Algoritmos em Grafos: Caminho mais curto

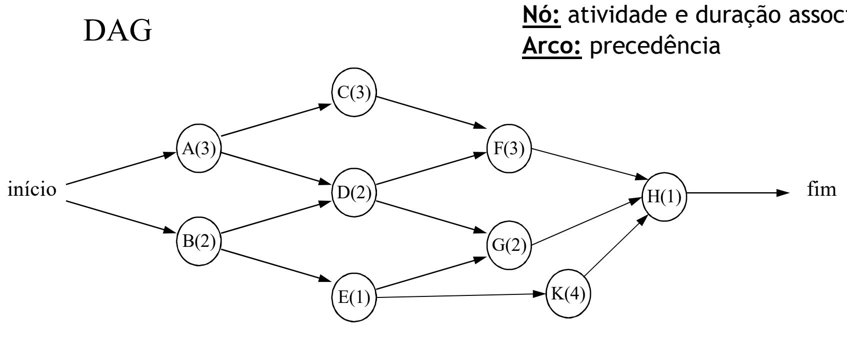
Aplicação à gestão de projetos

R. Rossetti, L. Ferreira, H. L. Cardoso, F. Andrade

FEUP, MIEIC, CAL

Grafo Nó-Atividade

Nó: atividade e duração associada
Arco: precedência



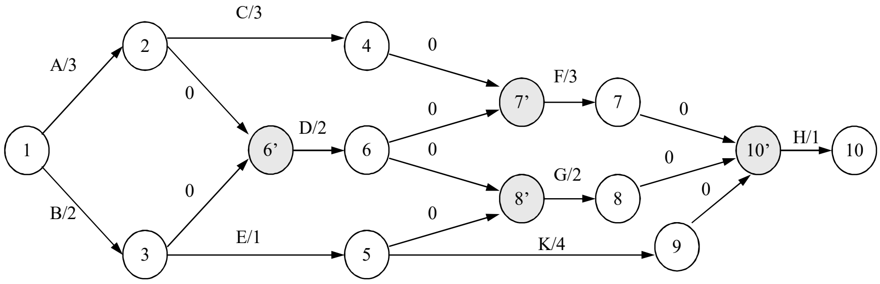
Qual a duração total mínima do projeto?

Que atividades podem ser atrasadas e por quanto tempo (sem aumentar a duração do projeto)?

Reformulação em Grafo Nó-Evento

DAG

Nó: evento – completar atividade
Arco: atividade e duração

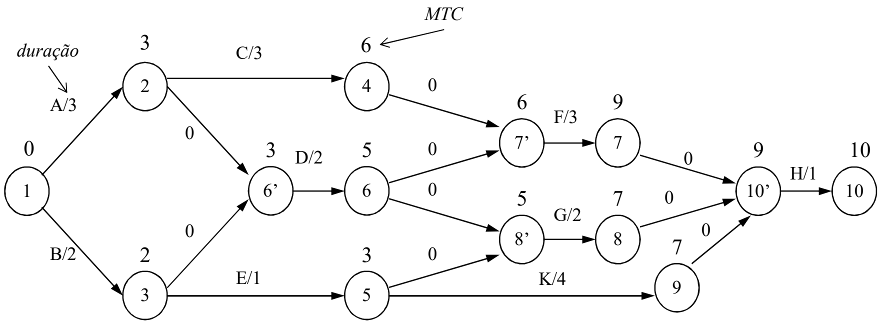


Introduzem-se nós e arcos extra para garantir precedências no caso de atividades com mais que uma antecessora

Menor Tempo de Conclusão

- **menor tempo de conclusão de uma atividade**
  ⇔ caminho mais comprido do evento inicial ao nó de conclusão da atividade
- adaptar algoritmo de caminho mais curto para grafos acíclicos
  - MTC(1) = 0
  - MTC(w) = max { MTC(v) + c(v,w) | (v, w) $\in$ E}

MTC : usar ordem topológica

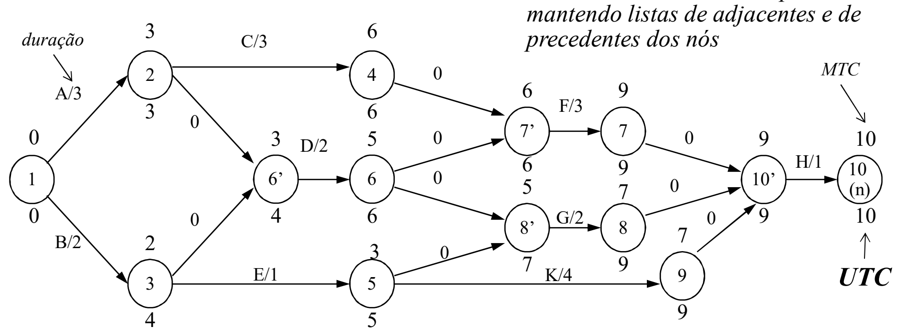


Último Tempo de Conclusão

- **último tempo de conclusão:** mais tarde que uma atividade pode terminar sem comprometer as que se lhe seguem
  - $UTC(n) = MTC(n)$
  - $UTC(v) = \min\{ UTC(w) - c(v, w) \mid (v, w) \in E\}$

UTC : usar ordem topológica inversa

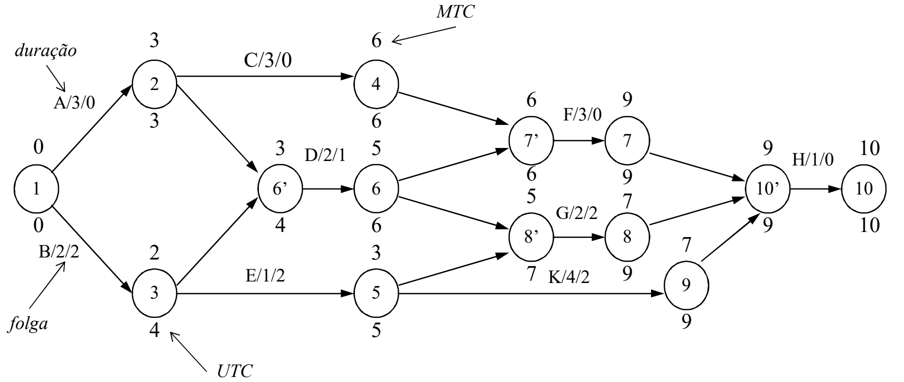
valores calculados em tempo linear mantendo listas de adjacentes e de precedentes dos nós



Folgas nas atividades

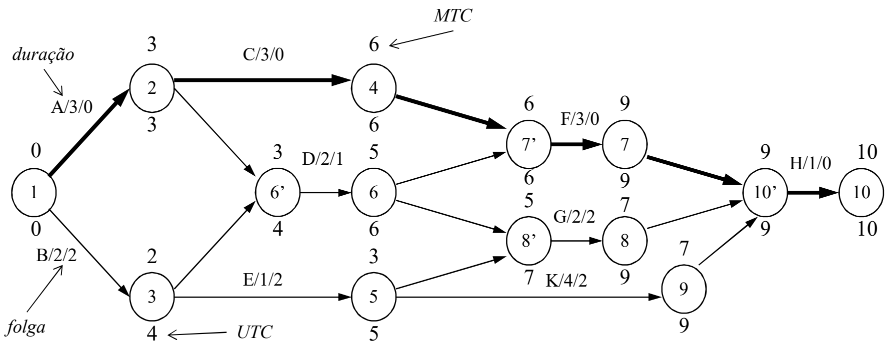
- **folga da atividade**

  $folga(v,w) = UTC(w) - MTC(v) - c(v,w)$



Folgas nas atividades

- **folga da atividade**
  folga(v,w) = UTC(w)-MTC(v)-c(v,w)



Caminho crítico: só atividades de folga nula (há pelo menos 1)

Referências e mais informação

- "Data Structures and Algorithm Analysis in Java", Second Edition, Mark Allen Weiss, Addison Wesley, 2006
- "Introduction to Algorithms", Second Edition, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, The MIT Press, 2001