Jacek Blazewicz, Klaus H. Ecker,
E. G. Coffman Jr., G. Finke (editors)

Scheduling in Computer and Manufacturing Systems

Dagstuhl Seminar Report (99431)

24.10.1999 - 29.10.1999

Overview

During the week of October 24 - 29, 1999, the Seminar on Scheduling in Computer and Manufacturing Systems was organized by J. Blazewicz (Poznan), E. G. Coffman, Jr. (Bell Labs, Murray Hill), K. Ecker (Clausthal) and G. Finke (Grenoble). Participants came from universities or research centers from Austria, Belarus, Belgium, Canada, France, Germany, Great Britain, Italy, Mexico, Poland, Switzerland, The Netherlands, U.S.A.

The objective of the seminar was to provide a forum for the discussion of current and ongoing research in scheduling. The seminar promoted an exchange of ideas covering the entire spectrum from case studies of real applications to recent advances in mathematical foundations. These various aspects of the scheduling area have been covered by 38 lectures which addressed classical application areas such as distributed processing, operating systems, dependable systems, flexible manufacturing, and others. It is worth pointing out that many lectures have been motivated by practical considerations, as for example machine break downs, batch scheduling, synchronous production, robotic cell scheduling, real-time scheduling, resource investment problem and others. But also exciting new areas have emerged such as those in modern communications, examples being wireless networks, multimedia networks, and the internet.

The seminar proceeded along three broad fronts:

- *Applications*, which include empirical studies of existing systems as well as numerical studies of the analysis and simulation of system models. Most of the studied applications came from the area of production scheduling and planning, such as just in time scheduling, due date assignment and project control, including special problems dealing with machine breakdowns, robotic cells, assembly scheduling, load balancing, minimizing the number of workers (human resources). Other presentations considered special problems from chemistry and oceanography, the design of schedulers e.g. for web applications, and planning examination sessions.

- *Algorithms* for various problems such as batch scheduling, resource scheduling, tardiness problems, shop problems, deadline and due date scheduling, real-time scheduling, on-line scheduling, single machine problems, time lags, scheduling with communication delays, and/or scheduling. The main concern in these presentations was the design and analysis of algorithms ranging from simple and tractable on-line and greedy rules to methods based on semi-enumerative approaches, branch and bound, local neighborhood search, and LP formulations.

- *Theory*, which includes recent results in the analysis of new and classical problems under novel (or multiple) criteria, dealing with particular assumptions on machines, tasks (e.g., release dates, precedence constraints, communication delays, multiprocessor tasks, bi-processor tasks), and other problems such as assembly scheduling problems and on-line scheduling. Typical questions discussed were the structure of problems and their relation to graph theory, complexity of problems including polynomial solvability, the design of algorithms and performance analysis, and the approximability of optimal solutions.

The participants were overwhelmed by the outstanding local organization and the marvelous facilities which created the atmosphere for a successful seminar.

On behalf of the participants,

J. Blazewicz, E. G. Coffman, K. Ecker, G. Finke

List of Participants

Evripidis Bampis	bampis@lami.univ-evry.fr
Philippe Baptiste	baptiste@utc.fr
Michael Bender	bender@cs.sunysb.edu
Jacek Blazewicz	blazewic@sol.put.poznan.pl
Nadia Brauner	nadia.brauner@ulg.ac.be
Peter Brucker	peter@mathematik.Uni-Osnabrueck.de
Edmund Burke	ekb@cs.nott.ac.uk
Jacques Carlier	Jacques.Carlier@utc.fr
Ed Coffman Jr	egc@bell-labs.com
Ulrich Dorndorf	udorndorf@acm.org
Frank Drews	frank.drews@tu-clausthal.de
Andreas Drexl	drexl@bwl.uni-kiel.de
Maciej Drozdowski	maciej_d@sol.put.poznan.pl
Klaus Ecker	ecker@informatik.tu-clausthal.de
Valery Gordon	gordon@newman.bas-net.by
Frédéric Guinand	frederic.guinand@univ-lehavre.fr
Leslie A. Hall	lah@jhu.edu
Han Hoogeven	slam@cs.uu.nl
Adam Janiak	janiak@ict.pwr.wroc.pl
Klaus Jansen	kj@informatik.uni-kiel.de
Joanna Jozefowska	JJozef@sol.put.poznan.pl
Sigrid Knust	sigrid@mathematik.Uni-Osnabrueck.de
Mikhail Kovalyov	koval@newman.Bas-net.by
Marek Kubale	kubale@pg.gda.pl
Wieslaw Kubiak	wkubiak@morgan.ucs.mun.ca
Jan Karel Lenstra	jkl@win.tue.nl
Ernst W. Mayr	mayr@in.tum.de
Rolf H. Möhring	moehring@math.tu-berlin.de
Jurek Nawrocki	nawrocki@put.poznan.pl
Klaus Neumann	neumann@wior.uni-karlsruhe.de
Bernard Penz	bernard.penz@gilco.inpg.fr
Erwin Pesch	e.pesch@uni-bonn.de
Cindy Phillips	caphill@mp.sandia.gov
Chris N. Potts	cnp@maths.soton.ac.uk
Kirk Pruhs	kirk@cs.pitt.edu
Christophe Rapine	christophe.rapine@imag.fr
Günter Schmidt	gs@itm.uni-sb.de
Jörn Schneider	js@dagstuhl.de
V. Sridharan	suhas@clemson.edu
Clifford Stein	cliff@cs.dartmouth.edu
Frederik Stork	stork@math.tu-berlin.de
Eugeniusz Toczylowski	e.toczylowski@ia.pw.edu.pl
Denis Trystram	denis.trystram@imag.fr
Marc Uetz	uetz@math.tu-berlin.de
Steef van de Velde	s.velde@fac.fbk.eur.nl
Jan Weglarz	weglarz@put.poznan.pl
Frank Werner	frank.werner@mathematik.uni-magdeburg.de
Marino Widmer	marino.widmer@unifr.ch
Gerhard Woeginger	gwoegi@opt.math.tu-graz.ac.at
Jürgen Zimmermann	zimmermann@wior.uni-karlsruhe.de

Abstracts

Scheduling with Machine Breakdowns

Susanne Albers and Guenter Schmidt

In scheduling theory the basic model assumes that a fixed set of machines is continuously available for processing throughout the planning horizon. This assumption might be justified in some cases but it does not apply if certain maintenance requirements, breakdowns or other constraints that cause the machines not to be available for processing have to be considered. Knowledge about machine availabilities might be complete or incomplete. In an online setting machine availabilities are not known in advance. Machine breakdowns are a typical example of events that arise online. Sometimes a scheduler has partial knowledge of the availabilities, i.e, he has some lookahead He might know of the next time interval where a machine requires maintenance or he might know when a broken machine will be available again. In an offline setting all machine availabilities are known prior to schedule generation.

We study a very basic scheduling problem with respect to limited machine availability: A set of jobs has to be scheduled on a set of identical machines so as to minimize the makespan. More specifically, let $J = \{J_i | i = 1, ..., n\}$ be a set of independent jobs to be scheduled. Job J_i has a processing time of p_i time units known in advance, $1 \leq i \leq n$. The jobs have to be scheduled on a set of machines that operate with the same speed. Preemptions of jobs are allowed. Each machine may work only on one job at a time, and each job may be processed by only one machine at a time. We wish to minimize the makespan i.e., the completion time of the last job that finishes. Machines may have different time intervals of availability. We emphasize here that we are interested in the online version of the problem where the machine availabilities are not known in advance. We also call an interval where a machine is not available a machine breakdown. Machines may break down or recover at arbitrary time instances. New machines may be added as well. We also consider the online problem with lookahead one i.e., a scheduler always knows the next point in time where the set of available machines changes. However, he does not have to know which machines break down or become available. In the previous literature, this setting is also referred to as nearly online.

We show that no online algorithm can construct optimal schedules and that no online algorithm can achieve a constant competitive ratio if there may be time intervals where no machine is available. We present an online algorithm that constructs schedules with an optimal makespan of $C_{\max}^{OPT}$ if a lookahead of one is given, i.e., the algorithm always knows the next point in time when the set of available machines changes. Finally we give an online algorithm without lookahead that constructs schedules with a nearly optimal makespan of $C_{\max}^{OPT} + \epsilon$, for any $\epsilon > 0$, if at any time at least one machine is available. The results demonstrate that not knowing machine availabilities in advance is of little harm.

A PTAS for the Average Weighted Completion Time Scheduling Problem on Unrelated Machines

Evripidis Bampis

In this talk, we presented an $O(n \log n)$ PTAS scheme for the problem of scheduling a set of independent jobs on a constant number of unrelated machines in order to minimize the weighted sum of completion times. We also discussed some related results in the presence of release dates.

Batching Identical Jobs

Philippe Baptiste

We study the problems of scheduling jobs, with different release dates and equal processing times, on two types of batching machines. All jobs of the same batch start and are completed simultaneously. On a serial batching machine, the length of a batch equals the sum of the processing times of its jobs and, when a new batch starts, a constant setup time s occurs. On a parallel batching machine, there is at most b jobs per batch and the length of a batch is the largest processing time of its jobs. We show that in both environments, for a large class of so called "ordered" objective functions, the problems are polynomially solvable by dynamic programming. This allows us to derive that the problems where the objective is to minimize the weighted number of late jobs, or the weighted flow time, or the total tardiness, or the maximal tardiness are polynomial. In other words, we show that $1|p\text{-}batch, b < n, r_i, p_i = p|F$ and $1|s\text{-}batch, r_i, p_i = p|F$, are polynomial for $F \in \{\sum w_i U_i, \sum w_i C_i, \sum T_i, T_{\max}\}$. The complexity status of these problems were unknown before.

New Algorithms and Metrics for Scheduling Job Streams

Michael A. Bender

There are several important objectives when designing a scheduler for a webserver, disk, or routing switch: maximizing throughput, minimizing average response times, and minimizing worst-case response times. We study the relationship between these objectives for the problem of scheduling a stream of jobs on a single processor. We introduce a new metric, maxstretch, that naturally balances average and worst-case response times. Then we describe algorithms to optimize this metric and show simulation results.

Synchronous Production, Synchronous Delivery: New Dynamic Scheduling Problems

Lyes Benyoucef, Yannick Frein and Bernard Penz

In order to reduce the time between the demand of the customers and the delivery of components, companies introduce in their organizations the *Just-In-Time* philosophy. To achieve this goal a new approach was proposed recently in car industries to supply bulky components of great diversity and very expensive to manage. This supply mode is called synchronous delivery. A special case is called synchronous production.

The principle of the synchronous mode is that the components are delivered in the order in which they will be got in the assembly line. This requires of course that the supplier knows in advance the sequence of orders. Afterwards the supplier has to produce the components and deliver it without any delay, following the order sequence. The objective of the supplier is to schedule his production respecting his constraints (load balancing for example), in order to minimize a criterion. This criterion can be the minimization of the number of production changeovers, the minimization of production and storage costs, etc.

Then we present a new dynamic scheduling problem which occurs in this context. A *Markov Decision Process* modeling is proposed to solve optimally the problem. Unfortunately this method is not polynomial in time and then, heuristic policies are developed and tested.

Links Between Task Precedence Graphs Revisited
JACEK BLAZEWICZ AND DANIEL KOBLER

Precedence constraints are a part of a definition of any scheduling problem. In this paper, we investigate relations between two graph representations of these constraints: task-on-arc and task-on-node, respectively. It is proved that the latter representation is more powerful. On the other hand, careful analysis reveals new classes of task-on-node graphs for which corresponding problems of scheduling preemptable tasks, are polynomially solvable.

One-cycle Complexity in Robotic Cells
NADIA BRAUNER, GERD FINKE AND WIESLAW KUBIAK

We consider a robotic cell consisting of a flow-shop in which the machines are served by a single robot. We concentrate on the case with only one part type produced. We study the complexity of finding the shortest one-cycle (i.e. cycle producing a single part each time). We call this problem the Robotic Cell Scheduling (RCS) problem.

We show that the RCS problem is NP-hard in the strong sense for symmetric distances with triangle inequality. For the equidistant case with the same processing time on each machine we give an optimization algorithm running in $O(1)$ time. The RCS problem is polynomial for additive distances as shown by Crama and van de Klundert.

Solving a Chemical Batch Scheduling Problem by Local Search
PETER BRUCKER

A chemical batch scheduling problem is modeled in two different ways as a discrete optimization problem. In the first model the batches of a processing task are represented as a chain of operations and one has a general shop problem with setups. In the second model the processing tasks are represented by jobs with generalized precedences between these jobs. Both models are used to solve the batch scheduling problem in a two-phase tabu search procedure by first finding a good solution for the second model and then improving this solution by using the other model. The method is tested on real world data.

Redundant Resources for the Cumulative Scheduling Problem

Jacques Carlier and Emmanuel Neron

This talk deals with the generation of redundant resources for the Cumulative Scheduling Problem (CusP). In a CuSP, we have to schedule n activities in a minimal makespan. Each activity i has a release date r_i, a processing time p_i and a tail q_i. It needs b_i units of a resource whose availability is equal to B. Several CuSPs can be obtained from the Resource Constrained Project Scheduling Problem (RCPSP) by relaxing all the resources except one. We associate with a CuSP a Multiple Elastic Preemptive Bound (MEPB) obtained by allowing more than one part of an activity to be executed simultaneously. MEPB is computed thanks to a parametric linear program. It is equal to the maximum of the optimal solutions of its dual. Next we associate a redundant resource with each optimal solution of the dual. It appears that the number of redundant resources is not so large. For instance for $B = 7$, it is equal to 6, for $B = 10$, to 20. So they can be practically used. Moreover we get all of them. Consequently we have efficient lower bounds and new adjustments of heads and tails. The interest of this approach is confirmed by the computational results on the RCPSP which are reported.

An Iterated Dynasearch Algorithm for the Single-Machine Total Weighted Tardiness Scheduling Problem

Richard K. Congram, Chris N. Potts and Steef L. van de Velde

This paper introduces a new neighborhood search technique, called dynasearch, that uses dynamic programming to search an exponential size neighborhood in polynomial time. While traditional local search algorithms make a single move at each iteration, dynasearch allows a series of moves to be performed. The aim is for the lookahead capabilities of dynasearch to prevent the search from being attracted to poor local optima. We evaluate dynasearch by applying it to the problem of scheduling jobs on a single machine to minimize the total weighted tardiness of the jobs. Dynasearch is more effective than traditional first-improve or best-improve descent in our computational tests. Furthermore, this superiority is much greater for starting solutions close to previous local minima. Computational results also show that an iterated dynasearch algorithm in which descents are performed a few random moves away from previous local minima is superior to other known local search procedures for the total weighted tardiness problem.

Periodic Loading Problem and Its Application

Adam Czajka, Jerzy Nawrocki and Piotr Zielinski

Real-time systems consist of periodic tasks, which have to be repeated with a given period. Some tasks must always be completed on time, i.e. before the end of the period. They are called hard real-time tasks and they have to be scheduled accordingly.

The problem considered here is how to find a statical (pre-run-time) schedule for a set of non-preemptable periodic tasks that have to be executed on one computer.

Each task is characterized with its duration d and period p. k-th instance of a task of period p has to be started and completed within the interval $[(k-1)p; kp]$. It is assumed here that periods form a geometrical progression with the quotient equal to 2, i.e. period of task j is equal to the period of task $j-1$ or it is twice as long as the period of task $j-1$. Binary periods have been applied in the MAFT architecture and in an implementation of the MIL-STD-1553. During the presentation we discussed the periodic loading approach to scheduling periodic tasks. The main idea is to split time into a sequence of frames. Each frame has the same size which equals the duration of the shortest period. Each instance of a task must be assigned to one of the frames - it is not allowed to place a task on the border between two frames.

We have presented worst-case analysis of two known heuristics: non-increasing duration (NID) and non-decreasing period (NDP). We have also proposed a new heuristic algorithm CNZ. From the worst-case point of view NID is much worse than NDP and CNZ, while the latter two are equally good. In average-case CNZ seems to be better than both NDP and CNZ. From computational experiments follows that when maximum duration is at least 6 times bigger than the minimal one, the CNZ's relative error is about 3 times smaller than that of NID and NDP.

Solving the Open Shop Scheduling Problem
Ulrich Dorndorf

We describe a branch-and-bound algorithm for solving the open shop scheduling problem which performs better than other exact algorithms. The key to the efficiency of our algorithm lies in the following approach: instead of analyzing and improving the search strategies for finding solutions, we focus on constraint propagation based methods for reducing the search space. Computational experiments for several sets of well-known benchmark problem instances are reported. For the first time, many problem instances are solved to optimality in a short amount of computation time.

Bounds for the Resource Investment Problem
Andreas Drexl and Alf Kimms

The resource investment problem deals with the issue of providing resources to a project such that a given deadline can be met. The objective is to make the resources available in the cheapest possible way. For each resource, expenses depend on the maximum amount required during the course of the project. In this paper we develop two lower bounds for this $\mathcal{NP}$–hard problem using Lagrangian relaxation and column generation techniques, respectively. Both procedures are capable of yielding feasible solutions as well. Hence, we also have two optimization guided heuristics. A computational study consisting of a set of 3210 instances compares both approaches and allows insight into the performance. E.g., for the instances from Möhring's test set it turns out that in 58% of the cases the heuristic solution derived on the basis of Lagrangian relaxation is optimal. Using column generation, the gap between the lower bound and the optimum objective function value is below 5% in 49% of the cases, it is below 10% in 68% of the cases, and it is below 20% in all cases.

Scheduling Multiprocessor Tasks for Mean Flow Time Criterion
MACIEJ DROZDOWSKI

Multiprocessor tasks are executed by more than one processor at the same moment of time. This work considers the problem of scheduling unit execution-time and preemptable multiprocessor tasks on m processors to minimize mean flow time and mean weighted flow time. We analyze complexity status of the problem. For parallel identical processors, when tasks have unit execution time and the number of processors is arbitrary the problem is shown to be computationally hard. Constructing an optimal preemptive schedule is also computationally hard in general. Polynomial algorithms are presented for scheduling unit execution time tasks when the number of processors is fixed, or the numbers of simultaneously required processors are powers of 2. The case of preemptable tasks requiring either 1 or m dedicated processors simultaneously is solvable in low-order polynomial time.

Repeated Execution of Non-Preemptive Tasks With Relative Timing Constraints
KLAUS H. ECKER

Real-Time systems, in particular parallel and distributed realizations, gained increasing interest during the last years. Among the many problems for the correct functioning of a real-time system is a careful scheduling of the computational activities. Examples are the periodic reading of instruments that inform about the status of the controlled system, evaluating the measured data, computing new data, and producing signals for setting actuator elements appropriately to ensure the expected functioning of the environment.

The main objective of this presentation is to discuss some scheduling problems, and especially contrast the periodic task paradigm to a new approach of repeated task execution guided by relative timing conditions. As will be seen, this approach offers greater flexibility in scheduling the tasks, and allows, as compared to periodic tasks, to reduce processor utilization while the response times still remain within their limits. As a consequence more free processor capacity is left to process sporadic tasks.

Scheduling with SLK an TWK Due Date Assignment
VALERY GORDON, JEAN-MARIE PROTH AND CHENGBIN CHU

A review of the research of due date assignment and scheduling problems where due dates are assigned according to SLK or TWK rules is presented. These methods of due date assignment represent examples of considering due dates as the decision variables of the scheduling problems and reflect the situation when the decision maker assigns due dates by estimating job flow times.

The SLK due date assignment method determines due date by adding a common *slack* to processing time of each job. In the TWK due date assignment method, where TWK stands for *total work* content, the due dates are equal to a multiple of job processing times and a multiplier common to all the jobs.

The results on algorithms and complexity of SLK and TWK due date assignment and scheduling problems are summarized. The research was supported by INTAS (project 96-0820).

Sensitivity Analysis for Scheduling Algorithm Design
Frédéric Guinand

This talk was an attempt to make use of sensitivity analysis not only for qualifying already existing algorithms but also for extracting properties from these algorithms for designing new ones. For that purpose we introduce a new graph called PPG (Precedence Processors Graph). Instead of directly considering algorithms, we try to determine which families of PPG characterize low sensitive schedule. The next step consists "simply" in developing the algorithm able to build schedules associated with such PPGs. This analysis can also give a new insight in scheduling with communication delays by making the structure of these communications more important than their number.

Production Planning and Scheduling of a Cast-to-Stock Metal Foundry
Erwin Hans and Steef van de Velde

We describe our algorithmic approach for the monthly production planning and scheduling problem of the in-house metal foundry of a company that produces central heating boilers to order. For operational safety reasons, however, the foundry casts entirely to stock. Every month, the master production schedule specifies the replenishment orders for the foundry. The foundry planning problem is to determine which and how many products are to be cast on each day of the next month as well as the sequence in which they are to be cast so as to level out the total needed amount of molten iron over the working days of the planning period subject to a variety of capacity constraints, the availability of resources and materials, and process requirements.

Our approach proceeds by decomposing the overall problem into a planning subproblem, where the replenishment orders need to be divided over the working days, and a scheduling subproblem, where the products per day as well as the working days need to be sequenced. We use mixed integer linear programming for the planning subproblem, a polynomial dynamic programming algorithm for sequencing the products per day, and iterative local improvement for sequencing the working days. Finally, we also describe how the resulting production schedule can be improved further. The quality of the production schedules obtained with our approach, a combination of mixed integer linear programming, shortest path algorithms, and iterative local improvement, is by far superior to the quality of the schedules found with the previous manual method: the amount of molten iron needed on any day deviates no more than 1% from the average, whereas with the manual method it used to deviate up to 5%; the approach saves on average two changeovers (of about 10 minutes each) per eight-hour working day; and the work-in-process is significantly less. Furthermore, the planning effort is significantly reduced: the manual method used to take about 2-3 days, whereas a typical planning session with the decision support system in which the our algorithmic approach has been embedded takes no more than half an hour.

Restarts Can Help in On-Line Scheduling
Han Hoogeveen

We consider a single-machine on-line scheduling problem where jobs arrive over time. A set of independent jobs has to be scheduled on a single machine that is continuously available from time zero onwards and that can handle at most one job at a time. Each job becomes available at its release date, which is not known in advance, and its processing time and delivery time become known at its arrival. The objective is to minimize the time by which all jobs have been delivered. In our model preemption is not allowed, but we are allowed to restart a job, that is, the processing of a job can be broken off to have the machine available to process an urgent job, but the time already spent on processing this interrupted job is considered to be lost. We propose an on-line algorithm and show that its performance bound is equal to $\frac{3}{2}$, which matches a known lower bound due to Vestjens. The performance bound of $\frac{3}{2}$ means a small improvement of the optimal worst-case bound of $\frac{1}{2}(\sqrt{5}+1) \approx 1,61803$ known for the same problem in which restarts are not allowed.

Single Machine Scheduling Problems with Deteriorating Jobs Dependent on Resources
Adam Janiak and A. Bachman

A new linear model of job processing time dependent on starting moment and constrained resources is introduced. We consider the single machine problems of scheduling jobs and resource allocation for the following three criteria: makespan, total completion time and maximum lateness. For the makespan criterion we prove the NP-completeness of the problem and present some polynomially-solvable cases. Since the remaining problems are NP-complete, we present only the polynomially-solvable cases for them.

A Linear Time Approximation Scheme for the Job Shop Scheduling Problem
Klaus Jansen, Roberto Solis-Oba and Maxim Sviridenko

In the job shop scheduling problem, there is a set $\mathcal{J} = \{J_1, \ldots, J_n\}$ of n jobs that must be processed on a group $M = \{1, \ldots, m\}$ of m machines. Each job J_j consists of a sequence of operations $O_{1j}, O_{2j}, \ldots, O_{\mu j}$, where O_{ij} must be processed on machine $m_{ij} \in \{1, \ldots, m\}$ during p_{ij} time units. The operations $O_{1j}, O_{2j}, \ldots, O_{\mu j}$ must be processed one after another in the given order and each machine can process at most one operation at a time.

The problem is to schedule the jobs so as to minimize the overall *makespan*, or schedule *length*, which is the time by which all jobs are completed. We study the preemptive and non-preemptive versions of the job shop scheduling problem when the number of machines and the number of operations per job are fixed. We present linear time approximation schemes for both problems, improving the best previous approximation algorithm with ratio $2+\epsilon$ by Smoys, Stein and Wein.

A Local Search Algorithm for the TSP with Time Windows and Generalized Precedence Constraints
Sigrid Knust

We consider a single-machine scheduling problem which arises as a subproblem in a job-shop environment where the jobs have to be transported between the machines by a single transport robot. The robot scheduling problem may be regarded as a generalization of the traveling salesman problem with time windows, where additionally generalized precedence constraints have to be respected. The objective is to determine a sequence of all nodes and corresponding starting times in the given time windows in such a way that all generalized precedence relations are respected and the sum of all traveling and waiting times is minimized.

We present a local search algorithm for this problem where an appropriate neighborhood structure is defined using problem-specific properties. In order to make the search process more efficient, we apply some techniques which accelerate the evaluation of the solutions in the proposed neighborhood considerably. Computational results are presented for test data arising from job-shop instances with a single transport robot.

Assembly Scheduling with Two Types of Batching
Mikhail Y. Kovalyov, Chris N. Potts and Vitaly A. Strusevich

Processing of each job includes $m+1$ operations. Each of the first m operations of a same job is a production of a component on a specific *component machine*. Component machines work in parallel. After the first m operations are completed, operation $m+1$ may start, which is to combine all the components into a final product on the *assembly machine*. It is assumed that the jobs are processing by the machines in *batches*. The following batching policies and availability rules are considered. Within a batch, the jobs are processed either sequentially or simultaneously. The batch processing time is then equal to the sum or maximum of the processing times of its jobs, respectively. The jobs in the same batch are either available at the same time when the whole batch is completed or each job is individually available when its processing is finished.

The objective is to find a batch schedule minimizing the makespan.

Computational complexities of various special cases of the problem with simultaneous job processing are established. For the problem with sequential job processing and individual job availability, an approximation algorithm is derived and its worst-case performance is analyzed.

Dedicated Scheduling of Biprocessor Tasks to Minimize Mean Flow Time
Marek Kubale

In the talk we investigate the computational complexity of scheduling of biprocessor tasks on dedicated processors from a mean flow time point of view. The first part of the talk is devoted to the case where arbitrary processing times are allowed. It is shown that problem P—fix_j=2,M=doublestar—ΣC_j is NP-hard.

On the other hand, if the scheduling graph is a path or a star then the problem becomes polynomially solvable in quadratic time. The second part deals with the case where the processing times are 0 or 1. We show that the UET problem is equivalent to finding the chromatic sum of the line-graph of G. The P—$fix_j = 2$,UET,M=subcubic-bipartite—ΣC_j problem is shown to be NP-hard, but P—$fix_j = 2$,UET,M=tree—ΣC_j is polynomially solvable. The last result can be generalized to bounded cyclicity graphs.

Fully Polynomial Time Approximation Schemes for CTV
Wieslaw Kubiak, Jinkiang Cheng and Mikhail Y. Kovalyov

We present two fully polynomial approximation schemes for the problem of minimizing completion time variance (CTV) of a finite set of jobs on a single machine. Though both have the same worst case time complexity, our computational experiments show that one finds an approximate solution with given accuracy significantly faster than the other for all instances tested. We also show bounds that the schemes can use to further reduce solution space when looking for an approximate solution.

On-line Scheduling Parallel Jobs
Ernst W. Mayr and Stefan Bischof

For the efficient use of parallel and (tightly coupled) distributed architectures, it is necessary to achieve good load balancing among the processing units. There are numerous problem variants in this area, and approaches and solutions of vastly differing complexity.

While most investigations of scheduling problems have been concerned with the off-line case, where all information about the jobs and their mutual precedence is known in advance, practical problems are often on-line, such that most information is not known until execution.

We present a number of scenarios for which we have developed simple scheduling and load balancing algorithms. Since the underlying problems are NP-hard, we content ourselves with approximation algorithms, for which we present optimal or nearly optimal quality guarantees (in the sense of "competitive ratio").

Some of the algorithms have been implemented as part of a large FE application on a network of workstations, for others we have performed extensive simulations in order to study their applicability and to supplement the theoretic analysis.

Risk Analysis in Project Scheduling
Rolf H. Möhring, Andreas Schulz, Frederik Stork and Marc Uetz

Deterministic models for scheduling and project control usually suffer from the fact that they assume complete information and neglect random influences that occur during project execution. A typical consequence is the underestimation of the expected project duration and cost frequently observed in practice.

The area of *stochastic scheduling* provides theory and methods to cope with these phenomena. I survey recent developments for problems where processing times are random but precedence and resource constraints are fixed. In this case, scheduling is done by a dynamic process of decisions (a *policy*) which are based on the observed past and the a priori knowledge of the distribution of processing times.

Two important topics are a new class of scheduling policies with good algorithmic properties and an analysis of the approximation behavior of simple priority policies in machine scheduling.

Net Present Value Project Scheduling - Basic Concepts and Schedule Generation Scheme
KLAUS NEUMANN

The paper deals with the problem of maximizing the net present value of a project subject to temporal constraints (given by minimum and maximum time lags between project activities) and resource constraints. The objective function of that problem is nonregular and not concave. The feasible region is generally disconnected and represents the union of finitely many polytopes, whereas the feasible region of the corresponding problem without resource constraints is a polytope. Different kinds of feasible schedules are introduced: so-called stable schedules correspond to extreme points, and pseudostable schedules correspond to local extreme points of the feasible region of the resource-constrained problem. There is always an optimal schedule to the latter problem which is pseudostable.

A very efficient steepest ascent algorithm for the problem without resource constraints is presented which, in each iteration, moves from a vertex of the feasible region to a different (not necessarily adjacent) vertex. For the problem with resource constraints, a schedule generation scheme is proposed which is based upon resolving resource conflicts by successively introducing additional temporal constraints in an appropriate way.

Branch and Propagate for Project Scheduling
ERWIN PESCH, ULRICH DORNDORF AND TOAN PHAN HUY

The resource-constrained project scheduling problem with generalized precedence constraints is a very general scheduling model with applications in areas such as make-to-order production planning. We describe a time-oriented branch-and-bound algorithm for the problem that uses constraint-propagation techniques which actively exploit the temporal and resource constraints of the problem in order to reduce the search space. Extensive computational experiments with systematically generated test problems show that the algorithm solves more problems to optimality and feasibility and performs better than other exact solution procedures which have recently been proposed, and that the truncated version of the algorithm is also a very good heuristic.

Scheduling with Duplication and Large Communication Delays
CHRISTOPHE RAPINE

We consider the problem of scheduling an application depicted by a directed acyclic graph in presence of communication delays. Performances of scheduling algorithms mainly depend on the granularity ratio ρ between communication and computation. On an unbounded number of processors, duplication of the tasks allows to get efficient approximation algorithms with guarantee independent of ρ. We show how list scheduling algorithm technics can reduce a schedule of guarantee λ on an unbounded number of processors to a schedule of guarantee $\mathcal{O}(\lambda\sqrt{\rho})$ on any fixed number of processors, using duplication. The algorithm results from a compromise between the earliest execution date of the tasks and the amount of duplication needed to get it, and provides the up-to-now best worst case guarantee on the problem. The question to know if this general scheduling problem with communication is approximable within a constant (i.e. independent of ρ) remains still open.

A Simple PTAS for Minimizing Average Completion Time
CLIFFORD STEIN

We consider the problem of scheduling n jobs with release dates on m machines so as to minimize their average weighted completion time. We present the first known polynomial time approximation schemes for several variants of this problem. Our results include PTASs for the case of identical parallel machines and a constant number of unrelated machines with and without preemption allowed. Our schemes are efficient: for all variants the running time for a $(1+\epsilon)$ approximation is of the form $f(1/\epsilon, m)poly(n)$. In this talk we focus on the one machine case, unweighted case, and give a particularly simple algorithm.

Scheduling with AND/OR Precedence Constraints
FREDERIK STORK, ROLF H. MÖHRING AND MARTIN SKUTELLA

A natural generalization of ordinary precedence constraints are so-called AND/OR precedence constraints. In an AND constraint, a job must wait *for all* its predecessors while in an OR constraint, a job has to wait *for at least one* of its predecessors. We provide a linear-time algorithm for deducing additional AND/OR precedence constraints that are implied by the given ones. We show that this algorithm can also be used to verify feasibility of the given constraints. Besides their theoretical value, these results have significant impact in practical applications such as scheduling and assembly sequencing; we show how to use our algorithm to improve solution procedures for resource-constrained project scheduling problems. We finally discuss the issue of computing earliest job starting times and a game-theoretic interpretation thereof.

Scheduling Malleable Tasks
DENIS R. TRYSTRAM, RENAUD LEPERE, GREGORY MOUNIE AND CHRISTOPHE RAPINE

This story started a few years ago in a similar place. The friendship meeting of a quiet and peaceful atmosphere created good conditions for imaging a new way for looking at some scheduling problems in the field of parallel processing. We know since a long time that it is necessary to take into account the impact of communication delays. Most of the models used until that time lead to hard problems because essentially the communications are taken into account explicitly, even for the simplest idealized models. Moreover, due to several reasons as the determination of the good granularity, the tasks of the application contain themselves some parallelism. The combination of both considerations leads to the concept of Malleable tasks (which are computational units that can be executed in parallel) where the communications are handled implicitly using a global penalty factor.

The goal of this talk was first to present the model and then, discuss some scheduling results on Malleable Tasks. These theoretical results were applied for solving an actual code in oceanography (simulation of the circulation in the Atlantic ocean).

Resource-Constrained Project Scheduling: From a Lagrangian Relaxation to Competitive Solutions
MARC UETZ AND FREDERIK STORK

List scheduling belongs to the classical and widely used algorithms for complex scheduling problems, but folklore priority lists usually do not capture much of the problem structure, thus resulting in poor performance. We use a Lagrangian relaxation for resource-constrained project scheduling, primarily designed to compute lower bounds on the optimum objective value, to obtain more appropriate priority lists. The Lagrangian relaxation can be efficiently solved as a minimum cut problem, and in each iteration of a (standard) subgradient optimization, we thus obtain resource-infeasible schedules. Priority lists are then obtained by an ordering according to so-called alpha-completion times of jobs.

Our computational results show that the schedules thus obtained improve considerably upon those obtained by folklore priority lists, and the procedure even compares to state-of-the-art local search algorithms. Perhaps more important, since lower bounds and feasible solutions are computed at the same time, the gap between lower and upper bounds can be drastically reduced within very reasonable time.

Recent Results in Discrete-Continuous Scheduling
JAN WĘGLARZ AND JOANNA JÓZEFOWSKA

We consider a project scheduling problem in which nonpreemptive activities simultaneously require for their processing m discrete, renewable resources and one renewable continuous resource. Resource requirements concerning discrete resources are fixed, whereas the continuous resource can be allotted arbitrarily from the interval (0,1]. Processing speed of activity i is given by the formula:

$$\dot{x}(t) = f_i[u_i(t)], x_i(0) = 0, x_i(C_i) = \tilde{x}_i \quad (1)$$

where $u_i(t)$ is the continuous resource amount allotted to i at t, f_i is continuous and concave, $f_i(0) = 0$, C_i is (unknown in advance) completion time of i, $\tilde{x}_i$ is known, $x_i(t)$ is the state of i at time t.

The criterion is the project duration. We show the transformation of the above problem to the Multimode Resource-Constrained Project Scheduling Problem (for discretized continuous resource allocations) and the results of the computational experiment with SA for different numbers of the continuous resource allotments (i.e. processing modes of activities).

Scheduling with Non-Regular Criteria
Frank Werner

In this talk we consider two-machine flow shop scheduling problems with a common due date and the objective to minimize the total penalty for arbitrary job-specific penalty functions depending on the deviation of the completion time from the due date. After summarizing some properties for single and parallel machine problems of this type, we give some properties of the two-machine flow shop problem. We present an enumerative algorithm for such problems and report on results with different asymmetric job-independent penalty functions. The algorithm is able to solve problems with up to 20 jobs quite satisfactorily.

Importance of Human Resources in Scheduling Problems
Marino Widmer

In this talk, two case studies dealing with human resource management are presented.

The first case study concerns the planning of an exam session. It is shown that it is possible to use layout techniques for solving this kind of timetabling problems. A heuristic method, including Lemonias algorithm and Hamming distances, taking into account the fact that a student has to pass no more than one exam per day, is presented.

The second case study deals with the minimization of the number of operators working on a rythmed flow line, in which each job stays at each workstation during 2 days, independently of its processing time. The ILP formulation of this problem is presented and a heuristic method, based on tabu search technique, is proposed.

Net Present Value Project Scheduling - Branch-and-Bound and Tabu-Search-Methods
Jürgen Zimmermann

We present a branch-and-bound-procedure for the resource-constrained net present value problem with general temporal constraints based on the concept of minimal delaying alternatives. Each enumeration node represents a resource-unconstrained net present value problem, which is solved by means of a dual method. Starting with the solution of the corresponding father-node represented by a spanning tree additional precedence constraints, which resolves current resource conflicts, are added to the spanning tree shifting appropriate subtrees. To improve generated feasible solutions, we eliminate superfluous precedence constraints added in course of the procedure, since there is always an optimal solution to the resource-constrained net present value problem which is pseudostable.

Furthermore, a tabu-search-procedure based on a spanning tree representation is discussed. The neighborhood is given by different shifts of subtrees and the tabu-list contains polytopes of the feasible region already investigated as well as arcs added to spanning trees of investigated solutions. In addition, local optimization steps are integrated.

An experimental performance analysis shows that the proposed procedures for net present value problems with and without resource constraints clearly outperform all methods known from literature.