8 June

Critical path method [not in P&S]

This presentation is from Walker, Introduction to Mathematical Programming (available in the Sorrells Engineering and Science Library).

A variation is described in Jungnickel (see syllabus).

Common managerial problem: Project consisting of a set of activities (or tasks).

- Each activity has zero or more immediate prerequisites, which are other activities that must be completed first before the given activity can begin.
- Each activity has a duration.
- Activities may be done in parallel (i.e., simultaneously) as long as neither is a prerequisite of the other.

Questions:

- Minimum time to completion of whole project?
- Which activities are most crucial to on-time project completion?
- Flexibility, if any, in completing the activities?

Example. Introduction of a new product.

Activity	Immediate prerequisites	Duration (months)
A. Design product	—	4
B. Develop marketing strategy	—	3
C. Design brochure	A	2
D. Produce prototype	A	6
E. Survey potential market	B, C	3
F. Test prototype	D	3
G. Develop pricing strategy	D, E	2
H. Develop production capability	A	4
I. Write implementation plan	F, G, H	1

We will use the critical path method (CPM) to analyze this project.

First step: Draw (and label) a CPM network.

— Nodes (vertices) represent points in time.

— Arrows (edges) represent activities and enforce precedence between two points in time.

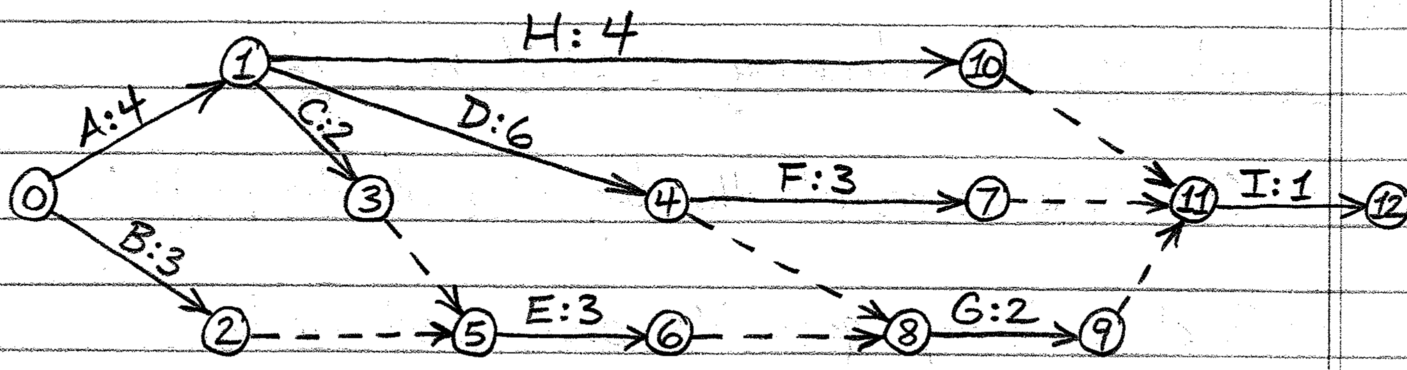
— "Dummy" edges do not represent activities — they are used to enforce precedence only. Drawn as dashed arrows.

8 June

Guidelines for drawing a CPM network

- One node represents the beginning of the project.
- Each activity gets its own end node.
- An activity with no prerequisites should point outward from the project start node.
- An activity with exactly one immediate prerequisite should point outward from that prerequisite's end node.
- An activity with more than one immediate prerequisite should have its own start node, with dummy edges from its prerequisites' end nodes to its start node.
- At the end, there should be a single node representing the completion of the project. If there are several activity end nodes that are sinks (no outgoing edges), make a new node representing the end of the project and join all of the sinks to this end node with dummy edges.

CPM network for previous example



Notes:

- Nodes are numbered 0, 1, 2, ... in the order they were drawn. Node 0 represents the beginning of the project, and node 12 represents the end.

- Edges (arrows) representing activities are labeled with the activity letter and the duration of the activity. The edge $0 \xrightarrow{A:4} 1$ represents activity A and means that the point in time represented by node 1 cannot occur earlier than 4 months after the point in time represented by node 0, so that activity A can be done in between.

- Dummy edges are drawn as dashed arrows. They just enforce precedence: $2 \dashrightarrow 5$ means that the point in time represented by node 5 cannot occur before the point in time represented by node 2.
  - You can think of dummy edges as being activities having duration zero.

8 June

Defn. For a node in a CPM network:
(Remember, a node represents a point in time.)

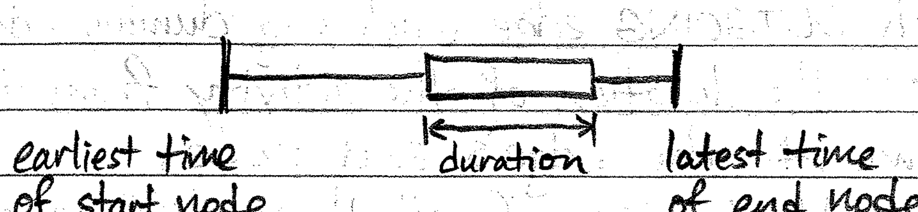
- The earliest time of the node is the first time at which it is possible for the node to occur if the project begins at time 0 and all precedence constraints are satisfied.

- The latest time of the node is the last time at which the node may occur without delaying the overall project completion beyond its earliest time.

- The slack of the node is the difference between its latest and earliest times.

Defn. For an activity in a CPM network:

- The float of the activity is the latest time of its end node, minus the earliest time of its start node, minus the duration of the activity.



Float is the amount of "wiggle room" in the time window available to the activity.

Defn. A <u>critical activity</u> is one whose float is zero.

— If a critical activity is delayed, the whole project will be delayed.

Defn. A <u>critical path</u> is a path from the start of the project to its completion consisting entirely of critical activities.

Rules for computing earliest times, latest times, floats

1. First compute <u>earliest times</u> by going FORWARD in time through the CPM network. The earliest time of node 0 (start of project) is 0. To compute the earliest time of any other node:
   - For each INCOMING edge (including dummy edges), ADD the earliest time of the edge's start node plus the duration of the activity. (The duration of a dummy edge is zero.)
   - Take the MAXIMUM of all these sums.
2. Then compute <u>latest times</u> by going BACKWARD in time through the CPM network. The latest time of the last node (project completion) is its earliest time. To compute the latest time of any other node:
   - For each OUTGOING edge (including dummy edges), SUBTRACT the duration of the activity from the latest time of the edge's end node.
   - Take the MINIMUM of all these differences.

8 June

Earliest times, latest times, floats, etc. — 2

3. After earliest times and latest times have been computed for all nodes, compute the <u>float</u> of each activity as

$$\left(\begin{array}{c}\text{latest time}\\\text{of end node}\end{array}\right) - \left(\begin{array}{c}\text{earliest time}\\\text{of start node}\end{array}\right) - (\text{duration}).$$

Notes:

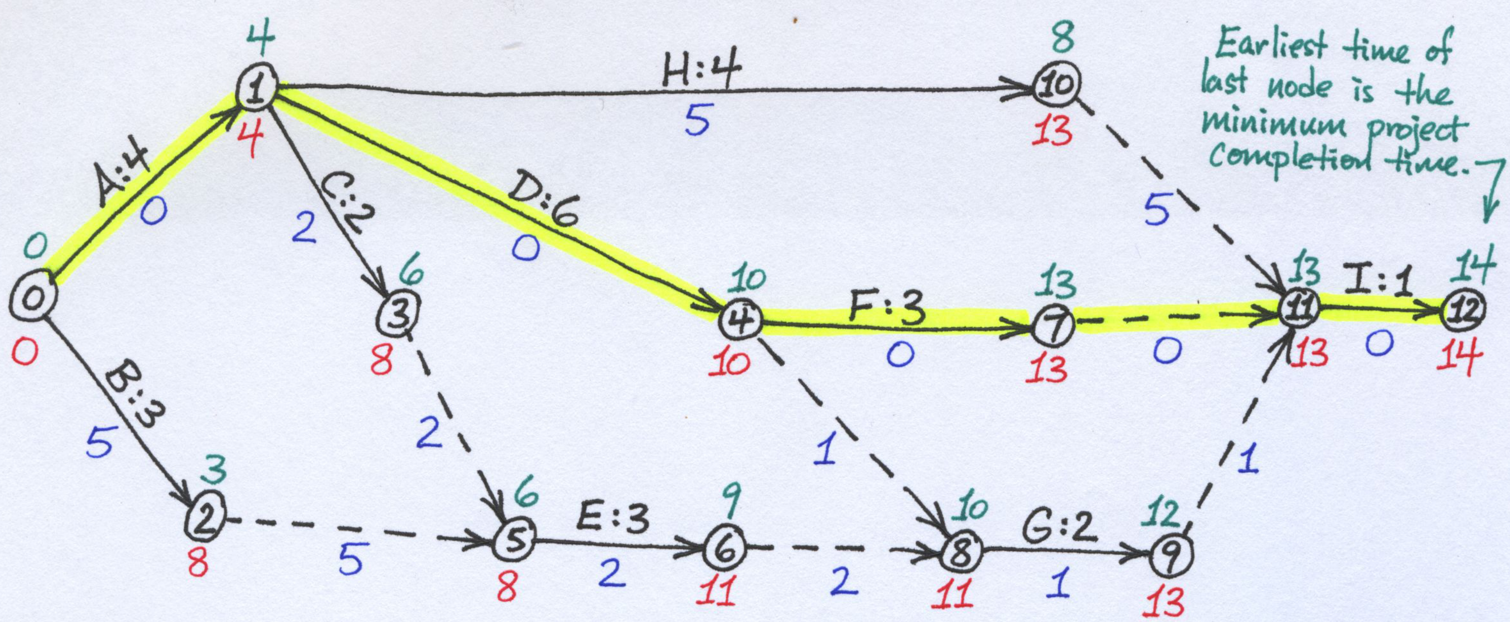
- The latest time of node 0 should always be 0. If you go forward through the network calculating earliest times, and then go backward calculating latest times, and you get something other than 0 for the latest time of node 0, then you made a mistake somewhere.

- Once you have calculated floats, it is easy to identify critical activities as those with float zero.

- There will always be at least one critical path (there may be more than one), and every critical activity is a part of at least one critical path.

8 June

Earliest times, latest times, floats: Example.



Earliest times (written above nodes)
• Compute going FORWARD →

Latest times (written below nodes)
• Compute going BACKWARD ←

Floats (written below activities)

Critical activities: A, D, F, I (and the dummy edge from node 7 to node 11)

Critical path: A — D — F — I.

These four activities must be started immediately once their prerequisites are finished, and their completion must not be delayed, or else the completion of the whole project will be delayed.

8 June.

Crash times.

Often it is possible to speed up the completion of an activity by devoting additional resources to it (e.g., overtime labor, additional machinery, expedited shipping, etc.).

In this context we will call the non-sped-up duration of an activity the <u>usual time</u>. The <u>crash time</u> is the absolute minimum completion time of an activity even if unlimited resources were available.

We will make the simplifying assumption that speedup costs are linear, i.e., the cost of speeding up the completion of an activity by k time units is k times a constant cost per unit.

Additional questions now:

- Given a deadline for project completion, what is the least expensive way to shorten the project to meet the deadline?
- Given a budget of resources that can be used to speed up activities, what is the shortest project completion time achievable?

Example. Suppose that some of the activities in the preceding example can be sped up:

Activity	Immediate prerequisites	Usual time (months)	Crash time (months)	Cost per month to speed up
A	—	4	2	\$4000
B	—	3	1	\$5000
C	A	2	—	—
D	A	6	3	\$4000
E	B,C	3	—	—
F	D	3	2	\$3000
G	D,E	2	—	—
H	A	4	2	\$4000
I	F,G,H	1	—	—

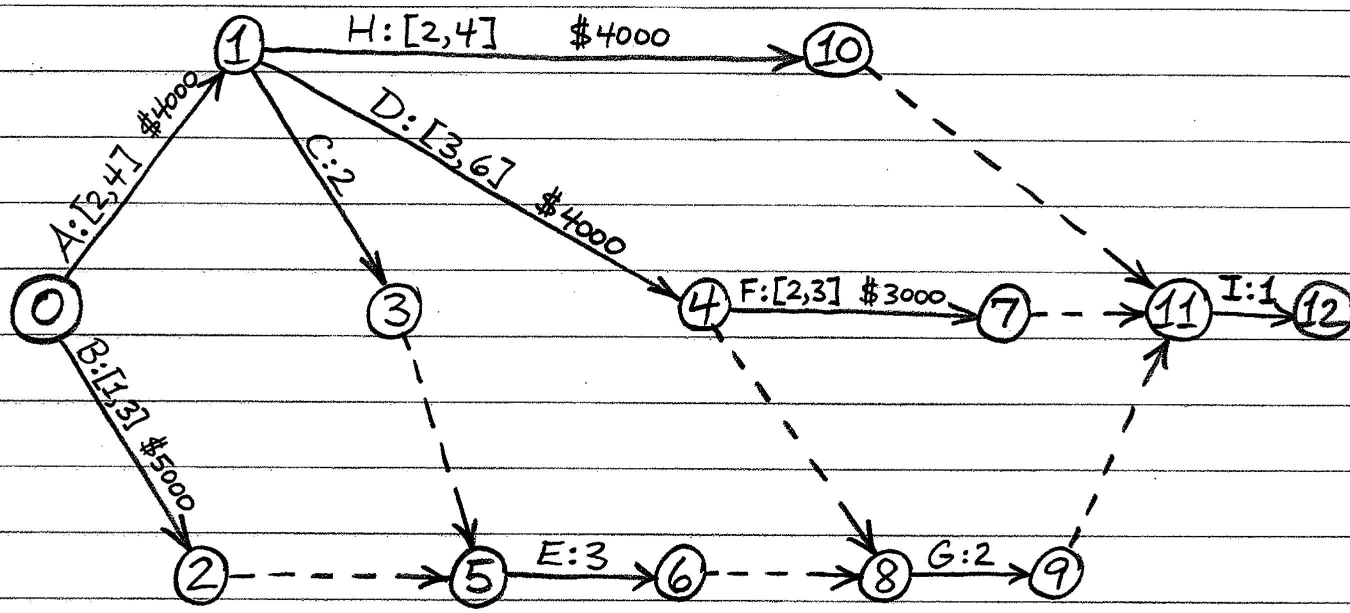
When drawing a CPM network for a problem like this, it is helpful to have all of this information in the drawing. One way to do this is to write the durations of activities that can be sped up as intervals; e.g., the duration of activity A is now a value in the interval $[2,4]$. The speedup costs can also be included in the edge labels, so now the arrow for activity A might look like

$$⓪ \xrightarrow{A:[2,4] \quad \$4000} ①$$

Labels for activities that cannot be sped up (C, E, G, and I) are the same as before.

8 June

New labeling of CPM network with crash times and speedup costs:



8 June

CPM: LP formulation.

Variables: Two kinds.

- Event variables: one for each node in the CPM network, indicating the time at which that node occurs. (Remember that nodes represent points in time.) These variables are called t_i.
  Domain: $t_i \geq 0$ for all i.
  - For the example: $t_0, t_1, t_2, \ldots, t_{12}$.
- Duration variables: one for each activity that can be sped up, indicating the chosen duration of that activity. Activities that cannot be sped up do not need duration variables, because their durations are constant. These variables are called d_j.
  Domain: $d_j \geq 0$ for all j.

  - For the example: d_A, d_B, d_D, d_F, d_H.

  Don't need duration variables for activities C, E, G, I, because their durations are constant (can't be sped up).

Constraints.

Network constraints: Enforce precedence and duration requirements expressed in CPM network. Every LP for a given project must include these constraints, regardless of the question being asked. Three types of network constraints:

- Sequence constraints: One for each activity. Enforce the requirement that the difference in time between the start node and the end node must be at least the duration of the activity.

  - For the example:

$$
\begin{aligned}
t_1 - t_0 &\geq d_A && \text{[for activity A]} \\
t_2 - t_0 &\geq d_B && \text{[for B]} \\
t_3 - t_1 &\geq 2 && \text{[for C: note constant duration]} \\
t_4 - t_1 &\geq d_D && \text{[for D]} \\
t_6 - t_5 &\geq 3 && \text{[for E]} \\
t_7 - t_4 &\geq d_F && \text{[for F]} \\
t_9 - t_8 &\geq 2 && \text{[for G]} \\
t_{10} - t_1 &\geq d_H && \text{[for H]} \\
t_{12} - t_{11} &\geq 1 && \text{[for I]}
\end{aligned}
$$

8 June

CPM LP formulation — 2

Network constraints — continued.

- Dummy constraints: One for each dummy edge in the CPM network. Enforce the precedence requirements represented by the dummy edges. (You can think of these as sequence constraints for activities having constant duration zero.)

  — For the example:

$$t_5 - t_2 \geq 0 \qquad t_{11} - t_7 \geq 0$$
$$t_5 - t_3 \geq 0 \qquad t_{11} - t_9 \geq 0$$
$$t_8 - t_4 \geq 0 \qquad t_{11} - t_{10} \geq 0$$
$$t_8 - t_6 \geq 0$$

- Duration constraints: A pair for each duration variable. Enforce the requirement that the duration be between the crash time and the usual time.

  — For the example:

[Note that each of these is actually two constraints: e.g., $d_A \geq 2$, $d_A \leq 4$.]

$$2 \leq d_A \leq 4 \qquad 2 \leq d_F \leq 3$$
$$1 \leq d_B \leq 3 \qquad 2 \leq d_H \leq 4$$
$$3 \leq d_D \leq 6$$

Other constraints are specific to the question being asked.

Often an expression for the <u>total speedup cost</u> appears, either as the objective function or as part of a constraint.

- For the example: Speeding up activity A from its usual time of 4 months to the duration d_A is a reduction of $4-d_A$ months, at a cost of \$4000 per month; so the speedup cost for activity A is $4000(4-d_A)$. Likewise for the other activities that can be sped up. So the total speedup cost is

$$4000(4-d_A) + 5000(3-d_B) + 4000(6-d_D) + 3000(3-d_F) + 4000(4-d_H).$$

LPs to answer questions about the example:

Least expensive way to reduce project completion time to 11 months?	Fastest project completion time given a budget of \$9000:
min [total speedup cost]	min t_{12} [project completion time]
s.t. $t_{12} \le 11$ [deadline]	s.t. [total speedup cost] ≤ 9000
[all network constraints]	[all network constraints]
All variables nonnegative.	All variables nonnegative.