The Critical Path Problem

Giovanni Righini

University of Milan

UNIVERSITÀ DEGLI STUDI DI MILANO

Project management problems

Project management is a typical application for graph optimization algorithms.

In its most general formulation a project management problem consists of arranging a set of activities on a time line, complying with

- minimum duration constraints for each activity,
- precedence constraints between activities,
- time constraints such as release dates, due dates and deadlines,
- limited capacities, limited resources.

Project management with infinite resources

The simplest case does not take limited resources into account.

We are given

- a set $\mathcal{J}$ of activities,
- a duration d_j for each activity $j \in \mathcal{J}$;
- a set $\mathcal{P}_j \subseteq \mathcal{J}$ of precedessors for each activity $j \in \mathcal{J}$.

The objective is to decide when to start and end each activity in order to minimize the overall project duration.

A mathematical formulation

The model of this problem uses a pair of variables s_j and e_j to represent the start time and the end time of each activity $j \in \mathcal{J}$.

A variable z indicates the completion time of the project which is assumed w.l.o.g. to start at time 0.

$$\begin{aligned} \text{minimize } & z \\ \text{s.t. } & e_j \geq s_j + d_j && \forall j \in \mathcal{J} \\ & s_j \geq e_i && \forall j \in \mathcal{J}, \forall i \in \mathcal{P}_j \\ & z \geq e_j && \forall j \in \mathcal{J} \\ & s, e \geq 0 \end{aligned}$$

This is a linear programming problem, but it can be solved by specialized algorithms instead of general LP methods.

A graphical representation

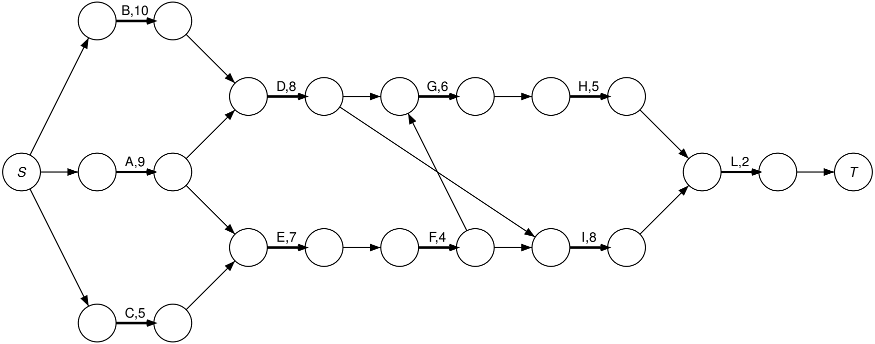
A graphical representation of the problem can be given in this way.
Consider a digraph $\mathcal{D} = (\mathcal{N}, \mathcal{A})$ with

- a start node and an end node for each activity $j \in \mathcal{J}$;
- an activity arc from the start node to the end node for each activity $j \in \mathcal{J}$;
- a precedence arc from the end node of activity $i \in \mathcal{J}$ to the start node of activity $j \in \mathcal{J}$ for each pair of activities such that $i \in \mathcal{P}_j$;
- a node S and a node T corresponding with the beginning and the end of the project;
- arcs from S to all start nodes without precedessors;
- arcs from all end nodes without successors to T;
- for each activity arc a weight d_j equal to the duration of the corresponding activity $j \in \mathcal{J}$;
- null weights for the precedence arcs.

An example: the data

Activity	Duration	Precedessors
A	9	-
B	10	-
C	5	-
D	8	A,B
E	7	A,C
F	4	E
G	6	D,F
H	5	G
I	8	D,F
L	2	H,I

An example: the digraph



The digraph with activity arcs (bolded) and precedence arcs.

The Critical Path Method

In Phase 1 the algorithm propagates node labels e from S to T.

Each label e_v represents the earliest time at which the event corresponding with node $v \in \mathcal{N}$ can occur, provided that the project cannot start before time 0. Hence e_T is the minimum completion time for the project.

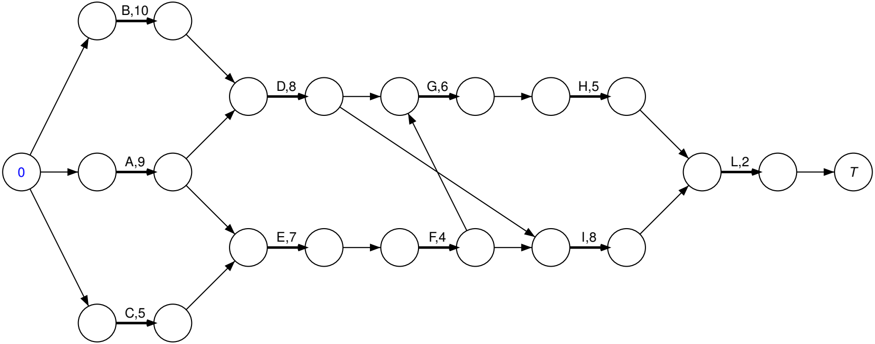
$$\forall v \in \mathcal{N} \quad e_v := \max_{u \in \mathcal{N}:(u,v) \in \mathcal{A}} \{e_u + d_{uv}\}.$$

In Phase 2 the algorithm propagates node labels l from T to S.

Each label l_v represents the latest time at which the event corresponding with node $v \in \mathcal{N}$ can occur, provided that the project cannot end after its optimal completion time computed in Phase 1.

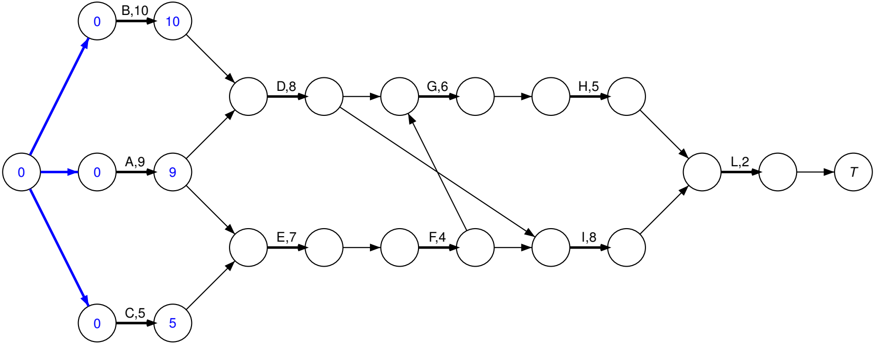
$$\forall v \in \mathcal{N} \quad l_v := \min_{u \in \mathcal{N}:(v,u) \in \mathcal{A}} \{l_u - d_{vu}\}.$$

Example: forward propagation of labels e.



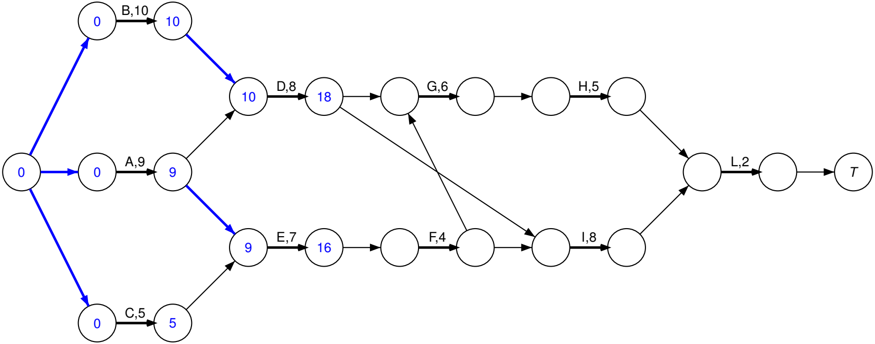
Forward propagation is initialized setting $e_S = 0$.

Example: forward propagation of labels *e*.

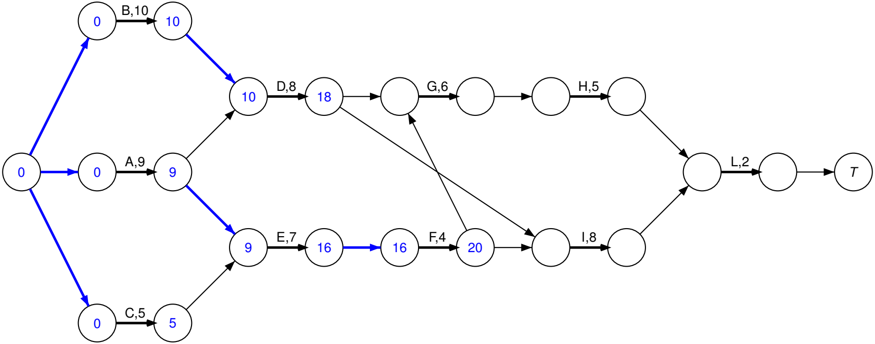


Blue precedence arcs indicate critical predecessors.

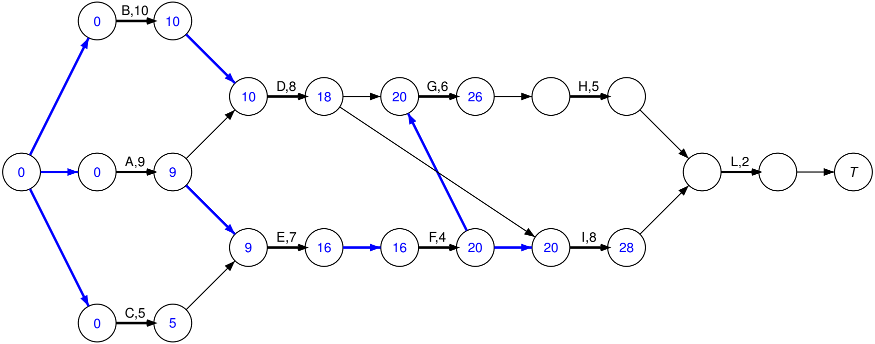
Example: forward propagation of labels *e*.



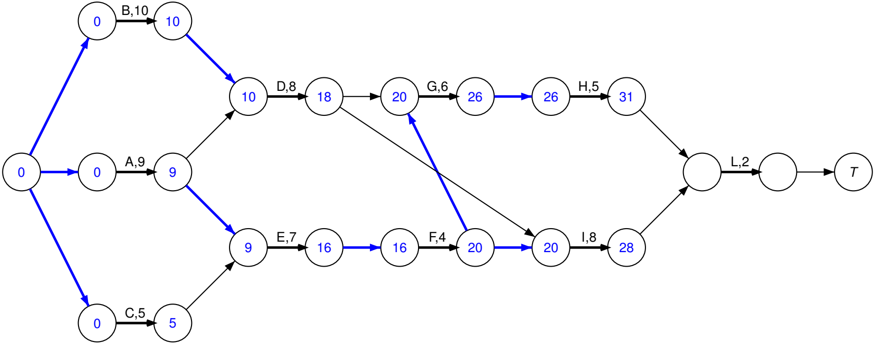
Example: forward propagation of labels *e*.



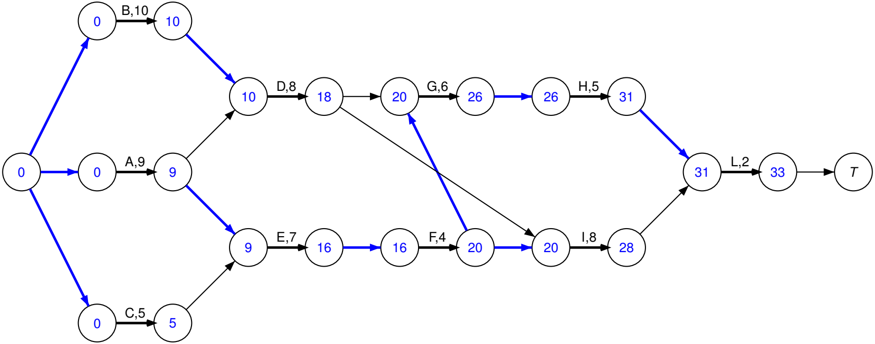
Example: forward propagation of labels e.



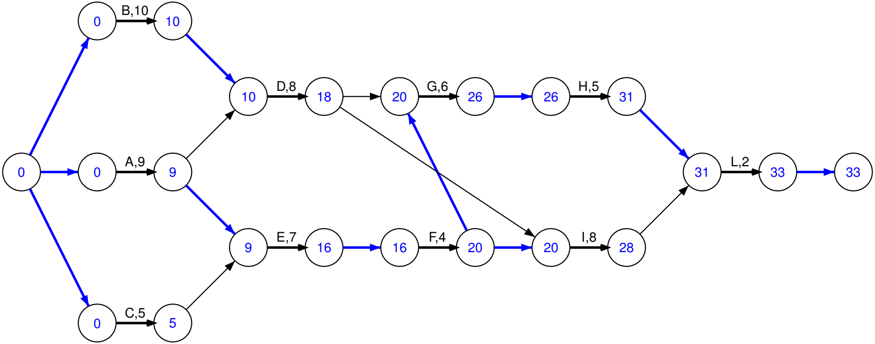
Example: forward propagation of labels *e*.



Example: forward propagation of labels *e*.

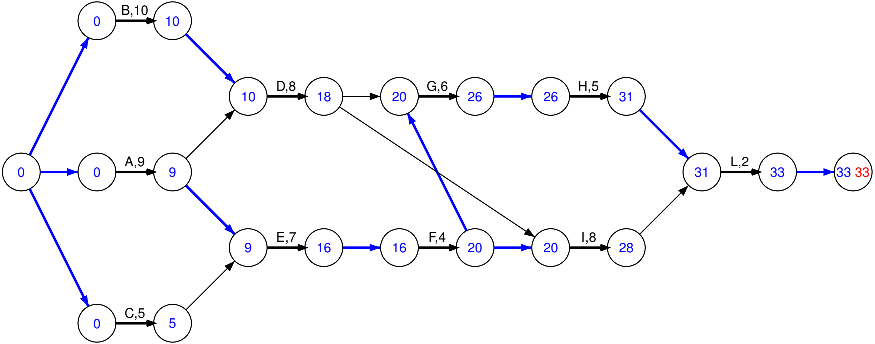


Example: forward propagation of labels e.



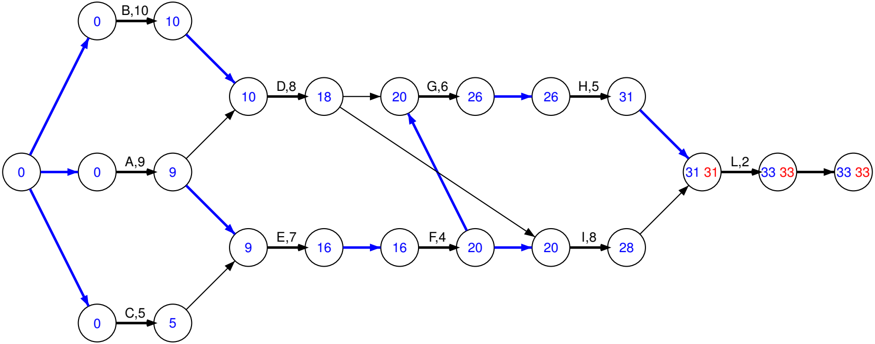
The minimum completion time z^* for the whole project is $e_T = 33$.
The **critical path** is made by activities A,E,F,G,H,L. These are **critical activities**: an ϵ delay in their duration results in an ϵ increase of the project duration.

Example: backward propagation of labels l.



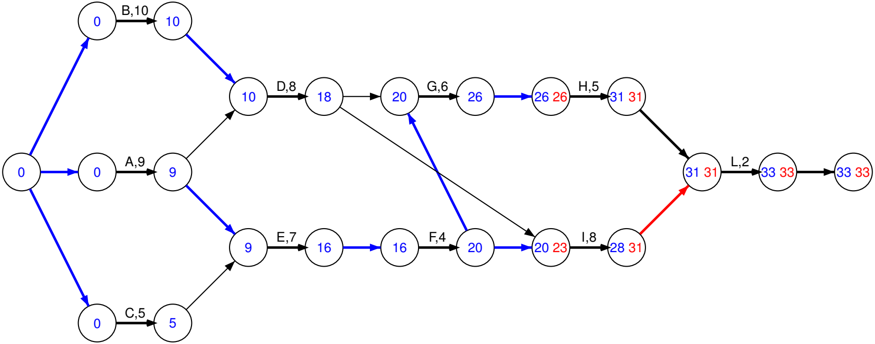
Backward propagation is initialized by setting $l_T = z^*$.

Example: backward propagation of labels l.

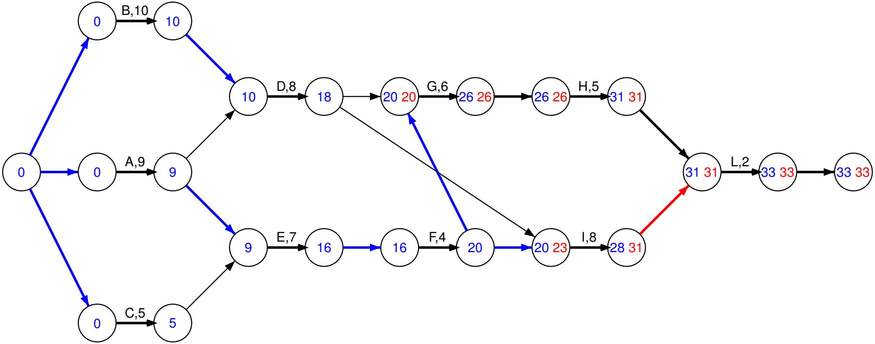


Red precedence arcs indicate critical successors.
Precedence arcs that should be both blue and red are indicated in thick black.

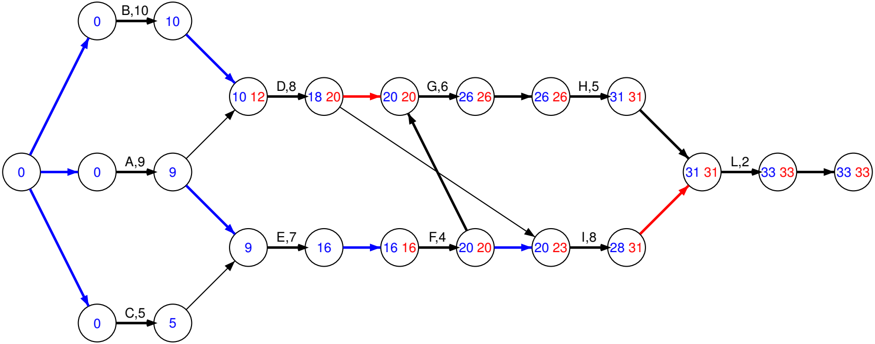
Example: backward propagation of labels l.



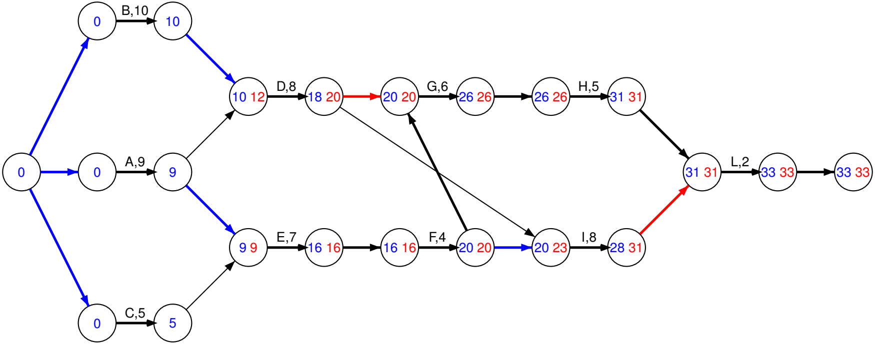
Example: backward propagation of labels l.



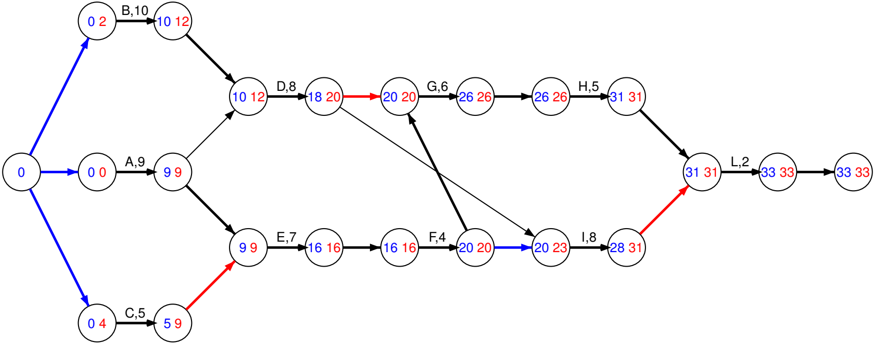
Example: backward propagation of labels l.



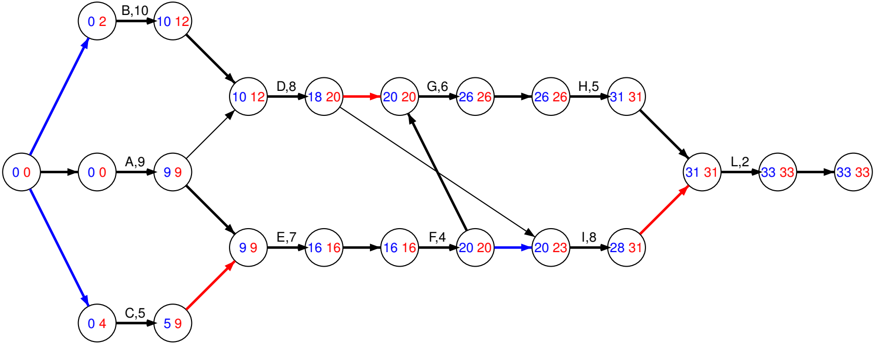
Example: backward propagation of labels l.



Example: backward propagation of labels l.



Example: backward propagation of labels l.



At the end of Phase 2 we must get $l_S = 0$ and the same critical path as in Phase 1.

The values of forward and backward labels along the **critical path** are the same. For the non-critical activities the difference between the labels is the allowed tolerance (slack) in their schedule.

Example: the results

Activity	Duration	Pred.	Earliest start	Latest start	Earliest end	Latest end	Slack
A	9	-	0	0	9	9	0
B	10	-	0	2	10	12	2
C	5	-	0	4	5	9	4
D	8	A,B	10	12	18	20	2
E	7	A,C	9	9	16	16	0
F	4	E	16	16	20	20	0
G	6	D,F	20	20	26	26	0
H	5	G	26	26	31	31	0
I	8	D,F	20	23	28	31	3
L	2	H,I	31	31	33	33	0

Program Evaluation and Review Technique (PERT)

It is also called *three-point estimation* technique, because each activity i is assumed to have an uncertain duration, described by three values:

- d_i^*: nominal duration;
- $\underline{d}_i$: optimistic duration;
- $\overline{d}_i$: pessimistic duration.

The uncertain duration d_i of each activity i is represented by a Beta probability distribution with standard deviation $\sigma_i = \frac{\overline{d}_i - \underline{d}_i}{6}$.

With these assumptions, the expected duration $\hat{d}_i$ of each activity i is

$$\hat{d}_i = \frac{\underline{d}_i + 4d_i^* + \overline{d}_i}{6}.$$

Program Evaluation and Review Technique (PERT)

Assuming all activities are independent,

- the expected duration of a sequence of activities is the sum of the expected durations of its activities:

$$\hat{d}(S) = \sum_{i \in S} \hat{d}_i$$

- the variance of a sequence is the sum of the variances of its activities:

$$\sigma^2(S) = \sum_{i \in S} \sigma_i^2$$

Activities that are expected to be critical are identified with CPM: S^*.
The expected duration of a whole project (duration of S^*) is also computed:

$$d(S^*) = \hat{d}(S^*) \pm \sigma(S^*),$$

where $\sigma(S^*) = \sqrt{\sigma^2(S^*)}$.

Uncertainty evaluation

The reliability of the result is estimated by assuming that the project duration is a random variable with normal distribution with expected value $d(S^*)$ and variance $\sigma^2(S^*)$.

In this way we can estimate the probability that the project duration be within an interval of width $l = k\sigma$ around the mean, for any choice of k.

A larger value of k corresponds to a more reliable estimate.