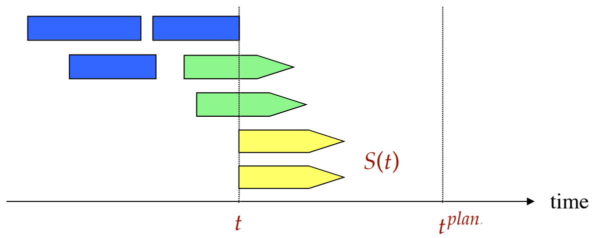
# §5 Scheduling policies

## Planning with policies – the dynamic view

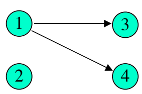



Decision at time $t$ (*non-anticipative*)

- start set $S(t)$ (possibly empty)
- fix tentative next decision time $t^{plan}$ (deliberate idleness)
- next decision time $= \min \{ t^{plan}$, next completion time $\}$

Optimal policies may require tentative decision times, see example from the introduction

### Policies: an example



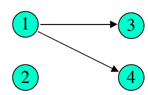

$m = 2$ machines

$X_j \sim exp(\lambda)$, independent

- common due date $d$
- penalties for lateness: $v$ for job 2, $w$ for jobs 3,4, $v << w$
- no interruption of jobs

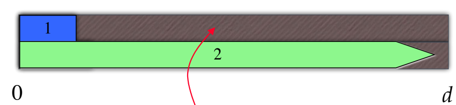
mimimize $\mathbb{E}[\sum \text{penalties}]$

### Example continued



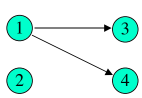

Start jobs 1 and 2 at $t = 0$

Danger: job 2 blocks machine



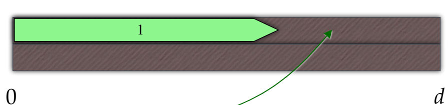

expensive jobs 3 and 4 only sequentially

### Example continued




Start only job 1 and wait for its completion

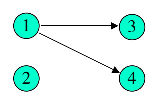
Danger: deadline is approaching




can do expensive jobs 3 and 4 in parallel

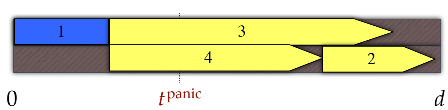
short span to deadline

### Example continued

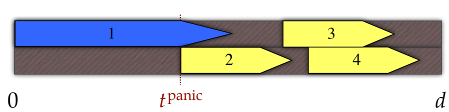



start 1 at 0, fix tentative decision time $t^{\text{panic}}$

if $C_1 \leq t^{\text{panic}}$ start 3 and 4, and then 2

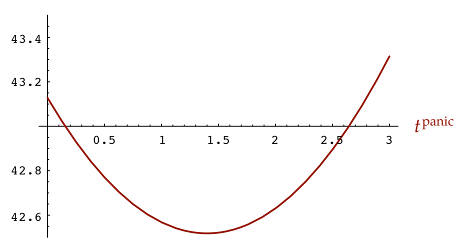



else start 2 at $t^{\text{panic}}$ and then 3 and 4




### Example completed

Expected cost for $\lambda = 1$, $d = 3$, $v = 10$, $w = 100$




$t^{\text{panic}} = 0$ and $\infty$ gives the other two policies

Characterization of policies as function:

$$\Pi : \mathbb{R}^n_> \to \mathbb{R}^n$$

$$x \mapsto \text{schedule } \Pi(x) \text{ respecting } G, \mathcal{F}$$

$x, y$ look the same to $\Pi$ at time $t$

$:\Longleftrightarrow$ $C(t)$ and $B(t)$ are the same for $\Pi(x)$ and $\Pi(y)$ at time $t$

$$x_j = y_j \quad \forall j \in C(t)$$

$$\tilde{x}_j = \tilde{y}_j \quad \forall j \in B(t)$$

defines an equivalence relation $E_t$ on $\mathbb{R}^n_>$

Notation: $\quad x \sim_t y$

Equivalent characterization of non-anticipativity)

$x \sim_t y$ and $\Pi[x](j) = t \Rightarrow \Pi[y](j) = t$ $\Big\}$ (NA)

So $\Pi : \mathbb{R}^n_> \to \mathbb{R}^n_>$ is a policy

$\Leftrightarrow$ $\Pi[x]$ is a feasible schedule $\forall x$

$\Pi$ fulfills (NA)

5.3 Theorem

a) $E_0 = \mathbb{R}^n_> \times \mathbb{R}^n_>$, i.e. $x \sim_{t=0} y \quad \forall x, y$

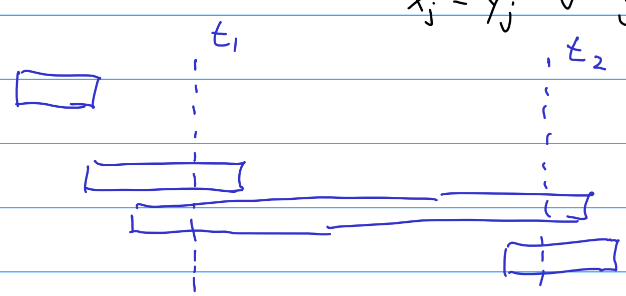
b) $t_1 < t_2 \Rightarrow E_{t_1} \supseteq E_{t_2}$ (or $x \sim_{t_2} y \Rightarrow x \sim_{t_1} y$)

c) $E_\infty = \{ (x,x) \mid x \in \mathbb{R}^n_> \}$ ($x \sim_\infty y \Rightarrow x = y$)

Proof: a) clear since $C(0) = \emptyset$ and $\tilde{x}_j = \tilde{y}_j = 0$

b) let $x \sim_{t_2} y \Rightarrow x_j = y_j \quad \forall j \in C(t_2)$

$$\tilde{x}_j = \tilde{y}_j \quad \forall j \in B(t_2)$$



$j \in C(t_1)$

$j \in B(t_1) \cap C(t_2)$

$j \in B(t_1) \cap B(t_2)$

$\leftarrow$ not important for $t_1$

$j \in C(t_1) \Rightarrow j \in C(t_2) \overset{x \sim_{t_2} y}{\Rightarrow} x_j = y_j$ at $t_1$

$j \in B(t_1) \Rightarrow j \in B(t_2)$ or $j \in C(t_2)$

(i) (ii)

(i) $j \in B(t_2) \Rightarrow \tilde{x}_j = \tilde{y}_j$ at $t_2$

$\Rightarrow \tilde{x}_j = \tilde{y}_j$ at $t_1$ (difference is $t_2 - t_1$)

(ii) $j \in C(t_2) \Rightarrow$ use lemma 5.1 at $t_2$

$\Rightarrow S_j$ and $C_j$ are the same for $x, y$

$\Rightarrow \tilde{x}_j = \tilde{y}_j$ at $t_1$

c) let $x \neq y$ say $x_j \neq y_j \overset{t = \infty}{\Rightarrow} \Pi$ sees a difference in the history $\square$

<u>Properties of planning rules</u>

$\Pi_1$ <u>dominates</u> $\Pi_2$ $:\Leftrightarrow$ $\Pi_1(x) \leq \Pi_2(x)$ (Notation $\Pi_1 \leq \Pi_2$)

$\uparrow$

componentwise

$\Pi$ is <u>minimal</u> in a class $\mathcal{P}$ of planning rules if

(i) $\Pi \in \mathcal{P}$

(ii) $\Pi' \in \mathcal{P}$, $\Pi' \leq \Pi \Rightarrow \Pi' = \Pi$

$\Pi$ is elementary $:\Leftrightarrow$ $\Pi$ starts jobs only at completion of other jobs (i.e. $t^{plan} = \infty$)

# Stability of policies

Data deficiencies, use of approximate methods (simulation) require stability condition:

uniform convergence $\longrightarrow$

$Q'$ approximates $Q$
$\kappa'$ approximates $\kappa$

$\leftarrow$ weak convergence of measures

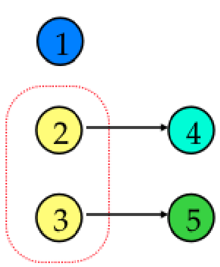
$Q_n \to Q :\iff \int f \, dQ_n \to \int f \, dQ \quad \forall f$ continuous bounded

$\Downarrow$

$OPT(Q',\kappa')$ approximates $OPT(Q,\kappa)$

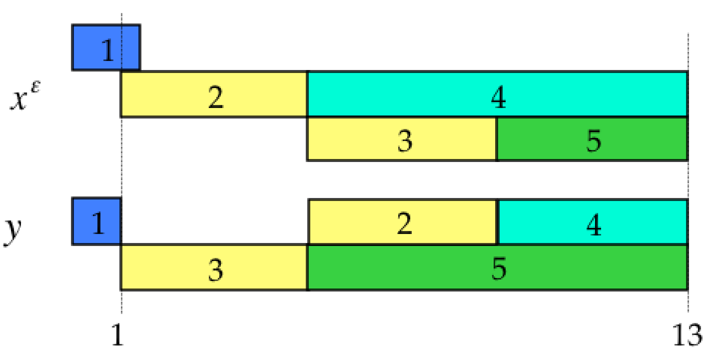
No stability for optimal policies in general!

# Excessive use of information yields instability



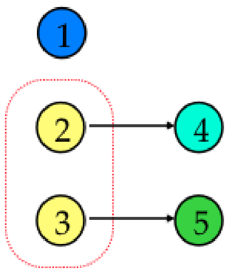
$$\min \mathbb{E}(C_{\max})$$

$$Q^{\varepsilon} : \begin{cases} x^{\varepsilon} = (1+\varepsilon, 4, 4, 8, 4) & \text{with prob. } ½ \\ y \;\; = (1, \quad\; 4, 4, 4, 8) & \text{with prob. } ½ \end{cases}$$




$\mathbb{E}_{Q^{\varepsilon}}(C_{\max}) = 13$

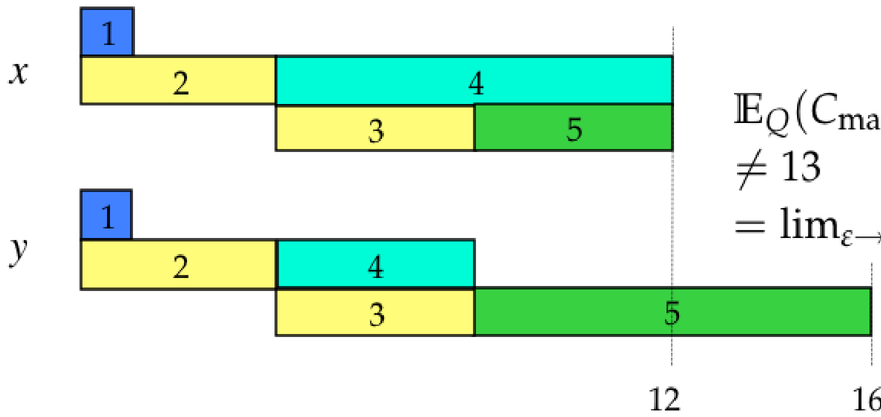
for $\varepsilon \to 0$



$\varepsilon \to 0 \;\Rightarrow\; Q^{\varepsilon} \to Q$ with

$$Q : \begin{cases} x = (1,4,4,8,4) \text{ with probability } \frac{1}{2} \\ y = (1,4,4,4,8) \text{ with probability } \frac{1}{2} \end{cases}$$

No info when 1 completes. So start 2 at $t = 0$




$\mathbb{E}_Q(C_{\max})$
$\neq 13$
$= \lim_{\varepsilon \to 0} \mathbb{E}_{Q^{\varepsilon}}(C_{\max})$

A class $\mathcal{P}$ of policies is called <u>stable</u>

if
$$\lim_{j \to \infty} \rho^{\mathcal{P}'}(\kappa, Q^j) = \rho^{\mathcal{P}'}(\kappa, Q)$$

for
- every sequence $Q^j \to Q$ (weak convergence)
- every countable subset $\mathcal{P}'$ of $\mathcal{P}$
- every continuous $\kappa$

weak requirement

Can show

5.4 Theorem: Let $\mathcal{P}$ be a stable class of policies. Then

(1) every $\Pi \in \mathcal{P}$ is continuous

(2) If $\mathcal{P}'$ is a finite class of continuous policies the $\mathcal{P} \cup \mathcal{P}'$ is stable

(3) $\rho^{\mathcal{P}}(\kappa^j, Q^j) \longrightarrow \rho^{\mathcal{P}}(\kappa, Q)$ for every sequences

$Q^j \to Q$ weak convergence

$\kappa^j \to \kappa$ uniform convergence

i.e. have stability in the strong sense

Proof: (1) Suppose $\Pi$ is not continuous

$\Rightarrow$ $x \to \Pi[x](j) + x_j$ discontinuous for some job $j$

$\Rightarrow$ $\exists$ sequence $x^k \to x$ with $\lim\limits_{k\to\infty} \Pi[x^k](j) + x_j^k \neq \Pi[x](j) + x_j$

Choose $\kappa(C_1, \dots, C_n) := C_j$

$Q^k$ = one point distribution in $x^k$

$Q$ = .... in $x$

$\mathcal{P}'$ = $\{\Pi\}$

$$\Rightarrow \lim_{k\to\infty} \rho^{\mathcal{P}'}(\kappa, Q^k) = \lim_k E_{Q^k}(\kappa^\Pi)$$

$$= \lim_k \left( \Pi[x^k](j) + x^k(j) \right) \neq \Pi[x](j) + x_j = \rho^{\mathcal{P}'}(\kappa, Q)$$

(weak stability condition not satisfied)

(2) show that stability holds by adding one continuous policy at a time

(3) more difficult, needs results on convergence of probability measures

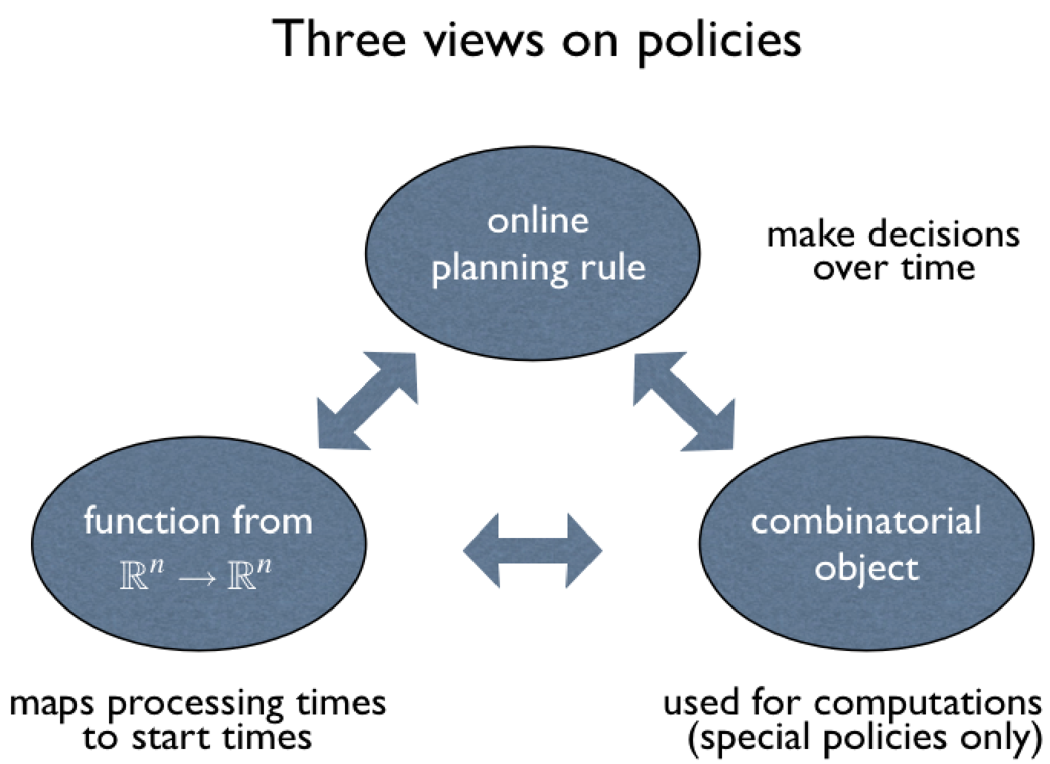
Exercises

5.1 Find an example for $\sum_j w_j C_j$ with independent processing times in which tentative decision times lead to better policies.
Can you do it without precedence constraints?

5.2* Try the same for $C_{max}$

5.3 Show that the class of elementary policies is unstable

Special classes of policies




## §6 Priority policies

combinatorial object $\hat{=}$ priority lists

A planning rule $\Pi$ is a priority planning rule or priority rule for $[G, \mathcal{F}]$ if

(1) $\Pi$ is elementary

(2) At every decision time $t$, there is a priority list $L(t)$

$$L(t): \; j_1 < j_2 < \dots < j_k$$

highest priority

on the set of still unscheduled (= unstarted) jobs

(3) $\Pi$ considers jobs $j$ in $L(t)$ one by one in the order of $L(t)$ and sets $S_j = t$ if possible w.r.t. $G, \mathcal{F}$ and the previously started jobs

$\Pi$ is *static* if every $L(t)$ is a sublist of $L(0)$. Otherwise $\Pi$ is called *dynamic*

For *priority policies*, the list $L(t)$ may only depend on the history up to time $t$

Smith's rule is a static priority rule, but not a priority policy

Smith's rule based on expected processing times

$$\frac{E(X_{j_1})}{w_{j_1}} \leq \frac{E(X_{j_2})}{w_{j_2}} \leq \dots \leq \dots$$

is a static priority polity

*Advantages of priority policies*

(1) $\boxed{\text{Every priority policy is minimal among all policies}}$

Suppose $\Pi' \leq \Pi$, $\Pi$ priority policy, consider fixed $x$

at $t = 0$ $\Pi[x]$ starts as many jobs as possible in the order of $L(0)$

$\Pi' \leq \Pi \Rightarrow \Pi'[x]$ has the same start set

Look at the first completion w.r.t. $x$

$\Pi'$ cannot start new jobs earlier, since resources are all used

$\Rightarrow$ same argument, $\Pi[x]$, $\Pi'[x]$ start the same jobs at the first completion

$\Rightarrow$ (iteration) $\Pi'[x] = \Pi[x] \overset{\text{all } x}{\Rightarrow} \Pi' = \Pi$

(2) $\boxed{\text{easy to implement}}$

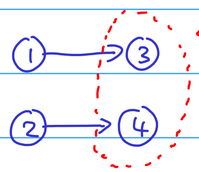
(3) $\boxed{\text{can give approximation guarantees for some problems}}$

e.g. $\dfrac{C_{max}(x)}{OPT\ C_{max}(x)} \leq 2 - \dfrac{1}{m}$ for $m$-machine problems with arbitrary processing times (later)

# Disadvantages of priority policies

(1) priority policies are neither continuous nor monotone and thus may cause instabilities

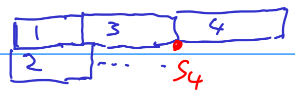
Expl:



← only forbidden set

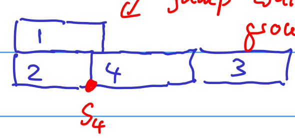
$L = 1 < 2 < 3 < 4$

$x_1 < x_2$



$x_1 > x_2$

jump while $x_1$ grows a little

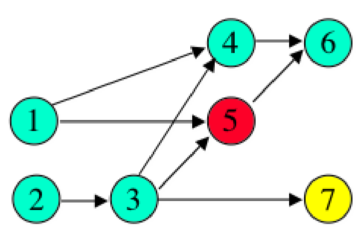


for fixed $x_2, x_3, x_4$ and $x_1 \in [x_2-\varepsilon, x_2+\varepsilon]$

$S_4 = \Pi[\cdot](4)$ is discontinuous and not monoton

(2) Graham anomalies $\sim$ 196?

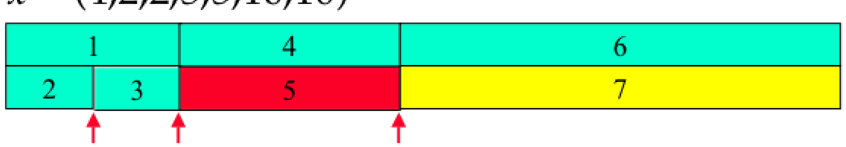
a) **Priority policies have anomalies**



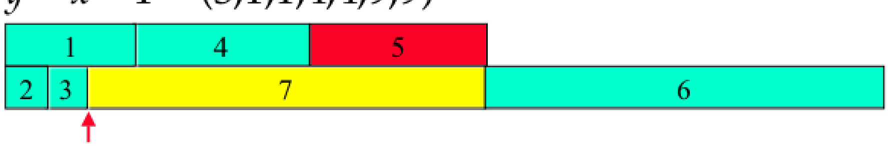
minimize makespan
on 2 identical machines

use priority list $1 < 2 < 3 \ldots$

$x = (4,2,2,5,5,10,10)$

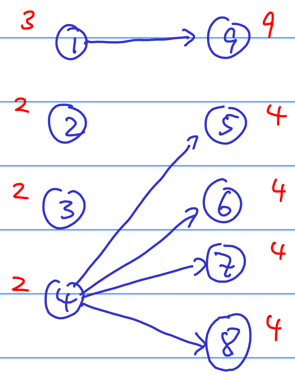


$y = x - 1 = (3,1,1,4,4,9,9)$



a) processing times get smaller

b) delete precedence constraints




$x = (3, 2, 2, 2, 4, 4, 4, 4, 9)$

$m = 2$

$\kappa = C_{max}$

$L = 1 < 2 < \ldots < 9$

$C_{max}(x)$ grows when $4 < 5$ and $4 < 6$ are deleted

c) $C_{max}(x)$ grows when more machines are added

take example b) but with $m = 3$

$C_{max}(x)$ grows when $m = 4$ machines are available

Exercise

6.1 Show that there no Graham anomalies for $m$-machine problems without precedence constraints. Show that, in this case, every priority policy is continuous and monotone (all priority policies static)

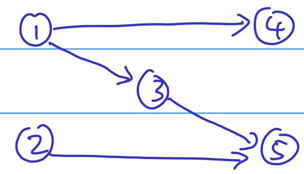
§7 Early start policies

Recall the ES-function induced by a partial order $H$ on $V$

$$ES_H[\cdot] : \mathbb{R}^n_{\geq} \to \mathbb{R}^n$$

$ES_H[x]$ = ES-schedule w.r.t. $H$ and $x$

Expl. H



$$ES_H[x] = (0, 0, x_1, x_1, \max\{x_1 + x_3, x_2\})$$

$\Downarrow$

$ES_H[\cdot]$ is a function with the same domain and range as a policy $\Pi$ for $[G, \mathcal{F}]$

$ES_H[\cdot]$ has nice properties :
- continuous
- monotone ...

Question: Given $(G, \mathcal{F})$, when is $ES_H[\cdot]$ a policy for $(G, \mathcal{F})$ ?

7.1 Theorem:

$ES_H[\cdot]$ is a planning rule for $[G, \mathcal{F}]$ iff

(1) $H$ <u>extends</u> $G$, i.e., $i <_G j \Rightarrow i <_H j$

$H$ is then called an <u>extension</u> of $G$

(2) No antichain of $H$ is forbidden, i.e.

$F \in \mathcal{F} \Rightarrow F$ is not an antichain of $H$

<u>Proof</u>

"$\Rightarrow$" by contradiction

(1) assume $i <_G j$ but $i \not<_H j$. Define $x \in \mathbb{R}^n_>$ by

$$x_k := \begin{cases} \varepsilon & k \neq i \\ 1 & k = i \end{cases} \qquad \varepsilon < \frac{1}{n^2}$$

$$ES_H[x](j) \underset{\substack{\uparrow \\ i \not<_H j}}{\leq} (n-2) \cdot \varepsilon < 1$$

$x_i = 1$

$\Rightarrow$ $j$ does not wait for $i$ in schedule $ES_H[x]$

$\Rightarrow$ $ES_H[x]$ does not respect $G$

$\Rightarrow$ $ES_H[\cdot]$ is not a planning rule for $[G, \mathcal{F}]$

(2) Suppose $F \in \mathcal{F}$ is an antichain of $H$. Define $x \in \mathbb{R}^n_>$ by

$$x_k := \begin{cases} \varepsilon & k \notin F \\ 2 & k \in F \end{cases} \qquad \varepsilon < \frac{1}{n^2}$$

Then, for $j \in F$, $ES_H[x](j) \leq (n-2)\varepsilon < 1$

$j$ has only short predecessors

$\Rightarrow$ at $t = 1$ all jobs in $F$ are busy

$\Rightarrow$ $ES_H$ is a not a planning rule for $[G, \mathcal{F}]$

"$\Leftarrow$" $H$ extends $G$ $\Rightarrow$ $ES_H[\cdot]$ respects $G$

no antichain of $H$ is forbidden $\Rightarrow$ $ES_H[\cdot]$ respects $\mathcal{F}$

only antichains can be processed simultaneously in $H$ $\square$

<u>Definition</u>: Call a partial order $H$ on $V$ <u>feasible</u> for $[G, \mathcal{F}]$ if

(1) it respects $G$, i.e., it extends $G$

(2) it respects $\mathcal{F}$, i.e., no antichain of $H$ is in $\mathcal{F}$

7.2 THEOREM

$ES_H[\cdot]$ is a planning rule for $[G, \mathcal{F}]$ iff $H$ is feasible for $[G, \mathcal{F}]$.

In this case, $ES_H[\cdot]$ is already a policy, i.e. non-anticipative

<u>Proof</u>: the first part follows from Thm. 7.1.

Show non-anticipativity:

So let $x \sim_t y$ and $ES_H[x](j) = t$

$\Downarrow$

$x, y$ have the same history up to time $t$

$\Rightarrow$ all $i \in Pred_H(j)$ have $x_i = y_i$

$\Rightarrow$ ( $ES_j$ = longest path in $Pred_H(j)$ ) $ES_H[x](j) = ES_H[y](j) = t$ $\square$

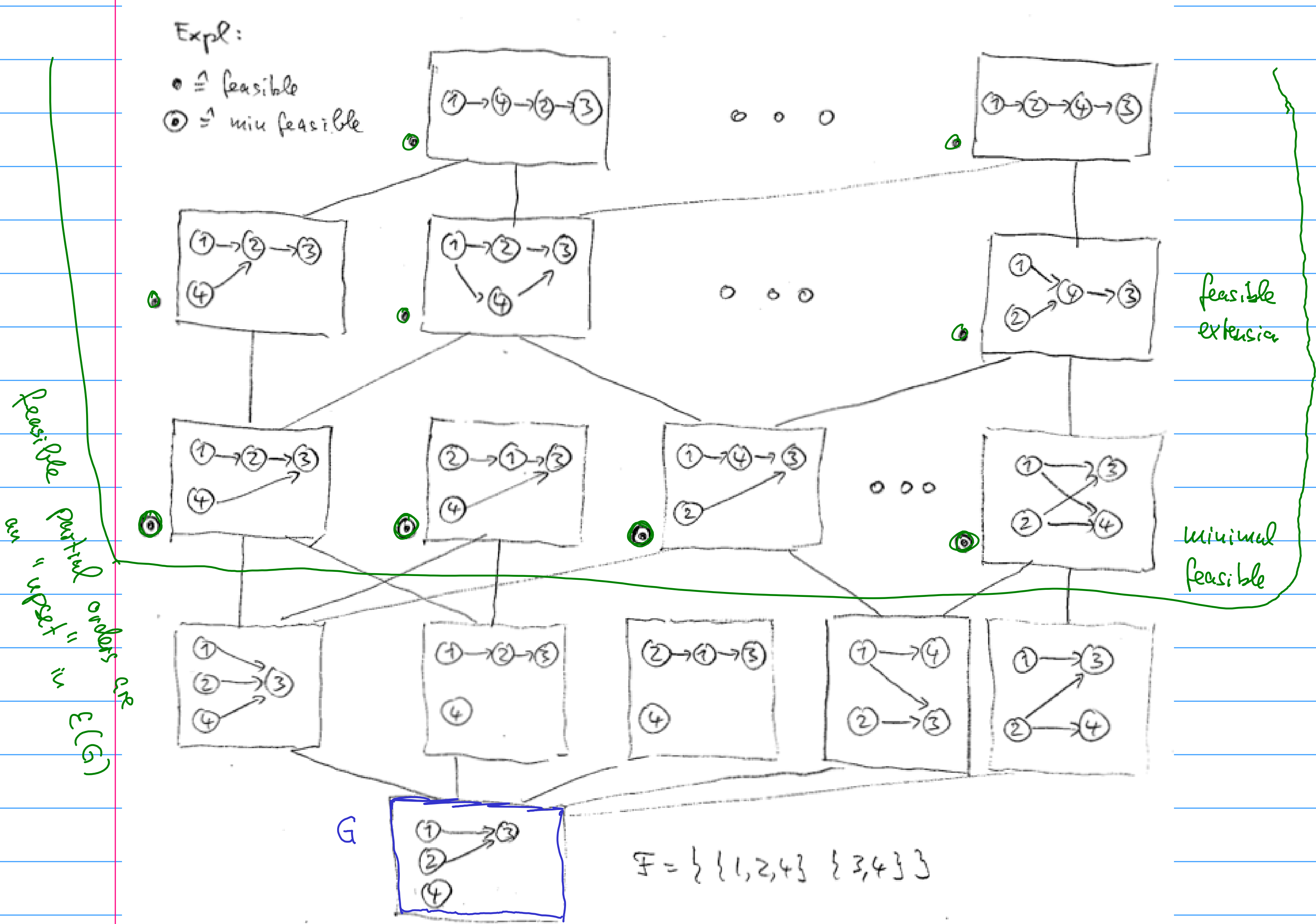
Let $\mathcal{E}(G)$ denote the set of all extensions of $G$ (including $G$)

$\mathcal{E}(G)$ is a partial order under the order relation

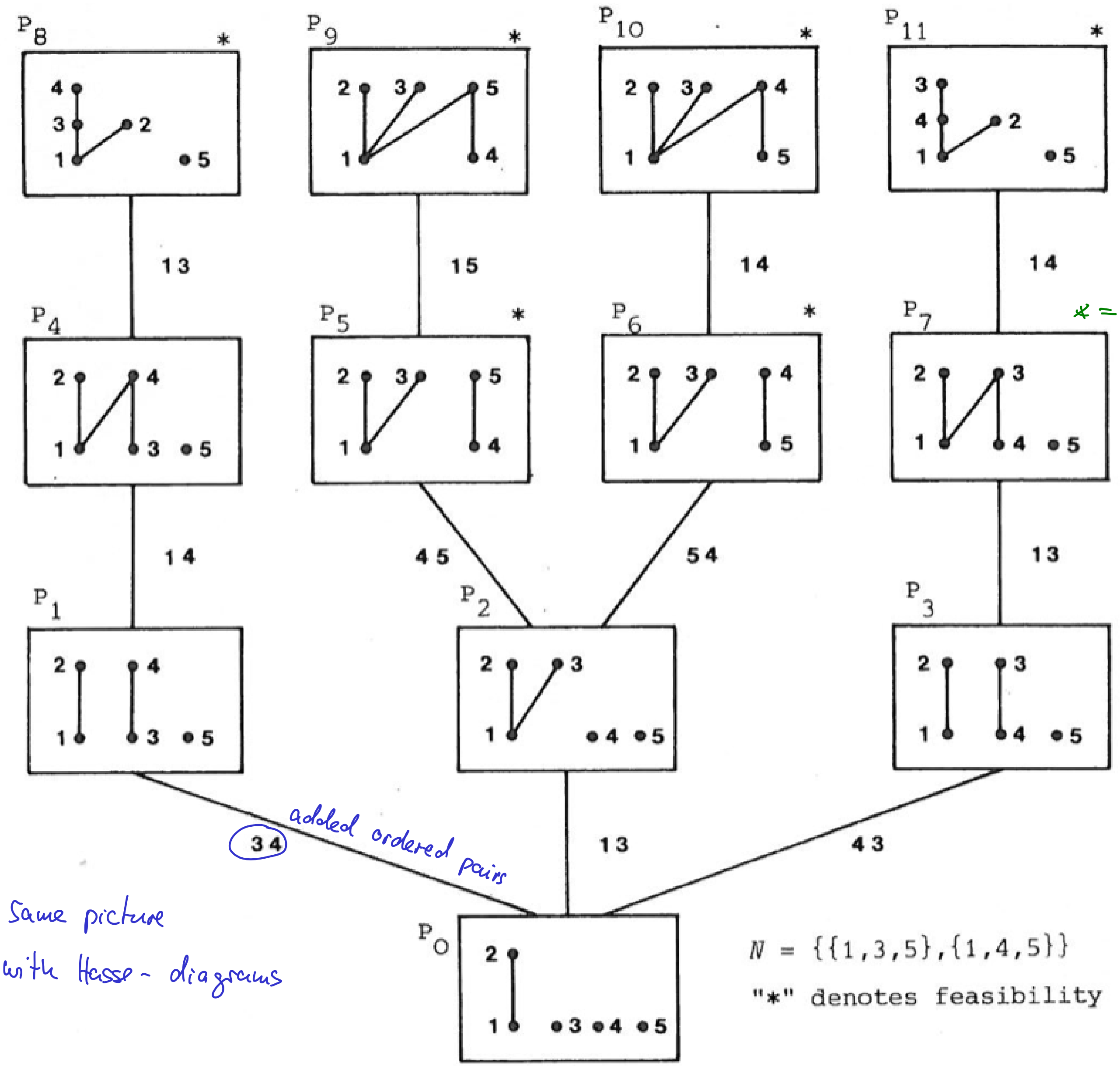
$$H_1 \prec H_2 \quad :\Leftrightarrow \quad H_2 \text{ extends } H_1$$

$$(\text{i.e. } i <_{H_1} j \Rightarrow i <_{H_2} j)$$

We call $\mathcal{E}(G)$ the <u>extension order</u> of $G$



* = feasible




added ordered pairs

Same picture with Hasse-diagrams

$N = \{\{1,3,5\},\{1,4,5\}\}$

"*" denotes feasibility

$\mathcal{E}(G)$ has nice properties (Exercises)

(1) $H_1 = (V, E_1)$, $H_2 = (V, E_2)$

$H_1$ is an immediate predecessor of $H_2$ $\Leftrightarrow$ $|E_2^{trans} \setminus E_1^{trans}| = 1$

partial order constraints (including transitive pairs) differ exactly by 1

$H_1 < H_2 \Leftrightarrow E_1^{trans} \subseteq E_2^{trans}$

(2) All maximal (w.r.t. $\leq$) chains from $H_1$ to $H_2$ have the same length

(3) $H_1$ and $H_2$ have a (unique) largest common predecessor $H = (V, E)$ with $E^{trans} = E_1^{trans} \cap E_2^{trans}$

(4) $H_1, H_2$ have a (unique) smallest common successor $H = (V, E)$ iff $E_1 \cup E_2$ is acyclic. Then $E^{trans} = (E_1 \cup E_2)^{trans}$

Note: (3) and (4) say that $\mathcal{E}(G)$, equipped with an artificial greatest element, is a semi-lattice

7.3 Consequences

(1) The set of feasible orders is "upwardly closed" in $\mathcal{E}(G)$
i.e. $H_1$ feasible, $H_1 \prec H_2 \Rightarrow H_2$ is feasible

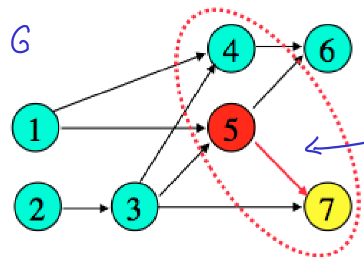
(2) $H_1 \prec H_2 \Rightarrow ES_{H_1} \leq ES_{H_2}$

(3) $ES_H$ is a minimal ES-policy $\iff$ $H$ is minimal feasible in $\mathcal{E}(G)$

(4) The set of ES-policies for $[G, \mathcal{F}]$ may be identified with the set of feasible orders in $\mathcal{E}(G)$

ES - policies avoid Graham anamolies

# An ES-policy for Graham's example

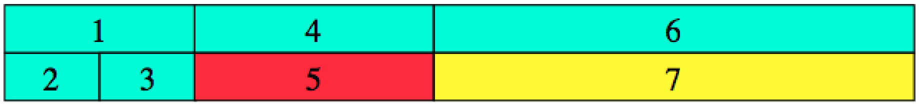



$\mathcal{F}$: 2 identical machines

$\Rightarrow F = \{4,5,7\}$ is only forbidden set

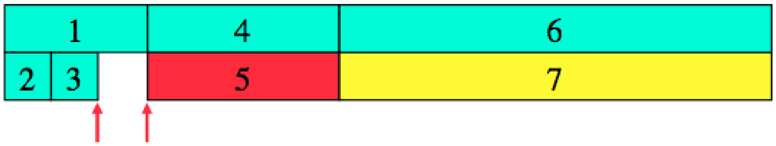
$5 < 7$ defines an extension H of G

$x = (4,2,2,5,5,10,10)$




$y = x - 1 = (3,1,1,4,4,9,9)$

ES-policy w.r.t. to extension H




ES policies lead to optimal schedules in the deterministic case

This is not the case for priority policies!

in the above example with processing times $y$, every priority policy would start job 7 at the completion of 3, leading to a non-optimal schedule

7.4 Theorem.

Consider $[G, \mathcal{F}]$. For every $x$ there is an ES-policy $\Pi = ES_H$ such that $OPT(\kappa, x) = \kappa^H(x)$ $\left(= \kappa^{ES_H}(ES_H[x], x)\right)$

[ Note that $\Pi$ and thus $H$ depend on $x$ and $\kappa$ ]

In particular:

$$OPT(\kappa, x) = \min\{\kappa^H(x) \mid H \text{ feasible}\}$$
$$= \min\{\kappa^H(x) \mid H \text{ minimal feasible}\}$$