DM87
SCHEDULING,
TIMETABLING AND ROUTING

Lecture 1

Course Introduction.
Scheduling: Terminology and Classification

Marco Chiarandini

Outline

1. Course Introduction

2. Scheduling
   Problem Classification

Outline

1. Course Introduction

2. Scheduling
   Problem Classification

Course Presentation

- Communication media
  - Blackboard (for private communications)
    Mail, Fora, Blog, Grades, Documents (photocopies),
  - Web-site `http://www.imada.sdu.dk/~marco/DM87/`
    Lecture plan, Syllabus, Links, Exam documents
- 40 hours of lectures + work at the exam project
- Schedule:
  1. Lectures:
     Mondays 12:00-13:45, Thursdays 8:15-10:00
     Weeks 5-10, 13-16
     Last lecture (preliminary date): Thursday, April 17
  2. Exam: June

Course Content

- Review of Optimization Methods:
  - Mathematical Programming,
  - Constraint Programming,
  - Heuristics
  - Problem Specific Algorithms (Dynamic Programming, Branch and Bound)
- Introduction to Scheduling, Terminology, Classification.
  - Single Machine Models
  - Parallel Machine Models
  - Flow Shops and Flexible Flow Shops
  - Job Shops, Open Shops
- Introduction to Timetabling, Terminology, Classification
  - Interval Scheduling, Reservations
  - Educational Timetabling
  - Workforce and Employee Timetabling
  - Transportation Timetabling
- Introduction to Vehicle Routing, Terminology, Classification
  - Capacited Vehicle Routing
  - Vehicle Routing with Time Windows

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

Course Material

- Literature
  - **Text book**: M.L. Pinedo, Planning and Scheduling in Manufacturing and Services; Springer Series in Operations Research and Financial Engineering, 2005. (388 DKK)
  - **Supplementary book**: M.L. Pinedo, Scheduling: Theory, Algorithms, and Systems; 2nd ed., Prentice Hall, 2002.
  - **Supplementary book**: P. Toth, D. Vigo, eds. The Vehicle Routing Problem, SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 2002.
  - **Supplementary Articles**: will be indicated during the course
- Slides
- Class exercises (participatory)

Useful Previous Knowledge for this Course

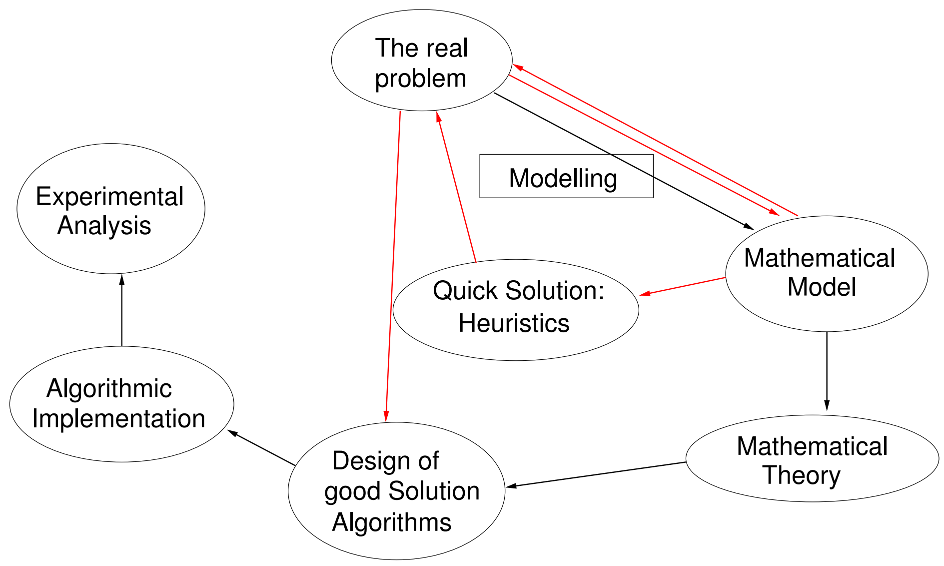
- Algorithms and data structures
- Programming A and B
- Networks and Integer Programming
- Heuristics for Optimization
- Software Methodologies and Engineering

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

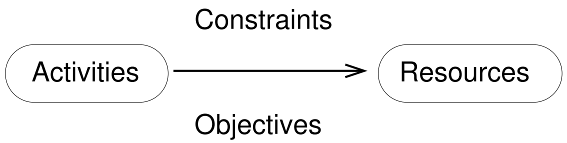
2. Scheduling
   Problem Classification

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

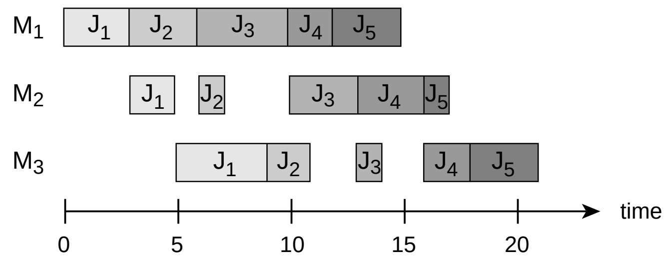
Given: a set of **jobs** $\mathcal{J} = \{J_1, \dots, J_n\}$ that have to be processed
by a set of **machines** $\mathcal{M} = \{M_1, \dots, M_m\}$
Find: a **schedule**,
i.e., a mapping of jobs to machines and processing times
subject to feasibility and optimization constraints.

Notation:
n, j, k jobs
m, i, h machines

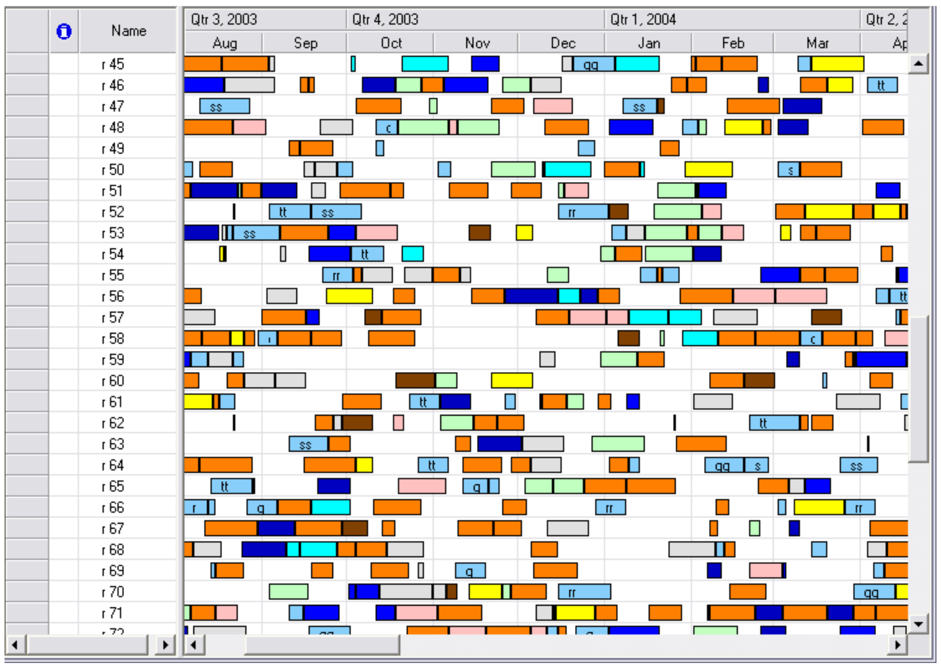
Visualization

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 $O_{j1}, O_{j2}, \dots, O_{jn_j}$ 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 $\alpha \,|\, \beta \,|\, \gamma$.

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

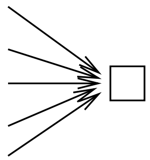
[R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan (1979): Optimization and approximation in deterministic sequencing and scheduling: a survey, Ann. Discrete Math. 4, 287-326.]

The $\alpha \,|\, \beta \,|\, \gamma$ Classification Scheme

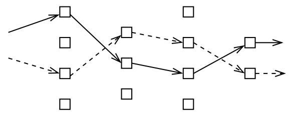
Machine Environment $\quad \boldsymbol{\alpha_1\alpha_2} \,|\, \beta_1 \dots \beta_{13} \,|\, \gamma$

- 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 ($\alpha_1 = X$)

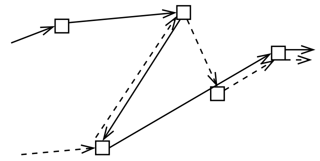
Single Machine



Flexible Flow Shop ($\alpha = FFc$)



Open, Job, Mixed Shop



The $\alpha \,|\, \beta \,|\, \gamma$ Classification Scheme

Job Characteristics $\quad \alpha_1\alpha_2 \,|\, \boldsymbol{\beta_1 \dots \beta_{13}} \,|\, \gamma$

- $\beta_1 = prmp$ 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

The $\alpha \,|\, \beta \,|\, \gamma$ Classification Scheme

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

- $\beta_8 = brkdwn$ 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

The $\alpha \mid \beta \mid \gamma$ 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 $U_j = \begin{cases} 1 & \text{if } C_j > d_j \\ 0 & \text{otherwise} \end{cases}$

The $\alpha \mid \beta \mid \gamma$ 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

The $\alpha \mid \beta \mid \gamma$ 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

Scheduling Tasks in a Central Processing Unit (CPU) [Ex. 1.1.3, textbook]

- 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

Gate Assignment at an Airport [Ex. 1.1.2, textbook]

- 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.

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 minimizes the given objective.

Classes of Schedules

Nondelay schedule

A feasible schedule is called nondelay if no machine is kept idle while an operation is waiting for processing.

There are optimal schedules that are nondelay for most models with regular objective function.

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.

There exists for $Jm||\gamma$ (γ regular) an optimal schedule that is active.
nondelay $\Rightarrow$ active
active $\not\Rightarrow$ nondelay

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.