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 need 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.

- 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

- 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

- 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

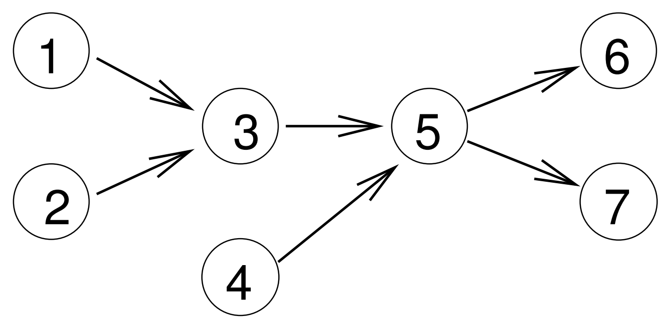
Classification - Machine environment

- 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 (i.e. $d_j = d$)

- 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

<u>Notations</u>:

- 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

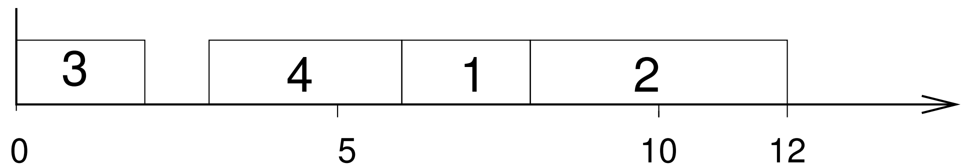
Classification - Objective criterion

- 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 ($\gamma = \sum (w_j) U_j$)

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

- $1|r_j|C_{max}$
  - 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
  - $n = 4$, $p = (2, 4, 2, 3)$, $r = (5, 4, 0, 3)$

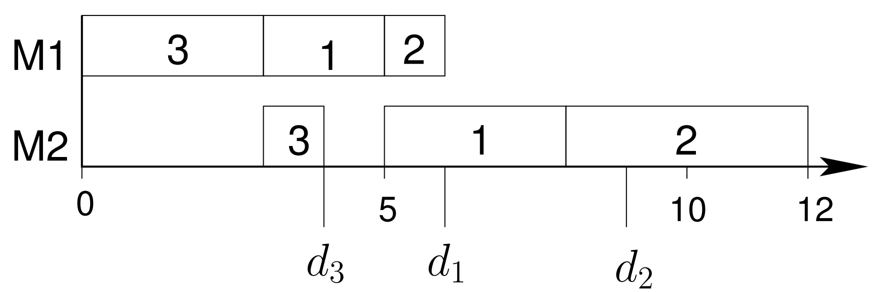
- $1|r_j|C_{max}$
  - given: n jobs with processing times $p_1, \ldots, p_n$ and release dates $r_1, \ldots, r_n$
  - jobs have to be scheduled without preemption on one machine taking into account the earliest starting times of the jobs, such that the makespan is minimized
  - $n = 4$, $p = (2, 4, 2,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



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

- $F2||\sum w_jT_j$
  - given n jobs with weights $w_1, \ldots, w_n$ and due dates $d_1, \ldots, d_n$
  - operations (i, j) with processing times $p_{ij},\ i = 1, 2;\ j = 1, \ldots, n$
  - jobs have to be scheduled on two machines such that operation $(2, j)$ is scheduled 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
  - $n = 3,\ p = \begin{pmatrix} 2 & 1 & 3 \\ 3 & 4 & 1 \end{pmatrix},\ w = (3, 1, 5),\ d = (6, 9, 4)$

- $F2||\sum w_j T_j$
  - given n jobs with weights $w_1, \ldots, w_n$ and due dates $d_1, \ldots, d_n$
  - operations (i, j) with processing times $p_{ij},\ i = 1, 2;\ j = 1, \ldots, n$
  - 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
  - $n = 3,\ p = \begin{pmatrix} 2 & 1 & 3 \\ 3 & 4 & 1 \end{pmatrix},\ w = (3, 1, 5),\ d = (6, 9, 4)$



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