side of the incoming rectangle corresponds to the resource needed to execute the task, and the horizontal side of the rectangle belongs to the time needed to execute the task. In the real life there exist such situations when one can modify the properties of the task (the sides of the rectangle) to use less resource, but using less resource means that it takes more time to execute the task. Or similarly, it is also possible to decrease the time needed to execute the job, but in this case the execution needs more resource. To handle this kind of relaxation it is supposed that one is allowed to change the rectangle keeping its area fixed.

There is a shelf-based 7.4803-competitive online algorithm in the standard model. For the special case when the rectangles arrive in a list ordered by decreasing height another shelf algorithm exists which is 2.5-competitive. In case of modifiable rectangles (keeping their area fixed) the best known algorithm is $\left(2-\sqrt{2}\right)\sqrt{\frac{5}{4}\sqrt{2}+2} \approx 1.1371$-competitive and it is proved that no online algorithm can have smaller competitive ratio than $\frac{3}{2\sqrt{2}} \approx 1.061$.

There are several further interesting questions. First it would be good to decrease the gaps, mainly by giving a reasonable lower bound in the standard model. A second question is whether allowing rotation can yield better algorithms. And finally it would be interesting to study such models where we have to pay some penalty for changing the size of the items.

Pointers into the literature:

References

1 Gy. Dósa, Y. He, New upper and lower bounds for online scheduling with machine cost, Discrete Optimization, **7(3)**, (2010), 125–135.

2 Gy. Dósa, Cs. Imreh, The generalization of scheduling with machine cost, http://www.inf.u-szeged.hu/~cimreh/machcost.pdf

3 Cs. Imreh, Online strip packing with modifiable boxes, Operations Research Letters, **29**, (2001), 79–86.

3.13 Online Scheduling with General Cost Functions

Sungjin Im (Duke University – Durham, US)

We consider a general online scheduling problem where the goal is to minimize $\sum_j w_j g(F_j)$, where w_j is the weight/importance of job J_j, F_j is the flow time of the job in the schedule, and g is an arbitrary non-decreasing cost function. Numerous natural scheduling objectives are special cases of this general framework:

- Weighted Flow Time: When $g(x) = x$, the objective becomes the total weighted flow time The total stretch is a special case of the total weighted flow time where $w_j = 1/p_j$.
- Weighted Flow Time Squared: If $g(x) = x^2$ then the scheduling objective is the sum of weighted squares of the flows of the jobs.
- Weighted Tardiness with Equal Spans: Assume that there is a deadline d_j for each job J_j that is equal to the release time of j plus a fixed span d. If $g(t) = 0$ for t not greater than the deadline d_j, and $g(t) = w_j(t - d_j)$ for t greater than the deadline $r_j + d$, then the objective is weighted tardiness.
- Weighted Exponential Flow: If $g(x) = a^x$, for some real value $a > 1$, then the scheduling objective is the sum of the exponentials of the flow, which has been suggested as an

appropriate objective for scheduling problems related to air traffic control, and quality control in assembly lines.

We show that the scheduling algorithm Highest Density First (HDF) is $(2+\epsilon)$-speed $O(1)$-competitive for all cost functions g *simultaneously*; see the pointer below. We also show that the HDF algorithm and this analysis are essentially optimal.

This raises a natural question if one can obtain analogous results in the stochastic setting. More concretely, what is the strongest statement one can make about *simultaneous* optimality with standard stochastic assumptions?

Pointers into the literature:

References

1 Sungjin Im, Benjamin Moseley, Kirk Pruhs: Online scheduling with general cost functions. SODA 2012:1254–1265.

3.14 Scheduling of Users with Time-Varying Service Rates

Peter Jacko (Lancaster University, GB)

We discuss the problem of developing a well-performing and implementable scheduler of users with wireless connection to the base station. The main feature of such real-life systems is the time-varying quality of the channel conditions, which turn into the time-varying service rate.

If the service rate is constant, the Smith's $c\mu$-rule is optimal for the single-server case. The variant with time-varying service rates is significantly more difficult, and it is unlikely that an optimal scheduler maintains such a simple structure. For instance, even for single-class users, threshold policies (of giving higher priority to users with higher transmission rate) are not necessarily optimal.

Practically important schedulers, however, are those as simple as possible, since the scheduling decisions are taken at milliseconds scale. Several schedulers have been proposed, but there are no (sub)optimality performance guarantees known at the moment, except for maximal stability or asymptotic (fluid) optimality, which only indicate what to do in the channel condition with highest service rate. Open problems further include adaptation to incomplete information and the multi-server case.

Pointers into the literature:

References

1 Sem Borst: User-level performance of channel-aware scheduling algorithms in wireless data networks. IEEE/ACM Transactions on Networking, 13(3):636–647, 2005.
2 Fabio Cecchi, Peter Jacko: Scheduling of Users with Markovian Time-Varying Service Rates. ACM Sigmetrics 2013.

3.15 Employee scheduling and rescheduling in call centers

Ger Koole (VU – Amsterdam, NL)

In call centers, many parameters are still uncertain the moment employees are scheduled. This leads to the necessity of real-time adjustments to the schedule. This requires different forms of flexibility in the initial schedule. Ideally, when making agent schedules the right amount of flexibility should be introduced. In this talk we discuss the different forms of parameter uncertainty, different ways to do rescheduling, and how this can be incorporated in the initial schedule.

3.16 Online scheduling algorithms analyzed by Dual Fitting

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

I shall talk about a general dual-fitting technique for analyzing online scheduling algorithms in the unrelated machines setting where the objective involves weighted flow-time and we allow the machines of the online algorithm to have slightly extra resources than the offline optimum (the resource augmentation model). In this framework, one can often analyze simple greedy algorithms by considering the dual (or Lagrangian dual) of the linear (or convex) program for the corresponding scheduling problem, and finding a feasible dual solution as the online algorithm proceeds. I shall also mention some recent applications of this technique for deadline scheduling problems.

3.17 Approximation in Dynamic Stochastic Scheduling

Nicole Megow (TU Berlin, DE)

Stochastic scheduling is concerned with scheduling problems in which job processing times are modeled as random variables with known probability distributions. The actual processing times are revealed only upon completion of the jobs. Such problems have been addressed since the 70s, but only more recently approximation results were derived. We give an overview of results and methods for obtaining provably good scheduling policies. This involves linear programming, lower bounding techniques borrowed from online scheduling, and index-based dynamic allocation rules known from multi-armed bandit problems. We discuss open problems, further research directions, and possible connections to other areas.

3.18 MapReduce and Distributed Scheduling

Benjamin J. Moseley (Toyota Technological Institute – Chicago, US)

Recently, the MapReduce parallel computing framework has become the de facto standard for processing large data. The MapReduce distributed framework consist of an elegant combination of sequential computation and network communication that naturally lends itself to efficient distributed data processing. In a MapReduce implementation there is a centralized job tracker that coordinates job scheduling. Designing new scheduling policies has been one of the active research topics in MapReduce because of the need to balance often contradictory needs, e.g., system utilization, fairness, and response times. In this talk, we will first focus on introducing the fundamentals of MapReduce. Then we will discuss several scheduling issues that arise in MapReduce as well as recent developments in the theoretical scheduling community that have addressed these issues.

3.19 Weakly Coupled Stochastic Decision Systems

Kamesh Munagala (Duke University – Durham, US)

Several problems in stochastic optimization and decision theory have the property that they are composed of many independent decision sub-problems coupled together by a few constraints. We present several examples of such problems from diverse application areas such as wireless communication, design of experiments, mechanism design, and budgeted allocations. We present unifying solution techniques based on linear programming and duality, and show connections to well-known heuristics used in practice. The talk is self-contained.

3.20 Stochastic k-TSP

Viswanath Nagarajan (IBM TJ Watson Research Center – Yorktown Heights, US)

We will discuss a stochastic variant of the k-TSP problem: given a set of locations with random rewards, find a path originating from a depot, that minimizes the expected distance to obtain a total reward of k. We present an approximation algorithm, and upper/lower bounds on the "adaptivity gap". The currently known results seem far from best-possible, and it would be interesting to obtain better results even for simple distributions such as Bernoulli. We will also discuss an open question regarding a submodular extension of this problem.

Pointers to some adaptive covering problems:

References

1 Michel X. Goemans and Jan Vondrak, Stochastic Covering and Adaptivity, LATIN 2006, 532–543.

2 Zhen Liu, Srinivasan Parthasarathy, Anand Ranganathan, Hao Yang, Near-optimal algorithms for shared filter evaluation in data stream systems, SIGMOD 2008, 133–146.
3 Sungjin Im, Viswanath Nagarajan and Ruben van der Zwaan, Minimum Latency Submodular Cover, ICALP (1) 2012, 485–497.

3.21 Job Scheduling Mechanisms for Large Computing Clusters

Seffi Naor (Technion – Haifa, IL)

We study mechanisms for online deadline-aware scheduling in large computing clusters. Batch jobs that run on such clusters often require guarantees on their completion time (i.e. deadlines). However, most existing scheduling systems implement fair-share resource allocation between users, an approach that ignores heterogeneity in job requirements and may cause deadlines to be missed. In our framework, jobs arrive dynamically and are characterized by their value and total resource demand (or estimation thereof), along with their reported deadlines. The scheduler's objective is to maximize the aggregate value of jobs completed by their deadlines. We circumvent known lower bounds for this problem by assuming that the input has slack, meaning that any job could be delayed and still finish by its deadline. Under the slackness assumption, we design a preemptive scheduler with a constant-factor worst-case performance guarantee.

3.22 Scheduling in queues with customer grouping

Sindo Nunez Queija (CWI – Amsterdam, NL)

We consider a queueing system where customers of the same type may be grouped in a single service. Delaying service is therefore advantageous to reduce the server load and, as consequence, the load the system can support is unbounded if the the number of different service types is finite. The adversarial effect of delaying service is, naturally, long waiting times of customers. Our performance measure is the mean (or, alternatively, the tail probability) of customer waiting times. Optimizing on performance (avoiding long waiting times) calls for a balanced trade-off of positive and negative effects of customer grouping. We are particularly interested in an extended version of this model in which the waiting times are bounded by deadlines (patience) that either are exactly known to the service system, or their distribution is given beforehand. The motivation for this model comes from content delivery data systems, in which identical content may be requested by many users within a short time interval. Transmitting the content to several users at once corresponds to customer grouping in our model. We will discuss some initial exploration results of this problem.

3.23 Stochastic Comparison of Multicast Pull and Push

Kirk Pruhs (University of Pittsburgh, US)

We consider a client server system where the server communicates to the client by broadcast. So if the server broadcasts a data item (page), then all clients waiting for that page will receive the page. We assume that clients have requests, where each request specifies a time of arrival and a page. Assume that our objective is the average time that a request has to wait to be satisfied. In a *pull* system, clients instantly forward requests to the server. So the server always knows what requests are outstanding. In a *push* system, requests are not forwarded to the server. So the server never knows what requests are outstanding. Let us assume for simplicity that all pages are of unit size (although varying sized pages is also of interest). So the server must decide at each point in time what page to broadcast.

Taking a worst-case view, the push problem is not that interesting, and the pull problem is resolved, there is a scalable algorithm and no better result is possible. Let us turn to a stochastic setting where requests for each data item are independent and Poisson. So the input distributions are completely specified by one arrival rate parameter for each page. In the push setting, is known that computing the optimal expected waiting time is NP-hard, and a polynomial-time approximation scheme is known.

I propose that it would be an interesting question to consider the power of pulling in the stochastic setting. That is, how much better can the average waiting time be if the server knows the requests? One could compare optimal schedules, or show that there is a push algorithm that is competitive with any pull algorithm, or show that there must be a large gap between any push algorithm and a particular pull algorithm.

Pointers into the literature:

References

1 Nicolas Schabanel: The Data Broadcast Problem with Preemption. STACS 2000:181–192.
2 Nikhil Bansal, Ravishankar Krishnaswamy, Viswanath Nagarajan: Better Scalable Algorithms for Broadcast Scheduling. ICALP 2010:324–335

3.24 Scheduling with time-varying cost: Deterministic and stochastic models

Roman Rischke (TU Berlin, DE)

We consider a natural generalization of classical scheduling problems in which using a time slots for processing a job causes some time-dependent cost in addition to the standard scheduling cost. Adding the cost consideration to classical scheduling increases the problem complexity significantly. Nevertheless, we also propose a related two-stage stochastic scheduling model with recourse. Suppose that the exact scheduling instance is not known in the first stage, we only have probabilistic information about scenarios, where a scenario represents a scheduling instance. In the first stage we may reserve time slots at some cheap price. In the second stage a particular scenario becomes known and we may buy additional time slots at a (scenario-dependent) higher price so as to find a feasible schedule for the realized scenario minimizing the total cost of buying and scheduling. Notice that the deterministic problem described above appears as the recourse problem in the second stage in the stochastic model. Although two-stage (multi-stage) stochastic optimization with recourse has received a lot of attention in the theory of approximation algorithms in the past decade, corresponding scheduling problems have hardly been addressed.

In this short presentation we want to advertise the very natural deterministic and stochastic scheduling problems taking into account costs for using/reserving time slots. Both problems are much more complex than classical scheduling problems and require new techniques. We give some first insights and show some relation to scheduling on a machine that may change its speed. We hope to foster further research activity in this hitherto little explored area.

This is joint work in progress with S. Leonardi, N. Megow, L. Stougie, C. Swamy and J. Verschae.

Pointers into the literature:

References

1 L. Epstein, A. Levin, A. Marchetti-Spaccamela, N. Megow, J. Mestre, M. Skutella, and L. Stougie. Universal sequencing on an unreliable machine. SIAM J. Comput., 41(3):565–586, 2012.

2 A. Gupta, M. Pal, R. Ravi, and A. Sinha. Sampling and cost-sharing: Approximation algorithms for stochastic optimization problems. SIAM J. Comput., 40(5):1361–1401, 2011.

3 J. Kulkarni and K. Munagala. Algorithms for cost aware scheduling. In Proc. of WAOA 2012, to appear 2013.

4 N. Megow and J. Verschae. Scheduling on a machine with varying speed: Minimizing cost and energy via dual schedules. CoRR, abs/1211.6216, 2012.

3.25 Online scheduling of jobs with fixed start times on related machines

Jiri Sgall (Charles University – Prague, CZ)

We consider online scheduling of identical jobs with fixed starting times revealed at those times on m uniformly related machines, with the objective of maximizing the number of completed jobs. A newly released job must be either assigned to start running immediately on a machine or otherwise it is dropped. It is also possible to drop an already scheduled job, but only completed jobs contribute their weights to the profit of the algorithm.

In the paper we show that a natural greedy algorithm is 4/3-competitive and optimal on m=2 machines, while for a large m, its competitive ratio is between 1.56 and 2. Furthermore, no algorithm is better than 1.5-competitive. It is an open problem to improve these bounds.

3.26 Scheduling Advertising Campaigns

Hadas Shachnai (Technion – Haifa, IL)

An advertising campaign is a series of advertisement messages that share a single idea and theme which make up an integrated marketing communication. Given a large set of campaigns that can be potentially delivered to a media audience, a service provider attempts to fully deliver a subset of campaigns that maximizes the total revenue, while satisfying constraints on the placement of ads that belong to the same campaign, as well as possible placement constraints among conflicting campaigns. In particular, to increase the number of viewers exposed to an ad campaign, one constraint is that each commercial break contains no more than a single ad from this campaign. Each ad has a given length (=size), which remains the same, regardless of the commercial break in which it is placed. This generic assignment problem defines a family of all-or-nothing variants of the generalized assignment problem (GAP). We design for these variants approximation algorithms with constant-factor worst-case performance guarantees.

3.27 Optimal queue-size scaling in switched networks

Devavrat Shah (MIT – Cambridge, US)

We consider a switched (queueing) network in which there are constraints on which queues may be served simultaneously; such networks have been used to effectively model input-queued switches, wireless networks and more recently data-centers. The scheduling policy for such a network specifies which queues to serve at any point in time, based on the current state or past history of the system. Designing a scheduling policy with optimal average queue-size for switched network has been a question of interest for a while now. As the main result, we shall discuss a new class of online scheduling policies that achieve optimal scaling for average queue-size for a class of switched networks including input-queued switches.

Talk is based on work with Neil Walton (U of Amsterdam)+ Yuan Zhong (UC Berkeley).

3.28 On Carry Over

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

Any schedule for a round robin tournament involves an order in which each team meets its opponents. We say that team i gives a *carry-over effect* to team j, if some other team t's match against i is followed by a match against team j. This is particularly relevant in physical, body-contact sports. For instance, if team i is a very strong, tough-playing side, one can imagine that its opponent, team t, is weakened by injuries or fatigue, which could be an advantage for its next opponent, team j. Moreover, the carry-over effect can also be relevant in a strictly psychological interpretation, when team t loses confidence and morale after a severe loss against the strong team i, again to the benefit of their next opponent, team j. The opposite may be true if team i is a weak team. Clearly, carry-over effects are unavoidable in any schedule, however, schedules can differ in the extent to which carry-over effects are balanced over the teams. We define c_{ij} as the number of times that team i gives a carry-over effect to team j in a schedule. The degree to which the carry-over effects are balanced is typically measured by the so-called carry-over effects value, which is defined as $\sum c_{i,j}^2$ (Russell 1980).

The table below gives the best known values for the carry-over effect, depending upon the number of teams. An asterisk denotes that the given value is minimum, i.e., no schedule exists achieving a value lower than the reported number.

Number of teams	Carry-over value
4	12* (Russell 1980)
6	60* (Russell 1980)
8	56* (Russell 1980)
10	108* (Anderson 1999, Eggermont 2011)
12	170 (Eggermont 2011)
14	234 (Anderson 1999)
16	240* (Russell 1980)
18	340 (Anderson 1999)
20	380* (Anderson 1999)
22	462* (Anderson 1999)
24	644 (Anderson 1999)

Pointers to the literature:

References

1 Anderson, I. (1999), Balancing carry-over effects in tournaments, in: *Combinatorial designs and their applications*, Chapman and Hall/CRC, page 1–16.

2 Eggermont, C. (2011), reachability Problems in Scheduling and Planning, PhD thesis Eindhoven University of Technology.

3 Russel, K. (1980), Balancing carry-over effects in round robin tournaments, Biometrika 67, 127–131.

3.29 Stochastic Optimal Control for a Class of Dynamic Resource Allocation Problems

Mark S. Squillante (IBM TJ Watson Research Center – Yorktown Heights, US)

We consider a class of general dynamic resource allocation problems within a stochastic optimal control theoretic framework. This class of problems arises in a wide variety of applications, each of which intrinsically involves resources of different types and demand with uncertainty and/or variability. The goal is to determine the allocation capacity for every resource type in order to serve the uncertain/variable demand and maximize the expected profit (utility) over a time horizon of interest based on the rewards, costs and flexibility associated with the different resources. We derive the optimal (online) control policy within a singular stochastic optimal control setting, which includes simple expressions for governing the dynamic adjustments to resource allocation capacities over time. Numerical experiments investigate various issues of both theoretical and practical interest, quantifying the benefits of our approach over alternative optimization approaches.

This talk is based on joint work with Xuefeng Gao, Yingdong Lu, Mayank Sharma and Joost Bosman.

3.30 Truthful Scheduling

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

A major question in algorithmic game theory is whether the presence of selfish agents affects the approximability of various classic optimization problems. Specifically, the following research agenda was suggested: *"to what extent is incentive compatible efficient computation fundamentally less powerful than 'classic' efficient computation?"*. Of particular interest are scheduling problems, where jobs are assigned for processing to agents, each controlling one machine, and who have some private information regarding their machines. In this paper, we consider the case of single-parameter agents with scheduling problems on uniformly related machines, which was among the first problems considered in the area of algorithmic mechanism design. The private information of an agent is the cost of processing one unit of work, which is also the inverse of the speed of the machine. We show that selfish agents do not affect the approximability of scheduling problems on uniformly related machines, by designing $(1+\varepsilon)$-approximation mechanisms for these problems for any $\varepsilon > 0$. Given that these problems are (strongly) NP-hard, this is the best possible result.

Non-preemptive scheduling problems on m uniformly related machines are defined as follows. We let the set of machines be denoted by $M = \{1, 2, \ldots, m\}$. We are given a set of jobs $J = \{1, 2, \ldots, n\}$, where each job j has a positive size p_j. The jobs need to be partitioned into m subsets $S_1, \ldots, S_m$, with S_i being the subset of jobs assigned to machine i. We let s_i denote the (actual) speed of machine i, meaning that the processing of job j takes $\frac{p_j}{s_i}$ time units if j is assigned to machine i. For such a solution (also known as a schedule), we let $L_i = (\sum_{j \in S_i} p_j)/s_i$ be the *completion time* or *load* of machine i. The *work* of machine i is $W_i = \sum_{j \in S_i} p_j = L_i \cdot s_i$, that is, the total size of the jobs which are assigned to i. We consider objective functions which are functions of the machine loads, $L_1, L_2, \ldots, L_m$. We consider a variety of objective functions (social goals), like minimizing the makespan (maximum load) and maximizing the minimum load (cover).

The setup of mechanism design for single-parameter agents operating uniformly related machines is as follows. Agents present bids to a mechanism, where the bid b_i of an agent i is the claimed cost per unit of work of its machine (the inverse of its claimed speed). Based on these bids, the mechanism allocates the jobs to the machines and also assigns payments to the agents. We assume that each agent is only interested in maximizing its own profit, which is its payment minus its (actual) cost of processing the jobs allocated to it. A mechanism is called *truthful* if reporting their true costs per unit of work is a dominant strategy for the agents. That is, this strategy maximizes the profit for each agent, regardless of the strategies of the other agents. In the case of single-parameter agents, a well-known necessary and sufficient condition for truthfulness is that the allocation algorithm is *monotone*, that is, the allocation algorithm must have the property that if an agent i increases its claimed speed (i.e., decreases its bid) while all other bids are unchanged, the work allocated to i does not decrease. More precisely, in such a case there exist simple payment functions that can be coupled with the (monotone) allocation algorithm to give a truthful mechanism. If the allocation algorithm runs in polynomial time, and the payments can be computed in polynomial time as well, then the resulting truthful mechanism can be implemented in polynomial time.

Pointers into the literature:

References

1 Aaron Archer. *Mechanisms for Discrete Optimization with Rational Agents.* PhD thesis, Cornell University, 2004.

2 L. Epstein and J. Sgall. Approximation schemes for scheduling on uniformly related and identical parallel machines. *Algorithmica*, 39(1):43–57, 2004.

3.31 Online Stochastic Matching

Clifford Stein (Columbia University, US)

Online matching is an important and well-studied problem and has received particular attention recently because of its connections to allocation problems in internet advertising. In the traditional worst-case model, on-line algorithms are well understood, with tight competitive ratios of 1/2 for deterministic algorithms and (e-1)/e for randomized algorithms.

Several recent works have considered relaxing the worst-case model and considering various probabilistic models for on-line matching. We will survey some of the models and results, and also describe an online algorithm in a model in which an adversary chooses the graph, but does not control the order in which the online nodes are revealed (This work is joint with Feldman, Korula, Henzinger, and Mirrokni). We will also show how these ideas generalize to online packing problems

3.32 Split scheduling with uniform setup times

Suzanne van der Ster (VU – Amsterdam, NL)

We consider the problem of scheduling a set of n jobs on m parallel machines with *job splitting* and *setup times.* This means that each job may be split into parts and multiple parts of the same job may be processed simultaneously. However, before a machine can start processing (a part of) a job, a fixed setup time s is required. (Note that the setup time is job-, machine-, and sequence-independent.) Job j has processing time p_j, for $j = 1, \dots, n$ and the objective is to minimise the sum of completion times of the jobs.

Given a schedule for the jobs, we define by M_j the set of all machines on which job j is scheduled. A job is called a d-job when is is scheduled on d machines. We call a job balanced in a schedule if it completes at the same time on all machines on which it is processed. We assume that jobs are numbered in non-decreasing order of their processing times. This order is also called the Shortest Processing Time first (SPT) order.

There is a polynomial-time algorithm solving the problem for the case of 2 machines. The properties below give us a handle to find an optimal schedule.

(a) There exists an optimal schedule such that on each machine the job parts are processed (started and completed) in SPT order of the corresponding jobs.
(b) There exists an optimal schedule in which all jobs are balanced.
(c) There are no 1-jobs after a 2-job in an optimal schedule satisfying the two properties above.

From these properties we derive that all 2-jobs are scheduled in SPT order at the end and the first 2-job is not shorter than the preceding 1-jobs. This implies that the 1-jobs can be scheduled in SPT order without increasing the completion time of the 2-jobs. We can simply consider each of the n jobs as the first 2-job, or perform a more careful update of consecutive solutions to improve on the running time.

In future research we want to study the problem for more than 2 machines. Already for 3 machines this is anything but straightforward. Properties (a) and (b) hold for any number of machines, but property (c) (or a generalization of it) does not. We have examples where in none of the optimal solutions $|M_j|$ is monotone in j. A few of the questions that could be considered:

- Does there exist an optimal schedule such that the jobs are started (or finished) in SPT order? (This is an option that is not ruled out by the counterexamples showing that a generalization of (c) does not hold.)
- What is the complexity of the problem for more than 2 machines? Or even, what is the complexity for only 3 machines?

The results given above, as well as a more extensive discussion of the problems we encounter already for three machines, can be found here.

References

1 F. Schalekamp, R. Sitters, S. van der Ster, L. Stougie, V. Verdugo, and A. van Zuylen. Split scheduling with uniform setup times. http://arxiv.org/abs/1212.1754

3.33 An infinite server system with customer-to-server packing constraints

Alexander Stolyar (Bell Labs – Murray Hill, US)

The model is motivated by the problem of efficient "packing" of virtual machines into physical host machines in a network cloud data center. There is an infinite number of servers and multiple flows of arriving customers of different types. Each server can simultaneously serve several customers, subject to some "packing" constraints. Service times of different customers are independent – even if customers share a server. Customers leave after their service is complete. The underlying objective is to minimize the number of occupied servers. We show that some versions of a greedy packing strategy are asymptotically optimal as the system scale (the average total number of customers in service) goes to infinity.

3.34 Analysis of Smith's Rule in Stochastic Scheduling

Marc Uetz (University of Twente, NL)

In a landmark paper from 1986, Kawaguchi and Kyan show that scheduling jobs according to ratios weight over processing time – a.k.a. Smith's rule – has a tight performance guarantee of 1.207 for minimizing the total weighted completion time in parallel machine scheduling. The talk addresses the performance of Smith's rule for the variant where processing times are exponentially distributed random variables. The current status of the problem is as follows. The expected performance of Smith's rule is no worse than $(2 - 1/m)$ times optimum, m being the number of machines Möhring, Schulz and Uetz, JACM 46, 1999, 924-942. This analysis is based on a LP relaxation, and hold for any random variable with coefficient of variation at most 1. On the other hand, the analysis of a (slightly modified) stochastic version of the Kawaguchi and Kyan instance yields an expected performance at least 1.243 times optimum Jagtenberg, Schwiegelshohn and Uetz, WAOA 2010. The open problem is to find the true performance bound for Smith's rule in stochastic parallel machine scheduling.

3.35 Appointment Scheduling with Slot Blocking

Peter van de Ven (IBM TJ Watson Research Center – Hawthorne, US)

Consider an electronic appointment booking system that allows patients to book their doctor appointment online; the system displays the available time slots and the patient selects an acceptable slot or leaves if none of the available slots are acceptable to him. Patients typically have some set of acceptable slots, and surveys shows that certain slots are popular (deemed acceptable by more patients) than others. The popular slots tend to be booked early in the reservation process, leaving only less desirable slots open. As a result, patients that arrive at a later stage may not find any acceptable slot upon arrival, and at the end of the reservation proces several unpopular slots may be left unoccupied, reducing the utilization of this doctor.

We propose to block the popular slots during initial stages of the reservation process, forcing early patients to select less popular slots. Making available the popular slots to later arrivals then allows for more flexibility and may result in higher utilization. We study the tradeoff between sending away early patients (blocking too long) and having little flexibility later on (blocking too little). This model has an interesting connection with the online bipartite matching problem.

3.36 Learning in Stochastic Scheduling

Tjark Vredeveld (Maastricht University, NL)

We consider a scheduling problem in which K classes of independent jobs have to be processed non-preemptively. The processing times of the jobs are assumed to be exponentially distributed with parameters depending on the class of each job. However, the precise values of the parameters are unknown to the scheduler. The objective is to minimize the sum of expected completion times.

The scheduler has certain beliefs on the values of the parameters for the different classes and the belief on a parameter of a certain class is updated after the completion of a job of this class.

As the Shortest Expected Processing Time (SEPT) rule is optimal for the single machine problem without parameter uncertainty and it is an simple rule, we are interested in how variants of this rule performance in the presence of parameter uncertainty. It has been shown that on a single machine, two variants of SEPT have a performance guarantee of 2, i.e., SEPT obtains a schedule with expected value at most two times the optimal value, and that this bound is tight whenever there are only two different classes of jobs.

An interesting question is what the performance guarantees of several variants of SEPT are when there is more than one machine and more than 2 job classes.

Based on joint work with S. Marbán and C. Rutten.

Pointers into the literature:

References

1 S. Marbán, C. Rutten, T. Vredeveld. Learning in stochastic machine scheduling. WAOA 2011:21-34
2 S. Marbán (2012). Pricing and Scheduling under Uncertainty. PhD Thesis, Maastricht University. (Chapters 4, 5)

3.37 FCFS infinite matching, queues with skill based routing, and organ transplants

Gideon Weiss (Haifa University, IL)

This talk is a survey of recent work with Ivo Adan and several other collaborators, including Cor Hurkens, Marko Boon, Ana Busic, and Jean Mairesse. It is based on earlier work of Rishi Talreja and Ward Whitt, and of Rene Caldentey and Ed Kaplan. In recent years it has become very important to investigate service systems which serve cus- tomers of several types, and which employ servers of various skills. Such service systems are referred to in current literature as queues with skill based routing. These have several types of customers, and several types of servers, and a bipartite compatibility graph to indicate which types of servers can serve which types of customers. Applications include such varied fields as call centers, outsourcing, manufacturing process, cloud computing and health systems. A somewhat different model is the matching of applicants and positions, of organ donors and patients, of adoptive parents and children, the so called marriage model. These two types of service models motivate our research, we note that these applications are also studied by schedulers and combinatorial optimizers. There is a significant difference between these two models: While in a queueing model the sequence of interarrival times and the sequence of service times get hopelessly entangled through the busy/idle cycles of the servers, this is absent from the matching applications: Here customers and servers play a symmetric role and each customer server encounter has no after effects on the rest of the system. The taxi rank is a simple example of that. A further simplification is to consider just the sequence of customers and of servers, ordered by arrivals and classified by types: We think of those as i.i.d infinite sequences of customer types (all the customers that will appear in the future) and of server types (all the services that will be given). For these sequences it is natural to define a FCFS matching: The first server is matched to the first compatible customer, and the nth server is matched to the first compatible customer that none of the previous n ? 1 servers picked up. This infinite FCFS model has a much more combinatorial flavor than the standard queueing models. It turns out that it is quite tractable. The key property which we discovered is that this infinite matching model is in some sense time reversible. Reversibility plays a key role in the theory of Markov chains and of queueing models. It is the property which underlies product form results. It is a sad fact that most queueing models are complicated and often intractable. Research on queueing networks would have gone nowhere if it weren't for Jackson's discovery that Jackson networks have steady state distribution. From that time onwards, product form results have been keenly sought after, as they seem the best way of getting explicit solutions and useful insight for more general models. Product form is always related to some form of reversibility. The reversibility of the infinite matching model underlies all our further results. Coming back to queues with skilled based routing, we focus on FCFS-ALIS, first come first served, assign longest idle server policy. This means that whenever a server becomes available he will go to the longest waiting customer in the system which he can serve, and whenever a customer arrives to find several idle servers he will be assigned to the longest idle compatible server. This policy has several attractive features: First and foremost it is fair to both customers and servers, in many systems e.g. organ donations, public housing assignment, FCFS is dictated by law. ALIS is the best way to equalize the efforts of the servers, and thus it encourages diligent service. The policy is also very natural and easy to implement, and it requires minimal information about the parameters of the system and its current state. As a result it is useful in systems in which load and staffing keep changing over the operating horizon. Our exact results are that under a FCFS-ALIS policy when arrivals are Poisson and services exponential we have product form queue length distributions, and closed form expressions for waiting times. Furthermore, these results can be used to obtain excellent approximations for much more general queueing models, under many server scaling. These include general link dependent service times, abandonments, and efficiency driven systems. We will present some examples of how to use these results in the design of call centers, and in the planning of an organ transplants policy.

4 Working Groups

Our experience is that it takes at least a year before one can start to assess the impact of a seminar with any confidence. But some initial collaborations arising from the seminar are:

- Based on collaborations in preparing for the seminar, Bert Zwart (from the stochastic community) and Nikhil Bansal (from the worst-case community) wrote a successful grant proposal for funding of one joint Ph.D. student.

- Nicole Megow and Leen Stougie (from the worst-case community) began a research collaboration on 2-stage stochastic optimization problems.
- Frits Spieksma (from the worst-case community) and Onno Boxma (from the stochastic community) have started a research collaboration on scheduling shipping locks.
- Martin Skutella, Maxim Sviridenko, and Marc Uetz (from the worst-case community) began a research collaboration on stochastic scheduling of unrelated machines, resulting in a paper, http://eprints.eemcs.utwente.nl/23246/
- Ger Koole and Rhonda Righter started a research collaboration on assessing various performance measures for call centers with abandonment.
- Urtzi Ayesta and Rhonda Righter started a research collaboration on scheduling of bandit processes in a random environment.
- Samir Khuller, and Christoph Dürr each began research with one of their Ph.D. students on one of the open problems presented in the short talks.

5 Open Problems

Open problems are included in the talk abstracts above.

Participants

- Samuli Aalto
Aalto University, FI
- Susanne Albers
HU Berlin, DE
- Konstantin Avrachenkov
INRIA Sophia Antipolis – Méditerranée, FR
- Urtzi Ayesta
LAAS – Toulouse, FR
- Yossi Azar
Tel Aviv University, IL
- Nikhil Bansal
TU Eindhoven, NL
- Vincenzo Bonifaci
National Research Council – Rome, IT
- Sem C. Borst
TU Eindhoven, NL
- Onno J. Boxma
TU Eindhoven, NL
- Marek Chrobak
University of California – Riverside, US
- Christoph Dürr
UPMC, Lab. LIP6 – Paris, FR
- Fritz Eisenbrand
EPFL – Lausanne, CH
- Thomas Erlebach
University of Leicester, GB
- Naveen Garg
Indian Inst. of Technology – New Dehli, IN
- Anupam Gupta
Carnegie Mellon University – Pittsburgh, US
- John Hasenbein
University of Texas – Austin, US
- Wiebke Höhn
TU Berlin, DE
- Sungjin Im
Duke University – Durham, US
- Csanad Imreh
University of Szeged, HU
- Peter Jacko
Lancaster University, GB
- Samir Khuller
University of Maryland, US
- Amit Kumar
Indian Inst. of Technology – New Dehli, IN
- Fei Li
George Mason University – Fairfax, US
- Alberto Marchetti-Spaccamela
University of Rome "La Sapienza", IT
- Monaldo Mastrolilli
IDSIA – Lugano, CH
- Nicole Megow
TU Berlin, DE
- Rolf H. Möhring
TU Berlin, DE
- Benjamin J. Moseley
Toyota Technological Institute – Chicago, US
- Kamesh Munagala
Duke University – Durham, US
- Viswanath Nagarajan
IBM TJ Watson Res. Center – Yorktown Heights, US
- Seffi Naor
Technion – Haifa, IL
- Sindo Nunez Queija
CWI – Amsterdam, NL
- Kirk Pruhs
University of Pittsburgh, US
- Rhonda Righter
University of California – Berkeley, US
- Roman Rischke
TU Berlin, DE
- Jiri Sgall
Charles University – Prague, CZ
- Hadas Shachnai
Technion – Haifa, IL
- Devavrat Shah
MIT – Cambridge, US
- René Sitters
VU – Amsterdam, NL
- Martin Skutella
TU Berlin, DE
- Frits C. R. Spieksma
K.U. Leuven, BE
- Mark S. Squillante
IBM TJ Watson Res. Center – Yorktown Heights, US
- Clifford Stein
Columbia University, US
- Sebastian Stiller
TU Berlin, DE
- Alexander Stolyar
Bell Labs – Murray Hill, US
- Leen Stougie
CWI – Amsterdam, NL
- Maxim I. Sviridenko
University of Warwick, GB
- Marc Uetz
University of Twente, NL
- Peter van de Ven
IBM TJ Watson Res. Center – Hawthorne, US
- Suzanne van der Ster
VU – Amsterdam, NL
- Rob van Stee
MPI für Informatik – Saarbrücken, DE
- Tjark Vredeveld
Maastricht University, NL
- Gideon Weiss
Haifa University, IL
- Gerhard J. Woeginger
TU Eindhoven, NL
- Prudence W. H. Wong
University of Liverpool, GB