Project Management

Lecture 2b

Network Planning Techniques: CPM-PERT

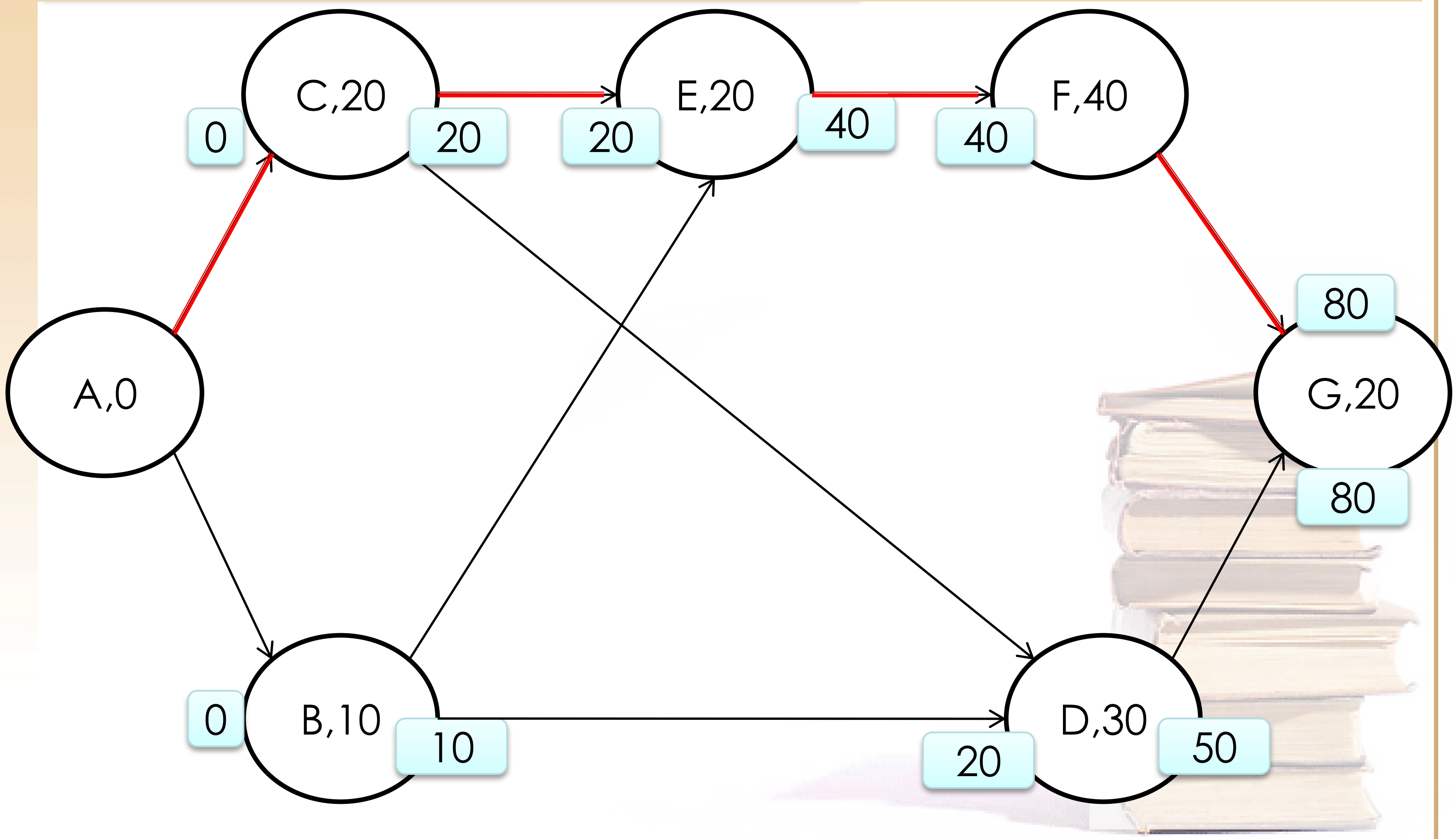
Instructor

Carmi Bogot

Critical Path Algorithm

- For large projects there are many paths
- An algorithm is needed to identify the CP efficiently
- Develop information about each task in the context of the project
- Times
  - Start time (S)
  - For each job: earliest start (ES)
    - When all predecessors are completed
  - Job duration t
  - Earliest Finish (EF) = (ES)+t
- Show algorithm using project graph

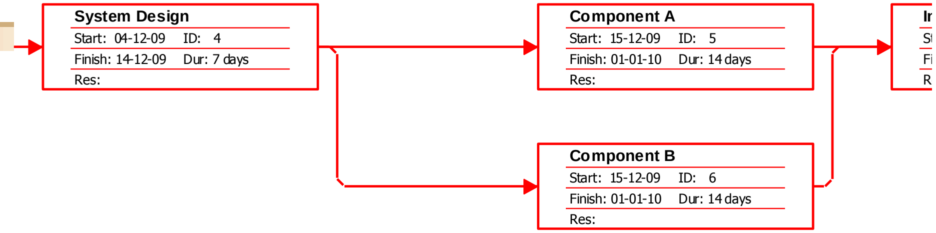
CP Algorithm - Graphical



Latest Start and Finish Times

- Set target finish time T >= F
- Clean car by Pesach
- When is the latest the project can be started?
- Late Finish (LF)
  - Latest time a job can be finished without delaying the project beyond target time T
- Late Start
  - LS = LF - t

Project Graph



- Each task drawn on a graph as circle or box
- Connect each job with predecessors unidirectional arrows
- No jobs before "start"
- No jobs after "finish"
- Total time of each path is the sum of the job times
- Longest total time -> "critical path"
- There can be multiple critical paths

Critical Path

- CP is the bottleneck
- Working on CP tasks affect project finish
- Duration of non-critical is not important
- “Crashing” focus on the few jobs in the CP
- How can a task be shortened
  - Can 9 women make a baby in a month?
- After completion a new CP might pop up
- Lengthening non critical tasks can also shorten he critical path

Slack

- With some tasks ES=LS -> no slack
- Total Slack of a task TS=LS-ES
- **Maximum amount of time a task may be delayed beyond its early start without delaying the project completion**
- Slack is an important managerial tool
- When T=F all critical tasks have TS=0

Main CPM Errors

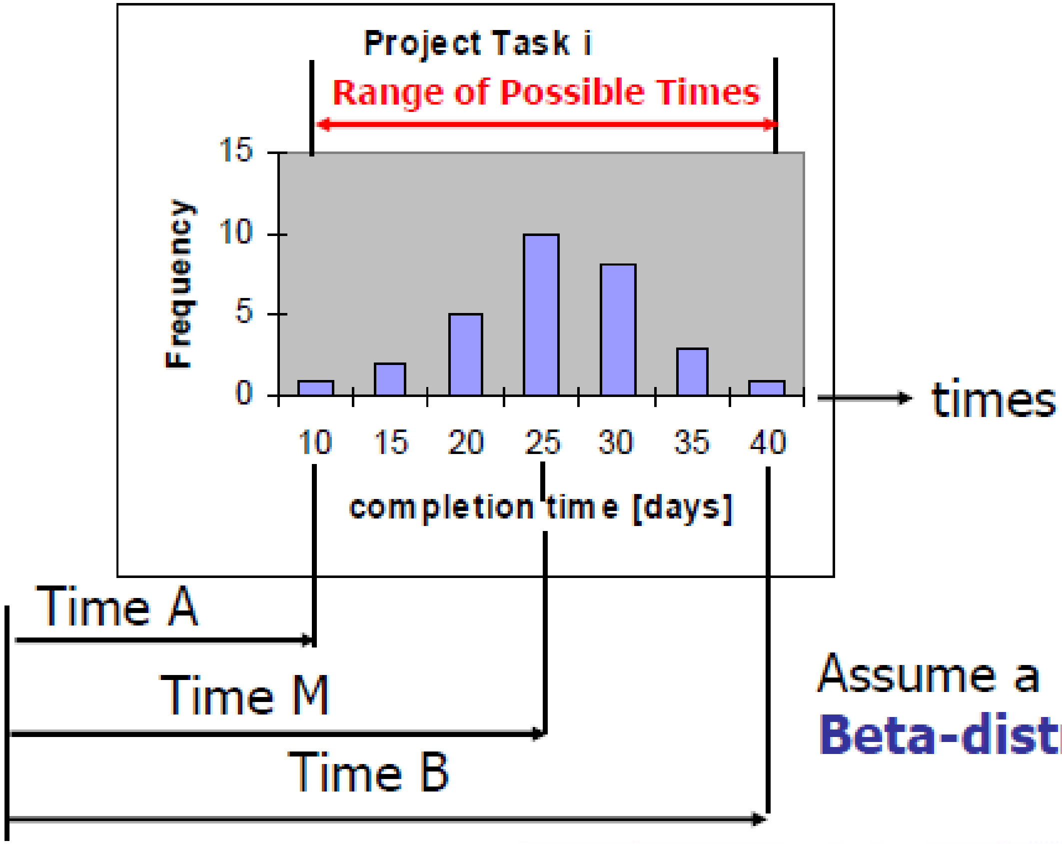
- Estimated job times are wrong
- Predecessor relationships contain cycles
- List of prerequisites contains more that immediate predecessors
  - (a) -> b, (b)->c, (a, b) -> c
- Missing tasks
- Missing dependancies

How long does a task take?

CPM versus PERT

- Difference how task duration is treated
- CPM assumes time estimates are <u>deterministic</u>
  - Obtain task duration form previous projects
  - Suitable for “construction” like projects
- PERT treats durations as probabilistic
  - PERT= CPM + probabilistic task times
  - Better to R&D
  - Captures schedule and risk

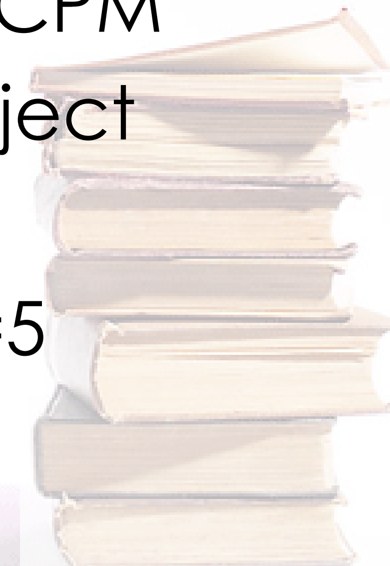
A-M-B Time Estimates



Assume a **Beta-distribution**

Expected Time and Variance

- Mean expected Time (TE) $$TE = \frac{A + 4M + B}{6}$$
- Time Variable (TV) $$TV = \sigma_t^2 = \left(\frac{B - A}{6}\right)^2$$
- EF and LF computed as with CPM
- Set T=F for the end of the project
- Assume Gaussian Distribution
- Example A=3, B=7 M=5 -> TE=5



Cost Calculations'

- Project cost can be computed if the cost for each task is known
- Direct costs will increase if we “crash: critical tasks
- Indirect (fixed , overhead) will decrease as the project gets shorter.
- Minimize sum of direct and fixed costs

Summary

- CPM is useful, despite criticism, to identify the critical path and focus on it
- Slack is precious
- PERT treats time as probabilistic
- Selective “crashing” can reduce total project cost
- CPM and PERT do not allow task iterations

HW1 Introduction

- You are Project Manager for a project to develop a corporate web site
- Plan the project
  - Task list
  - Project graph
  - Critical path
  - Slack times
  - “managerial” issues