A Comparison of Relaxation-Based Enumeration Schemes in Production Scheduling

Ch. Schwindt, Clausthal University of Technology
supported by German Research Foundation DFG

Outline

1. Production scheduling problem
2. Generic scheduling model
3. Relaxation-based enumeration schemes
4. Avoiding redundancy
5. Performance analysis
6. Conclusions
1 Production scheduling problem

Operations
- Processing of production order (job) on machine
- Execution of chemical process (task) on processing unit
- Performance of activity in project using personnel and equipment

Temporal relationships
- Precedence constraints arising from process plans or recipes
- Release dates, deadlines
- Quarantine times, shelf life times

Resources
- Machinery, tools, manpower
- Storage facilities, intermediate products

Problem: Determine production schedule (assignment of start times to operations) complying with temporal relationships and resource constraints
2 Generic scheduling model

Resource-constrained scheduling model

- Operations $i$ with processing times $p_i$, including production start $i = 0$
- Temporal relationships: minimum and maximum time lags $d_{ij}^{\text{min}}$ and $d_{ij}^{\text{max}}$ between start times of operations $i, j$
- Manpower, machinery: renewable resources $k$ with capacities $R_k$ and requirements $r_{ik}$
- Storage facilities, intermediate products: cumulative resources $l$ with minimum and maximum inventory levels $R_l$ and $\bar{R}_l$ and requirements $r_{il}$ ($r_{0l}$: initial stock)
2. Generic scheduling model

Replace operations by events

- Split each operation \( i \neq 0 \) in start and completion events \( e = s(i) \) and \( f = c(i) \)
- Define time lags \( \delta_{ef} \) between events \( e \) and \( f \)
  - Fixed processing times \( p_i \): \( \delta_{ef} = p_i, \delta_{fe} = -p_i \) with \( e = s(i) \) and \( f = c(i) \)
  - Minimum and maximum time lags \( d_{ij}^{\text{min}} \) and \( d_{ij}^{\text{max}} \): \( \delta_{ef} = d_{ij}^{\text{min}}, \delta_{fe} = -d_{ij}^{\text{max}} \) with \( e = s(i) \) and \( f = s(j) \)

Replace renewable resources by cumulative resources

- Renewable resources \( k \): transform into cumulative resources \( l \) with \( R_l = 0, R_l = R_k \), \( r_{el} = r_{ik} \) for \( e = s(i) \) and \( r_{fl} = -r_{ik} \) for \( f = c(i) \)

Eliminate maximum inventory levels, normalize minimum inventory levels

- Maximum inventory levels \( R_i \): introduce cumulative resources \( l' \) with inventory levels \( R_{ll'} = -R_l, R_{ll'} = \infty \) and requirements \( r_{el'} = -r_{el} \), put \( R_l := \infty \)
- Minimum inventory levels \( R_l \): put \( r_{0l} := r_{0l} - R_l, R_l := 0 \)
Generic scheduling model

**Notation**

- $V$ Set of events $e$
- $E \subseteq V \times V$ Temporal relation
- $N = (V, E, \delta), D$ Event-on-node network, distance matrix
- $\mathcal{R}$ Set of cumulative resources
- $S_e, S = (S_e)_{e \in V}$ Occurrence time of event $e$, schedule
- $r_l(S, t) = \sum_{e \in V : S_e \leq t} r_{el}$ Inventory level of resource $l$ at time $t$ given schedule $S$
- $f(S)$ Objective function, e.g., $f(S) = \max_{e \in V} S_e$
- $S$ Set of feasible schedules (feasible region)


\[
\begin{align*}
\text{Minimize} & \quad f(S) \\
\text{subject to} & \quad r_l(S, t) \geq 0 \quad (l \in \mathcal{R}, \ t \geq 0) \\
& \quad S_f - S_e \geq \delta_{ef} \quad ((e, f) \in E) \\
& \quad S_0 = 0, \ S_e \geq 0 \quad (e \in V) \\
\end{align*}
\]  

\[(\text{PSP})\]
3 Relaxation-based enumeration schemes

3.1 Basic scheme

Scheduling is (Bell and Park 1990) . . .

- defining precedence relationships between events competing for same resources (Sequencing: hard)

- optimizing objective function subject to prescribed time lags and established precedence relationships (Temporal scheduling: tractable)
3.2 Resolving inventory shortages

- Schedule \( \hat{S} \) not resource-feasible: determine some time \( t \geq 0 \) with \( r_t(\hat{S}, t) < 0 \)
- Determine set \( A := \{ e \in V \mid \hat{S}_e > t, \ r_{el} > 0 \} \)
- Compute minimal delaying alternatives \( B \): \( \subseteq \)-minimal set of events \( f \) with \( \hat{S}_f \leq t \) and \( r_t(\hat{S}, t) - \sum_{f \in B} r_{fl} \geq 0 \)
- Add precedence relationships between sets \( A \) and \( B \)
  - Release dates: Fest et al. (1999)
    \[ S_f \geq \min_{e \in A} \hat{S}_e \quad (f \in B) \]
  - Ordinary precedence constraints (branch over all \( e \in A \)): De Reyck, Herroelen (1998)
    \[ S_f \geq S_e \quad (f \in B) \]
  - Disjunctive precedence constraints: Neumann et al. (2001)
    \[ \min_{f \in B} S_f \geq \min_{e \in A} S_e \]
Example

Inventory shortage at time \( t = 10 \), \( A = \{3, 5\} \), \( B_1 = \{2\} \), \( B_2 = \{4\} \), \( \min_{e \in A} \hat{S}_e = 12 \)

<table>
<thead>
<tr>
<th>RD</th>
<th>( S_2 \geq 12 )</th>
<th>Schedule ( S^1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( S_4 \geq 12 )</td>
<td>Schedule ( S^2 )</td>
</tr>
<tr>
<td>OPC</td>
<td>( S_2 \geq S_5 )</td>
<td>Schedule ( S^1 )</td>
</tr>
<tr>
<td></td>
<td>( S_4 \geq S_5 )</td>
<td>Schedule ( S^3 )</td>
</tr>
<tr>
<td></td>
<td>( S_2 \geq S_3 )</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>( S_4 \geq S_3 )</td>
<td>—</td>
</tr>
<tr>
<td>DPC</td>
<td>( S_2 \geq \min{S_3, S_5} )</td>
<td>Schedule ( S^1 )</td>
</tr>
<tr>
<td></td>
<td>( S_4 \geq \min{S_3, S_5} )</td>
<td>Schedule ( S^3 )</td>
</tr>
</tbody>
</table>
4 Avoiding redundancy

4.1 Partitioning the feasible region

- Consider enumeration node $u$ with search space $Q$
- Compute minimal delaying alternatives $B_1, \ldots, B_\nu$
- Define disjunctive precedence constraints $\min_{f \in B_\mu} S_f \geq \min_{e \in A} S_e$ belonging to sets
  \[ P_\mu := \{ S \in Q \mid \min_{f \in B_\mu} S_f \geq \min_{e \in A} S_e \} \]
- Enumerate child nodes $v_1, \ldots, v_\nu$ with search spaces
  \[ Q_\mu := P_\mu \setminus \left[ \bigcup_{\lambda=1}^{\mu-1} P_\lambda \right] \]
- $\bigcup_{\mu=1}^{\nu} (Q_\mu \cap S) = Q \cap S$ and $Q_\lambda \cap Q_\mu = \emptyset$ for all $\lambda \neq \mu$
- Construction of sets $Q_\mu$
  - Introduce disjunctive precedence constraint $\min_{f \in B_\mu} S_f \geq \min_{e \in A} S_e$ at node $v_\mu$
  - Introduce reverse constraint $\min_{e \in A} S_e \geq \min_{f \in B_\mu} S_f + 1$ at all nodes $v_{\mu+1}, \ldots, v_\nu$
4.2 Generalized subset dominance

- Release dates, ordinary precedence constraints: time lags $\delta_{ef}$
- Associate a distance matrix $D(u)$ with each enumeration node $u$
- Node $u$ dominated by node $v$ if $Q(u) \subseteq Q(v)$, i.e., $D(u) \geq D(v)$: Neumann, Zimmermann (2002)
- Perform depth-first search: enumeration nodes $v$
  - on active path from root $r$ to active node $u$
  - bud nodes
  - fully explored (all descendant nodes explored)
- Generalized subset dominance rule: fathom node $u$ if
  - there exists bud node $v$ with $D(v) \leq D(u)$: S. (1998)
  - there exists fully explored node $v$ with distance one from active path and $D(v) \leq D(u)$: De Reyck, Herroelen (1998)
- Each search space $Q(u)$ explored only once
5 Performance analysis

Test bed

- Test set from literature with 90 instances comprising 50 events and 10 resources each
- Pentium IV PC with 1.8 GHz clock pulse and 512 MB RAM, time limit 10 seconds
- Branch-and-bound algorithms for makepan problem coded under MS Visual C++ 6.0
  - RD(-SSD): release dates (+ subset dominance)
  - OPC(-SSD): ordinary precedence constraints (+ subset dominance)
  - DPC(-PFR): disjunctive precedence constraints (+ partitioning of feasible region)

Computational results

<table>
<thead>
<tr>
<th></th>
<th>RD</th>
<th>RD-SSD</th>
<th>OPC</th>
<th>OPC-SSD</th>
<th>DPC</th>
<th>DPC-PFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number instances solved</td>
<td>71</td>
<td>79</td>
<td>74</td>
<td>79</td>
<td>87</td>
<td>90</td>
</tr>
<tr>
<td>Number of nodes explored</td>
<td>49045</td>
<td>15784</td>
<td>7383</td>
<td>1229</td>
<td>1110</td>
<td>204</td>
</tr>
<tr>
<td>CPU time total [ms]</td>
<td>2117</td>
<td>1304</td>
<td>2047</td>
<td>1622</td>
<td>413</td>
<td>254</td>
</tr>
<tr>
<td>CPU time first solution [ms]</td>
<td>2</td>
<td>1</td>
<td>140</td>
<td>81</td>
<td>8</td>
<td>59</td>
</tr>
</tbody>
</table>
6 Conclusions

Summary

- Production scheduling problem
- Generic scheduling model with cumulative resources
- Different relaxation-based enumeration schemes
  ▷ Release dates
  ▷ Ordinary precedence constraints
  ▷ Disjunctive precedence constraints
- Avoid redundancy by partitioning feasible region or subset dominance

Further research

- Integration of further constraints
  ▷ Sequence-dependent changeover times
  ▷ Multi-purpose intermediate storages
- Application to process scheduling problems
Cumulative resources


Enumeration schemes


Redundancy avoidance