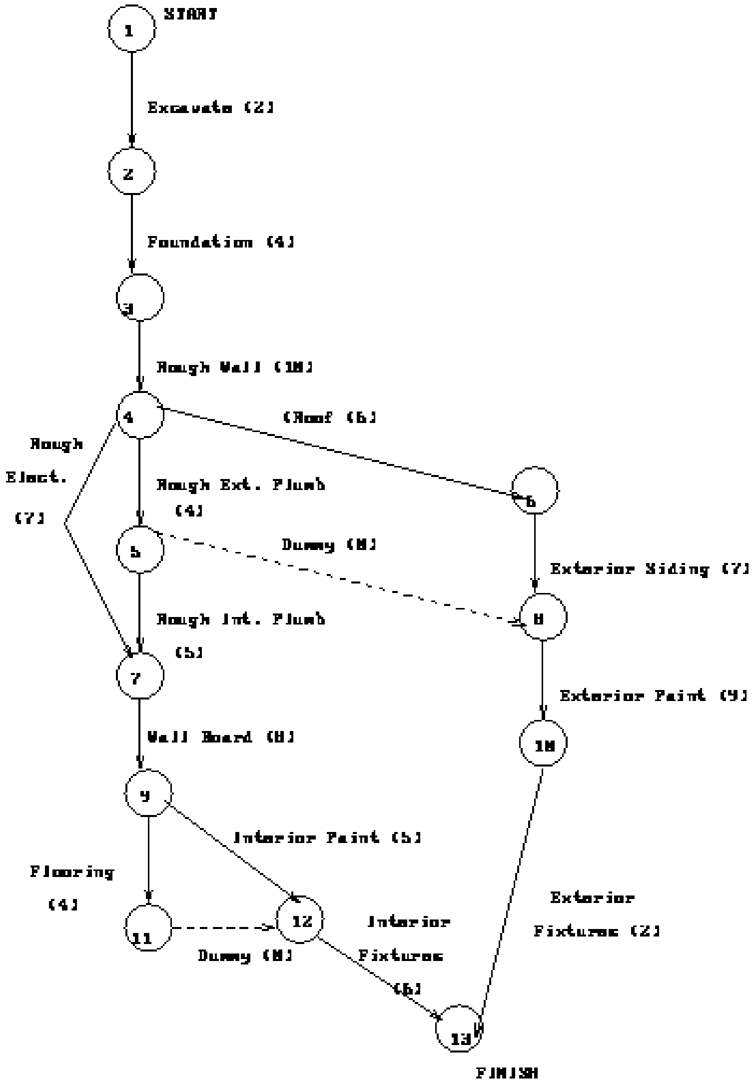
PERT/CPM

The final network models we discuss are the PERT and CPM models for project scheduling. The successful management of large projects, be they construction, transportation, financial, or what-have-you, relies on careful scheduling and coordinating of various tasks. CPM (Critical Path Method) and PERT (Project Evaluation and Review Technique) attempt to analyze project scheduling. This allows for better control and evaluation of the project. For instance, CPM allows you to answer such questions as:

- How long will the project take?
- When will we be able to start a particular task?
- If this task is not completed on time, will the entire project be delayed?
- Which tasks should we speed up (crash) in order to finish the project earlier?

CPM and PERT have been used in a wide variety of projects, from designing and producing computers to scheduling a complex surgery. The essential ideas behind these techniques are quite straightforward (not as complicated as maximum flow for instance). We will begin with CPM.

CPM uses a **project network** to portray graphically the relationships among the tasks in a project. This is illustrated in the following diagram for building a house. Here excavation must be done before foundation and the foundation done before the rough wall. After that, three tasks may be done (rough electrical, rough exterior plumbing, and roof).



Each arc of the project represents a task, or **activity**. This is the activity on arc (AOA) representation. Each node represents an event (like finishing foundation is node 3). The direction of the arcs gives the sequences in which activities must be achieved. An event must precede the beginning of successive activities, and an event occurs only when all the activities immediately preceding it occur.

There are two special nodes, the start node and the finish node. The start node has no activities enter it; the finish node has no activity leave it.

Each arc has two roles: it represents an activity and it defines the precedence relationships among the activities. Sometimes it is necessary to add arcs that only represent precedence relationships. These **dummy arcs** are represented by dashed arrows. In our example, the arc from 5 to 8 represents the fact that exterior plumbing must be completed in order to begin exterior painting.

There are generally rules restricting the form of a project network. Some of these are:

- An activity is represented by no more than one arc,
- Two nodes can be connected by no more than one arc,

- The project network can have no directed cycles.

To satisfy the second requirement, dummy nodes can be added (like node 11). Because of the third requirement, we can number the nodes so that if one node precedes another then the first gets a lower number than the second (as in our example).

Once we have the form of a project network, we can estimate the time required by each activity. Dummy activities get time 0. This number is in parentheses on the preceding diagram. These times are used to calculate two basic quantities: the **early time** and the **late time**.

The early time is the earliest time for an event assuming each activity is started as soon as possible. This is obtained by making a forward pass through the network. The start of the project is set to have early time 0. Every following event gets an early time based on the latest time a preceding event finishes. This is illustrated in table 3.1.

Event	Preceding Event	Early Time + Activity Time	Early Time
1	—	—	0
2	1	0+2	2
3	2	2+4	6
4	3	6+10	16
5	4	16+4	20
6	4	16+6	22
7	4	16+7	
	5	20+5	25
8	5	20+0	
	6	22+7	29
9	7	25+8	33
10	8	29+9	38
11	9	33+4	37
12	9	33+5	38
	11	37+0	
13	10	38+2	
	12	38+6	44

Table 1: Calculation of Early Time

The late time for an event is the latest time an event can occur without delaying the completion of the project beyond its early time. In this project, it takes 44 days to build a house. What is the latest the roofing can finish (or equivalently the latest exterior siding can start) without delaying the entire project?

The late time is calculated in a manner analogous to the early time. This time, begin at the end and work backwards. Here is the result for this project.

Event	Following Event	Late Time - Activity Time	Late Time
13	—	—	44
12	13	44–6	38
11	12	38–0	38
10	13	44–2	42
9	12	38–5	33
	11	38–4	
8	10	42–9	33
7	9	33–8	25
6	8	33–7	26
5	8	33–0	
	7	25–5	20
4	7	25–7	
	6	26–6	
	5	20–4	16
3	4	16–10	6
2	3	6–4	2
1	2	2–2	0

Table 2: Calculation of Late Time

Therefore, roofing must be done by time 26 or the completion of the building will be delayed. Note that it can finish as early as 22 days and as late as 26 days without affecting the completion time. Associated with each activity is a value called the **total float** for the activity. The total float for an activity is the amount of time an activity can be delayed beyond its earliest time without affecting the final completion time of the project. The total float for an arc from **i** to **j** is the difference between the late time of job **j** and (the early time for job **i** plus the activity time).

An alternative measure for the flexibility of an activity is the **free float**. This is the amount of time an activity can be delayed without pushing *any* job past its early time. This is calculated as the difference between the early time of job **j** and (the early time of job **i** plus the activity time). The following table illustrates this concept.

Activity	Total Float	Free Float
(1,2)	2–(0+2) = 0	2–(0+2) = 0
(2,3)	6–(2+4) = 0	6–(2+4) = 0
(3,4)	16–(6+10) = 0	16–(6+10 = 0
(4,5)	20–(16+4) = 0	20–(16+4) = 0
(4,6)	26–(16+6) = 4	22–(16+6) = 0
(4,7)	25–(16+7) = 2	25–(16+7) = 2
(5,7)	25–(20+5) = 0	25–(20+5) = 0
(6,8)	33–(22+7) = 4	29–(22+7) = 0
(7,9)	33–(25+8) = 0	33–(25+8) = 0
(8,10)	42–(29+9) = 4	38–(29+9) = 0
(9,11)	38–(33+4) = 1	37–(33+4) = 0
(9,12)	38–(33+5) = 0	38–(33+5) = 0
(10,13)	44–(38+2) = 4	44–(38+2) = 4
(12,13)	44–(38+6) = 0	44–(38+6) = 0

Table 3: Calculation of Float

If an activity with a total float of 0 is delayed for whatever reason, then the entire project is delayed. Such an activity is called a **critical activity**. Any path (there may be more than one) from the start node to the finish node made up solely of critical activities is a **critical path**. In our example, a critical path is 1-2-3-4-5-7-9-12-13.

This information on early and late times, critical jobs, and critical paths in invaluable to a manager, for it allows her to investigate the effect of possible improvements to the project plan.

Using Linear Programming

It is possible to use linear programming to find a critical path and to determine the overall completion time. Although this approach is much slower than the above algorithm, we will be able to extend the model to identify activities that should be ‘‘crashed’’ (speeded up, perhaps at some cost).

To do this, let x_j be the time that event **j** occurs. The objective, for this example is to minimize $x_{13} - x_1$. Each activity corresponds to a constraint. For instance, the activity between 1 and 2 is the constraint

$$x_2 \geq x_1 + 2.$$

The other constraints are

$x_3 \geq x_2 + 4$
$x_4 \geq x_3 + 10$
$x_5 \geq x_4 + 4$
$x_6 \geq x_4 + 6$
$x_7 \geq x_4 + 7$
$x_7 \geq x_5 + 5$
$x_8 \geq x_5 + 0$
$x_8 \geq x_6 + 7$
$x_9 \geq x_7 + 8$
$x_{10} \geq x_8 + 9$
$x_{11} \geq x_9 + 4$
$x_{12} \geq x_9 + 5$
$x_{12} \geq x_{11}$
$x_{13} \geq x_{10} + 2$
$x_{13} \geq x_{12} + 6$

If x_1 is set to 0, optimizing this linear program will give the optimal completion time. Furthermore, one critical path will be identified. The arcs on this path will correspond to constraint with a shadow price of -1.

- Crashing the Project

Michael A. Trick
Wed Nov 8 14:14:52 EST 1995