

---

# Project Management

## Network Diagrams

Prof. Mauro Mancini

e-mail: [Mauro.Mancini@polimi.it](mailto:Mauro.Mancini@polimi.it)

tel.: +39-02-23994057

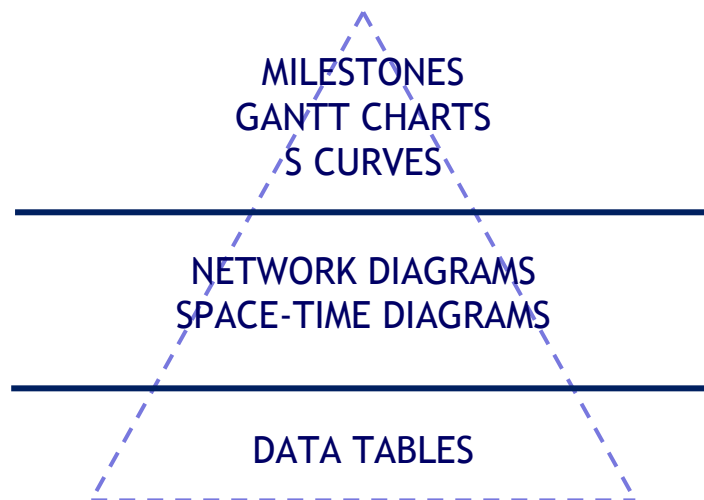
POLITECNICO DI MILANO

Department of Management, Economics and Industrial Engineering

Mauro Mancini ©

## Planning's Instruments

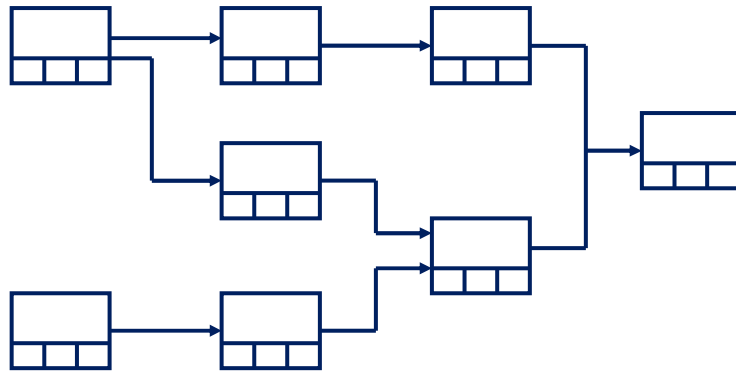
---



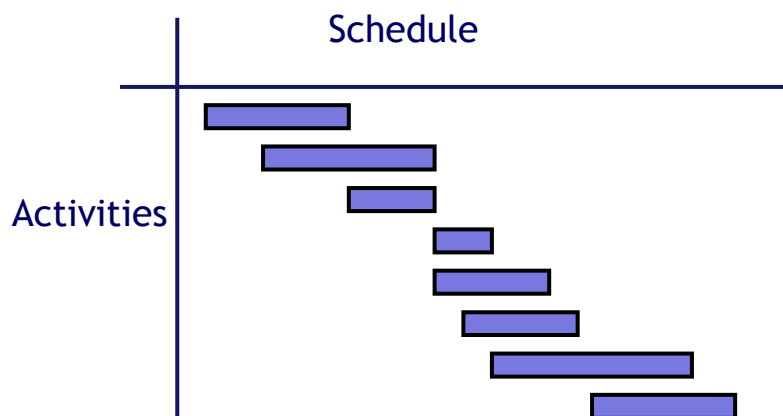
Mauro Mancini ©

2

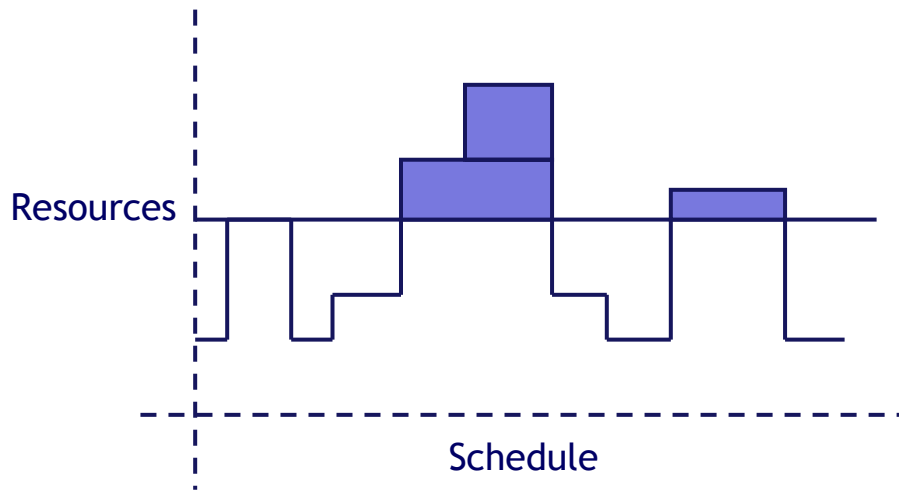
## Precedence Chart



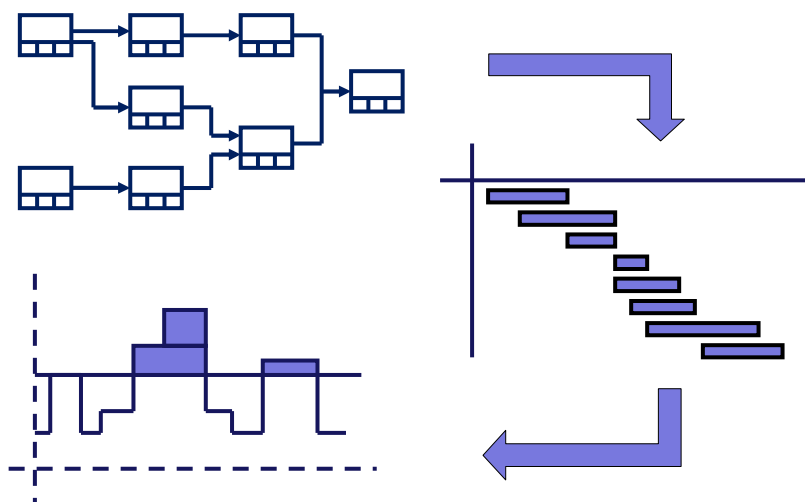
## Gantt Chart



## Resources Scheduling



## Pattern



## Gantt Chart

Gantt Chart shows when the project and each activity start and end against a horizontal timescale (bar chart)

### Steps:

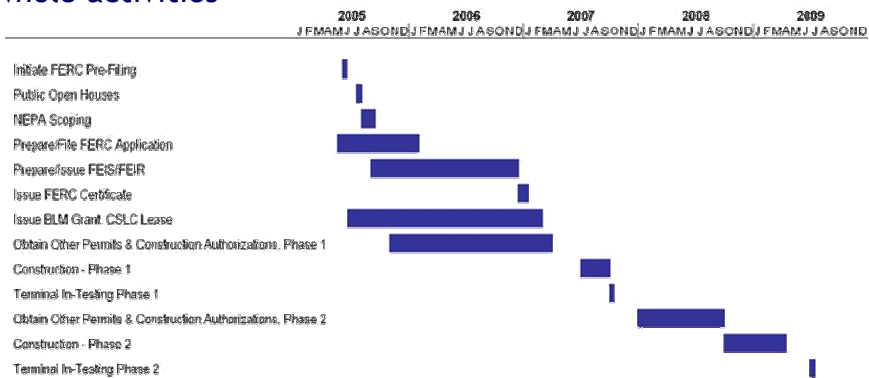
- Activities have to be sequenced
- Activity's duration has to be estimated
- Gantt Chart can be drafted and refined

## Gantt Chart Advantages

Easy elaboration

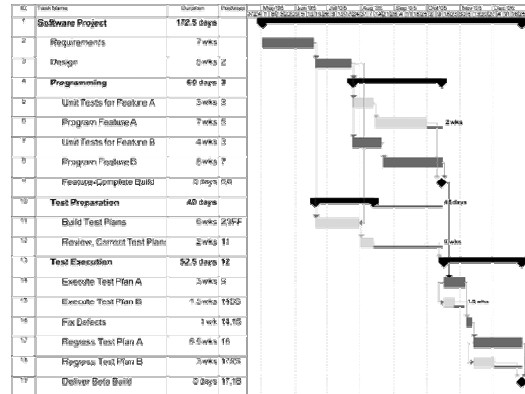
Direct graphics

Possibility to easily evaluate the resources requested by the whole activities



## Gantt Chart Limitations

Suitable for projects composed by few activities  
It (*usually*) doesn't indicate the logical relationships between activities



## Network Diagrams

Unlike the Gantt chart, the network diagrams show a systemic project representation. In this way they allow to define the project like the whole linked activities oriented to a unique objective.



## Network Diagrams

---

They allow to link together activities (or work package) making possible:

- to give a inter-functional visibility
- to individualize the critical activities
- to simulate the impact from time and cost variations or, more generally, from eventual problems or corrective actions

## Graphs

---

A graph is a set of nodes linked by arrows, their sequence define a path.

If the arrows have a sense, the graph is called oriented.

## Graphs types

---

Without circuits

With circuits

With certain paths

With uncertain paths

With “AND nodes”

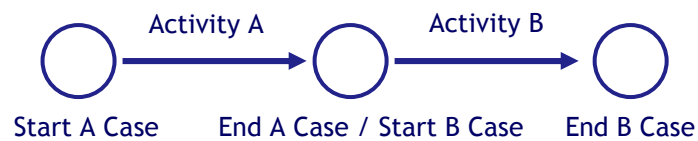
With “OR nodes”

With probabilistic node

## Graphs types

---

### Activity on arrow

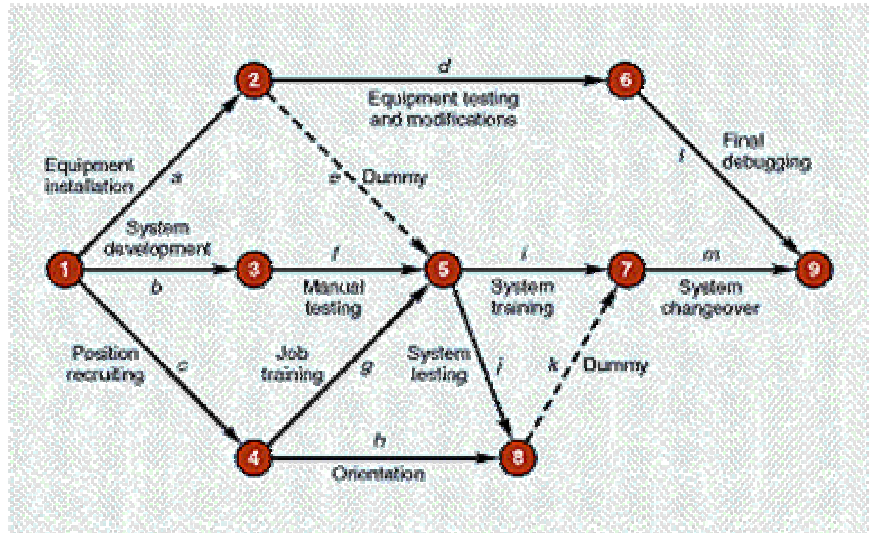


### Activity on node



Activity A - Activity B  
Connection

## Example 1 - Activity on arrow



## AOA Networks

FINISH TO START links

Dummy activities requested

The dummy activities are not characterized by duration or resources' use. They only establish precedence constraints between the activities

## AON Networks

### Link types

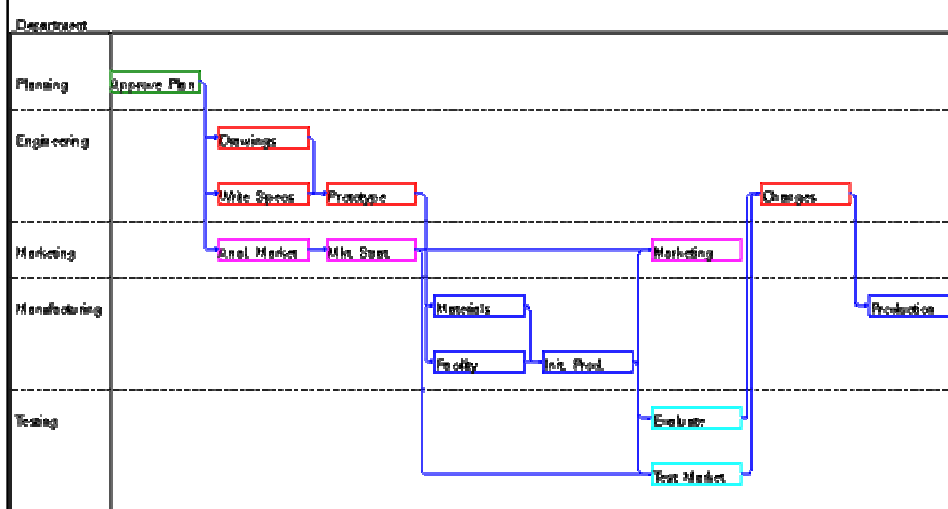
- Finish to Start
- Start to Start
- Finish to Finish
- Start to Finish

Each activity can have more than one “input and output” link

Between two activities is possible to have two different types of link

Dummy events (milestones) can be used

## Example 2 - Activity on node

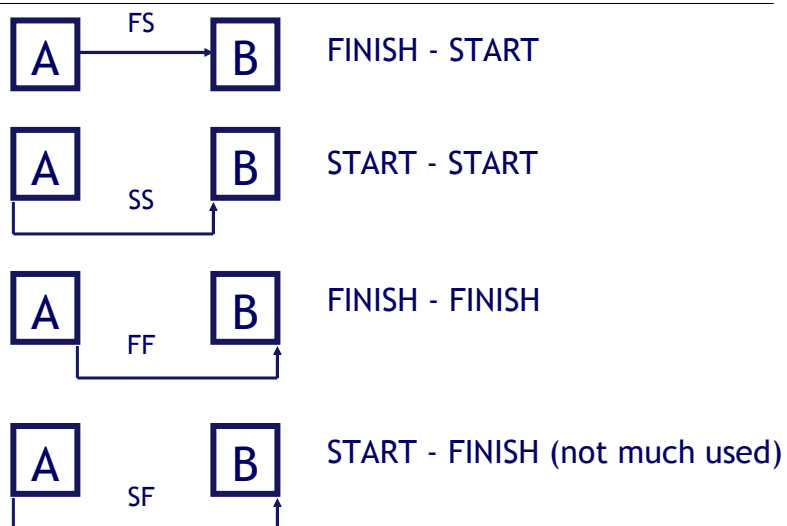


## Types of Graphs Without Circuits

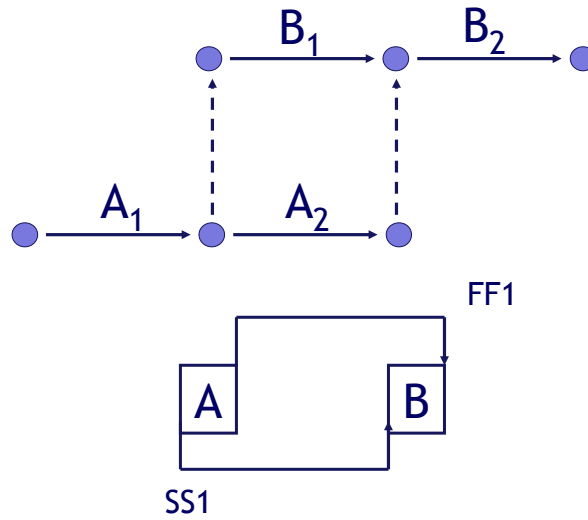
	Certain Durations		Uncertain Durations
Link types	FS	FS-FF-SS-SF	FS
Methods	CPM	PDM	Pert
Output (Durations)	Deterministic		Probabilistic

Uncertain Paths  
 Uncertain Durations  Simulations  
 Every Links

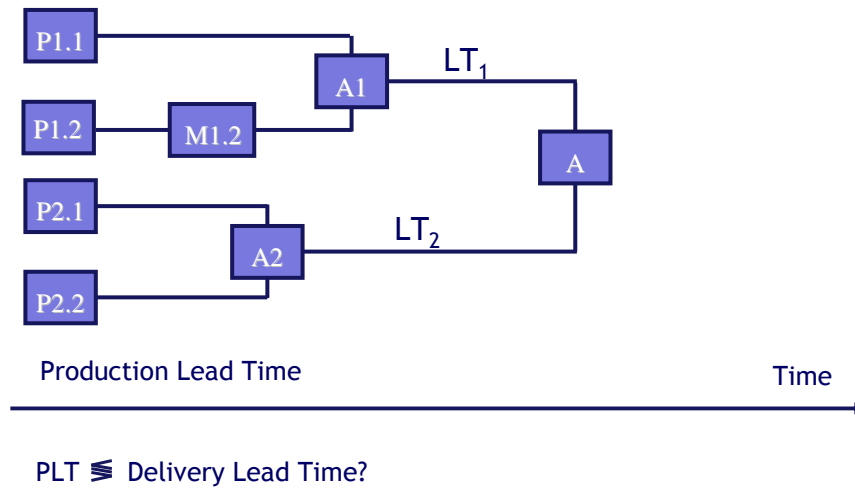
## Links types



## Links types



## Lead Time Tree



## Terminology and definitions

---

EOT <sub>i</sub>	Earliest Occurrence Time	(event i)
LOT <sub>i</sub>	Latest Occurrence Time	(event i)
EST <sub>ik</sub>	Earliest Starting Time	(activity ik)
EFT <sub>ik</sub>	Earliest Finishing Time	(activity ik)
LST <sub>ik</sub>	Latest Starting Time	(activity ik)
LFT <sub>ik</sub>	Latest Finishing Time	(activity ik)
t <sub>ik</sub>	Time	(activity ik)

## Terminology and definitions

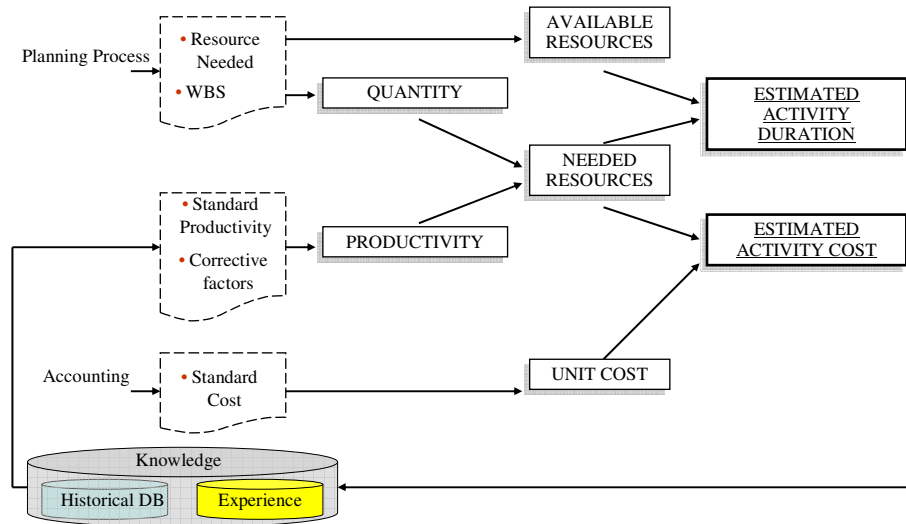
---

$i = 0$  indicates the start project node  
 $i = N$  indicates the end project node

$A_i$  = set of nodes  $x$  for which the  $xi$  arc exists  
and converges on  $i$

$B_i$  = set of nodes  $y$  for which the  $iy$  arc exists  
and comes from  $i$

## Activity Duration



## Earliest and Latest Dates Calculation

The algorithm to calculate the network activities' dates is composed by two phases:

Forward pass

(earliest dates calculation)

Backward pass

(latest dates calculation)

## Earliest and Latest Dates Calculation

---

Dates are scheduled in the two versions:  
“**earliest**”, anticipating the activities that are not in the critical paths to the earliest dates  
“**latest**”, postponing anticipating the activities that are not in the critical paths to the latest dates

### CONVENTION:

activity start = unit of time start

activity end = unit of time end

$$FT = ST + d - 1 \quad d = FT - ST + 1$$

$$ST = FT - d - 1$$

## Forward Pass

---

Beginning from the start project node (node 0) end going toward the end project node all the earliest start and finish dates for each event/activity are computed

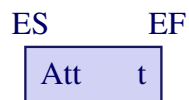
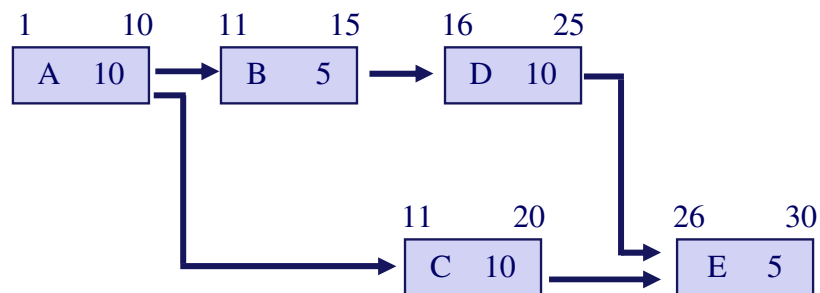
$$EOT_0 = 0$$

$$EST_{ik} = EOT_i + 1$$

$$EFT_{ik} = EST_{ik} + t_{ik} - 1$$

$$EOT_k = \max_{j \in Ak} (EFT_{jk})$$

## Forward Pass



$$EF = ES + \text{duration} - 1$$

## Backward Pass

It is necessary to consider the “end project” event (node N) and to set its maximal date ( $LOT_N$ ) equal to the lowest date  $EOT_N$  computed in the previous pass

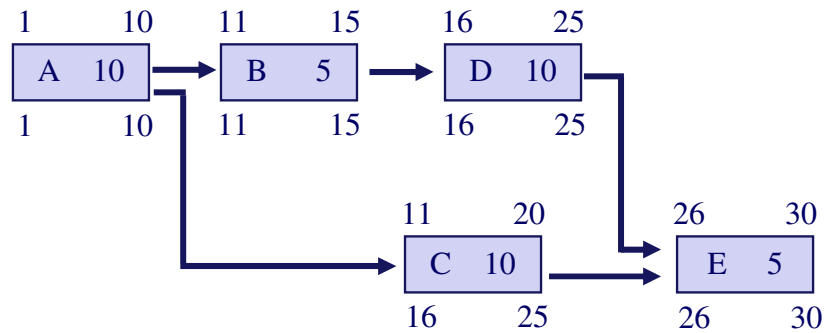
$$LOT_N = EOT_N$$

$$LFT_{ik} = LOT_k$$

$$LST_{ik} = LFT_{ik} - t_{ik} + 1$$

$$EOT_i = \min_{l \in Bi} (LST_{il}) - 1$$

## Backward Pass



$$LS = LF - \text{duration} + 1$$

## Backwad Pass

From the resulting dates is possible to know:

- The internal interrelationship logic
- The activities durations

It is possible to assign to  $LOT_N$  values (different from  $EOT_N$ ) imposed by external constraints to the network logic (contractual expiration date, ...)

It is established:

- if degrees of freedom exist
- where to share eventual negative TF

## Exercise (1/3)

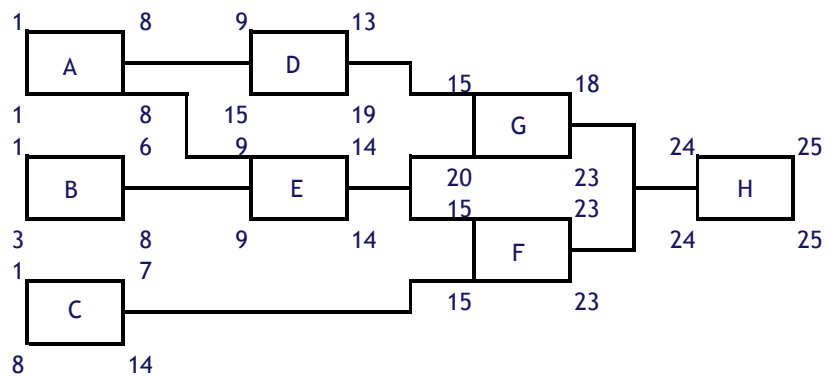
From the data in the table draw:

- the precedence diagram
- the allocated resources profile
- the Gantt chart

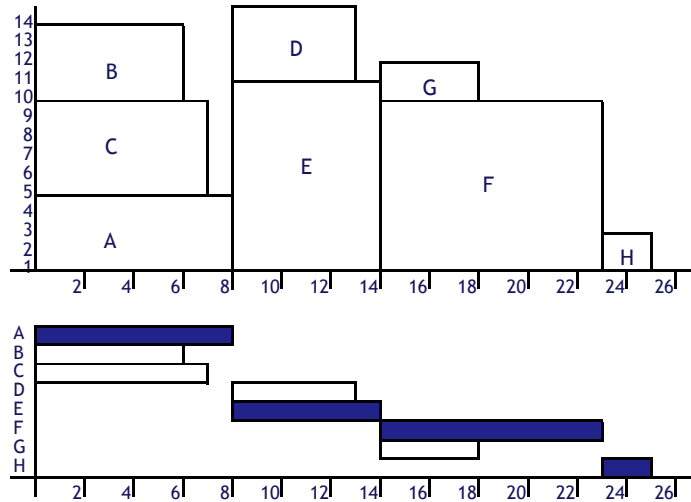
Assuming that there are infinite resources available and not temporal constraints

WP	Precedence	Duration	res. per day
A	/	8	4
B	/	6	4
C	/	7	5
D	A	5	4
E	A,B	6	10
F	E,C	9	9
G	D,E	4	2
H	G,F	2	2

## Exercise (2/3)



## Exercise (3/3)



## Float Calculation

According to the comparison among the earliest and latest dates, through the floats examination it is possible to individualize the project critical activities:

The following types of float are defined:

- Total float
- Free float
- Independent float

## Total Float

---

Total float is the maximum interval available to postpone or extend the execution for a single activity, without change the project deadline

## Total Float

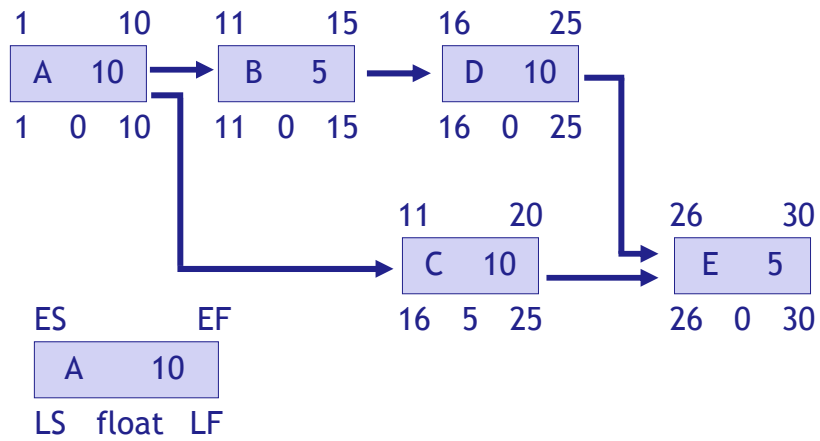
---

It is calculated using the following relationships:

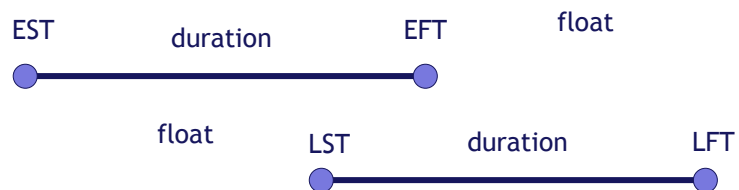
$$\begin{aligned}Sti_k &= \boxed{LFT_{ik} - EFT_{ik}} = LFT_{ik} - (EST_{ik} + t_{ik} - 1) = \\ &= (LFT_{ik} - t_{ik} + 1) - EST_{ik} = \boxed{LST_{ik} - EST_{ik}} = \\ &= (LOT_k - t_{ik} + 1) - EOT_i - 1 = \boxed{LOT_k - EOT_i - t_{ik}}\end{aligned}$$

It represents the maximum interval available to reallocate the activities

## Float Calculation

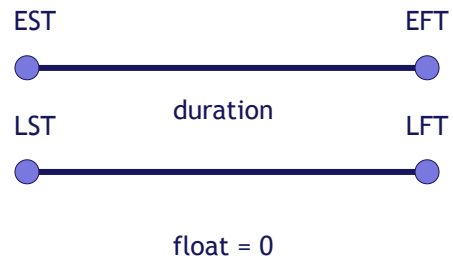


## Not Critical Activities



## Critical Activities

---



## Free Float (FF)

---

Float available for the activity without need to postpone the earliest start date for the successive node (k)

Hypothesis: the starting node (i) is earliest realized

## Free Float (FF)

---

The following relationships are assumed:

$$\begin{aligned} SL_{ik} &= EOT_k - EFT_{ik} = EOT_k - (EST_{ik} + t_{ik} - 1) = \\ &= \boxed{EOT_k - EOT_i - t_{ik}} \end{aligned}$$

It always results:  $SL_{ik} \leq ST_{ik}$

## Independent Float

---

Acceptable delay for an activity with the hypothesis that the starting date was the latest start date and the finish activities (successive activity starting) must be realized at the earliest start date. The independent float does not influence the previous and successive activities

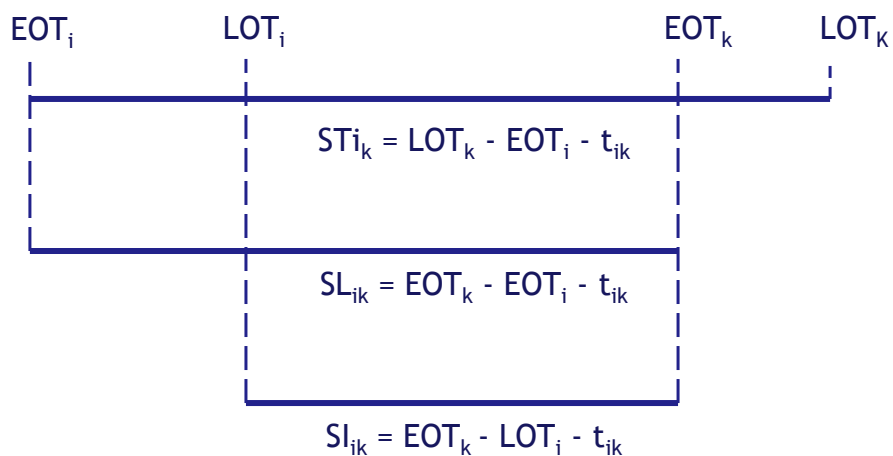
## Independent Float

It is calculated with the following relationship:

$$SI_{ik} = EOT_k - LOT_i - t_{ik}$$

It always results  $SI_{ik} \leq SL_{ik} \leq ST_{ik}$

## Float Graphic Representation



## Temporal Constraints on Nodes

---

- NET “not earlier than”  
(forward pass)
- NLT “not later than”  
(backward pass)
- ON  
(forward e backward pass)

## Critical Path

---

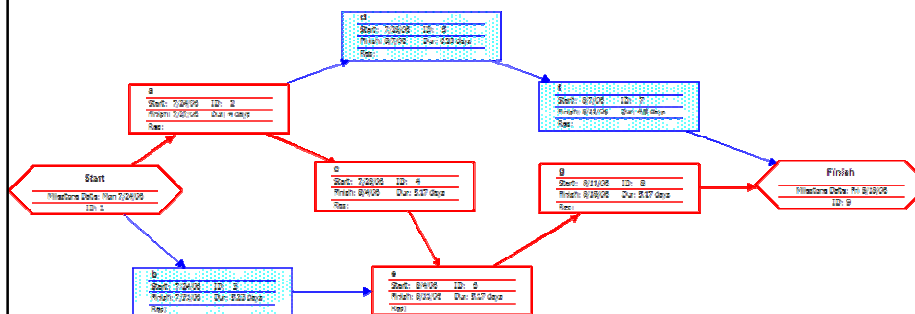
It is the activities chain characterized by the maximal comprehensive duration or by the minimal float

A delay for any activity belonging to the Critical Path implies a delay for the whole project.

In order to reduce the whole project duration (10-20%) is necessary to operate on the critical path's activities.

## Critical Path

- If  $EOT_N = LOT_N$   $TF=0$
- If there are temporal constraints on the nodes the critical path could not be the path with the maximal duration.



Mauro Mancini ©

51

## Network Diagrams' Limits

- In order to achieve reliable analysis it is necessary
  - to consider a great amount of activities
  - to use sophisticated logical links
- It necessary to consider a great amount of data about:
  - activities' attributes
  - How the attributes change changing the activities

Mauro Mancini ©

52

## Network Diagrams' Limits

---

- The problems about data's updating are increased by the method analyticity
- After each updating it is necessary to critically verify the results, analyzing the development for the logical project concatenations
- The method does not allow to carry out "previsione a finire" on the base of advancing data

## CPM tips

---

- If you need to accelerate the schedule, do it by fast-tracking and crashing.
- Watch out! Accelerating the schedule may increase the number of critical activities. (from 10 percent to 40-50 percent of activities will result critical).
- Sprinkle major milestones over your CPM chart. It helps you see woods (milestones) and trees (activities).

## CPM tips

- Color-code activities performed by various resource providers in order to identify their interfaces and provide their coordination.
- Develop template CPM charts. Then, use them consistently to develop the schedule for new projects.

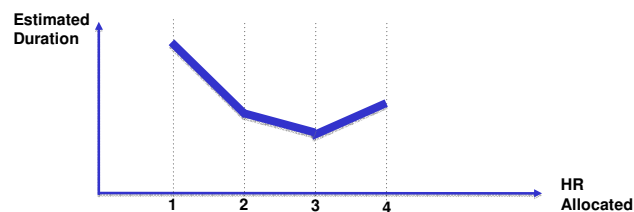
## Cost Critical Path Method (CPM)

It considers the time/costs trade-off

- “normal” duration/cost
- “crashed” duration/cost

It tries to reduce the project duration to the least necessary expense

- find the critical path
- compact the critical path activities beginning from the least expensive
- repeat the process until  $dC/dT < dBenefits/dT$



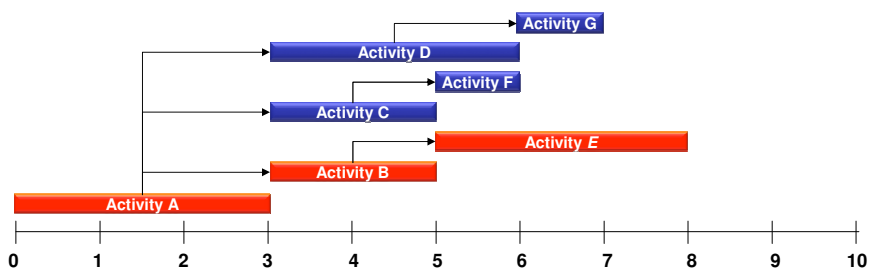
## Cost Critical Path Method (CPM): example

ACTIVITY	PRED.	NORMAL TIME	NORMAL COST	CRASHED TIME	CRASHED COST	dC/dT
A	-	3	30	2	40	10
B	A	2	30	1	65	35
C	A	2	60	/	/	/
D	A	3	60	2	75	15
E	B	3	30	2	45	15
F	C	1	30	/	/	/
G	D	1	60	/	/	/

suppose that the customer would give a prize of "40" for every unitary reduction of delivery times

Mauro Mancini ©

## Cost Critical Path Method (CPM): example



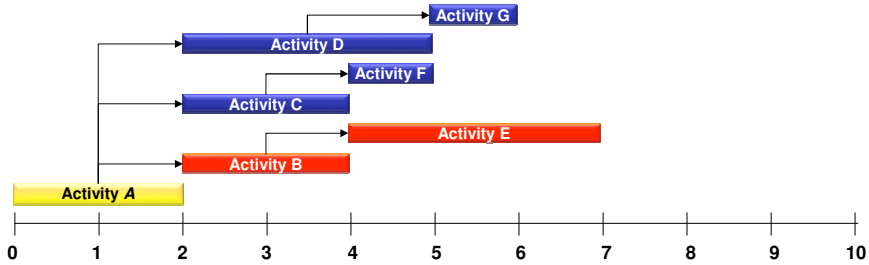
Activity	Pred.	Early Start	Early Finish	Late Start	Late Finish	dC/dT
A	-	0	3	0	3	10
B	A	3	5	3	5	35
C	A	3	5	5	7	/
D	A	3	6	4	7	15
E	B	5	8	5	8	15
F	C	5	6	7	8	/
G	D	6	7	7	8	/

Project duration= 8  
Project Cost= 210  
Critical Activities A, B, E

*A is the candidate for crashing*

Mauro Mancini ©

## Cost Critical Path Method (CPM): example



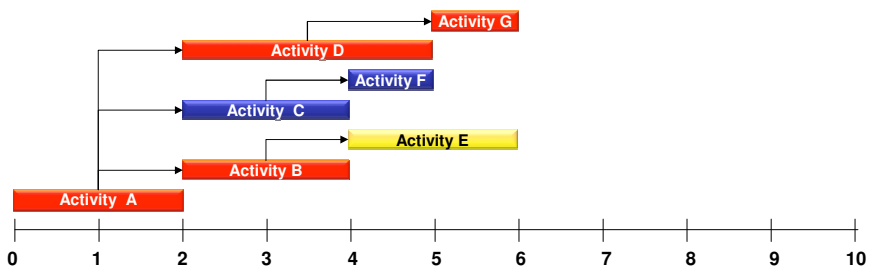
Activity	Pred.	Early Start	Early Finish	Late Start	Late Finish	dC/dT
A	-	0	3	0	3	na
B	A	3	5	3	5	35
C	A	3	5	5	7	/
D	A	3	6	4	7	15
E	B	5	8	5	8	15
F	C	5	6	7	8	/
G	D	6	7	7	8	/

Project Duration = 7  
 Project Cost = 220  
 Marginal Cost = 10  
 Marginal Revenue = 40  
 Profit = 30  
 Critical Activities A, B, E

*E is the candidate for crashing*

Mauro Mancini ©

## Cost Critical Path Method (CPM): example



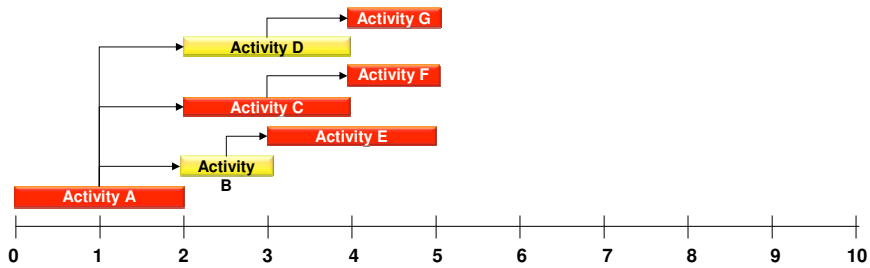
Activity	Pred.	Early Start	Early Finish	Late Start	Late Finish	dC/dT
A	-	0	3	0	3	na
B	A	3	5	3	5	35
C	A	3	5	5	7	/
D	A	3	6	4	7	15
E	B	5	8	5	8	na
F	C	5	6	7	8	/
G	D	6	7	7	8	/

Project Duration = 6  
 Project Cost = 235  
 Marginal Cost = 15  
 Marginal Revenue = 40  
 Profit = 25  
 Critical activities: A, B, E and A, D, G

*D e B are both candidate for crashing*

Mauro Mancini ©

## Cost Critical Path Method (CPM): example

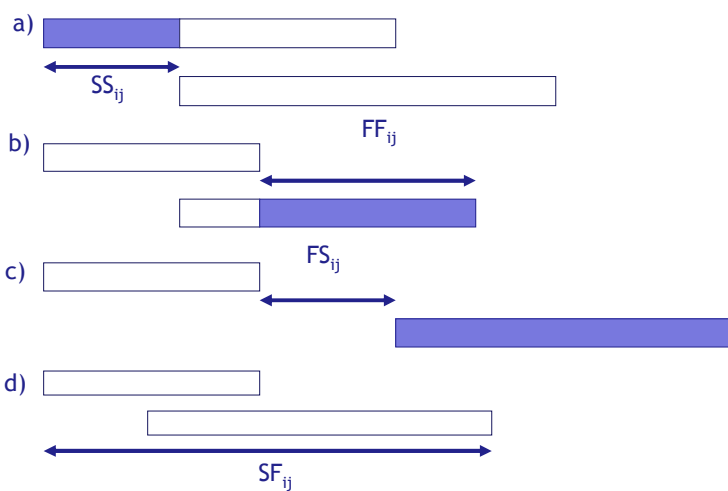


Activity	Pred.	Early Start	Early Finish	Late Start	Late Finish	dC/dT
A	-	0	3	0	3	na
B	A	3	5	3	5	na
C	A	3	5	5	7	/
D	A	3	6	4	7	na
E	B	5	8	5	8	na
F	C	5	6	7	8	/
G	D	6	7	7	8	/

Project Duration = 5  
 Project Cost = 285  
 Marginal Cost = 50  
 Marginal Revenue = 40  
 Profit = -10  
 Critical Activities: A, B, E; A, D, G and A, C, F  
 The profit is negative, we refuse to crash B and D

Mauro Mancini ©

## Link Between Activities - Representation Methods



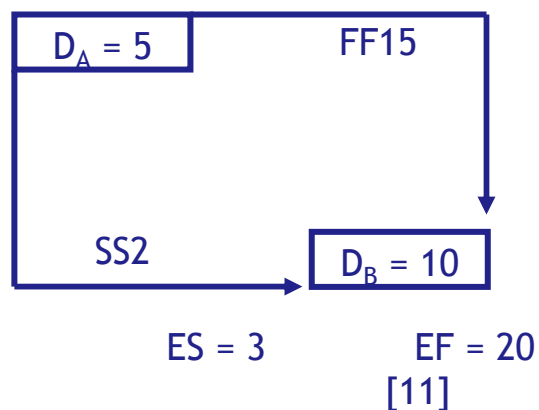
Mauro Mancini ©

62

## Link Between Activities - Representation Methods

- $SS_{ij}$  Start to Start  
Minimal number of units of time that must be past from the previous activity's start to allow the successive activity to start
- $FF_{ij}$  Finish to Finish  
Minimal number of units of time required to complete the successive activity after the previous activity's completion.
- $FS_{ij}$  Finish to Start  
Minimal number of units of time that must be past from the previous activity's completion to allow the successive activity to start
- $SF_{ij}$  Start to Finish  
Minimal number of units of time that must be past from the previous activity's start to the successive activity's completion

## Precedence Diagrams



## Precedence Diagrams - Forward Pass

It is necessary to consider the constraint relative to the considered activity' predecessors (Start e Finish)

$$FS_{ij} \quad ES_j = EF_i + lag + 1$$

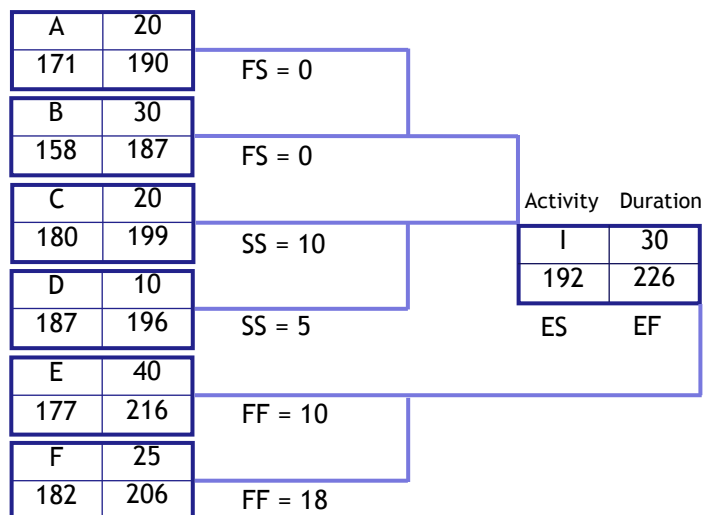
$$SS_{ij} \quad ES_j = ES_i + lag$$

$$FF_{ij} \quad EF_j = EF_i + lag$$

$$SF_{ij} \quad EF_j = ES_i + lag - 1$$

In case of many predecessors it is necessary to consider the maximal date

## Precedence Diagrams - Forward Pass - Example



## Precedence Diagrams - Forward Pass - Example

---

$$ES (I/A) = 190 + 0 + 1 = 191$$

$$ES (I/B) = 187 + 0 + 1 = 188$$

$$ES (I/C) = 180 + 10 = 190$$

$$ES (I/D) = 187 + 5 = \underline{192}$$

$$EF (I) = 192 + 30 - 1 = 221$$

$$EF (I/E) = 216 + 10 = \underline{226}$$

$$EF (I/F) = 206 + 18 = 224$$

## Precedence Diagrams - Forward Pass - Example

---

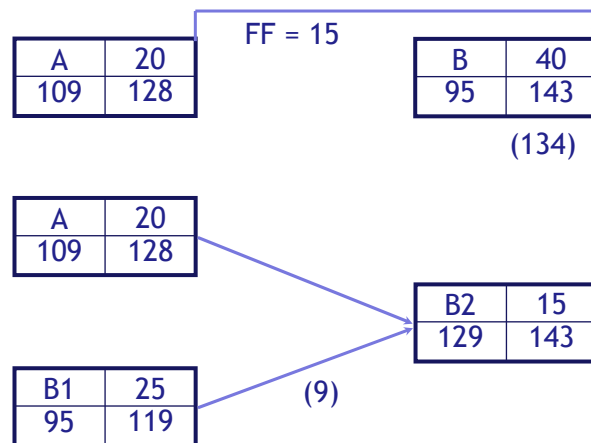
In general:

$$EF = ES + d - 1$$

In presence of external predecessors for its finish (FF and SF), it is possible having incongruities.



## Precedence Diagrams - Forward Pass - Splitting Example



## Precedence Diagrams - Backward Pass

It is necessary to consider the constraint relative to the considered activity' successors (Start e Finish)

$$FS_{ij} \quad LF_j = LS_i - \text{lag} - 1$$

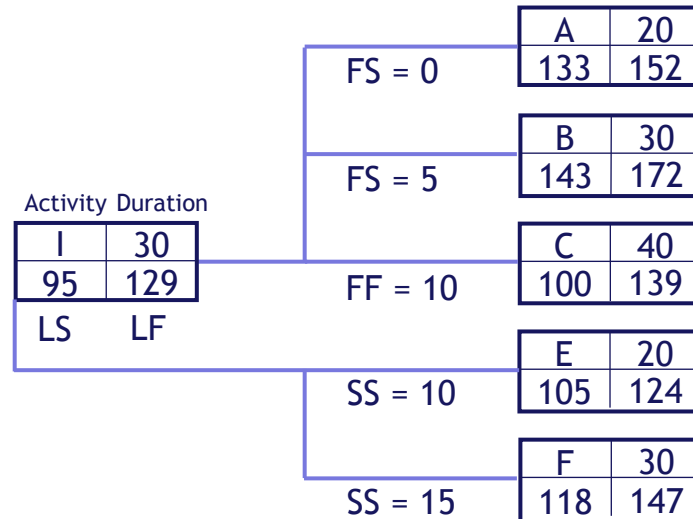
$$SS_{ij} \quad LS_j = LS_i - \text{lag}$$

$$FF_{ij} \quad LF_j = LF_i - \text{lag}$$

$$SF_{ij} \quad LS_j = LF_i - \text{lag} - 1$$

In case of many predecessors it is necessary to consider the minimal date

## Precedence Diagrams - Backward Pass - Example



## Precedence Diagrams - Backward Pass - Example

$$LF (I/A) = 133 - 0 - 1 = 132$$

$$LF (I/B) = 143 - 5 - 1 = 137$$

$$LF (I/C) = 139 - 10 = \underline{129}$$

$$LS (I) = 129 - 30 + 1 = 100$$

$$LS (I/E) = 105 - 10 = \underline{95}$$

$$LS (I/F) = 118 - 15 = 103$$

## Precedence Diagrams - Backward Pass - Example

In general:

$$LS = LF - d - 1$$

In presence of external predecessors for its start (FS and SS), it is possible having incongruities.



## Precedence Diagrams - Backward Pass - Splitting Example

