A Simulation Methodology for Worst-Case Response Time Estimation of Distributed Real-Time Systems

Soheil Samii, Sergiu Rafiliu, Petru Eles, Zebo Peng

Embedded Systems Laboratory

Department of Computer and Information Science

Linköpings universitet

Sweden

Outline

- Motivation and background
- Simulation environment
- Example
- Solution overview
- Experiments
- Conclusions

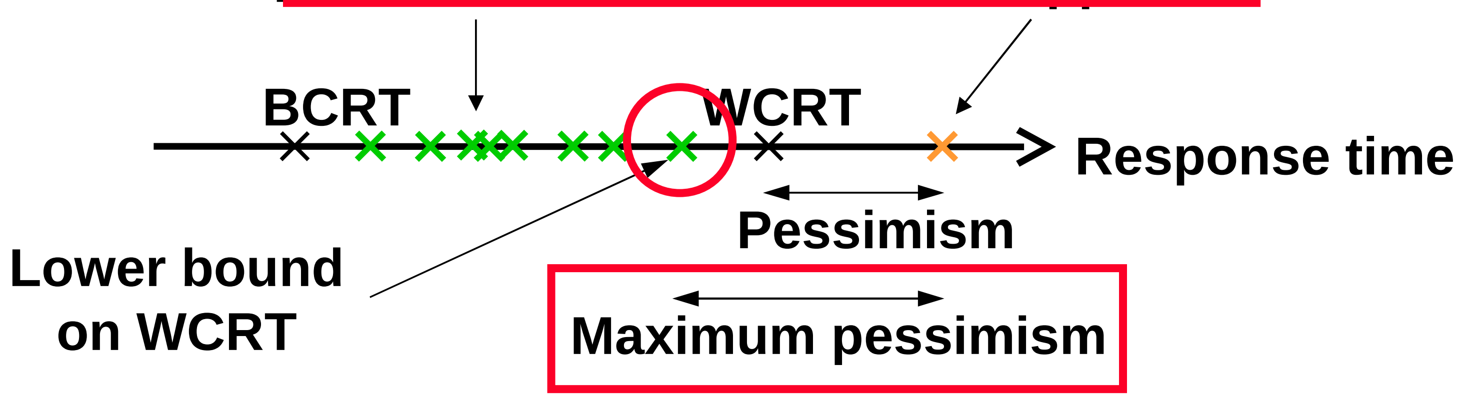
Motivation and background

- Real-time systems: timing characteristics are of interest
- In this paper: worst-case response times (WCRTs) of the processes in the applications
- Analytical methods
  - Pessimistic upper bounds
  - May lead to overdesigned systems and underutilized resources
  - Available only for restricted application models and execution platforms (e.g., communication protocols)

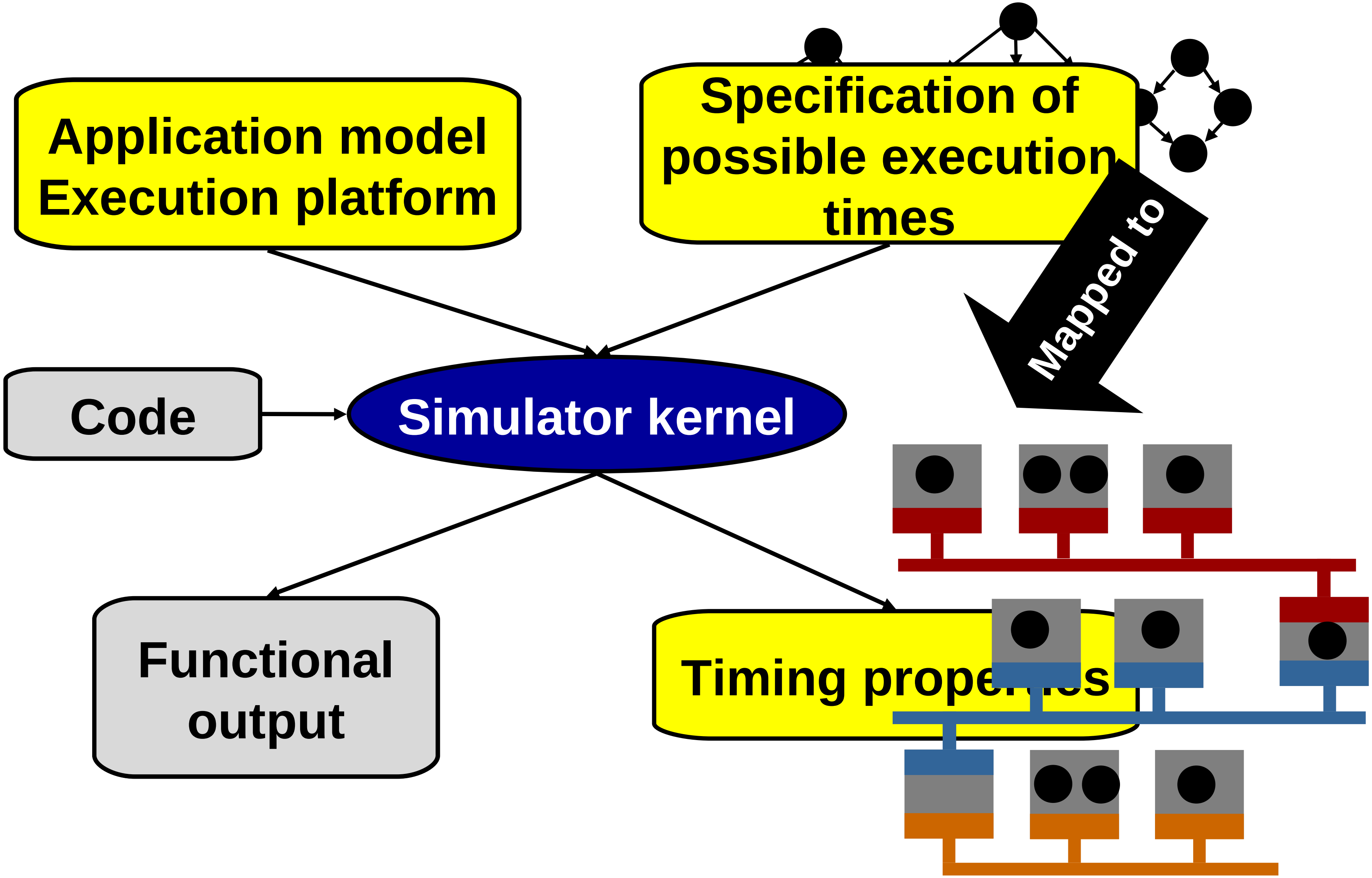
Motivation and background

- Simulation-based approach
  - Practical when no analysis is available
  - Not pessimistic (but optimistic lower bounds)
    - Avoid overdesign
    - C
    - V ... .
      p
      I
      f ... und on WCRT

How to drive the simulator towards WCRT?

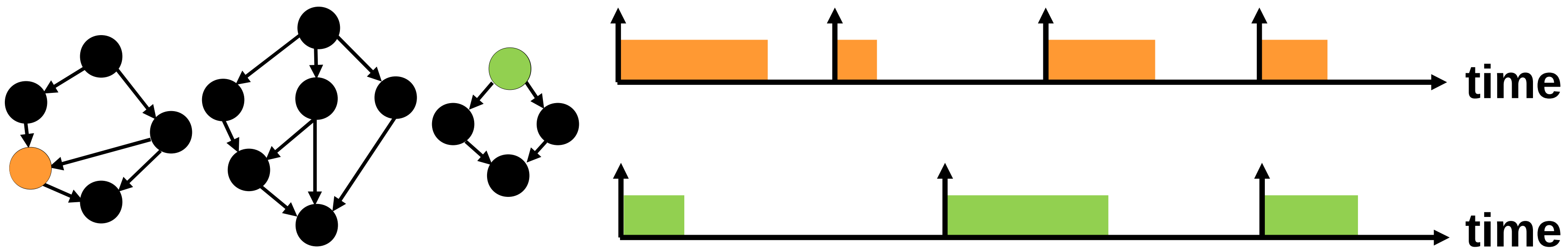


Simulation environment



Soheil Samii, Sergiu Rafiliu, Petru Eles, Zebo Peng

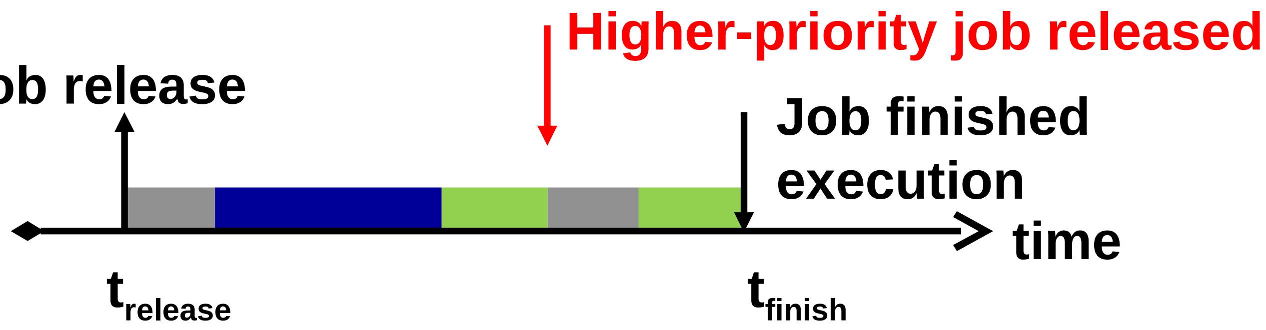
Application model



- Jobs are released at certain moments in time
  - Periodic release
- A job has an execution time
  - Execution time in [BCET, WCET]

Soheil Samii, Sergiu Rafiliu, Petru Eles, Zebo Peng

Response time



Response time = t_{finish} - $t_{release}$

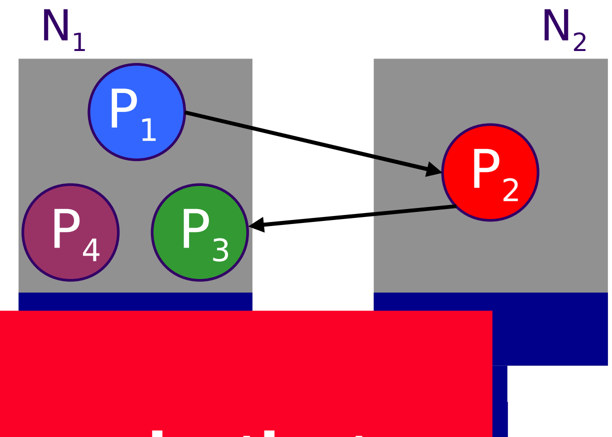
- **Response time of a job (of a process)**
  - **Its execution time**
  - **Execution of higher-priority jobs**
  - **Time to wait for messages (communication delay)**

Observations

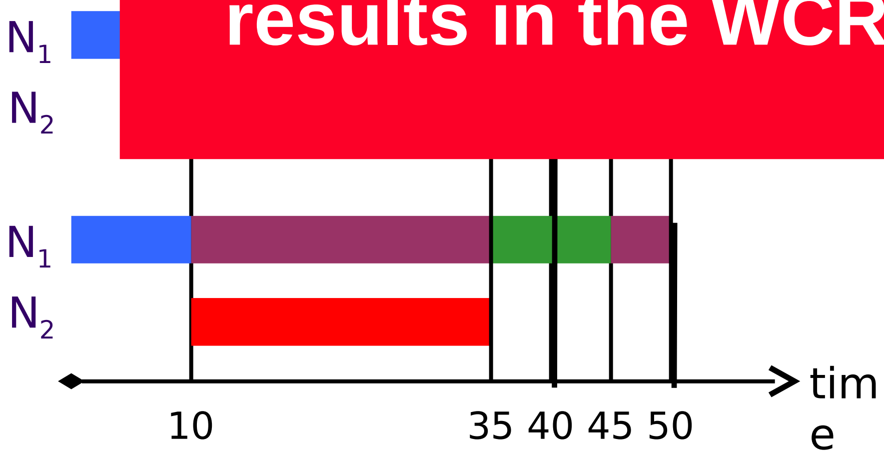
- **The number of execution scenarios is huge**
  - **Most of them do not lead to the WCRT**
- **The scenario where all jobs execute for their WCET does not necessarily produce the WCRT**

Example

- C_{P1}=10, C_{P2} in [25, 35], C_{P3}=10, C_{P4}=30 (execution times)
- P4 has lowest priority
- Instantaneous communication



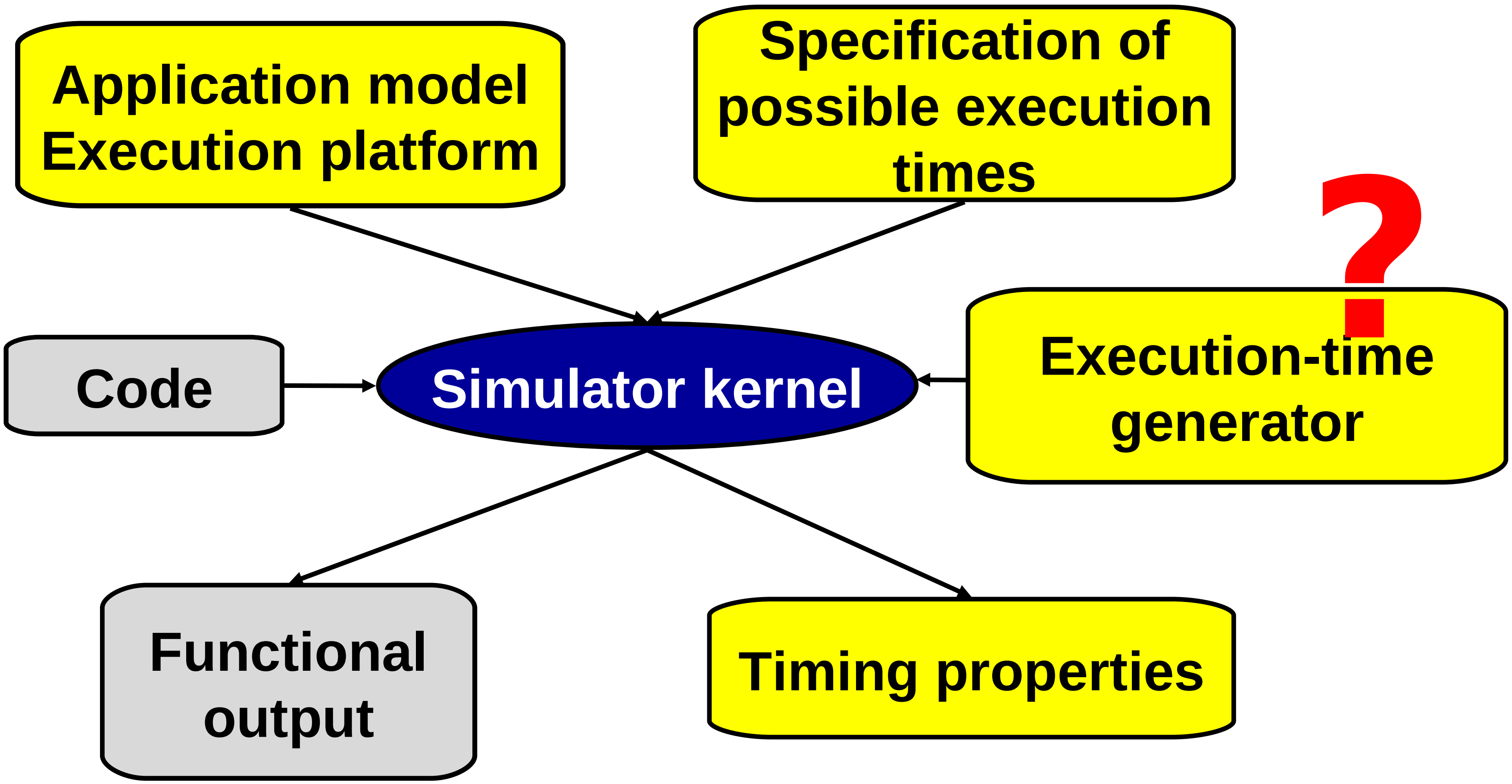
How to produce the scenario that results in the WCRT of a process?



C_{P2}=25 ⇒ R_{P4}=50

Scheduling anomaly

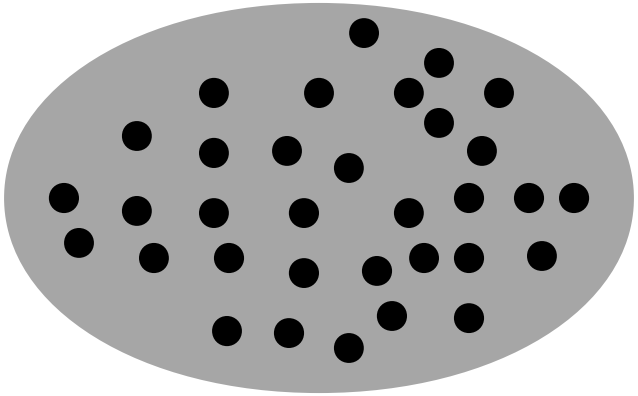
Simulation environment



Solution overview

- Choose between all points in [BCET, WCET]
- Intelligently reduce the execution time candidates to a discrete set

Execution-time space

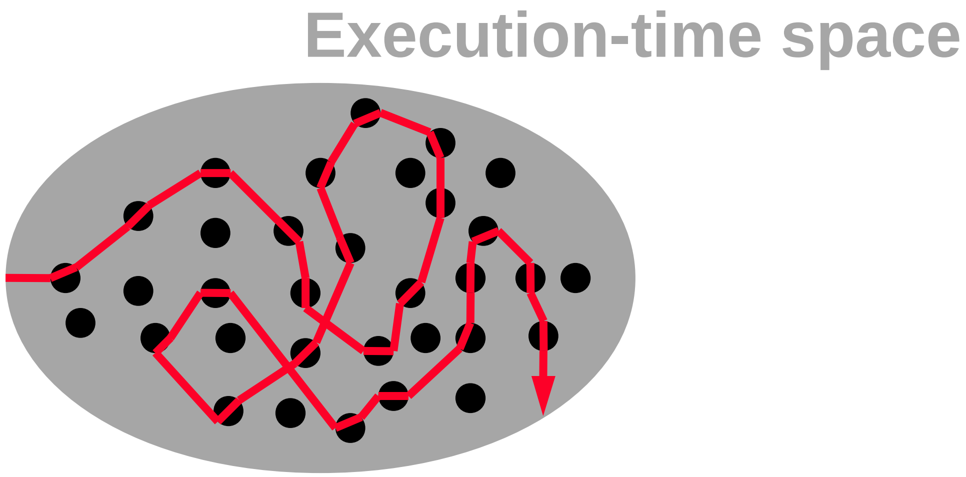


Reduced execution-time space

Reduced space cannot be simulated in affordable time

Solution overview

- **How to explore the reduced execution-time space to reach a good solution?**

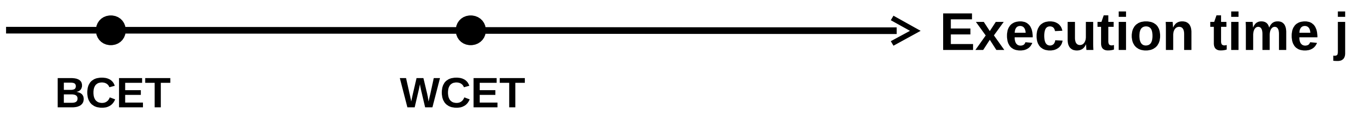


Reduced execution-time space

1. **Execution-time space reduction**
2. **Execution-time space exploration**

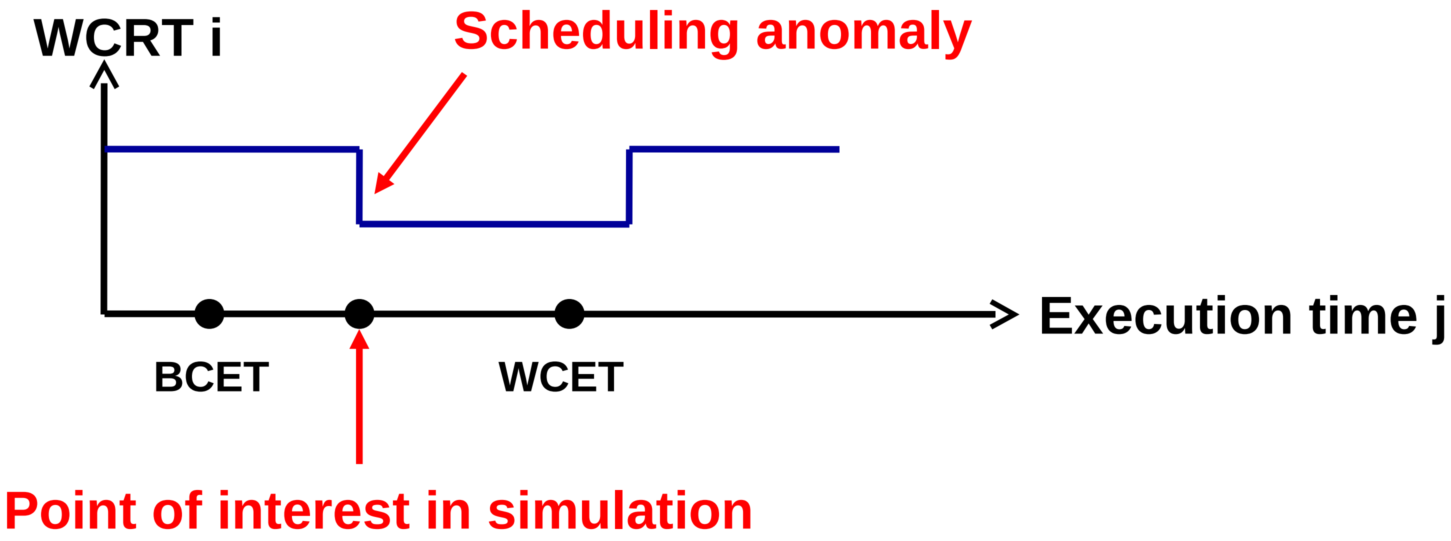
Execution-time space reduction

- Corner-case reduction (CC)
  - For each job, choose either the BCET or the WCET
  - Intuition and experiments: extreme cases produce usually large response times



Execution-time space reduction

- Improved corner-case reduction (ICC)
  - Find additional points (related to scheduling anomalies)
  - Analysis by Racu and Ernst (RTAS'06)



Execution-time space exploration

- How do we choose job execution times at a given point during simulation?

- Random exploration
  - Initial space of execution times
    - Choose randomly
  - Corner-cases (CC) and improved corner-cases (ICC)
    - Randomly
    - Intuition and experiments: more towards WCET

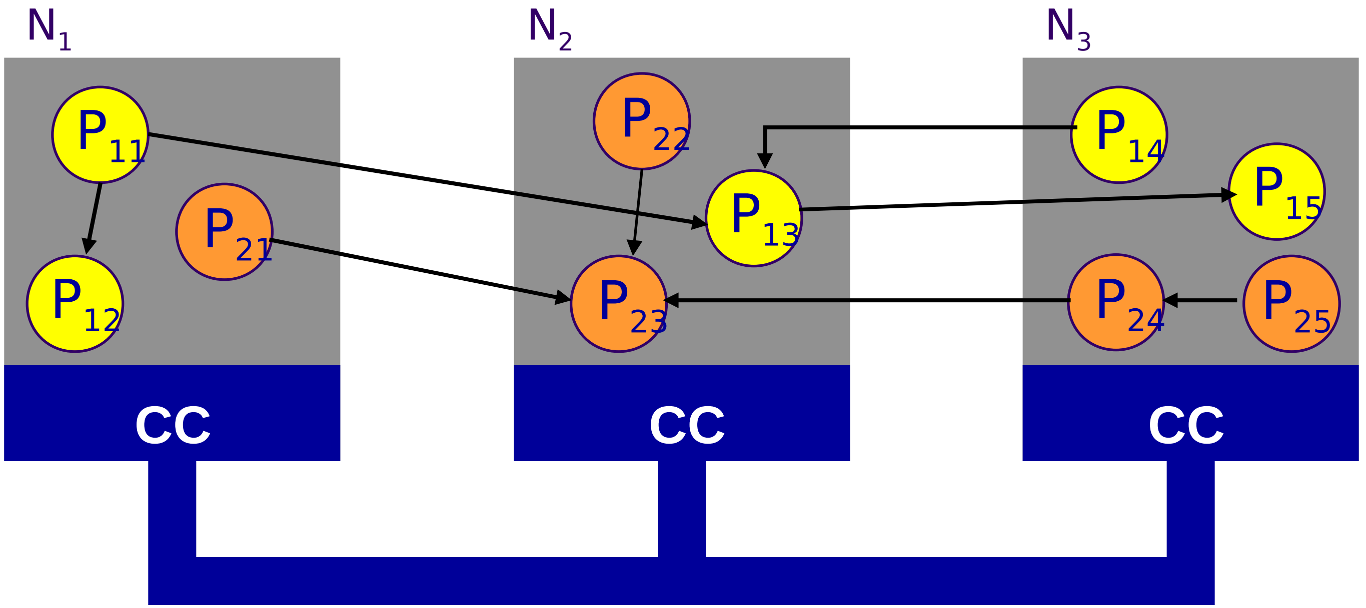
Execution-time space exploration

- Optimization problem
  - Cost function: The response time of a process
    - Given by the simulator
  - Variables: job execution times
    - Execution-time generator
- Genetic algorithm-based exploration
  - Developed for CC and ICC

Summary of approaches

	Initial	CC	ICC
Random	R-Initial	R-CC	R-ICC
GA	-	GA-CC	GA-ICC

Experiments: System architecture



Processes

- Execution time in [BCET,WCET]
- Jobs released periodically
- Priority-based scheduling

Messages

- CAN: message priorities
- FlexRay: TDMA + dynamic segment

Experiments

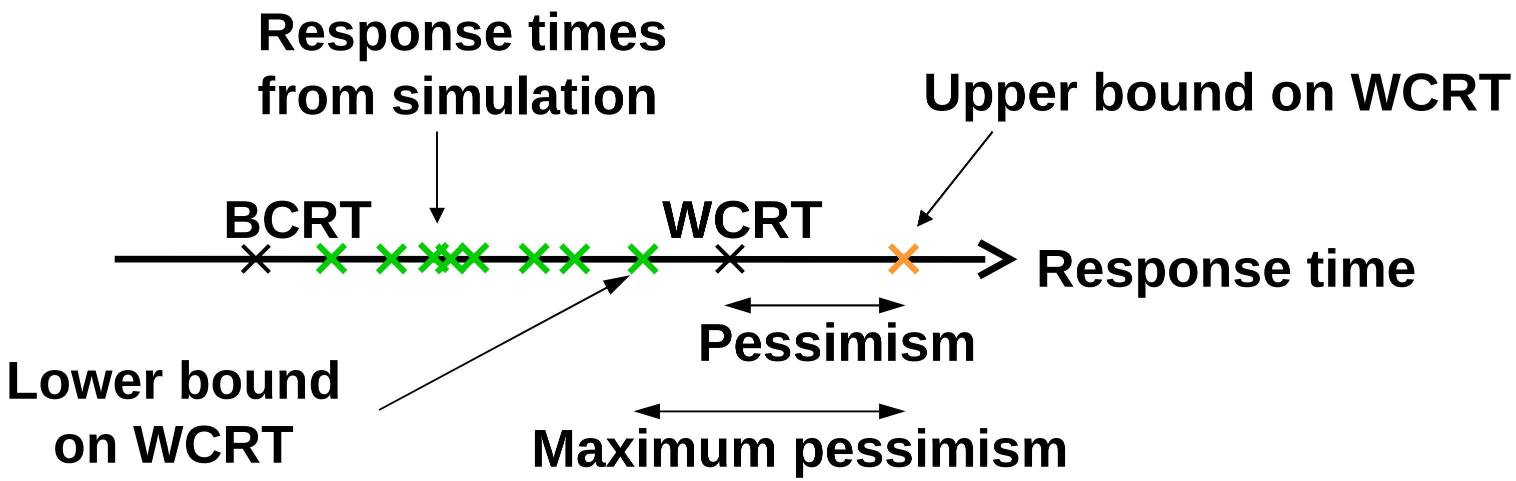
- Compare the approaches with respect to producing large response times
  - Generated applications with varying timing characteristics and varying data dependency structures
- Reference point: in-house analysis tool (WCRT is unknown)
  - Ratio = R_sim / R_analysis
- For each approach:
  - Average ratio
  - Number of times the approach found the best solution (among all approaches)

Experiments

Approach	Average ratio [%]	Frequency [%]
R-Initial	77.6	0
R-CC	87.3	30
R-ICC	87.4	32.9
GA-CC	88.0	41.4
GA-ICC	90.5	97.1
Only WCET	83.7	0

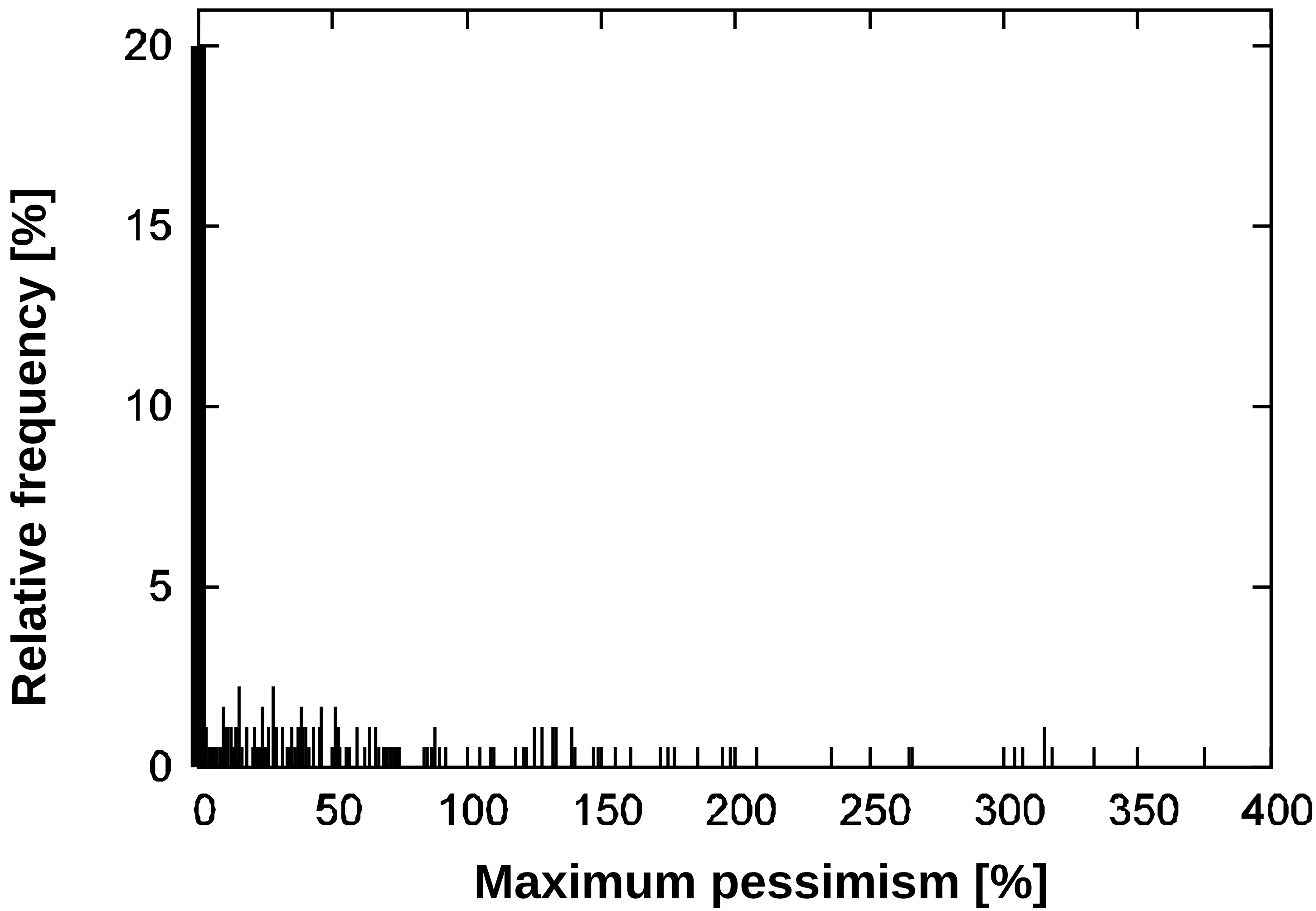
- **All approaches have run for the same amount of time**
  - **Up to 10 minutes**
  - **On average: 100 seconds**

Experiments: Pessimism estimation



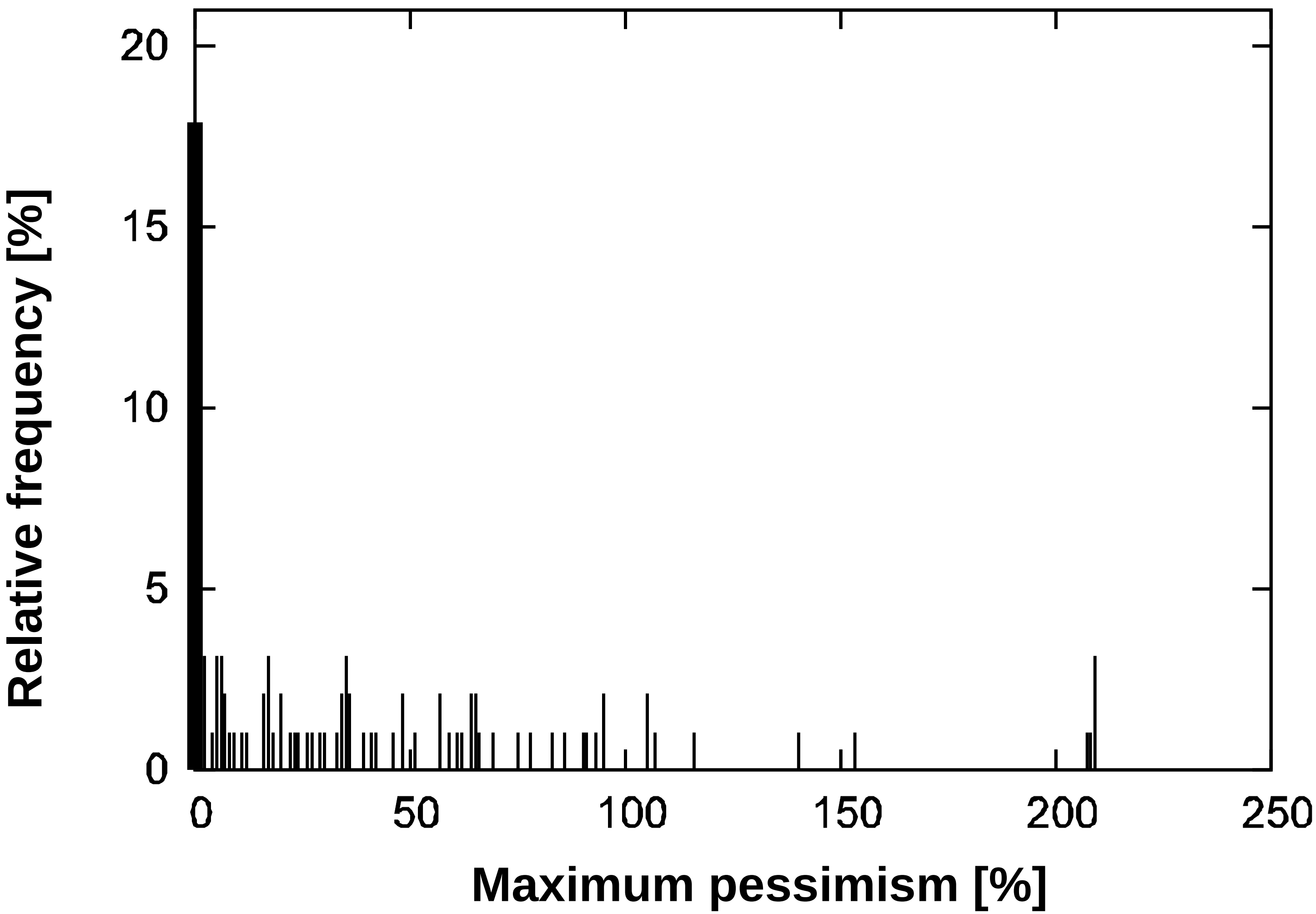
- Maximum pessimism = (R_analysis – R_sim) / R_sim
- CAN- and FlexRay-based systems

Pessimism estimation - CAN



Soheil Samii, Sergiu Rafiliu, Petru Eles, Zebo Peng

Pessimism estimation - FlexRay



Conclusions

- Simulation methodology for WCRT estimation of distributed real-time systems
  - Reduce the space of execution times
  - Efficient exploration strategy
- Useful approach:
  - No analysis tool available
  - Avoid overdesign when deadline misses can be tolerated
  - Validate a timing analysis

A Simulation Methodology for Worst-Case Response Time Estimation of Distributed Real-Time Systems

Soheil Samii, Sergiu Rafiliu, Petru Eles, Zebo Peng

Embedded Systems Laboratory

Department of Computer and Information Science

Linköpings universitet

Sweden

Case study

- Automotive cruise-controller application: 28 processes mapped to 5 computation nodes
- Analyzed 2 processes that produce the control data

- CAN implementation
  - 35.2% and 8.5% pessimism
- FlexRay implementation
  - 39.6% and 6.7% pessimism

- Pessimism relatively small → the implementations are tight and cost efficient