Temporal Scheduling of Projects with Time-Overlap Trade-Offs

Christoph Schwindt
Philipp Benke

Clausthal University of Technology

New Perspectives in Scheduling Theory
Hangzhou, October 9–13, 2011

Outline

1. Problem definition
   - Concurrent engineering projects
   - Temporal constraints
   - Temporal scheduling problems
2. Structural issues
3. Temporal scheduling methods
   - Earliest completion times
   - Latest completion times
4. Performance analysis
5. Conclusions

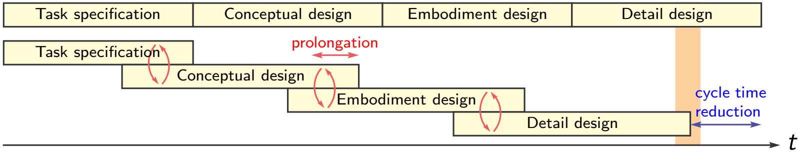
Outline

1. Problem definition
   - Concurrent engineering projects
   - Temporal constraints
   - Temporal scheduling problems
2. Structural issues
3. Temporal scheduling methods
   - Earliest completion times
   - Latest completion times
4. Performance analysis
5. Conclusions

Concurrent engineering projects

Concurrent engineering approach

- Industrial development projects organized in consecutive phases
- In high-tech sectors ability to place new products within tight market entry time windows constitutes decisive success factor
- Concurrent engineering approach: parallelize consecutive development phases to shorten cycle time of development project
- Additional integration and coordination efforts due to feedback loops between phases: trade-off between overlappings and durations (time-overlap trade-off)



Concurrent engineering projects

Overlap times and activity durations

- Development project consists of n activities $i \in V$ (phases, working packages, milestones, project beginning, project end)
- Precedence relationships $(i, j) \in E$ between activities, maximum project duration $\overline{d}$
- Overlapping of activities i, j with $(i, j) \in E$ during time

  $$\ell_{ij} = C_i - S_j$$

  leads to increasing duration of activity j

  $$p_j = p_j(\ell_j) \quad \text{with } \ell_j = (\ell_{ij})_{(i,j)\in E}$$

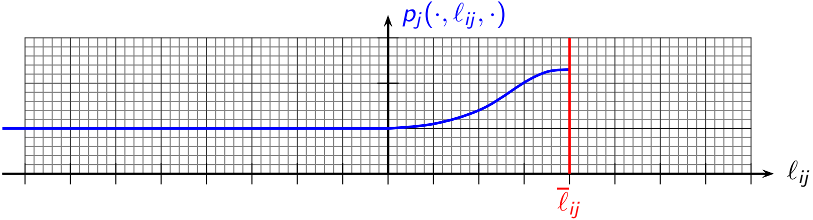
- Overlap times ℓ_{ij} bounded by $\overline{\ell}_{ij}$

Concurrent engineering projects

Properties of duration functions $p_j(\ell_j)$

- p_j componentwise differentiable
- p_j componentwise constant when overlapping is avoided,
  p_j componentwise nondecreasing when overlapping is realized

$$\frac{\partial p_j}{\partial \ell_{ij}}(\ell_{ij}) = 0 \text{ for } \ell_{ij} < 0, \quad \frac{\partial p_j}{\partial \ell_{ij}}(\ell_{ij}) \geq 0 \text{ for } \ell_{ij} \in [0, \overline{\ell}_{ij}]$$

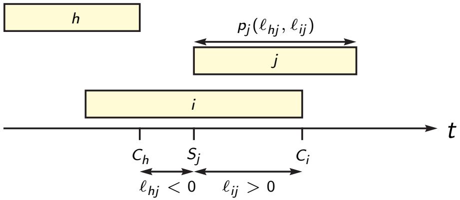


Temporal constraints

Maximum start-to-end time lags

- Negative overlap time ℓ_{ij} means that j is started $-\ell_{ij}$ time units after completion of j
- Maximum overlap time $\overline{\ell}_{ij}$ induces maximum start-to-end time lag between activity j and activity i

$$\ell_{ij} = C_i - S_j \leq \overline{\ell}_{ij} \quad \Leftrightarrow \quad C_i \leq S_j + \overline{\ell}_{ij}$$



Temporal constraints

Semantic power: How to model . . .

- project deadline: maximum start-to-end time lag between project beginning $j = 0$ and project end $i = n + 1$

$$C_{n+1} \leq S_0 + \overline{d} \quad \Leftrightarrow \quad \overline{\ell}_{(n+1)0} = \overline{d}$$

- ordinary precedence constraint between i and j

$$S_j \geq C_i \quad \Leftrightarrow \quad \overline{\ell}_{ij} = 0$$

- minimum end-to-start time lag $d_{ij}^{min} > 0$ between i and j

$$S_j \geq C_i + d_{ij}^{min} \quad \Leftrightarrow \quad \overline{\ell}_{ij} = -d_{ij}^{min}$$

Semantic limitations

- Due to overlap-dependent activity durations only modeling of maximum start-to-end and minimum end-to-start relationships

Temporal scheduling problems

Earliest and latest start and completion times

- Determine earliest and latest start and completion times ES_h, LS_h, EC_h, LC_h of activities $h \in V$
- Earliest/latest start time problem for activity h

$$(TSP_h^S) \left\{ \begin{array}{lll} \text{Min./Max. } S_h & & \\ \text{subject to } & \ell_{ij} = S_i + p_i(\ell_i) - S_j & ((i,j) \in E) \\ & \ell_{ij} \leq \overline{\ell}_{ij} & ((i,j) \in E) \\ & S_0 = 0,\ S_j \geq 0 & (j \in V) \end{array} \right. \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \mathcal{S}_T$$

- Earliest/latest completion time problem for activity h

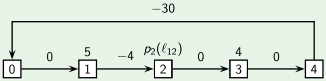
$$(TSP_h^C) \left\{ \begin{array}{lll} \text{Min./Max. } C_h & & \\ \text{subject to } & \ell_{ij} = C_i - [C_j - p_j(\ell_j)] & ((i,j) \in E) \\ & \ell_{ij} \leq \overline{\ell}_{ij} & ((i,j) \in E) \\ & C_0 = 0,\ C_j \geq p_j(\ell_j) & (j \in V) \end{array} \right. \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \mathcal{C}_T$$

Outline

1. Problem definition
   - Concurrent engineering projects
   - Temporal constraints
   - Temporal scheduling problems
2. Structural issues
3. Temporal scheduling methods
   - Earliest completion times
   - Latest completion times
4. Performance analysis
5. Conclusions

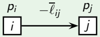
Example 1

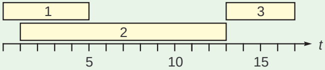
Earliest start schedule *ES* is not feasible

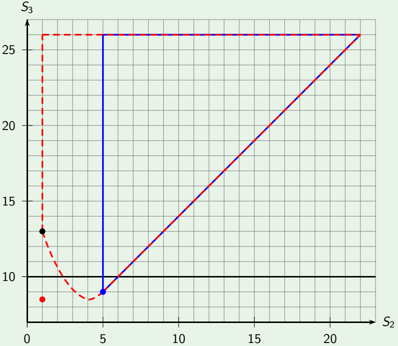


$p_2(\ell_{12}) = 0.5(\ell_{12}^{+})^2 + 4$

Legend

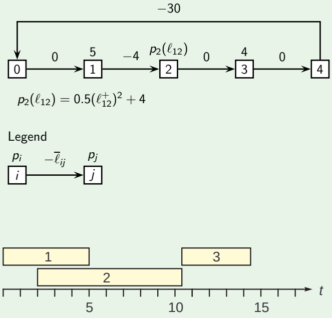


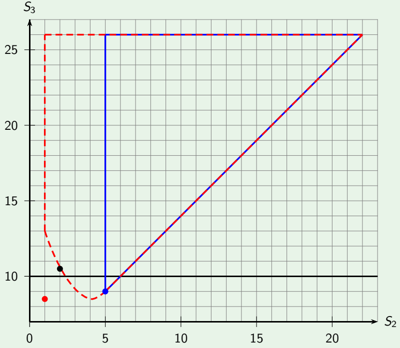




Example 1

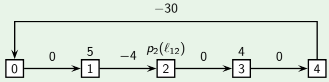
Earliest start schedule *ES* is not feasible





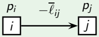
Example 1

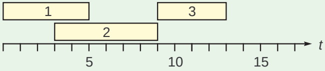
Earliest start schedule *ES* is not feasible

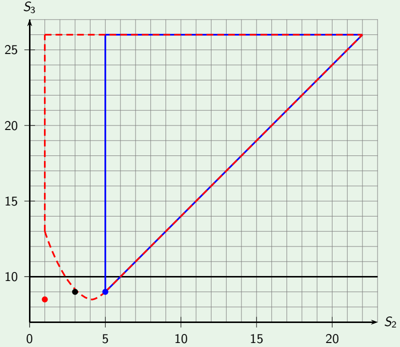


$p_2(\ell_{12}) = 0.5(\ell_{12}^{+})^2 + 4$

Legend

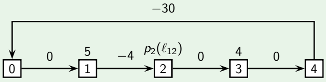




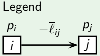


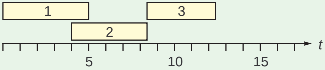
Example 1

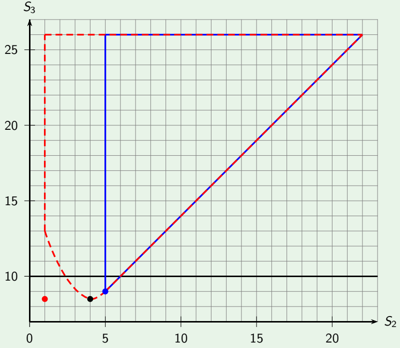
Earliest start schedule *ES* is not feasible



$p_2(\ell_{12}) = 0.5(\ell_{12}^+)^2 + 4$

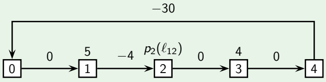






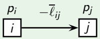
Example 1

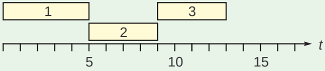
Earliest start schedule *ES* is not feasible

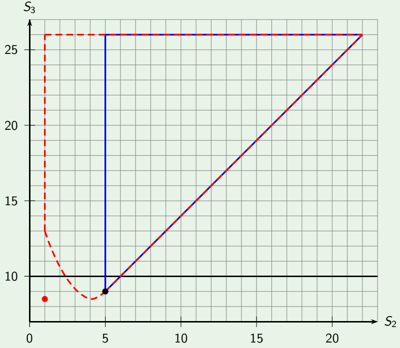


$p_2(\ell_{12}) = 0.5(\ell_{12}^+)^2 + 4$

Legend

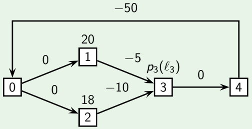




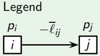


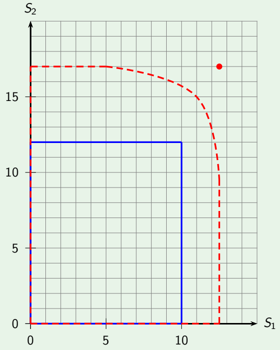
Example 2

Latest start schedule LS is not feasible



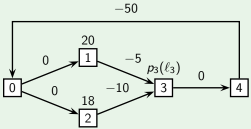
$p_3(\ell_{13}, \ell_{23}) = 0.01(\ell_{13}^+)^2 + 0.05(\ell_{23}^+)^2 + 20$





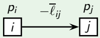
Example 2

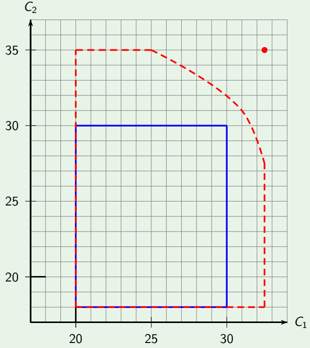
Latest completion schedule LC is not feasible



$$p_3(\ell_{13}, \ell_{23}) = 0.01(\ell_{13}^+)^2 + 0.05(\ell_{23}^+)^2 + 20$$

Legend





Feasibility of earliest completion schedule

Proposition

1. The earliest completion schedule $EC = (EC_j)_{j \in V}$ is feasible.
2. Let $\ell = \ell(EC)$ be the minimal vector of overlap times belonging to schedule EC. For each $(i, j) \in E$ it holds that

$$\ell_{ij} = \min\{EC_i, \min_{(h,j) \in E} (\overline{\ell}_{hj} + EC_i - EC_h)\} \qquad (1)$$

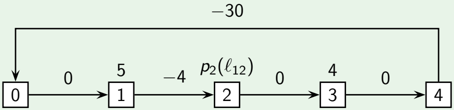
or ℓ_{ij} satisfies the following two optimality conditions:

$$\frac{d}{d\ell_{ij}} p_j((\ell_{ij} + EC_h - EC_i)_{(h,j) \in E}) = 1 \qquad (2)$$

$$\frac{d^2}{d\ell_{ij}^2} p_j((\ell_{ij} + EC_h - EC_i)_{(h,j) \in E}) \geq 0 \qquad (3)$$

Computation of optimal overlap times

Example 1 (cont'd)



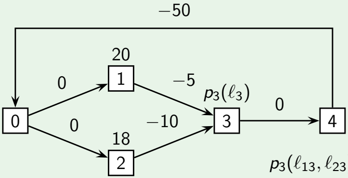
$p_2(\ell_{12}) = 0.5(\ell_{12}^+)^2 + 4$

- Equation (1): $\ell_{12} = \min\{EC_1, \overline{\ell}_{12}\} = \min\{5, 4\} = 4$
- Equation (2): $\frac{d}{d\ell_{12}} p_2(\ell_{12}) = \ell_{12} = 1$
- Equation (3): $\frac{d^2}{d\ell_{12}} p_2(\ell_{12}) = 1 > 0$
- $\ell_{12} = 4$: $p_2(\ell_{12}) = 12$, $C_2 = EC_1 - \ell_{12} + p_2(\ell_{12}) = 5 - 4 + 12 = 13$
- $\ell_{12} = 1$: $p_2(\ell_{12}) = 4.5$, $C_2 = EC_1 - \ell_{12} + p_2(\ell_{12}) = 5 - 1 + 4.5 = 8.5$

  $\Rightarrow EC_2 = 8.5$

Computation of optimal overlap times

Example 2 (cont'd)



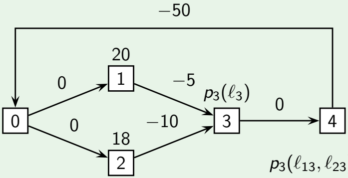
$$p_3(\ell_{13}, \ell_{23}) = 0.01(\ell_{13}^+)^2 + 0.05(\ell_{23}^+)^2 + 20$$

- Equation (1):

$$\ell_{13} = \min\{EC_1, \overline{\ell}_{13}, \overline{\ell}_{23} + EC_1 - EC_2\} = \min\{20, 5, 10 + 20 - 18\} = 5$$

Computation of optimal overlap times

Example 2 (cont'd)



$p_3(\ell_{13}, \ell_{23}) = 0.01(\ell_{13}^+)^2 + 0.05(\ell_{23}^+)^2 + 20$

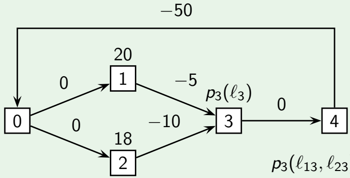
- Equation (2):

$$\begin{aligned} p_3(\ell_{13}) &= 0.01\ell_{13}^2 + 0.05(\ell_{13} + EC_2 - EC_1)^2 + 20 \\ &= 0.06\ell_{13}^2 - 0.2\ell_{13} + 20.2 \end{aligned}$$

$$\frac{d}{d\ell_{13}} p_3(\ell_{13}) = 0.12\ell_{13} - 0.2 = 1 \quad \Leftrightarrow \quad \ell_{13} = 10$$

Computation of optimal overlap times

Example 2 (cont'd)



$p_3(\ell_{13}, \ell_{23}) = 0.01(\ell_{13}^+)^2 + 0.05(\ell_{23}^+)^2 + 20$

- Equation (3): $\frac{d^2}{d\ell_{13}} p_3(\ell_{13}) = 0.12 > 0$
- $\ell_{13} = 5$: $p_3(\ell_{13}) = 20.7$, $C_3 = EC_1 - \ell_{13} + p_3(\ell_{13}) = 20 - 5 + 20.7 = 35.7$
- $\ell_{13} = 10$: $p_3(\ell_{13}) = 24.2$, $C_3 = EC_1 - \ell_{13} + p_3(\ell_{13}) = 20 - 10 + 24.2 = 34.2$

  $\Rightarrow EC_3 = 34.2$

Overview of structural issues

General duration functions $p_i(\ell_i)$			
Start times		Completion times	
$ES \notin \mathcal{S}_T$	$LS \notin \mathcal{S}_T$	$EC \in \mathcal{C}_T$	$LC \notin \mathcal{C}_T$
$\mathcal{S}_T$ not convex		$\mathcal{C}_T$ not convex	
$[ES_i, LS_i]$ not feasible		$[EC_i, LC_i]$ feasible	

Convex duration functions $p_i(\ell_i)$			
Start times		Completion times	
$ES \notin \mathcal{S}_T$	$LS \notin \mathcal{S}_T$	$EC \in \mathcal{C}_T$	$LC \notin \mathcal{C}_T$
$\mathcal{S}_T$ convex		$\mathcal{C}_T$ convex	
$[ES_i, LS_i]$ feasible		$[EC_i, LC_i]$ feasible	

Constant durations functions $p_i(\ell_i)$			
Start times		Completion times	
$ES \in \mathcal{S}_T$	$LS \in \mathcal{S}_T$	$EC \in \mathcal{C}_T$	$LC \in \mathcal{C}_T$
$\mathcal{S}_T$ convex		$\mathcal{C}_T$ convex	
$[ES_i, LS_i]$ feasible		$[EC_i, LC_i]$ feasible	

Outline

1. Problem definition
   - Concurrent engineering projects
   - Temporal constraints
   - Temporal scheduling problems
2. Structural issues
3. Temporal scheduling methods
   - Earliest completion times
   - Latest completion times
4. Performance analysis
5. Conclusions

Temporal scheduling methods

Overview

- Earliest completion times
  - EC can be computed with efficient label-correcting algorithm
- Latest completion times
  - LC_j has to be calculated separately for each activity j
  - Since interval $[EC_j, LC_j]$ does only contain feasible completion times C_j: perform binary search on interval $[EC_j, \overline{d}]$
  - For each C_j check whether or not C_j is feasible using modified label-correcting algorithm
- Earliest and latest start times
  - ES_j: Determine maximum $\ell_{ij} \leq \min_{(h,j)\in E}(\overline{\ell}_{hj} + EC_i - EC_h)$ such that $C_j = EC_i - \ell_{ij} + p_j((\ell_{ij} + EC_h - EC_i)_{(h,j)\in E}) \leq LC_j$
  - LS_j: Introduce dummy activity i with $p_i = 0$ and $\overline{\ell}_{ij} = 0$, LS_j coincides with LC_i

Earliest completion schedule

Label-correcting algorithm

for all $i \in V \setminus \{0\}$ **do** put $C_i := -\infty$;
put $C_0 := 0$, $Q := \{0\}$; (* Q is a queue *)
while $Q \neq \emptyset$ **do**
  pop i off queue Q;
  for all $(i, j) \in E$ **do**
    calculate ℓ_j^* using equations (1)–(3);
    if $C_j < C_i - \ell_{ij}^* + p_j(\ell_j^*)$ **then**
      update $C_j := C_i - \ell_{ij}^* + p_j(\ell_j^*)$;
      if $C_j > \overline{\ell}_{n+1,0}$ **then** terminate; (* $\mathcal{C}_T = \emptyset$ *)
      if $j \notin Q$ **then** push j into queue Q;
    end if
  end for
end while
return C;

Check feasibility of completion time

Modified label-correcting algorithm checking feasibility of C_h

for all $i \in V \setminus \{h\}$ **do** put $C_i := -\infty$;
put $Q := \{i\}$; (* Q is a queue *)
while $Q \neq \emptyset$ **do**
  pop i off queue Q;
  for all (i, j) **do**
    calculate ℓ_j^*;
    if $C_j < C_i - \ell_{ij}^* + p_j(\ell_j^*)$ **then**
      if $j = h$ **then return false**; (* C_h is not feasible *)
      $C_j := C_i - \ell_{ij}^* + p_j(\ell_j^*)$;
      if $j \notin Q$ **then** push j into queue Q;
    end if
  end for
end while
return true; (* C_h is feasible *)

Outline

Performance analysis

Test bed

- Temporal scheduling methods implemented under MS Visual C++ 6.0 Developer Studio
- Intel Pentium 1.7 GHz PC with 524 MB RAM running under Windows 2000 professional
- Full factorial design experiment with 2,160 instances

Symbol	Parameter	Values
p	number of (sub-)projects	2, 5, 10
n	number of real activities per project	10, 20, 50
OS	order strength	0.25, 0.5

- Duration functions: nonconvex 3rd-order polynomials

Performance analysis

Impact of number of projects on CPU times [ms]

	$p = 2$	$p = 5$	$p = 10$
CPU_{EC}	0.5	2.3	3.0
CPU_{LC}	824.3	5,572.8	9,729.0

Impact of number of activities per project on CPU times [ms]

	$n = 10$	$n = 20$	$n = 50$
CPU_{EC}	0.4	0.9	4.4
CPU_{LC}	266.6	1,462.3	14,397.2

Impact of order strength on CPU times [ms]

	$OS = 0.25$	$OS = 0.5$
CPU_{EC}	1.6	2.2
CPU_{LC}	3,604.2	7,146.5

Outline

Conclusions

Summary

- Concurrent engineering projects
- Trade-off between overlap times and activity durations
- Earliest completion schedule is feasible, remaining extremal schedules are not feasible
- Algorithms perform reasonably well for project portfolios with up to 500 activities

Expansions

- Priority-rule based method for resource-constrained variant of the problem
- Objective function: earliness-tardiness cost with respect to due date
- Sequential version: mean CPU time less than one second