# On the Tractability of Digraph-Based Task Models

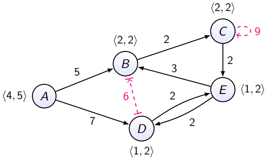
Martin Stigge

Uppsala University, Sweden

Joint work with Pontus Ekberg, Nan Guan and Wang Yi

# Analysis of Abstract Models

- Model hard real-time systems
  - Analysis: Guarantee deadlines
  - *Expressiveness* of models?
  - *Efficiency* of analysis?

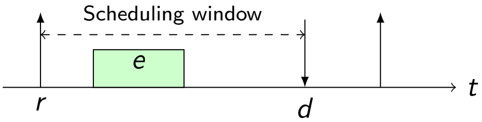



**Question**

- How expressive can a model be?
- ... with a *tractable* feasibility test?

# Context: Real-Time Task Models

- System is composed of *tasks*, releasing *jobs* $J = (r, e, d)$
  - Release time $r$
  - Worst-case execution time $e$
  - Deadline $d$

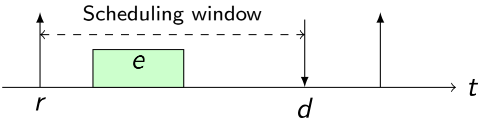



- Feasibility: *Can we schedule s.t. all jobs meet their deadlines?*
- In this work:
  - Preemptive schedules
  - On uniprocessors
  - Independent jobs

In this setting: EDF is optimal.
*Feasible* $\Leftrightarrow$ *Schedulable with EDF*

# Context: Real-Time Task Models

- System is composed of *tasks*, releasing *jobs* $J = (r, e, d)$
  - Release time $r$
  - Worst-case execution time $e$
  - Deadline $d$



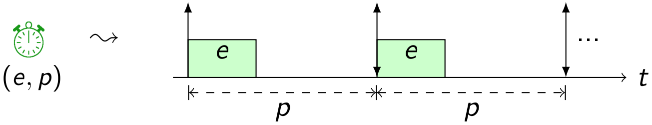

- Feasibility: *Can we schedule s.t. all jobs meet their deadlines?*
- In this work:
  - Preemptive schedules
  - On uniprocessors
  - Independent jobs

$\}$ In this setting: EDF is optimal.
*Feasible* $\Leftrightarrow$ *Schedulable with EDF*

# The Liu and Layland (L&L) Task Model

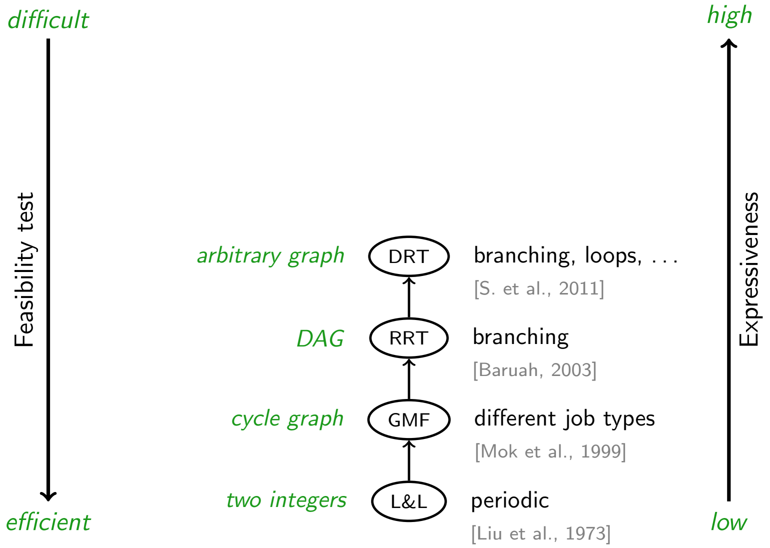
(Liu and Layland, 1973)

- Tasks are *periodic*
  - Job WCET $e$
  - Minimum inter-release delay $p$ (implicit deadline)

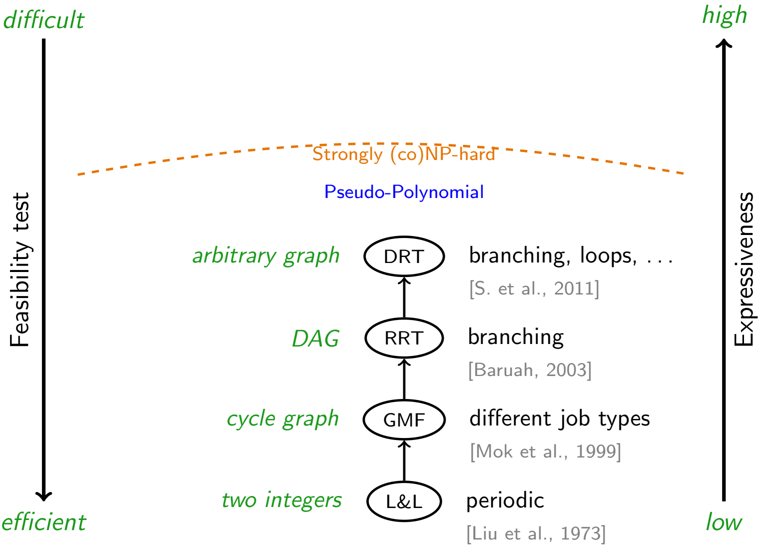



- Advantages: Well-known model; *efficient* schedulability test
- Disadvantage: Very *limited expressiveness*

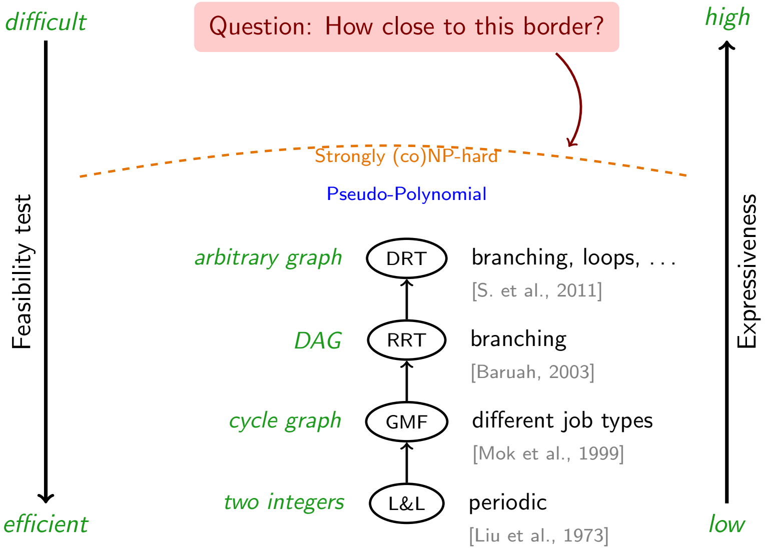
# Hierarchy of Models



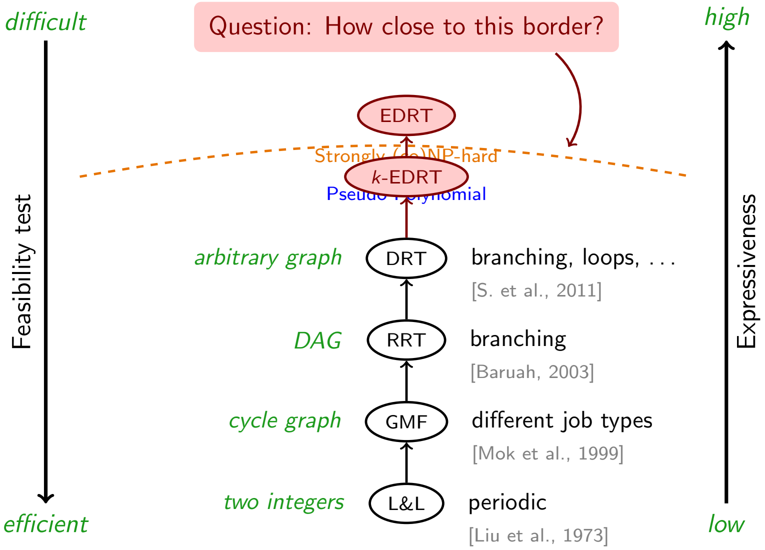
# Hierarchy of Models



# Hierarchy of Models

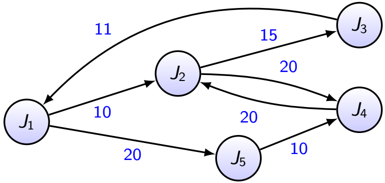


# Hierarchy of Models



# The Digraph Real-Time (DRT) Task Model

- Branching, cycles (loops), ...
- *Directed graph* for each task
  - Vertices $J$: jobs to be released (with WCET and deadline)
  - Edges $(J_i, J_j)$: minimum inter-release delays $p(J_i, J_j)$

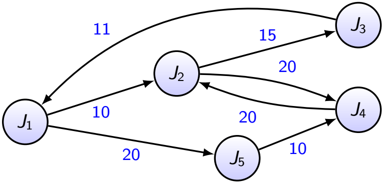



**Theorem (S. et al., RTAS 2011)**

*For DRT task systems $\tau$ with a utilization bounded by any $c < 1$, feasibility can be decided in pseudo-polynomial time.*

# The Digraph Real-Time (DRT) Task Model

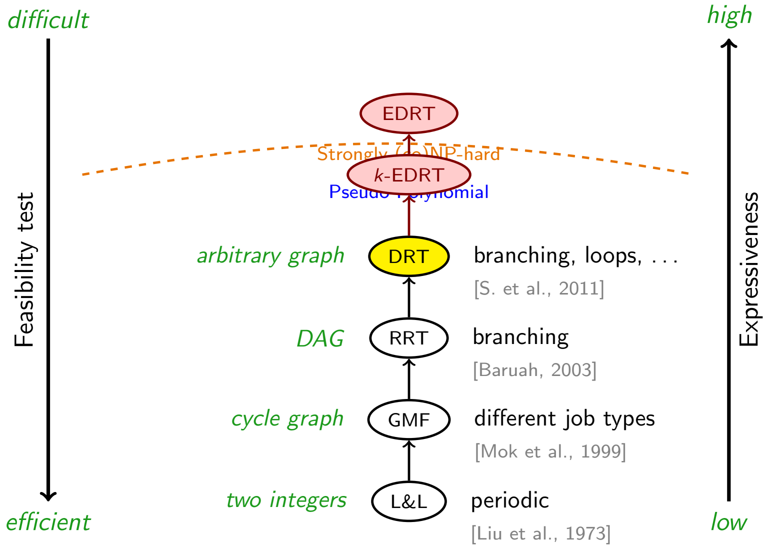
- Branching, cycles (loops), ...
- *Directed graph* for each task
  - Vertices $J$: jobs to be released (with WCET and deadline)
  - Edges $(J_i, J_j)$: minimum inter-release delays $p(J_i, J_j)$



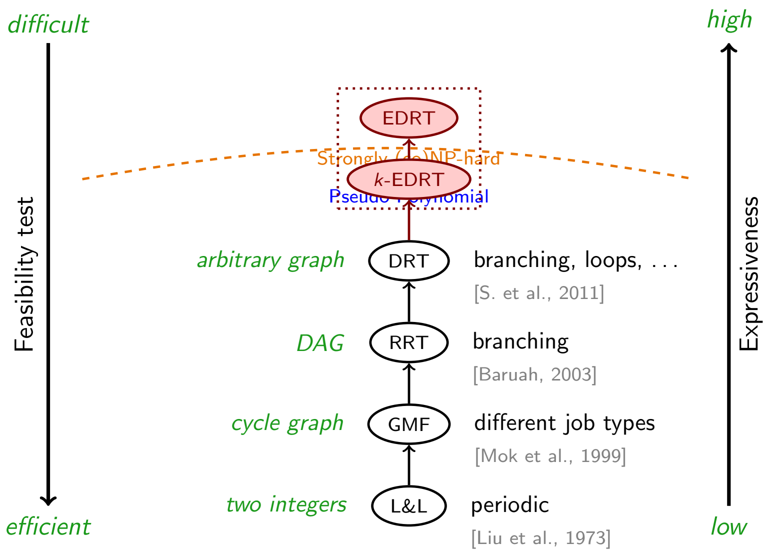

**Theorem (S. et al., RTAS 2011)**

*For DRT task systems $\tau$ with a utilization bounded by any $c < 1$, feasibility can be decided in pseudo-polynomial time.*

# Hierarchy of Models

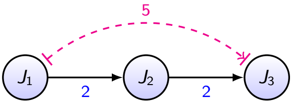


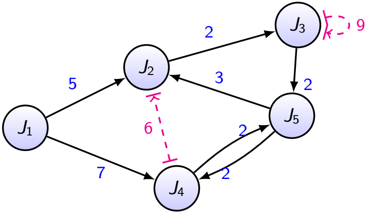
# Hierarchy of Models



# Extending DRT: Global Timing Constraints – *This Work*

- Delays in DRT only for *adjacent* jobs
- What about adding *global delay constraints*?

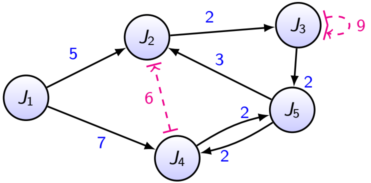



- Motivation:
  - Mode sub-structures
  - Burstiness
  - ...

# Extended DRT (EDRT) – *This Work*

- Extends DRT with *global delay constraints*
- *Directed graph* for each task
  - Vertices $J$: jobs to be released (with WCET and deadline)
  - Edges $(J_i, J_j)$: minimum inter-release delays $p(J_i, J_j)$
  - $k$ global constraints $(J_i, J_j, \gamma)$



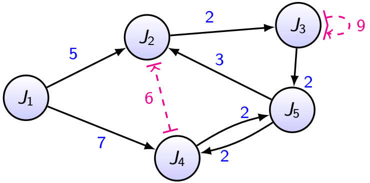

**Theorem (Our technical result)**

*For $k$-EDRT task systems with bounded utilization, feasibility is*

1. *decidable in pseudo-polynomial time if $k$ is constant, and*
2. *strongly coNP-hard in general.*

# Extended DRT (EDRT) – *This Work*

- Extends DRT with *global delay constraints*
- *Directed graph* for each task
  - Vertices $J$: jobs to be released (with WCET and deadline)
  - Edges $(J_i, J_j)$: minimum inter-release delays $p(J_i, J_j)$
  - $k$ global constraints $(J_i, J_j, \gamma)$

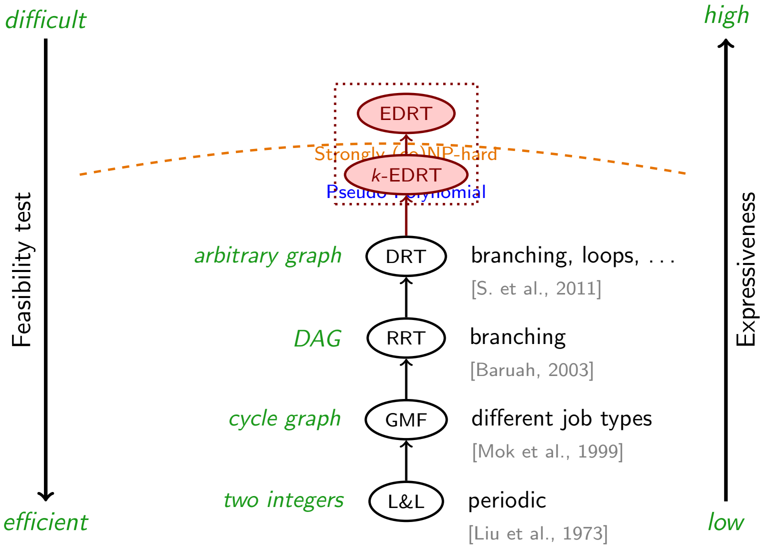



## Theorem (Our technical result)

*For $k$-EDRT task systems with bounded utilization, feasibility is*

1. *decidable in pseudo-polynomial time if $k$ is constant, and*
2. *strongly coNP-hard in general.*

# Hierarchy of Models

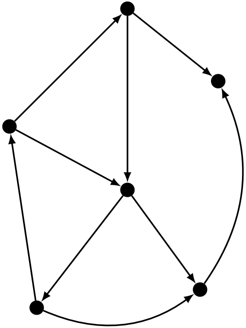


# Fahrplan

# Fahrplan

# Hardness for EDRT

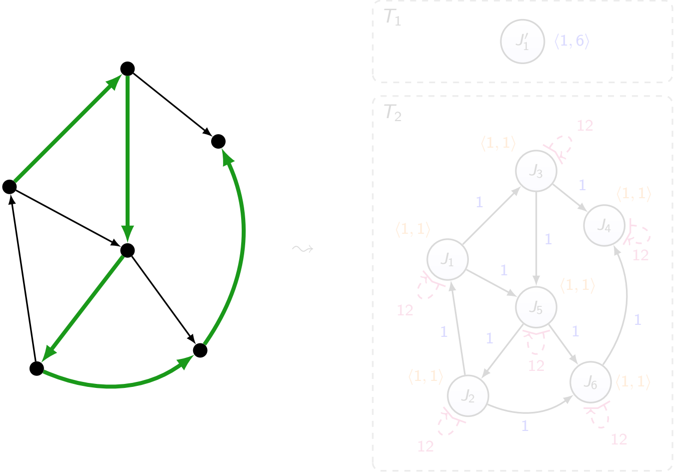
- Number of constraints now *not constant*
- Reduction from *Hamiltonian Path Problem* (strongly NP-hard)



$\rightsquigarrow$

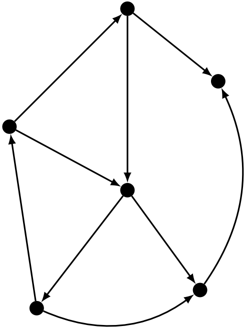
# Hardness for EDRT

- Number of constraints now *not constant*
- Reduction from *Hamiltonian Path Problem* (strongly NP-hard)

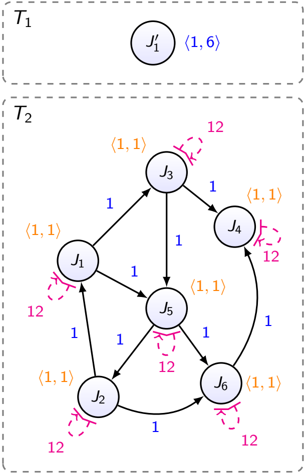


# Hardness for EDRT

- Number of constraints now *not constant*
- Reduction from *Hamiltonian Path Problem* (strongly NP-hard)

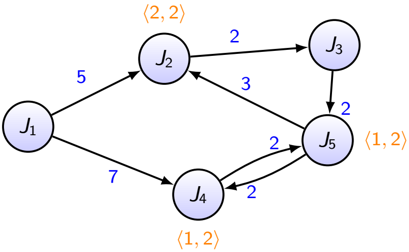


$\rightsquigarrow$



# Analysing $k$-EDRT: The Problem

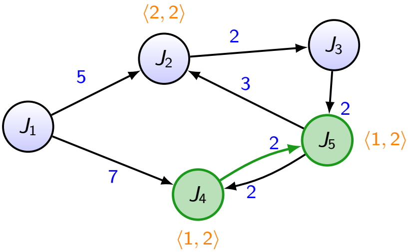
- Detour: Analyzing DRT
  - Using demand bound functions
  - Compute exec. demand and deadline for *all paths* in $G(T)$




- Problem: Constraints ignored during path exploration
  - $\langle 2, 4 \rangle$ is lacking constraint information
  - ... about the *active constraint*

# Analysing $k$-EDRT: The Problem

- Detour: Analyzing DRT
  - Using demand bound functions
  - Compute exec. demand and deadline for *all paths* in $G(T)$

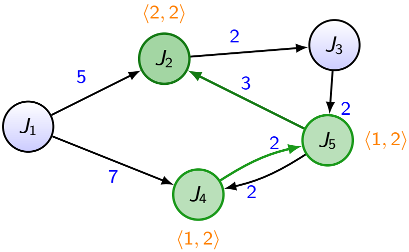



*Demand pair:* $\langle 2, 4 \rangle$

- Problem: Constraints ignored during path exploration
  - $\langle 2, 4 \rangle$ is lacking constraint information
  - ... about the *active constraint*

# Analysing $k$-EDRT: The Problem

- Detour: Analyzing DRT
  - Using demand bound functions
  - Compute exec. demand and deadline for *all paths* in $G(T)$



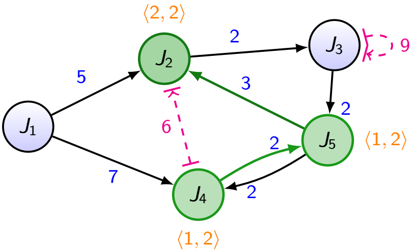

*Demand pair:* $\langle 2, 4\rangle$

*New demand pair:* $\langle 4, 7\rangle$

- Problem: Constraints ignored during path exploration
  - $\langle 2, 4\rangle$ is lacking constraint information
  - ... about the *active constraint*

# Analysing $k$-EDRT: The Problem

- Detour: Analyzing DRT
  - Using demand bound functions
  - Compute exec. demand and deadline for *all paths* in $G(T)$



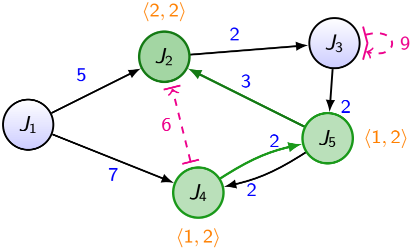

*Demand pair:* $\langle 2, 4 \rangle$

*New demand pair:* $\langle 4, 7 \rangle$

- Problem: Constraints ignored during path exploration
  - $\langle 2, 4 \rangle$ is lacking constraint information
  - ... about the *active constraint*

# Analysing $k$-EDRT: The Problem

- Detour: Analyzing DRT
  - Using demand bound functions
  - Compute exec. demand and deadline for *all paths* in $G(T)$



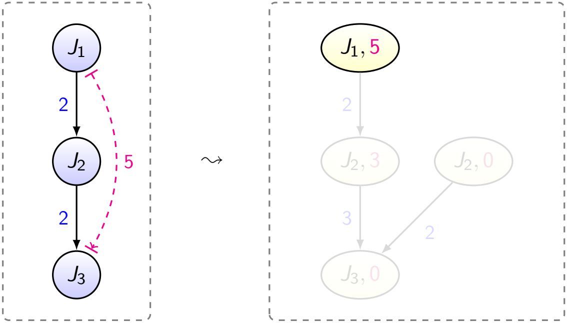

*Demand pair:* $\langle 2,4 \rangle$

*New demand pair:* $\cancel{\langle 4,7 \rangle}$ $\langle 4,8 \rangle$

- Problem: Constraints ignored during path exploration
  - $\langle 2,4 \rangle$ is lacking constraint information
  - ... about the *active constraint*

# Analysing $k$-EDRT: Solution

- Translate $k$-EDRT into *plain DRT*
  - Represent active constraints as *countdowns*
  - Store countdown values in DRT vertices
  - Preserve demand bound function

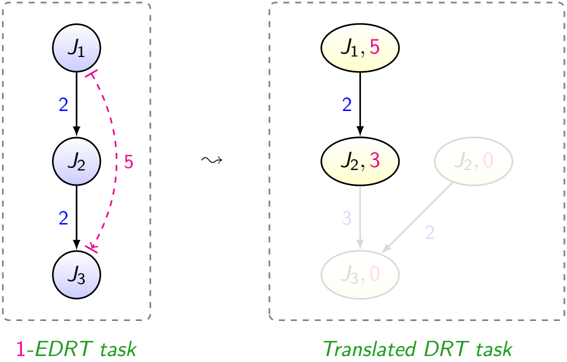



*1-EDRT task* *Translated DRT task*

- Optimizations: Efficient on-the-fly translation

# Analysing $k$-EDRT: Solution

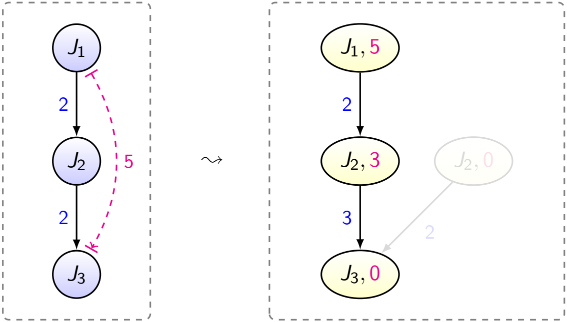
- Translate $k$-EDRT into *plain DRT*
  - Represent active constraints as *countdowns*
  - Store countdown values in DRT vertices
  - Preserve demand bound function




- Optimizations: Efficient on-the-fly translation

# Analysing $k$-EDRT: Solution

- Translate $k$-EDRT into *plain DRT*
  - Represent active constraints as *countdowns*
  - Store countdown values in DRT vertices
  - Preserve demand bound function



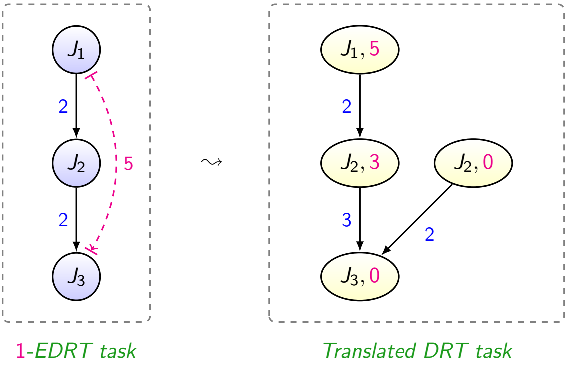

1-*EDRT task*

*Translated DRT task*

- Optimizations: Efficient on-the-fly translation

# Analysing $k$-EDRT: Solution

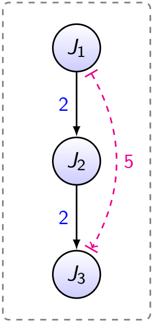
- Translate $k$-EDRT into *plain DRT*
  - Represent active constraints as *countdowns*
  - Store countdown values in DRT vertices
  - Preserve demand bound function




- Optimizations: Efficient on-the-fly translation

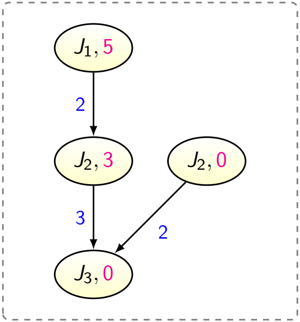
# Analysing $k$-EDRT: Solution

- Translate $k$-EDRT into *plain DRT*
  - Represent active constraints as *countdowns*
  - Store countdown values in DRT vertices
  - Preserve demand bound function




1-*EDRT task*

$\rightsquigarrow$

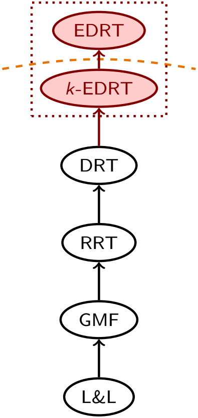



*Translated DRT task*

- Optimizations: Efficient on-the-fly translation

# Summary and Outlook

- Introduced Extension to DRT Task Model
  - *Global delay constraints*
- Establishes tractability borderline for feasibility test
  - Constant number of constraints: *tractable*
  - Unbounded number of constraints: *intractable*
- Ongoing work:
  - Global constraints for simpler models (RRT, GMF)
  - Interaction with Resource Sharing Protocols (cf. talk tomorrow)



# Q & A

Thanks!