Project Scheduling Under Generalized Precedence Relations
A Survey of Structural Issues, Solution Approaches, and Applications

Christoph Schwindt
Operations Management Group

14th International Conference on Project Management and Scheduling, Munich
Outline

1 Introduction
   - Precedence relations
   - Resource constraints
   - Objective functions
   - Project scheduling problems with generalized precedence relations

2 Temporal analysis

3 Structural issues
   - Characterization of the feasible region
   - Efficient solutions
   - Generic solution approaches

4 Solution approaches for the project duration problem

5 Expansions
   - Multi-mode problem
   - Preemptive problem

6 Applications

7 Conclusions
Project scheduling problem

Consider project with \( n \) activities \( i \in V \) of durations \( p_i \in \mathbb{Z}_{\geq 0} \).

**Project scheduling problem**: assign execution times to each activity \( i \)

\[
y_i : \mathbb{R}_{\geq 0} \rightarrow \{0, 1\} \text{ such that } \int_{0}^{\infty} y_i(t) \, dt = p_i
\]

**Non-preemptive problem**: activities cannot be interrupted

\( \rightarrow \) Solution to scheduling problem specified by start times \( S_i \) or completion times \( C_i = S_i + p_i \) of all activities \( i \in V \)

**Preemptive scheduling problem**: activities can be interrupted and resumed later on

\( \rightarrow \) Solution to scheduling problem specified by trajectories

\[
z_i : \mathbb{R}_{\geq 0} \rightarrow [0, 1], \quad t \mapsto z_i(t) = \frac{1}{p_i} \int_{0}^{t} y_i(t) \, dt
\]

for all activities \( i \in V \)
Precedence relations and resource constraints

- Activities have to be scheduled subject to **precedence relations** and **resource constraints** so as to optimize one or several measures of project performance.

- **Precedence relations**: elements \((i, j)\) of binary relation \(E \subseteq V \times V\) on activity set \(V\) defining conditions on execution times of activities \(i\) and \(j\).

- Pairs \((i, j)\) may be associated by some **extra data** like time lags \(\delta_{ij}\) or execution percentages \(\xi_i\) and \(\xi_j\).

- **Resource constraints**: limited availability of manpower, machinery, materials, money, energy supply, . . .

- Scheduling goals formulated as **objective function(s)** in decision variables \(S_i, C_i\) or functions \(y_i, z_i\).
Types of precedence relations

1. **Ordinary precedence relations** (Kelley 1961)
   
   
   \[(i, j) : S_j \geq C_i\]

2. **Generalized precedence relations** (Roy 1964)

   \[(i, j, \delta_{ij}) : S_j \geq S_i + \delta_{ij}\]

3. **Feeding precedence relations** (Kis 2005, Alfieri et al. 2011)

   \[(i, j, \xi_i) : S_j \geq \min\{t \mid z_i(t) = \xi_i\}\]

4. **Generalized work precedence relations** (Quintanilla et al. 2012)

   \[(i, j, \xi_i, \xi_j) : \max\{t \mid z_j(t) = \xi_j\} \geq \min\{t \mid z_i(t) = \xi_i\}\]

5. **Generalized feeding precedence relations** (S. and Paetz 2014)

   \[(i, j, \xi_i, \xi_j, \delta_{ij}) : \max\{t \mid z_j(t) = \xi_j\} \geq \min\{t \mid z_i(t) = \xi_i\} + \delta_{ij}\]
Example

Generalized feeding precedence relation \((i, j, 0.25, 0.4, 3)\)
Resource constraints

- Different types of resources considered in project scheduling: renewable, nonrenewable, doubly-constrained, storage, partially renewable, continuous resources

- In this talk: renewable resources $k$ from a set $\mathcal{R}$
  - Each resource $k \in \mathcal{R}$ consists of $R_k \in \mathbb{N}$ identical units (capacity)
  - Each activity uses $r_{ik} \in \mathbb{Z}_{\geq 0}$ units when being in progress

- Resource constraints: joint requirements of activities $i$ for resources must not exceed the resource capacities at any point in time

\[
\sum_{i \in V} r_{ik} y_i(t) \leq R_k \quad (k \in \mathcal{R}; t \geq 0)
\]
Objective functions

- Scheduling goals specified by single or several objective functions $f$

- In this talk: single-criterion problems

- Regular objective function
  $C \leq C' \Rightarrow f(C) \leq f(C')$
  - Project duration $f(C) = \max_{i \in V} C_i$
  - Total tardiness cost $f(C) = \sum_{i \in V} w_i(C_i - d_i)^+$

- Nonregular objective functions
  - Net present value $f(C) = \sum_{i \in V} c_i^F \beta C_i$
  - Total squared utilization cost (resource leveling)
    $f(y) = \sum_{k \in R} c_k \int_0^\infty \left( \sum_{i \in V} r_{ik} y_i(t) \right)^2 dt$
Resource-constrained project scheduling problem

- **General project scheduling problem** with generalized (feeding) precedence relations and renewable-resource constraints

\[
(P) \quad \begin{align*}
\text{Min.} & \quad f(y) \\
\text{s. t.} & \quad \int_0^\infty y_i(t) \, dt = p_i \quad (i \in V) \\
& \quad t_j^+(\xi_j) \geq t_j^- (\xi_i) + \delta_{ij} \quad ((i, j) \in E) \\
& \quad \sum_{i \in V} r_{ik} y_i(t) \leq R_k \quad (k \in R; t \geq 0)
\end{align*}
\]

- **Non-preemptive version** with \( A(S, t) := \{ i \in V \mid S_i \leq t < S_i + p_i \} \)

\[
(P) \quad \begin{align*}
\text{Min.} & \quad f(S) \\
\text{s. t.} & \quad S_j \geq S_i + \delta_{ij} \quad ((i, j) \in E) \\
& \quad \sum_{i \in A(S, t)} r_{ik} \leq R_k \quad (k \in R; t \geq 0) \\
& \quad S_i \geq 0 \quad (i \in V)
\end{align*}
\]

Set of feasible schedules: \( S \), set of time-feasible schedules: \( S_T \)
Time-constrained project scheduling problem

\[
(P_T) \begin{cases} 
\text{Min.} & f(S) \\
\text{s. t.} & S_j \geq S_i + \delta_{ij} \quad ((i, j) \in E) \\
& S_i \geq 0 \quad (i \in V)
\end{cases}
\]

Generalized precedence relations \((i, j, \delta_{ij})\) represent minimum and maximum time lags between starts of activities \(i\) and \(j\)

- \(\delta_{ij} = p_i\): ordinary precedence relation \(S_j \geq S_i + p_i = C_i\)
- \(\delta_{ij} > p_i\): delayed precedence relation \(S_j \geq C_i + (\delta_{ij} - p_i)\)
- \(0 \leq \delta_{ij} < p_i\): minimum time lag allowing overlapping of \(i\) and \(j\)
- \(\delta_{ij} < 0\): maximum time lag of \(-\delta_{ij}\) between starts of \(j\) and \(i\)

\[S_j \geq S_i + \delta_{ij} \iff S_i \leq S_j - \delta_{ij}\]

Completion-to-start, completion-to-completion, and start-to-completion time lags can be transformed into start-to-start time lags
MPM (activity-on-node) project network (Roy 1964)

- Generalized precedence relations represented by MPM network $N = (V, E, \delta)$
- Activities correspond to nodes, precedence relations to arcs
- Introduce nodes 0 and $n + 1$ for project beginning and termination
- Nonnegativity conditions $S_i \geq 0$ can be replaced by $S_0 = 0$

Example: MPM network for project with four real activities
Modeling practical constraints

- Release date \( r_i \) of activity \( i \): \( \delta_{0i} = r_i \)
- Quarantine time \( q_i \) of activity \( i \): \( \delta_{i(n+1)} = p_i + q_i \)
- Deadline \( \overline{d}_i \) for completion of activity \( i \): \( \delta_{i0} = -\overline{d}_i + p_i \)
- Fixed start time \( t_i \) for activity \( i \): \( \delta_{0i} = t_i, \delta_{i0} = -t_i \)
- Simultaneous start of activities \( i \) and \( j \): \( \delta_{ij} = \delta_{ji} = 0 \)
- Simultaneous completion of activities \( i \) and \( j \): \( \delta_{ij} = p_i - p_j, \delta_{ji} = p_j - p_i \)
- Processing activities \( i, j \) immediately one after another: \( \delta_{ij} = p_i, \delta_{ji} = -p_i \)
- Minimum overlapping time \( \ell_{ij} \) of \( i \) and \( j \): \( \delta_{ij} = \ell_{ij} - p_i, \delta_{ji} = \ell_{ij} - p_j \)
- Maximum makespan \( C_{max}^U \) for activity set \( U \): \( \delta_{ij} = -C_{max}^U + p_i \) for \( i, j \in U \)
- Time-varying resource capacities: dummy activities with fixed start times
- Time-varying resource requirements: sequence of sub-activities pulled tight
- ...
Temporal analysis with MPM

- **MPM**: Metra Potential Method
- Interpret project network as electric circuit
- **Potential**: assignment $S : V \rightarrow \mathbb{R}_{\geq 0}$
- **Tensions**: differences $S_j - S_i$ of potentials
- Generalized precedence relations: lower bounds $\delta_{ij}$ on tensions $S_j - S_i$
- Dual $(D)$ of problem $(P_T)$ with $f(S) = \sum_{i \in V \setminus \{0\}} S_i - (n + 1)S_0$

\[
(D) \begin{cases}
\text{Max.} & \sum_{(i,j) \in E} \delta_{ij} \cdot \varphi_{ij} \\
\text{s. t.} & \sum_{(i,j) \in E} \varphi_{ij} - \sum_{(j,i) \in E} \varphi_{ji} = \begin{cases} -1 & \text{for } i \in V \setminus \{0\} \\ (n + 1) & \text{for } i = 0 \end{cases} \\
& \varphi_{ij} \geq 0 \quad ((i,j) \in E)
\end{cases}
\]

is longest-walk problem in $N$
Temporale analysis with MPM

Fundamentals
- $S_T \neq \emptyset$ iff $N$ does not contain any cycle of positive length
- Induced time lag $d_{ij} := \min_{S \in S_T} (S_j - S_i) =$ length of longest walk from $i$ to $j$ in $N$ (“distance”)
- Earliest start time $ES_i = d_{0i}$, latest start time $LS_i = -d_{i0}$

Algorithms and complexities (with $m := |E|$)
- $S_T \neq \emptyset$ and single time lag $d_{ij}$: transformation of Bianco and Caramia (2010) to unit-capacity transshipment problem, $O(m)$
- All time lags (distance matrix $D = (d_{ij})_{i,j \in V}$): Floyd-Warshall-Algorithm, $O(n^3)$
- Update of distance matrix after increase of single $d_{ij}$: Algorithm of Bartusch et al. (1988), $O(n^2)$
- Earliest and latest schedules $ES$ and $LS$: label-correcting algorithm for longest-walk calculations, $O(mn)$
Resource-constrained problem \( (P) \): Complexity and decomposition

- Problem \( (P) \) is \( \mathcal{NP} \)-hard
- The feasibility variant of problem \( (P) \) is \( \mathcal{NP} \)-complete

Decomposition theorem (Neumann and Zhan 1995)

An instance of problem \( (P) \) is feasible if and only if for each strong component \( G \) of project network \( N \) there exists a feasible subschedule for the execution of all activities of \( G \).

- Classical schedule-generation schemes must be modified to avoid or to cope with deadlocks
- Decomposition theorem is basis of heuristic decomposition methods
**Example**

Assume $R = 3$ and schedule activities with **serial schedule-generation** scheme

- **Deadlock** for activity $i = 2$ after three iterations
- **Conclusion**: start times of activities cannot be fixed during scheduling
Bartusch’s Lemma

- Forbidden set $F \subseteq V$: $\sum_{i \in F} r_{ik} > R_k$ for some $k \in \mathcal{R}$
- Forbidden set $F$ broken up by schedule $S$: $\mathcal{A}(S, t) \not\supseteq F$ for all $t \geq 0$

Lemma (Bartusch et al. 1988)

1. An $\subseteq$-minimal forbidden set $F$ is broken up by schedule $S$ iff $F$ contains two activities $i, j$ with $S_j \geq S_i + p_i$.
2. Schedule $S$ is resource-feasible iff all $\subseteq$-minimal forbidden sets $F$ are broken up.

Consequences:

- Resource constraints can be expressed as disjunctions of ordinary precedence relations $(i, j)$
- Feasible region is union of finitely many relation polyhedra

$$S_T(\rho) = \{ S \in S_T \mid S_j \geq S_i + p_i \text{ for all } (i, j) \in \rho \}$$
Covering of $S$ by relation polyhedra

Capacity $R = 2$

$F_1 = \{1, 2, 3\}$,
$F_2 = \{1, 2, 4\}$

1) $3 \rightarrow 1, 1 \rightarrow 4$
2) $3 \rightarrow 1, 2 \rightarrow 4$
3) $3 \rightarrow 1, 4 \rightarrow 1$
4) $3 \rightarrow 1, 4 \rightarrow 2$
5) $3 \rightarrow 2, 1 \rightarrow 4$
6) $3 \rightarrow 2, 2 \rightarrow 4$

Legend:
① : $S_T(\rho_1)$
② : $S_T(\rho_2)$
③ : $S_T(\rho_3)$
④ : $S_T(\rho_4)$
⑤ : $S_T(\rho_5)$
⑥ : $S_T(\rho_6)$

$\bigcup$ : $S$
Feasible relations (S. 2005)

Definition: Feasible relation

Relation $\rho$ with $\emptyset \neq S_T(\rho) \subseteq S$ is called feasible relation.

- Condition $S_T(\rho) \neq \emptyset$: ordinary precedence relations $(i, j) \in \rho$ are compatible with generalized precedence relations $(i', j') \in E$
- Condition $S_T(\rho) \subseteq S$: all schedules $S$ satisfying the ordinary precedence relations $(i, j) \in \rho$ are resource-feasible

Induced strict order, schedule-induced order, iso-order set

- Relation network $N(\rho) = (V, E \cup \rho, \delta)$ with $\delta_{ij} = p_i$ for $(i, j) \in \rho$
- Distance matrix $D(\rho)$ associated with network $N(\rho)$
- Relation $\rho$ induces strict order $\Theta(\rho) := \{(i, j) \mid d_{ij}(\rho) \geq p_i\}$
- Schedule $S$ induces strict order $\theta(S) := \{(i, j) \mid S_j \geq S_i + p_i\}$
- Iso-order set $S_{\bar{T}}^{=} (\theta) := \{S \in S_T \mid \theta(S) = \theta\}$
Example

Relation network for $\rho = \{(3, 2), (4, 2), (5, 1)\}$ and strict order $\Theta(\rho)$
Checking feasibility of relations (Kaerkes and Leipholz 1977)

1. $S_T \neq \emptyset$ iff $N(\rho)$ does not contain any cycle of positive length
2. $S_T(\rho) \subseteq S$:
   - For each resource $k$ weight activities $i \in V$ with $r_{ik}$
   - Condition is satisfied iff for each resource $k$, weight of any antichain $A_k(\rho)$ of $\Theta(\rho)$ does not exceed $R_k$
   - Maximum-weight antichain can be computed in $O(n^3)$ time by solving maximum-cut problem in precedence graph of $\Theta(\rho)$

Example: Maximum-weight antichain for $\rho = \{(3, 2), (4, 2), (5, 1)\}$

- Weight nodes with requirements $r_{ik}$
- Determine maximum $0 - (n + 1)$-cut
- Here: $A(\rho) = \{4, 5\}$
Schedule types (Neumann et al. 2000)

- **Active schedules**: Minimal points of $S$
- **Stable schedules**: Extreme points of $S$
- **Pseudostable schedules**: Local extreme points of $S$
- **Quasiactive schedules**: Minimal points of relation polyhedra
- **Quasistable schedules**: Vertices of relation polyhedra

![Diagram illustrating schedule types]
Classes of objective functions and efficient solutions

- **Regular** functions $\rightarrow$ project duration $S_{n+1}$
- **Linear(-izable)** functions $f(S)$
  $\rightarrow$ subset makespan $\max_{i \in U} (S_i + p_i) - \min_{i \in U} S_i$
- **Binary monotonic** functions: monotonicity in binary directions $s \in \{0, 1\}^n$
  $\rightarrow$ net present value $\sum_{i \in V} c_i F^{S_i+p_i}$
- **Locally regular** functions: $f$ regular on iso-order sets $S_T$
  $\rightarrow$ total resource availability cost $\sum_{k \in R} c_k \max_{t \geq 0} r_k(S, t)$
- **Locally concave** functions: $f$ concave on iso-order sets $S_T$
  $\rightarrow$ total squared utilization cost $\sum_{k \in R} c_k \int_0^\infty r_k^2(S, t) dt$

<table>
<thead>
<tr>
<th>Objective function</th>
<th>Efficient solutions</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>Active schedules</td>
<td>$\mathcal{NP}$-complete</td>
</tr>
<tr>
<td>Linear</td>
<td>Stable schedules</td>
<td>$\mathcal{NP}$-complete</td>
</tr>
<tr>
<td>Binary monotonic</td>
<td>Pseudostable schedules</td>
<td>$\mathcal{NP}$-complete</td>
</tr>
<tr>
<td>Locally regular</td>
<td>Quasiactive schedules</td>
<td>polynomial</td>
</tr>
<tr>
<td>Locally concave</td>
<td>Quasistable schedules</td>
<td>polynomial</td>
</tr>
</tbody>
</table>

Christoph Schwindt
Clausthal University of Technology

Project Scheduling Under Generalized Precedence Relations
Generic solution approaches

Regular, linear, and binary-monotonic objective functions

- Time-constrained scheduling problem \((P_T)\) efficiently solvable
  - longest-walk calculations
  - linear programming
  - recursive algorithms, e.g., De Reyck and Herroelen (1998b)
  - steepest-descent algorithms, e.g., S. and Zimmermann (2001)

- Apply relaxation-based procedure providing feasible relation \(\rho\)

- Minimize \(f\) on relation polyhedron \(S_T(\rho)\)

Objective function (only) locally regular or locally concave

- Time-constrained scheduling problem \((P_T)\) intractable

- Apply schedule-construction procedure providing minimal or extreme point of some relation polyhedron \(S_T(\rho)\)
Relaxation-based procedure

Schedule-generation scheme for problem \((P)\)

1. Set \(\rho := \emptyset\);
2. If \(S_T(\rho) = \emptyset\): STOP; // no feasible schedule found
3. Verify feasibility of \(\rho\) by solving maximum-cut problems;
4. If \(\rho\) is feasible: compute minimizer of \(f\) on \(S_T(\rho)\) and STOP;
5. Determine resource \(k\) such that antichain \(A_k(\rho)\) is forbidden;
6. Select \(\subseteq\)-minimal set \(B \subset A_k(\rho)\) such that \(A := A_k(\rho) \setminus B\) is not forbidden, and select some \(i \in A\);
7. Set \(\rho := \rho \cup (\{i\} \times B)\), and go to step 2;

- Combination \((i, B)\) is called a minimal delaying mode (De Reyck and Herroelen 1998a)
- Procedure can also be used for problems with stochastic processing times \(\tilde{p}_i\); resulting relation defines an ES-policy (Radermacher 1981)
Example

Relaxation-based procedure

(a) \( \rho = \emptyset \)

(b) \( \rho = \{(3, 2)\} \)

(c) \( \rho = \{(3, 2), (4, 2)\} \)

(d) \( \rho = \{(3, 2), (4, 2), (5, 1), (5, 2)\} \)
Schedule-construction procedure

Schedule-generation scheme for problem \((P_T)\)

1. Set \(C := \{0\}, \ S_0 := 0,\ \text{and} \ ES_i := d_{0i}, \ LS_i := -d_{i0} \ \text{for all} \ i \in V;\)
2. Select some \(i \in C\) and some \(j \in V \setminus C;\)
3. Select time \(S_j \in \{S_i + \delta_{ij}, S_i + p_i, S_i - p_j, S_i - \delta_{ji}\} \cap [ES_j, LS_j];\)
4. Add \(j\) to \(C;\)
5. Update \(ES_h\) and \(LS_h\) for all \(h \in V \setminus C;\)
6. If \(C \neq V,\) go to step 2;

- **Locally regular objective function:** select \(t \in \{S_i + \delta_{ij}, S_i + p_i\}\)
- Pairs \((i, j)\) selected in step 2 form a spanning tree (spanning outtree) of relation network \(N(\rho)\) rooted at node 0
- In case of resource constraints: determine \(\subseteq\)-minimal feasible relation \(\rho\) and apply procedure on network \(N(\rho)\) instead of \(N\)
### Example

#### Schedule-construction procedure for quasiactive schedule

<table>
<thead>
<tr>
<th>Iteration</th>
<th></th>
<th></th>
<th>time t</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration</th>
<th></th>
<th></th>
<th>time t</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>2</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>6</td>
<td>16</td>
</tr>
</tbody>
</table>

![Graph showing schedule construction](image)

Legend:
- \( p_i \)  
- \( \delta_{ij} \)  
- \( p_j \)
Solution approaches for the project duration problem

- Minimization of project duration has received largest attention in literature

- Four categories of solutions approaches
  - Adaptations of schedule-construction procedures and metaheuristics for RCPSP
  - Relaxation-based branch-and-bound procedures
  - Constraint-programming based approaches
  - Mixed-integer linear programming formulations and related algorithms
Schedule-construction procedures

- Serial/parallel SGS iteratively fix start times of activities
- When procedure is trapped in deadlock: call unscheduling procedure
- No guarantee to find a feasible solution, but very effective on benchmark instances

Unscheduling procedure (Franck et al. 2001)

1. Set $\Delta := t - LS_j$; // $t$ is earliest resource-feasible start time of $j$
2. Determine $U := \{i \in C \mid LS_j = S_i - d_{ji}\}$;
3. For all $i \in U$: set $ES_i := ES_i + \Delta$;
4. Remove all $h$ with $S_h \geq \min_{i \in U} S_i$ from set $C$;
5. Update earliest and latest start times and return to schedule-generation scheme;

Priority-rule based methods, tabu search, and genetic algorithm by Franck et al. (2001) and evolutionary algorithm by Ballestín et al. (2011) based on serial SGS with unscheduling
Example

- $j = 2$, $t = 7$, $LS_j = 3$, $\Delta = 4$
- $\mathcal{U} = \{i \in \mathcal{C} \mid LS_j = S_i - d_{ji}\} = \{1\}$
- Set $ES_1 := 4$ and unschedule activities $i = 1, 3, 4$

(a) $i = 3, 4, 1$

(b) $i = 2$

(c) $i = 4$
Relaxation-based approaches

- Start with solution $\hat{S} = ES$ to time-constrained problem ($P_T$)
- Identify some time $t$ with $r_k(\hat{S}, t) > R_k$ for some resource $k$
- Branch over alternatives to resolve the resource conflict at time $t$
- Partition forbidden set $\mathcal{A}(\hat{S}, t)$ in minimal delaying alternative $B$ and feasible set $A = \mathcal{A}(\hat{S}, t) \setminus B$

- **Ordinary precedence relations** (De Reyck and Herroelen 1998a)
  \[ S_j \geq S_i + p_i \quad (j \in B) \quad \text{for some } i \in A \]

- **Release dates** (Fest et al. 1999)
  \[ S_j \geq \delta_{0j} := \min_{i \in A}(\hat{S}_i + p_i) \quad (j \in B) \]

- **Disjunctive precedence relations** (S. 1998)
  \[ S_j \geq \min_{i \in A}(S_i + p_i) \quad (j \in B) \]
Constraint-programming approaches
(Dorndorf et al. 2000, Schutt et al. 2013)

- Associate decision variables $S_i$ with domains $\Delta_i = \{ES_i, \ldots, LS_i\}$
- Try to reduce domain sizes by applying consistency tests like precedence, interval capacity, or disjunctive consistency tests
- When consistency tests reach fixed point, perform dichotomic start-time branching for activity $i$ with smallest earliest start time $t = \min \Delta_i$: $S_i = t \lor S_i \geq t + 1$
- Replace domain $\Delta_i$ by $\{t\}$ or $\{t + 1, \ldots, LS_i\}$
- Propagate update to other domains by applying consistency tests

Best results for project duration problem obtained by Schutt et al. (2013) from combining start-time branching with SAT representation and lazy clause generation

Alternative approach by Cesta et al. (2002) based on formulation as CSP for posting precedence relations in minimal forbidden sets
Mixed-integer linear programming (Bianco and Caramia 2012)

- In general, MILP formulation for RCPSP easily adapted to generalized precedence relations
- MILP model of Bianco and Caramia (2012)
  - Binary variables $s_{it} = 1$ if $i$ has been started by time $t$
  - Binary variables $f_{it} = 1$ if $i$ has been completed by time $t$
  - Variables $z_{it} \in [0, 1]$ keeping execution percentage of $i$ by time $t$
  - Coupling constraints: $z_{i(t+1)} - z_{it} = \frac{1}{p_i} (s_{it} - f_{it})$
  - Temporal constraints: $\sum_{t=1}^{T} s_{it} \geq \sum_{t=1}^{T} s_{jt} + \delta_{ij}$
  - Resource constraints: $\sum_{i \in V} r_{ik} p_i \cdot (z_{it} - z_{i(t-1)}) \leq R_k$

- Branch-and-bound algorithm based on MILP formulation
  - Each level of enumeration tree associated with one activity $i$
  - Branch over $\bigvee_{t=ES_i,...,LS_i} \{s_{it} = 1\}$
  - Lower bounds obtained by Lagrangian relaxation of resource constraints
Experimental performance analysis for project duration problem

- Algorithms evaluated on ProGen/max data sets
- Results for test set CD (540 instances, $n = 100$, $|\mathcal{R}| = 5$)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$t_{cpu}$</th>
<th>$p_{feas}$</th>
<th>$p_{opt}$</th>
<th>$p_{inf}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>De Reyck and Herroelen (1998a)</td>
<td>3</td>
<td>97.3</td>
<td>54.8</td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>97.5</td>
<td>56.4</td>
<td>1.4</td>
</tr>
<tr>
<td>S. (1998)</td>
<td>3</td>
<td>98.1</td>
<td>58.0</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>98.1</td>
<td>62.5</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>98.1</td>
<td>63.4</td>
<td>1.9</td>
</tr>
<tr>
<td>Fest et al. (1999)</td>
<td>3</td>
<td>92.2</td>
<td>58.1</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>98.1</td>
<td>69.4</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>98.1</td>
<td>71.1</td>
<td>1.9</td>
</tr>
<tr>
<td>Dorndorf et al. (2000)</td>
<td>3</td>
<td>97.8</td>
<td>66.2</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>98.1</td>
<td>70.4</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>98.1</td>
<td>71.1</td>
<td>1.9</td>
</tr>
<tr>
<td>Bianco and Caramia (2012)</td>
<td>3</td>
<td>98.1</td>
<td>67.6</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>98.1</td>
<td>71.8</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>98.1</td>
<td>72.2</td>
<td>1.9</td>
</tr>
<tr>
<td>Schutt et al. (2013)</td>
<td>1</td>
<td>97.9</td>
<td>78.1</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>98.1</td>
<td>89.8</td>
<td>1.9</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>98.1</td>
<td>94.0</td>
<td>1.9</td>
</tr>
</tbody>
</table>

The multi-mode version of problem \((P)\)

- Each activity \(i\) can be executed in one of a finite number of execution modes \(m \in M_i\)
- Executions modes \(m\) differ in durations \(p_{im}\) and resource requirements \(r_{ikm}\) (renewable and nonrenewable resources)
- Generalized precedence relations \(\delta_{ij}\) depend on modes \(m_i\) and \(m_j\)

Feasibility variant of time-constrained problem \((P_T)\) \(\mathcal{NP}\)-complete
Preemptive problem ($\bar{P}$) (S. and Paetz 2014)

- Activities can be interrupted at any point in time
- Generalization of problem ($P$) since preemption can be prevented by generalized feeding precedence relations of type ($i, i, 1.0, 0.0, -p_i$)

($\bar{P}$) can be reduced to canonical form with nonpositive completion-to-start time lags

- Up to $2n - 1$ slices needed, one and the same antichain can be in progress several times, number of interruptions bounded by $n(n - 1)$
- Subproblem with given positive antichains still $\mathcal{NP}$-hard
Practical applications including generalized precedence relations

- Technical constraints in civil engineering (Bartusch et al. 1988)
- Lot streaming in manufacturing (Neumann and S. 1997)
- Perishable intermediate products in process scheduling (Neumann et al. 2002)
- Minimum and maximum durations of service activities (Mellentien et al. 2004)
- Minimum and maximum time lags between build-up and test activities in automotive R&D projects (Bartels and Zimmermann 2009)
- Overlapping of activities in aggregate production scheduling (Alfieri et al. 2011)
- Maximum duration of validity for statutory permissions in nuclear power plant dismantling (Bartels et al. 2011)
- Maximum makespan for activity sequences at service centers (Quintanilla et al. 2012)
Conclusions

- Generalized precedence relations needed to formulate real-life scheduling constraints
- Efficient temporal analysis based on Roy’s Metra Potential Method
- Feasibility variant of resource-constrained problems \( \mathcal{NP} \)-complete
- Classical schedule-generation schemes lead to deadlocks
- Unscheduling techniques, relaxation-based approaches, constraint programming methods, mixed-integer programming formulations

Significant recent advances, e.g.:
- Linear-time algorithm for checking feasibility of temporal constraints
- Very effective constraint-programming approaches for project duration problem

Avenues for future research
- Preemptive project scheduling under gpr’s
- Stochastic/robust project scheduling under gpr’s
- Lazy clause generation approach for different objective functions
References


References

Scheduling project networks with resource constraints and time windows.

A new formulation of the resource-unconstrained project scheduling problem with
generalized precedence relations to minimize the completion time.
*Networks*, 56:263–271.

An exact algorithm to minimize the makespan in project scheduling with scarce
resources and generalized precedence relations.

A constraint-based method for project scheduling with time windows.
References

A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations.

An optimal procedure for the resource-constrained project scheduling problem with discounted cash flows and generalized precedence relations.

The multi-mode resource-constrained project scheduling problem with generalized precedence relations.

References

Resource-constrained project scheduling with time windows: A branching scheme based on dynamic release dates.

Truncated branch-and-bound, schedule-construction, and schedule-improvement procedures for resource-constrained project scheduling.
OR Spektrum, 23:297–324.

Robust local search for solving RCPSP/max with durational uncertainty.
Journal of Artificial Intelligence Research, 43:43–86.

Generalized network functions in flow networks.
References

Critical path planning and scheduling: Mathematical basis.

A branch-and-cut algorithm for scheduling of projects with variable-intensity activities.

Scheduling the factory pick-up of new cars.

Active and stable project scheduling.
References


References

Cost-dependent essential systems of es-strategies for stochastic scheduling problems.

Physionomie et traitement des problèmes d’ordonnancement.
In Carré, D., Darnaut, P., Guitard, P., Nghiem, P., Pacaud, P., de Rosinski, J.,
Dunod, Paris.

A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags.

Solving RCPSP/max by lazy clause generation.
References

Verfahren zur Lösung des ressourcenbeschränkten Projektdauerminimierungsproblems mit planungsabhängigen Zeitfenstern.
Shaker, Aachen.

Resource Allocation in Project Management.

Continuous preemption problems.

A steepest ascent approach to maximizing the net present value of projects.