Project Scheduling Under Generalized Precedence Relations

A Survey of Structural Issues, Solution Approaches, and Applications

Christoph Schwindt

Operations Management Group

TU Clausthal
Clausthal University of Technology

14th International Conference on
Project Management and Scheduling, Munich

Outline

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

$$y_i : \mathbb{R}_{\geq 0} \to \{0,1\} \text{ such that } \int_0^\infty y_i(t)\,dt = p_i$$

- 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

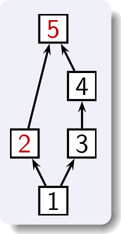
$$z_i : \mathbb{R}_{\geq 0} \to [0,1], \quad t \mapsto z_i(t) = \tfrac{1}{p_i} \int_0^t y_i(t)\,dt$$

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 δ_{ij} or execution percentages ξ_i and ξ_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)

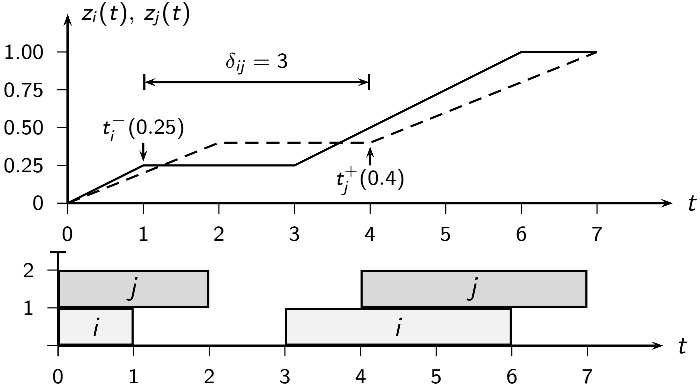
   $$(i,j,\xi_i,\xi_j) : \max\{t \mid z_j(t) = \xi_j\} \geq \min\{t \mid z_i(t) = \xi_i\}$$

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

   $$(i,j,\xi_i,\xi_j,\delta_{ij}) : \underbrace{\max\{t \mid z_j(t) = \xi_j\}}_{t_j^+(\xi_j)} \geq \underbrace{\min\{t \mid z_i(t) = \xi_i\}}_{t_i^-(\xi_i)} + \delta_{ij}$$

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) $f(y) = \sum_{k \in \mathcal{R}} c_k \int_0^\infty \left(\sum_{i \in V} r_{ik} y_i(t) \right)^2 dt$

Resource-constrained project scheduling problem

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

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

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

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

Set of feasible schedules: $\mathcal{S}$, set of time-feasible schedules: $\mathcal{S}_T$

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, δ_{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

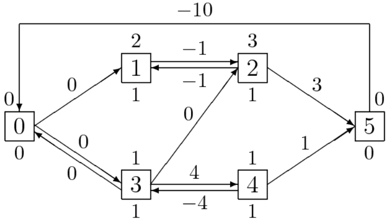
$$S_j \geq S_i + \delta_{ij} \Leftrightarrow S_i \leq S_j - \delta_{ij}$$

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 ℓ_{ij} of i and j: $\delta_{ij} = \ell_{ij} - p_i$, $\delta_{ji} = \ell_{ij} - p_j$
- Maximum makespan C^U_{max} for activity set U: $\delta_{ij} = -C^U_{max} + p_i$ 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 $S : V \to \mathbb{R}_{\geq 0}$
- Tensions: differences $S_j - S_i$ of potentials
- Generalized precedence relations: lower bounds δ_{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

Temporal analysis with MPM

Fundamentals

- $\mathcal{S}_T \neq \emptyset$ iff N does not contain any cycle of positive length
- Induced time lag $d_{ij} := \min_{S \in \mathcal{S}_T}(S_j - S_i) =$ 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 $m := |E|$)

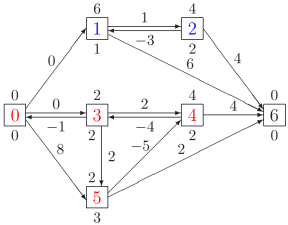
- $\mathcal{S}_T \neq \emptyset$ and single time lag d_{ij}: transformation of Bianco and Caramia (2010) to unit-capacity transshipment problem, $\mathcal{O}(m)$
- All time lags (distance matrix $D = (d_{ij})_{i,j \in V}$): Floyd-Warshall-Algorithm, $\mathcal{O}(n^3)$
- Update of distance matrix after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

Resource-constrained problem (P): Complexity and decomposition

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

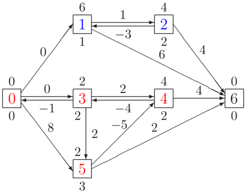
Decomposition theorem (Neumann and Zhan 1995)

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.

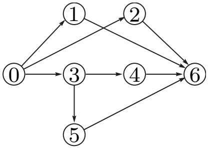


- Classical schedule-generation schemes must be modified to avoid or to cope with deadlocks
- Decomposition theorem is basis of heuristic decomposition methods

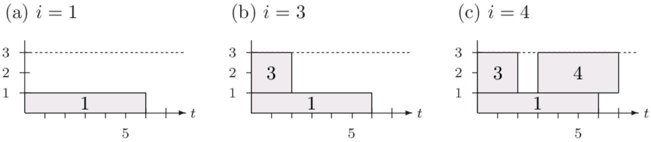
Example



Precedence graph



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)

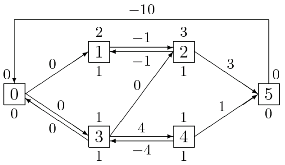
1. An $\subseteq$-minimal forbidden set F is broken up by schedule S iff F contains two activities i, j with $S_j \geq S_i + p_i$.
2. Schedule S is resource-feasible iff all $\subseteq$-minimal forbidden sets F are broken up.

Consequences:

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

$$\mathcal{S}_T(\rho) = \{S \in \mathcal{S}_T \mid S_j \geq S_i + p_i \text{ for all } (i,j) \in \rho\}$$

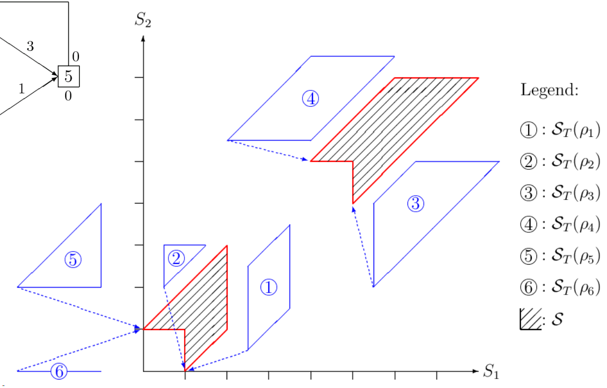
Covering of $\mathcal{S}$ by relation polyhedra



Capacity $R = 2$

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

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



Feasible relations (S. 2005)

Definition: Feasible relation

Relation ρ with $\emptyset \neq \mathcal{S}_T(\rho) \subseteq \mathcal{S}$ is called feasible relation.

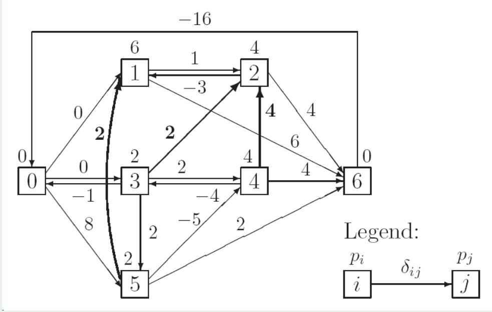
- Condition $\mathcal{S}_T(\rho) \neq \emptyset$: ordinary precedence relations $(i, j) \in \rho$ are compatible with generalized precedence relations $(i', j') \in E$
- Condition $\mathcal{S}_T(\rho) \subseteq \mathcal{S}$: all schedules S satisfying the ordinary precedence relations $(i, j) \in \rho$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

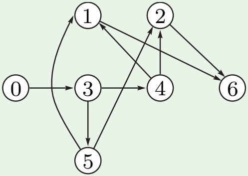
Induced strict order, schedule-induced order, iso-order set

- 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 ρ 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 $\mathcal{S}_T^{=}(\theta) := \{S \in \mathcal{S}_T \mid \theta(S) = \theta\}$

Example

Relation network for $\rho = \{(3,2),(4,2),(5,1)\}$ and strict order $\Theta(\rho)$

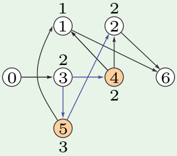


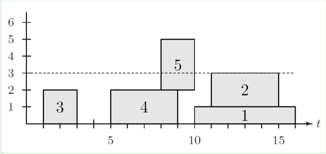


Checking feasibility of relations (Kaerkes and Leipholz 1977)

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

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

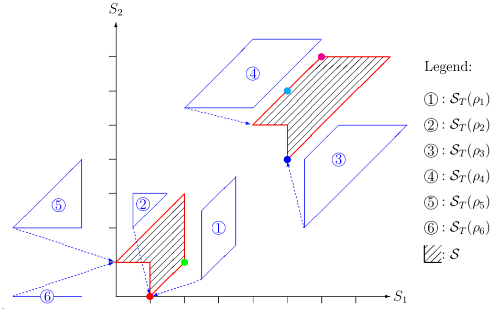




- Weight nodes with requirements r_{ik}
- Determine maximum $0-(n+1)$-cut
- Here: $\mathcal{A}(\rho) = \{4,5\}$

Schedule types (Neumann et al. 2000)

- Active schedules: Minimal points of $\mathcal{S}$
- Stable schedules: Extreme points of $\mathcal{S}$
- Pseudostable schedules: Local extreme points of $\mathcal{S}$
- Quasiactive schedules: Minimal points of relation polyhedra
- Quasistable schedules: Vertices of relation polyhedra



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 $\sum_{i \in V} c_i^F \beta^{S_i + p_i}$
- Locally regular functions: f regular on iso-order sets $\mathcal{S}_T^{=}$
  $\rightarrow$ total resource availability cost $\sum_{k \in \mathcal{R}} c_k \max_{t \geq 0} r_k(S,t)$
- Locally concave functions: f concave on iso-order sets $\mathcal{S}_T^{=}$
  $\rightarrow$ total squared utilization cost $\sum_{k \in \mathcal{R}} c_k \int_0^{\infty} r_k^2(S,t)\, dt$

Objective function	Efficient solutions	Verification
Regular	Active schedules	$\mathcal{NP}$-complete
Linear	Stable schedules	$\mathcal{NP}$-complete
Binary monotonic	Pseudostable schedules	$\mathcal{NP}$-complete
Locally regular	Quasiactive schedules	polynomial
Locally concave	Quasistable schedules	polynomial

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 ρ
- Minimize f on relation polyhedron $\mathcal{S}_T(\rho)$

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 $\mathcal{S}_T(\rho)$

Relaxation-based procedure

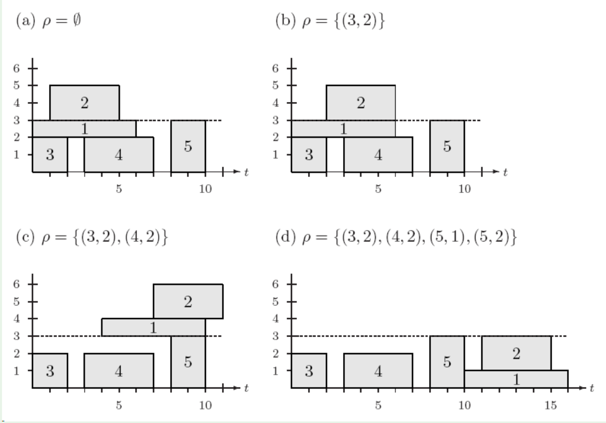
Schedule-generation scheme for problem (P)

1. Set $\rho := \emptyset$;
2. If $\mathcal{S}_T(\rho) = \emptyset$: STOP; // no feasible schedule found
3. Verify feasibility of ρ by solving maximum-cut problems;
4. If ρ is feasible: compute minimizer of f on $\mathcal{S}_T(\rho)$ and STOP;
5. Determine resource k such that antichain $\mathcal{A}_k(\rho)$ is forbidden;
6. Select $\subseteq$-minimal set $B \subset \mathcal{A}_k(\rho)$ such that $A := \mathcal{A}_k(\rho) \setminus B$ 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 $\widetilde{p}_i$; resulting relation defines an *ES*-policy (Radermacher 1981)

Example

Relaxation-based procedure



Schedule-construction procedure

Schedule-generation scheme for problem (P_T)

1. Set $\mathcal{C} := \{0\}$, $S_0 := 0$, and $ES_i := d_{0i}$, $LS_i := -d_{i0}$ for all $i \in V$;
2. Select some $i \in \mathcal{C}$ and some $j \in V \setminus \mathcal{C}$;
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 $\mathcal{C}$;
5. Update ES_h and LS_h for all $h \in V \setminus \mathcal{C}$;
6. If $\mathcal{C} \neq V$, go to step 2;

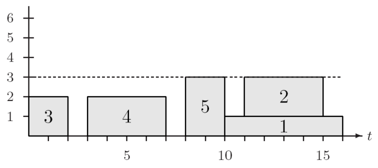
- Locally regular objective function: select $t \in \{S_i + \delta_{ij}, S_i + p_i\}$
- Pairs (i, j) selected in step 2 form a spanning tree (spanning outtree) of relation network $N(\rho)$ rooted at node 0
- In case of resource constraints: determine $\subseteq$-minimal feasible relation ρ and apply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

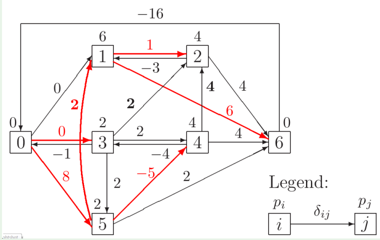
Example

Schedule-construction procedure for quasiactive schedule

Iteration	i	j	time t
1	0	3	0
2	0	5	8
3	5	4	3

Iteration	i	j	time t
4	5	1	10
5	1	2	11
6	1	6	16





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

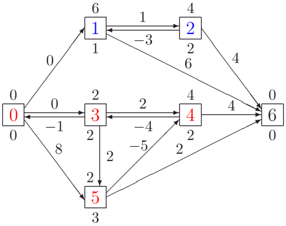
- Serial/parallel SGS iteratively fix start times of activities
- When procedure is trapped in deadlock: call unscheduling procedure
- No guarantee to find a feasible solution, but very effective on benchmark instances

Unscheduling procedure (Franck et al. 2001)

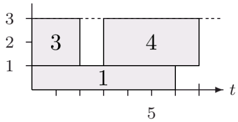
1. Set $\Delta := t - LS_j$; // t is earliest resource-feasible start // time of j
2. Determine $\mathcal{U} := \{i \in \mathcal{C} \mid LS_j = S_i - d_{ji}\}$;
3. For all $i \in \mathcal{U}$: set $ES_i := ES_i + \Delta$;
4. Remove all h with $S_h \geq \min_{i \in \mathcal{U}} S_i$ from set $\mathcal{C}$;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

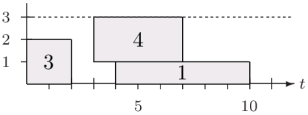


(c) $i = 4$

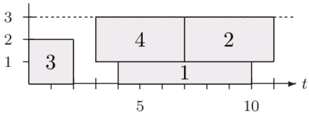


- $j = 2,\ t = 7,\ LS_j = 3,\ \Delta = 4$
- $\mathcal{U} = \{i \in \mathcal{C} \mid LS_j = S_i - d_{ji}\} = \{1\}$
- Set $ES_1 := 4$ and unschedule activities $i = 1, 3, 4$

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



(b) $i = 2$



Relaxation-based approaches

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

  - Ordinary precedence relations (De Reyck and Herroelen 1998a)

    $$S_j \geq S_i + p_i \quad (j \in B) \qquad \text{for some } i \in A$$

  - Release dates (Fest et al. 1999)

    $$S_j \geq \delta_{0j} := \min_{i \in A}(\widehat{S}_i + p_i) \quad (j \in B)$$

  - 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 $t = \min \Delta_i$: $S_i = t \vee S_i \geq t + 1$
- Replace domain Δ_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 $\vee_{t=ES_i,\ldots,LS_i}\{s_{it} = 1\}$
  - Lower bounds obtained by Lagrangian relaxation of resource constraints

Experimental performance analysis for project duration problem

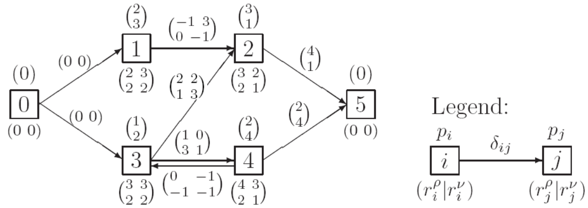
- Algorithms evaluated on ProGen/max data sets[1]
- Results for test set CD (540 instances, $n = 100$, $|\mathcal{R}| = 5$)

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

[1] http://www.wiwi.tu-clausthal.de/en/abteilungen/produktion/forschung/schwerpunkte/project-generator/

The multi-mode version of problem (P)

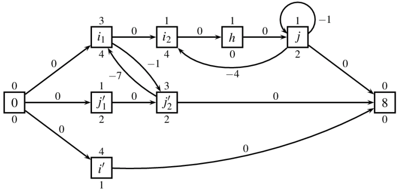
- Each activity i can be executed in one of a finite number of execution modes $m \in \mathcal{M}_i$
- Executions modes m differ in durations p_{im} and resource requirements r_{ikm} (renewable and nonrenewable resources)
- Generalized precedence relations δ_{ij} depend on modes m_i and m_j

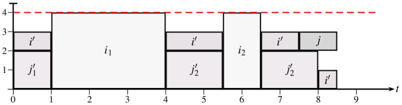


- Feasibility variant of time-constrained problem (P_T) $\mathcal{NP}$-complete
- Relaxation-based branch-and-bound algorithms by De Reyck and Herroelen (1999) and Heilmann (2003), MILP model by Sabzehparvar and Seyed-Hosseini (2008)

Preemptive problem ($\overline{P}$) (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 $(i, i, 1.0, 0.0, -p_i)$
- $(\overline{P})$ 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

Alfieria, A., Toliob, T., and Urgo, M. (2011).
A project scheduling approach to production planning with feeding precedence relations.
International Journal of Production Research, 49:995–1020.

Ballestín, F., Barríos, A., and Valls, V. (2011).
An evolutionary algorithm for the resource-constrained project scheduling problem with minimum and maximum time lags.
Journal of Scheduling, 14:391–406.

Bartels, J. H., Gather, T., and Zimmermann, J. (2011).
Dismantling of nuclear power plants at optimal npv.
Annals of Operations Research, 186:407–427.

Bartels, J. H. and Zimmermann, J. (2009).
Scheduling tests in automotive R&D projects.
European Journal of Operational Research, 193:805–819.

References

Bartusch, M., Möhring, R. H., and Radermacher, F.-J. (1988).
Scheduling project networks with resource constraints and time windows.
Annals of Operations Research, 16:201–240.

Bianco, L. and Caramia, M. (2010).
A new formulation of the resource-unconstrained project scheduling problem with generalized precedence relations to minimize the completion time.
Networks, 56:263–271.

Bianco, L. and Caramia, M. (2012).
An exact algorithm to minimize the makespan in project scheduling with scarce resources and generalized precedence relations.
European Journal of Operational Research, 219:73–85.

Cesta, A., Oddi, A., and Smith, S. F. (2002).
A constraint-based method for project scheduling with time windows.
Journal of Heuristics, 8:109–136.

References

De Reyck, B. and Herroelen, W. S. (1998a).
A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations.
European Journal of Operational Research, 111:152–174.

De Reyck, B. and Herroelen, W. S. (1998b).
An optimal procedure for the resource-constrained project scheduling problem with discounted cash flows and generalized precedence relations.
Computers and Operations Research, 25:1–17.

De Reyck, B. and Herroelen, W. S. (1999).
The multi-mode resource-constrained project scheduling problem with generalized precedence relations.
European Journal of Operational Research, 119:538–556.

Dorndorf, U., Pesch, E., and Phan-Huy, T. (2000).
A branch-and-bound algorithm for the resource-constrained project scheduling problem.
Mathematical Methods of Operations Research, 52:413–439.

References

Fest, A., Möhring, R. H., Stork, F., and Uetz, M. (1999).
Resource-constrained project scheduling with time windows: A branching scheme based on dynamic release dates.
Technical Report 596, Technical University of Berlin.

Franck, B., Neumann, K., and Schwindt, C. (2001).
Truncated branch-and-bound, schedule-construction, and schedule-improvement procedures for resource-constrained project scheduling.
OR Spektrum, 23:297–324.

Fu, N., Lau, H. C., Varakantham, P., and Xiao, F. (2012).
Robust local search for solving RCPSP/max with durational uncertainty.
Journal of Artifcial Intelligence Research, 43:43–86.

Kaerkes, R. and Leipholz, B. (1977).
Generalized network functions in flow networks.
Operations Research Verfahren, 27:225–273.

References

Kelley, J. E. (1961).
Critical path planning and scheduling: Mathematical basis.
Operations Research, 9:296–320.

Kis, T. (2005).
A branch-and-cut algorithm for scheduling of projects with variable-intensity activities.
Mathematical Programming, 103:515–539.

Mellentien, C., Schwindt, C., and Trautmann, N. (2004).
Scheduling the factory pick-up of new cars.
OR Spectrum, 26:579–601.

Neumann, K., Nübel, H., and Schwindt, C. (2000).
Active and stable project scheduling.
Mathematical Methods of Operations Research, 52:441–465.

References

Neumann, K. and Schwindt, C. (1997).
Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production.
OR Spektrum, 19:205–217.

Neumann, K., Schwindt, C., and Trautmann, N. (2002).
Advanced production scheduling for batch plants in process industries.
OR Spectrum, 24:251–279.

Neumann, K. and Zhan, J. (1995).
Heuristics for the minimum project-duration problem with minimal and maximal time lags under fixed resource constraints.
Journal of Intelligent Manufacturing, 6:145–154.

Quintanilla, S., Pérez, A., Lino, P., and Valls, V. (2012).
Time and work generalised precedence relationships in project scheduling with pre-emption: An application to the management of service centres.
European Journal of Operational Research, 219:59–72.

References

Radermacher, F. J. (1981).
Cost-dependent essential systems of es-strategies for stochastic scheduling problems.
Methods of Operations Research, 42:17–31.

Roy, B. (1964).
Physionomie et traitement des problèmes d'ordonnancement.
In Carré, D., Darnaut, P., Guitard, P., Nghiem, P., Pacaud, P., de Rosinski, J., Roy, B., and Sandier, G., editors, *Les Problèmes d'Ordonnancement*, pages 1–18. Dunod, Paris.

Sabzehparvar, M. and Seyed-Hosseini, S. M. (2008).
A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags.
Journal of Supercomputing, 44:257–273.

Schutt, A., Feydy, T., Stuckey, P. J., and Wallace, M. G. (2013).
Solving RCPSP/max by lazy clause generation.
Journal of Scheduling, 16:273–289.

References

Schwindt, C. (1998).
Verfahren zur Lösung des ressourcenbeschränkten Projektdauerminimierungsproblems mit planungsabhängigen Zeitfenstern.
Shaker, Aachen.

Schwindt, C. (2005).
Resource Allocation in Project Management.
Springer, Berlin.

Schwindt, C. and Paetz, T. (2014).
Continuous preemption problems.
In Schwindt, C. and Zimmermann, J., editors, *Handbook on Project Management and Scheduling: Vol. I*, Interational Handbook on Information Systems, Berlin. Springer.

Schwindt, C. and Zimmermann, J. (2001).
A steepest ascent approach to maximizing the net present value of projects.
Mathematical Methods of Operations Research, 53:435–450.