INTRODUCTION TO THE SCHEDULING THEORY

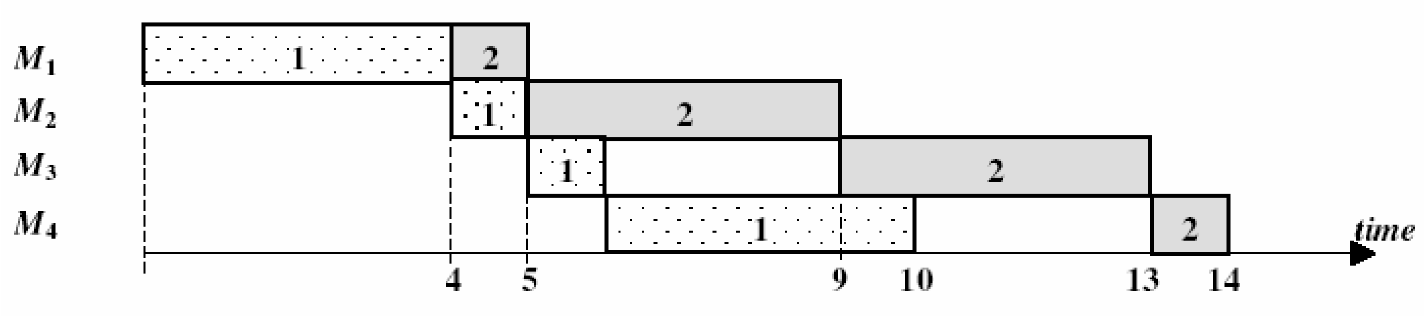
- ***Scheduling concerns optimal allocation or assignment of resources, over time, to a set of tasks or activities.***
- ***The allocation of resources over time to perform a collection of tasks' (Baker 1974)***

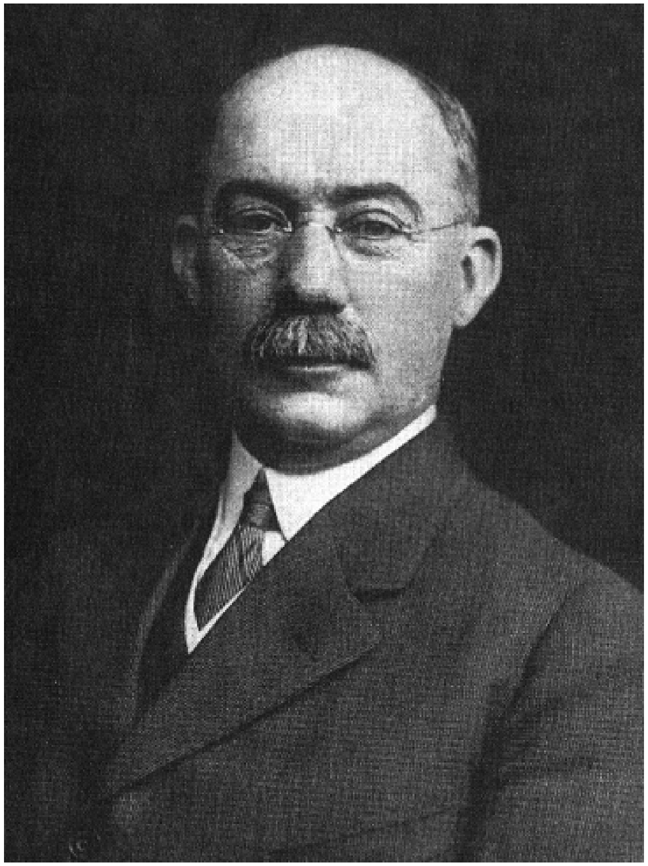
- As an independent branch of Operational Research, Scheduling Theory appeared in the beginning of the 50s. Scheduling theory can be applied to many areas including
  - computer systems
  - manufacturing
  - agriculture
  - health care
  - transport
  - materials handling
  - logistics and SC in general
  - …
- Scheduling deals with the problems of optimal arrangement, sequencing and timetabling.
- Scheduling is a decision-making process of allocating ***limited resources*** to ***activities (jobs)*** over time.

Resources:
- machines at a workshop
- cranes to ships in container port
- runways at an airport,
- crews at transportation means
- handling equipment to tasks
- ....

***Tasks (jobs)*:**
- operations in a workshop,
- loading unloading ships
- takeoffs and landings,
- stages at a construction project,
- internal transportation, loading unloading
- ....

- **A schedule is a job sequence determined for every resource of the processing system**.
- *Standard scheduling requirements say:*
  - *a job cannot be processed by two or more machines at a time,*
  - *a machine cannot process two or more jobs at the same time.*
- Depending on the type of scheduling system, specific constraints should be satisfied (jobs may be released at different times, there may be allowed preemption of jobs by other jobs, etc.)
- Gantt chart is a horizontal bar chart that graphically displays the time relationships between the different tasks in a project:
  - the X-axis represents the time,
  - the Y-axis represents machines,
  - a color and/or pattern code may be used to indicate operations of the same job.





Henry Laurence Gantt (1861 - 1919) was an industrial engineer and a disciple of Frederick W. Taylor. He developed his now famous charts during World War I to compare production schedules with their realizations. Gantt discussed the underlying principles in his paper "Efficiency and Democracy," which he presented at the annual meeting of the American Society ofMechanical Engineers in 1918. The Gantt charts currently in use are typically a simplification of the originals, both in purpose and in design.

- Scheduling theory covers more than **10,000 different models**. The models are specified according to three-field classification $\alpha|\beta|\gamma$ where
  - α specifies the machine environment,
  - β specifies the job characteristics, and
  - γ determines the optimality criterion.

➢ MACHINE ENVIRONMENT

- Single stage systems
  - ✓ If there is a single machine, each job should be processed by that machine exactly once.
  - ✓ If there are several parallel machines each job can be processed by any machine.
- Multistage systems
  - ✓ Each job should be processed on each machine from the set. All machines are different.

➢ JOB ENVIRONMENT

- There are n jobs N={1,… ,n}.

Processing time of job ***j*** on machine ***i*** is p_{ij}. If there is a single machine, then processing time of job ***j*** does not depend on machine number and it is denoted by p_j.
For job ***j*** there may be given also
rj – release time (the time the job arrives at the system),
dj – due date (the time the job is promised to the customer),
wj – weight (the importance of job ***j***).

➢ OPTIMALITY CRITERION

- The schedule can be characterised by starting or completion times of all operations of the jobs.
- The objective is to construct a schedule, that minimizes a given objective function F. Usually function F depends on job completion times C_j, j=1,…,n, where C_j is the completion time of the last operation of job j.

- The most common objective functions are

Makespan	$C_{\max} = \max\{C_j \mid j=1,\ldots,n\}$
Total completion time	$\sum C_j = \sum_{j=1}^{n} C_j$
Total weighted completion time	$\sum w_j C_j = \sum_{j=1}^{n} w_j C_j$

- Other objective functions depend on due dates dj. We define for each job j

$$L_j = C_j - d_j \quad \text{- lateness of job } j,$$
$$E_j = \max\{0, d_j - C_j\} \quad \text{- earliness,}$$
$$T_j = \max\{0, C_j - d_j\} \quad \text{- tardiness,}$$
$$U_j = \begin{cases} 0 & \text{if } C_j \le d_j \\ 1 & \text{otherwise} \end{cases} \quad \text{- unit penalty.}$$

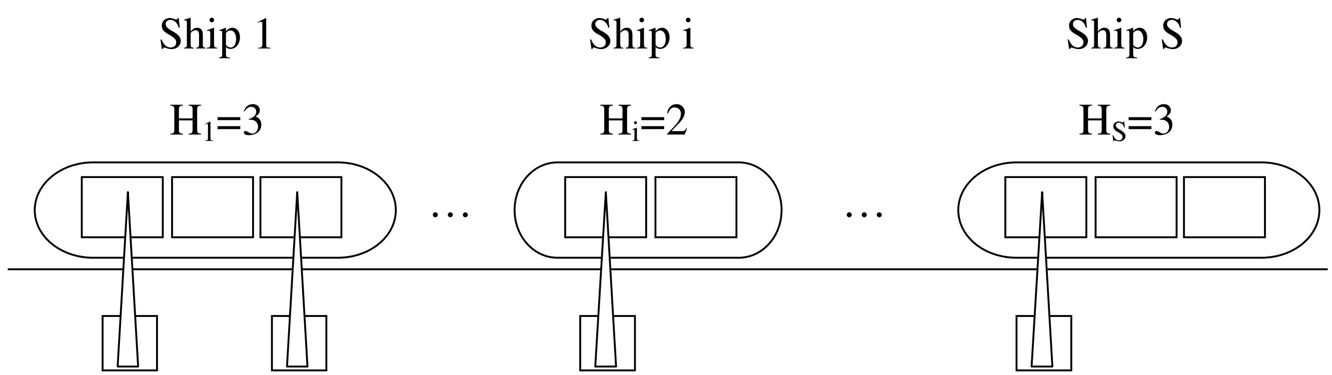
- The corresponding objective functions can be defined as follows:

Maximum lateness	$L_{\max} = \max\{L_j \mid j=1,\ldots,n\}$
Total (weighted) tardiness	$\sum_{j=1}^{n} T_j \; (\sum_{j=1}^{n} w_j T_j)$
Total (weighted) number of late jobs	$\sum_{j=1}^{n} U_j \; (\sum_{j=1}^{n} w_j U_j)$

THE CRANE SCHEDULING PROBLEM (DAGANZO 1989)

as a manufacturing job environment scheduling problem where single task jobs (ship holds) should be handled by a set of identical machines (cranes)

- We consider a berth of fixed length with a fixed number of cranes, **C**, serving a variety of ships. The sizes of ships are described by the number of holds they have; $\mathbf{H_i}$ shall denote the number of holds for ship **i**.
- All holds are assumed to take roughly equivalent berth length. Thus, berth length can be expressed by the number of holds that can be served simultaneously, **B**. In other words, we assume that the berth can be visualized as being divided into **B** identical “slots”; thus, ship **i** needs $\mathbf{H_i}$ slots to be handled



- Time lost while cranes are moving is ignored
- If ship has only one hold it can depart as soon as job is completed. Then, if two or more cranes are not allowed to work on hold simultaneously, the problem is classical open shop, parallel machine scheduling problem.

simple solutions to this problem exist (see, for example, the related works by McNaughton, 1959

- When ships have multiple holds, a job cannot leave until all jobs on the same ship have been completed, which delays the departures and complicates the analysis. This problem can be modeled as a resource constrained project scheduling problem with precedence constraints.

3.2 *Practical considerations*

This subsection discusses approaches that can be taken when a mathematical programming solution is not effective. It is divided in three parts: a simple case that can be solved exactly, a discussion of some scheduling principles leading to a heuristic procedure, and the results of some numerical tests.

3.2.1 *Ships with open decks.* This section applies to ships with no holds and cranes that can all be allocated to the same ship. The situation is not prevalent, but may arise in dry or liquid bulk terminals with just a few loaders (cranes). It is also assumed that $\tau_i = 0$ for all ships. We use t_i to denote the total work time required by ship i. Under these conditions, the following strategy is optimal (see Appendix B):

(i) Rank ships in decreasing order of cost to workload ratio, c_i/t_i, so that: $c_1/t_1 \geq c_2/t_2 \geq \ldots$.

(ii) Allocate all cranes to ship 1 until it is served, then to ship 2, etc., until all ships have been served.

3.2.2 *Crane scheduling principles.* When either $\tau_i \neq 0$ or ships have multiple holds, a simple exact solution could not be found. Nevertheless, optimal solutions have properties that lead to useful scheduling principles.

Principle 1. A crane should not be idle if there is some work that it can do.

Principle 2. If cranes work on a ship, at least one of them should work on the "maximum hatch."

Principle 2 can be generalized to cases when more than one crane can work on a hatch. Then, as many cranes as possible should work on the maximum hatch.

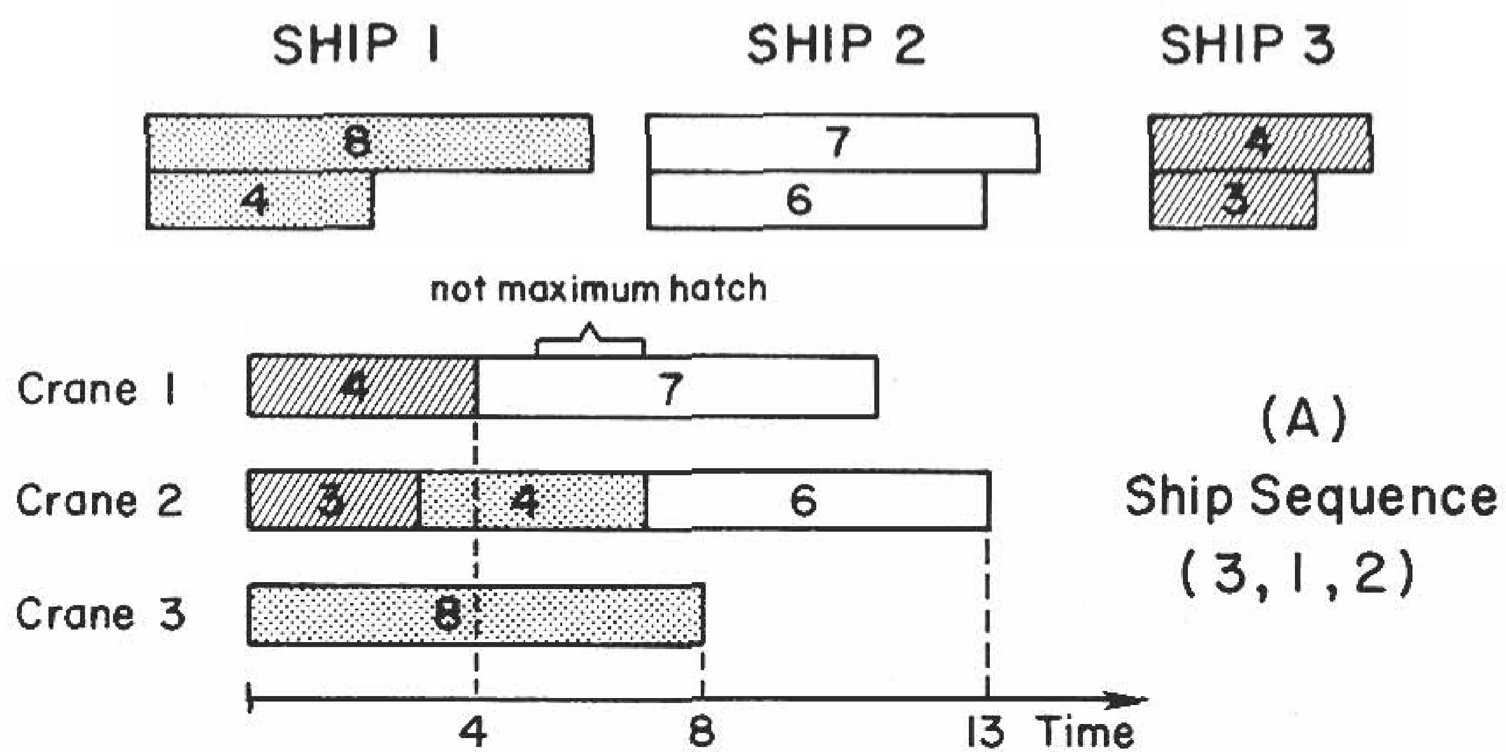
These principles, coupled with the overall goal of serving lightly loaded, expensive ships first, suggest the following scheduling strategy:

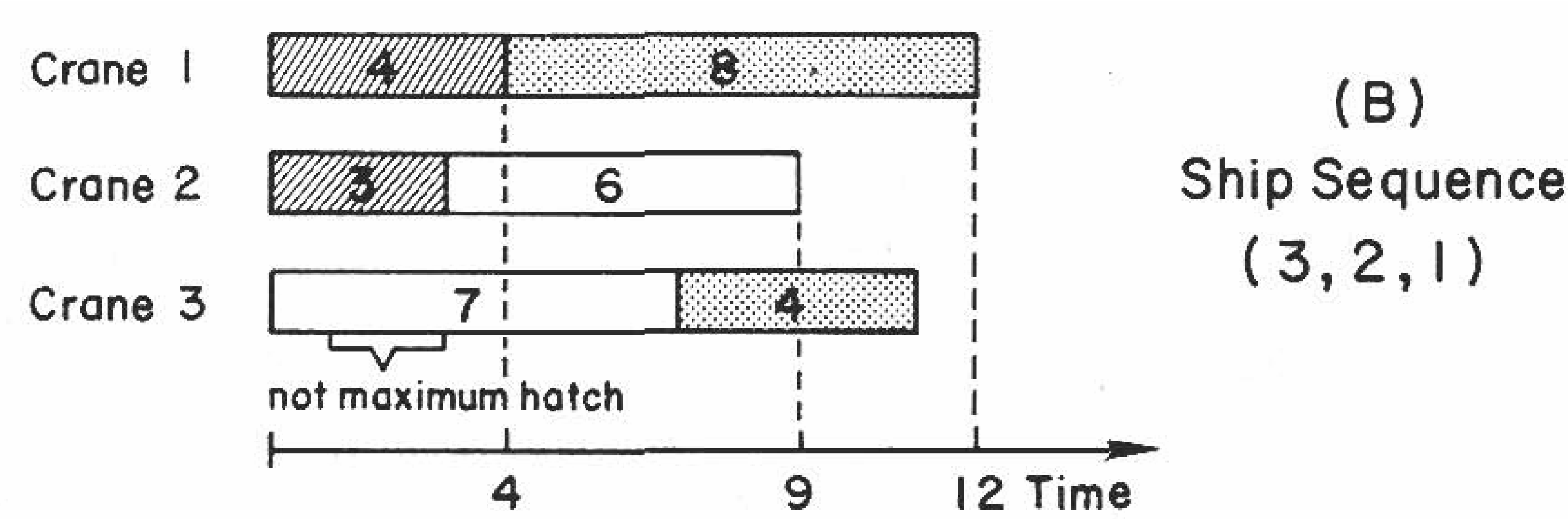
(i) Rank ships in a desired service order, and use this order to rank all the holds. Rank holds by ship first, and then within a ship by the amount of work required.

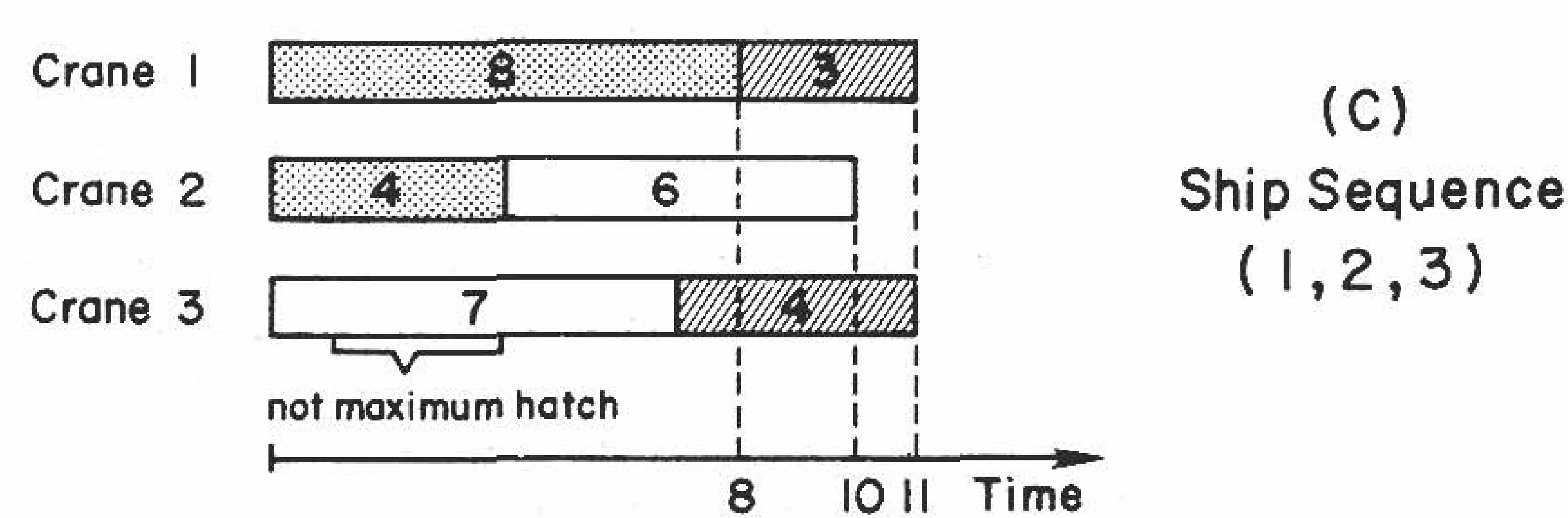
(ii) Allocate all the holds to cranes in the order implied by the above step. Always choose the crane that is next to become available:

The step (i) ranking should reflect the ship's costs and their loading patterns. For realistic problems, only a few ship sequences should be reasonable; they all can be tried. For example, if ships are not highly unbalanced, it may make sense to rank them according to the ratio of "cost" to "total work needed," and to explore minor perturbances to this ordering.

Figure 1 shows by means of bar diagrams (Gantt charts) the result of the strategy




Crane 1
Crane 2
Crane 3
4
8
3
6
7
4
not maximum hatch
4
9
12 Time
(B)
Ship Sequence
(3,2,1)


Crane 1
Crane 2
Crane 3
8
3
4
6
7
4
not maximum hatch
8
10 11 Time
(C)
Ship Sequence
(1,2,3)