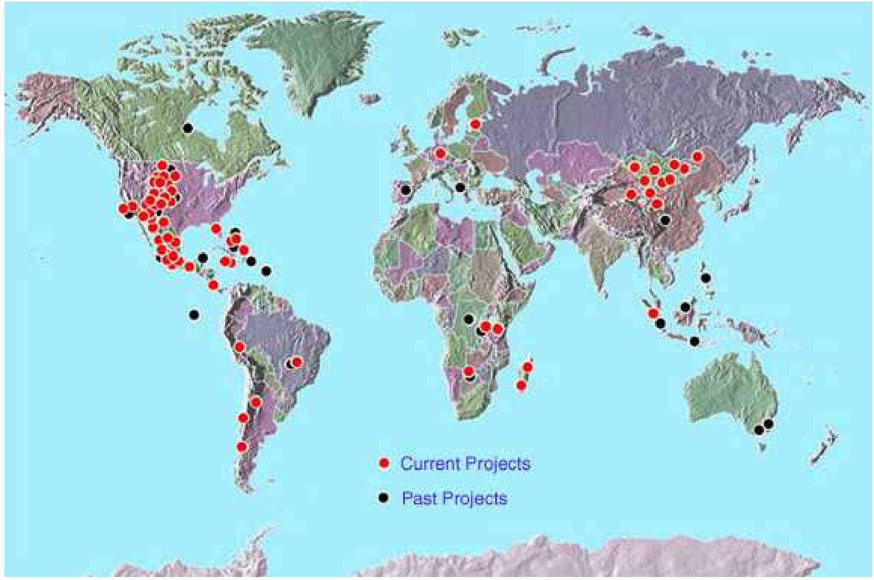
Project Scheduling with Sequence-Dependent Changeover Times: A Branch-and-Bound Approach

Outline

1. Forbidden sets and maximum cuts
2. Breaking up forbidden sets
3. Schedule-generation scheme
4. Computational experience
5. Conclusions

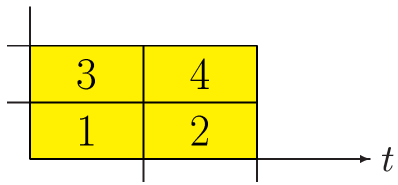


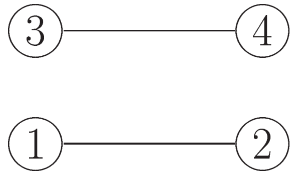
1 Forbidden sets and maximum cuts

- **Weak triangle inequality** $\vartheta^k_{hi} + p_i + \vartheta^k_{ij} \geq \vartheta^k_{hj} \quad (h, i, j \in V_k)$
  - ▷ Relation $O_k(S)$ is strict order in set $\overline{V}_k$
  - ▷ $\mathcal{A}_k(S)$: longest antichain of $O_k(S)$ = maximum-weight stable set in comparability graph of $O_k(S)$

- **Example:** $O_k(S)$ may not be interval order

ϑ^k_{ij}	1	2	3	4
1	—	0	0	1
2	0	—	1	0
3	0	1	—	0
4	1	0	0	—





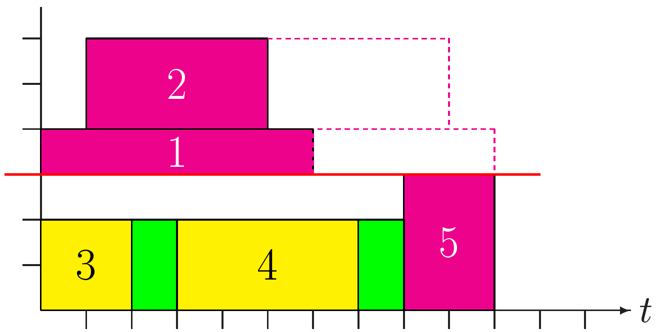
- $F \subseteq V$ **forbidden set** of activities:

$$\sum_{i \in F} r_{ik} > R_k \text{ for some } k \in \mathcal{R}$$

- $\mathcal{A} \subseteq V$ **active set** for given schedule S and resource k:

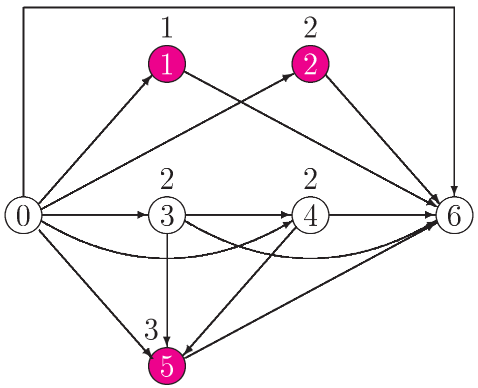
$$[S_i, S_i + p_i + \vartheta_{ij}^k[\cap [S_j, S_j + p_j + \vartheta_{ji}^k[\neq \emptyset \text{ for all } i, j \in \mathcal{A}$$

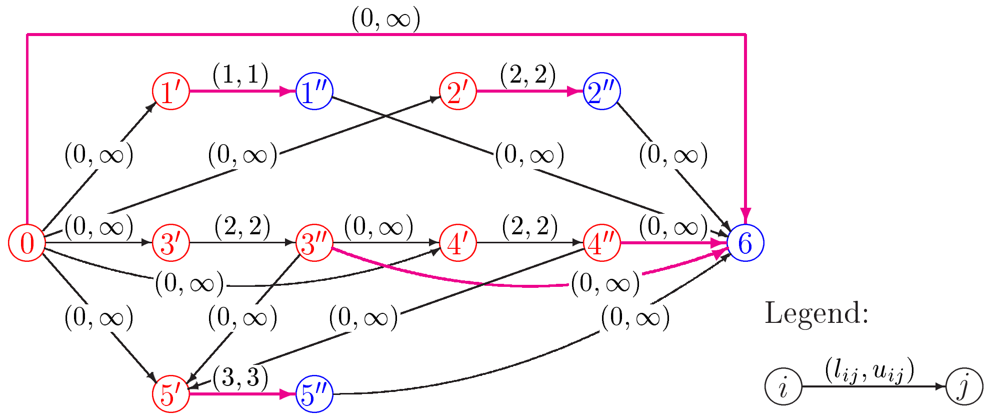
- $\mathcal{A}$ active set iff $\mathcal{A}$ is antichain in $O_k(S)$
- Requirement $\sum_{i \in \mathcal{A}} r_{ik} \leq \overline{r}_k(S)$
- Schedule S changeover-feasible only if no active set is forbidden
- For each $k \in \mathcal{R}$ find active set $\mathcal{A}_k(S)$ with maximum requirement $\sum_{i \in \mathcal{A}_k(S)} r_{ik}$



How to find an active set with maximum requirement?

- Split nodes $i \in V_k$ of precedence graph $G_k(S)$ into two nodes i' and i''
- Link nodes i' and i'' by arc (i', i'') with lower and upper capacities $l_{i'i''} = u_{i'i''} = r_{ik}$
- **Proposition (Möhring 1985).**
  Maximum $(0, n+1)$-cuts C in $\overline{G}_k(S)$ are uniformly directed
- Any path from 0 to $n+1$ is cut exactly once
- Set $U := \{i \in V_k \mid (i', i'') \in C\}$ is an active set
- Since $l_{0i'} = l_{i''(n+1)} = l_{i''j'} = 0$: $\sum_{i \in U} r_{ik} =$ capacity of $C = \overline{r}_k(S)$
- Set U coincides with active set $\mathcal{A}_k(S)$ of maximum requirement





2 Breaking up forbidden sets

- Given forbidden set F, $B \subset F$ **minimal delaying alternative** for F if
  - ▷ $F \setminus B$ is not forbidden
  - ▷ $F \setminus B'$ is forbidden for any $B' \subset B$

- Breaking up forbidden set $F = \mathcal{A}_k(S)$:
  Introduce **disjunctive precedence constraint**

  $$\min_{j \in B} S_j \geq \min_{i \in A}(S_i + p_i + \vartheta_{ij}^k)$$

  between set $A := F \setminus B$ and minimal delaying alternative B

- Minimizing f subject to temporal and disjunctive precedence constraints can be done in $\mathcal{O}(n^2\nu + \min[n, \nu]\overline{d}\nu \log n)$ time

3 Schedule-generation scheme

- **Resource relaxation**

$$\left.\begin{array}{ll} \text{Minimize} & f(S) \\ \text{subject to} & S_0 = 0 \\ & S_j - S_i \geq \delta_{ij} \quad (\langle i,j \rangle \in E) \\ & \overline{r}_k(S) \leq R_k \quad (k \in \mathcal{R}) \end{array}\right\} \quad (PS\infty|temp, s_{ij}|reg)$$

- **Schedule-generation scheme**

01 Solve resource relaxation: schedule S

02 Determine $\mathcal{A}_k(S)$ for all $k \in \mathcal{R}$

03 IF no $\mathcal{A}_k(S)$ forbidden THEN stop (∗ schedule S is feasible ∗)

04 ELSE branch over minimal delaying alternatives B for set $\mathcal{A}_k(S)$

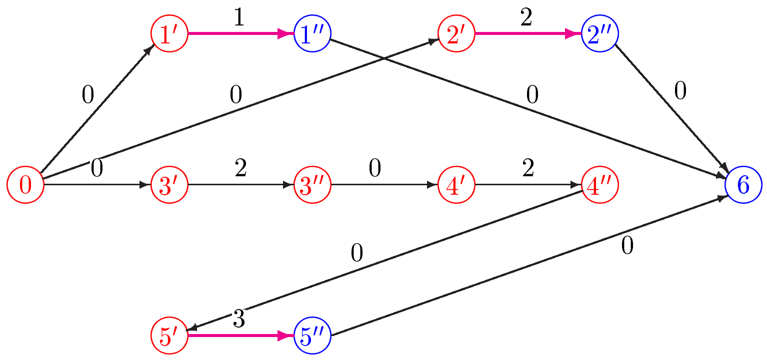
05 In each node add disjunctive precedence constraints constraints
$\min_{j\in B} S_j \geq \min_{i\in A}(S_i + p_i + \vartheta_{ij}^k)$ to resource relaxation

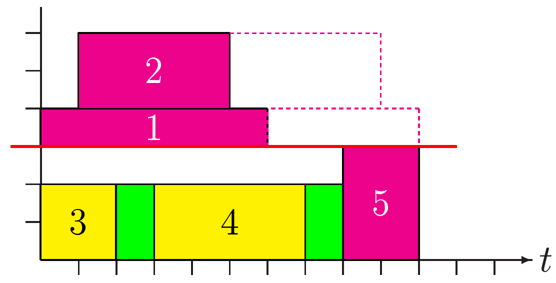
06 Select one enumeration node and GOTO 01

Example (cntd.):

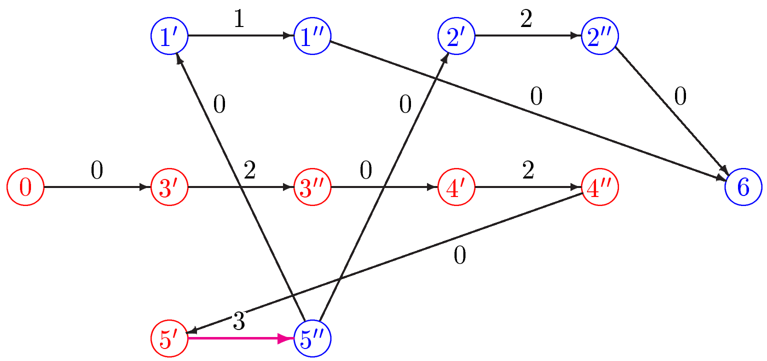
$\mathcal{A}_k(S) = \{1, 2, 5\}$

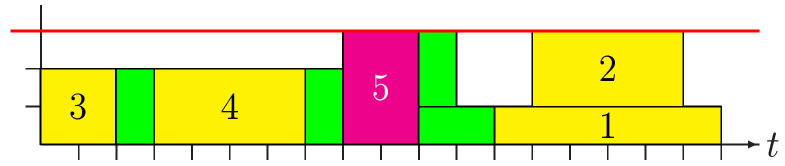
$(A, B) = (\{5\}, \{1, 2\})$





$\mathcal{A}_k(S) = \{5\}$





4 Computational experience

- Testset: 360 instances with 10, 20, 50, 100 activities and 5 resources each generated by ProGen/max
- $RF = 0.75$, $RS \in \{0, 0.25, 0.5\}$, $OS \in \{0.25, 0.5, 0.75\}$
- $\vartheta_{ij}^k \approx 0.25 p_i$, variation coefficient $vc \approx 0.75$
- Objective function: Project duration S_{n+1}
- Pentium PII with 333 MHz and 128 MB RAM
- Branch-and-bound algorithm in C with time limit of 100 seconds

	p_{opt}	p_{uns}	p_{feas}	p_{unk}	Δ_{LB}
$n = 10$	76.67%	23.33%	0.0%	0.0%	6.34%
$n = 20$	65.56%	27.78%	6.67%	0.0%	6.59%
$n = 50$	28.89%	21.11%	40.00%	10.00%	8.84%
$n = 100$	15.56%	14.44%	44.44%	25.56%	7.07%

5 Conclusions

- Summary
  - ▷ Relax resource constraints
  - ▷ Checking changeover-feasibility of a schedule: minimum flow problems
  - ▷ Maximum cuts provide (forbidden) active sets
  - ▷ Break up forbidden sets by disjunctive precedence constraints

- Future Research
  - ▷ Local search procedures based on concepts presented
  - ▷ Material flows between sites: Cumulative resources and transshipment problems
  - ▷ Stochastic durations/changeover times
    - ∗ Replace schedules by strict orders O, starting with $O = \emptyset$
    - ∗ Check changeover-feasibility of strict orders O by computing maximum cuts
    - ∗ Expand strict orders O by pairs (i, j): precedence constraints $S_j \geq S_i + p_i + \vartheta_{ij}^k$