Critical Path Method (Chapter 10)

CPM can be used to help with the scheduling of big projects.

**PROJECT**
- defined by a list of activities. Each activity has
  1. an estimated completion time
  2. a set of precedence relationships b/w activities

**CPM GOAL**
- determine minimum completion time
- highlight critical activities (those that most affect completion time)

Firstly we develop the project network:

We need:
(i) the list of jobs
(ii) an estimate of the duration of each job
(iii) the predecessors of each job (those jobs that must be completed before this job can commence)

(*NO CYCLES ALLOWED*)

We use event network approach (activity-on-arc representation):
- activities are represented by directed arcs
  with values = duration of activity
- precedence relationship established through nodes
  → these represent events: - completion OR both!
  of some set of activities

NB: NETWORKS ARE NOT UNIQUE
→ WE AIM TO GET A REASONABLY EFFICIENT ONE

**NOTATION and DEFINITIONS**

\( t_{m,n} \) - duration of job represented by arc \((m,n)\)

**T** → earliest completion time of entire project

**ET(n)** - early event time for node \(n\)
- is the earliest time that event corresponding to node \(n\) can occur (all preceding jobs completed)

**LT(n)** - late event time for node \(n\)
- is the latest time that event corresponding to node \(n\) can occur without delaying \(T\)

Clearly: \(LT(n) = T = ET(n)\)

\(n\) nodes in network

→ to compute the latest event times for all other nodes we work recursively backwards through the network

**TF(m,n)** - total float corresponding to arc \((m,n)\)
- amount which time duration of activity can be increased without delaying completion of entire project

**CRITICAL ACTIVITY**
- those activities whose total float is 0.

**CRITICAL PATH**
- path from node 1 to node \(N\) consisting of critical activities

**FF(m,n)** - free float corresponding to arc \((m,n)\)
- amount which the duration of activity can be increased without delaying the start of subsequent activity beyond its earliest possible starting time
\[ \begin{align*} 
ET(i) &= 0 \\
ET(n) &= \max_{m \in P(n)} \{ ET(m) + t_{m,n} \}, \quad n=2, \ldots, N \\
LT(N) &= T = ET(N) \\
LT(n) &= \min_{m \in E(n)} \{ LT(m) - t_{n,m} \}, \quad n=1, \ldots, N-1
\end{align*} \]

(NB: LT(1) must always equal 0.)

\[ TF(m,n) = LT(n) - ET(m) - t_{m,n} \]

- latest possible completion
- earliest possible commencement
- duration

\[ FF(m,n) = ET(n) - ET(m) - t_{m,n} \]

**EXAMPLE:**

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>DURATION</th>
<th>IMMEDIATE PREDECESSORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>5</td>
<td>A, B</td>
</tr>
<tr>
<td>D</td>
<td>11</td>
<td>A, B</td>
</tr>
<tr>
<td>E</td>
<td>8</td>
<td>C, E</td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>C, E</td>
</tr>
<tr>
<td>G</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**RESTRICTIONS/RULES**
- tell us how the project network must ultimately look
- two nodes can be connected by at most one arc
  (introduce dummy activity of zero duration)
- each activity represented by only one arc

**EVENT NETWORK:**

- This network has:
  - one origin node (node 1)
  - one terminal node (node 6)
  - each activity ends at a higher labelled node than from which it began.

- Thus this network representation for the project is a valid project network.

- Need to calculate ET(n), LT(n), TF(m,n) & FF(m,n)
  - for all nodes
  - for all activities

  \[ \rightarrow \text{once we have all } ET(n) \text{ and } LT(n), \text{ fill these in} \]

  \[ \rightarrow \text{state the critical activities} \]

  \[ \rightarrow \text{write down the critical path.} \]