The PERT Method

Optimization Methods in Management Science
Master in Management
HEC Lausanne

Dr. Rodrigue Oeuvray

Fall 2019 Semester

The PERT Method

PERT: Project Evaluation and Review Technique

We consider a complex project:

- a set of tasks with a known duration
- precedence constraints between some tasks

We would like to **order** the tasks and to plan their **start dates** in order to:

- satisfy the precedence constraints
- minimize the total duration of the implementation of the project
- be able to assess the impact of a task delay on the whole project duration

Problem Formulation

This problem is modeled as a **network**:

- the vertices represent the project **tasks**
- an arc connects two vertices i and j if it exists a precedence constraint between the tasks associated with vertices i and j
- the **weight** c_{ij} of the arc (i, j) is equal to the **duration** d_i of the task/vertex i

We also add:

- a vertex α representing the **commencement** of the work, which has a **null** duration and preceding all the tasks without any predecessor
- a vertex ω representing the **end** of the work, with a **null** duration and succeeding all the tasks with no successor

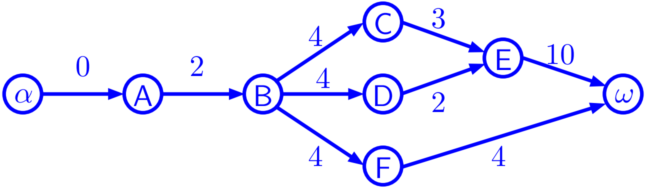
Problem Formulation (Cont'd)

- The graph associated with a project has **no circuit**, only one vertex with no predecessor (α), and only one vertex with no successor (ω)
- It is called the **PERT network**
- Determining the **minimal** duration of the project is equivalent to computing a **longest** path between α and ω

Example

A cable network company wishes to increase the number of channels it offers to its customers. Tasks and precedences of this project are given below:

Task	Description	Duration (in week)	Precedence
A	Choice of the channels	2	—
B	Administrative work	4	A
C	Order of the decoders	3	B
D	Installation of the antennas	2	B
E	Installation of the decoders	10	C,D
F	Modification of the billing	4	B



The Critical Path Algorithm

Input: a PERT network $R = (V, E, c)$ associated with a project and a topological order of its vertices (α and ω are labeled 1 and n respectively)

Output: the minimal duration D of a project and, for each task i, its **earliest** start date δ_i and its **latest** start date φ_i

(1) Update of δ_k

$\delta_1 = 0$

For $k = 2$ to n, set $\delta_k = \max\{\delta_j + c_{jk} \mid j \in Pred(k)\}$

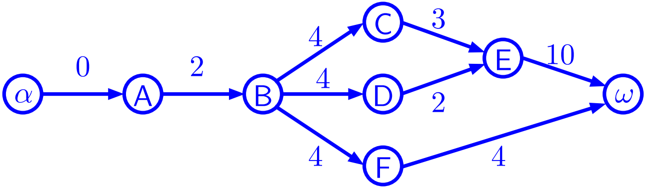
(2) Update for φ_k

$D = \delta_n$, $\varphi_n = \delta_n$

For $k = n - 1$ to 1, set $\varphi_k = \min\{\varphi_j - c_{kj} \mid j \in Succ(k)\}$

Example (Cont'd)

Task	α	A	B	C	D	E	F	ω
k	1	2	3	4	5	6	7	8
Duration d_k	0	2	4	3	2	10	4	0
$Pred(k)$	—	α	A	B	B	C, D	B	E, F
$Succ(k)$	A	B	C, D, F	E	E	ω	ω	—
$\delta_k / Pred(k)$	$0/-$	$0/\alpha$	$2/A$	$6/B$	$6/B$	$9/C$	$6/B$	$19/E$
$\varphi_k / Succ(k)$	$0/A$	$0/B$	$2/C$	$6/E$	$7/E$	$9/\omega$	$15/\omega$	$19/-$



- $\delta_k = \max\{\delta_j + c_{jk} \mid j \in Pred(k)\}$, $\delta_1 = 0, k = 2, \ldots, n$
- $\varphi_k = \min\{\varphi_j - c_{kj} \mid j \in Succ(k)\}$, $\varphi_n = \delta_n, k = (n-1), \ldots, 1$

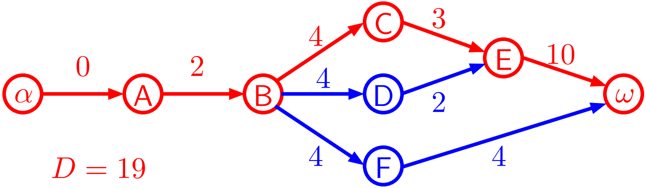
Terminology

- A task i is **critical** if $\delta_i = \varphi_i$. Every delay in its implementation impacts the total duration of the project
- A **critical** path is a path from α to ω which is composed solely of critical tasks
- The **length** of a critical path corresponds to the **minimal** duration of the project

Example (Cont'd)

In our example, critical tasks are α, A, B, C, E et ω. The unique critical path is

$$C = (\alpha, (\alpha, A), A, (A, B), B, (B, C), C, (C, E), E, (E, \omega), \omega).$$



Its length is 19 and corresponds to the minimal duration of the project