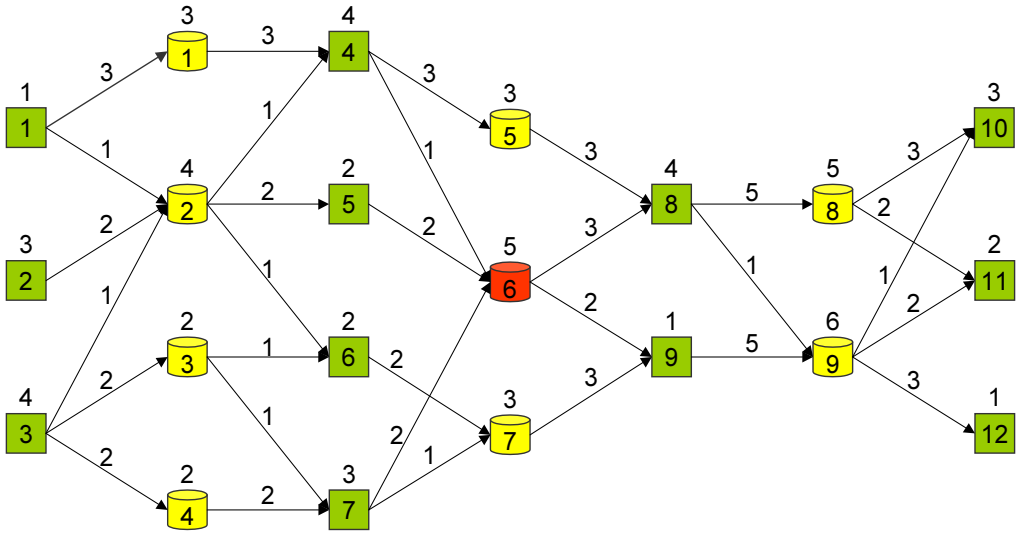
Scheduling with Storage Resources

Ch. Schwindt,
University of Karlsruhe (TH)

Outline

1. Problem
2. Model
3. Solution methods
   3.1 Branch-and-bound
   3.2 Priority-rule method
4. Conclusions

1 Problem



- **Given:**
  - ▷ Set of operations executed on dedicated processing units
  - ▷ Set of input and output products depleted and replenished in batch mode
  - ▷ Input and output quantities for operations
  - ▷ Storage facilities of finite capacity for stocking products
  - ▷ Initial stocks and prescribed safety stocks for products
  - ▷ Minimum and maximum time lags between start times of operations

- **Sought:**
  - ▷ Feasible production schedule minimizing some regular objective function

2 Model

- **Notations**
  - ▷ O: Set of operations $i = 0, 1, \ldots, n, n+1$ with processing times p_i ($p_0 = p_{n+1} = 0$)
  - ▷ $\mathcal{R}$: Set of storage resources k with safety stocks $\underline{R}_k$ and storage capacities $\overline{R}_k$
  - ▷ r_{ik}: Increase in inventory level of resource k by execution of operation i
  - ▷ $O_k^- = \{i \in O \mid r_{ik} < 0\}$, $O_k^+ = \{i \in O \mid r_{ik} > 0\}$: Sets of depleting and replenishing operations for resource k
  - ▷ δ_{ij}: Time lag between linked operations $(i,j) \in E \subseteq O \times O$, distances d_{ij}
  - ▷ $S = (S_0, S_1, \ldots, S_{n+1})$: Production schedule
  - ▷ $r_k(S,t) = \sum_{i \in O_k^- : S_i \le t} r_{ik} + \sum_{i \in O_k^+ : S_i + p_i \le t} r_{ik}$: Inventory in resource k at time t
  - ▷ $f : \mathbb{R}_{\ge 0}^{n+2} \to \mathbb{R}$: Regular objective function in start times S_i ($i \in O$)

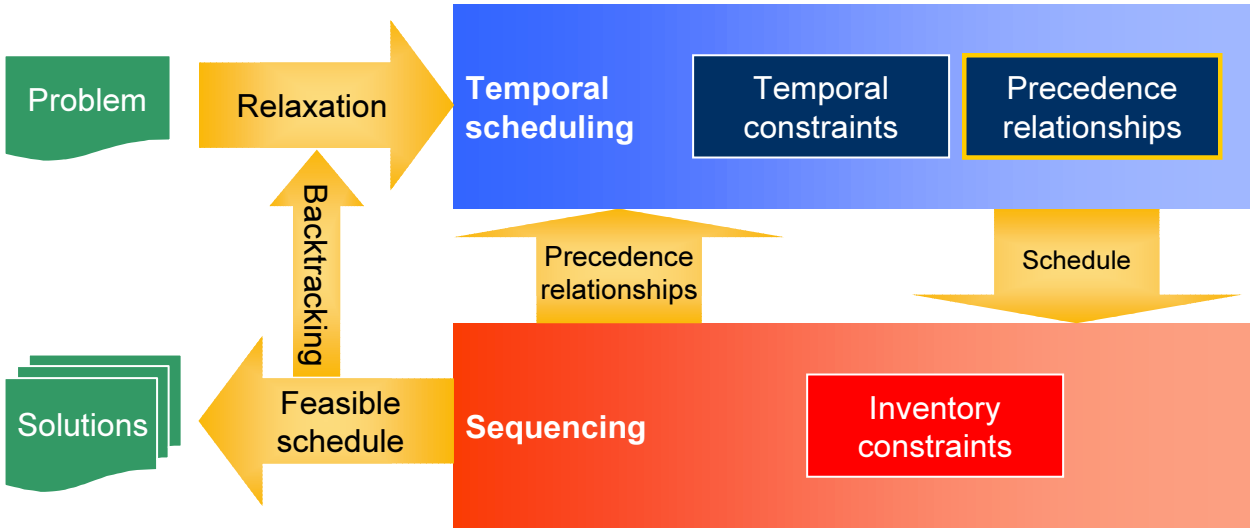
- **Model**

$$\begin{cases} \text{Minimize} & f(S) & \\ \text{subject to} & \underline{R}_k \le r_k(S,t) \le \overline{R}_k & (k \in \mathcal{R},\ t \ge 0) \\ & S_j - S_i \ge \delta_{ij} & ((i,j) \in E) \\ & S_i \ge 0 & (i \in O) \end{cases}$$

3 Solution Methods

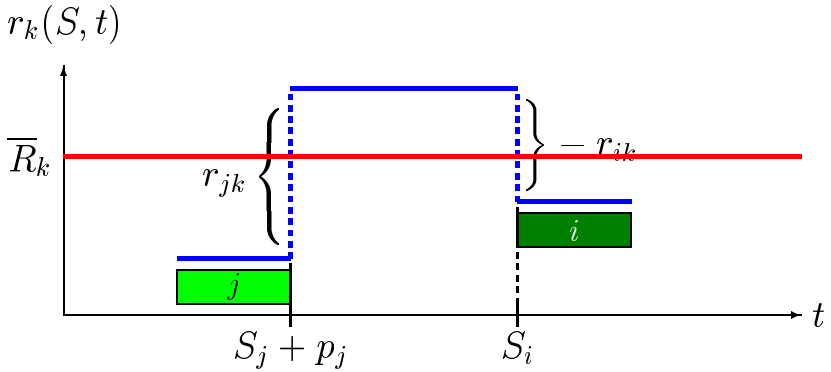
3.1 Branch-and-Bound

- Scheduling is ...
  - ▷ defining precedence relationships between operations competing for scarce resources (Sequencing: hard)
  - ▷ optimizing objective function subject to prescribed time lags and established precedence relationships (Temporal scheduling: tractable)
- Enumeration scheme



Resolving resource conflicts

- Inventory excess at time t: $r_k(S,t) > \overline{R}_k$



  - ▷ Delay completion of some operation $j \in O_k^+$ up to start of some operation $i \in O_k^-$
  - ▷ Precedence relationship $S_j + p_j \geq S_i$: $\delta_{ij} = -p_j$ (maximum time lag)

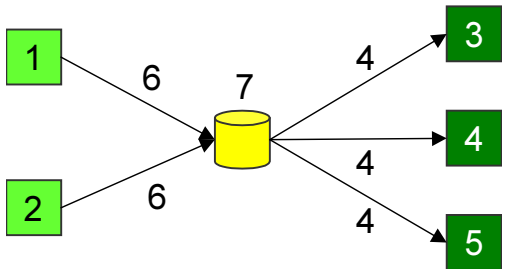
- Inventory shortage at time t: $r_k(S,t) < \underline{R}_k$

  - ▷ Delay start of some operation $j \in O_k^-$ up to completion of some operation $i \in O_k^+$
  - ▷ Precedence relationship $S_j \geq S_i + p_i$: $\delta_{ij} = p_i$ (minimum time lag)

3.2 Priority-rule method

Why classical priority-rule methods don't work

- Stepwise expand partial schedule by scheduling one eligible operation in each iteration
- Machine scheduling or project scheduling with renewable resources: partial schedules are feasible
- Scheduling with storage resources:
  - ▷ Material-availability constraints and storage-capacity constraints
  - ▷ Feasibility of partial schedules would require simultaneous scheduling of several operations



  - ▷ Allow for infeasible partial schedules

Two-phase approach

- Phase 1: Scheduling subject to material-availability constraints
  - ▷ Relax storage-capacity constraints
  - ▷ Operation j eligible if
    - ○ all operations $i \in O$ with $d_{ij} \geq 0$ and $d_{ji} < 0$ have been scheduled
    - ○ for resulting partial schedule, terminal inventories do not fall below safety stocks
  - ▷ Select some eligible activity j^* according to priority indices $\pi(j)$
  - ▷ Schedule j^* at time $t^* = \min\{t \geq ES_{j^*} \mid r_k(S^{\mathcal{C}}, \tau) + r_{jk} \geq \underline{R}_k \text{ for } k \in \mathcal{R},\ \tau \geq t\}$
- Pegging by precedence relationships according to FIFO strategy
  - ▷ Iterate operations $i \in O_k^+$ in order of nondecreasing $S_i + p_i$ and allot output of $i \in O_k^+$ to operations $j \in O_k^-$ in order of nondecreasing S_j
  - ▷ Introduce time lag $\delta_{ij} = p_i$ between $i \in O_k^+$ and $j \in O_k^-$ consuming output of i
- Phase 2: Scheduling subject to precedence and storage-capacity constraints
  - ▷ Relax material-availability constraints
  - ▷ Proceed analogously to phase 1, replacing material-availability with storage-capacity constraints

Serial schedule-generation scheme (phase 1)

$u := 0$;
2: $S_0 := 0$, $\mathcal{C} := \{0\}$;
for all $i \in O$ **do** ($*$ initialize ES_i and LS_i $*$)
  $ES_i := d_{0i}$, $LS_i := -d_{i0}$;
while $\mathcal{C} \neq O$ **do**
  $\mathcal{E} := \{j \in O \setminus \mathcal{C} \mid Pred(j) \subseteq \mathcal{C},\ \sum_{i \in \mathcal{C} \cup \{j\}} r_{ik} \geq \underline{R}_k \text{ for } k \in \mathcal{R}\}$;
  if $\mathcal{E} = \emptyset$ **then** terminate;
  $j^* := \min\{j \in \mathcal{E} \mid \pi(j) = \text{ext}_{h \in \mathcal{E}} \pi(h)\}$;
  $t^* := \min\{t \geq ES_{j^*} \mid r_k(S^{\mathcal{C}}, \tau) + r_{j^*k} \geq \underline{R}_k \text{ for } k \in \mathcal{R},\ \tau \geq t\}$;
  if $t^* > LS_{j^*}$ **then** ($*$ unschedule and restart $*$)
    $u := u + 1$;
    if $u > \overline{u}$ **then** terminate;
    $\mathcal{U} := \{i \in \mathcal{C} \mid LS_{j^*} = S_i - d_{j^*i}\}$;
    for all $i \in \mathcal{U}$ **do** $d_{0i} := S_i + t^* - LS_{j^*}$;
    update distances d_{ij} for all $i, j \in O$ and **goto** line 2;
  else ($*$ schedule j^* at time t^* $*$)
    $S_{j^*} := t^*$, $\mathcal{C} := \mathcal{C} \cup \{j^*\}$;
    for all $j \in O \setminus \mathcal{C}$ **do** ($*$ update ES_j and LS_j $*$)
      $ES_j := \max(ES_j, S_{j^*} + d_{j^*j})$;
      $LS_j := \min(LS_j, S_{j^*} - d_{jj^*})$;
return S;

4 Experimental performance analysis

- Branch-and-bound algorithm
- Randomized multi-pass priority-rule based method
- 36 problem instances with 12 to 90 operations
- Storage capacity and storage time settings FIS, FWQ, FWS

No.	n	FIS	FWQ	FWS	LB	S^{bb}_{n+1}	Term.	S^{pr}_{n+1}
1	12	no	no	no	4	12	$*$	12
2	24	no	no	no	8	16	$*$	16
3	36	no	no	no	12	20	$*$	20
4	48	no	no	no	16	24	$*$	24
5	60	no	no	no	20	28	$*$	28
6	72	no	no	no	24	32	$*$	32
7	15	no	yes	no	7	15	$*$	15
8	30	no	yes	no	11	19	$*$	19
9	24	no	yes	no	15	23	$*$	23
10	60	no	yes	no	19	27	$*$	27
11	75	no	yes	no	23	32		32
12	90	no	yes	no	27	35	$*$	35
13	15	no	no	yes	4	12	$*$	12
14	30	no	no	yes	8	16	$*$	16
15	45	no	no	yes	12	20	$*$	∞
16	60	no	no	yes	16	24	$*$	24
17	75	no	no	yes	20	28	$*$	28
18	90	no	no	yes	24	32	$*$	32
19	12	yes	no	no	4	12	$*$	12
20	24	yes	no	no	8	16	$*$	16
21	36	yes	no	no	12	20	$*$	20
22	48	yes	no	no	16	24	$*$	24
23	60	yes	no	no	20	28	$*$	28
24	72	yes	no	no	24	32	$*$	32
25	15	yes	yes	no	7	∞	$*$	∞
26	30	yes	yes	no	11	19	$*$	19
27	45	yes	yes	no	15	23	$*$	25
28	60	yes	yes	no	19	27	$*$	27
29	75	yes	yes	no	23	32		32
30	90	yes	yes	no	27	35		37
31	15	yes	no	yes	4	12	$*$	12
32	30	yes	no	yes	8	16	$*$	16
33	45	yes	no	yes	12	∞		∞
34	60	yes	no	yes	16	24	$*$	24
35	75	yes	no	yes	20	28	$*$	28
36	90	yes	no	yes	24	32	$*$	32

5 Conclusions

- Scheduling with storage resources
  - ▷ Operations consuming input products and producing output products
  - ▷ Prescribed safety stocks and limited storage capacities for products
  - ▷ Minimum and maximum time lags between operations
- Branch-and-bound method
  - ▷ Relax inventory constraints
  - ▷ Branch over alternative precedence relationships resolving resource conflicts
- Priority-rule method
  - ▷ Two-phase method
  - ▷ Ensure material availability by precedence relationships
- Talk by Norbert Trautmann
  - ▷ Renewable resources
  - ▷ Sequence-dependent changeover times on processing units
  - ▷ Performance analysis comparing priority-rule to branch-and-bound method