§ 5 SCHEDULING POLICIES

Scheduling problems are "highly" NP-hard

$\Rightarrow$ need polynomial (approximation) methods

Study them with varying processing times in mind

- i.e. independent of x $\leftarrow$ may change while G and $\mathcal{F}$ are fixed

- and to a large extent also independent of κ

Def: A <u>planning rule</u> for $[G, \mathcal{F}]$ is a function

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

that assigns to each vector x of processing times a schedule $\Pi[x]$ respecting G, $\mathcal{F}$ and x

Since every planning rule is per def. a function, we can speak of <u>continuous</u> or <u>convex</u> planning rules

For a fixed performance measure κ and a fixed planning rule Π, the function

$$\kappa^\Pi : \mathbb{R}^n_{>} \to \mathbb{R}^1 \quad \text{with} \quad \kappa^\Pi(x) := \kappa\big(\underbrace{\Pi[x] + x}_{\substack{\text{vector of completion times} \\ \text{w.r.t. } \Pi[x] \text{ and } x}}\big)$$

gives the performance cost resulting from planning according to Π.

For a fixed class $\mathcal{P}$ of planning rules, the optimal value for fixed x is given by

$$\rho^{\mathcal{P}}(\kappa, x) := \inf \{ \kappa^{\Pi}(x) \mid \Pi \in \mathcal{P} \}$$

If the processing times are random, we want to find a planning rule from $\mathcal{P}$ that is <u>best on average</u>, i.e.

$$\rho^{\mathcal{P}}(\kappa) := \inf \{ \underbrace{E(\kappa^{\Pi})}_{\uparrow} \mid \Pi \in \mathcal{P} \}$$

expected cost of planning rule Π

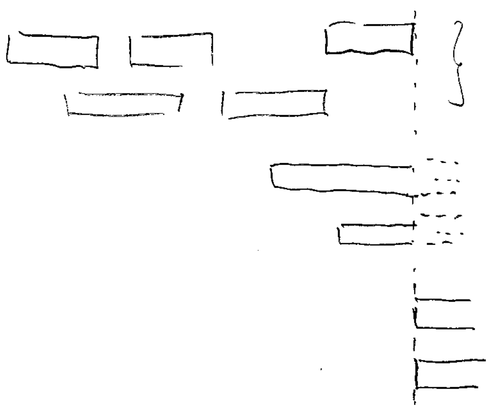
<u>Dynamic representation of policies</u>

different, implicit representation

<u>decision times</u>

$$\left.\begin{array}{l} t = 0 \\ \text{completion of a job} \\ \text{times at which information} \\ \qquad \text{becomes available} \end{array}\right\} \text{make a decision}$$

Situation at a decision time t



} set $C(t)$ of completed jobs

} set $B(t)$ of busy jobs

} $\underbrace{\text{set } S(t) \text{ of jobs started at } t}_{\text{part of the decision taken at } t}$

other part = next tentative decision time t^{planned}

actual next decision time $t^{\text{next}} = \min\{t^{\text{planned}}, \text{completion of a job}\}$

<u>Def:</u> A planning rule is <u>non-anticipative</u> or a <u>policy</u> if t^{planned} and $S(t)$ depend only on the history up to time t

history:
- processing times + starting times of completed jobs
- current processing times + starting times of busy jobs
- time t

[gives info about: conditional distribution obtained from joint distribution of processing times conditioned on the history]

$$\left.\begin{array}{l}\text{history} = \text{state}\\ \text{decision} = \text{action}\end{array}\right\} \text{ in dynamic programming}$$

<u>Expl.</u> Smith's rule : not non-anticipative

matching algorithm: non-anticipative

$\hat{=}$ ES of an extension of G

Expl: non-anticipative vs anticipative leads to worse optimal value g

G

① ② ③

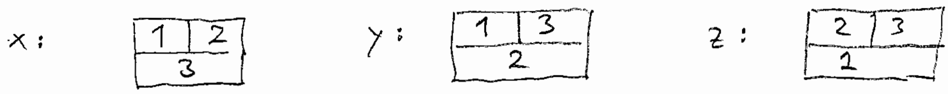
joint distribution:

realization	probability
$x = (1, 1, 2)$	$1/3$
$y = (1, 2, 1)$	$1/3$
$z = (2, 1, 1)$	$1/3$

$K = C_{max}$

2-machine problem

Best anticipative planning rule : do the best for every realization

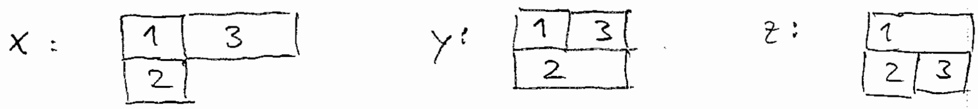


$\Rightarrow C_{max} = 2$ for every realization $\Rightarrow E(C_{max}^{\pi}) = 2$

Best non-anticipative planning rule :

must make a decision at $t = 0$ without knowing the future, i.e. the realization

symmetry: w.l.o.g. let $S(0) = \{1, 2\}$



3 2 2

$\Rightarrow E(C_{max}^{\pi}) = 2\frac{1}{3}$

$\Downarrow$

There is a loss of $\frac{1}{3}$ due to non-anticipativity

= due to uncertainty about processing times

typical for decision making under uncertainty

Observation: Best anticipative planning rule Π^A is given by

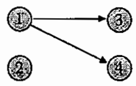
$\Pi^A[x] :=$ optimal feasible schedule for x

$\Rightarrow$ Loss of a policy Π due to uncertainty is

$$E(\kappa^{\Pi}) - E(\kappa^{\Pi^A})$$

Example: Tentative decision times are necessary

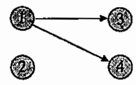
Policies – An Example



- $m = 2$ machines $\Rightarrow F = \{2,3,4\}$ forbidden
- $X_j \sim \exp(a)$, independent

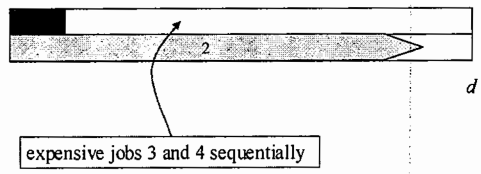
- common due date d
- penalties for lateness: v for job 2, w for jobs 3,4, $v \ll w$

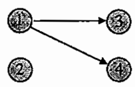
Minimize $E(\Sigma$ penalties $)$



Start jobs 1 and 2 at $t = 0$

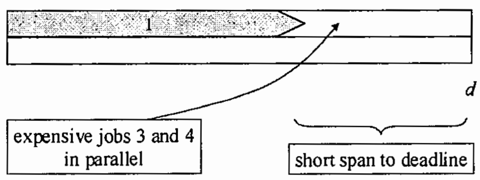
Danger: job 2 blocks machine





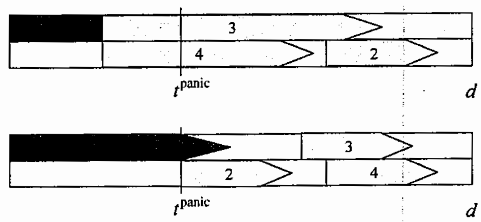
Start only job 1 and wait for its completion

Danger: deadline is approaching



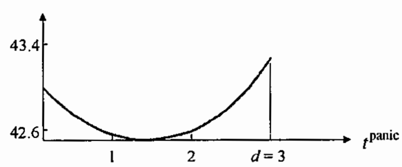
Start 1 at time 0. Fix tentative decision time t^{panic}

If $C_1 \le t^{\text{panic}}$ start 3 and 4 at C_1, else start 2 at t^{panic}



Jobs may start when no other jobs end

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



Expl shows that tentative decision times are needed

May require that $S(t) \neq \emptyset$ at tentative decision times

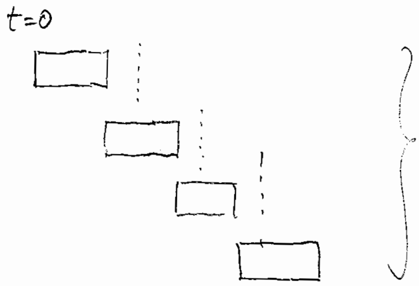
(otherwise consider policy without that decision time, which defines the same policy as a function)

$\Downarrow$

Consider at most $2n-1$ decision times in a realization x

$t=0$, every completion except last

+ at most $n-1$ tentative decision times (since a job must be started)



achieves $2n-1$

5.1 LEMMA: Let Π be a policy (i.e. non-anticipative). Then the history at decision time t is completely determined by

- the processing time x_j of all completed jobs
- the current processing time $\bar{x}_j$ of all busy jobs
- the time t

[i.e. do not need start times and completion times to determine the history]

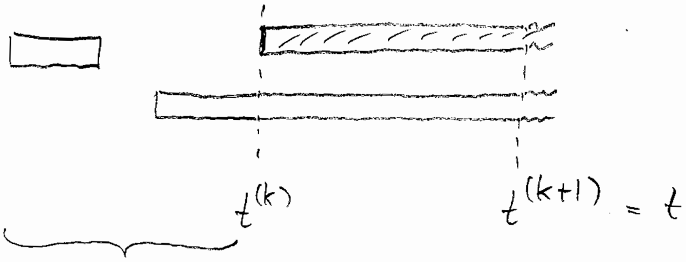
Proof: induction along the decision times $t^{(1)}=0$, $t^{(2)}$, $t^{(3)}$ ...

$t^{(1)}=0 \overset{\Pi \text{ policy}}{\Rightarrow}$ Π determines $S(t^{(1)})$ uniquely

$\Rightarrow$ $S_j = 0$ for all $j \in S(t^{(1)})$ [start times recovered]

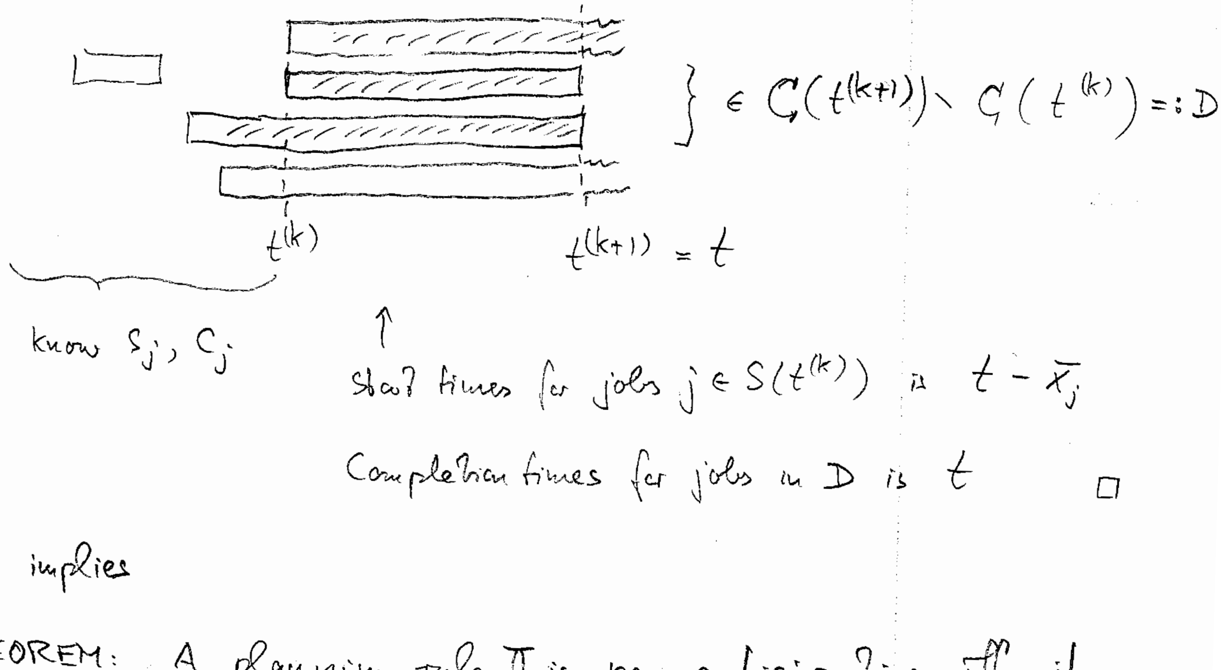
Now suppose that Lemma holds for history up to decision time $t^{(k)}$.
Show this for $t^{(k+1)}$

<u>Case 1:</u> $t^{(k+1)}$ is no completion time [$\Rightarrow$ no job completes between $t^{(k)}, t^{(k+1)}$]



here we know by the inductive assumption
all start times S_j and completion times C_j $\Rightarrow$ know all S_j, C_j up to $t^{(k+1)}$
obtain S_j of "new" jobs as $S_j = t - \bar{x}_j$
no new completion

<u>Case 2</u> $t^{(k+1)}$ is a completion time [$\Rightarrow$ first completion after $t^{(k)}$]



start times for jobs $j \in S(t^{(k)})$ is $t - \bar{x}_j$

Completion times for jobs in D is t $\square$

Lemma 5.1 implies

5.2 THEOREM: A planning rule Π is non-anticipative iff it fulfills the following condition:

$$(NA)\begin{cases} \text{if } x, y \in \mathbb{R}^n \text{ look the same to } \Pi \text{ at time } t \text{ [w.r.t. Lemma 5.1]} \\ \text{and } \Pi[x](j) = t \quad \text{then} \quad \Pi[y](j) = t \end{cases}$$

Theorem 5.2 expresses non-anticipativity as a condition (NA) suitable for the function interpretation of policies

x, y <u>look the same to Π at time t</u>

$\iff$ $x_j = y_j$ for all $j \in C(t)$
$\bar{x}_j = \bar{y}_j$ for all $j \in B(t)$

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

Notation: $(x, y) \in E_t$
$x \sim_t y$

5.3 THEOREM: a) $E_0 = \mathbb{R}^n_> \times \mathbb{R}^n_>$ i.e. all x, y look the same to Π

b) $t_1 < t_2 \Rightarrow E_{t_1} \supseteq E_{t_2}$

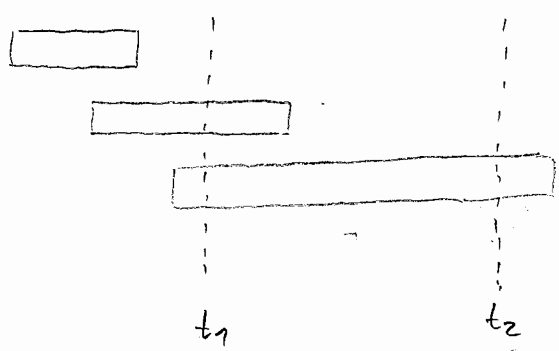
i.e. equivalence classes get smaller as time passes

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

i.e. in the end all vectors look different to Π

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

b) let $x \sim_{t_2} y \Rightarrow x_j = y_j$ for all $j \in C(t_2)$
$\bar{x}_j = \bar{y}_j$ ---- $\in B(t_2)$



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

$j \in C(t_1) \Rightarrow j \in C(t_2) \Rightarrow x_j = y_j$

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

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

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

$j \in C(t_2) \Rightarrow$ (Lemma 5.1 at time t_2)

S_j and C_j are the same under x and y

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

c) Let $x \neq y$ say $x_j \neq y_j$

$\Rightarrow$ at $t = \infty$ Π sees a difference $\square$

PROPERTIES OF PLANNING RULES

Π_1 <u>dominates</u> Π_2 $:\Longleftrightarrow$ $\Pi_1[x] \underset{\substack{\uparrow \\ \text{componentwise}}}{\leq} \Pi_2[x]$ for all x

Notation $\Pi_1 \leq \Pi_2$

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

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

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

Π is <u>elementary</u> $:\Longleftrightarrow$ Π starts jobs only at completion of other jobs (i.e. $t^{\text{planned}} = \infty$)

Stability of policies

Stability of policies

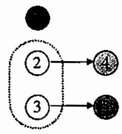
Data deficiencies, use of approximate methods (simulation) require stability condition:

$\tilde{Q}$ approximates Q
$\tilde{\kappa}$ approximates κ $\Longrightarrow$ OPT$(\tilde{Q},\tilde{\kappa})$ approximates OPT(Q,κ)

$Q^j \to Q$ weak convergence of probability measures

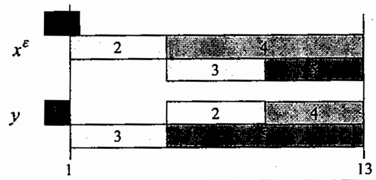
$\kappa^j \to \kappa$ uniform convergence

Excessive use of information yields instability

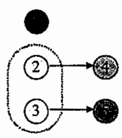


$\min E(C_{\max})$

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



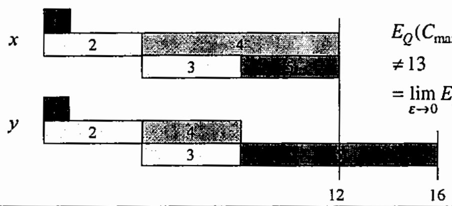
$E_{Q^\varepsilon}(C_{\max}) = 13$
for $\varepsilon \to 0$



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

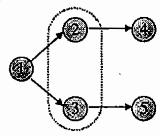
$$Q : \begin{cases} x = (1,4,4,8,4) \text{ with probability } \frac{1}{2} \\ y = (1,4,4,4,8) \text{ with probability } \frac{1}{2} \end{cases}$$

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



$E_Q(C_{\max}) = 14$
$\neq 13$
$= \lim_{\varepsilon \to 0} E_{Q^\varepsilon}(C_{\max})$

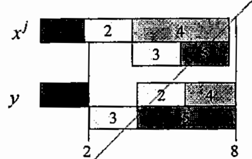
Excessive information yields instability



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

$$Q^j : \begin{cases} x^j = (2-\frac{1}{j},2,2,4,2) \text{ with probability } \frac{1}{2} \\ y \;\; = (2, \quad 2,2,2,4) \text{ with probability } \frac{1}{2} \end{cases}$$

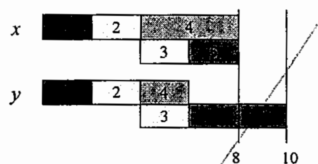
Exploit info when 1 completes



$\Rightarrow E_{Q^j}(C_{\max}) \xrightarrow[j\to\infty]{} 8$

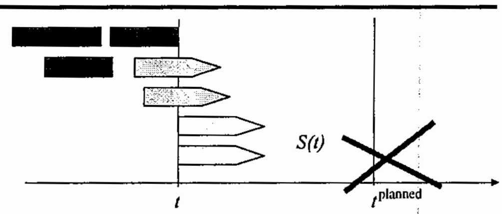
$$Q^j \xrightarrow[j\to\infty]{} Q : \begin{cases} x = (2,2,2,4,2) \text{ with probability } \frac{1}{2} \\ y = (2,2,2,2,4) \text{ with probability } \frac{1}{2} \end{cases}$$

No info when 1 completes. So start 2 after 1



$\Rightarrow E_Q(C_{\max}) = 9$

Robust information and decisions



Robust information at time t

- which jobs have completed by t
- which jobs are running at t

Start jobs only at completions of other jobs

↳ hope for stability

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

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

for - every sequence $Q^j \to Q$ of probability distributions
- every countable subset $\mathcal{P}'$ of $\mathcal{P}$
- every continuous κ

weak stability 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 then $\mathcal{P} \cup \mathcal{P}'$ is stable

In particular, every finite class of continuous policies is stable

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

$Q^j \to Q$ weak convergence

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

i.e. have stability in the strong sense

Proof: (1) Suppose Π is not continuous

$\Rightarrow \quad x \to \underbrace{\Pi[x](j) + x_j}_{\text{completion } C_j \text{ of job } j \text{ under } \Pi}$ discontinuous for some j

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

Choose $\kappa(C_1, \dots, C_n) := C_j$, $\quad Q^k$ = one point distribution in x^k

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad Q$ = ... in x

$\Rightarrow$ For $\mathcal{P}' = \{\Pi\}$ we have

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

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

(2) simple, add one policy at a time

(3) more difficult, need results on weak convergence of probability measures, without proof $\square$.

Exercises:

5.1 Find an example for $\sum w_j C_j$ 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