Scheduling

Tim Nieberg
Research Institute for Discrete Mathematics

Course Information

Lecturer	Jun.-Prof. Dr. T. Nieberg
	room: 210
	e-mail: nieberg@or.uni-bonn.de
Secretary	D. Kijper
	tel.: 0228/73 8747

Information on the web:

- http://www.or.uni-bonn.de/lectures/ss10/ss10.html

Some slides used with friendly permission from J.L. Hurink (Universiteit Twente).

Material

- Pinedo, Michael L:
  Planning and Scheduling in Manufacturing and Services
  Series: Springer Series in Operations Research and Financial Engineering, second edition,2009.
- Brucker, Peter:
  Scheduling Algorithms 4th ed., 2004, Springer Verlag Berlin, Hardcover, ISBN: 3-540-20524-1
- Pinedo, Michael L:
  Scheduling: Theory, Algorithms, and Systems 2nd ed., 2002, Prentice Hall, ISBN 0-13-028138-7

Goals

main goals of the course 'Scheduling':

- get knowledge of basic model in scheduling theory
- get knowledge on basic solution techniques for models
- learn about application of scheduling models

What is Scheduling?

- decision making in manufacturing and service industries
- allocation of scare resources to tasks over time

Two main areas of application:

- manufacturing models
- service models

Remark: we only consider deterministic models

Planning of the subjects (tentative)

Lecture	Date	Subject
Lecture 1	04/13	Introduction
Lecture 2	04/20	Single machine models
Lecture 3	04/27	Single machine models
Lecture 4	05/04	Single machine models
Lecture 5	05/11	Parallel machine models
Lecture 6	05/18	Shop scheduling models
Lecture 7	06/01	Shop scheduling models
Lecture 8	06/08	Shop scheduling models
Lecture 9	06/15	Interval scheduling, Reservations and Timetabling
Lecture 10	06/22	Models in Transportation
Lecture 11	06/29	Models in Transportation
Lecture 12	07/06	On-Line Scheduling
Lecture 13	07/13	Wrap-Up
Lecture 14	07/20	slack

Examples: Paper Bag Factory

- factory producing paper bags for different goods
- raw material: rolls of paper
- 3-stage production process
  - printing the logo
  - gluing the sides of the bag
  - sewing one or both ends of the bag
- at each stage several machines for processing
- set of production orders specified by
  - quantity and type of bag
  - committed delivery date

Examples: Paper Bag Factory

- processing times proportional to the quantities
- late delivery leeds to a penalty, magnitude depends on
  - importance of the client
  - tardiness of the delivery
- switching on a machine from production of one bag-type to another, leads to setup time
- objectives:
  - minimize total penalty costs
  - minimize total time spent on setups

Examples: Routing and Scheduling of Airplanes

- airline has a fleet of different aircrafts
- given a set of flights characterized by
  - origin and destination
  - scheduled departure and arrival time
  - customers demand (predicted by the marketing department)
- assigning a particular type of aircraft to a specific flight leg leads to an estimated profit (based on demand)

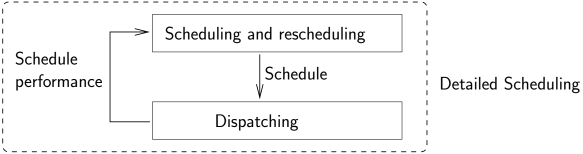
Examples: Routing and Scheduling of Airplanes

- problem: combine different flight legs to round-trips and assign an aircraft to them
- constraints:
  - turn around time at an airport
  - law regulation on duration of a crew duty
  - ...
- goal: maximize profit

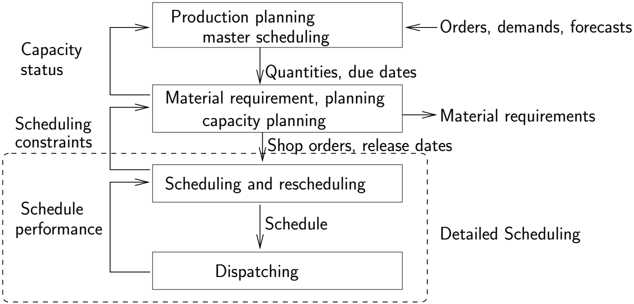
Scheduling Function in an Enterprise

- the scheduling function interacts with many other functions
- interactions are system-dependent
- often take place in an enterprise-wide information system; enterprise resource planning (ERP) system
- often scheduling is done interactively with a decision support system linked to the ERP system

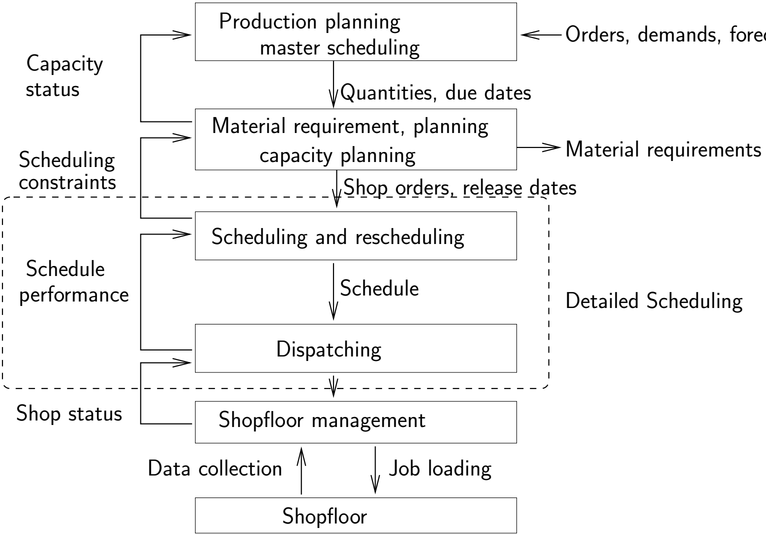
Scheduling in Manufacturing



Scheduling in Manufacturing

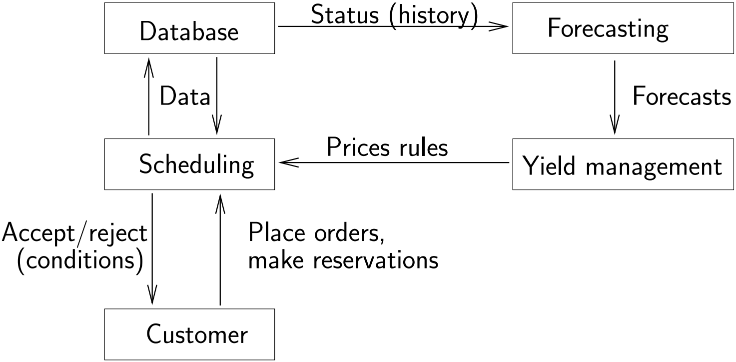


Scheduling in Manufacturing



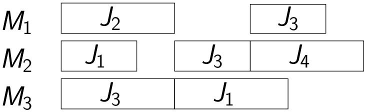
Scheduling in Services

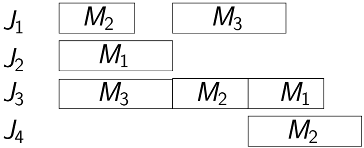
Remark: scheduling function in service organization is much more diverse than in manufacturing



Scheduling models (manufacturing)

- scheduling concerns optimal allocation or assignment of resources, over time, to a set of tasks or activities
  - m machines $M_1, \ldots, M_m$
  - n jobs $J_1, \ldots, J_n$
- schedule may be represented by Gantt charts





Classification of Scheduling Problems

General Notations:

- m machines $1, \ldots, m$
- n jobs $1, \ldots, n$
- (i, j) processing of job j on machine i (called an operation)
- data for jobs:
  - p_{ij}: processing time of operation (i, j)
  - if a job needs to be processed only on one machine or has only one operation:
    - p_j processing time of job j
  - r_j: release date of job j (earliest starting time)
  - d_j: due date of job j (committed completion time)
  - w_j: weight of job j (importance)

Classification of Scheduling Problems

(Many) Scheduling problems can be described by a three field notation $\alpha|\beta|\gamma$, where

- α describes the machine environment
- β describes the job characteristics, and
- γ describes the objective criterion to be minimized

Remark: A field may contain more than one entry but may also be empty.

Classification - Machine environment

- Single machine ($\alpha = 1$)
- Identical parallel machines ($\alpha = P$ or Pm)
  - m identical machines;
    if $\alpha = P$, the value m is part of the input and if $\alpha = Pm$, the value is considered as a constant (complexity theory)
  - each job consist of a single operation and this may be processed by any of the machines for p_j time units
- Uniform parallel machines ($\alpha = Q$ or Qm)
  - m parallel machines with different speeds $s_1, \ldots, s_m$
  - $p_{ij} = p_j / s_i$
  - each job has to be processed by one of the machines
  - if equal speeds: same situation as for identical machines

Classification - Machine environment

- Unrelated parallel machines ($\alpha = R$ or Rm)
  - m different machines in parallel
  - $p_{ij} = p_j / s_{ij}$, where s_{ij} is speed of job j on machine i
  - each job has to be processed by one of the machines
- Flow Shop ($\alpha = F$ or Fm)
  - m machines in series
  - each job has to be processed on each machine
  - all jobs follow the same route: first machine 1, then machine 2, etc
  - if the jobs have to be processed in the same order on all machines, we have a **permutation** flow shop

Classification - Machine environment

- Flexible Flow Shop ($\alpha = FF$ or FFm)
  - a flow shop with m stages in series
  - at each stage a number of machines in parallel
- Job Shop ($\alpha = J$ or Jm)
  - each job has its individual predetermined route to follow
  - a job does not have to be processed on each machine
  - if a job can visit machines more than once, this is called **recirculation** or **reentrance**
- Flexible Job Shop ($\alpha = FJ$ or FJm)
  - machines replaced by work centers with parallel identical machines

Classification - Machine environment

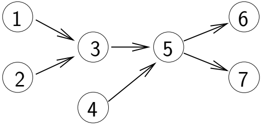
- Open Shop ($\alpha = O$ or Om)
    - each job has to be processed on each machine once
    - processing times may be 0
    - no routing restrictions (this is a scheduling decision)

Classification - Job characteristics

- release dates (r_j is contained in β field)
  - if r_j is not in β field, jobs may start at any time
  - if r_j is in β field, jobs may not start processing before their release date r_j
- preemption (*pmtn* or *prmp* is contained in β field)
  - processing of a job on a machine may be interrupted and resumed at a later time even on a different machine
  - the already processed amount is not lost
- unit processing times ($p_j = 1$ or $p_{ij} = 1$ in β field)
  - each job (operation) has unit processing times
- other 'obvious' specifications (e.g. $d_j = d$)

Classification - Job characteristics

- precedence constraints (*prec* in β field)
  - between jobs precedence relations are given: a job may not start its processing before another job has been finished
  - may be represented by an acyclic graph (vertices = jobs, arcs = precedence relations)



  - special forms of the precedences are possible: if the graph is a chain, intree or outtree, *prec* is replaced by *chain*, *intree* or *outtree*

Classification - Objective criterion

Notations:

- C_{ij}: completion time of operation (i, j)
- C_j: completion time of job j (= completion time of last operation)
- $L_j = C_j - d_j$: lateness of job j
- $T_j = \max\{C_j - d_j, 0\}$: tardiness of job j
- $U_j = \begin{cases} 1 & \text{if } C_j > d_j \\ 0 & \text{otherwise} \end{cases}$: unit penalty
- Note: due dates implicit in β field

Classification - Objective criterion

- Makespan ($\gamma = C_{max}$)
  - $C_{max} = \max\{C_1, \ldots, C_n\}$
- Maximum lateness ($\gamma = L_{max}$)
  - $L_{max} = \max\{L_1, \ldots, L_n\}$
- Total completion time ($\gamma = \sum C_j$)
  - can be used to measure the Work-In-Progress (WIP)
- Total weighted completion time ($\gamma = \sum w_j C_j$)
  - represents the weighted flow times of the jobs
- Total (weighted) tardiness ($\gamma = \sum (w_j) T_j$)
- (weighted) number of tardy jobs ($\gamma = \sum (w_j) U_j$)

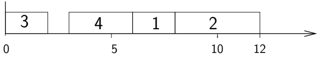
Remark: the mentioned classification gives only an overview of the basic models; lots of further extensions can be found in the literature!

Classification - Examples

- $1|r_j|C_{max}$
  - given: n jobs with processing times $p_1, \ldots, p_n$ and release dates $r_1, \ldots, r_n$
  - jobs have to be scheduled without preemption on one machine taking into account the earliest starting times of the jobs, such that the makespan is minimized
  - $n = 4$, $p = (2, 4, 2, 3)$, $r = (5, 4, 0, 3)$

Classification - Examples

- $1|r_j|C_{max}$
  - given: n jobs with processing times $p_1, \ldots, p_n$ and release dates $r_1, \ldots, r_n$
  - jobs have to be scheduled without preemption on one machine taking into account the earliest starting times of the jobs, such that the makespan is minimized
  - $n = 4,\ p = (2, 4, 2, 3),\ r = (5, 4, 0, 3)$



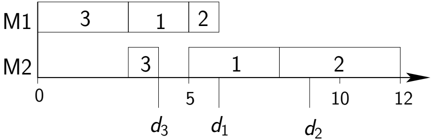
Feasible Schedule with $C_{max} = 12$ (schedule is optimal)

Classification - Examples

- $F2||\sum w_j T_j$
  - given n jobs with weights $w_1, \dots, w_n$ and due dates $d_1, \dots, d_n$
  - operations (i,j) with processing times $p_{ij},\ i = 1,2;\ j = 1, \dots, n$
  - jobs have to be scheduled on two machines such that operation $(2,j)$ is schedules on machine 2 and does not start before operation $(1,j)$, which is scheduled on machine 1, is finished and the total weighted tardiness is minimized
  - $n = 3,\ p = \begin{pmatrix} 2 & 1 & 3 \\ 3 & 4 & 1 \end{pmatrix},\ w = (3,1,5),\ d = (6,9,4)$

Classification - Examples

- $F2||\sum w_j T_j$
  - given n jobs with weights $w_1, \dots, w_n$ and due dates $d_1, \dots, d_n$
  - operations (i,j) with processing times $p_{ij},\ i = 1,2;\ j = 1, \dots, n$
  - jobs have to be scheduled on two machines such that operation $(2,j)$ is schedules on machine 2 and does not start before operation $(1,j)$, which is scheduled on machine 1, is finished and the total weighted tardiness is minimized
  - $n = 3,\ p = \begin{pmatrix} 2 & 1 & 3 \\ 3 & 4 & 1 \end{pmatrix},\ w = (3,1,5),\ d = (6,9,4)$



$$\sum w_j T_j = 3(8-6) + (12-9) \\ +5(4-4) = 9$$

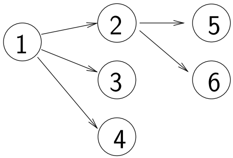
Classes of Schedules

- Nondelay Schedules:
  A feasible schedule is called a nondelay schedule if no machine is kept idle while a job/an operation is waiting for processing

  Example: $P3|prec|C_{max}$

  $n = 6$

  $p = (1, 1, 2, 2, 3, 3)$



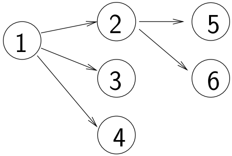
Classes of Schedules

- Nondelay Schedules:
  A feasible schedule is called a nondelay schedule if no machine is kept idle while a job/an operation is waiting for processing

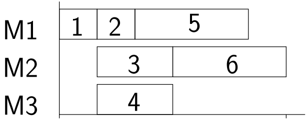
Example: $P3|prec|C_{max}$

$n = 6$

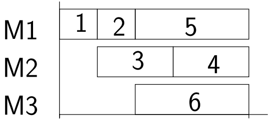
$p = (1, 1, 2, 2, 3, 3)$



Best nondelay:



Optimal



Classes of Schedules

Remark: restricted to non preemptive schedules

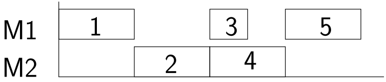
- Active Schedules:
  A feasible schedule is called active if it is not possible to construct another schedule by changing the order of processing on the machines and having at least one job/operation finishing earlier and no job/operation finishing later.
- Semi-Active Schedules:
  A feasible schedule is called semi-active if no job/operation can be finishing earlier without changing the order of processing on any one of the machines.

Classes of Schedules

Examples of (semi)-active schedules:

Prec: $1 \longrightarrow 2;\ 2 \longrightarrow 3$

not semi-active

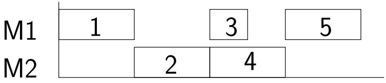


Classes of Schedules

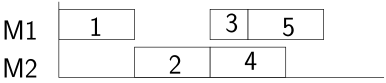
Examples of (semi)-active schedules:

Prec: $1 \longrightarrow 2;\ 2 \longrightarrow 3$

not semi-active



semi-active

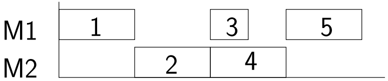


Classes of Schedules

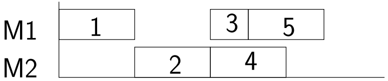
Examples of (semi)-active schedules:

Prec: $1 \longrightarrow 2;\ 2 \longrightarrow 3$

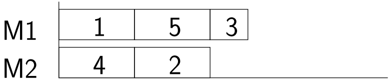
not semi-active



semi-active



active

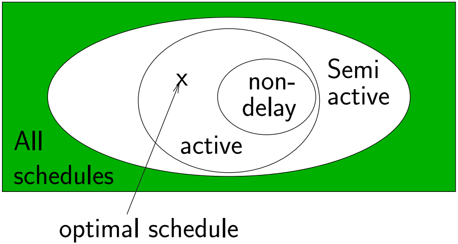


Classes of Schedules

<u>Properties</u>:

- every nonpreemptive nondelay schedule is active
- every active schedule is semiactive
- if the objective criterion is regular, the set of active schedules contains an optimal schedule (regular = non decreasing with respect to the completion times)

<u>Summary</u>:



Research topics for Scheduling

- determine border line between polynomially solvable and NP-hard models
- for polynomially solvable models
  - find the most efficient solution method (low complexity)
- for NP-hard models
  - develop enumerative methods (DP, branch and bound, branch and cut, ...)
  - develop heuristic approached (priority based, local search, ...)
  - consider approximation methods (with quality guarantee)