Introduction to production scheduling

Industrial Management Group
School of Engineering
University of Seville

Introduction to production scheduling

- Scheduling
- Production scheduling
- Gantt Chart
- Scheduling environment
- Constraints
- Scheduling objectives

Scheduling (I)

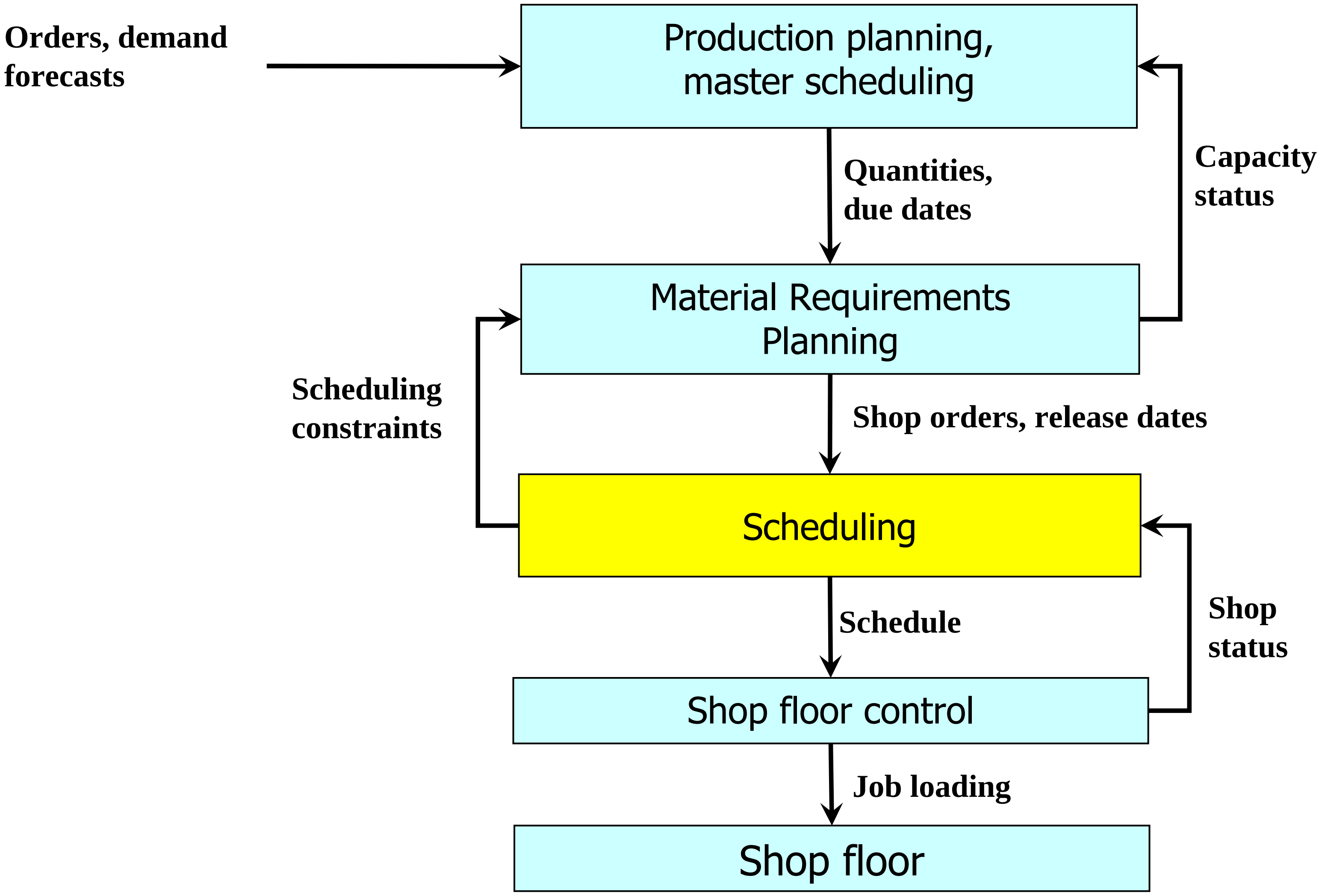
- Definition
  - Scheduling deals with the efficient allocation of tasks over resources
  - The general scheduling problem is, given a number of tasks and a number of resources, set the dates when each task should be accomplished on each resource

- We are interested in scheduling in the manufacturing context, although it has many applications in other fields

Scheduling (II)

- Real-life examples
  - Timetabling
    - Scheduling of the PC room at Duisburg University: a number of different courses (tasks) have to be given using the PC room (resource)
  - Workforce scheduling
    - Assign shifts for nurses and doctors in a Hospital
  - Sports scheduling

Production Scheduling (I)



Production Scheduling (II)

- The MRP tell us the quantities of products to manufacture in every time bucket
  - However, MRP does not make any assumption about the resources (i.e. labour, machines) currently available in the factory
    - E.g. two different components have to be manufactured in the same section. How to schedule them?
- Therefore, we have a number of jobs (j) to be manufactured over a number of machines (i)
  - The production scheduling problem deals with obtaining the date for each job to enter on each machine
  - Not necessarily physical machines, they may be stages (consisting on several machines or labour) in a manufacturing process

Production Scheduling (III)

- Jobs have to be manufactured in each machine in a certain order (known as route) during a certain time period (known as processing time)
  - Processing time of job j in machine i is usually denoted by $p_{i,j}$
- Both route and time period are given by the technological process

Example1: Computer assembly (I)

- Tomcat Ltd. is a company that assembles computers. Three main steps can be distinguished in this process:
  - Motherboard & microprocessor are installed
  - Peripheral devices are plugged in the motherboard
    - The number and type of devices that have been order by each customer
  - The computer (all its components) are tested
    - This process depends on the number and type of components
- The route through the steps is given by the technological process and does not depend on the specific order of the customer

Example1: Computer assembly (II)

- The plant in which Tomcat Ltd. assembles the computers is organised in three sections, according to the three main steps:
  - Section 1: Motherboard & microprocessor
  - Section 2: Peripheral devices
  - Section 3: Computer test

- On each section, one worker is performing the corresponding operation
  - Obviously, a new order cannot start until the worker completes the current order
  - Let us assume that an order cannot overtake another order, i.e. the job sequence is the same for all steps

Example1: Computer assembly (III)

- Let us assume that we have three orders (computers to manufacture).
  - According to the nature of each order (components, type, etc.), we can have some estimate of the average times (minutes) for each step:

	Motherboard	Devices	Test
Order #1	2	15	7
Order #2	3	10	5
Order #3	2	12	5

- The objective of the company is to keep the average time to assembly a computer as lowest as possible

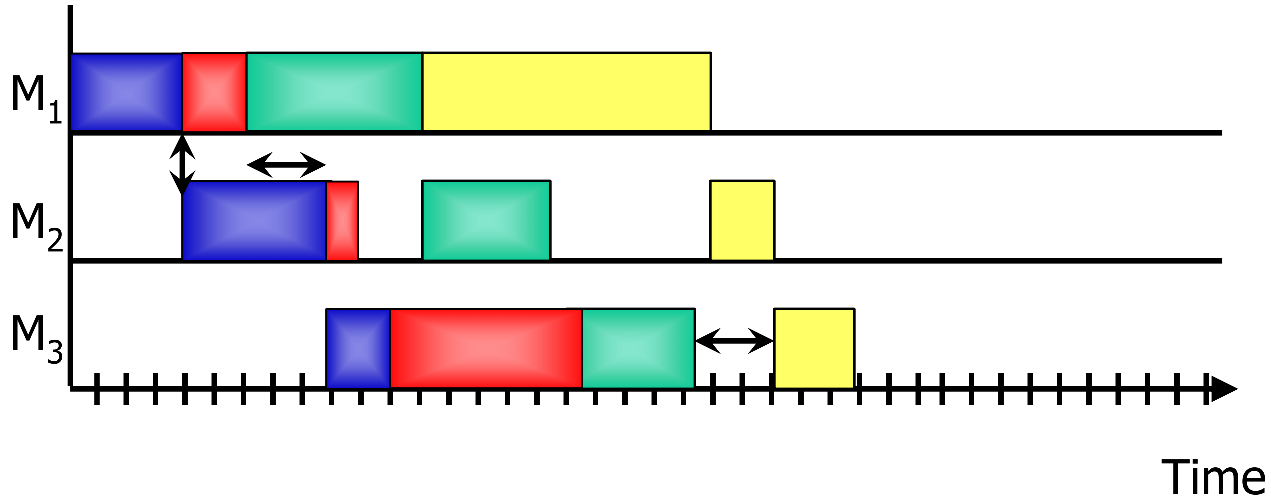
Example1: Computer assembly (IV)

- Which of the following sequences is the most convenient for the objective of the company:
  - #1, #2, #3?
  - #3, #2, #1?
  - Or the order of the jobs is not relevant for the final result?

	Motherboard	**Devices**	**Test**
Order #1	2	15	7
Order #2	3	10	5
Order #3	2	12	5

Gantt chart

- A representation of an specific solution of a scheduling problem in terms of the machines and jobs
  - Eg. job1, job2, job3, job4
  - Sequence [1,2,3,4]



Example2: Representation of solutions (I)

- Using the data of the Example1, try to represent the sequences [1,2,3] and [3,2,1] by a Gantt chart
- job1 , job2 , job3

	Motherboard	Devices	Test
Order #1	2	15	7
Order #2	3	10	5
Order #3	2	12	5

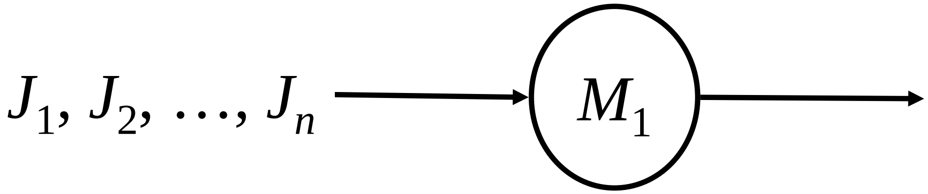
Solution of Example2

Scheduling environment (I)

- The environment or framework of a scheduling problem refers to the way the jobs must visit the machines:
  - Single machine
  - Parallel machines
  - Flow shops
  - Flexible flow shops
  - Job shops
  - Flexible job shops
  - Open shop
- The environment largely determines the difficulty of the scheduling problem

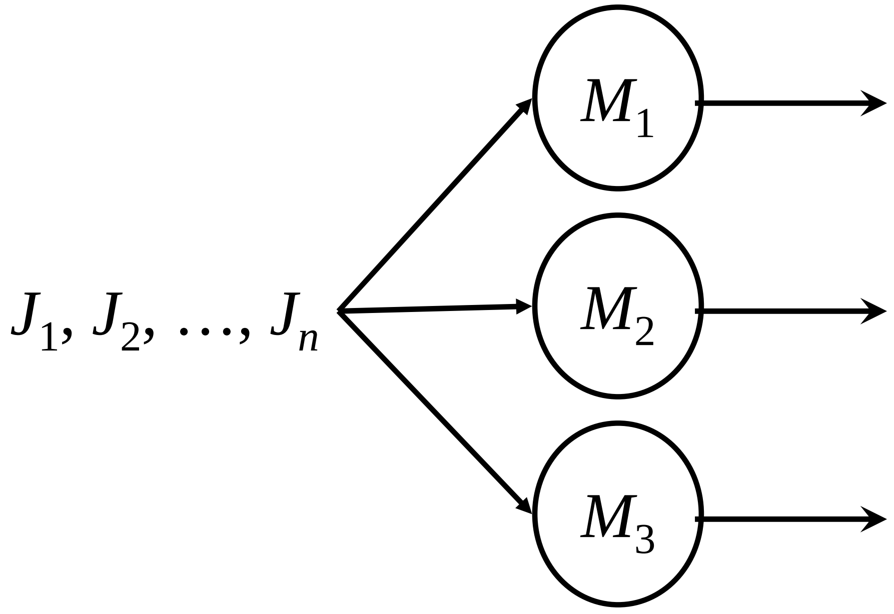
Scheduling environment (II)

- Single machine
  - The simplest of all machine environments
  - One may reduce the different steps (sections) in the plant to a single machine
  - Interesting case: bottleneck process, the important issue is scheduling jobs in the bottleneck



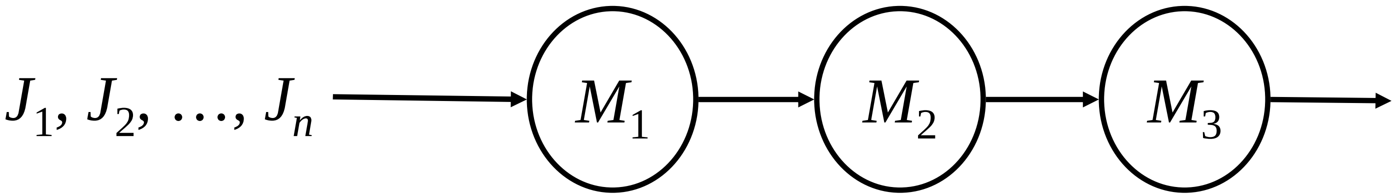
Scheduling environment (III)

- Parallel machines
  - All machines are identical
  - A job can be processed on any machine
  - Generalization of the single machine
  - Special case of the flexible flow shop and flexible job shop



Scheduling environment (IV)

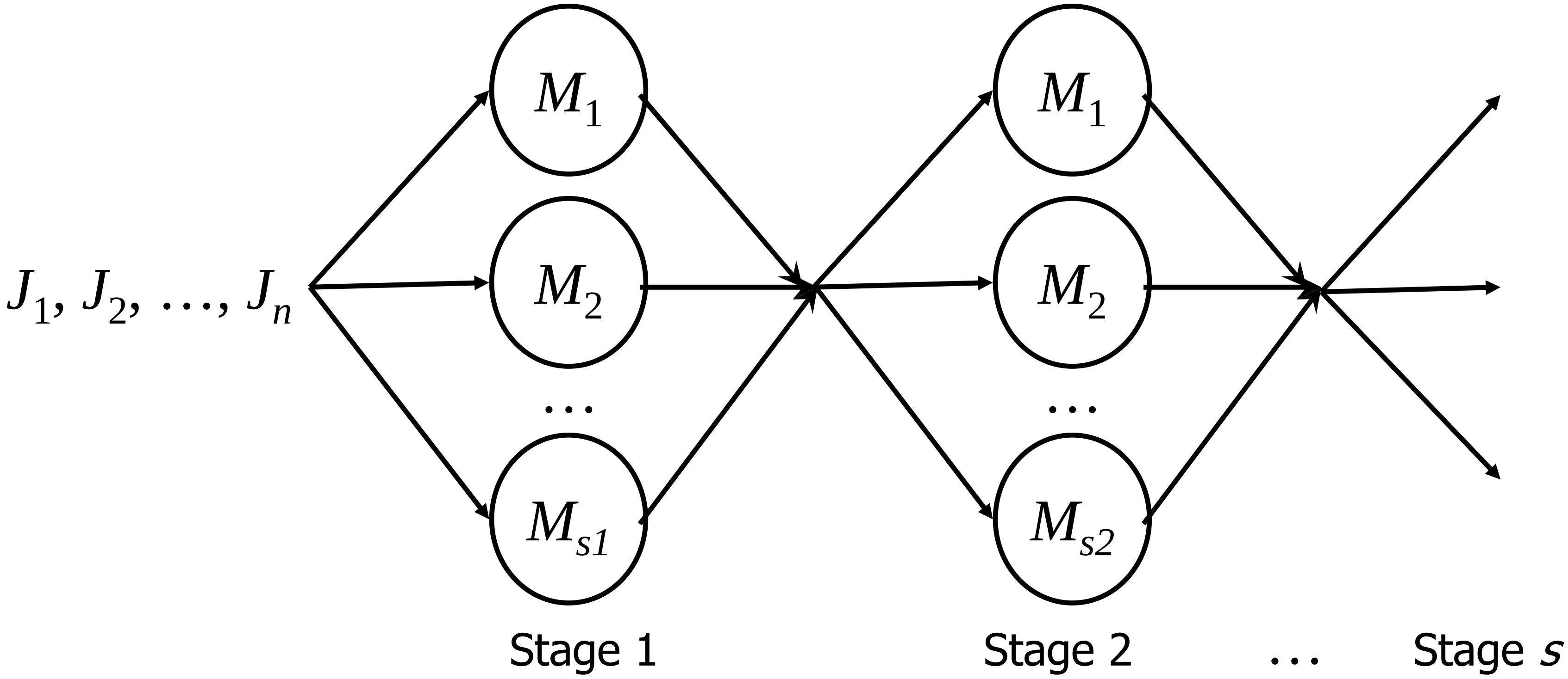
- Flow shops
  - All jobs have the same routing



  - Additionally, most of the times it is consider than the sequence is the same for all machines
    - Permutation flow shop, e.g. in Tomcat Ltd

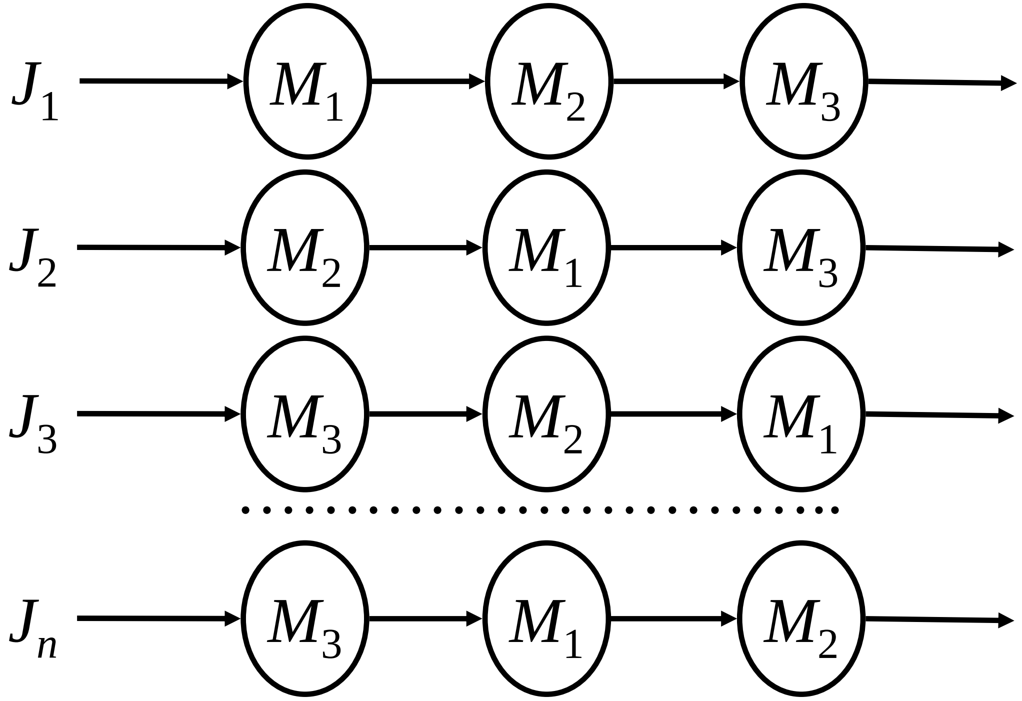
Scheduling environment (V)

- Flexible flow shop
  - s stages with m_s machines in parallel
  - A job can be processed on any machine at each stage
  - Generalization of parallel machines and flow shop



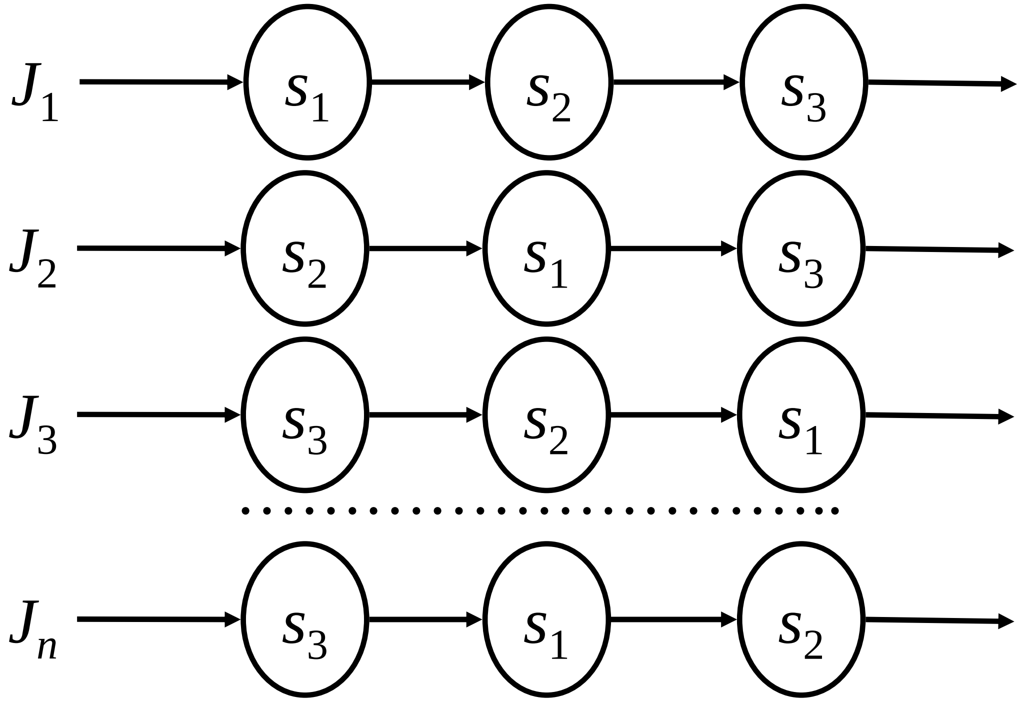
Scheduling environment (VI)

- Job shops
  - Each job has, in general, a different route to be processed by the machines
  - It is one of the most complex cases



Scheduling environment (VII)

- Flexible Job shops
  - Each job has, in general, a different route to be processed at all stages
  - Each stage has m_s machines in parallel
  - It is even more complex than the job shop



Constraints

- There may be additional constraints for the scheduling problem:
  - Release dates
  - Setup-times
  - Pre-emption
  - Precedence constraints
  - Blocking
  - No-wait
  - ....

Scheduling objectives (I)

- A scheduling objective is a measure to evaluate the quality of certain schedule
  - In real-life situations, there are many (sometimes conflicting) objectives
- There are based on the completion times:
  - $C_{i,j}$: Time in which job j is finished in machine I
  - C_j: Time in which the job j is finished in the last machine
- It is easy to see that, for the flow shop case, the completion time for a job in the position $[k]$ is:

$$C_{[k],j} = \max(C_{[k-1],j}\,;\, C_{[k],j-1}) + p_{[k],j}$$

assuming $C_{[k],0} = 0$, and $C_{[0],j} = 0$

Scheduling objectives (II)

- Rather often, not all jobs (customers) are equally important
  - Therefore, one can assign a weight w_j to each job representing the relative importance of each job
- In general, one can distinguish two types of objectives:
  - Due date related objectives
  - Non-due date related objectives

Scheduling objectives (III)

Due date related objectives (I)

- For this kind of problems, we assume that each job j has, in general, a due date d_j and a release date r_j
  - The due date represents the commitment of the company with a customer
  - The release date implies the non availability of raw materials from the beginning
- When each job has a due date, a basic objective is fulfilling this due date
  - Indicator of the service level
  - However, finishing the order as soon as possible (much before the due date) is not a good idea
    - Inventory costs

Scheduling objectives (IV)

Due date related objectives (II)

- Lateness of a job: $L_j = C_j - d_j$
  - Maximum lateness: $L_{max} = max(L_j)$
  - Average (total) lateness: $\underline{L} = \Sigma L_{ij} / n$
  - Weighted lateness: $wL = \Sigma w_j L_j$
- Tardiness of a job: $T_j = max(0, L_j)$
  - Maximum tardiness $T_{max} = max(T_j)$
  - Average (total) tardiness: $\underline{T} = \Sigma T_j / n$
  - Weighted tardiness: $wT = \Sigma w_j T_j$
  - Number of tardy jobs: $U = \Sigma U_j$; $U_j = 0$ if $T_j = 0$, $U_j = 1$ if $T_j \neq 0$
- Earliness of a job: $E_j = max(0, - L_j)$
  - Maximum earliness $E_{max} = max(E_j)$
  - Average (total) earliness: $\underline{E} = \Sigma E_j / n$
  - Weighted earliness: $wE = \Sigma w_j E_j$

Scheduling objectives (V)

Non due date objectives

- Machine utilisation
  - Makespan: $C_{max} = max(C_j)$
  - Idle time
    - Time after finishing one job and before starting the next one
    - It can be shown that minimising makespan is equivalent to minimising idle time
- Average lead time
  - Average (total) completion time: ΣC_j
  - Average weighted completion times $\Sigma w_j C_j$

Break