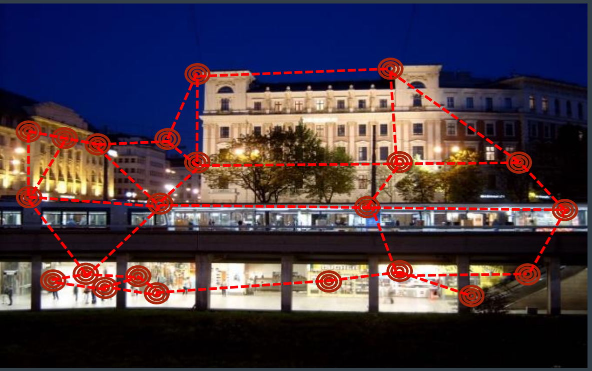


Digital Evil Twins

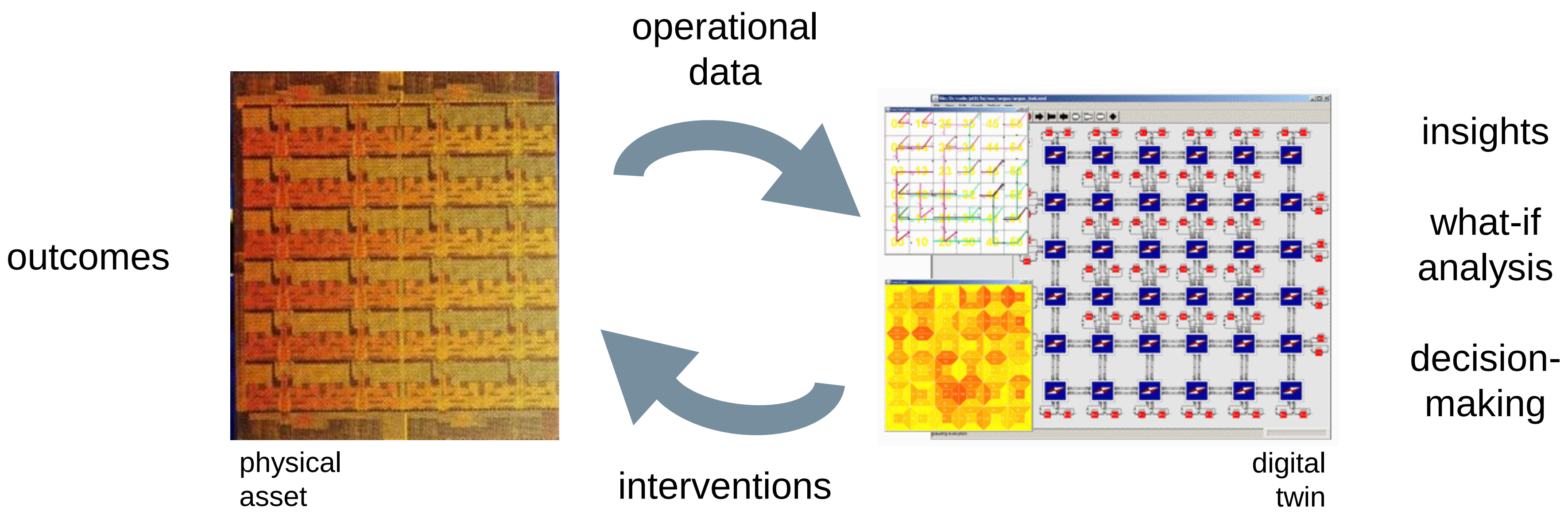
Identification of worst-case behaviours in real-time systems

Leandro Soares Indrusiak

Real-Time and Distributed Systems Group
Department of Computer Science
University of York

Digital Twin

- Virtual representation of an object or system that spans its lifecycle
    - updated from real-time data
    - uses simulation, machine learning and reasoning to help decision making



operational data

outcomes

insights

what-if analysis

decision-making

physical asset

interventions

digital twin

Evil Twin

- Common plot device used (and maybe overused) in fictional narratives where one of the characters is the evil identical twin, clone or counterpart of another character

Digital Evil Twin

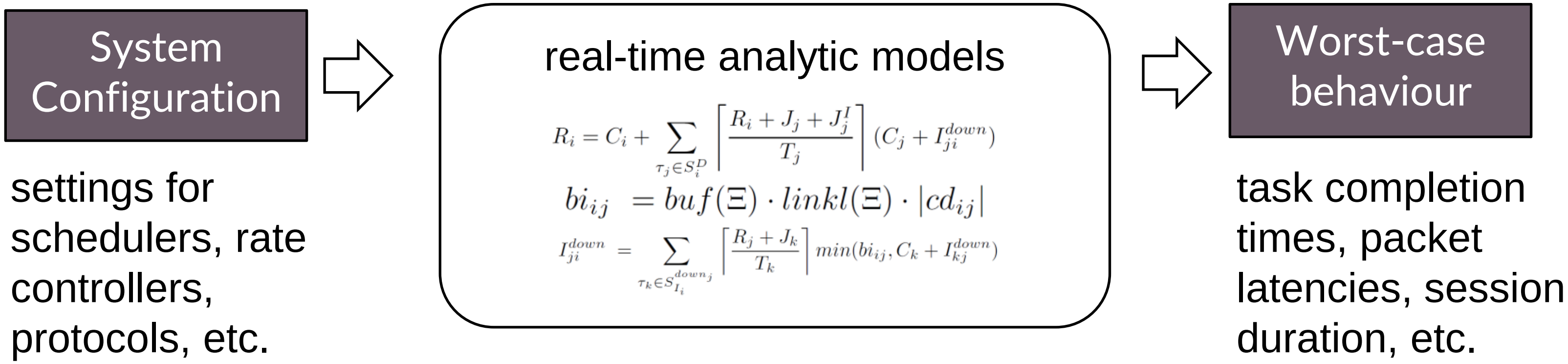
- Digital twin representing the **worst-case behaviour** of the physical object or system
  - performance, energy dissipation, temperature, etc.
  - useful for performance-sensitive systems
    - stress testing
    - load admission control
    - safety bounds
    - optimisation
  - regular digital twins are not suitable for the job
    - higher complexity
    - focus on actual behaviour
      - worst-case behaviour may have not yet been experienced by physical system

Outline

- Digital Evil Twins ✓
- Identification of worst-case behaviours using real-time analysis
  - guiding optimisation
- Semi-automatic synthesis of worst-case real-time analysis models
- Open areas of research

Real-Time Analysis

- Family of analytical models aiming to establish if a system meets its timing requirements even in the worst-case scenario
  - based on decades of research by the real-time systems community
  - e.g. response-time analysis, real-time calculus, network calculus



System Configuration

settings for schedulers, rate controllers, protocols, etc.

real-time analytic models

$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down})$$

$$bi_{ij} = buf(\Xi) \cdot linkl(\Xi) \cdot |cd_{ij}|$$

$$I_{ji}^{down} = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil min(bi_{ij}, C_k + I_{kj}^{down})$$

Worst-case behaviour

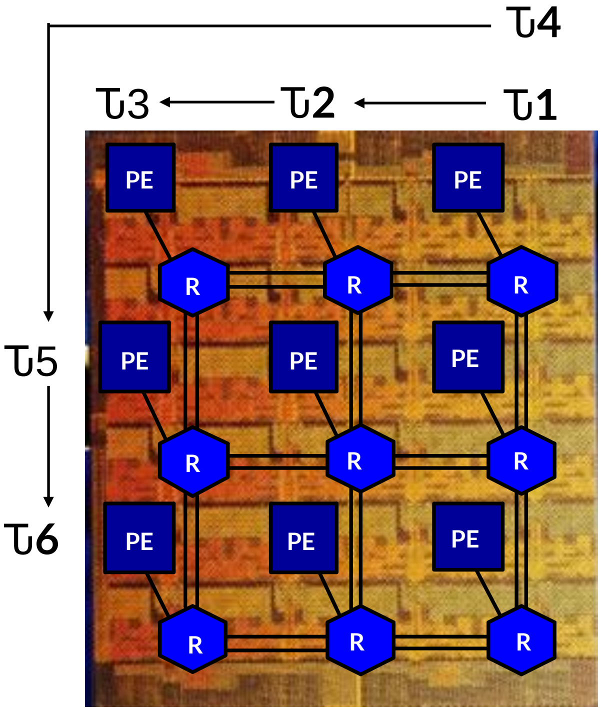
task completion times, packet latencies, session duration, etc.

Real-time analysis ≠ real-time analytics

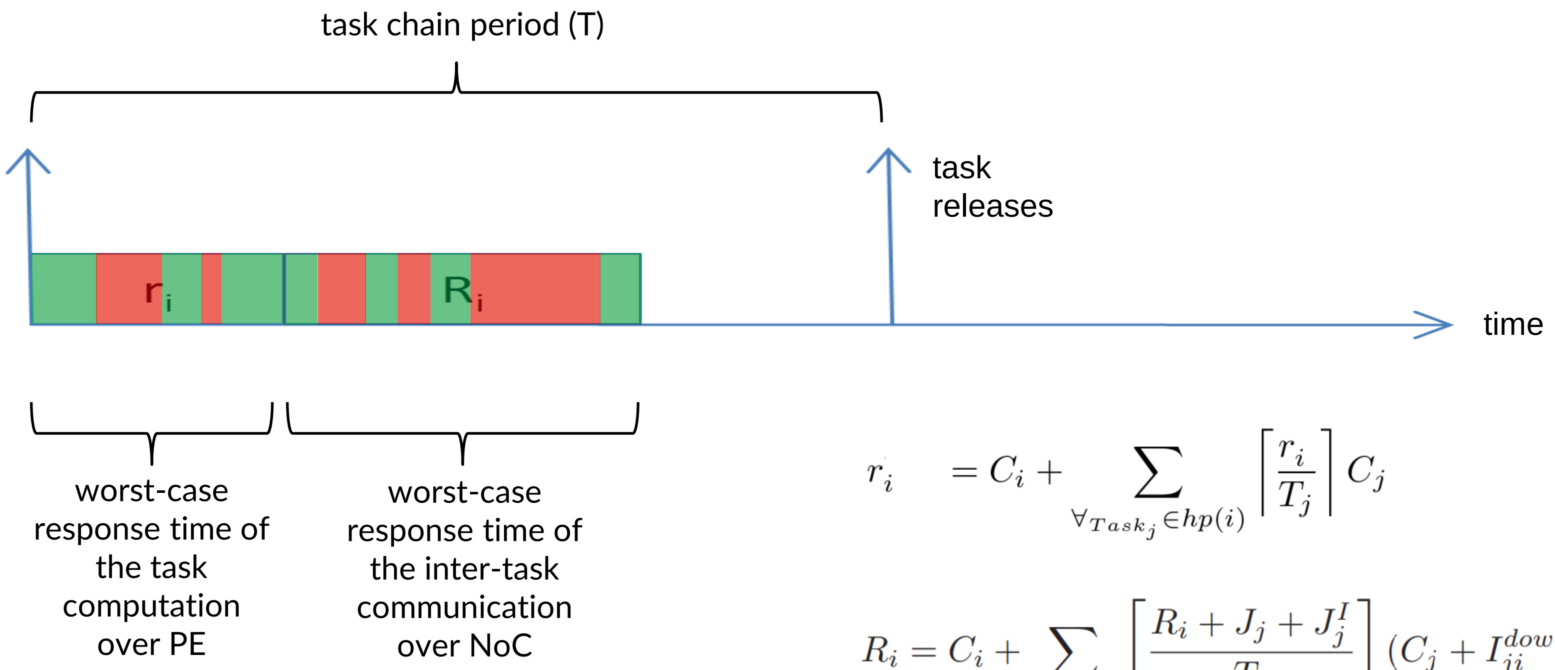
Real-Time Analysis

- Example: chains of software tasks running over multiple processing elements of a network-on-chip interconnect

- System configuration
  - wormhole switching
  - XY routing
  - priority-preemptive arbitration
  - credit-based flow control
  - sporadic tasks



Real-Time Analysis


task chain period (T)
task releases
r_i
R_i
time
worst-case response time of the task computation over PE
worst-case response time of the inter-task communication over NoC

$$r_i = C_i + \sum_{\forall_{Task_j} \in hp(i)} \left\lceil \frac{r_i}{T_j} \right\rceil C_j$$

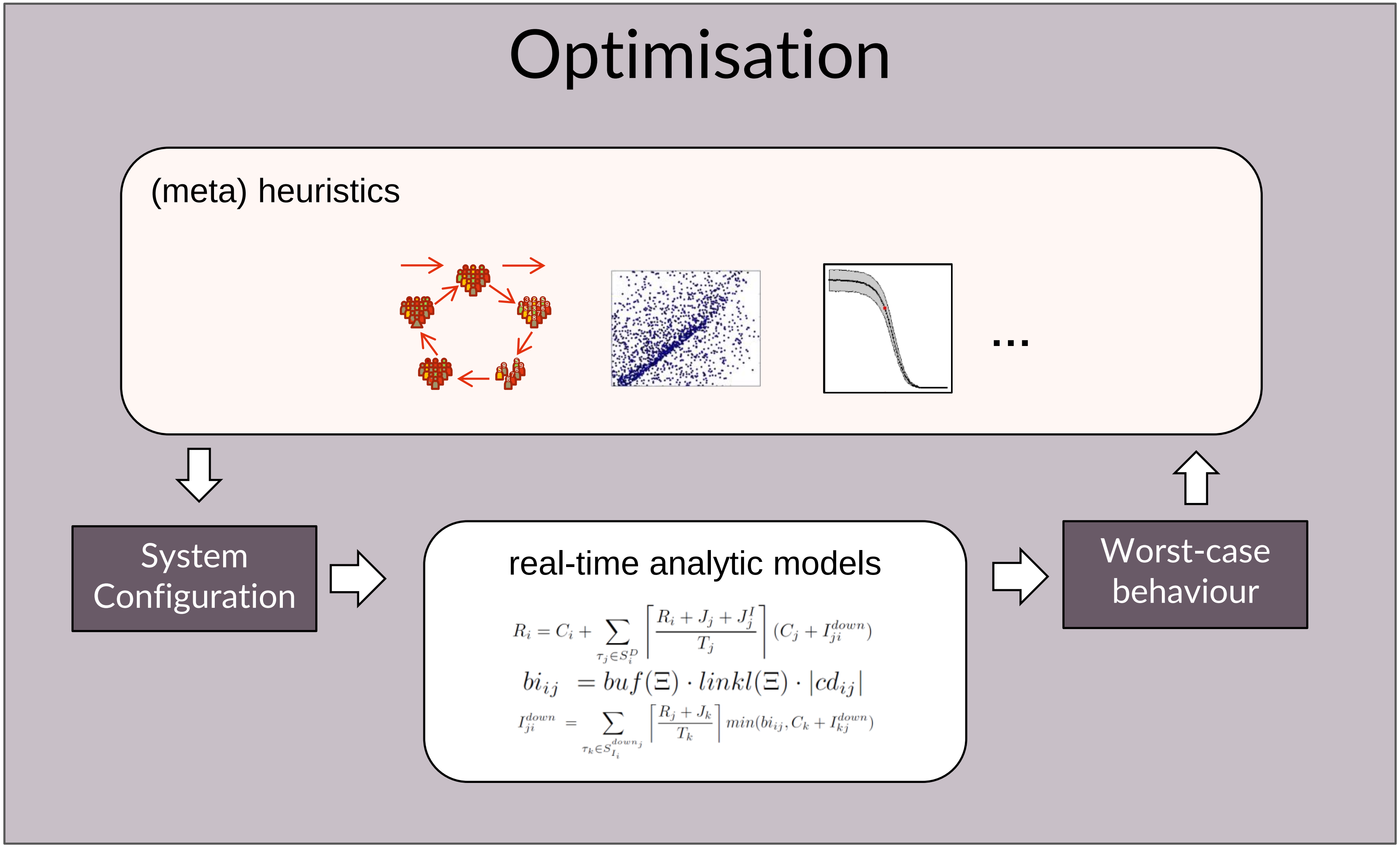
$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down})$$

- Worst-case end-to-end behaviour of each task of the chain is $r_i + R_i$
- Can be used to ensure that each task of the chain will always have all the data it needs by the time it starts its execution

L.S. Indrusiak, “End-to-End Schedulability Tests for Multiprocessor Embedded Systems based on Networks-on-Chip with Priority-Preemptive Arbitration”, Journal of Systems Architecture, v. 60, n. 7, Aug 2014.

L.S. Indrusiak, A. Burns, B. Nikolic, “Buffer-aware bounds to multi-point progressive blocking in priority-preemptive NoCs”, in Design Automation and Test in Europe (DATE), 2018. (Best paper award)

Optimising worst-case behaviour



Optimisation

(meta) heuristics

...

System Configuration

real-time analytic models

$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down})$$

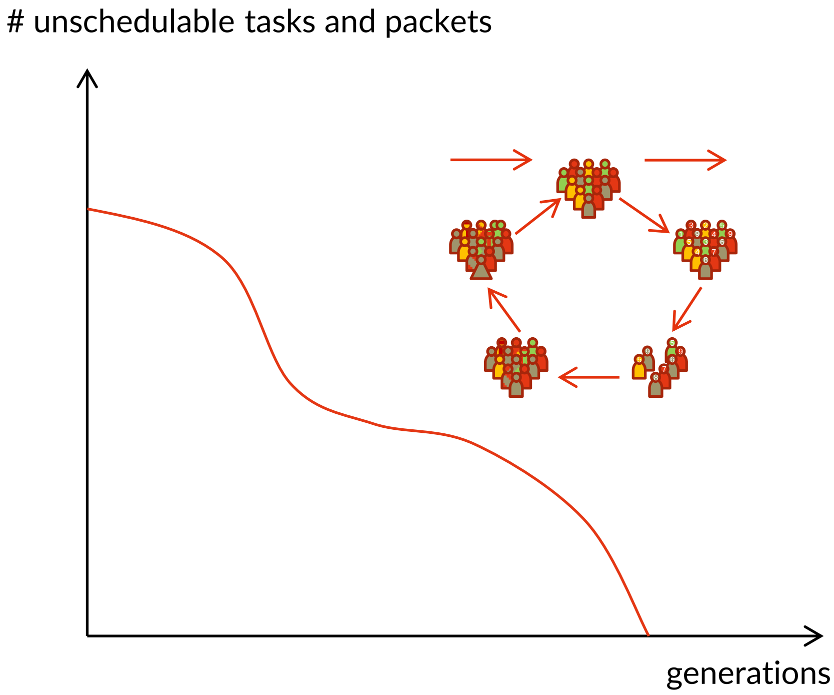
$$bi_{ij} \;\; = buf(\Xi) \cdot linkl(\Xi) \cdot |cd_{ij}|$$

$$I_{ji}^{down} \;\; = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil min(bi_{ij}, C_k + I_{kj}^{down})$$

Worst-case behaviour

Optimising worst-case behaviour

- task allocation for performance (ReCoSoC 2011, 2012, ACM Computing Surveys 2017, Leibniz Trans Emb Sys 2017), energy dissipation (ISVLSI 2013, RTNS 2013) and security (ReCoSoC 2017, Microprocessors & Microsystems 2019)
- memory partitioning (PDP 2018)
- multi-mode operation (RTNS 2015, ISORC 2016, EURASIP JES 2017)
- priority assignment (PDP 2015)
- interconnect routing and topology (under review)



L.S. Indrusiak, P. Dziurzanski, A. K. Singh, Dynamic Resource Allocation in Embedded, High-Performance and Cloud Computing, River Publishers, 2016.

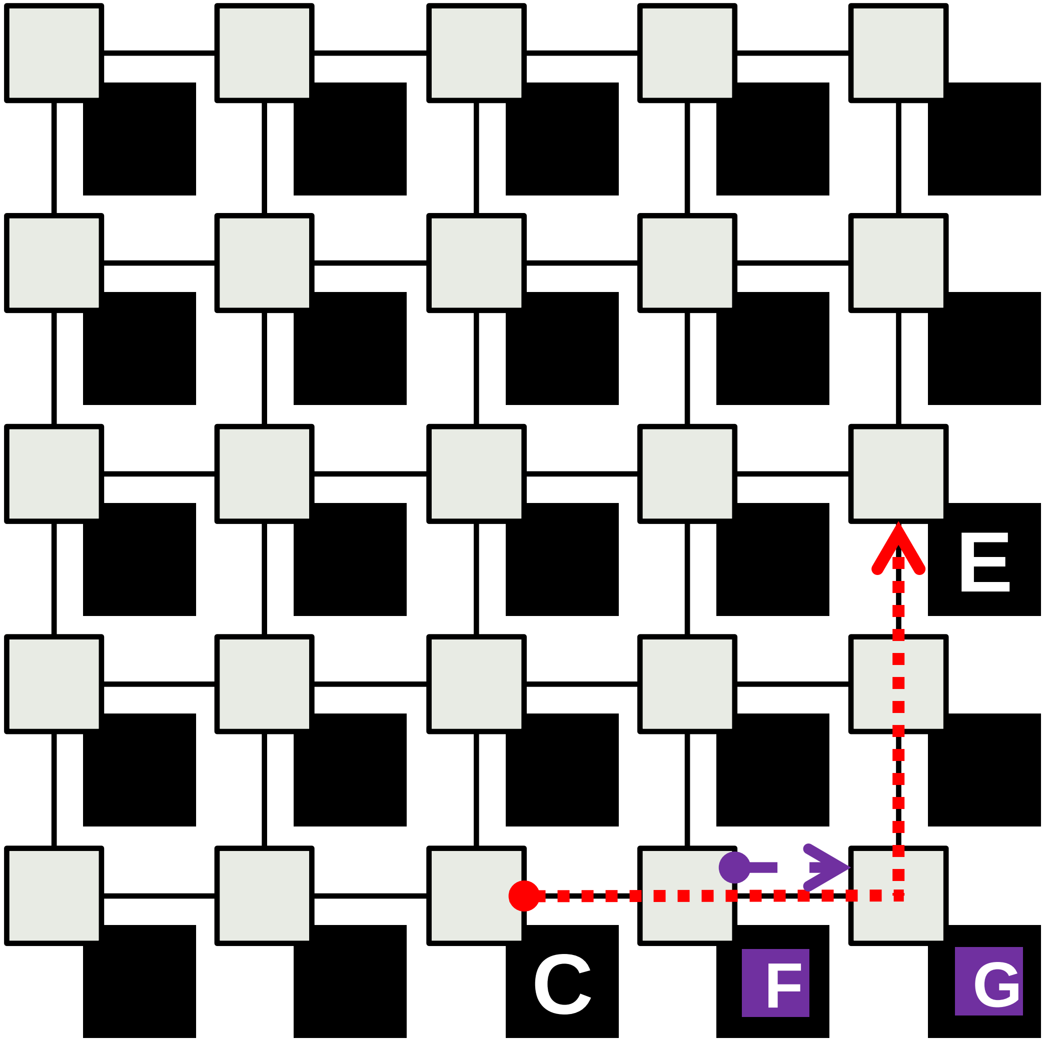
Resilience against timing attacks

- Secure network-on-chip supporting hard real-time guarantees
  - packet flows must meet their deadlines even in worst-case scenarios
  - packet flows may carry sensitive information
  - vulnerable to timing attacks, i.e. side channel attacks where the timing of packets can be correlated to the sensitive information they carry

- Clear trade-off between
  - time predictability
  - timing attack resilience

L.S. Indrusiak, J. Harbin, C. Reinbrecht, J. Sepulveda, “Side-channel protected MPSoC through secure real-time networks-on-chip”, Microprocess. Microsystems, v.68, p. 34-46, 2019.

Example



Route randomisation

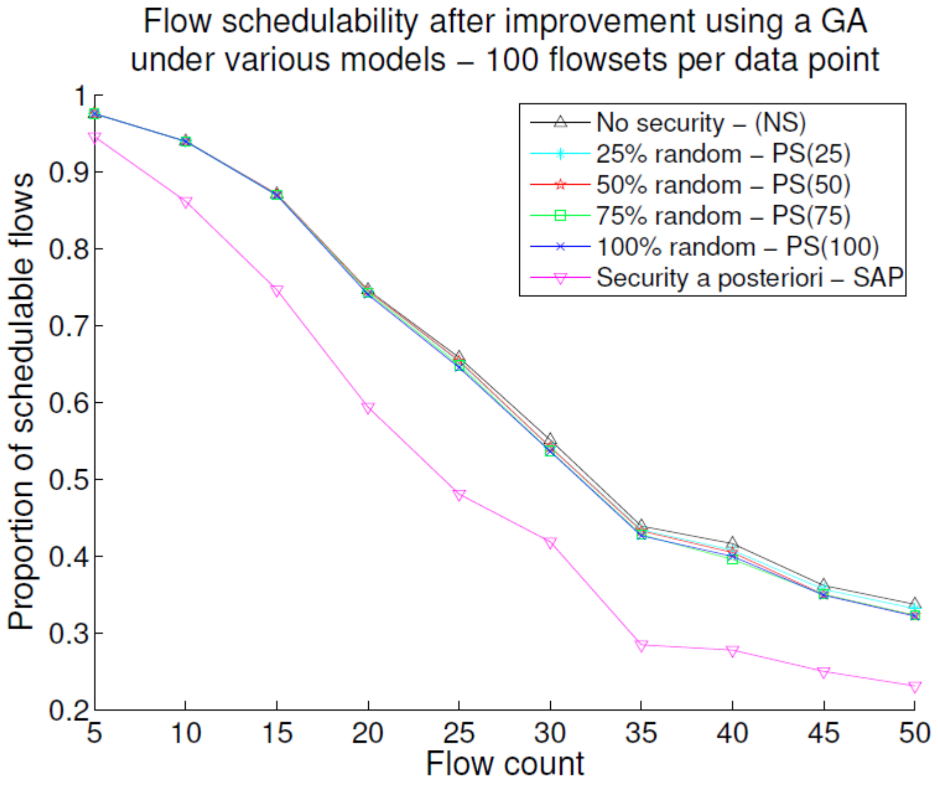
- Reduces attack surface
  - attackers can’t easily monitor timing behaviour
- Increases communication latency variability
  - additional timing interference inflates worst-case
  - analytical models must take that into account
- Optimisation approach
  - randomise as much as possible until worst-case is barely acceptable

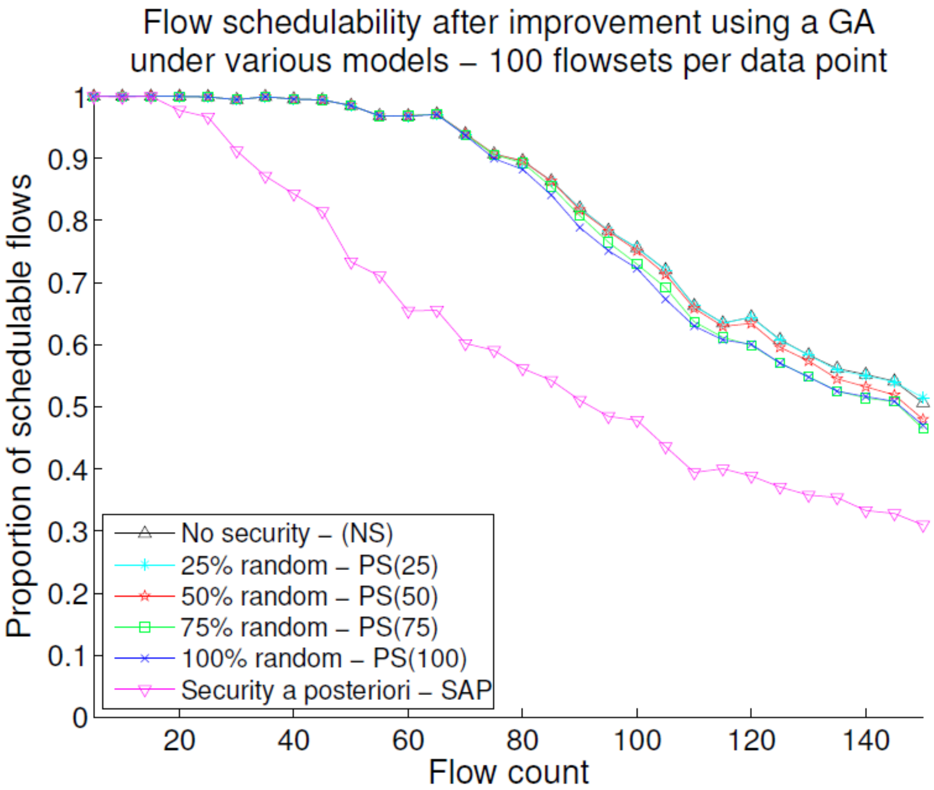
L.S. Indrusiak, J. Harbin, C. Reinbrecht, J. Sepulveda, “Side-channel protected MPSoC through secure real-time networks-on-chip”, Microprocess. Microsystems, v.68, p. 34-46, 2019.

Experimental results

4x4 mesh network-on-chip

8x8 mesh network-on-chip





L.S. Indrusiak, J. Harbin, C. Reinbrecht, J. Sepulveda, “Side-channel protected MPSoC through secure real-time networks-on-chip”, Microprocess. Microsystems, v.68, p. 34-46, 2019.

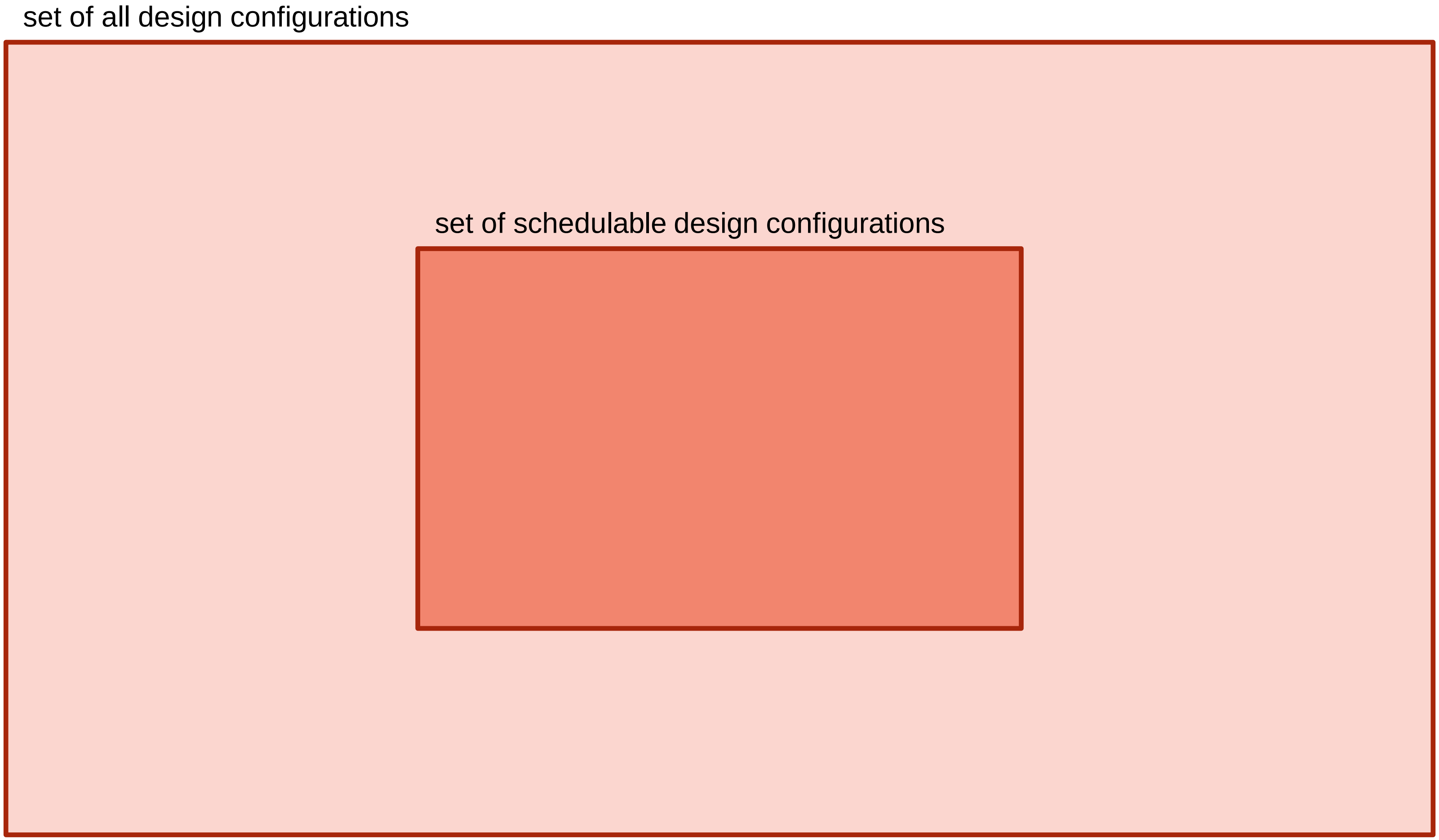
Outline

- Digital Evil Twins ✓
- Identification of worst-case behaviours using real-time analysis ✓
  - guiding optimisation
- Semi-automatic synthesis of worst-case real-time analysis models
- Open areas of research

Synthesis of worst-case analytical models

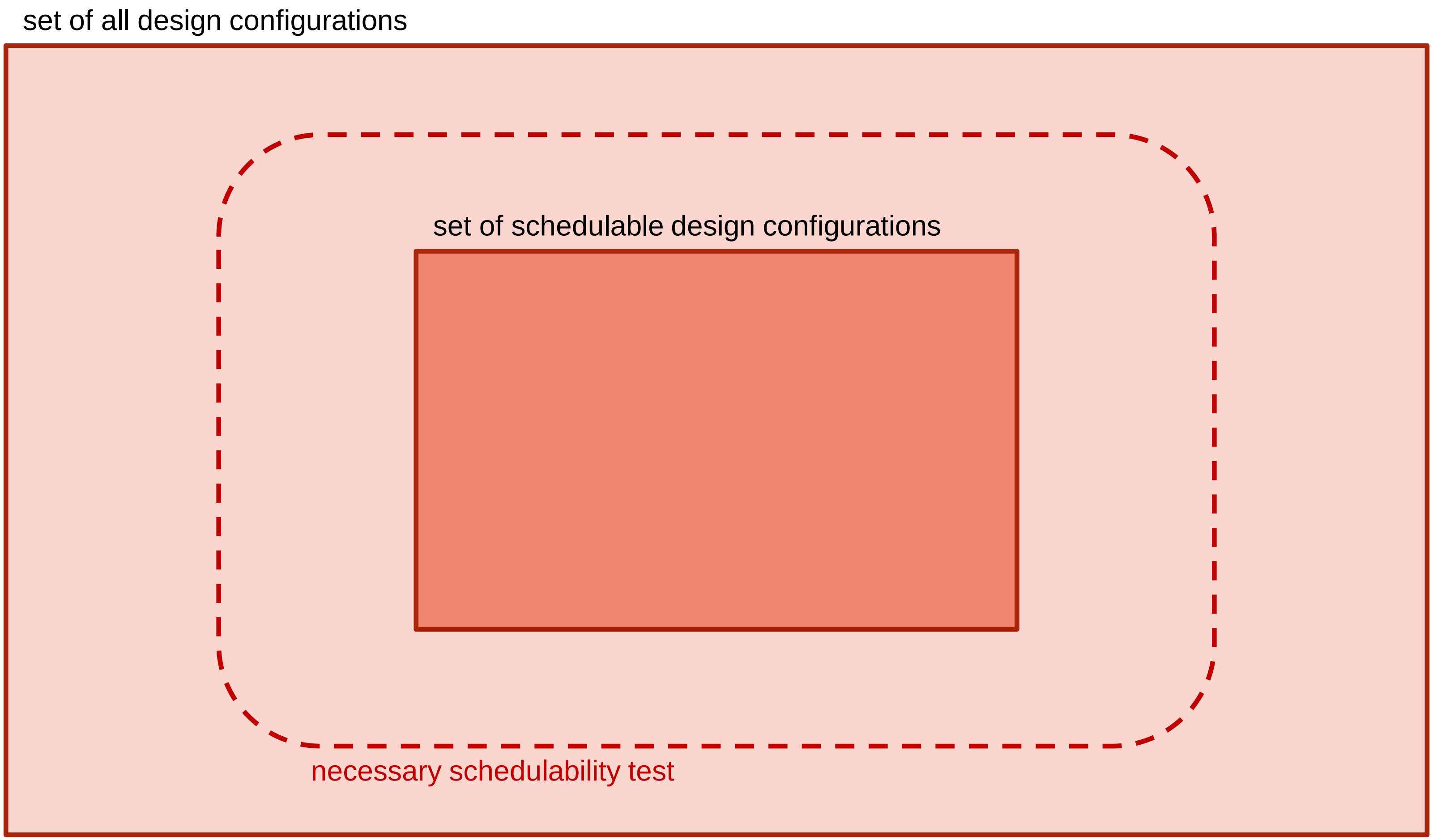
- Worst case models are manually derived by researchers and practitioners
    - error-prone, impractical for many complex systems
- Approach for evolutionary synthesis
    - candidate models (equations) encoded according to a grammar
    - evolution: fitness of candidate equations given by how well they capture the worst case behaviour of the system, observed in a large number of simulation scenarios

P. Dziurzanski, R. Davis, L. S. Indrusiak, “Synthesizing Real-Time Schedulability Tests using Evolutionary Algorithms: A Proof of Concept”, Real-Time Systems Symposium (RTSS), 2019.



set of all design configurations

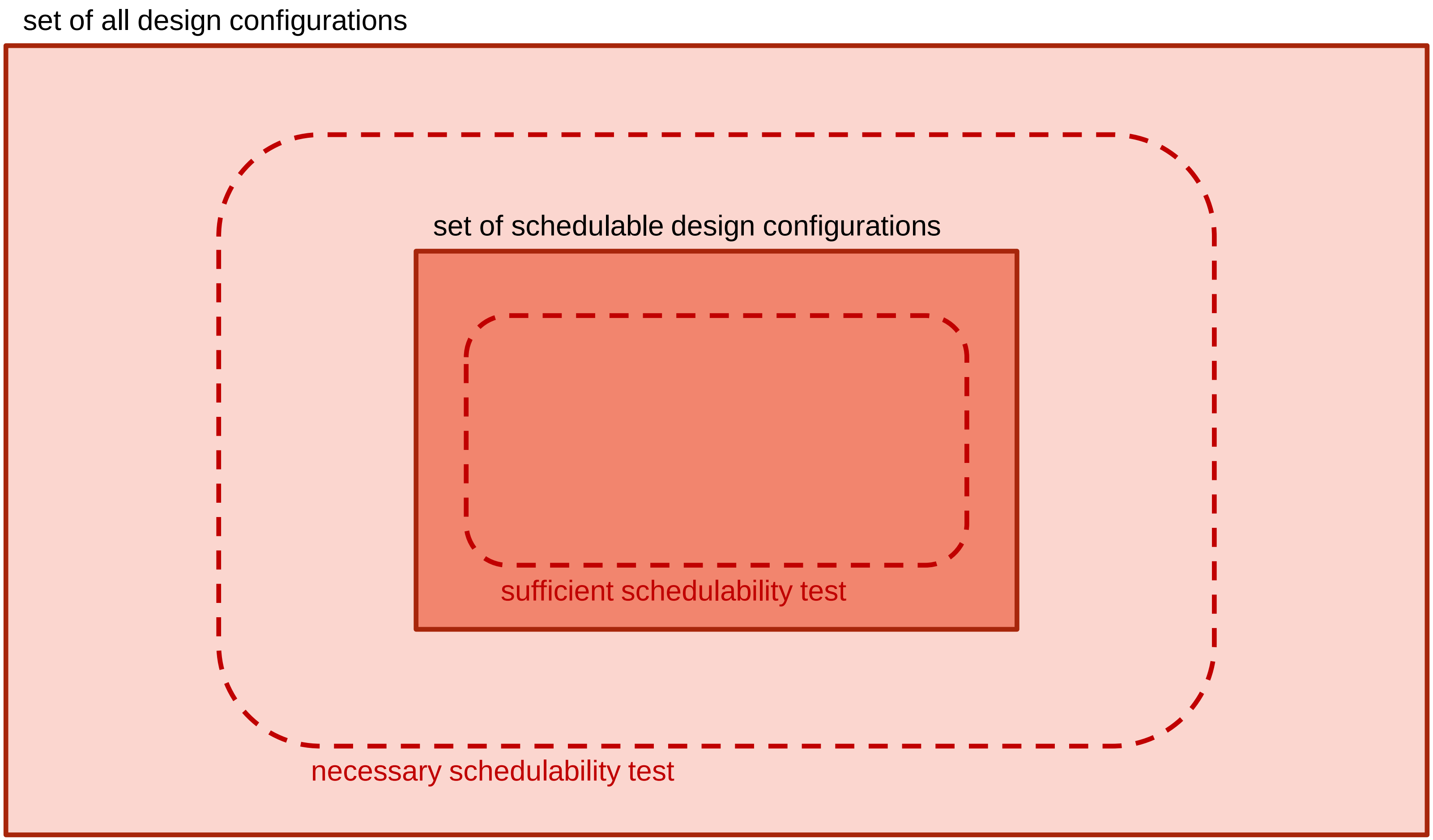
set of schedulable design configurations



set of all design configurations

set of schedulable design configurations

necessary schedulability test

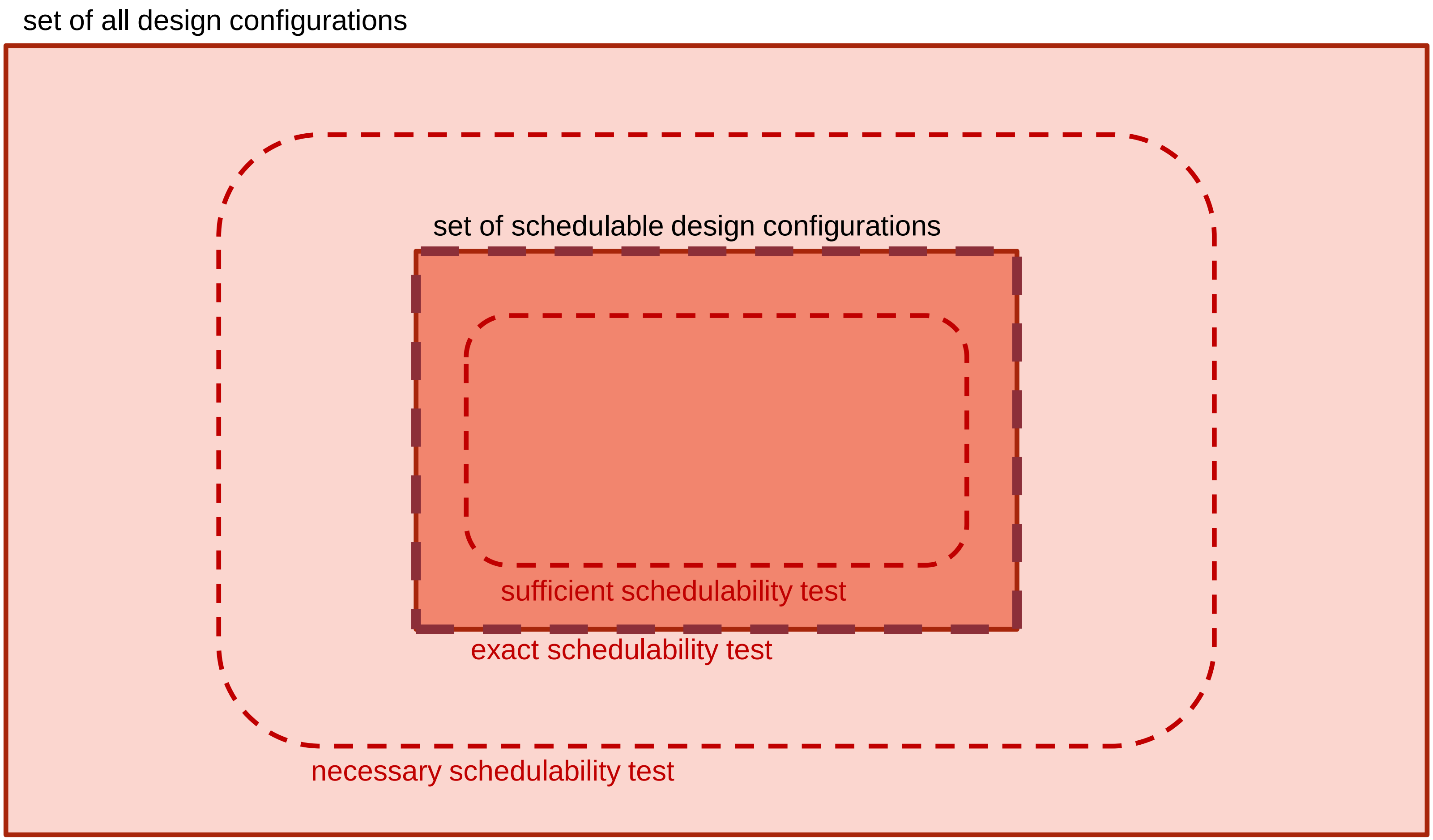


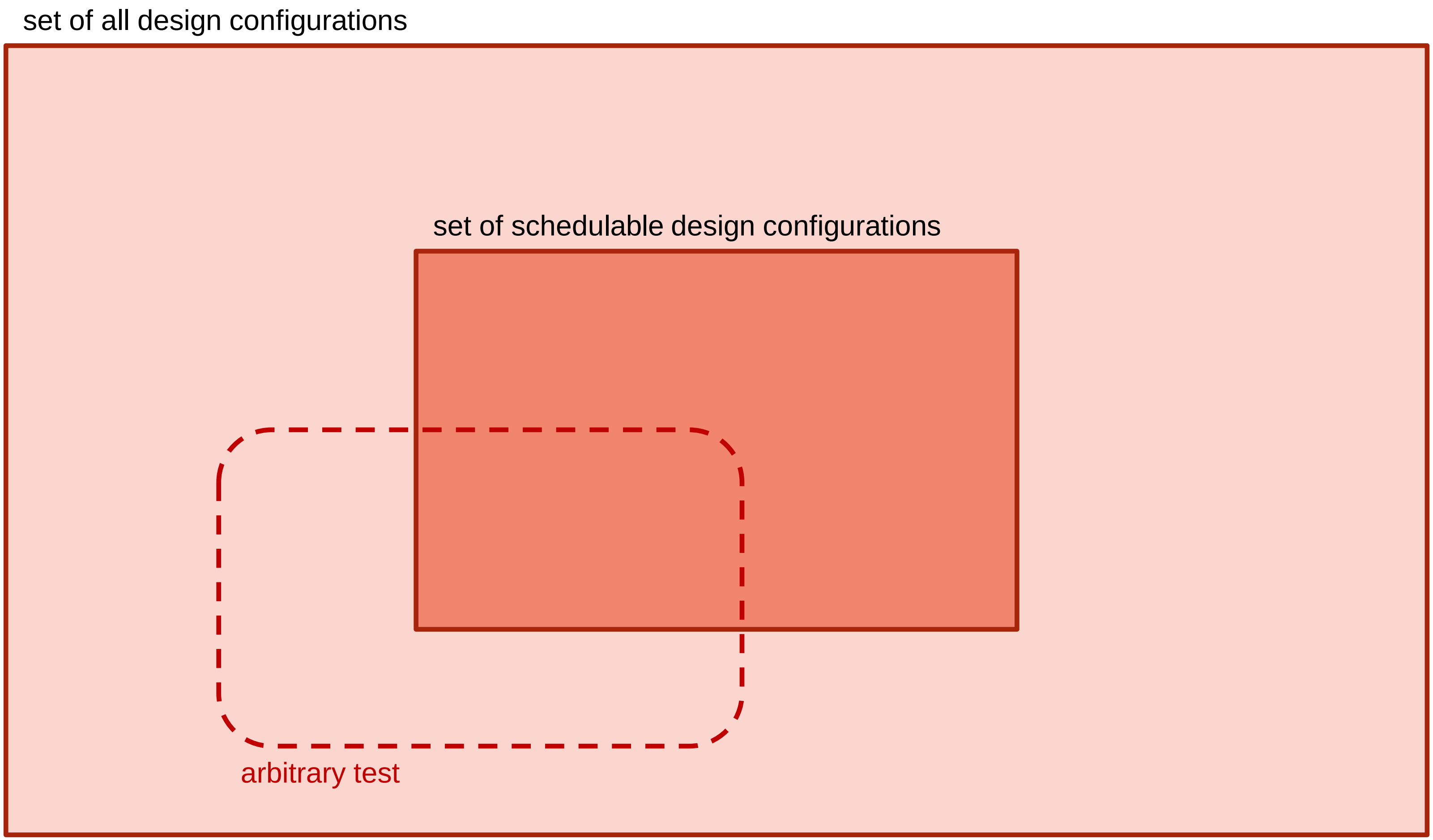
set of all design configurations

set of schedulable design configurations

sufficient schedulability test

necessary schedulability test

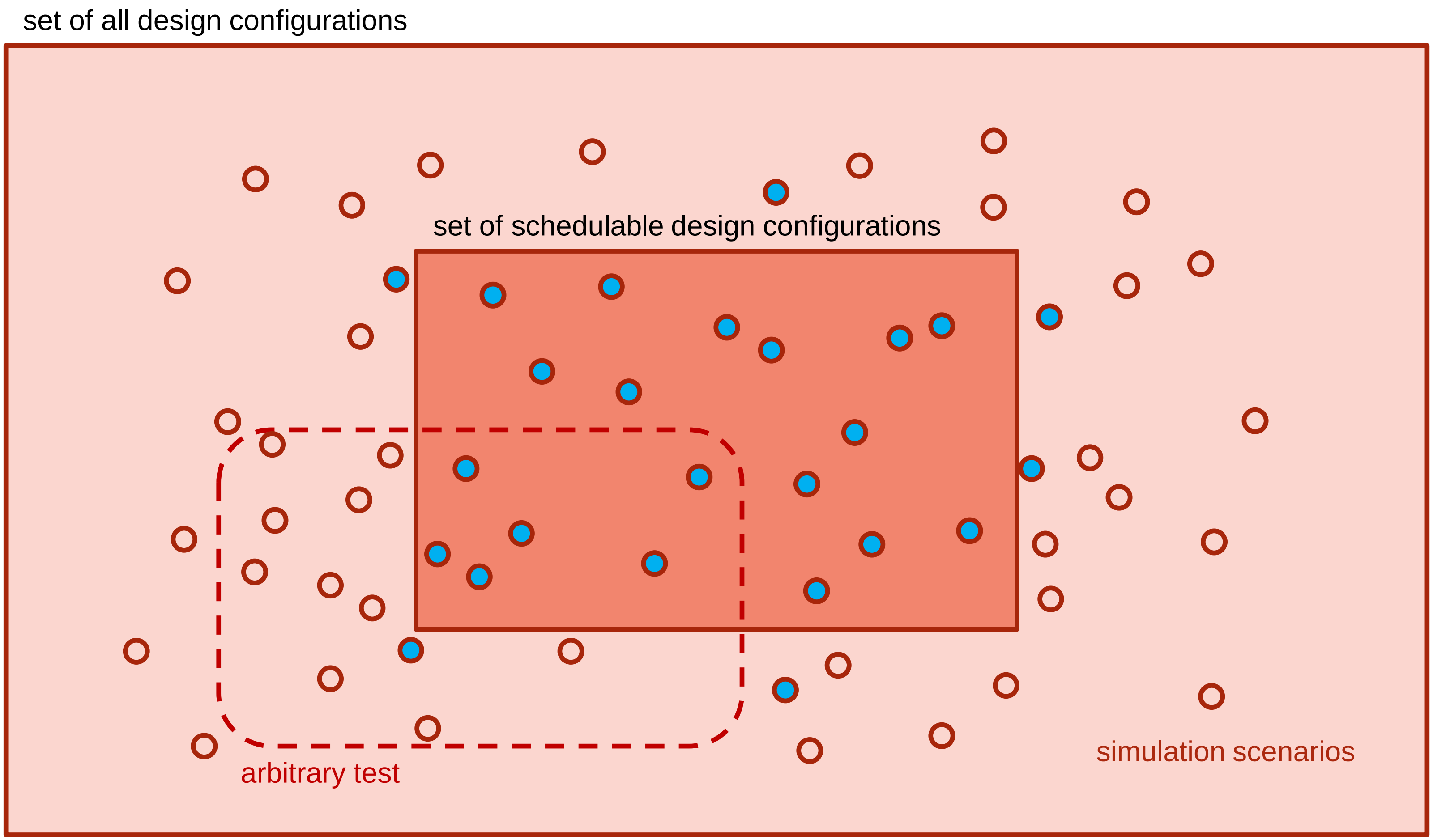

set of all design configurations
set of schedulable design configurations
sufficient schedulability test
exact schedulability test
necessary schedulability test

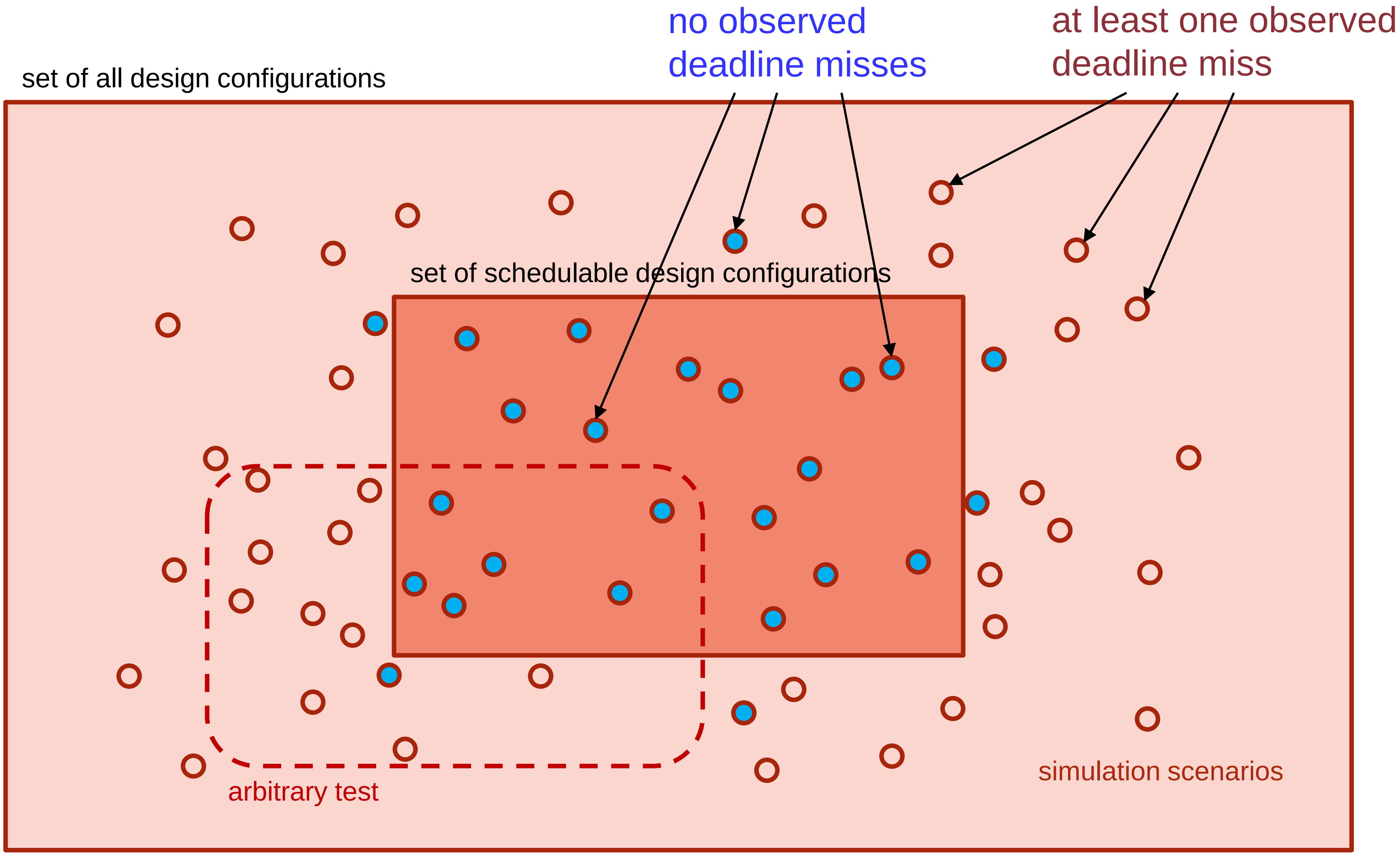


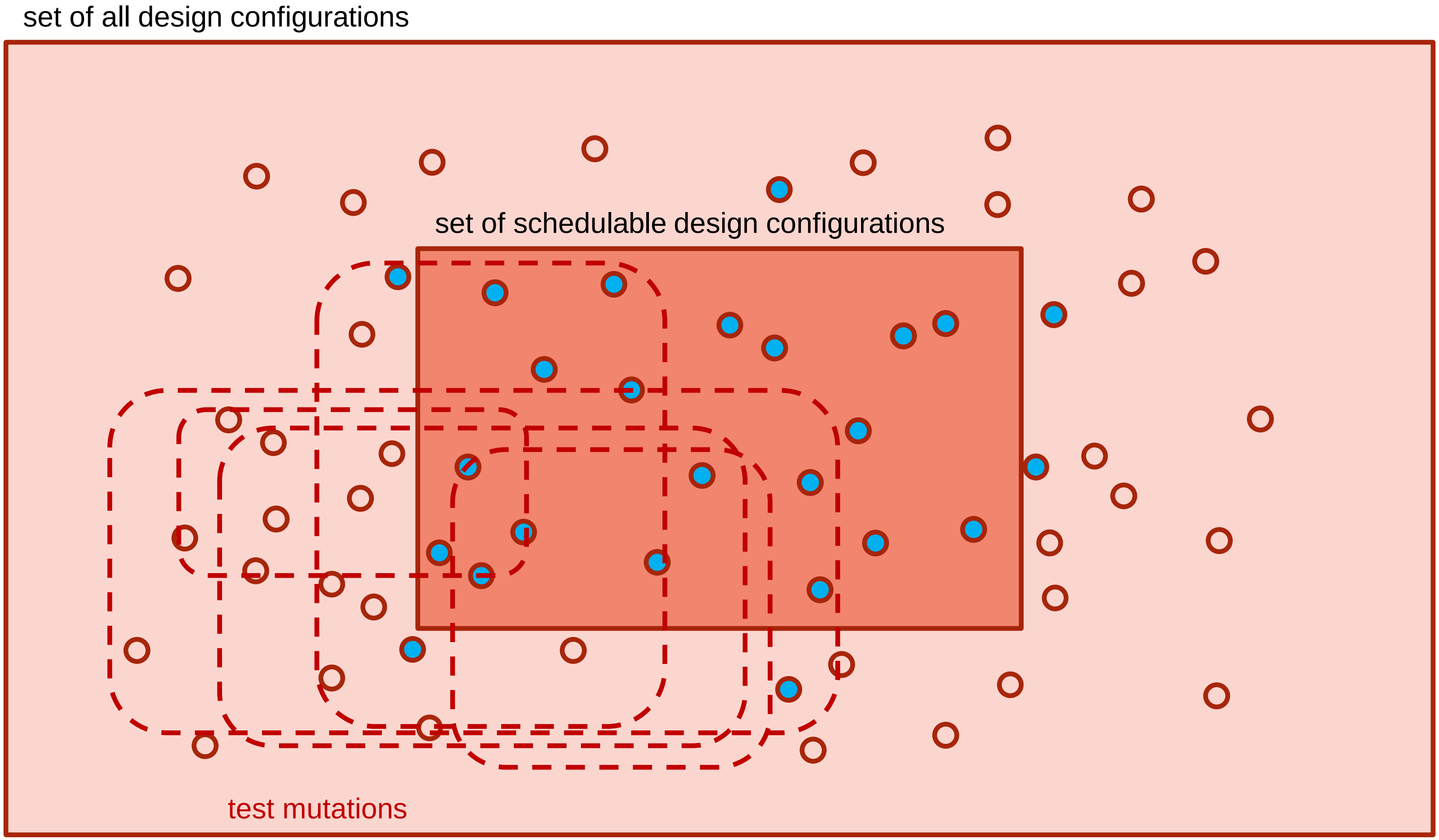
set of all design configurations

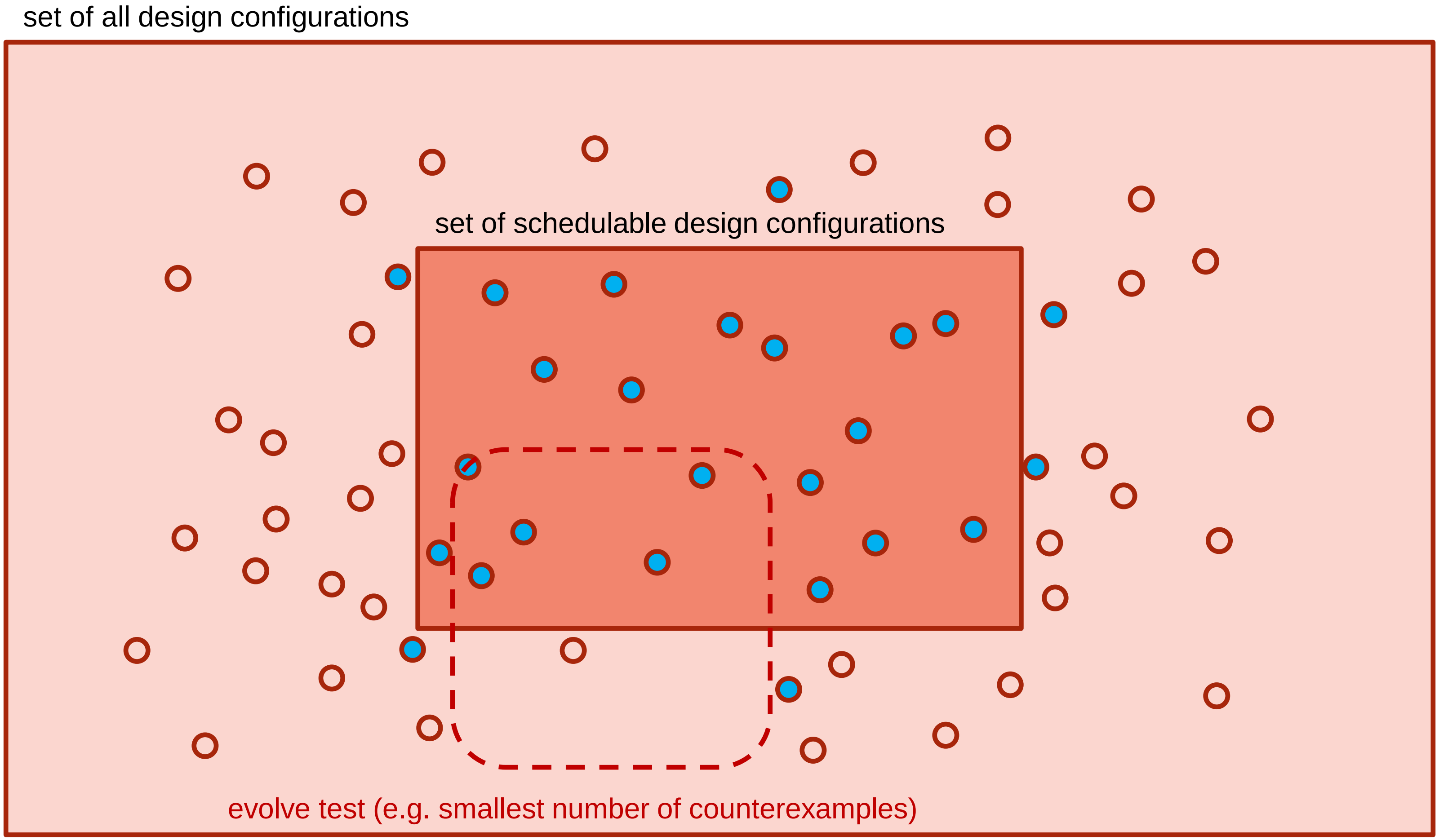
set of schedulable design configurations

arbitrary test


set of all design configurations
set of schedulable design configurations
arbitrary test
simulation scenarios


no observed deadline misses
at least one observed deadline miss
set of all design configurations
set of schedulable design configurations
arbitrary test
simulation scenarios

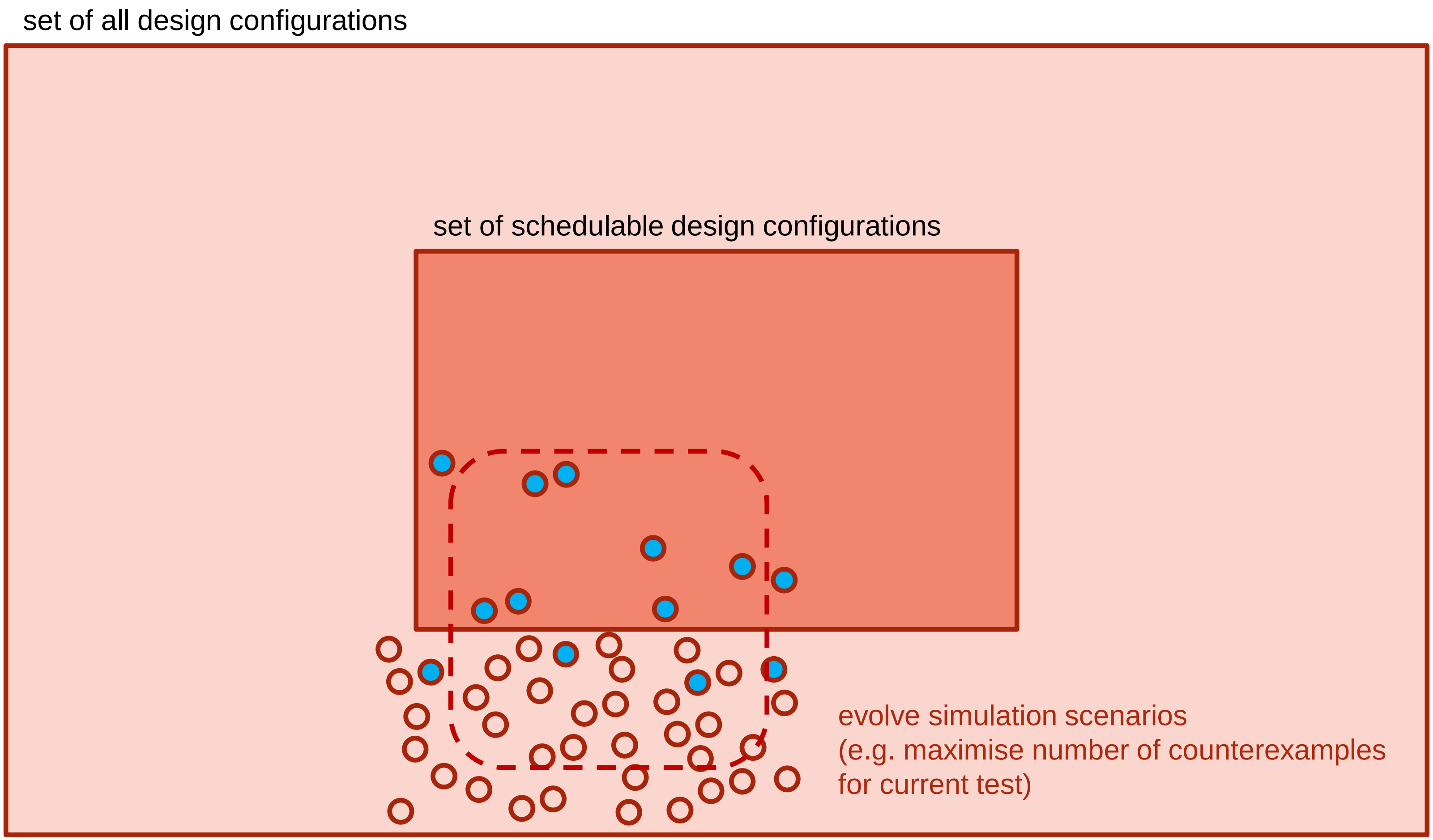

set of all design configurations
set of schedulable design configurations
test mutations



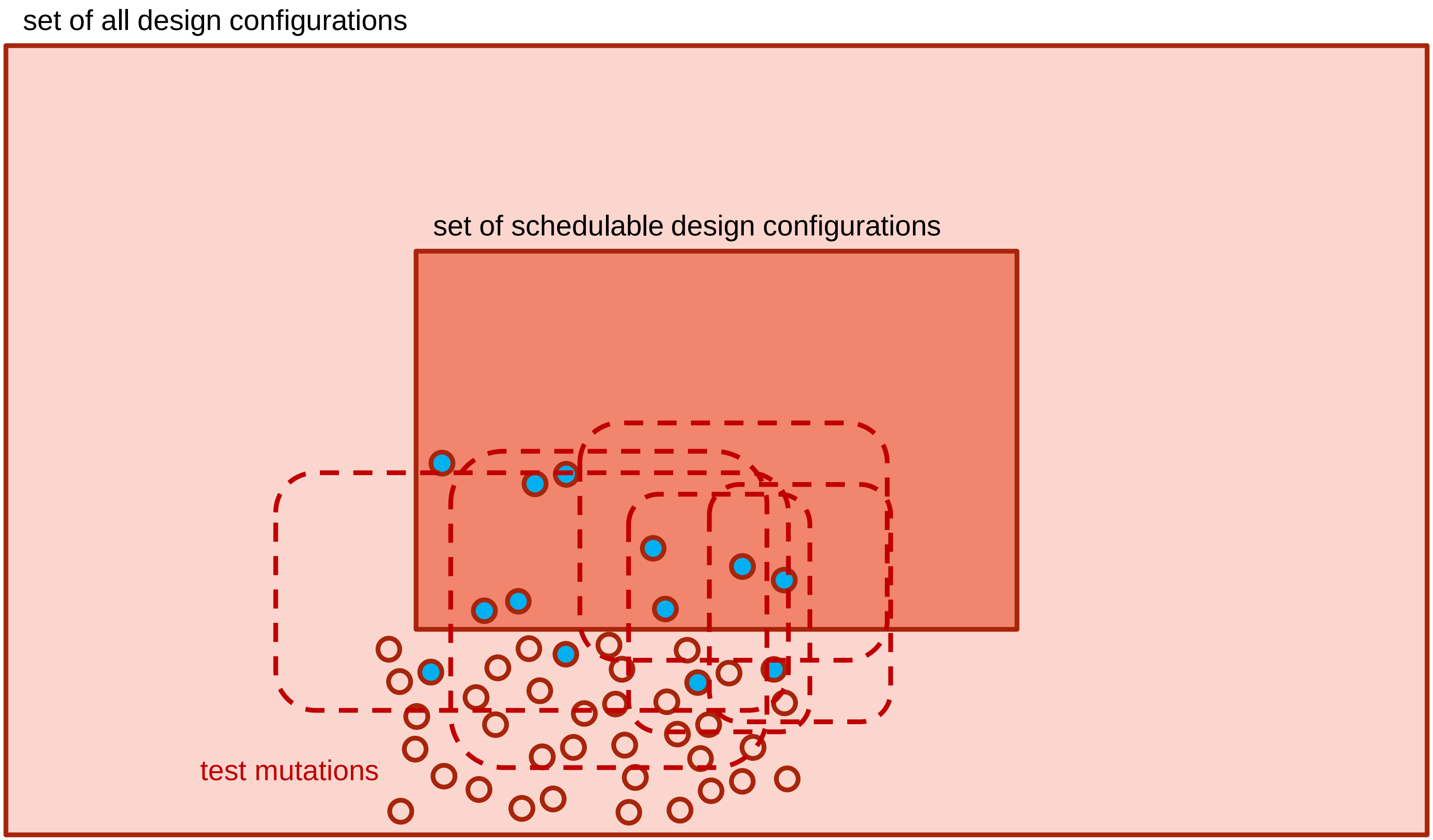
set of all design configurations

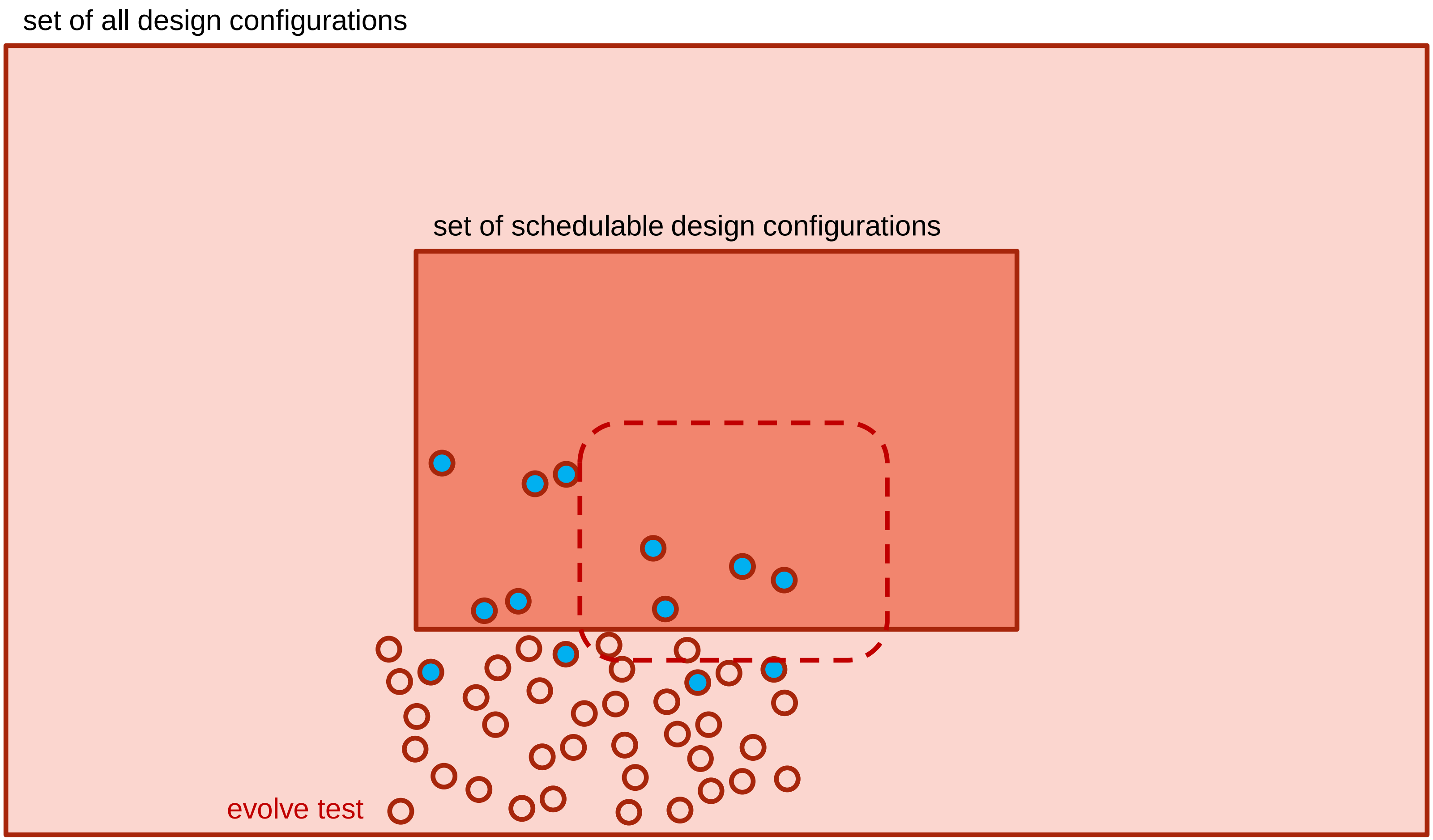
set of schedulable design configurations

evolve test (e.g. smallest number of counterexamples)


set of all design configurations
set of schedulable design configurations

evolve simulation scenarios
(e.g. maximise number of counterexamples
for current test)

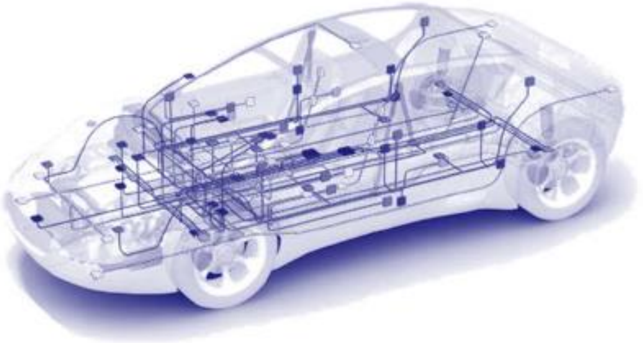

set of all design configurations
set of schedulable design configurations
test mutations


set of all design configurations
set of schedulable design configurations
evolve test

Synthesis of worst-case analytical models

- Proof of concept was able to evolve equations providing sufficient worst-case analysis for hard real-time messages over an automotive CAN bus (in-vehicle network used in most cars)

$$R_i = J_i + C_i + B_i + \sum_{k \in hp(i)} \left\lceil \frac{R_i - J_i + C_i + J_k}{T_k} \right\rceil C_k$$



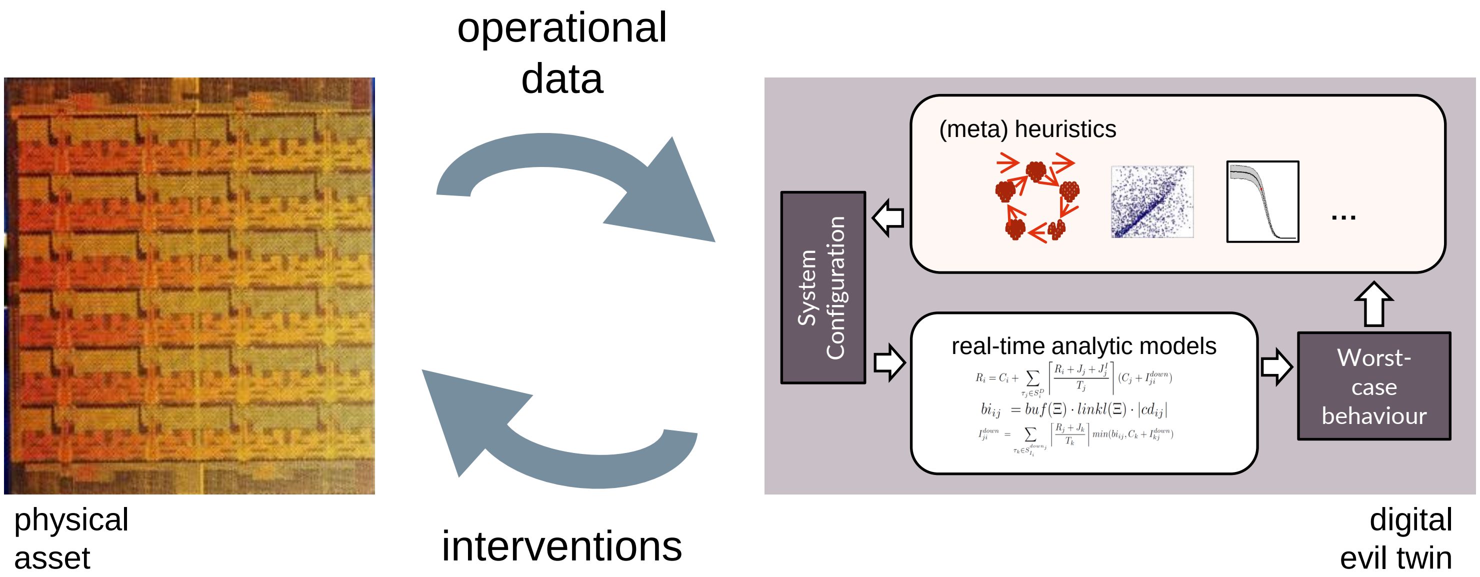
- Expert assistance still needed
  - e.g. to provide a proof that the test is sufficient

P. Dziurzanski, R. Davis, L. S. Indrusiak, “Synthesizing Real-Time Schedulability Tests using Evolutionary Algorithms: A Proof of Concept”, Real-Time Systems Symposium (RTSS), 2019.

Outline

- Digital Evil Twins ✓
- Identification of worst-case behaviours using real-time analysis ✓
  - guiding optimisation
- Semi-automatic synthesis of worst-case real-time analysis models ✓
- Open areas of research

Digital Evil Twins – open areas of research



operational data

interventions

physical asset

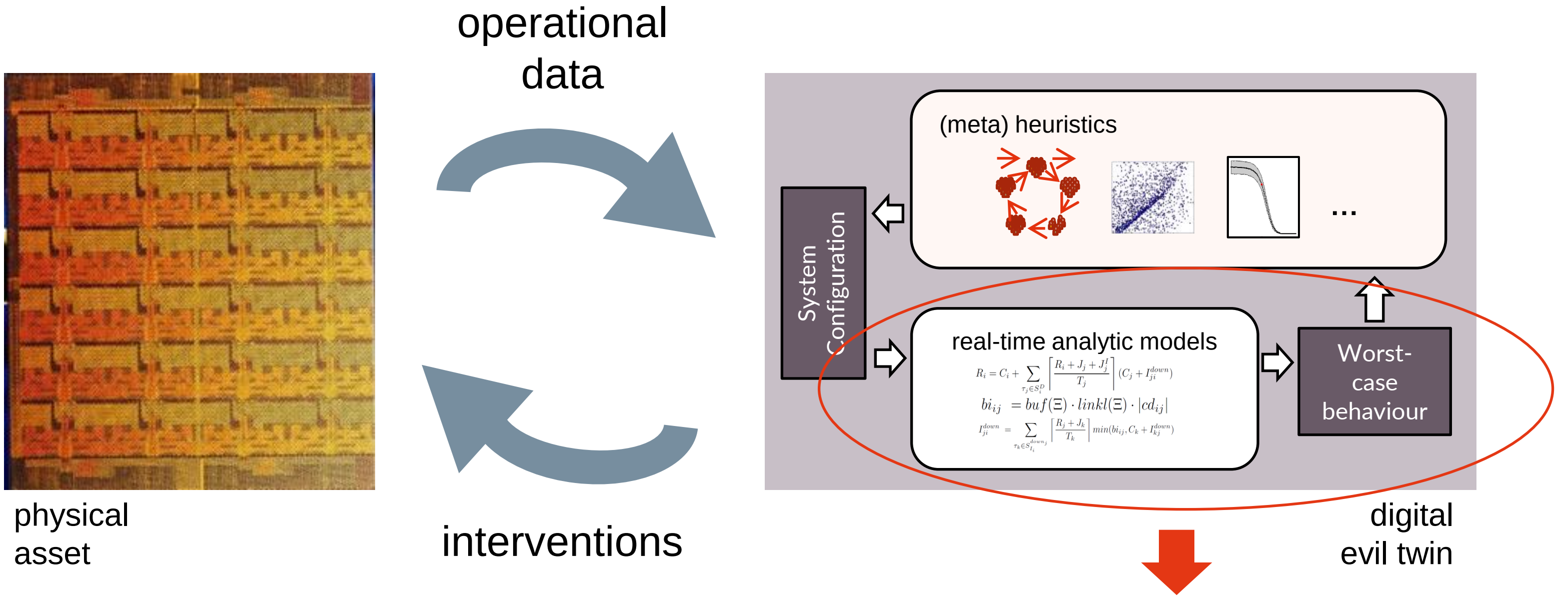
digital evil twin

$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down})$$

$$bi_{ij} \; = buf(\Xi) \cdot linkl(\Xi) \cdot |cd_{ij}|$$

$$I_{ji}^{down} \; = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil min(bi_{ij}, C_k + I_{kj}^{down})$$

Digital Evil Twins – open areas of research



$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down})$$

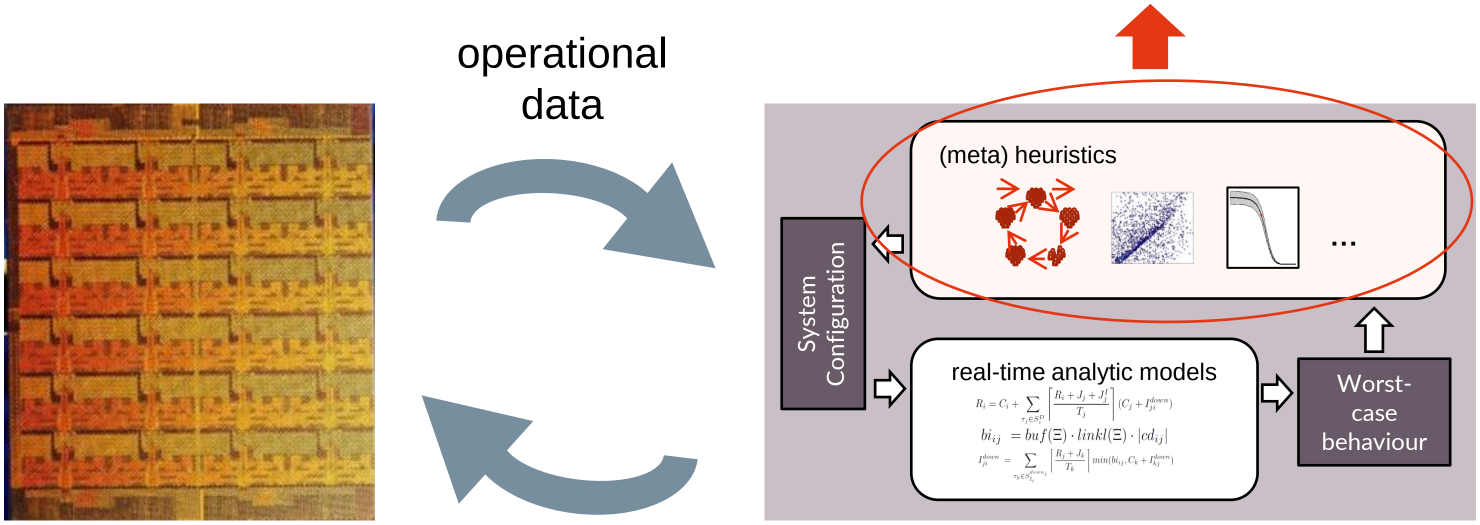
$$bi_{ij} = buf(\Xi) \cdot linkl(\Xi) \cdot |cd_{ij}|$$

$$I_{ji}^{down} = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil min(bi_{ij}, C_k + I_{kj}^{down})$$

current approaches: created manually or semi-automatically using simulation
coming next: improve analysis models with operational data (or with digital twin, if available)

Digital Evil Twins – open areas of research

current approaches: search-based metaheuristics (e.g. GAs, simulated annealing)
coming next: ML

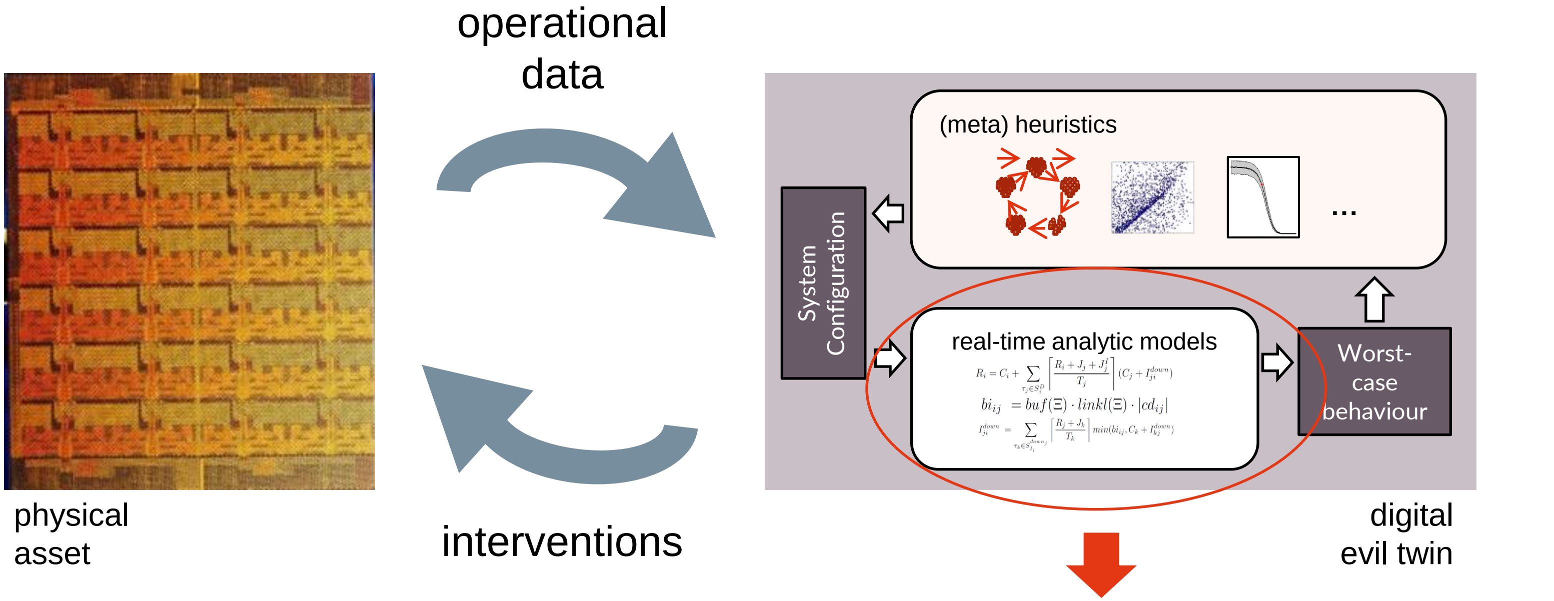


$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down})$$

$$bi_{ij} \ = buf(\Xi) \cdot linkl(\Xi) \cdot |cd_{ij}|$$

$$I_{ji}^{down} \ = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil min(bi_{ij}, C_k + I_{kj}^{down})$$

Digital Evil Twins – open areas of research



operational data

interventions

physical asset

digital evil twin

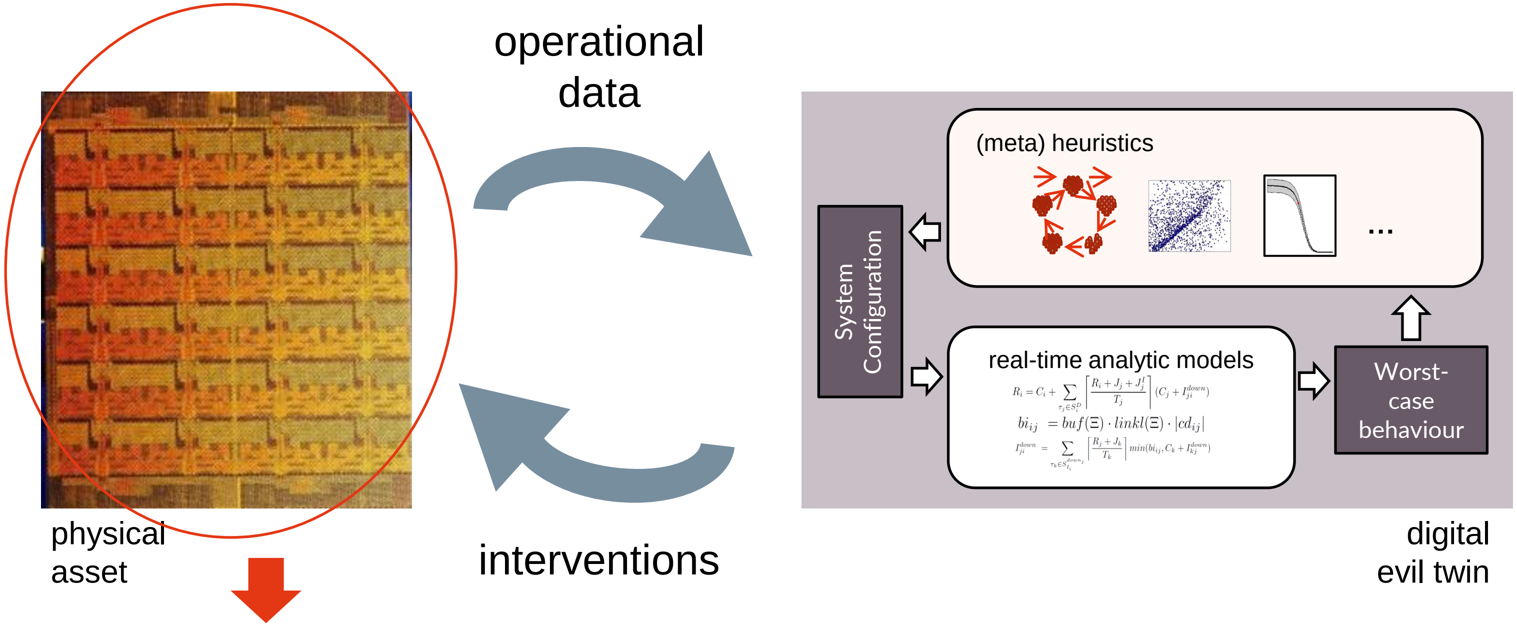
$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down})$$

$$bi_{ij} = buf(\Xi) \cdot linkl(\Xi) \cdot |cd_{ij}|$$

$$I_{ji}^{down} = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil min(bi_{ij}, C_k + I_{kj}^{down})$$

coming next: improve performance through parallelisation, hardware acceleration, exploring trade-off between model speed and tightness

Digital Evil Twins – open areas of research



operational data

interventions

physical asset

digital evil twin

System Configuration

(meta) heuristics

…

real-time analytic models

$$R_i = C_i + \sum_{\tau_j \in S_i^D} \left\lceil \frac{R_i + J_j + J_j^I}{T_j} \right\rceil (C_j + I_{ji}^{down})$$

$$bi_{ij} = buf(\Xi) \cdot linkl(\Xi) \cdot |cd_{ij}|$$

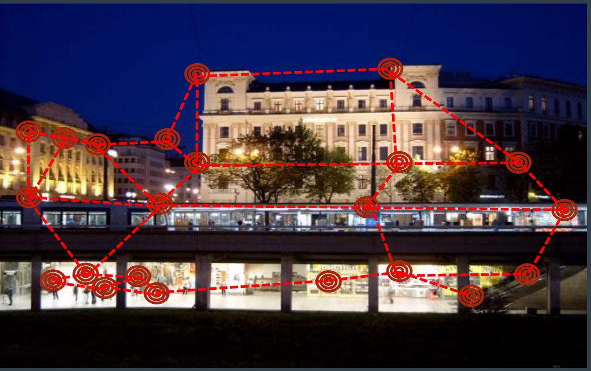
$$I_{ji}^{down} = \sum_{\tau_k \in S_{I_i}^{down_j}} \left\lceil \frac{R_j + J_k}{T_k} \right\rceil min(bi_{ij}, C_k + I_{kj}^{down})$$

Worst-case behaviour

current approaches: on-chip multiprocessors, IoT networks, automotive networks

coming next: ???

collaborations welcome



Digital Evil Twins

Identification of worst-case behaviours in real-time systems

Leandro Soares Indrusiak

Real-Time and Distributed Systems Group
Department of Computer Science
University of York