DMP204
SCHEDULING,
TIMETABLING AND ROUTING

Lecture 1

Introduction to Scheduling: Terminology, Classification

Outline

1. Course Introduction
2. Scheduling
   Problem Classification
3. Complexity Hierarchy

Outline

1. Course Introduction
2. Scheduling
   Problem Classification
3. Complexity Hierarchy

Course Presentation

Communication media

- Black Board (BB). What we use:
  - Mail
  - Announcements
  - Course Documents (for Photocopies)
  - Blog – Lecture Diary
  - Electronic hand in of the exam project
- Web-site http://www.imada.sdu.dk/~marco/DM204/
  - Lecture plan and slides
  - Literature and Links
  - Exam documents

- Schedule

Third quarter 2008		Fourth quarter 2008	
Tuesday	10:15-12:00	Wednesday	12:15-14:00
Friday	8:15-10:00	Friday	10:15-12:00

- $\sim$ 27 lectures

Evaluation

- Final Assessment (10 ECTS)
  - Oral exam: 30 minutes + 5 minutes defense project
    meant to assess the base knowledge
  - Group project:
    free choice of a case study among few proposed ones
    Deliverables: program + report
    meant to assess the ability to apply
- Schedule: Project hand in deadline + oral exam in June

Course Content

- General Optimization Methods
  - Mathematical Programming,
  - Constraint Programming,
  - Heuristics
  - Problem Specific Algorithms (Dynamic Programming, Branch and Bound)
- Scheduling
  - Single and Parallel Machine Models
  - Flow Shops and Flexible Flow Shops
  - Job Shops
  - Resource-Constrained Project Scheduling
- Timetabling
  - Interval Scheduling, Reservations
  - Educational Timetabling
  - Workforce and Employee Timetabling
  - Transportation Timetabling
- Vehicle Routing
  - Capacited Vehicle Routing
  - Vehicle Routing with Time Windows

Course Material

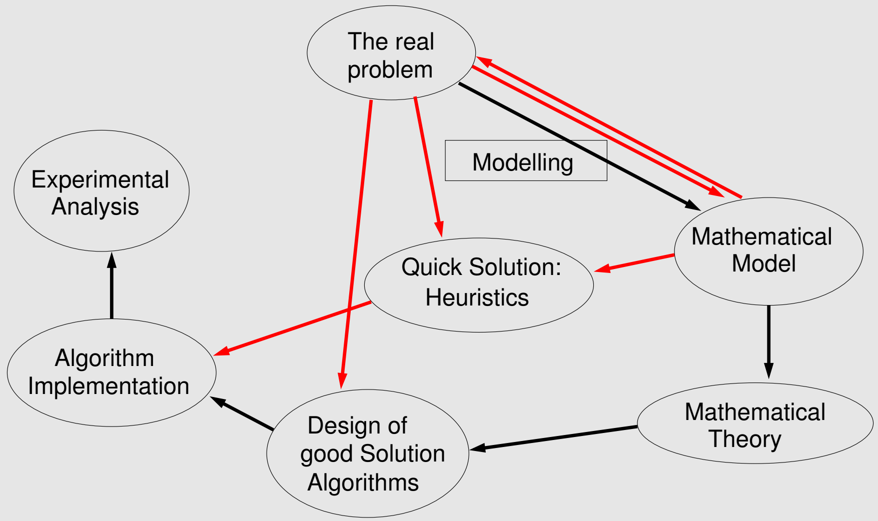
- Literature
  - B1 Pinedo, M. Planning and Scheduling in Manufacturing and Services Springer Verlag, 2005
  - B2 Pinedo, M. Scheduling: Theory, Algorithms, and Systems Springer New York, 2008
  - B3 Toth, P. & Vigo, D. (ed.) The Vehicle Routing Problem SIAM Monographs on Discrete Mathematics and Applications, 2002
- Slides
- Class exercises (participatory)

Course Goals and Project Plan

How to Tackle Real-life Optimization Problems:

- Formulate (mathematically) the problem
- Model the problem and recognize possible similar problems
- Search in the literature (or in the Internet) for:
  - complexity results (is the problem NP-hard?)
  - solution algorithms for original problem
  - solution algorithms for simplified problem
- Design solution algorithms
- Test experimentally with the goals of:
  - configuring
  - tuning parameters
  - comparing
  - studying the behavior (prediction of scaling and deviation from optimum)

The problem Solving Cycle



Outline

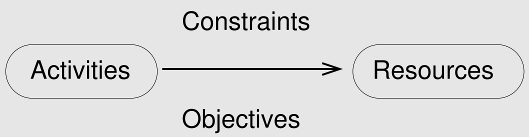
1. Course Introduction
2. Scheduling
   Problem Classification
3. Complexity Hierarchy

Scheduling

- Manufacturing
  - Project planning
  - Single, parallel machine and job shop systems
  - Flexible assembly systems
    Automated material handling (conveyor system)
  - Lot sizing
  - Supply chain planning
- Services

$\Rightarrow$ different algorithms

Problem Definition



Problem Definition

Given: a set of **jobs** $\mathcal{J} = \{J_1, \ldots, J_n\}$ that have to be processed by a set of **machines** $\mathcal{M} = \{M_1, \ldots, M_m\}$

Find: a **schedule**,

i.e., a mapping of jobs to machines and processing times subject to feasibility and optimization constraints.

Notation:

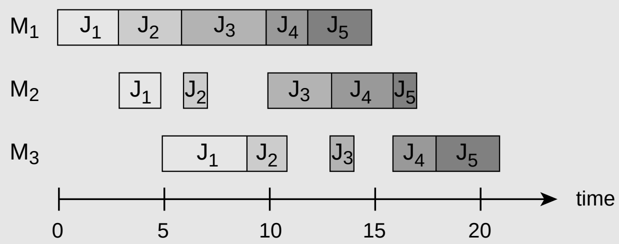
n, j, k jobs

m, i, h machines

Visualization

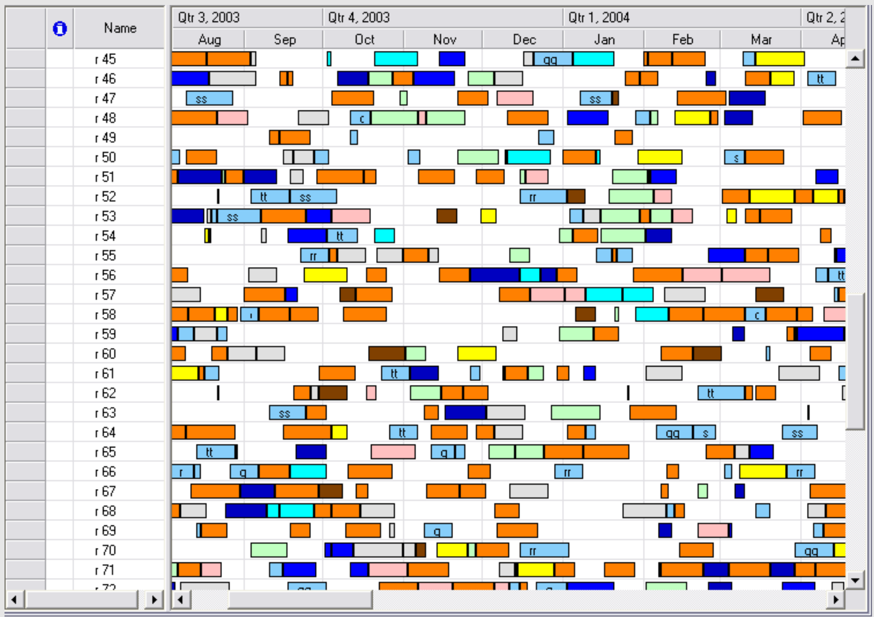
Scheduling are represented by Gantt charts

- machine-oriented



- or job-oriented

  ...



Data Associated to Jobs

- Processing time p_{ij}
- Release date r_j
- Due date d_j (called deadline, if strict)
- Weight w_j

- A job J_j may also consist of a number n_j of operations $O_{j1}, O_{j2}, \ldots, O_{jn_j}$ and data for each operation.
- Associated to each operation a set of machines $\mu_{jl} \subseteq \mathcal{M}$

Data that depend on the schedule (dynamic)

- Starting times S_{ij}
- Completion time C_{ij}, C_j

Problem Classification

A scheduling problem is described by a triplet $\alpha \mid \beta \mid \gamma$.

- α machine environment (one or two entries)
- β job characteristics (none or multiple entry)
- γ objective to be minimized (one entry)

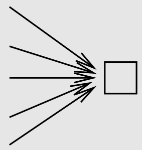
[R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan (1979): Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 4, 287-326.]

$\alpha \mid \beta \mid \gamma$ Classification Scheme

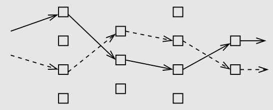
Machine Environment $\boldsymbol{\alpha_1\alpha_2} \mid \beta_1 \dots \beta_{13} \mid \gamma$

- single machine/multi-machine ($\alpha_1 = \alpha_2 = 1$ or $\alpha_2 = m$)
- parallel machines: identical ($\alpha_1 = P$), uniform p_j/v_i ($\alpha_1 = Q$), unrelated p_j/v_{ij} ($\alpha_1 = R$)
- multi operations models: Flow Shop ($\alpha_1 = F$), Open Shop ($\alpha_1 = O$), Job Shop ($\alpha_1 = J$), Mixed (or Group) Shop ($\alpha_1 = X$)

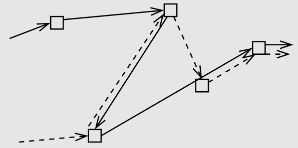
Single Machine



Flexible Flow Shop ($\alpha = FFc$)



Open, Job, Mixed Shop



$\alpha \mid \beta \mid \gamma$ Classification Scheme

Job Characteristics $\alpha_1\alpha_2 \mid \boldsymbol{\beta_1 \dots \beta_{13}} \mid \gamma$

- $\beta_1 = prmp$ presence of preemption (resume or repeat)
- β_2 precedence constraints between jobs (with $\alpha = P, F$) acyclic digraph $G = (V, A)$
  - $\beta_2 = prec$ if G is arbitrary
  - $\beta_2 = \{chains, intree, outtree, tree, sp\text{-}graph\}$
- $\beta_3 = r_j$ presence of release dates
- $\beta_4 = p_j = p$ preprocessing times are equal
- ($\beta_5 = d_j$ presence of deadlines)
- $\beta_6 = \{s\text{-}batch, p\text{-}batch\}$ batching problem
- $\beta_7 = \{s_{jk}, s_{jik}\}$ sequence dependent setup times

$\alpha \mid \beta \mid \gamma$ Classification Scheme

Job Characteristics (2) $\alpha_1\alpha_2 \mid \boldsymbol{\beta_1 \dots \beta_{13}} \mid \gamma$

- $\beta_8 = brkdwn$ machines breakdowns
- $\beta_9 = M_j$ machine eligibility restrictions (if $\alpha = Pm$)
- $\beta_{10} = prmu$ permutation flow shop
- $\beta_{11} = block$ presence of blocking in flow shop (limited buffer)
- $\beta_{12} = nwt$ no-wait in flow shop (limited buffer)
- $\beta_{13} = recrc$ Recirculation in job shop

$\alpha \mid \beta \mid \gamma$ Classification Scheme

Objective (always $f(C_j)$) $\alpha_1\alpha_2 \mid \beta_1\beta_2\beta_3\beta_4 \mid \gamma$

- Lateness $L_j = C_j - d_j$
- Tardiness $T_j = \max\{C_j - d_j, 0\} = \max\{L_j, 0\}$
- Earliness $E_j = \max\{d_j - C_j, 0\}$
- Unit penalty $U_j = \begin{cases} 1 & \text{if } C_j > d_j \\ 0 & \text{otherwise} \end{cases}$

$\alpha \mid \beta \mid \gamma$ Classification Scheme

Objective $\alpha_1\alpha_2 \mid \beta_1\beta_2\beta_3\beta_4 \mid \gamma$

- Makespan: Maximum completion $C_{max} = \max\{C_1, \ldots, C_n\}$ tends to max the use of machines
- Maximum lateness $L_{max} = \max\{L_1, \ldots, L_n\}$
- Total completion time $\sum C_j$ (flow time)
- Total weighted completion time $\sum w_j \cdot C_j$ tends to min the av. num. of jobs in the system, ie, work in progress, or also the throughput time
- Discounted total weighted completion time $\sum w_j(1 - e^{-rC_j})$
- Total weighted tardiness $\sum w_j \cdot T_j$
- Weighted number of tardy jobs $\sum w_j U_j$

All regular functions (nondecreasing in $C_1, \ldots, C_n$) except E_i

$\alpha \mid \beta \mid \gamma$ Classification Scheme

Other Objectives $\alpha_1\alpha_2 \mid \beta_1\beta_2\beta_3\beta_4 \mid \gamma$

Non regular objectives

- Min $\sum w'_j E_j + \sum w"_j T_j$ (just in time)
- Min waiting times
- Min set up times/costs
- Min transportation costs

Exercises

Gate Assignment at an Airport

- Airline terminal at a airport with dozes of gates and hundreds of arrivals each day.
- Gates and Airplanes have different characteristics
- Airplanes follow a certain schedule
- During the time the plane occupies a gate, it must go through a series of operations
- There is a scheduled departure time (due date)
- Performance measured in terms of on time departures.

Exercises

Scheduling Tasks in a Central Processing Unit (CPU)

- Multitasking operating system
- Schedule time that the CPU devotes to the different programs
- Exact processing time unknown but an expected value might be known
- Each program has a certain priority level
- Minimize the time expected sum of the weighted completion times for all tasks
- Tasks are often sliced into little pieces. They are then rotated such that low priority tasks of short duration do not stay for ever in the system.

Exercises

Paper bag factory

- Basic raw material for such an operation are rolls of paper.
- Production process consists of three stages: (i) printing of the logo, (ii) gluing of the side of the bag, (iii) sewing of one end or both ends.
- Each stage consists of a number of machines which are not necessarily identical.
- Each production order indicates a given quantity of a specific bag that has to be produced and shipped by a committed shipping date or due date.
- Processing times for the different operations are proportional to the number of bags ordered.
- There are setup times when switching over different types of bags (colors, sizes) that depend on the similarities between the two consecutive orders
- A late delivery implies a penalty that depends on the importance of the order or the client and the tardiness of the delivery.

Solutions

Distinction between

- sequence
- schedule
- scheduling policy

Feasible schedule

A schedule is feasible if no two time intervals overlap on the same machine, and if it meets a number of problem specific constraints.

Optimal schedule

A schedule is optimal if it is feasible and it minimizes the given objective.

Classes of Schedules

Semi-active schedule

A feasible schedule is called semi-active if no operation can be completed earlier without changing the order of processing on any one of the machines. (local shift)

Active schedule

A feasible schedule is called active if it is not possible to construct another schedule by changing the order of processing on the machines and having at least one operation finishing earlier and no operation finishing later. (global shift without preemption)

Nondelay schedule

A feasible schedule is called nondelay if no machine is kept idle while an operation is waiting for processing. (global shift with preemption)

- There are optimal schedules that are nondelay for most models with regular objective function.
- There exists for $Jm||\gamma$ (γ regular) an optimal schedule that is active.
- nondelay $\Rightarrow$ active but active $\not\Rightarrow$ nondelay

Outline

1. Course Introduction

2. Scheduling
   Problem Classification

3. Complexity Hierarchy

Complexity Hierarchy

Reduction

A search problem Π is (polynomially) reducible to a search problem Π' ($\Pi \longrightarrow \Pi'$) if there exists an algorithm $\mathcal{A}$ that solves Π by using a hypothetical subroutine $\mathcal{S}$ for Π' and except for $\mathcal{S}$ everything runs in polynomial time. [Garey and Johnson, 1979]

NP-hard

A search problem Π' is NP-hard if

1. it is in NP
2. there exists some NP-complete problem Π that reduces to Π'

In scheduling, complexity hierarchies describe relationships between different problems.

Ex: $1||\sum C_j \longrightarrow 1||\sum w_j C_j$

Interest in characterizing the borderline: polynomial vs NP-hard problems

Problems Involving Numbers

Partition

- **Input:** finite set A and a size $s(a) \in \mathbf{Z}^+$ for each $a \in A$
- **Question:** is there a subset $A' \subseteq A$ such that

$$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)?$$

3-Partition

- **Input:** set A of $3m$ elements, a bound $B \in \mathbf{Z}^+$, and a size $s(a) \in \mathbf{Z}^+$ for each $a \in A$ such that $B/4 < s(a) < B/2$ and such that $\sum_{a \in A} s(a) = mB$
- **Question:** can A be partitioned into m disjoint sets $A_1, \ldots, A_m$ such that for $1 \leq i \leq m$, $\sum_{a \in A_i} s(a) = B$ (note that each A_i must therefore contain exactly three elements from A)?