

§8 Constructing ES-policies

8.1 Lemma: H is feasible for $[G, F]$ iff

(1) $G \prec H$

(2) for every $F \in \mathcal{F}$, there are $i_F, j_F \in F$ with $i_F <_H j_F$

[the resource conflict given by F is settled

by letting j_F wait for i_F

[we also say that F is destroyed by $i_F < j_F$]

\leftarrow basis for algorithms

Proof: obvious, since no F is an antichain of H \square

Idea: solve conflicts on forbidden sets (iteratively) by telling "who must wait for whom"

May be contradictory if the choice is made arbitrarily at time 0

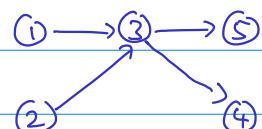
8.2 Example:



$F: \{\{2,3\}, \{3,4\}, \{1,2,4\}\}$

$\Rightarrow 2 \cdot 2 \cdot 6$ independent choices of $i_F < j_F$
only 9 lead to (feasible) orders

e.g. $2 < 3, 3 < 4, 1 < 4 \Rightarrow$



$2 < 3, 3 < 4, 4 < 1 \Rightarrow$ contradiction

$E_H \cup \{(2,3), (3,4), (4,1)\}$ not acyclic

Fact: $H + \text{choice of } i_F < j_F$ defines an ES-policy

$\Leftrightarrow H + \text{choice is acyclic}$

Organize construction in a tree, the conflict settling tree

root = G

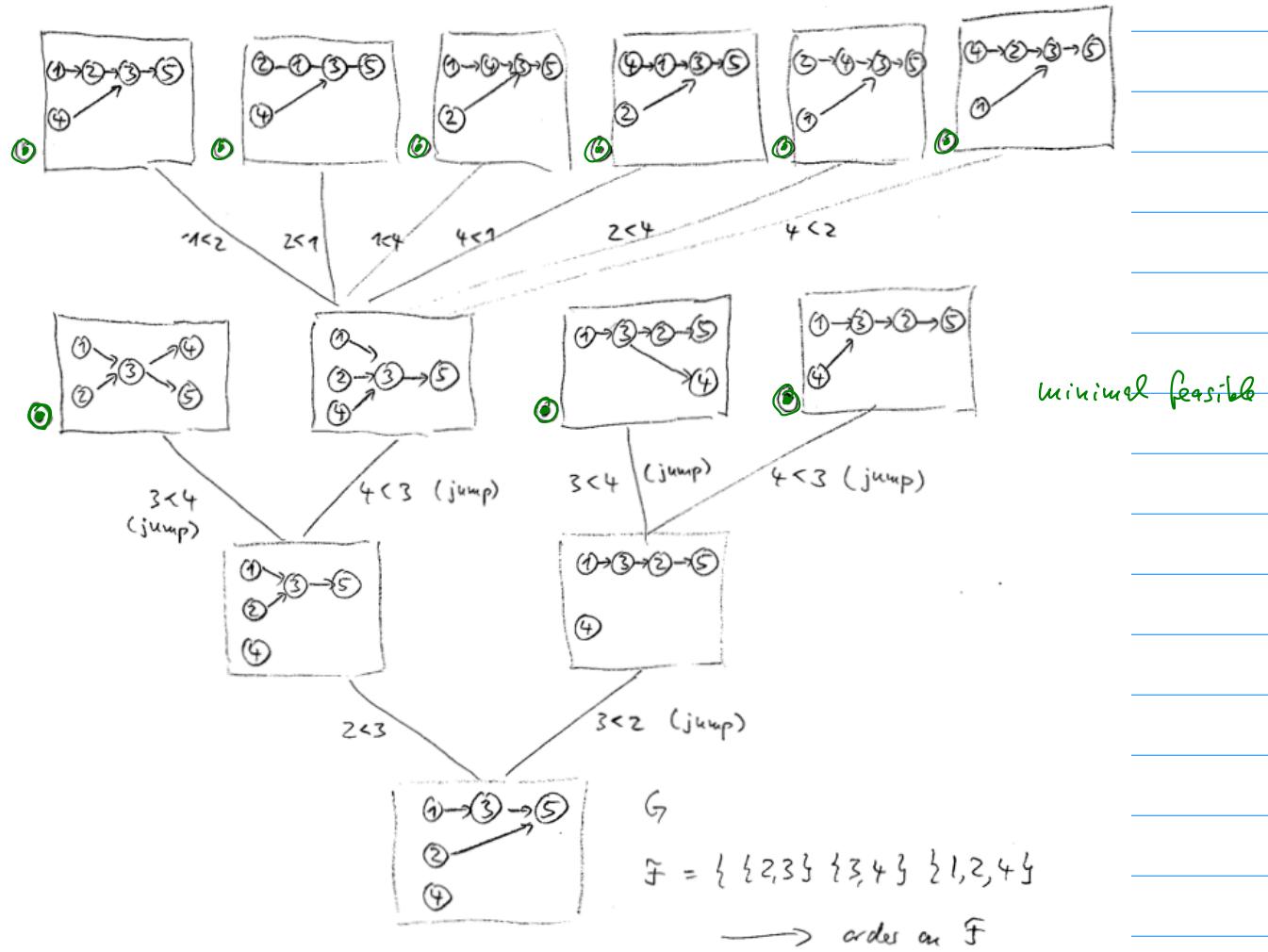
nodes = extensions of G

children of a node H = all extensions H' obtained by settling
the conflict on one yet unsettled forbidden set

may use a suitable ordering of the forbidden set

(can be done in the deterministic case together
with the construction of a schedule)

leaves = feasible order



8.3 Theorem: The conflict settling tree is a suborder (not necessarily induced) of $E(G)$ containing all minimal feasible orders

Proof:

- suborder is trivial

- not induced:

all the order relations $H_1 \prec H_2$ in the tree are also present in $E(G)$.

But there may be more

- all minimal feasible orders are contained

Let H minimal feasible and F_1, \dots, F_k be the ordering defining the tree

Lemma 8.1 \Rightarrow for every F_r there is some $i_{F_r} < j_{F_r}$ in H

adding these in the order l, \dots, k yields H

Note: not every leaf of the tree is minimal feasible □

Remarks

- Conflict tree may be used for Branch & Bound algorithms for calculating an optimal schedule for given K and x
- it can be combined with the dynamic decision view, i.e. branch only at decision times

In this way, only a subset of all forbidden sets will be considered

8.5 REMARK: In the stochastic case, the optimal values by priority policies / ES-policies are incomparable

[remember: in the deterministic case, ES is better]

8.6 Example

$$(1) \text{ OPT}^{\text{PRIORITY}} < \text{OPT}^{\text{ES}}$$

G: $\begin{array}{c} \textcircled{1} \xrightarrow{\text{?}} \textcircled{3} \\ \textcircled{2} \xrightarrow{\text{?}} \textcircled{4} \end{array}$ $\Sigma = \{3, 4\}$ $x^1 = (1, 2, 1, 1)$ prob. $1/2$
 $x^2 = (2, 1, 1, 1)$ prob $1/2$

$$K = C_{\max}$$

π priority policy

$$\pi[x^1]$$

1	3	4	(
2			

⋮

3

$$\pi[x^2] : \begin{array}{|c|c|} \hline 1 & 3 \\ \hline 2 & 4 \\ \hline \end{array} \vdots$$

$$\mathbb{E}[K^{\pi}] = 3$$

minimal feasible orders



$$E[K^{H_1}] = 3.5$$

$$E[\kappa^{H_2}] = 3.5$$

$$(2) \quad OPT^{ES} < OPT^{PRIORITY}$$

already in the deterministic case

Exercises:

- 8.1 Construct an example in which the conflict settling tree contains leaves that are not minimal feasible

8.2 What is the complexity of adding a precedence constraint $i_p < j_F$ to an order H ? What is the best data structure for the orders in the conflict settling tree to allow fast updating w.r.t. to adding precedence constraints?