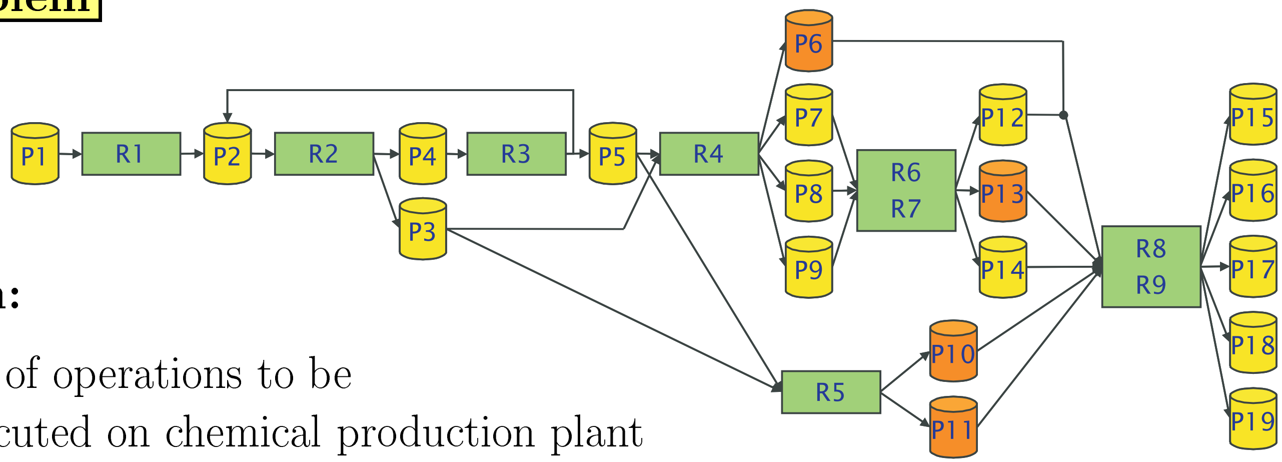
Scheduling Continuous Material Flows

Ch. Schwindt,
University of Karlsruhe

Outline

1. Problem
2. Model
3. Solution Method
4. Implementationary Issues
5. Conclusions

1 Problem



- **Given:**
  - ▷ Set of operations to be executed on chemical production plant
  - ▷ Plant operated in continuous production mode
  - ▷ Operations executed on processing units controlled by operators
  - ▷ Intermediate products buffered in storage facilities of finite capacity
  - ▷ Safety stock for storable intermediate products
  - ▷ Minimum and maximum time lags between operations
- **Sought:**
  - ▷ Feasible schedule minimizing some convex objective function

2 Model

- **Notations**
  - ▷ $\mathcal{O}$: Set of operations $i = 1, \ldots, n$ with processing time p_i
  - ▷ $\mathcal{C}$: Set of cumulative resources k with safety stock $\underline{R}_k$ and capacity $\overline{R}_k$
  - ▷ r_{ik}: Increase in inventory level of k after completion of i
  - ▷ δ_{ij}: Time lag between linked operations $(i,j) \in E$
  - ▷ $S = (S_1, \ldots, S_n)$: Production schedule
  - ▷ $x_i(S,t)$: Portion of operation $i \in \mathcal{O}$ processed by time t
  - ▷ $r_k(S,t) = \sum_{i\in\mathcal{O}} r_{ik} x_i(S,t)$: Inventory in resource $k \in \mathcal{C}$ at time t
  - ▷ $f : \mathbb{R}^n_{\geq 0} \to \mathbb{R}$: Convex objective function in start times S_i $(i \in \mathcal{O})$

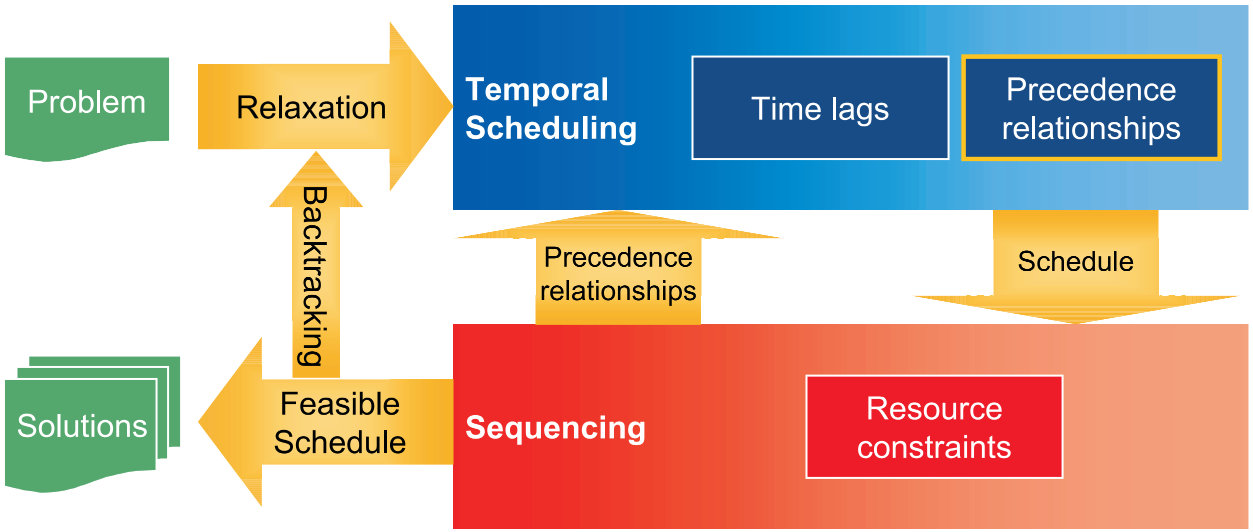
- **Model**

$$\left\{\begin{array}{lll} \text{Minimize} & f(S) & \\ \text{subject to} & \underline{R}_k \leq r_k(S,t) \leq \overline{R}_k & (k \in \mathcal{C},\ t \geq 0) \\ & S_j - S_i \geq \delta_{ij} & ((i,j) \in E) \\ & S_i \geq 0 & (i \in \mathcal{O}) \end{array}\right.$$

3 Solution Method

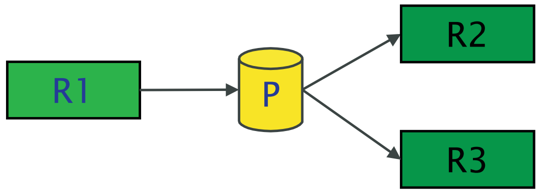
- Scheduling is . . .
  - ▷ defining precedence relationships between operations competing for scarce resources (Sequencing: hard)
  - ▷ optimizing objective function subject to prescribed time lags and established precedence relationships (Temporal scheduling: tractable)

- Solution method



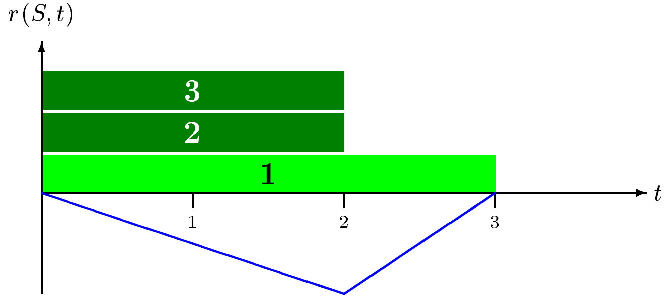
Cumulative resources in continuous mode: An example

- **Given:**
  - ▷ Operations $i = 1, 2, 3$ replenishing/depleting cumulative resource at constant rates
  - ▷ $p_1 = 3$, $r_1 = 2$, $S_1 = 0$
  - ▷ $p_2 = p_3 = 2$, $r_2 = r_3 = -1$



- **Sought:** Start times S_2 and S_3 such that
  - ▷ inventory permanently nonnegative ($\underline{R} = 0$, $\overline{R} = \infty$)
  - ▷ makespan $C_{\max} = \max_{i \in \mathcal{O}}(S_i + p_i)$ minimum

- **Relaxation:**
  - ▷ Disregard resource constraints
  - ▷ Earliest schedule $S = (0, 0, 0)$
  - ▷ Inventory falls below safety stock



Settling inventory shortage at time t

▷ *Select operations settling the shortage*

▷ *Optimize simultaneous displacements under resource constraints*

- Partition $\mathcal{O}_k := \{j \in \mathcal{O} \mid r_{jk} \neq 0\}$ into two sets A and B
- **Set A**
  - ▷ Operations $j \in \mathcal{O}_k^- := \{j \in \mathcal{O}_k \mid r_{jk} < 0\}$ have to be completed by deadline t
  - ▷ Operations $j \in \mathcal{O}_k^+ := \{j \in \mathcal{O}_k \mid r_{jk} > 0\}$ are released at ready time t

$$\left.\begin{array}{ll} S_j + p_j \leq t & (j \in A \cap \mathcal{O}_k^-) \\ S_j \geq t & (j \in A \cap \mathcal{O}_k^+) \end{array}\right\} \qquad (1)$$

- Inventory shortage at time t caused by operations $j \in A$: $\underline{R}_k - \sum_{j \in A \cap \mathcal{O}_k^-} r_{jk}$

- **Set B**
  - ▷ Operations $j \in \mathcal{O}_k^-$ cannot be completed before time t
  - ▷ Operations $j \in \mathcal{O}_k^+$ must not be started after time t
  - ▷ Schedule operations $j \in B$ such that net production not less than shortage by A:

$$\sum_{j \in B} r_{jk} x_j(S,t) \geq \underline{R}_k - \sum_{j \in A \cap \mathcal{O}_k^-} r_{jk}$$

  - ▷ Introduce decision variables x_j:

$$0 \leq x_j \leq 1 \quad (j \in B) \tag{2}$$

$$\sum_{j \in B} r_{jk} x_j \geq \underline{R}_k - \sum_{j \in A \cap \mathcal{O}_k^-} r_{jk} \tag{3}$$

$$\left.\begin{array}{ll} S_j + p_j x_j \geq t & (j \in B \cap \mathcal{O}_k^-) \\ S_j + p_j x_j \leq t & (j \in B \cap \mathcal{O}_k^+) \end{array}\right\} \tag{4}$$

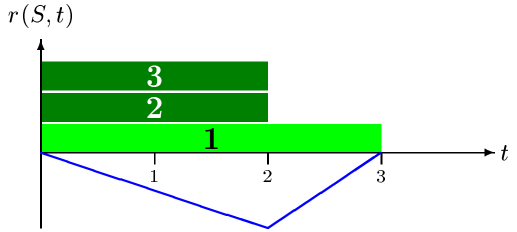
- **Finiteness of solution method**
  - ▷ Constraints (1) to (4) definitively settle inventory shortage at time t
  - ▷ There exists $i \in \mathcal{O}_k^+$ with $r_k(S, S_i) < \underline{R}_k$ or $i \in \mathcal{O}_k^-$ with $r_k(S, S_i + p_i) < \underline{R}_k$
  - ▷ Constant t in inequalities (1) and (4) can be replaced with decision variable S_i $(+p_i)$
  - ▷ All shortages settled after n iterations

- **Case of inventory excess**
  - ▷ Interchange sets $\mathcal{O}_k^-$ and $\mathcal{O}_k^+$ in inequalities (1) and (4)
  - ▷ Replace (3) with

$$\sum_{j \in B} r_{jk} x_j \leq \overline{R}_k - \sum_{j \in A \cap \mathcal{O}_k^+} r_{jk} \tag{3'}$$

  - ▷ Feasible schedule after $2n$ iterations

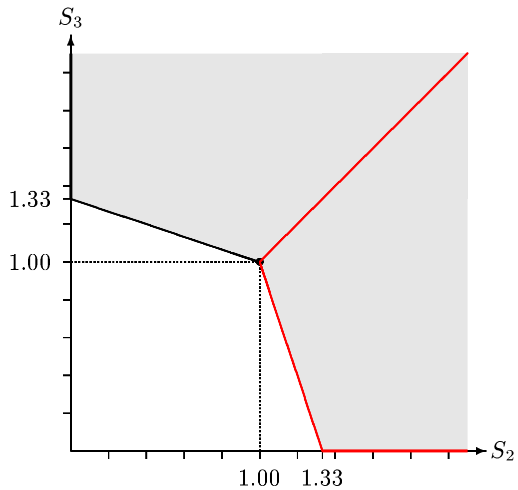
Example (ctd.)



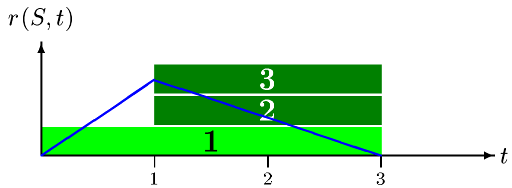
$i = 3,\ t = S_i + p_i$

$A = \{3\},\ B = \{1, 2\}$

$$\begin{aligned} \text{Minimize} \quad & \max(3, S_2 + 2, S_3 + 2) \\ \text{subject to} \quad & S_2 - S_3 + 2x_2 \geq 2 \\ & -S_3 + 3x_1 \geq 2 \\ & -x_2 + 2x_1 \geq 1 \\ & 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1 \end{aligned}$$



Optimal solution:
$x_1 = 1,\ x_2 = 1,\ S_2 = 1,\ S_3 = 1$



4 Implementationary Issues

- Enumeration of partitions $\{A, B\}$ for given inventory shortage
  - ▷ Binary tree, each level belongs to one $j \in \mathcal{O}_k$
  - ▷ For each $j \in \mathcal{O}_k$ branch over $j \in A$ or $j \in B$
  - ▷ Leaves of tree are partitions $\{A, B\}$
  - ▷ In each node solve temporal scheduling problem with relaxation

$$\sum_{j \in B} r_{jk} x_j \geq \underline{R}_k - \sum_{j \in A \cap \mathcal{O}_k^-} r_{jk} - \sum_{j \in \mathcal{O}_k^+ \setminus A \setminus B} r_{jk} \qquad (3'')$$

    of (3) by some dual method
  - ▷ Stop branching as soon as conflict is settled in resulting schedule S
  - ▷ Resume branching if conflict reappears later on

- Binary branch-and-bound tree of height $\mathcal{O}(n^2)$

5 Conclusions

- Storage problems in process scheduling
  - ▷ Storage capacity settings
  - ▷ Storage homogeneity settings
  - ▷ Storage time settings
- Scheduling continuous material flows
  - ▷ Disregard resource constraints
  - ▷ Settle conflicts by partitioning sets of operations
  - ▷ For given partition, replace resource constraint by linear inequality
- Future Research
  - ▷ Development of efficient heuristics based on concepts presented
  - ▷ Integration of further constraints like setup times or calendars