

§9 Preselective policies

Generalize ES-planning by relaxing the "rule" for solving the conflict on a forbidden set

ES-policies

for every $F \in \bar{F}$ choose $i_F, j_F \in F$ and add $i_F < j_F$ to G

i.e., choose a waiting job j_F and a job i_F for which j_F must wait

Preselective planning rules (will show that they are policies)

for every $F \in \bar{F}$, choose a waiting job $j_F \in F$ that must wait

for any job $\in F$, i.e. j_F can start after the first job in $F \setminus \{j_F\}$ completes

The sequence $s = (j_{F_1}, j_{F_2}, \dots, j_{F_k})$ of waiting jobs for $\bar{F} = \{\bar{F}_1, \dots, \bar{F}_k\}$ is called a selection for \bar{F}

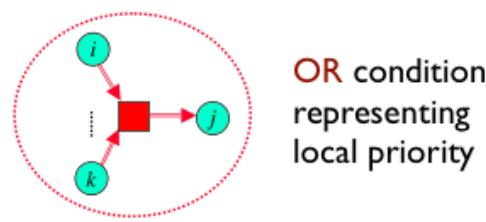
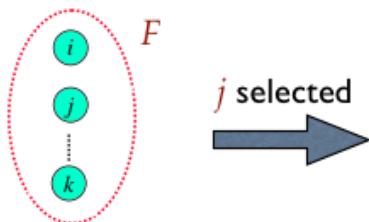
Idea: do early start scheduling w.r.t. to G and the waiting conditions resulting from the selection

Preselective policies

F is forbidden set $\Leftrightarrow F$ cannot be scheduled simultaneously but every proper subset can

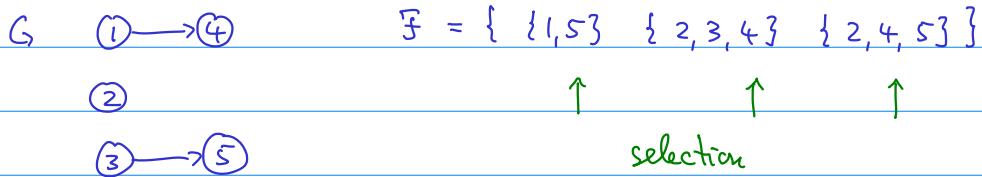
Solve conflict on every F by selecting a waiting job $j_F \in F$

j_F must wait until some job from F is completed

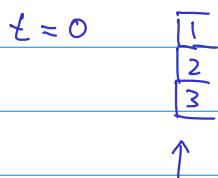


do early start scheduling w.r.t. original precedence constraints and OR conditions resulting from forbidden sets

9.1 Example



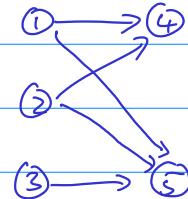
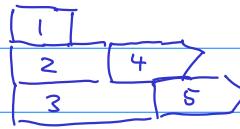
what are the resulting schedules (depending on x)



consider possible completions

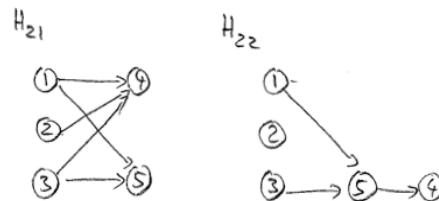
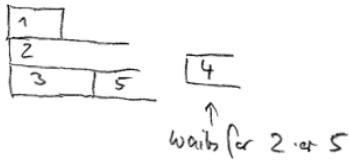
can be started at $t=0$

(1) $x_1 < x_2 < x_3$

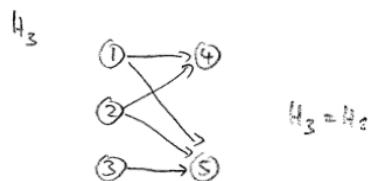
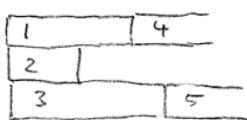


is ES of this feasible order

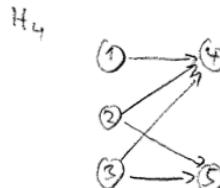
(2) $x_1 < x_3 < x_2$



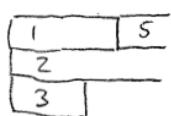
(3) $x_2 < x_1 < x_3$



(4) $x_2 < x_3 < x_1$

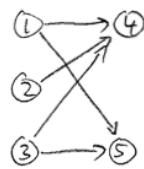


$$(5) \quad x_3 < x_1 < x_2$$

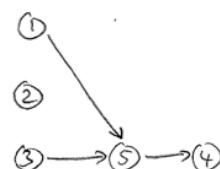


$\boxed{4}$
waits for
2 or 5

$$H_{51} = H_{21}$$



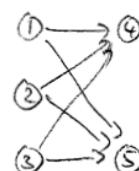
$$H_{52} = H_{22}$$



$$(6) \quad x_3 < x_2 < x_1$$



$$H_6$$



$$H_6 \succ H_1$$

Cases with equality are subsumed by (1)-(6)

Example shows:

- depending on x we obtain a different feasible order H
with $\pi[x] = ES_H[x]$ (*holds for every elementary policy*)
- $\pi = \min \{ ES_H \mid H \text{ induced by } \pi[x] \text{ for some } x \}$
(*does not hold for every elementary policy*)

Questions:

- When is a selection contradictory? Can this be easily checked?
- Does a "feasible" selection define a policy?
I.e. is a preselective policy non-anticipative?
- What is the relationship with ES-policies?
Is $\pi = \min \{ ES_H \mid H \dots \}$ for some set of feasible ES-policies?
(as in the example)
- How to calculate the start times of a preselective policy?
- Do preselective policies still have nice properties?

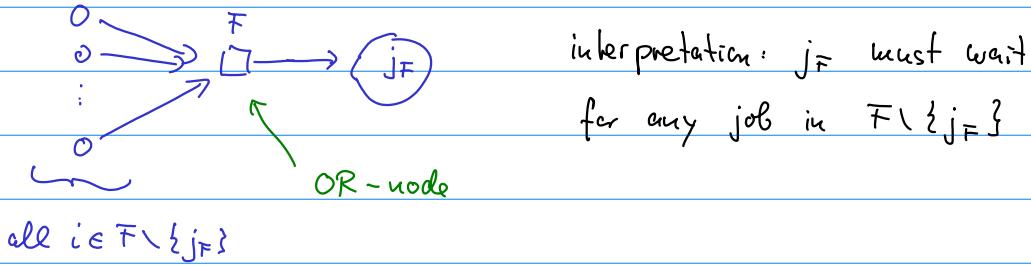
Preselective planning rules and AND/OR networks



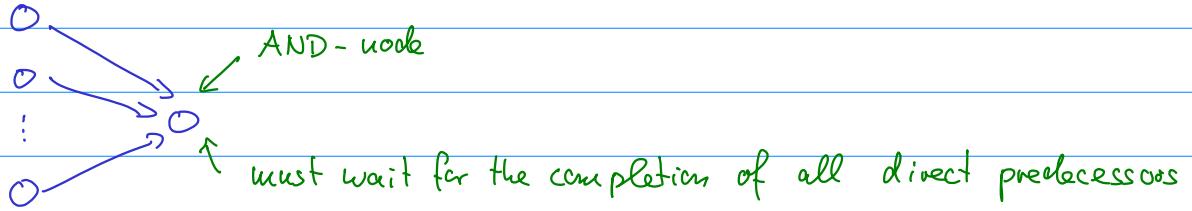
combinatorial interpretation of
preselective planning rules

Consider a selected job $j \in F \in \mathcal{F}$

\Rightarrow induces an OR-precedence constraint



ordinary precedence constraints



AND/OR network = network with AND/OR nodes

Observation: every selection defines an AND/OR-network

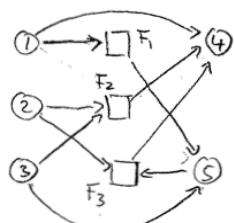
$$G \quad \begin{matrix} ① \rightarrow ④ \\ ② \\ ③ \rightarrow ⑤ \end{matrix} \quad \mathcal{F}: \begin{matrix} \{1,5\} & \{2,3,4\} & \{2,4,5\} \end{matrix}$$

$\uparrow \quad \uparrow \quad \uparrow$

defines selection $s = (5, 4, 4)$

9.1 EXAMPLE (continued)

$s = (5, 4, 4)$ defines



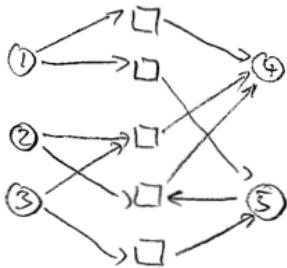
$$\mathcal{F}_1 = \{1,5\}$$

$$\mathcal{F}_2 = \{2,3,4\}$$

$$\mathcal{F}_3 = \{2,4,5\}$$

Observation: every precedence constraint $i < j$ can be interpreted as OR-constraint $(i) \rightarrow \square \rightarrow (j)$

9.1. EXAMPLE (continued) Represent all given prec. constraints by OR-pre. constraints



Then the AND/OR-network
is bipartite

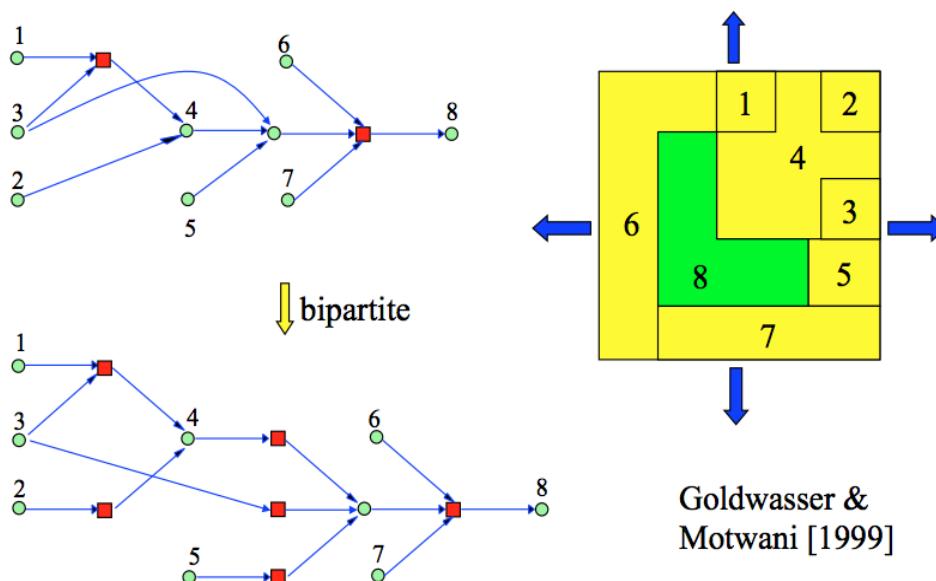
Consequences

- (1) Resulting AND/OR-network is bipartite
- (2) May just speak of a system \mathcal{W} of waiting conditions

$$(x, j) \stackrel{?}{=} x \left\{ \begin{array}{c} o \\ o \\ \vdots \\ o \end{array} \right\} \xrightarrow{\square} j$$

So a system of waiting conditions \Leftrightarrow bipartite AND/OR network

An application: Disassembly

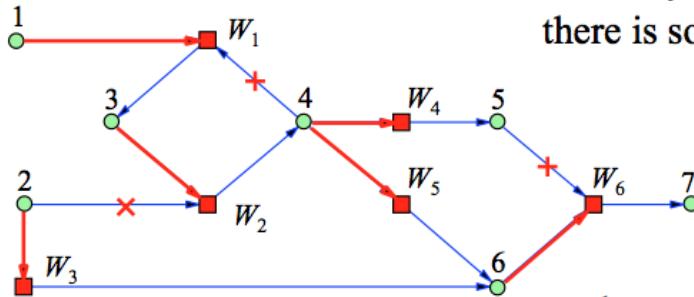


Question (1) When is a system of waiting conditions feasible

Feasibility of waiting conditions \mathcal{W}

A realization of \mathcal{W} is an acyclic graph $R = (V, A)$ on V s.t.

for every waiting condition (X, j) ,
there is some $i \in X$ preceding j



red arcs define a realization



special case: linear realization



An algorithm for testing feasibility

- Tries to construct a linear realization for \mathcal{W}
- Imitates topological sort for digraphs

List $L := []$

while (there is a job $i \in V$ that is not a waiting job of a condition in \mathcal{W})

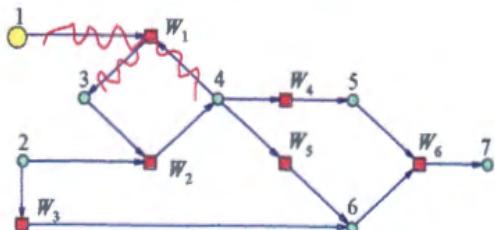
begin

 insert i at the end of L and delete it from V

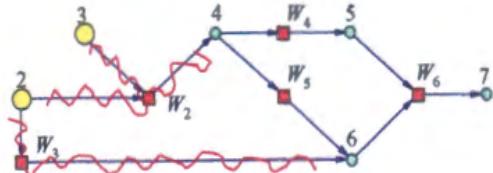
 if (some waiting condition (X, j) becomes satisfied)
 delete (X, j) from \mathcal{W}

end

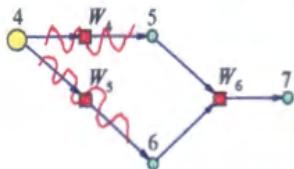
return L



$L = [1,]$



$L = [1, 2, 3,]$



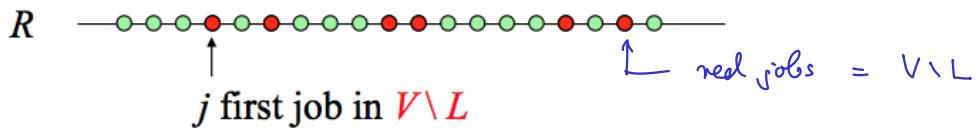
$L = [1, 2, 3, 4, 5, 6, 7]$

Correctness of the algorithm

\mathcal{W} feasible $\Leftrightarrow L$ contains all jobs

" \Leftarrow " is trivial

" \Rightarrow ": Let R be a linear realization but $L \neq V$



j not in $L \Rightarrow$ there is a waiting condition (X, j) with $X \subseteq V \setminus L$
 \Rightarrow all jobs in X are red

but in R , there must be a job $i \in X$, i.e., a red job, before j
 \Rightarrow contradiction

Consequences and further properties

- $V \setminus L$ corresponds to a [generalized cycle](#)
i.e., a set C fulfilling
 $j \in C \Rightarrow$ there is a waiting condition (X, j) with $X \subseteq C$
- The list L is an \subseteq -maximal feasible subset of V
- Every linear realization can be constructed by the algorithm

Def

(1)

(2)

Combinatorial properties:

- The linear realizations form the basic words of an antimatroid
[Korte & Lovasz \[1984\]](#)
- In the graph of linear realizations (adjacency \Leftrightarrow transposition),
vertex connectivity = minimum degree [Naatz \[2000\]](#)

Proof of (1) :

Let $V \setminus L \neq \emptyset$. Consider $C := V \setminus L$ and

$W_C :=$ all waiting conditions (X, j) with $X \cup \{j\} \subseteq C$

$\Rightarrow W_C$ is a relaxation of W

No job of C can be first in any linear realization of W_C

\Rightarrow no job of C can be in any linear realization of W

$\Rightarrow L$ is the \subseteq -maximal unique set of jobs that can be
in a (partial) linear realization

Proof (2) :

choose jobs in the algorithm in the order of the given

linear realization $R \Rightarrow$ algorithm produces $L = R$

Question (2) Does a feasible selection define a policy

9.5 Theorem: Every feasible selection s for $[G, F]$ defines a preselective policy (i.e. the planning rule is non-anticipative)

Proof needs the following lemma

9.6 Lemma: Let s be a feasible selection for $[G, \bar{s}]$

Then, for every vector x of processing times, the earliest start

$\underline{ES}_W[x]$ given by the system of waiting conditions defined by G and s is well defined

Proof: s feasible $\Rightarrow W$ is feasible

Consider the OR-nodes as dummy jobs

let $S = (S_1, \dots, S_N)$ be a vector of times S_j associated with the "real" jobs $j \in V$ and the "dummy" jobs $w \in W$ for a given x .

S is feasible for W

$$\Leftrightarrow \left\| \begin{array}{l} S_w \geq \min_{\text{arc } (j) \rightarrow w} [S_j + x_j] \quad \text{for OR-nodes } w \\ S_j \geq \max_{\text{arc } w \rightarrow j} [S_w + 0] \quad \text{for AND-nodes } j \end{array} \right.$$

$$\hookrightarrow \left\| \begin{array}{l} S_w \geq \min_{\text{arc } (j) \rightarrow w} [S_j + d_{jw}] \\ S_j \geq \max_{\text{arc } w \rightarrow j} [S_w + d_{wj}] \end{array} \right. \quad \text{special case of a system of min-max inequalities}$$

the general case of min-max inequalities has arbitrary d_{jw} and d_{wj} values (even negative)

$$\left\| \begin{array}{l} S_w \geq \min_{(j) \rightarrow w} [S_j + d_{jw}] \\ S_j \geq \max_{w \rightarrow j} [S_w + d_{wj}] \end{array} \right. \quad \left. \begin{array}{l} \text{is feasible if there} \\ \text{are } S_1, \dots, S_N \\ \text{filling all inequalities} \\ (\text{w.l.o.g. } S_j, S_w \geq 0) \end{array} \right.$$

9.7 Lemma

$j \in V$ AND node: $S_j \geq \max_{(w,j) \in A} \{S_w + d_{wj}\}$

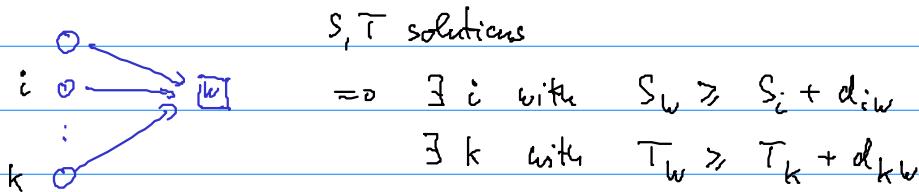
$w \in W$ OR node: $S_w \geq \min_{(j,w) \in A} \{S_j + d_{jw}\}$

} min-max system

$(\infty, \infty, \dots, \infty)$
is a solution

(S_1, \dots, S_n) and (T_1, \dots, T_n) solutions
 $\Rightarrow (\min\{S_1, T_1\}, \dots, \min\{S_n, T_n\})$ solution

Proof: a) Consider OR-node w

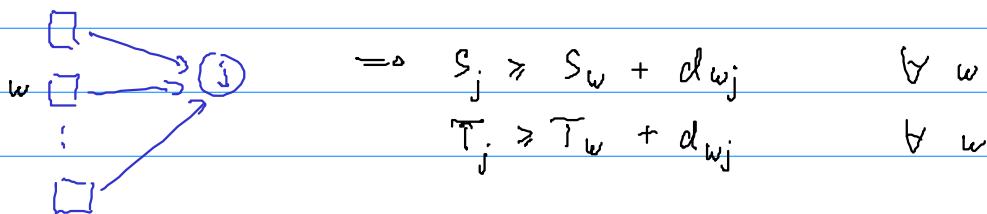


$$\Rightarrow S_w, T_w \geq \min \{ S_i + d_{iw}, T_k + d_{kw} \} \stackrel{w.l.o.g.}{=} S_i + d_{iw}$$

$$\Rightarrow \min \{ S_w, T_w \} \geq S_i + d_{iw} \geq \min \{ S_i, T_i \} + d_{iw}$$

\Rightarrow OR-inequality for node w is fulfilled by $\min \{ S, T \}$

b) Consider AND-node j



$$\Rightarrow \min \{ S_j, T_j \} = S_j \stackrel{w.l.o.g.}{\geq} S_w + d_{wj} \geq \min \{ S_w, T_w \} + d_{wj} \quad \forall w \quad \square$$

Proof of Lemma 9.6 continued:

W feasible \Rightarrow every (linear) realizer R of W defines a solution of the min-max system (by taking ES_R)
 \Rightarrow (Lemma 9.6) \exists unique minimal feasible solution $S = (S_1, \dots, S_N) \geq 0$
 \Rightarrow every S_j is then the earliest start of j w.r.t. W \square

Proof of Theorem 9.5

s feasible, W associated waiting conditions for $s, G \left\{ \begin{array}{l} \text{planning rule induced by } s \\ \text{all waiting conditions fulfilled with the same times} \end{array} \right. \Rightarrow \pi = ES_W$

Proof of non-anticipativity is similar to that for ES-policies:

let x, y look the same at time t to π and $\pi[x](j) = t$

if

$x_i = y_i \wedge$ jobs completed up to t

$\bar{x}_i = \bar{y}_i \wedge$ jobs busy at t

\Rightarrow all waiting conditions fulfilled with the same times

$\Rightarrow \pi[y](j) = t \quad \square$

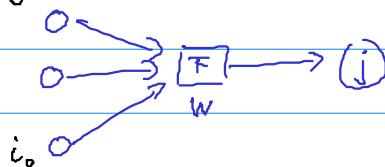
Question (3) What is the relationship with early start policies?

9.8 Theorem: let s be a feasible selection for $[G, \mathcal{F}]$, π be the associated prescriptive policy, and W the associated system of waiting conditions. Then

$$\pi = ES_W = \min \{ ES_R \mid R \text{ is a realizer of } W \}$$

Proof: Consider $F \in \mathcal{F}$ and the corresponding OR-node

let j be the waiting job



Then there is some $i_0 \in F \setminus \{j\}$ s.t. deleting all other acts $(i) \rightarrow [W]$
yields the same minimal solution S for a given x

if

modified problem
is tighter

S is unique minimal solution for tight problem

and F is settled by letting j wait for i_0 .

$\Rightarrow [G + (i_0, j), F - \{F\}]$ has the same min. solution S for x

Iteration $\Rightarrow (G + \{(i_F, j_F) \mid F \in F\}, \emptyset)$ has the same min
solution S

$=: R$

Claim: R is a realizer of w

clear by construction \square

So $S_j = ES_R[x](j)$ and R is acyclic

\Rightarrow min solution $S = ES_R$ \square

Note: We have not carefully distinguished between S in
the AND/OR network and S defined only on V .

The real meaning should be clear from the context

Question (4) How to compute earliest start times for a preselective policy?

\rightarrow § 10

Question (5) Do preselective policies have nice properties?

9.9 THEOREM: Let π be a policy for $[G, F]$

Then the following conditions are equivalent:

(i) π is preselective

(2) Π is monotone

$$[x \leq y \Rightarrow \Pi[x] \leq \Pi[y]]$$

(3) Π is continuous

Remark: (2), (3) show that Graham anomalies of type a)

occur in pairs (\Leftrightarrow non-monotonicity \Leftrightarrow non-continuity)

Proof: here only (1) \Rightarrow (2), (1) \Rightarrow (3)

The other directions are shown in § 11

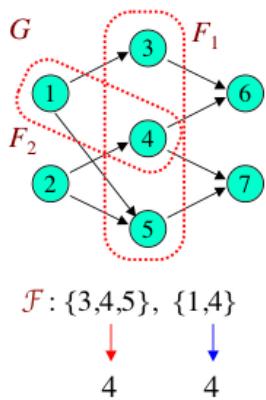
$$\Pi = \text{ES}_W = \min \{ \text{ES}_R \mid R \text{ realizer of } W \}$$



continuous, monotone, convex

preserves continuity, monotonicity, but not convexity

Preselective policies and AND/OR networks

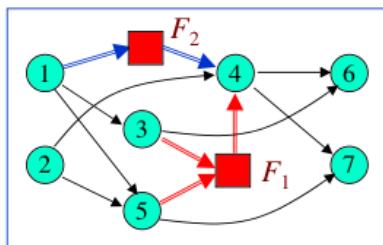


$$\mathcal{F}: \{3,4,5\}, \{1,4\}$$

↓ ↓

4 4

this choice of
waiting jobs
defines policy Π



AND/OR network representing Π

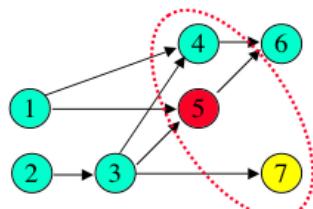
start in policy Π

= min/max of path lengths

= min/max of sums of processing times

$\Rightarrow \boxed{\Pi \text{ is continuous and monotone}}$

No anomalies for preselective policies



2 identical parallel machines
 $\Rightarrow F = \{4, 5, 7\}$ is only forbidden set

7

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

1	4	6
2	3	5

↑

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

1	4	6
2	3	5

↑

↑

Exercises

9.1

9.1 Theorem 9.8 shows that a preselective policy is the minimum of ES-policies. Consider now a set \mathcal{H} of feasible orders and set $\Pi = \min \{ \text{ES}_H \mid H \in \mathcal{H} \}$. Give necessary and sufficient conditions (on \mathcal{H}) for Π being a policy.

9.2

9.2 Let $\text{OPT}^{\text{PRES}}(K, Q) = \min \{ E_Q(K^\Pi) \mid \Pi \text{ preselective} \}$

be the optimum value over the class of preselective policies.

Show that there are instances with

$$\text{OPT}^{\text{PRES}}(K, Q) < \text{OPT}^{\text{ES}}(K, Q)$$

but that OPT^{PRES} and $\text{OPT}^{\text{PRIOR}}$ are incomparable in general