



Work-in-Progress: Partial-Order Reduction in Reachability-based Response-Time Analyses

RTSS 2021

Sayra Ranjha, Mitra Nasri, Geoffrey Nelissen



State of the art

Fixed-point iteration-based
[Audsley'93, Davis'07]

- Fast
- Exact in specific cases

Reachability-based
[Guan'07, Sun'16, Yalcinkaya'19]

- Exact in more general cases
- Poor scalability

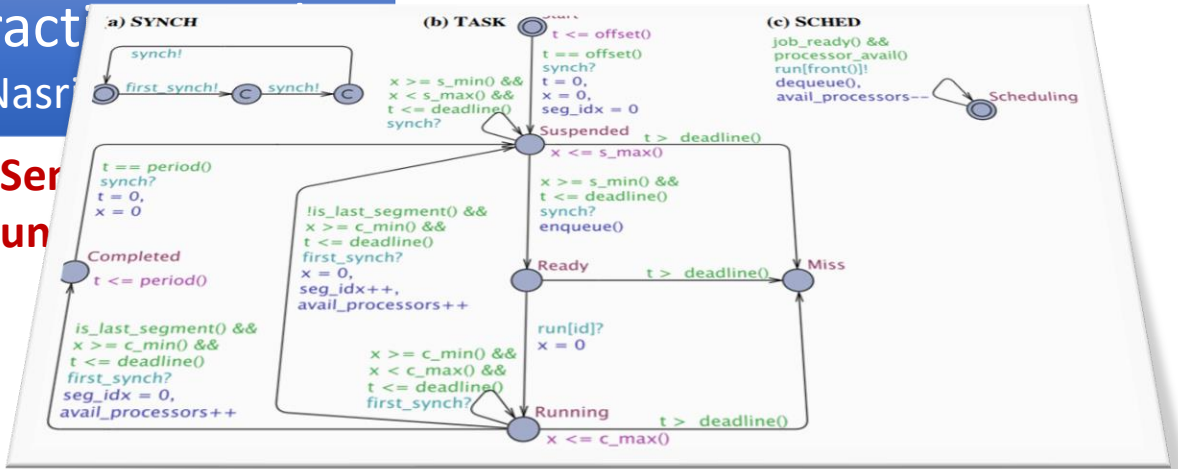
$$R_i^{(0)} = C_i + \sum_{j=1}^{i-1} C_j$$

$$R_i^{(k)} = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{R_i^{(k-1)}}{T_j} \right\rceil C_j$$

Schedule-abstract

Nasri'17, Nasri'18, Nasri'19

- More scalable
- Exact in more general cases
- Ser
- un



[Audsley'93] N. Audsley, A. Burns, M. Richardson, K. Tindell, and A. J. Wellings, "Applying New Scheduling Theory to Static Priority Pre-emptive Scheduling," Software Engineering Journal, vol. 8, no. 5, pp. 284–292, 1993.

[Davis'07] R. Davis, A. Burns, R. Bril, and J. Lukkien, "Controller Area Network (CAN) schedulability analysis : refuted, revisited and revised," Real-Time Systems, vol. 35, no. 3, pp. 239–272, 2007.

[Guan'07] N. Guan, Z. Gu, Q. Deng, S. Gao, and G. Yu, "Exact Schedulability Analysis for Static-Priority Global Multiprocessor Scheduling Using Model-Checking," in SEUS, 2007, p. 263–272.

[Sun'16] Y. Sun and G. Lipari, "A Pre-Order Relation for Exact Schedulability Test of Sporadic Tasks on Multiprocessor Global Fixed-Priority Scheduling," Real-Time Systems, vol. 52, no. 3, p. 323–355, 2016.

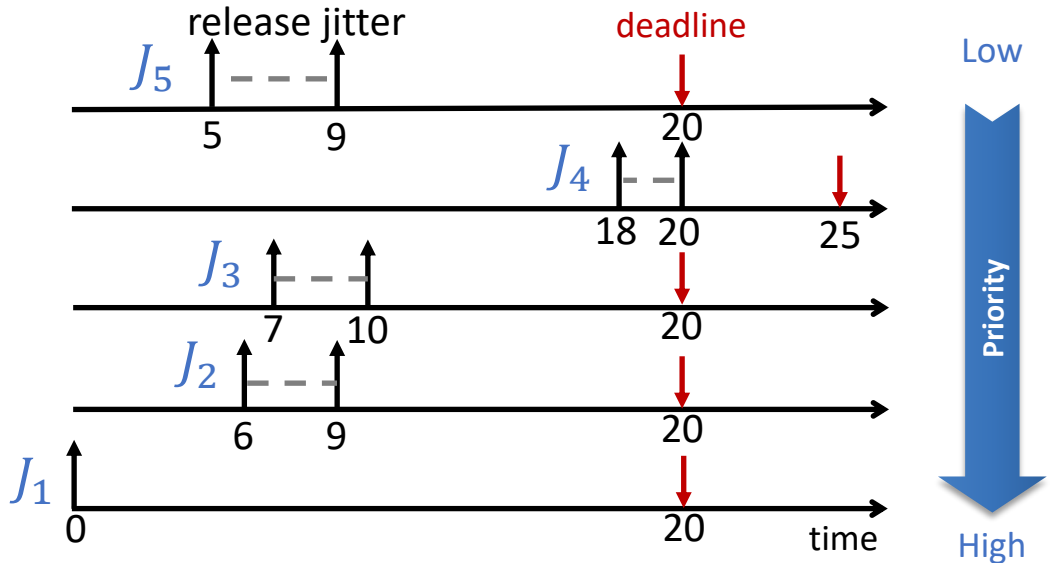


Schedule-abstraction graph

Release jitter Execution time variation

Job	Release time		BCET	WCET	Deadline	Priority
	Earliest	Latest				
J_1	0	0	7	13	20	1
J_2	6	9	1	2	20	2
J_3	7	10	1	2	20	3
J_4	18	20	1	3	25	4
J_5	5	9	1	2	20	5

Non-preemptive

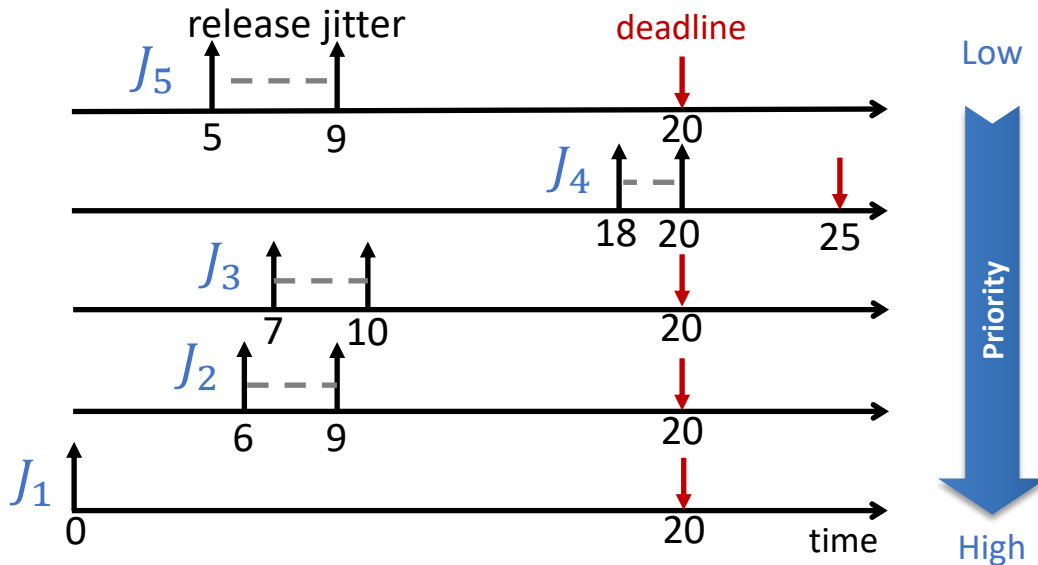


Executed on a single core



Schedule-abstraction graph

Job	Release time		BCET	WCET	Deadline	Priority
	Earliest	Latest				
J_1	0	0	7	13	20	1
J_2	6	9	1	2	20	2
J_3	7	10	1	2	20	3
J_4	18	20	1	3	25	4
J_5	5	9	1	2	20	5

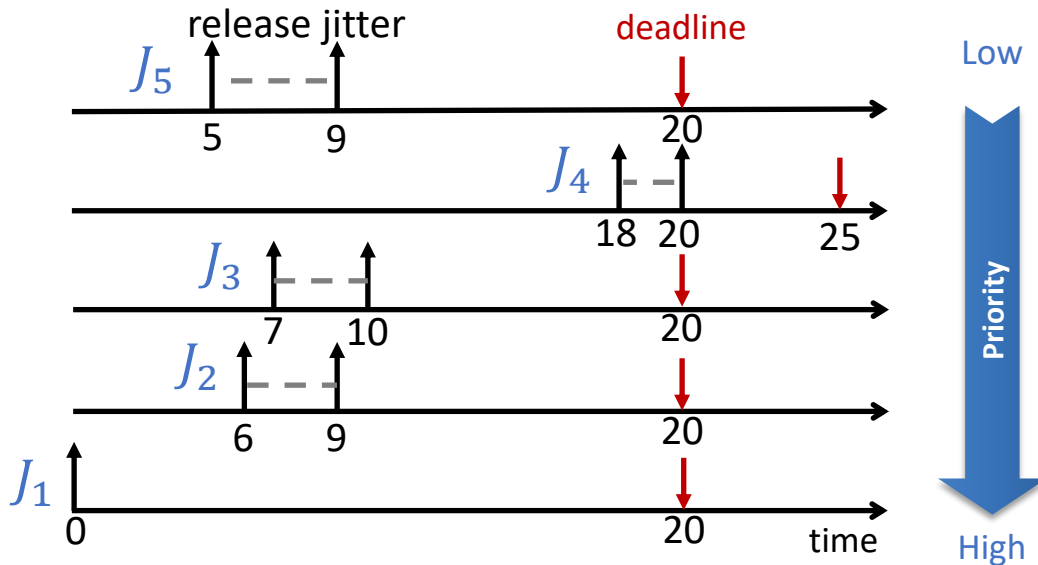
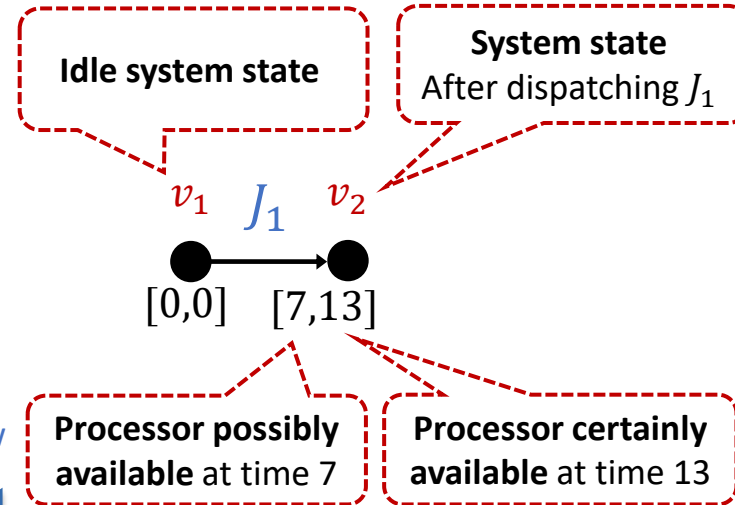


Represent the space of all possible decisions a scheduler can take



Schedule-abstraction graph

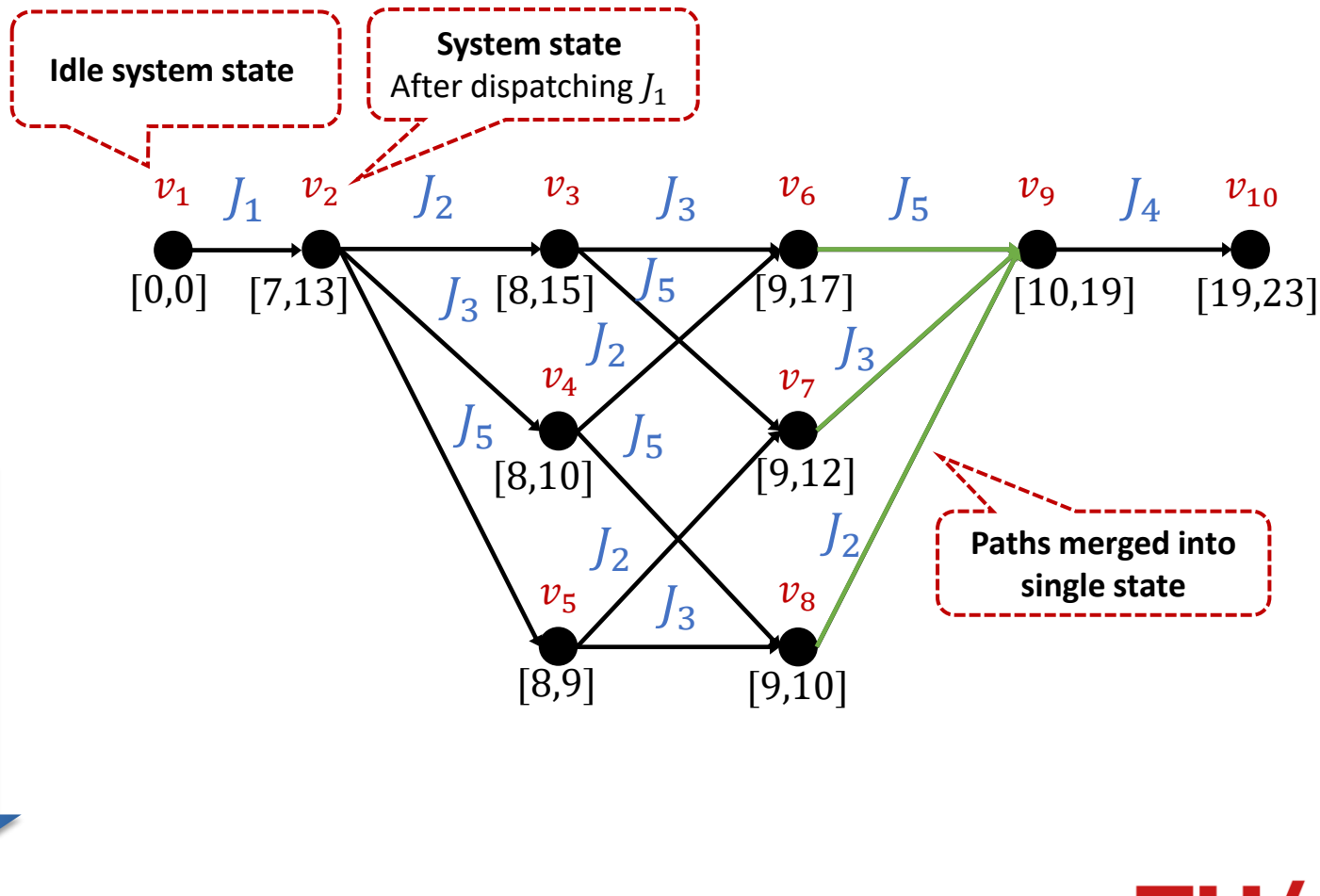
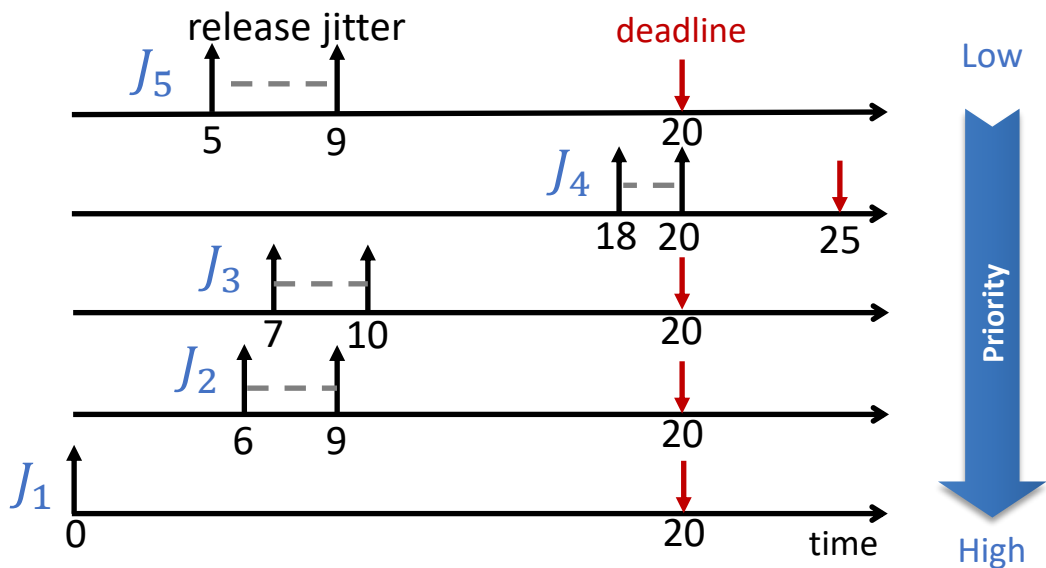
Job	Release time		BCET	WCET	Deadline	Priority
	Earliest	Latest				
J_1	0	0	7	13	20	1
J_2	6	9	1	2	20	2
J_3	7	10	1	2	20	3
J_4	18	20	1	3	25	4
J_5	5	9	1	2	20	5





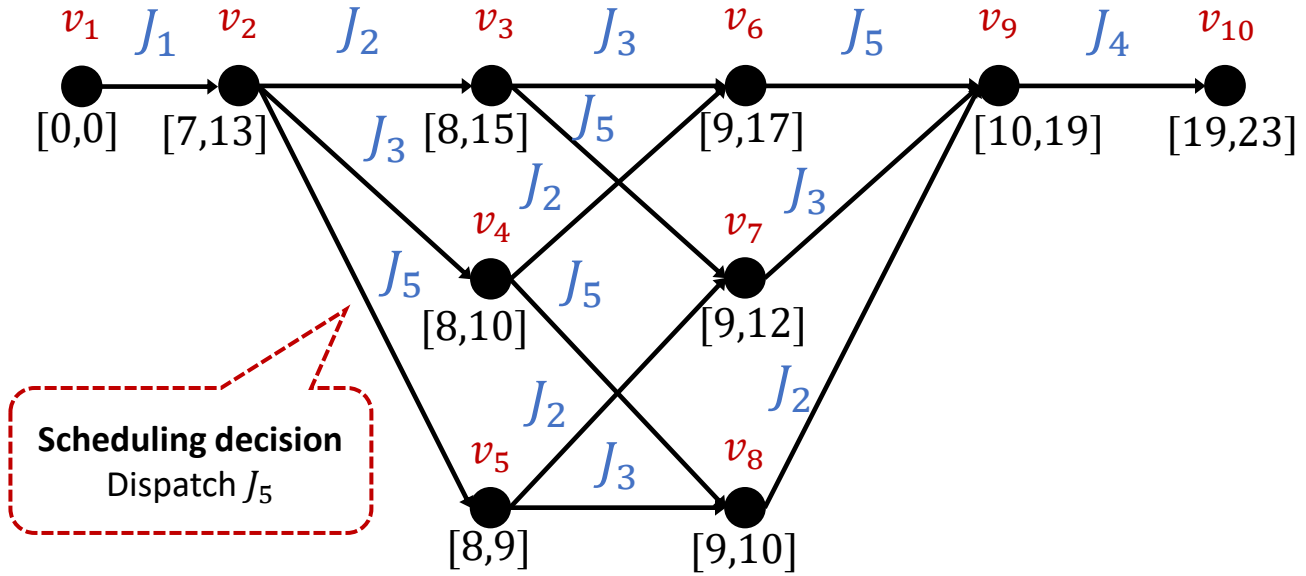
Schedule-abstraction graph

Job	Release time		BCET	WCET	Deadline	Priority
	Earliest	Latest				
J_1	0	0	7	13	20	1
J_2	6	9	1	2	20	2
J_3	7	10	1	2	20	3
J_4	18	20	1	3	25	4
J_5	5	9	1	2	20	5

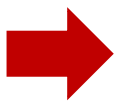




Schedule-abstraction graph



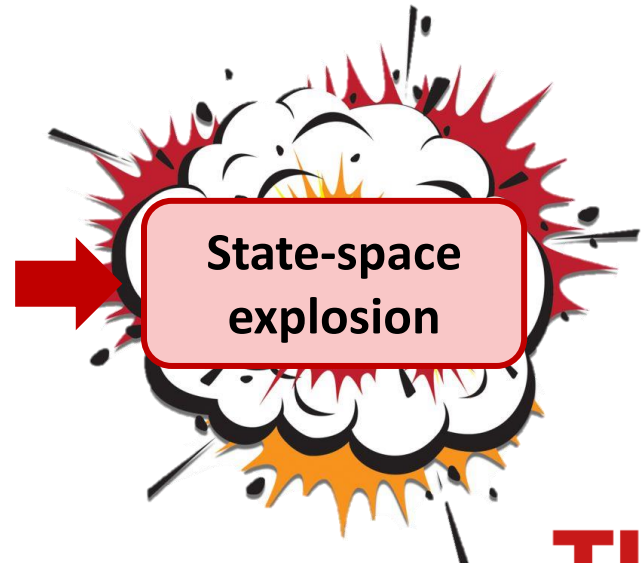
Each edge can only account for a **single** scheduling decision



Combinatorial exploration in case of **timing uncertainties**

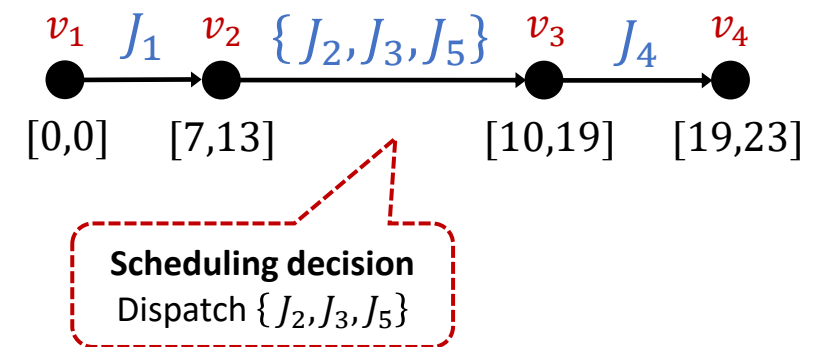
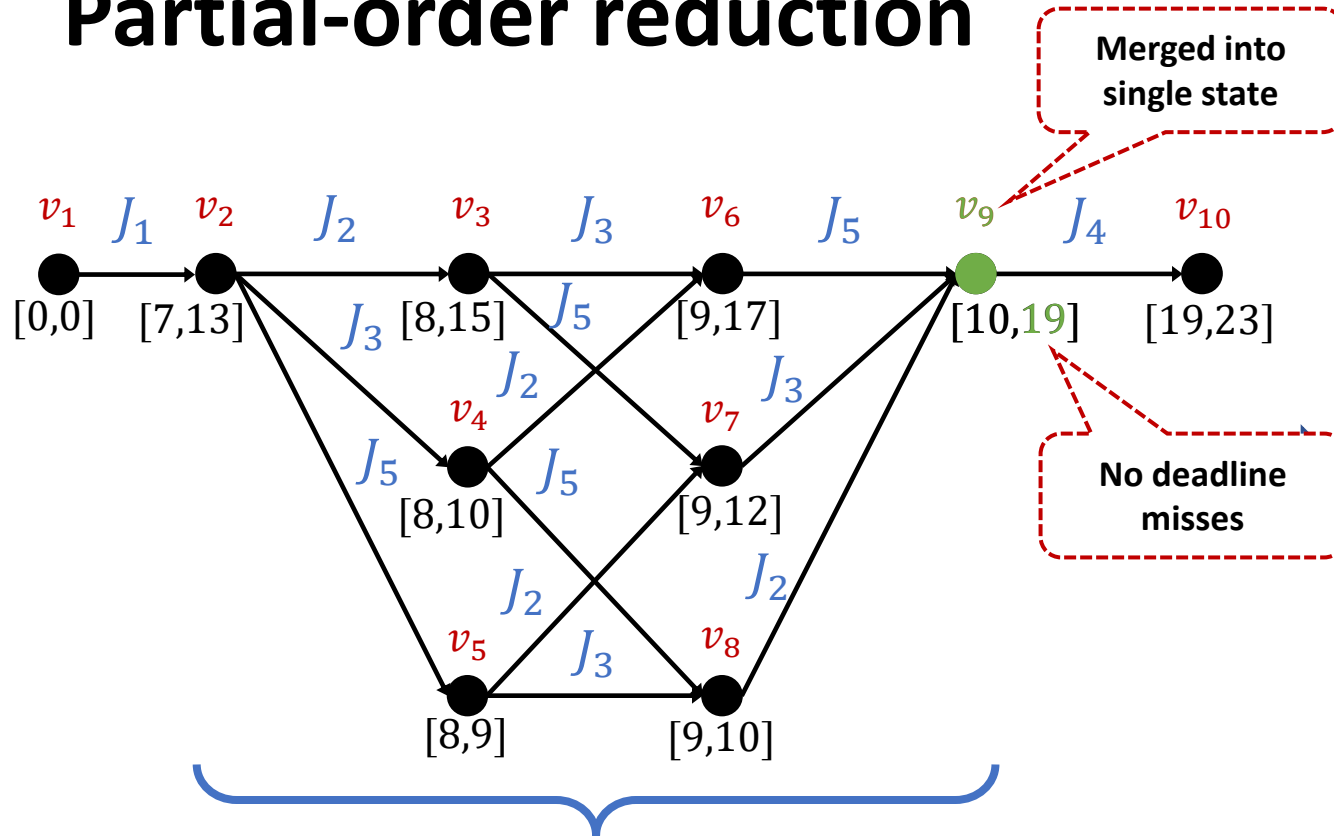


State-space explosion





Partial-order reduction



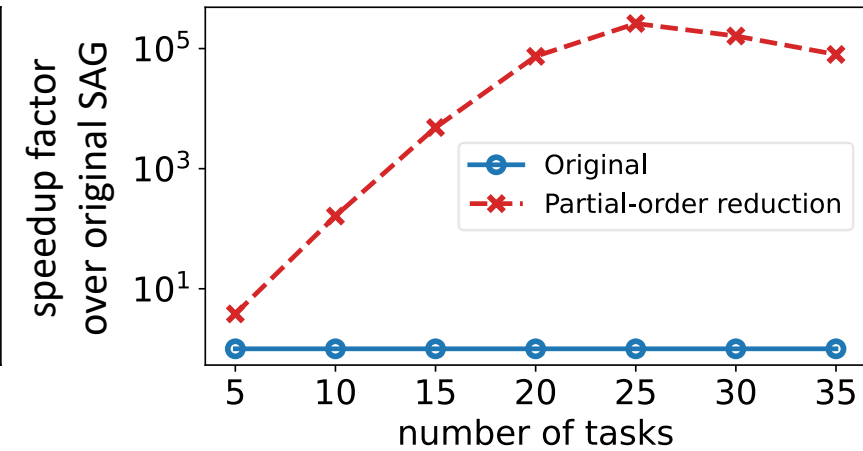
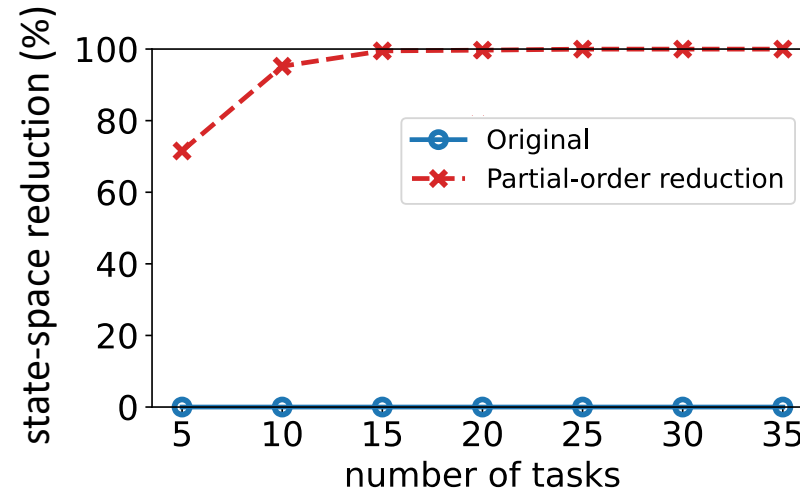
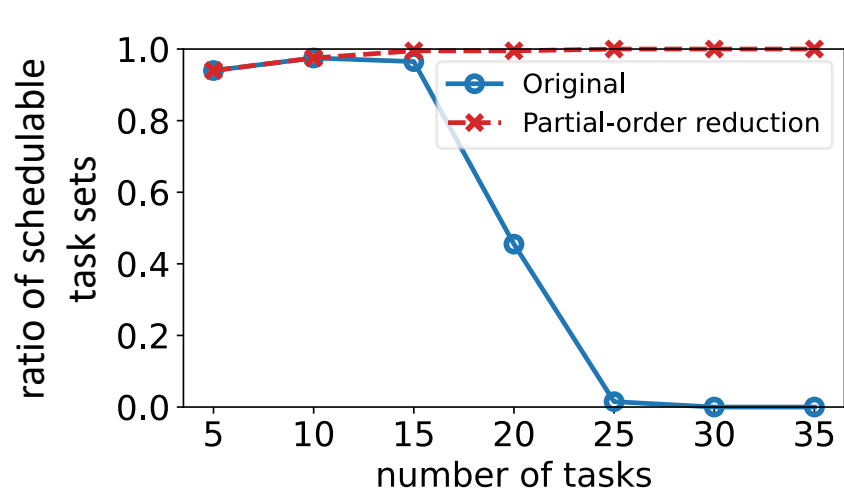
Combinatorial exploration of $\{J_2, J_3, J_5\}$

Not relevant to asserting schedulability

Use partial-order reduction to avoid combinatorial exploration



Empirical evaluation



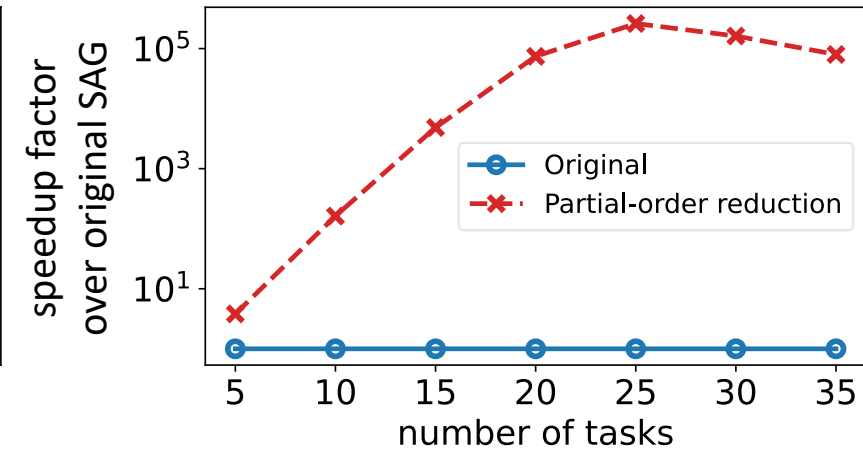
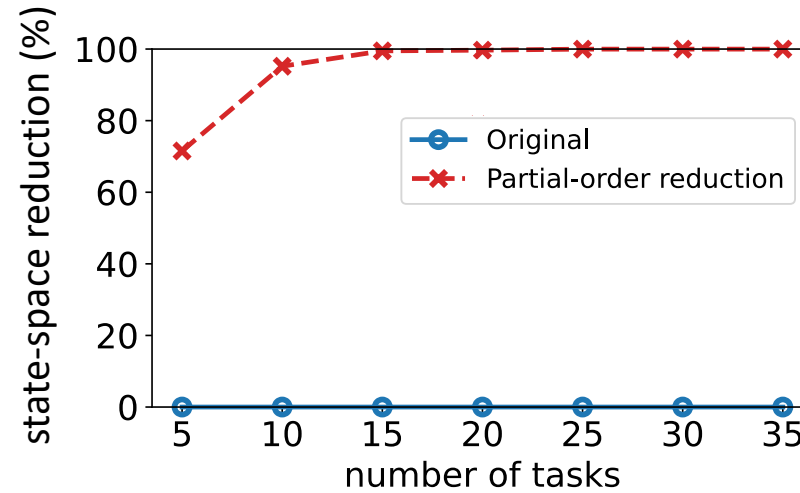
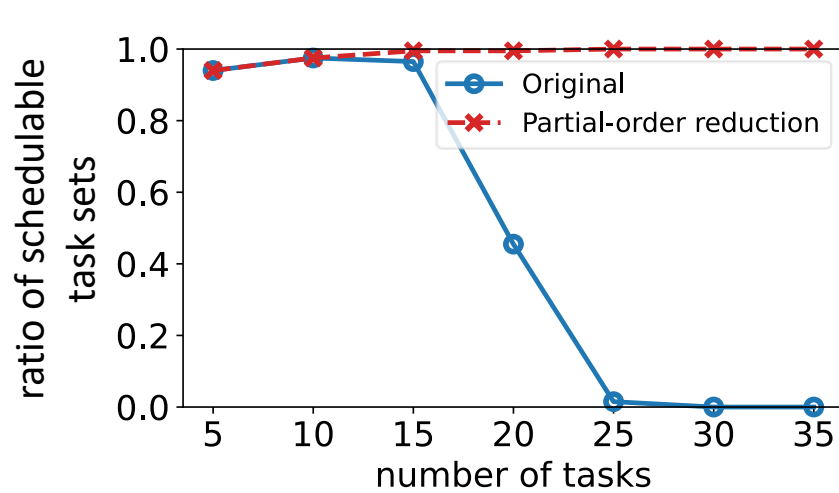
Increased schedulability

State-space reduced by 98% on average

Five orders of magnitude faster on average

Periodic tasks, log-uniform period distribution, release jitter 100 μ s, BCET 0, 30% utilization

Empirical evaluation



Increased schedulability

**State-space reduced by
98% on average**

**Five orders of magnitude
faster on average**

Periodic tasks, log-uniform period distribution, release jitter 100 μ s, BCET 0, 30% utilization