Manpower Routing and Scheduling with Temporal Dependencies Between Tasks

Anders Dohn

Jesper Larsen

Matias Sevel Rasmussen

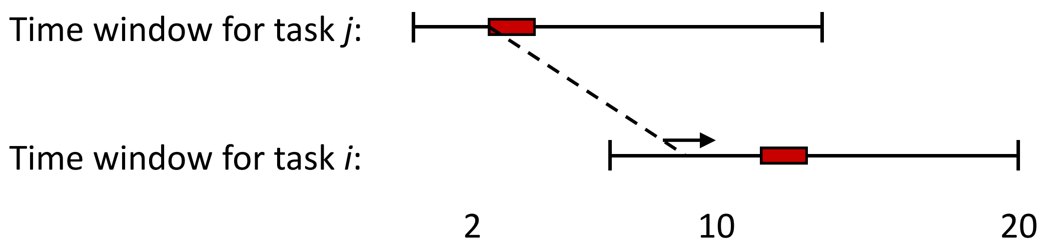
Technical University of Denmark
Department of Management Engineering

Outline

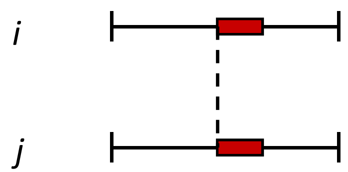
- What is a temporal dependency?
- Incentive for introducing temporal dependencies.
- Modeling temporal dependencies.
- Branching on time windows.
- Results from practical applications.
- Conclusions.

Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark
DTU Management Engineering
Department of Management Engineering

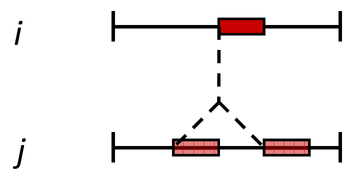
Tasks with Temporal Dependencies



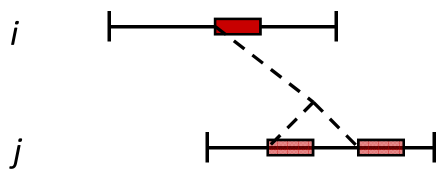
Synchronization:



Overlap:



Min/max gap:



Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark

Temporal Dependencies in Practice

- Ground handling in airports
  - Synchronization (Job teaming)
  - Overlap (Not used in practice yet, but is requested by users)
- Home care crew scheduling
  - Synchronization (Mainly for lifting)
  - Overlap (Lifting)
  - Min and max gap (E.g. medication and laundry)
- Allocation of technicians to service jobs [Li et al. 2005].
- Dial-a-Ride for disabled persons [Rousseau et al. 2003].
- Aircraft fleet assignment and routing [Ioachim et al. 1999].
- Machine scheduling with precedence constraints [van den Akker et al. 2006].

The General Temporal Dependency Constraint

$$t_i + p_{ij} \leq t_j$$

Column Generation - Mathematical Model

$$\textbf{min} \quad \sum_{k\in K}\sum_{r\in R^k} c_r^k \lambda_r^k + \sum_{i\in N} c_i \Lambda_i$$

$$\textbf{s.t.} \quad \sum_{k\in K}\sum_{r\in R^k} a_{ir}^k \lambda_r^k + \Lambda_i = 1 \qquad \forall i \in N$$

$$\sum_{r\in R^k} \lambda_r^k = 1 \qquad \forall k \in K$$

$$\sum_{k\in K}\sum_{r\in R^k} t_{ir}^k \lambda_r^k + p_{ij} \leq \sum_{k\in K}\sum_{r\in R^k} t_{jr}^k \lambda_r^k + M(\Lambda_i + \Lambda_j) \qquad \forall (i,j,p_{ij}) \in P$$

$$\lambda_r^k, \Lambda_i \in \{0,1\} \; \forall k \in K, r \in R^k, i \in N$$

Variables:

$$\lambda_r^k = \begin{cases} 1 \text{ if route } r \text{ is chosen for team } k \\ 0 \text{ otherwise} \end{cases}$$

$$\Lambda_i = \begin{cases} 1 \text{ if task } i \text{ is uncovered} \\ 0 \text{ otherwise} \end{cases}$$

Sets:

N	Tasks
K	Teams / Vehicles
R^k	Routes
P	Temporal Dependencies

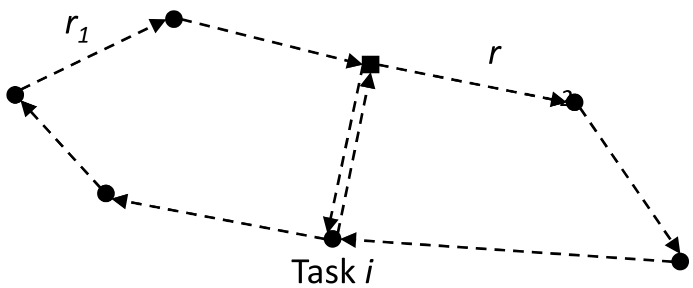
Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark

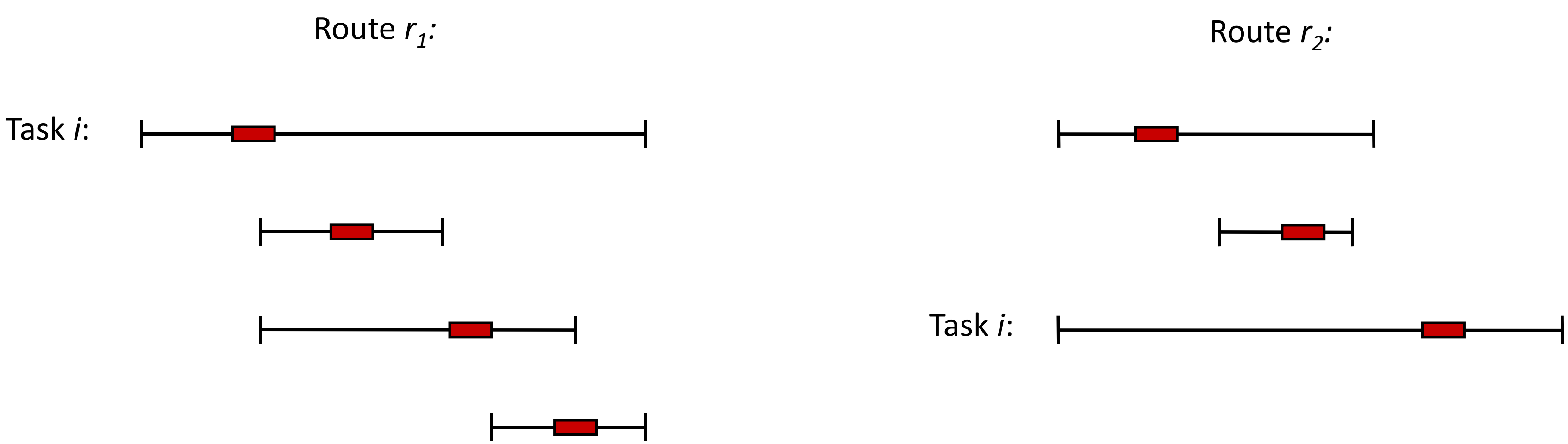
Solution Approaches

- Relaxing the temporal dependency constraints:
  - The master problem is a Set Partitioning Problem.
  - The subproblem is an Elementary Shortest Path Problem with Time Windows.
  - Temporal dependencies are enforced by branching.
- Solving the presented set partitioning problem with temporal dependency constraints:
  - The master problem is a Set Partitioning Problem with additional non-binary constraints.
  - The subproblem is an Elementary Shortest Path Problem with Time Windows and Linear Node Costs.
    - Only the acyclic case has been considered in the literature [Ioachim et al. 1997].

Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark

Visualizing the Routes

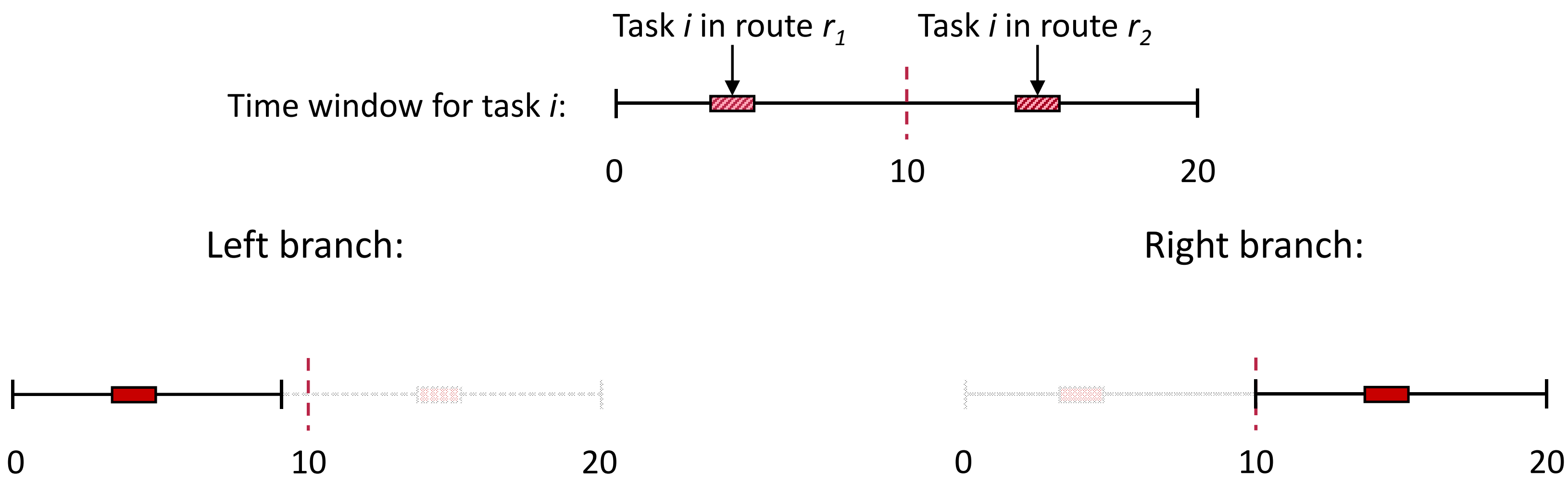




Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark

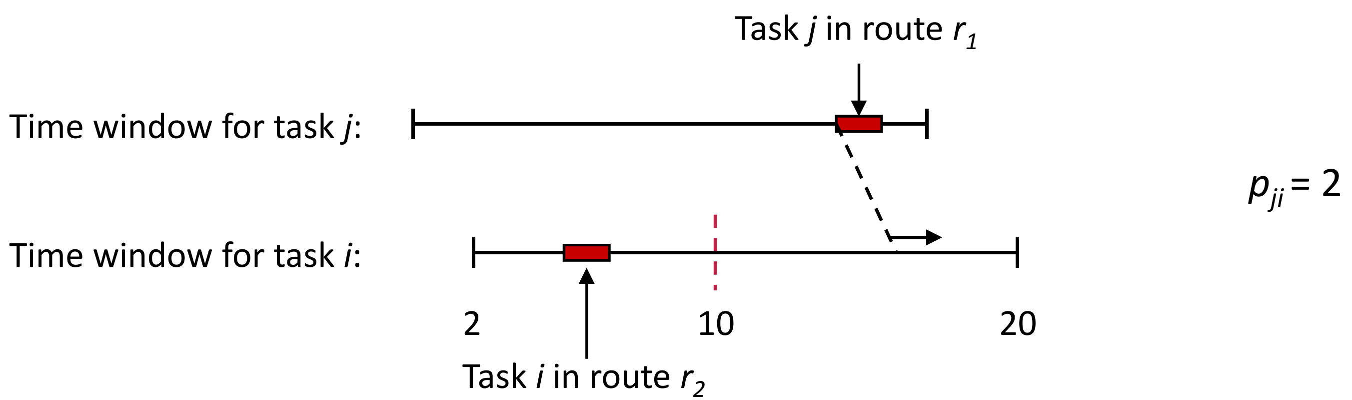
Branching on Time Windows

- Will remove most fractional values.
- Will enforce all temporal dependencies.
- Proposed as branching strategy to solely remove fractional values in traditional VRPTW [Gélinas et al. 1995].

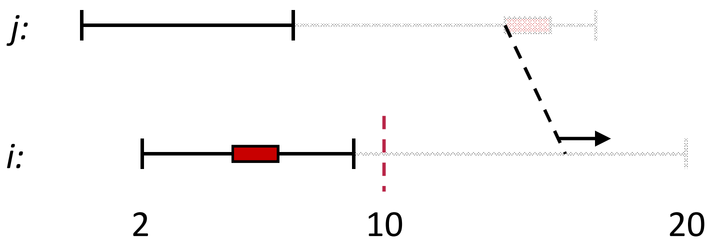


Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark
DTU Management Engineering
Department of Management Engineering

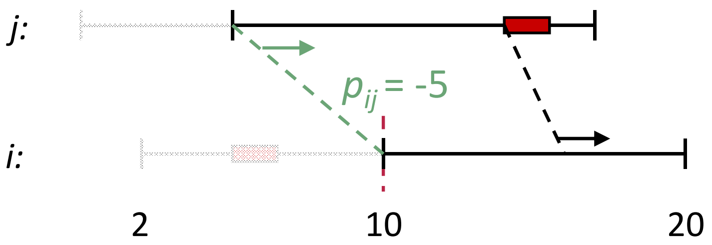
Branching on Time Windows



Left branch:



Right branch:



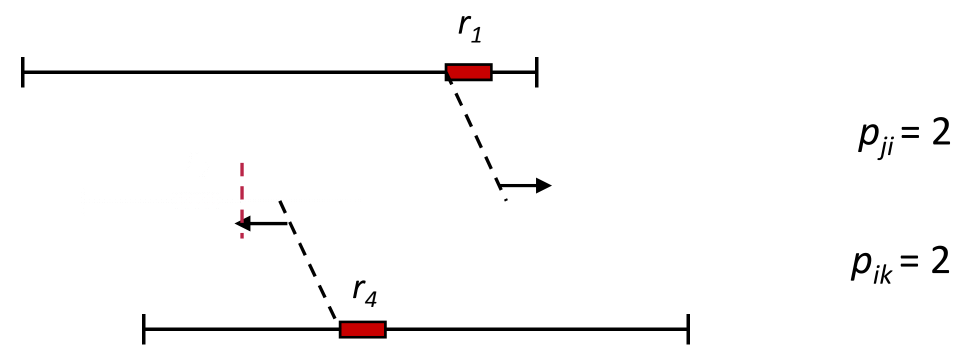
Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark

Branching on Time Windows

Task *j* in route r_1:

Task *i* in route r_2
and route r_3:

Task *k* in route r_4:



Infeasible routes:

	Left branch:	Right branch:
Branching candidate 1:	r_3, r_1	r_2
Branching candidate 2:	r_3, r_1	r_2, r_4
Branching candidate 3:	r_1	r_2, r_3, r_4

Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark

Results - Ground Handling in Airports

- Real-life data from two of Europe's major airports.
- 12 data instances of varying size
  (12-27 teams, 100-300 tasks):
  - 20-60 synchronization constraints.
  - Optimal solutions for 11 of the 12 datasets.
  - Solutions close to the lower bound for the last dataset.
- Solution time:
  - 6 instances: less than 1 hour.
  - Remaining 6 instances: up to 10 hours.

Results - Home Care Crew Scheduling

- Real-life instances: 6-15 caretakers, 60-150 visits per day.
- Current practice: Tailored heuristic and manual planning
  - Number of uncovered tasks reduced by 50%.
  - In manual planning: Time windows and competence requirements were modified to be able to find feasible solutions.
    - Constraints for up to 20% of tasks were modified in some instances.
    - These adjustments are not allowed in the column generation based optimization.
  - Transportation time is approximately the same as in manual planning.
- Solution time: Less than 1 hour.

Conclusions

- There is a clear incentive for introducing temporal dependencies in the models.
  - Certain combinations are found frequently in practice (e.g. synchronization and overlap)
- The general Temporal Dependency Constraint has been introduced and included in column generation.
  - Various practical problems can be modeled.
- Practical applications show encouraging results.

Anders Dohn - adh@imm.dtu.dk
Technical University of Denmark
DTU Management Engineering
Department of Management Engineering

Thank you for your attention.