FAI LAB 11: Scheduling

Paolo Morettin

2024-25

Planning with time and resources

Planning with time and resources

- **Scheduling**: assigning a start time to a set of actions that:
  - Have a **duration**
  - May **depend on** other actions (forming **jobs**)
  - Require consumable/reusable **resources**

Planning with time and resources

- **Scheduling**: assigning a start time to a set of actions that:
    - Have a **duration**
    - May **depend on** other actions (forming **jobs**)
    - Require consumable/reusable **resources**
- **Approach**:

(1) find a partially-ordered plan (regardless of the duration)

(2) schedule the actions

Planning with time and resources

- **Scheduling**: assigning a start time to a set of actions that:
  - Have a **duration**
  - May **depend on** other actions (forming **jobs**)
  - Require consumable/reusable **resources**
- **Approach**:

  (1) find a partially-ordered plan (regardless of the duration)

  (2) schedule the actions
- **GOAL**: Find a schedule that satisfies all the constraints while minimizing the **makespan** (total duration)

The Critical-Path Method

- Each action has:
    - **ES** earliest possible start time
    - **LS** latest possible start time
- Find $[ES, LS]$ given a **partially-ordered plan**

The Critical-Path Method

- Each action has:
  - **ES** earliest possible start time
  - **LS** latest possible start time
- Find $[ES, LS]$ given a **partially-ordered plan**
- They can be computed recursively:

The Critical-Path Method

- Each action has:
  - **ES** earliest possible start time
  - **LS** latest possible start time
- Find $[ES, LS]$ given a **partially-ordered plan**
- They can be computed recursively:

$$ES(Start) = 0$$

The Critical-Path Method

- Each action has:
  - **ES** earliest possible start time
  - **LS** latest possible start time
- Find $[ES, LS]$ given a **partially-ordered plan**
- They can be computed recursively:

$$ES(Start) = 0$$
$$ES(B) = max_{A \mid A \prec B}[ES(A) + D(A)]$$

The Critical-Path Method

- Each action has:
  - **ES** earliest possible start time
  - **LS** latest possible start time
- Find $[ES, LS]$ given a **partially-ordered plan**
- They can be computed recursively:

$$
\begin{aligned}
&ES(Start) = 0 \\
&ES(B) = max_{A \mid A \prec B}[ES(A) + D(A)] \\
&LS(Finish) = ES(Finish)
\end{aligned}
$$

The Critical-Path Method

- Each action has:
  - **ES** earliest possible start time
  - **LS** latest possible start time
- Find $[ES, LS]$ given a **partially-ordered plan**
- They can be computed recursively:

$$
\begin{aligned}
ES(Start) &= 0 \\
ES(B) &= max_{A \mid A \prec B}[ES(A) + D(A)] \\
LS(Finish) &= ES(Finish) \\
LS(A) &= min_{B \mid A \prec B}[LS(B) - D(A)]
\end{aligned}
$$

The Critical-Path Method

- Each action has:
  - **ES** earliest possible start time
  - **LS** latest possible start time
- Find $[ES, LS]$ given a **partially-ordered plan**
- They can be computed recursively:

$$
\begin{aligned}
ES(Start) &= 0 \\
ES(B) &= max_{A \mid A \prec B}[ES(A) + D(A)] \\
LS(Finish) &= ES(Finish) \\
LS(A) &= min_{B \mid A \prec B}[LS(B) - D(A)]
\end{aligned}
$$

- **Slack**$(A) = LS(A) - ES(A)$
- An action A is in the **critical path** IFF $Slack(A) = 0$

The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$

$$A \prec C \prec D \prec E$$

$$F \prec G$$

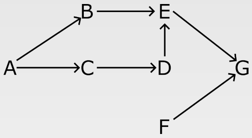
The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$

$$A \prec C \prec D \prec E$$

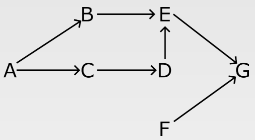
$$F \prec G$$



The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$
$$A \prec C \prec D \prec E$$
$$F \prec G$$



$ES(A) = 0$

$ES(B) = 1$

$ES(C) = 1$

$ES(D) = 3$

$ES(E) = 8$

$ES(F) = 0$

$ES(G) = 10$

$ES(Finish) = 12 = LS(Finish)$

The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$
$$A \prec C \prec D \prec E$$
$$F \prec G$$

$$\begin{tikzcd}
 & B \arrow[r] & E \arrow[dr] & \\
A \arrow[ur] \arrow[r] & C \arrow[r] & D \arrow[u] & G \\
 & & F \arrow[ur] &
\end{tikzcd}$$

$ES(A) = 0$ $LS(A) = 1$

$ES(B) = 1$ $LS(B) = 6$

$ES(C) = 1$ $LS(C) = 2$

$ES(D) = 3$ $LS(D) = 4$

$ES(E) = 8$ $LS(E) = 9$

$ES(F) = 0$ $LS(F) = 0$

$ES(G) = 10$ $LS(G) = 10$

$ES(Finish) = 12 = LS(Finish)$

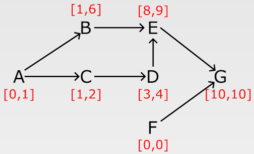
The Critical-Path Method: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$A \prec B \prec E \prec G$

$A \prec C \prec D \prec E$

$F \prec G$



Makespan = ?

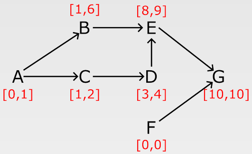
The Critical-Path Method: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$A \prec B \prec E \prec G$

$A \prec C \prec D \prec E$

$F \prec G$

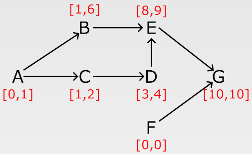


Makespan $=$ 12

The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$
$$A \prec C \prec D \prec E$$
$$F \prec G$$



Makespan = 12

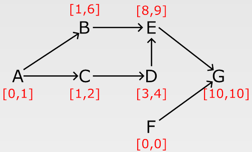
How does the makespan changes if:

- $D(F)$ is increased by 3?

The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$
$$A \prec C \prec D \prec E$$
$$F \prec G$$



Makespan $= 12$

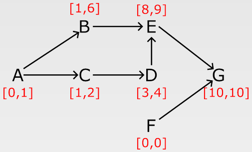
How does the makespan changes if:

- $D(F)$ is increased by 3? **$+3$**

The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$
$$A \prec C \prec D \prec E$$
$$F \prec G$$



Makespan $= 12$

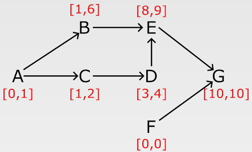
How does the makespan changes if:

- $D(F)$ is increased by 3? **$+3$**
- $D(B)$ is increased by 3?

The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$
$$A \prec C \prec D \prec E$$
$$F \prec G$$



Makespan $= 12$

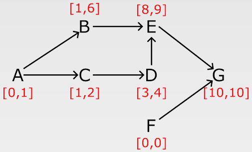
How does the makespan changes if:

- $D(F)$ is increased by 3? **$+3$**
- $D(B)$ is increased by 3? **Same**

The Critical-Path Method: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$$A \prec B \prec E \prec G$$
$$A \prec C \prec D \prec E$$
$$F \prec G$$



Makespan $= 12$

How does the makespan changes if:

- $D(F)$ is increased by 3? **+3**
- $D(B)$ is increased by 3? **Same**
- $D(B)$ is increased by 8?

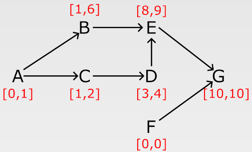
The Critical-Path Method: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

$A \prec B \prec E \prec G$

$A \prec C \prec D \prec E$

$F \prec G$



Makespan $= 12$

How does the makespan changes if:

- $D(F)$ is increased by 3? **$+3$**
- $D(B)$ is increased by 3? **Same**
- $D(B)$ is increased by 8? **$+3$**

The Critical-Path Method: exercises

Action:	*Start*	A	B	C	D	E	F	G	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

Same durations as before, but with dependencies:

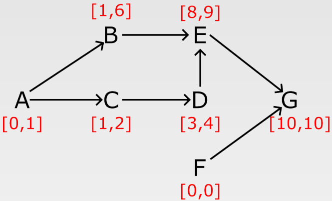
$$A \prec C$$
$$B \prec E \prec G$$
$$D \prec E$$
$$F \prec G$$

Adding resources

- No required resources: scheduling is computationally **easy**
- Finite resources makes it **much harder**
- The output of the CPM can guide the search

Adding resources: exercises

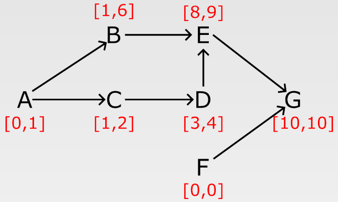
Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0



We now have **two** reusable resources. Each action requires **one**.

Adding resources: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

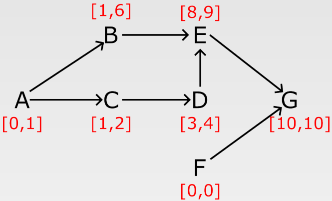


We now have **two** reusable resources. Each action requires **one**.

Let's try lexicographic order first ($Y > X$ can be allocated to R_i iff X was already allocated)

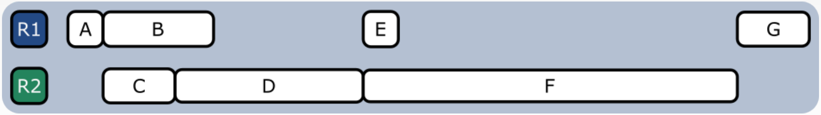
Adding resources: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0



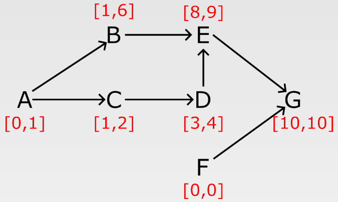
We now have **two** reusable resources. Each action requires **one**.

Let's try lexicographic order first ($Y > X$ can be allocated to R_i iff X was already allocated) **Makespan $=$ 20**



Adding resources: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0

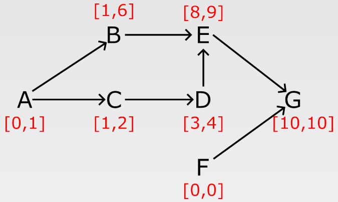


We now have **two** reusable resources. Each action requires **one**.

Let's actually consider the output of CPM.

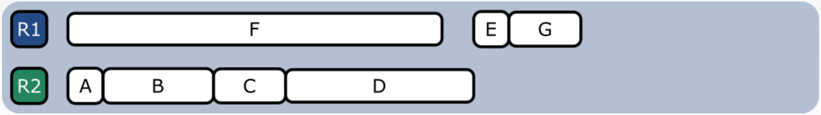
Adding resources: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0



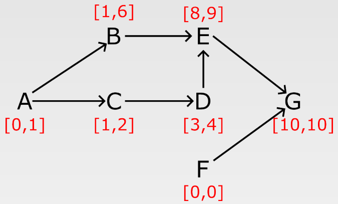
We now have **two** reusable resources. Each action requires **one**.

Let's actually consider the output of CPM. **Makespan $=$ 14**



Adding resources: exercises

Action:	*Start*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*Finish*
Duration:	0	1	3	2	5	1	10	2	0



We now have **two** reusable resources. Each action requires **one**.

Let's actually consider the output of CPM. **Makespan = 14**