

§6 Priority policies

combinatorial object $\hat{=}$ priority list

A planning rule Π is a priority planning rule or priority rule for $[G, F]$
if

(1) Π is elementary

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

$$L(t) : j_1 < j_2 < \dots < j_k \quad (\text{unstarted jobs at } t)$$



highest priority

on the set of still unscheduled jobs

(3) Π considers jobs j in $L(t)$ one by one in the order of $L(t)$
and sets $s_j = t$ if possible with $[G, F]$ and the
previously started jobs at t

Π is static, if every $L(t)$ is a sublist of $L(0)$

Otherwise Π 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

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

is a static priority policy

Advantages of priority policies

(1) Every priority policy is minimal among all policies

Suppose $\pi' \leq \pi$, π priority policy, consider fixed x

at $t=0$, $\pi(x)$ starts as many jobs as possible in the order of $L(0)$

$\pi' \leq \pi \Rightarrow \pi'[x]$ starts the same jobs as π

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

π' cannot start new jobs earlier, since resources are all used

\Rightarrow same argument, $\pi(x)$, $\pi'[x]$ start the same set at the first completion

\Rightarrow (iteration) $\pi'[x] = \pi[x] \stackrel{\text{all } x}{\Rightarrow} \pi' = \pi$

(2) easy to implement

(3) can give approximation guarantees for some problems

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

Disadvantages of priority policies

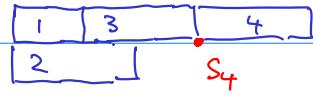
(1) priority policies are neither continuous nor monotone and thus may cause instabilities

Expl. $① \rightarrow ③$: \leftarrow only forbidden set $L: 1 < 2 < 3 < 4$

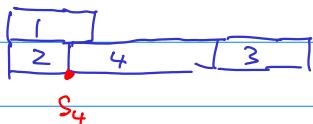
$② \rightarrow ④$:

$$x_1 < x_2$$

$$x_2 < x_1 + x_3$$



$$x_1 > x_2$$



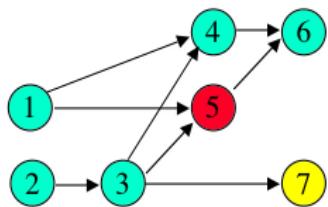
for fixed x_2, x_3, x_4

and $x_1 \in [x_2 - \varepsilon, x_2 + \varepsilon]$

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

(2) Graham anomalies $\sim 1960 +$

Priority policies have anomalies



minimize makespan
on 2 identical machines

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

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

1	4	6
2	3	5

↑ ↑ ↑

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

1	4	5	6
2	3	7	

↑

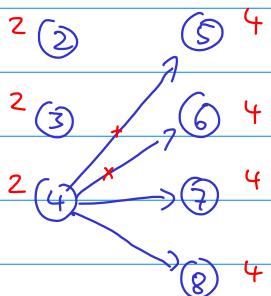
b) delete precedence constraints

$$\stackrel{3}{(1)} \rightarrow \stackrel{9}{(9)} \quad 9 \leftarrow x_j$$

$$m = 2$$

$$k = C_{\max}$$

$$L: 1 < 2 < 3 \dots$$



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

a) processing times
get smaller,
but C_{\max} grows

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 are no Graham anomalies for m -machine problems without precedence constraints.

Show that, in this case, every priority policy is continuous and monotone

↑
static