Earliness/Tardiness Scheduling by Constraint Programming

Jan Kelbel, Zdeněk Hanzálek
Department of Control Engineering
Faculty of Electrical Engineering, Czech Technical University
Karlovo nám. 13, 121 35 Prague 2
e-mail: kelbelj@fel.cvut.cz

Motivation

- **Lacquer production scheduling**
- Case study from AMETIST project
  - scheduling lacquer production – 29 lacquer orders
  - production of each order consists of a set of operations – extended job-shop scheduling problem
  - penalties for orders finished late (after due date)
  - storage cost for orders finished too early

  } ETSP
  - using Timed Automata approach

- ILOG OPL Studio as CP tool
- Compare CP results with TA

Related Work

[Behrmann, G., Brinksma, E., Hendriks, M., Mader, A., 2005] Production Scheduling by Reachability Analysis – A Case Study.

Original paper from AMETIST project with the case study

[Baker, K.R., Scudder, G.D., 1990] Sequencing with earliness and tardiness penalties: A review.

[Sourd, F., Kedad-Sidhoum, S., 2003] The One Machine Scheduling With Earliness and Tardiness Penalties. Branch and Bound

[Beck, J.C., Refalo, P. 2003] – A Hybrid Approach to Scheduling with Earliness and Tardiness Costs. Job-shop ETSP, hybrid CSP and MIP approach

Introduction to CP

Constraint Satisfaction Problem (CSP) over finite domains

- finite set of variables
- for each variable a finite set of possible values (called domain)
- finite set of constraints restricting possible combinations of values of variables

CSP is solved using

- constraint propagation (arc consistency)
- search
  – search tree – variable selection, value assignment
  – search strategy

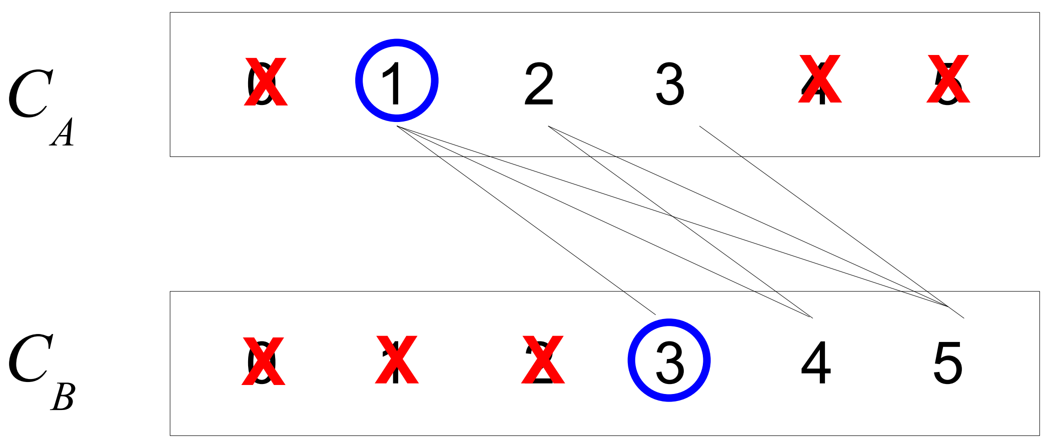
Example: Scheduling with CP

Tasks T_A and T_B,

$$T_A \quad p_A = 1 \quad C_A \in \{0, \ldots, 5\}$$

$$T_B \quad p_B = 2 \quad C_B \in \{0, \ldots, 5\}$$

$$T_A < T_B \quad \Rightarrow \quad C_A \le C_B - p_B$$



E/T Job Shop

- Set of jobs $J = \{ J_{1,} \ldots, J_n \}$
- For each job set of tasks with precedences

$$T_j = \{ T_{j,1}, \ldots, T_{j,n_j} \}$$

$$T_{j,i} < T_{j,i+1} \quad for\,all \quad i \in \{ 1, \ldots, n_j - 1 \}$$

- Each task uses a resource from set *R*

Job Shop

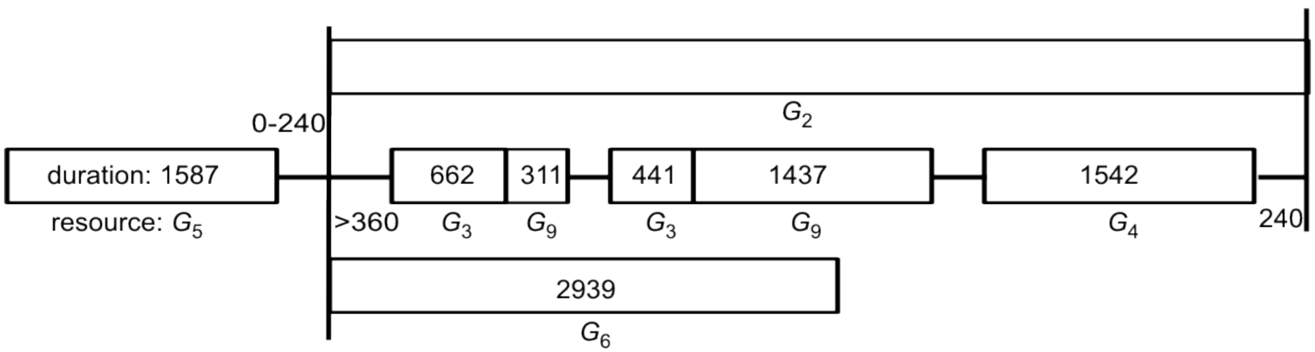
- Cost function of a job

$$f_j = max(\underbrace{\alpha_j (d_j - C_j)}_{earliness \quad cost}, \underbrace{\beta_j (C_j - d_j)}_{tardiness \quad cost})$$

Description of The Case Study

- Each job is specified by
  - release date
  - due date
  - earliness and tardiness costs

    earliness tardiness job-shop
  - quantity of lacquer
  - one of three lacquer recipes



- Resources
  unary / cumulative
  operating hours – full time / in two shifts / in three shifts

- Feasibility / Minimization of sum of earliness/tardiness costs

Three Case Studies

- Basic case study (**BF**)

  Feasibility of a basic model

- Extended case study (**EO**)

  Optimization, in addition to basic model

  - operating hours of resources
  - breakable tasks
  - changeover time and cost on one resource group
  - processing time of tasks according to quantity of lacquer

- Extended case study with performance factor (**EOP**)

  in addition to **EO**

  - extended processing times due to breakdown or maintenance (EOP instance)

Reducing the state space

- Non-overtaking (**NO**) heuristic constraint – precedence relation between tasks of jobs of the same lacquer type (sorted according to due date)
  Tasks of a job with earlier due date start earlier.
  [from the original AMETIST paper]

- Simplified objective function for **extended case study**
  - due to Earliness/Tardiness costs ratio 1/50 for majority of jobs due dates → deadlines (infinite tardiness cost)
  - Single earliness cost for all jobs j, $\alpha_j = 1$.

$$F = \sum_{j \in J} \alpha_j (d_j - C_j)$$

  - Special case – Cost optimization (**CO**), job specific α_j

Search

- Conversion to a problem of minimizing total waiting time by reversing time axis (to make it easier for built-in search procedures)
  - Deadline → release date
  - Completion time → starting time

$$F = \sum_{j \in J} \alpha_j (S_j - r_j)$$

- Search procedure defining the shape of the search tree – time-directed labeling
  1. sort jobs by earliest starting time in nondecreasing order
  2. for each job state two alternatives
     - assign earliest starting time as starting time of job
     - increase earliest starting time
- Search strategies used
  - Depth First Search (**DFS**)
  - Limited Discrepancy Search (**LDS**)

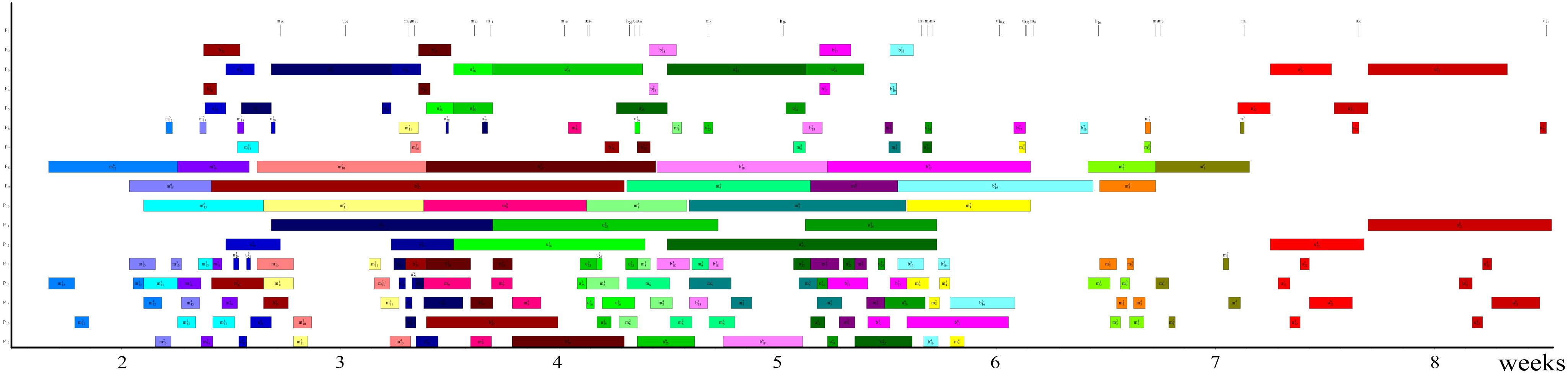
Results for the Case Study

- size of the case study instances:
  - 139 tasks (110 of them breakable) in 29 jobs
  - time for the schedule – 9 weeks in 1 minute resolution
  - time for each job – 2 weeks

	DFS-NO		LDS-NO		LDS		LDS CO		timed automata	
	cost	CPU	cost	CPU	cost	CPU	cost	CPU	cost	CPU
BF	-	0.09 s	-	0.09 s	-	*	-	*	-	0.1 s
EO	455,758	1374 s	455,758	18.4 s	457,523	181 s	454,246	185 s	*	*
EOP	**1,234,959**	1026 s	**1,186,851**	12.7 s	**920,240**	220 s	**886,535**	206 s	**≈ 2,100,000**	≈ 600 s

- value not applicable

* value not available



Search Procedure for ETSP

- Improved version – for earliness and **tardiness** costs
- denoted as Cost Directed Labeling (CDL)
  1. sort last tasks of jobs by size of domain in nondecreasing order
  2. for each job select t in domain Cj leading to smallest cost and state two alternatives
     - assign $Cj = t$
     - assign $Cj \neq t$

- improved solution of EOP case study instance

  from the cost 886,535 to 777,249

Conclusion and Future Work

- With CP a better solution was found than with timed automata
- ET Job Shop
- Hybrid CP/MIP methods are better

Future Work

- Hybrid CP/Graph algorithm method