TD 608
Project Management and Analysis

Part I
Project Conception and Execution

Milind Sohoni

Lecture 6

The Scheduling Problem

Recall:

ID	Duration	After
VC1	1 mo	-
VC2	1 mo	VC1
AP	0.5 mo	VC1
SG	1 mo	VC1
SP	1 mo	SG, AP
TP	0.5 mo	SP
F	2 mo	AP
W	2 mo	SP
T	0.5 mo	TP, W
FP	0.5 mo	T, VC2 W, F, SP
	Target 10	

Note: the target end-time is set to an abstract number, say 10.

Objectives:

- Compute feasible start and end times
  - No task should start before its precedences have ended.
- Identify the critical path
  - Tasks for which delays will impact the project
- Compute slacks
  - maximum delays for non-critical tasks

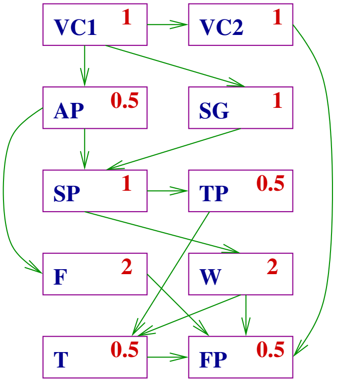
- We could do this *ad hoc* for our small project.
- For larger projects, it needs a systematic procedure.

CPM is one such scheme.

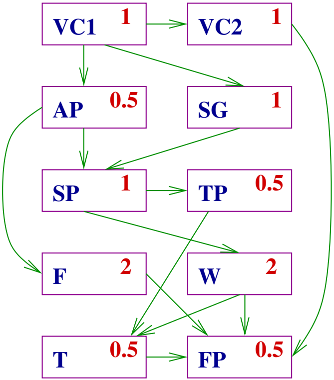
The Scheduling Problem

Recall:

ID	Duration	After
VC1	1 mo	-
VC2	1 mo	VC1
AP	0.5 mo	VC1
SG	1 mo	VC1
SP	1 mo	SG, AP
TP	0.5 mo	SP
F	2 mo	AP
W	2 mo	SP
T	0.5 mo	TP, W
FP	0.5 mo	T, VC2 W, F, SP
	Target 10	



The Activity Graph

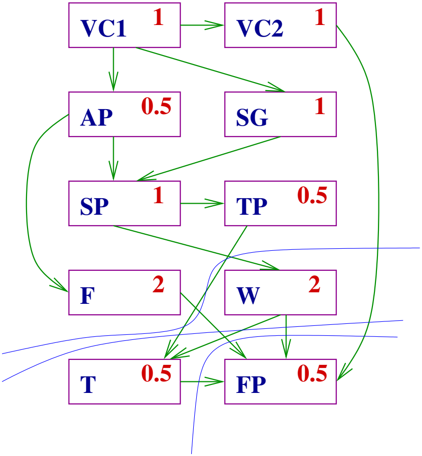


We will construct a *graph* where every node is an activity and every directed edge is a precedence :

Note that:

- The relevant data about an activity, i.e., its label and its duration is stored at the node.
- If our modellling is correct, then the graph has no cycles
- In this case, there is an activity, *viz.* FP which has no successor.

Starting the Sequencing



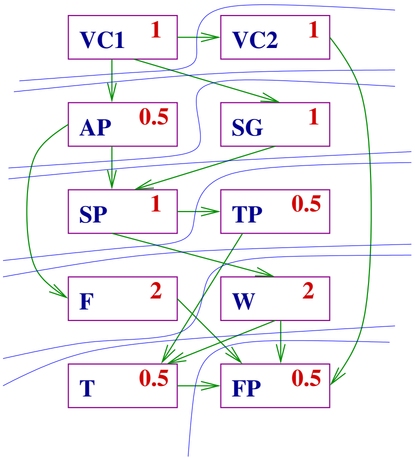
Next , we repeatedly

- We remove a node which has no successors
- we remove all edges leading into this node
- We maintain the sequence in which we remove these nodes

If the graph has no cycles, then this process terminates by exhausting all nodes!

Sequence: : FP, T, W

Getting the sequence



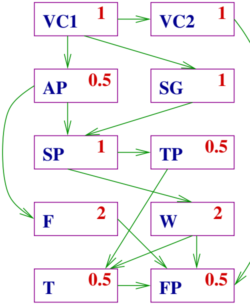
Next , we repeatedly

- We remove a node which has no successors
- we remove all edges leading into this node
- We maintain the sequence in which we remove these nodes

If the graph has no cycles, then this process terminates by exhausting all nodes!

Sequence : : FP, T, W, F, TP, SP, SG, AP, VC2, VC1

Starting the assignment

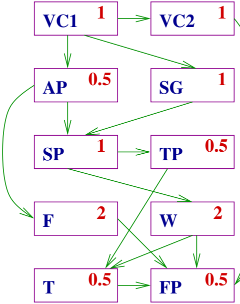


We go back to the graph and produce a table of the nodes in the reverse order of the sequence.

ID	Duration	After	Start	End
FP	0.5 mo	T, VC2 W, F, SP	9.5	10
	Target 10			

- We copy the target time as the end-time of the last activity
- The start time is `endtime-duration`=9.5

The next few nodes

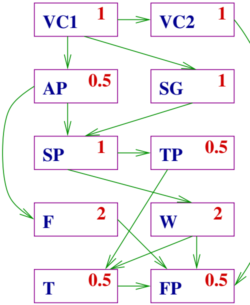


Sequence : : FP, T, W, F, TP, SP, SG, AP, VC2, VC1

ID	Duration	After	Start	End
T	0.5 mo	TP, W	9	9.5
FP	0.5 mo	T, VC2 W, F, SP	9.5	10

- For the next task, we se all its successors
- In this case, for T, the only successor is FP
- We put the end-time as the earliest of all succesor start-times.
- We put the start time as `endtime-duration`.

Continuing like this ...



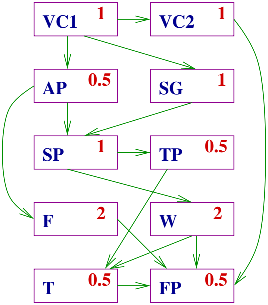
Sequence : : FP, T, W, F, TP, SP, SG, AP, VC2, VC1

ID	Duration	After	Start	End
SP	1 mo	SG, AP	6	7
TP	0.5 mo	SP	8.5	9
F	2 mo	AP	7.5	9.5
W	2 mo	SP	7	9
T	0.5 mo	TP, W	9	9.5
FP	0.5 mo	T, VC2 W, F, SP	9.5	10

Coming to SP, we see that:

- Successors are TP and W, with start-times 7 and 8.5.
- Thus 7 is the minimum and the end-time of SP.
- The end-time is as before.

Finally

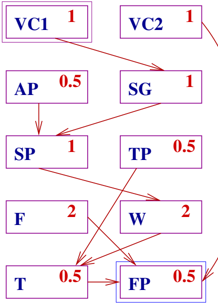


ID	Duration	After	Start	End
VC1	1 mo	-	4	5
VC2	1 mo	VC1	8.5	9.5
AP	0.5 mo	VC1	5.5	6
SG	1 mo	VC1	5	6
SP	1 mo	SG, AP	6	7
TP	0.5 mo	SP	8.5	9
F	2 mo	AP	7.5	9.5
W	2 mo	SP	7	9
T	0.5 mo	TP, W	9	9.5
FP	0.5 mo	T, VC2 W, F, SP	9.5	10

The earliest start-time is the project start-time.

- We finally obtain the start and end times of all activities.
- Note that this is quite different from the *ad hoc* schedule we had obtained earlier, e.g., VC2.

Critical Path

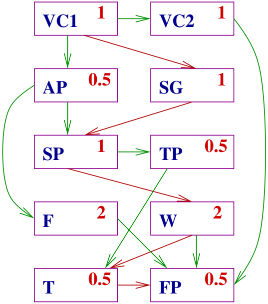


Crtical Path are all activities wherein delays will affect the whole project.

- All activities with the same start-time as the project start are initial activities.
- All activities with the same end-time as the project end are final activities.
- All edges connecting activities which do not have a gap are called critical edges.
- All activites which lie on any path from initial to final tasks along critical edges, are critical activities

Here, the critical path has VC1, SG, SP, W, T, FP, same as before!

Slacks



Slack : Window in which an activity needs to be completed without affecting project time.

- For any activity locate all edges coming in. The latest end-time of these is the start-time of the window.
- For any activity locate all edges going out. The earliest start-time of these is the end-time of the window.
- For every critical activity, the interval is the alloted window, and there is no slack.
- For a non-critical activity, the window is larger. For example, for AP:
  - VC1 comes in, and ends at 5.
  - SP, F do out and start at 6, 7.5.
  - window is [5,6]
- Thus AP of duration 0.5 may start anytime between [5,0.5]

Variations and other Mobilization Problems

We have seen how to schedule from the project-finish. A similar analysis can be made from the project-start end.

The basic points are:

- Assumes a unique event without a predecessor.
- Peels of one event after another, without predecesors. This gives us the sequence.
- Start-times of activities alloted as latest end-times of predecessors.

The general model is of course, Linear Programming, OR

PERT

- A variation of the CPM method which allows for uncertainty in the durations.

The Assignment Problem

- Efficient allocation resources to activities.

Inventory control and ware-housing

- Planning the availability of resources while minimizing storage costs.

Sourcing and Supply

- Optimum purchase strategy from various suppliers under different cost regimes.

A Problem and Variation

Conisder the following task list:

Id	Dur.	After	Labour
A	3	-	3
B	2	A	1
C	2	A	4
D	1	A	3
E	6	B,C,D	7
F	3	D	4
G	3	E,F	7
H	1	E,F	1
I	5	F	6
J	2	G,H,I	2

The last column indicates the units of labour which are needed to be deployed for the duration of the task.

- Compute a minimum time duration schedule, assuming a start-time of 0.
- What are the critical paths in this schedule.
- Based on this schedule, compute the total labour requirement at any time moment.
- Is there any other schedule of minimum time where the peak labour requirement is lower?