

§2 The deterministic project scheduling model

Every job j has a fixed deterministic processing time $x_j \in \mathbb{R}_+$

→ processing time vector $x = (x_1, x_2, \dots, x_n) \in \mathbb{R}_+^n$

Note: usual notation is p_j for processing time

we need the p for probabilities

we use x_j because processing times are seen as variables

no preemption (no interruption of jobs)

precedence constraints given by a partial order G G or $<$
used synonymously

schedule $S =$ vector $S = (S_1, \dots, S_n)$ of start times for the jobs

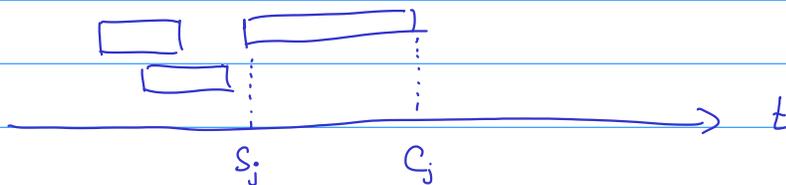
$$S_i \geq 0$$

S respects G iff $[i < j \Rightarrow S_i + x_i \leq S_j]$

j must wait for the completion of i

$C_j := S_j + x_j$ is the completion time of j

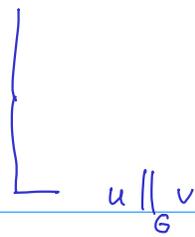
Schedules are represented by Gantt charts



resource constraints are modeled by a system

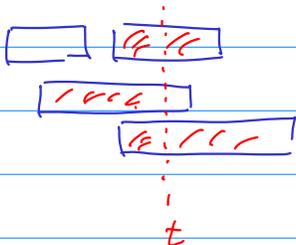
$\mathcal{F} = \{F_1, F_2, \dots, F_k\}$ of forbidden sets or bottlenecks

Each F_i is an antichain of G



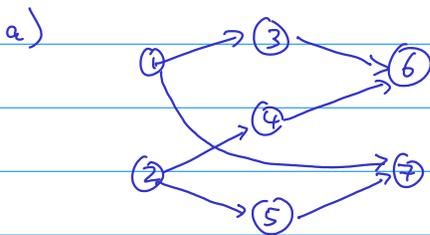
that must not be scheduled simultaneously at any moment in time during project execution, but every proper subset can

A schedule S respects \mathcal{F} if, for every $F_i \in \mathcal{F}$, for every t $\{j \mid s_j < t < c_j\} \not\subseteq F_i$



2.1 lemma: Resource constraints given by forbidden sets model precisely constant resource requirements and constant availabilities
 i.e. constant amount during processing of a job constant availability during project execution

Proof by example



j	1	2	3	4	5	6	7
$r_1(j)$	2	1	1	-	-	-	-
$r_2(j)$	-	1	1	1	2	2	2

↑
constant requirements

availabilities $R_1 = 2$ units of resource 1 } availabilities
 $R_2 = 3$... 2

$$\Rightarrow \mathcal{F} = \left\{ \underbrace{\{1,2\}}_{F_1} \underbrace{\{5,6\}}_{F_2} \underbrace{\{6,7\}}_{F_3} \underbrace{\{3,4,5\}}_{F_4} \underbrace{\{3,4,7\}}_{F_5} \right\}$$

Every system \mathcal{F} (with no $F_i \subseteq F_j$) of antichains of G can be obtained in this way, even with $\tau_i(j) \in \{0,1\}$

j	1	2	3	4	5	6	7	
$\tau_1 \stackrel{\Delta}{=} F_1$	1	1						$R_1 = 1$
τ_2					1	1		1
τ_3						1	1	1
τ_4			1	1	1			$R_4 = 2$
$\tau_5 \stackrel{\Delta}{=} F_5$			1	1			1	2

$R_i = |F_i| - 1$

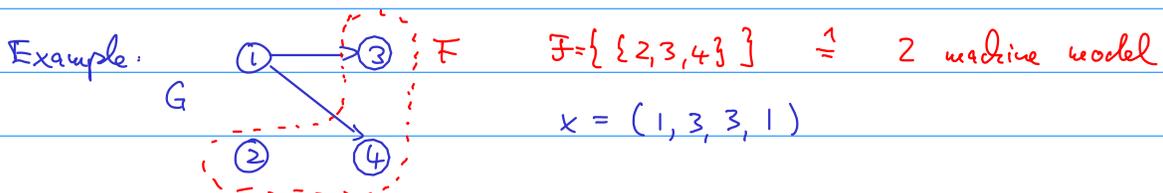
b) m parallel identical machines (m -machine problem)

j	1, 2	...	n		
$\tau_1(j)$	1	1		1	$R_1 = m$

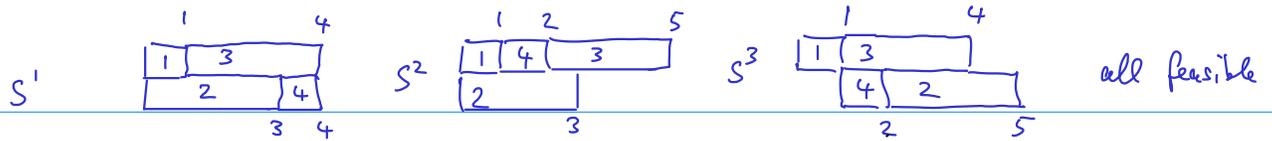
$\Rightarrow \mathcal{F} = \{ \text{all antichains of size } m+1 \}$

Remark: a) $|\mathcal{F}|$ can be exponentially large
 b) often it suffices that \mathcal{F} is given implicitly but \mathcal{F} is a good way to design algorithms

Schedule S is feasible for G, x, \mathcal{F} if S respects G, \mathcal{F}



possible schedules (with most left-shifted jobs)



all feasible

Differentiate between feasible schedules by a regular measure of performance ("cost" function)

$\kappa: \mathbb{R}_+^n \rightarrow \mathbb{R}^1$ non-decreasing in every component

$\kappa(C_1, \dots, C_n) \hat{=} \text{cost of performing the project according to schedule } S \quad [C = S + x]$

$$=: \kappa(S, x)$$

Expl: (i) $\kappa(C_1, \dots, C_n) = \max \{C_1, \dots, C_n\} =: C_{\max}$

makespan, project duration

S^1 is optimal for C_{\max} (the only one)

(ii) $\kappa(C_1, \dots, C_n) = \sum_j C_j$ sum of completion times

models average completion times

S^2 is the only optimal schedule for $\sum C_j$

(iii) $\kappa(C_1, \dots, C_n) = \sum_j w_j C_j$ weighted sum of completion times

weights ≥ 0 , model importance of jobs

with the right weights, S^3 is the only optimal one for $\sum w_j C_j$

(iv) $\sum_j w_j T_j$ $T_j = \text{tardiness of job } j$

$$= \max \{0, C_j - d_j\}$$

\downarrow due date of job j



optimality of a schedule depends on cost functions

may restrict to "left-shifted" schedules since κ is non-decreasing

special case: no resource constraints

Is there a "best" schedule for G, x ?

Yes: Early start schedule

$$ES_G[x](j) : \begin{cases} 0 & j \text{ is minimal in } G \\ \max_{(i,j) \in E} \{ ES_G[x](i) + x_i \} & \text{otherwise} \end{cases}$$

2.2 Lemma

a) $ES_G[x]$ is a schedule that respects G

b) $ES_G[x] \leq S$ for every schedule S that respects G
vector \uparrow vector
componentwise

c) $ES_G[x](j) = \text{length of a longest chain in } G | \text{Pred}(j)$

set C of jobs with $u \leq_G v$ for all $u, v \in C$

$$= \max \left\{ \sum_{i \in C} x_i \mid C \text{ is a maximal chain in } G | \text{Pred}(j) \right.$$

under \leq

d) $ES_G[\cdot] : \mathbb{R}_>^n \rightarrow \mathbb{R}_{\geq}^n$ is convex, monotone, continuous

Proof: c) Induction over $|\text{Pred}(j)|$

base case: $|\text{Pred}(j)| = 0$

$$\max \{ \dots \} = \max \emptyset = 0 \quad \left. \begin{array}{l} \text{Definition of ES} \\ 0 \end{array} \right\} \checkmark$$

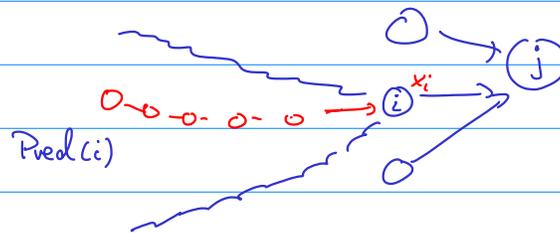
inductive step: $|\text{Pred}(j)| > 0$

$$ES(j) = \max_{(i,j) \in E} \{ ES(i) + x_i \}$$

↑
 $|Pred(i)| < |Pred(j)| \Rightarrow$ can use inductive hypothesis on i

$$= \max_{(i,j) \in E} \left\{ \max_{C \in \text{Pred}(i)} \left\{ \sum_{k \in C} x_k \right\} + x_i \right\}$$

every chain ending in j is a chain in $Pred(i)$
 + arc (i,j) for some $i \in InPred(j)$



$$= \max_{C \text{ is a maximal chain in } Pred(j)} \left\{ \sum_{k \in C} x_k \right\}$$

b) follows from c) since length of a largest chain in $Pred(j)$ is a lower bound on S_j for any schedule S that respects G

a) follows from recursive definition of ES

$$d) \quad ES_G[x](j) = \max_{C \text{ in } Pred(j)} \underbrace{\sum_{i \in C} x_i}_{\text{linear function of } x}$$

max of linear functions

\Rightarrow all properties hold □

Consequences

a) Given κ , then minimum cost of planning according to $ES_G[x]$ is
$$\kappa(ES_G[x], x) =: \kappa^G(x)$$

cost of planning according to G

b) Can consider $\kappa^G(\cdot) : \mathbb{R}^n \rightarrow \mathbb{R}^1$ as a cost or performance function

c) special case $\kappa = C_{\max} = 0$

$C_{\max}^G(x) =$ length of a longest chain in G

CPM-method

(longest chain = critical path)

Resume

If the world is simple (no resource constraints)

then the early bird rule is optimal

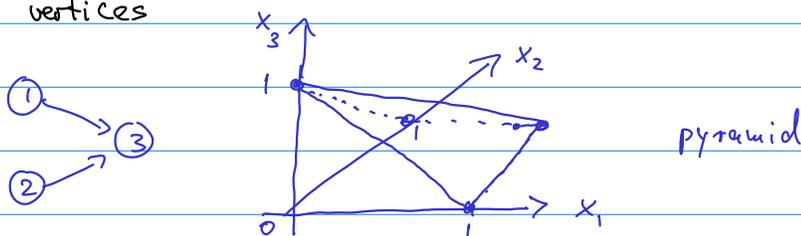
Homework

2.1 S minimizes $L_{\max} \Leftrightarrow S$ minimizes T_{\max}
$$\begin{array}{ccc} \uparrow & & \uparrow \\ \text{Lateness } L_j = C_j - d_j & & T_j = \max\{0, C_j - d_j\} \end{array}$$

2.2 $\sum w_j C_j$ and $\sum w_j L_j$ are equivalent (same w_j)
! i.e. have the same optimal schedules

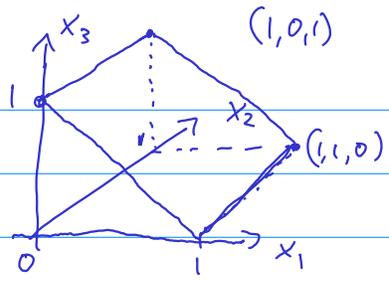
2.3 * Describe the makespan polytope of a partial order G
$$\{x \in \mathbb{R}_+^n \mid C_{\max}^G(x) \leq t\} \quad t \text{ fixed } (t=1)$$

by its vertices



① → ③

②



prism