Control-Quality Optimization for Distributed Embedded Systems with Adaptive Fault Tolerance

Soheil Samii[1,2], Unmesh D. Bordoloi[2], Petru Eles[2], Zebo Peng[2], Anton Cervin[3]

[1]Embedded Intelligent Solutions, Semcon AB

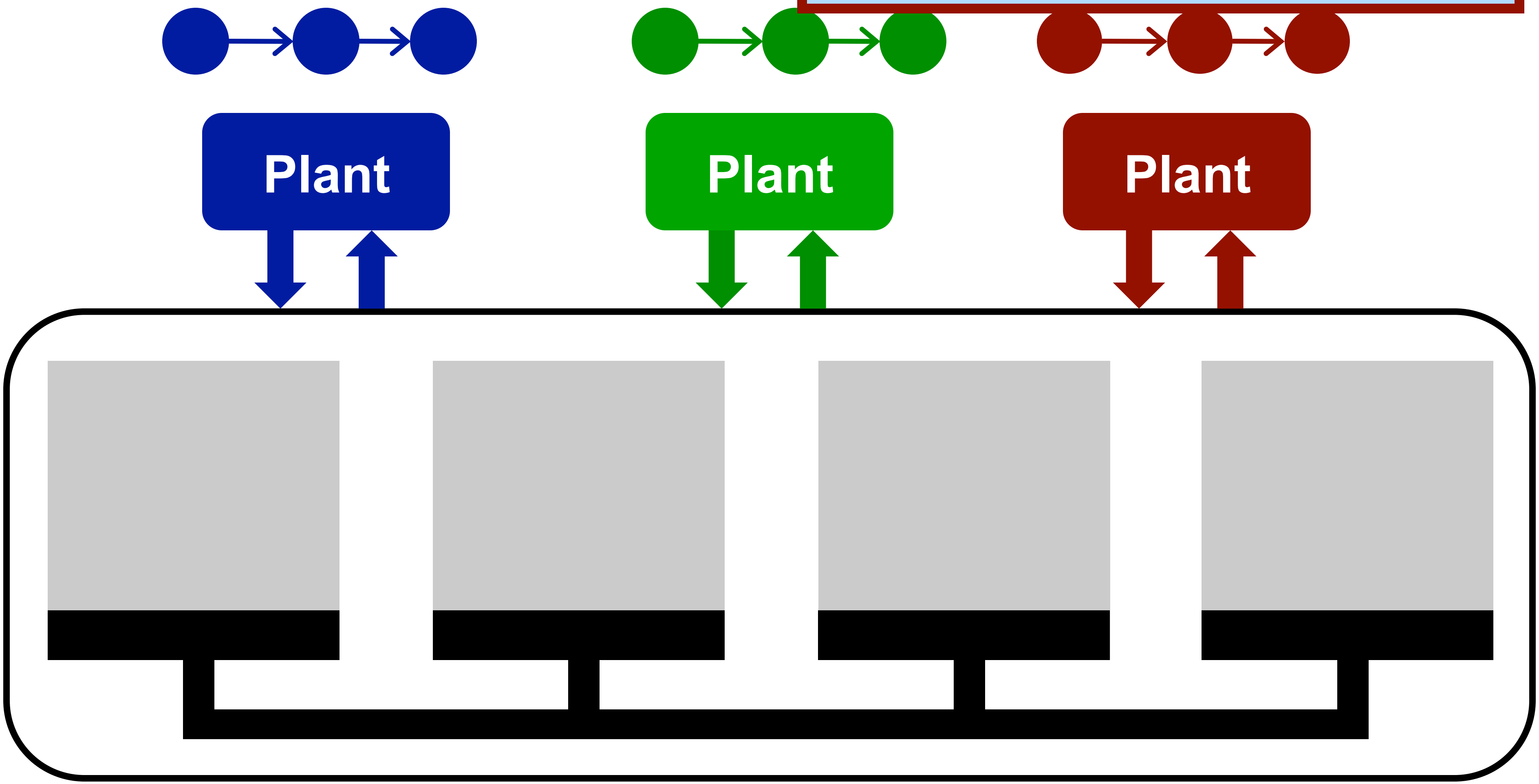
[2]Dept. of Computer and Information Science, Linköping University

[3]Dept. of Automatic Control, Lund University

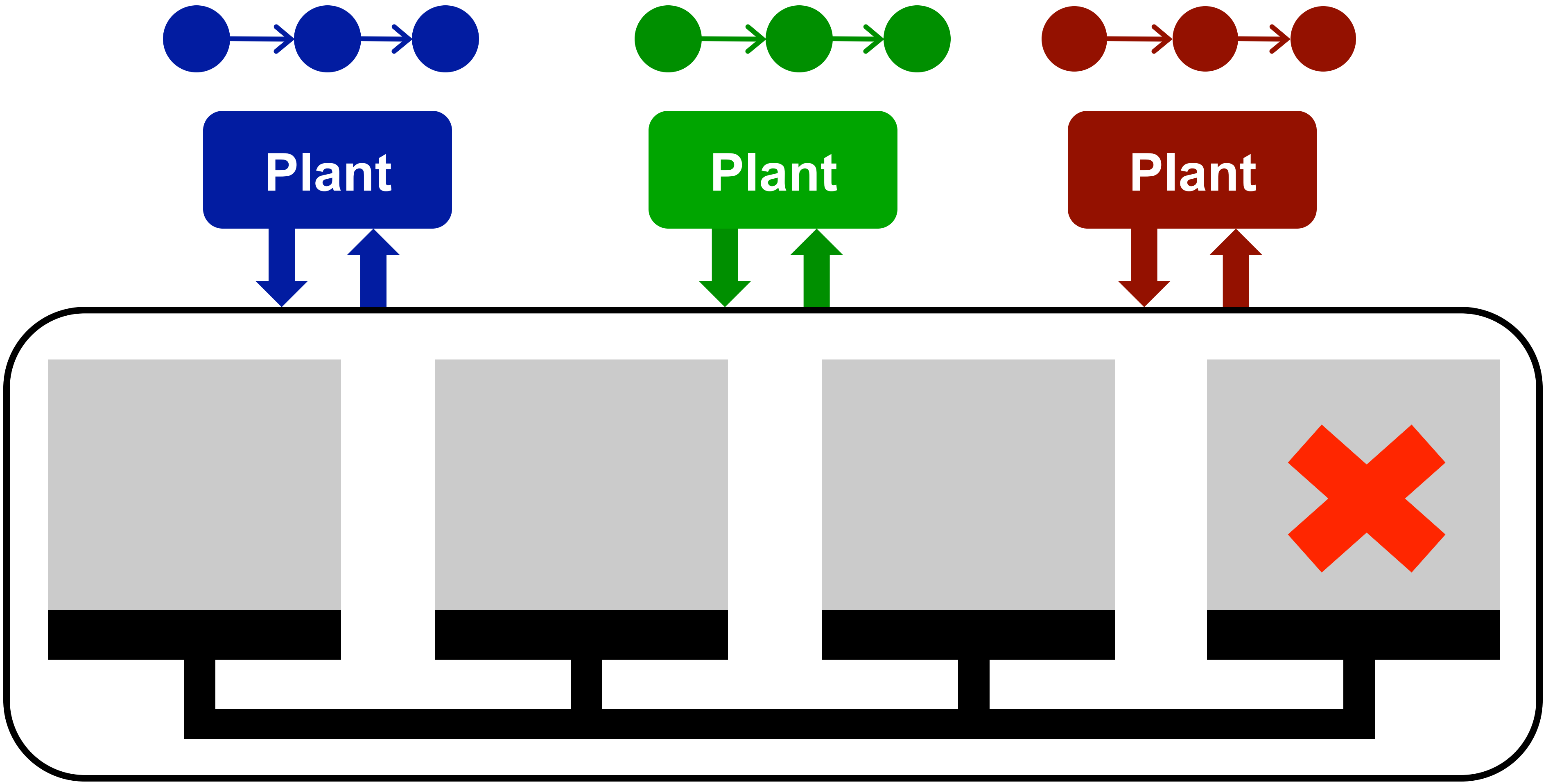
Sweden

Motivation

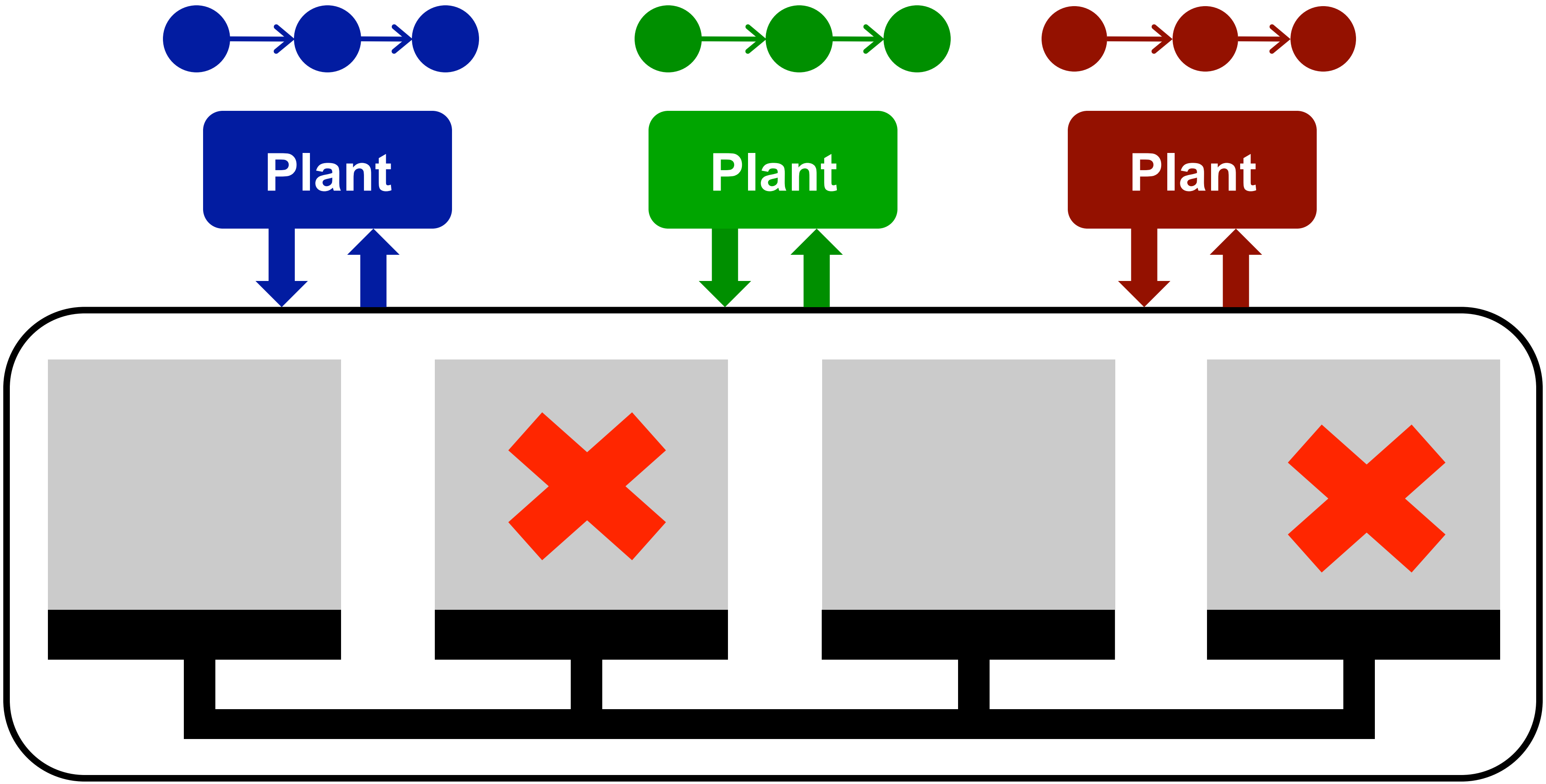
- **Control quality**
  - **Periods, control laws**
  - **Mapping, schedule**



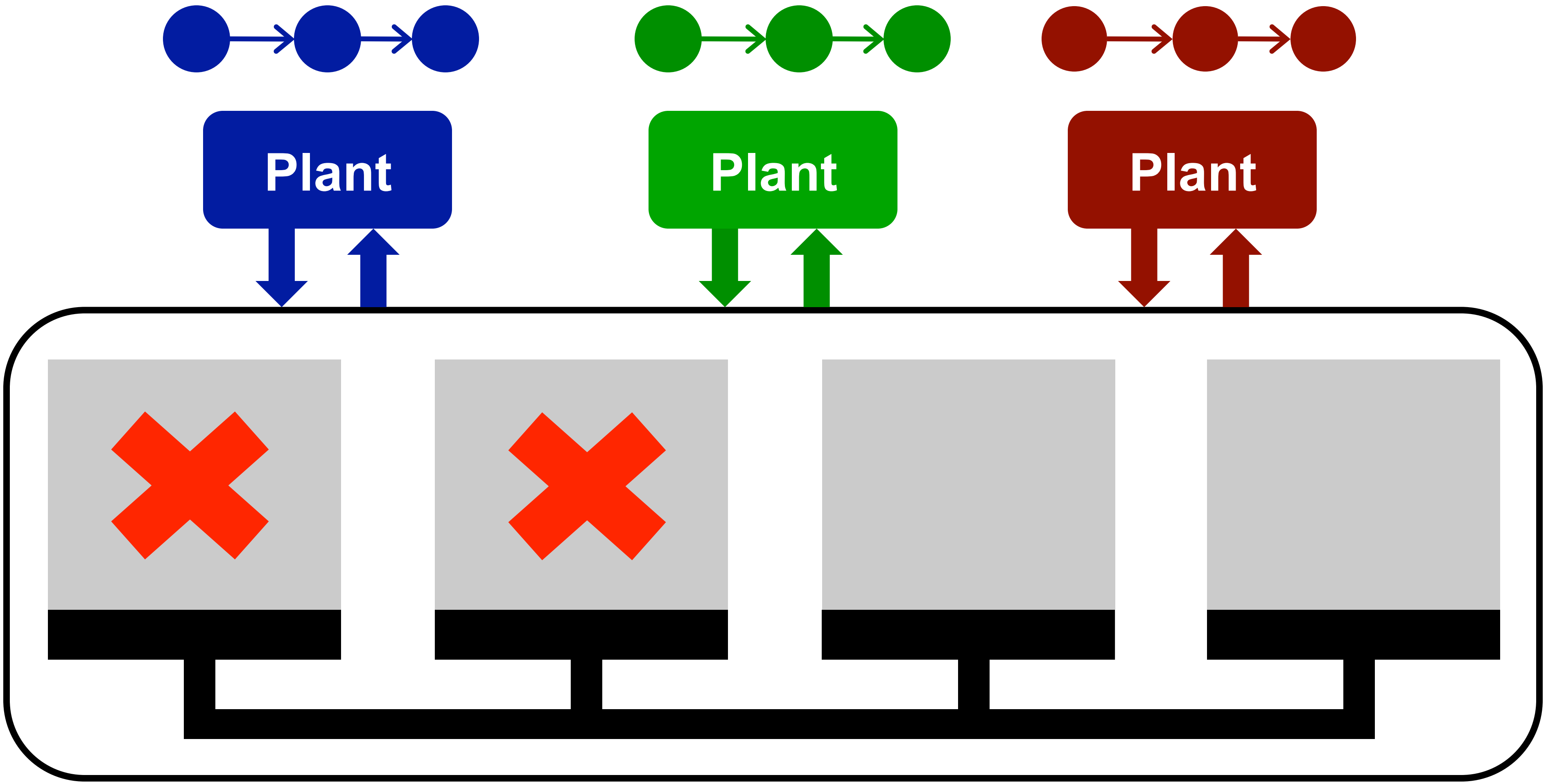
Motivation



Motivation

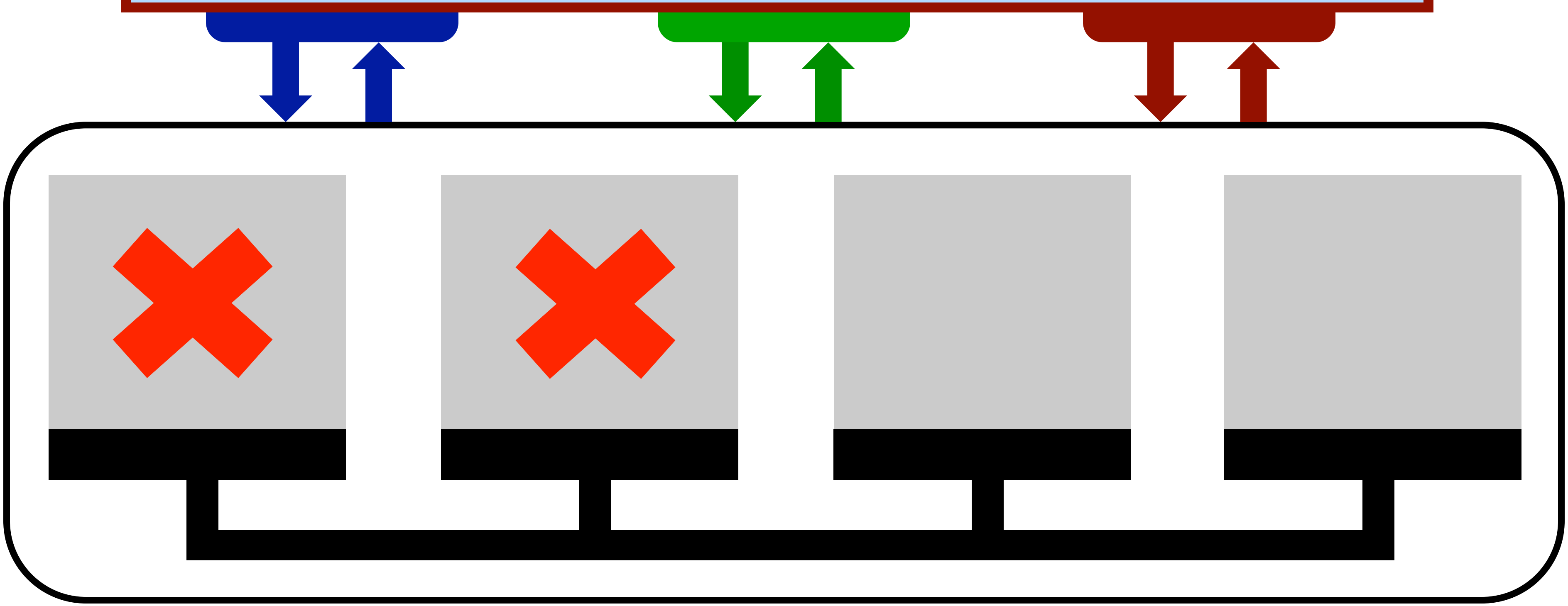


Motivation



Motivation

- **Node faults lead to new <u>configurations</u>**
- **Unpractical to synthesize solutions for all configurations**



Overview of our approach

- **Classify <u>feasible</u> configurations**
  - **Sufficient computation capacity**
  - **Availability of external interfaces to sensors and actuators**

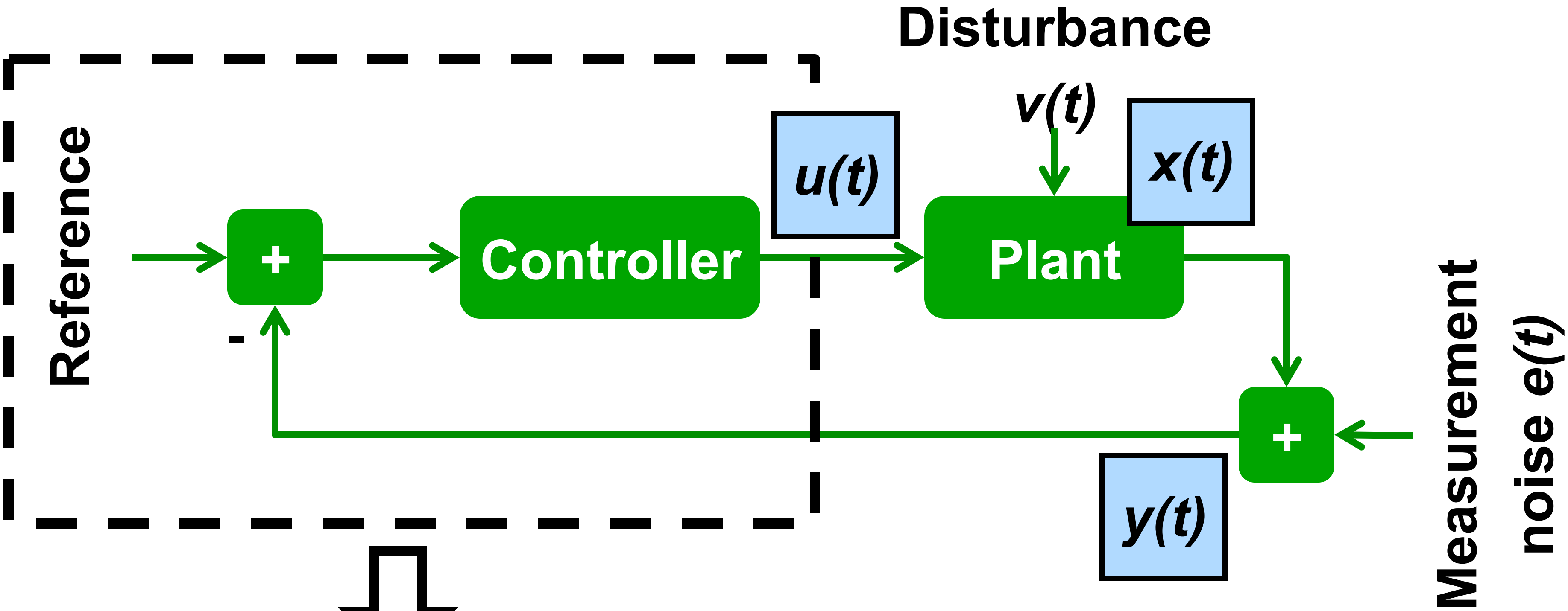
- **Synthesis of a certain set of base configurations is sufficient to satisfy fault-tolerance requirements**

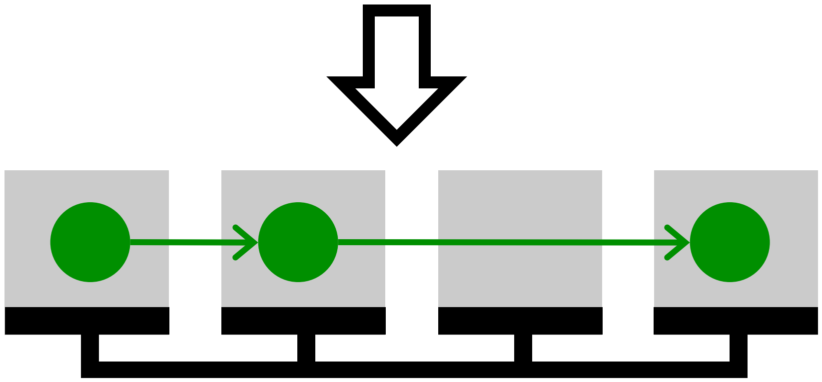
- **Design optimization for additional configurations to optimize control quality**

Outline

- **System model**
- **Example: Distributed control systems with faults**
- **Base configurations**
- **Control-quality optimization**
- **Experiments**

System model





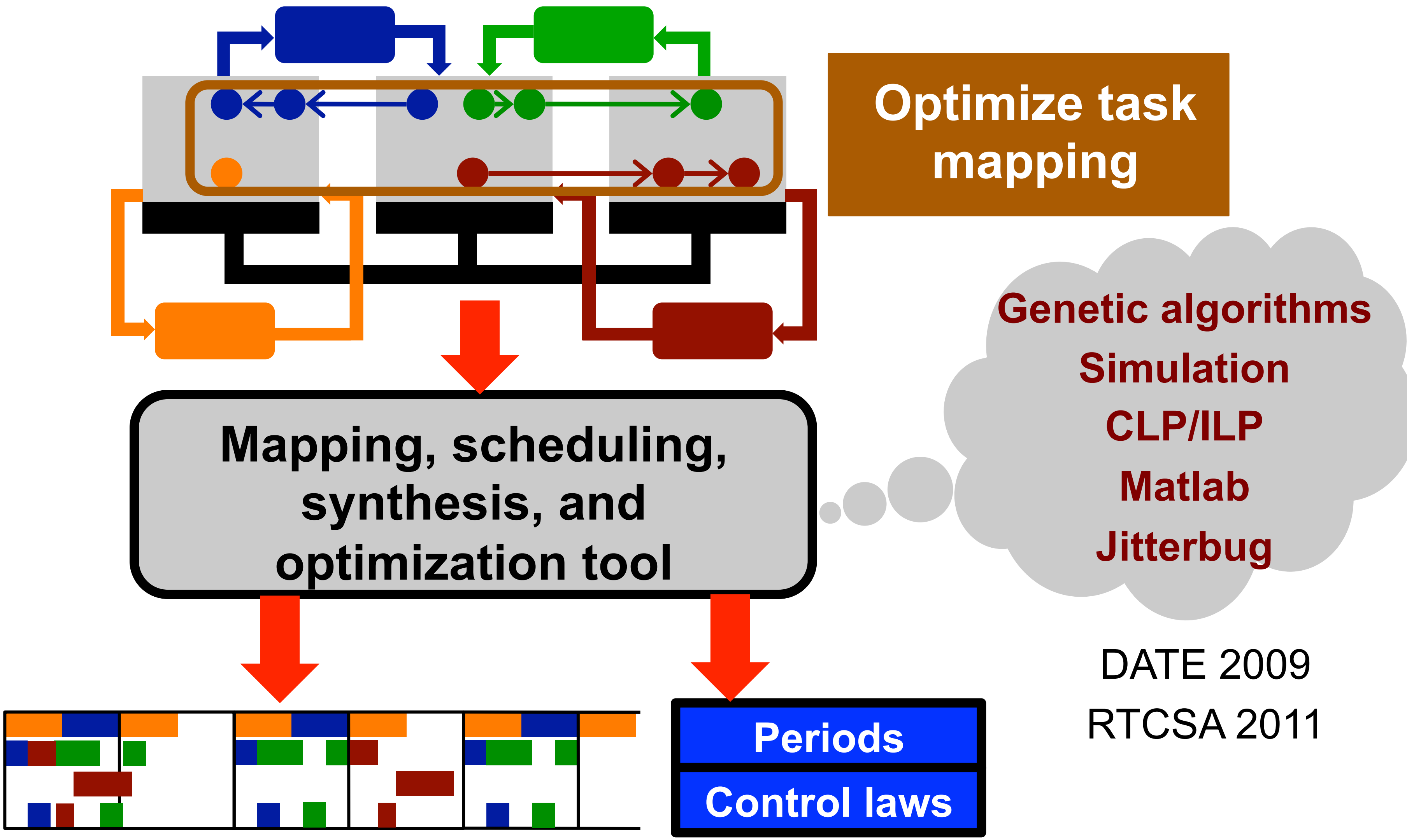
$$dx(t)/dt = Ax(t) + Bu(t) + v(t)$$

$$y(t) = Cx(t) + e(t)$$

Control quality

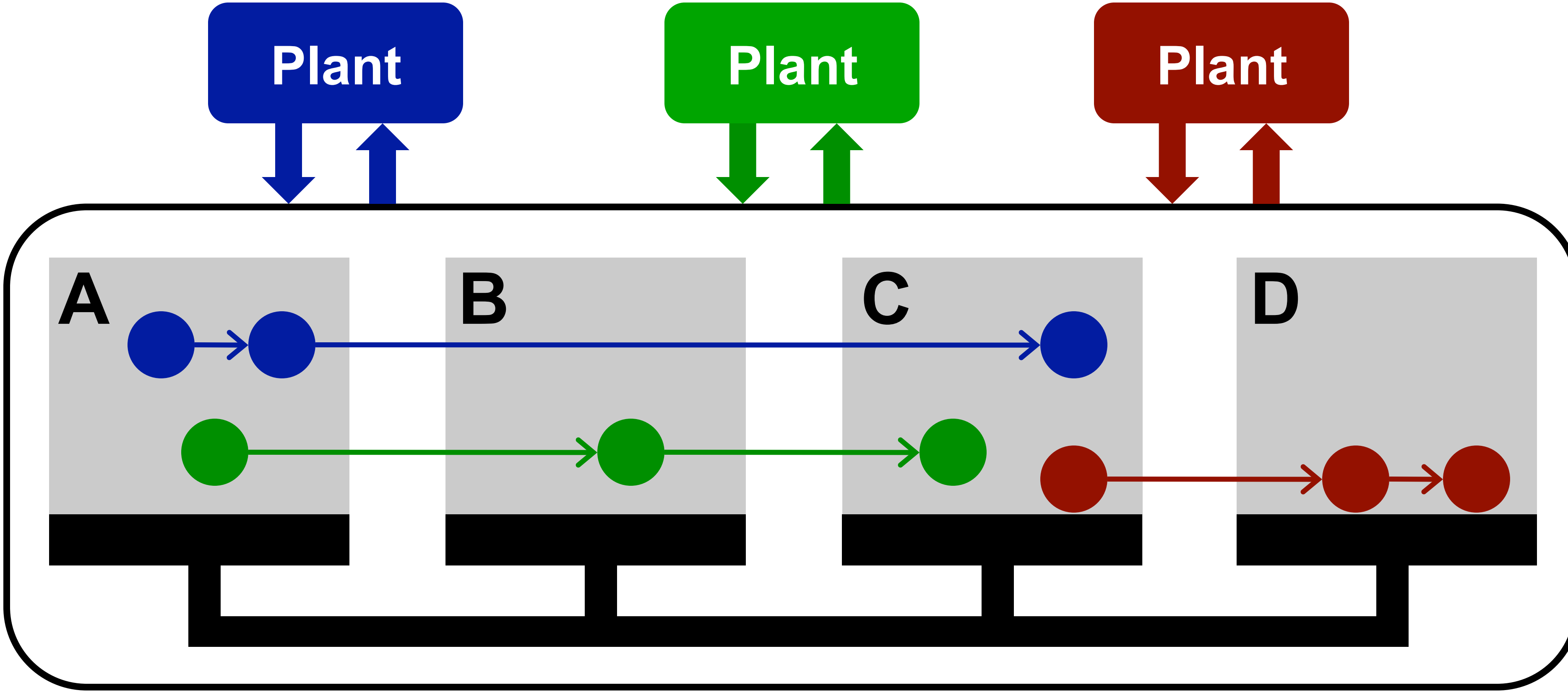
- **Quadratic cost: $J = E\{ x^T Q_1 x + u^T Q_2 u \}$**
- **Depends on**
  - **the sampling period,**
  - **the control law, and**
  - **the mapping and schedule (delays between sampling and actuation)**
- **”Jitterbug” (Lund University)**

Co-Design Tool for Distributed Control



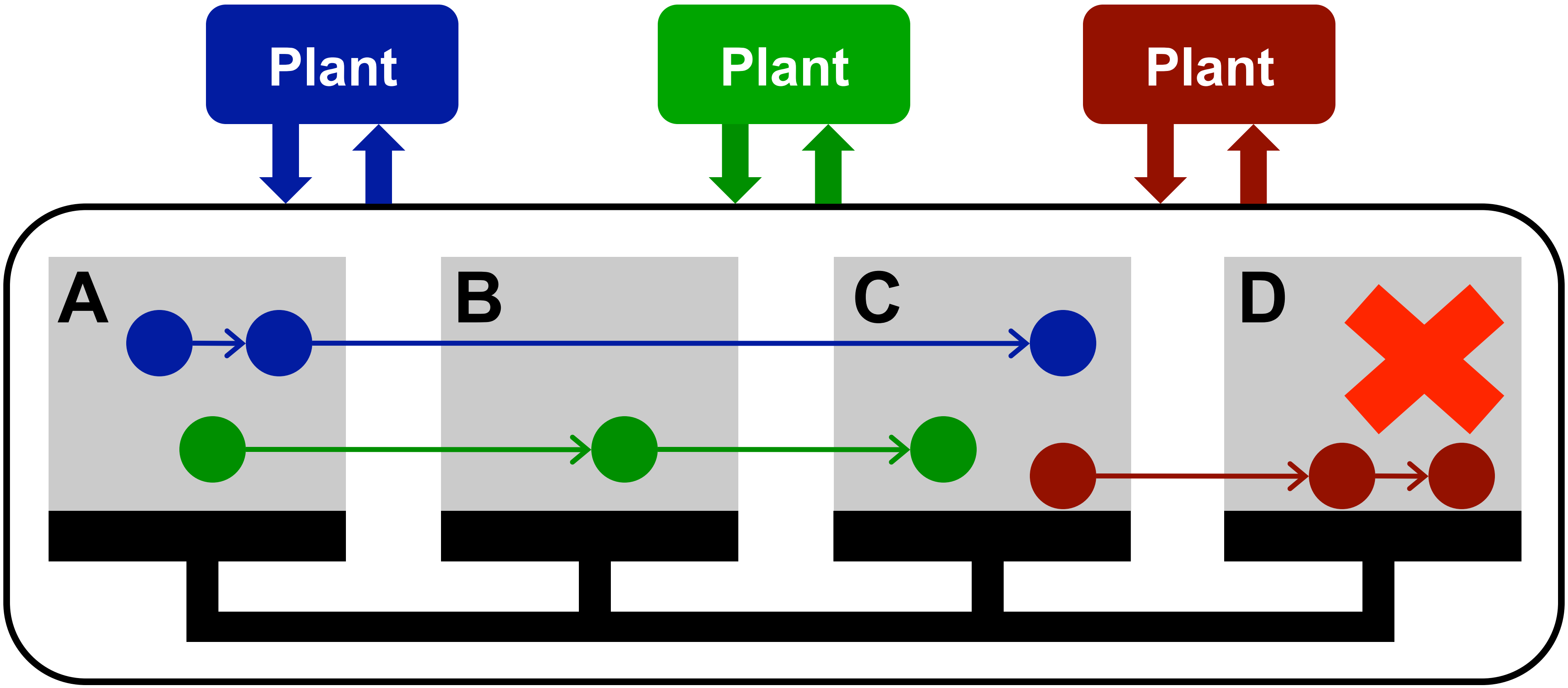
DATE 2009
RTCSA 2011

Example



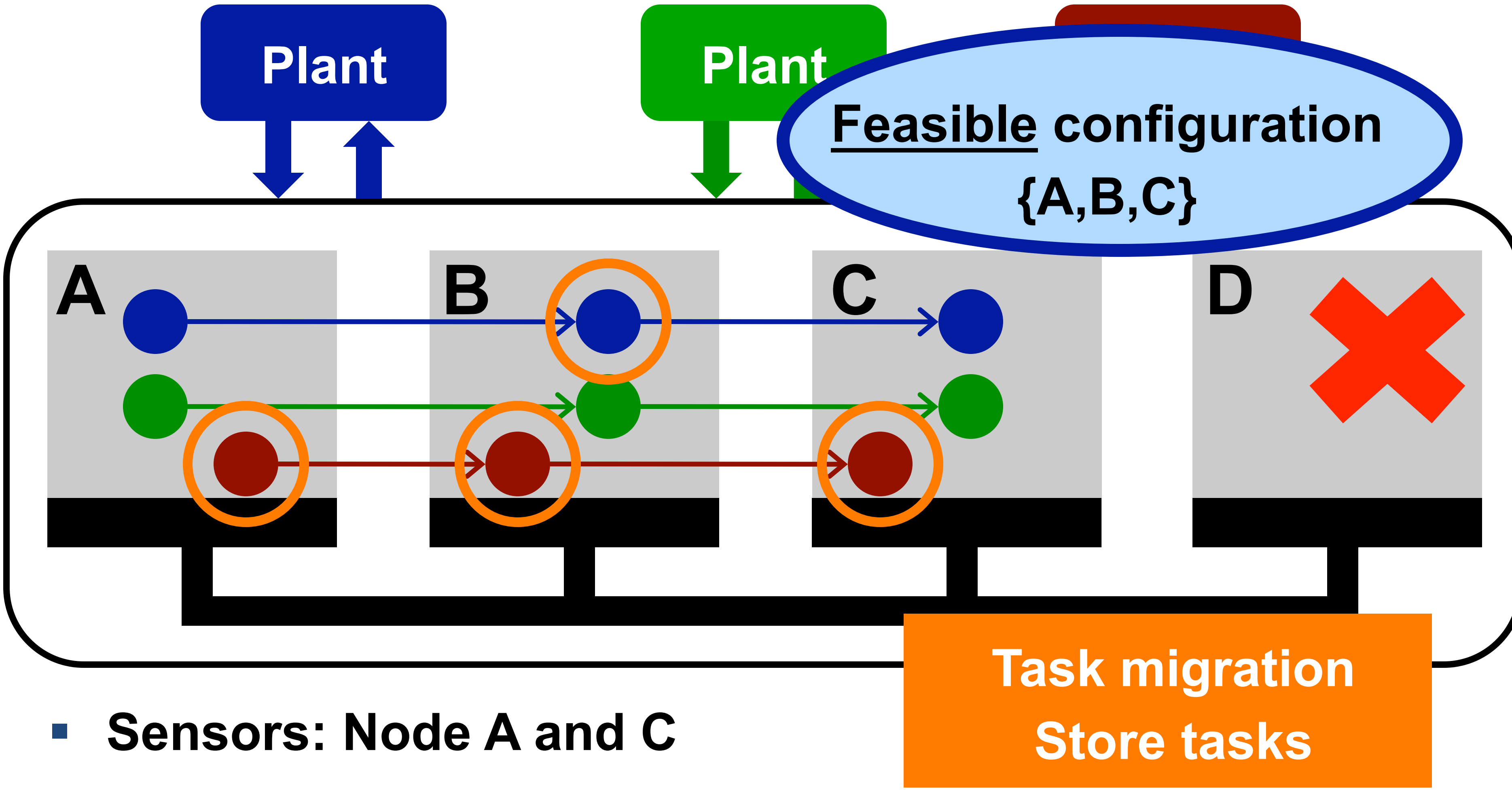
- **Sensors: Node A and C**
- **Actuators: Node C and D**

Configurations



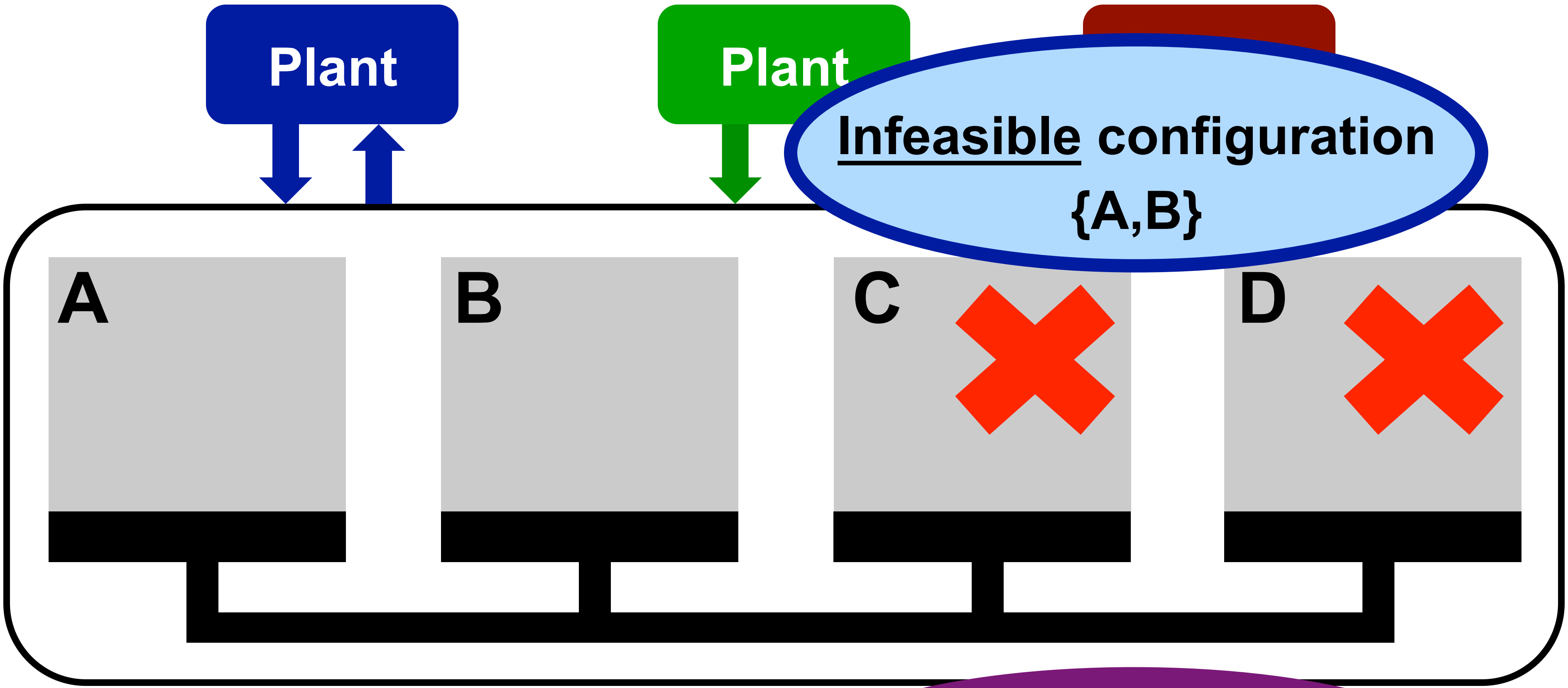
- **Sensors: Node A and C**
- **Actuators: Node C and D**

Configurations



- Sensors: Node A and C
- Actuators: Node C and D

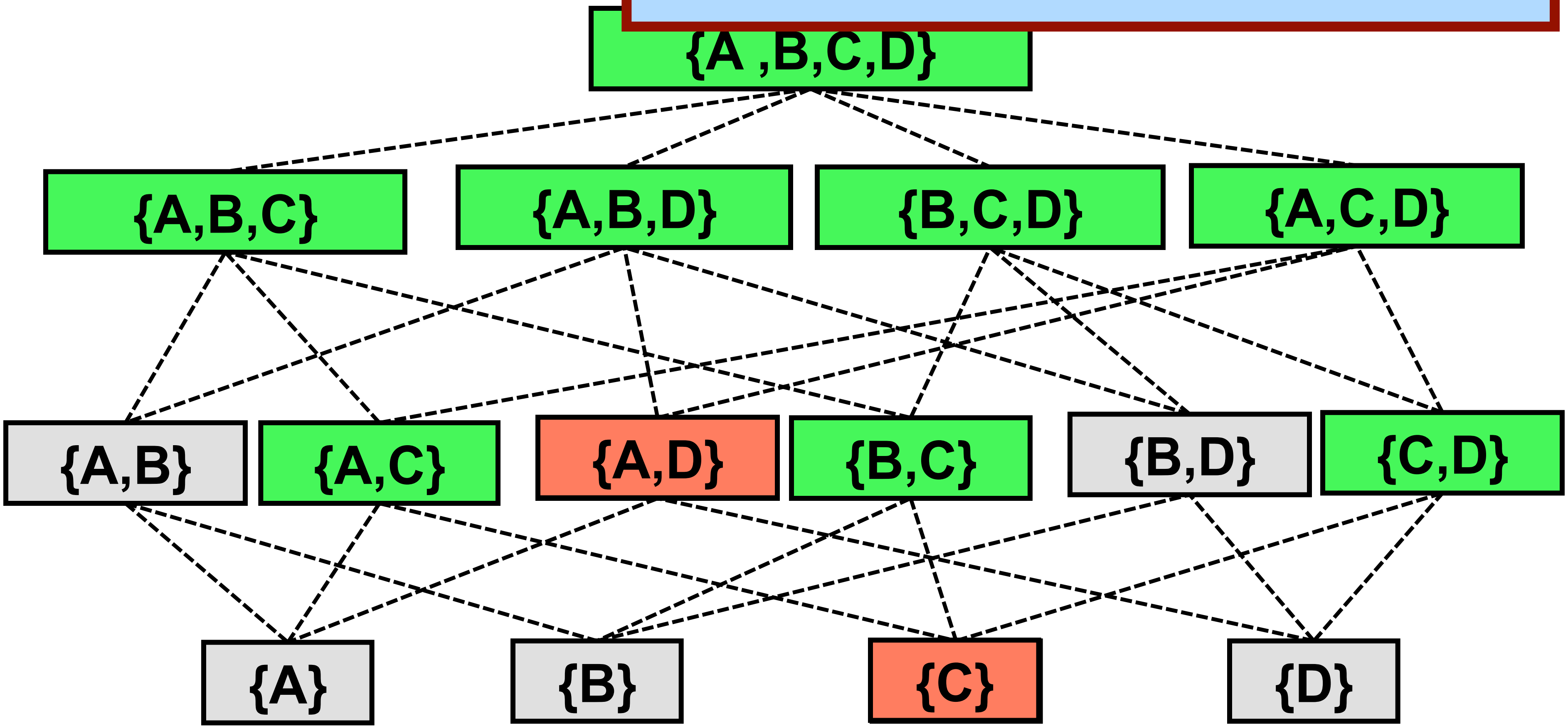
Configurations



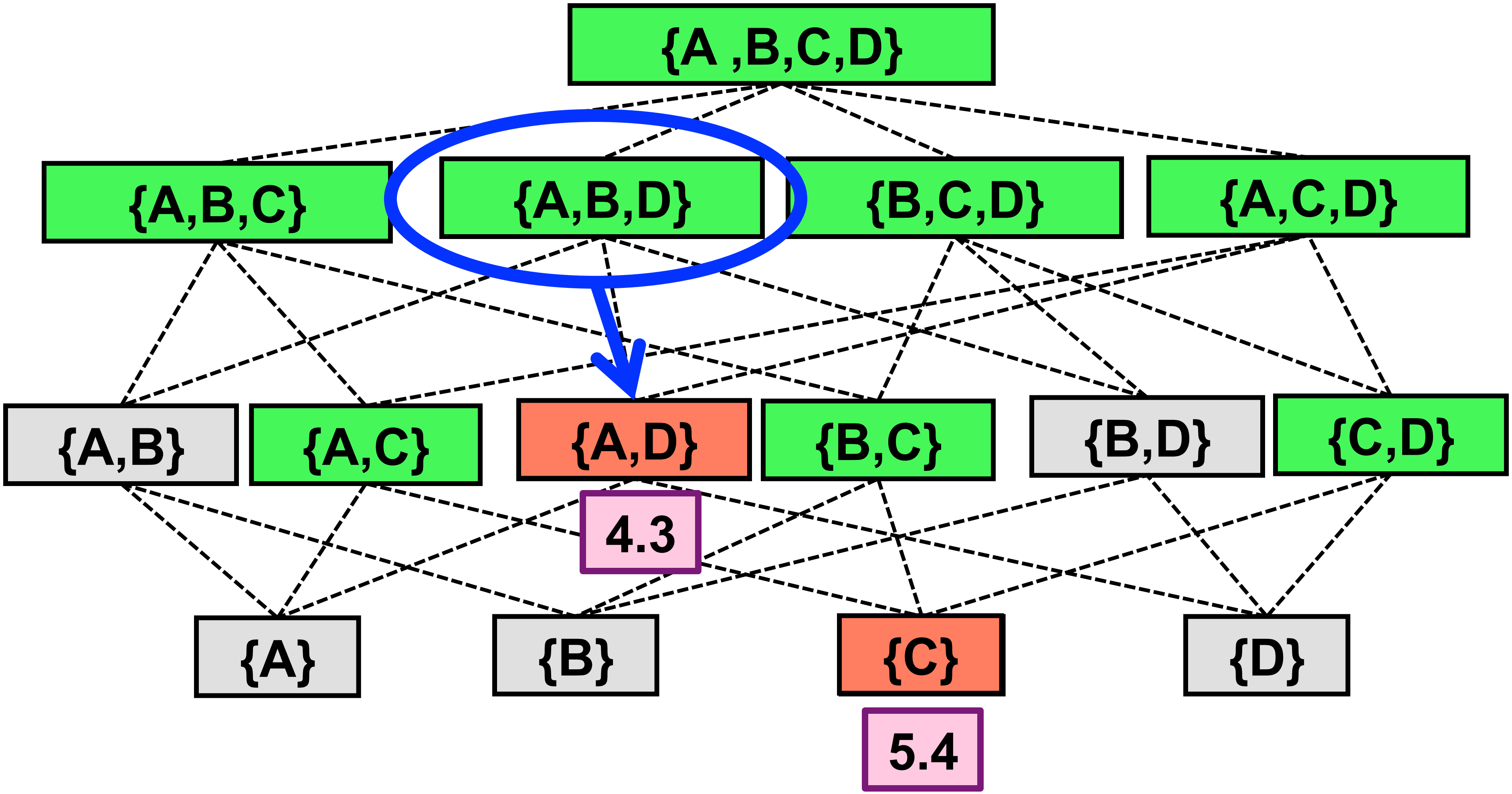
- Sensors: Node A and C
- Actuators: Node C and D

Configurations

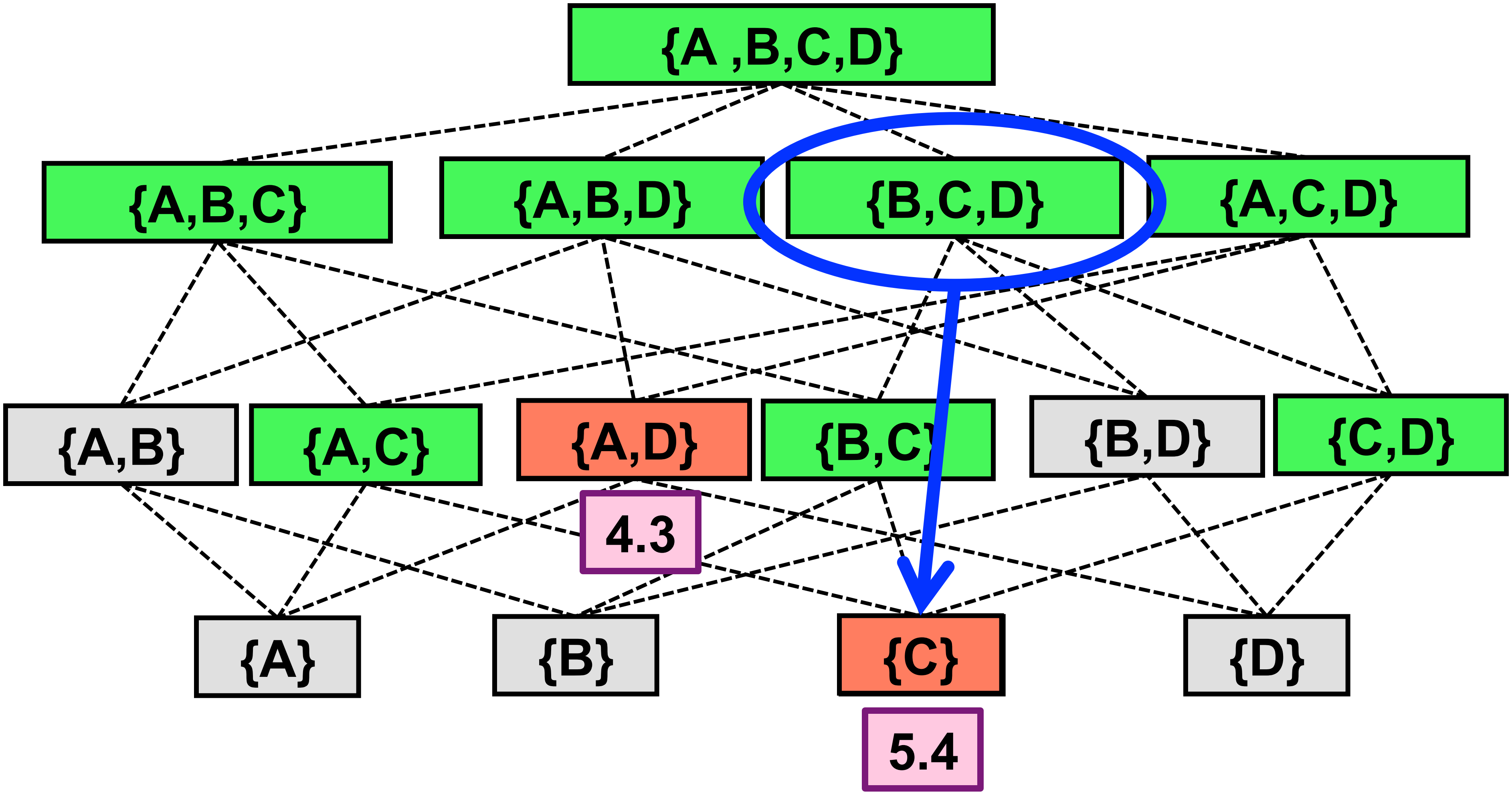
- **Synthesize mapping, schedule, and control laws for each <u>base</u> configuration**



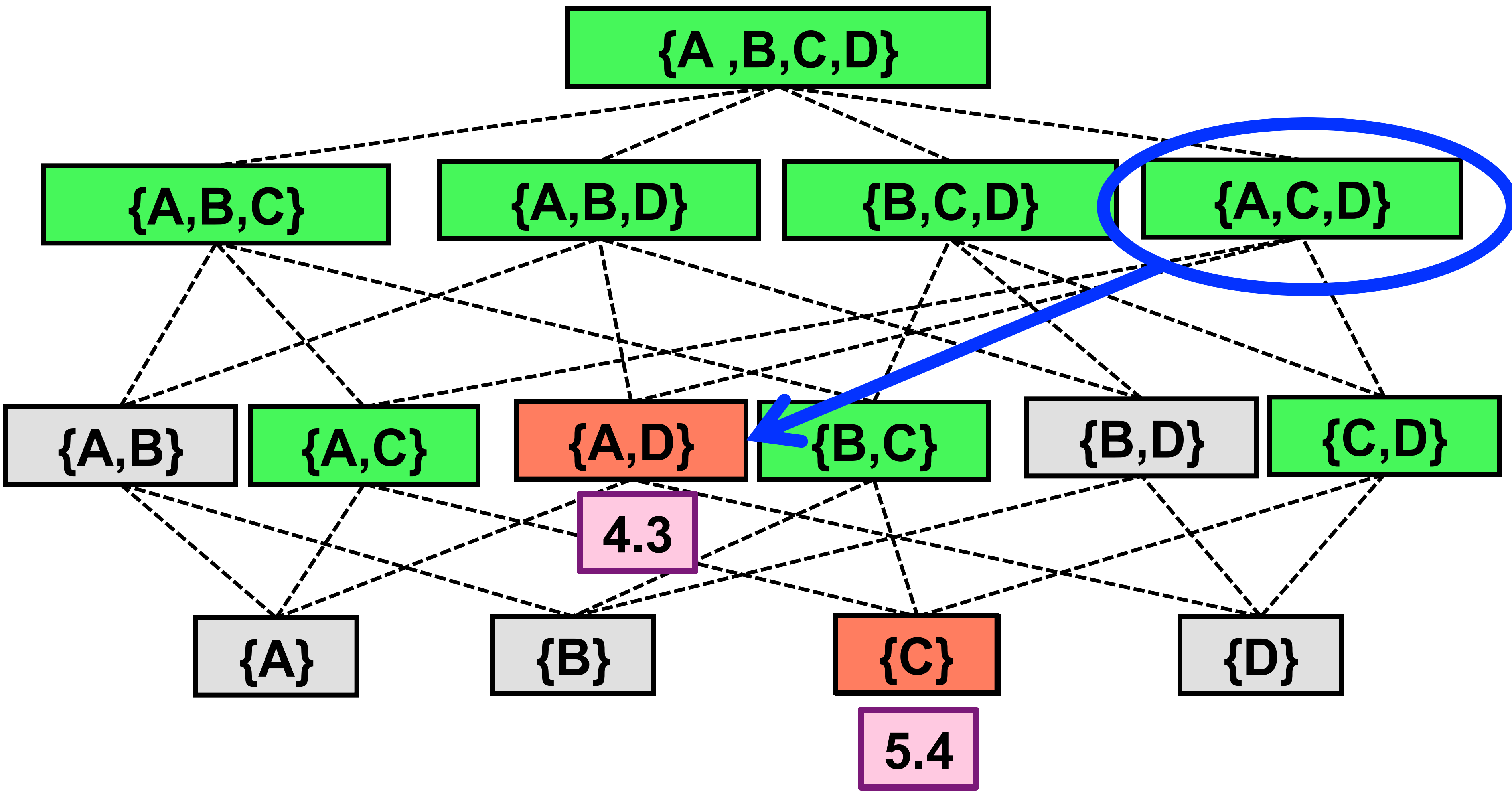
Configurations



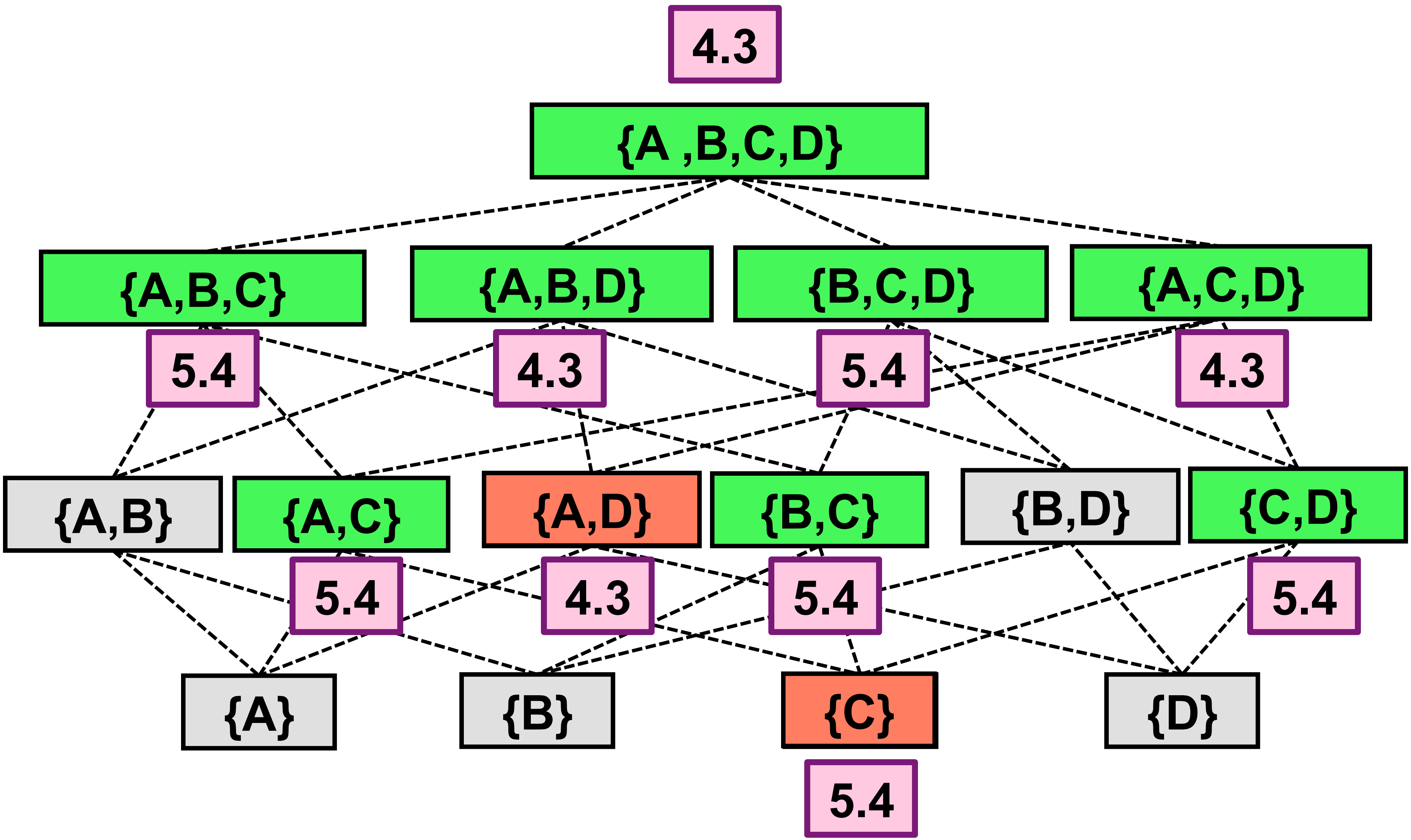
Configurations



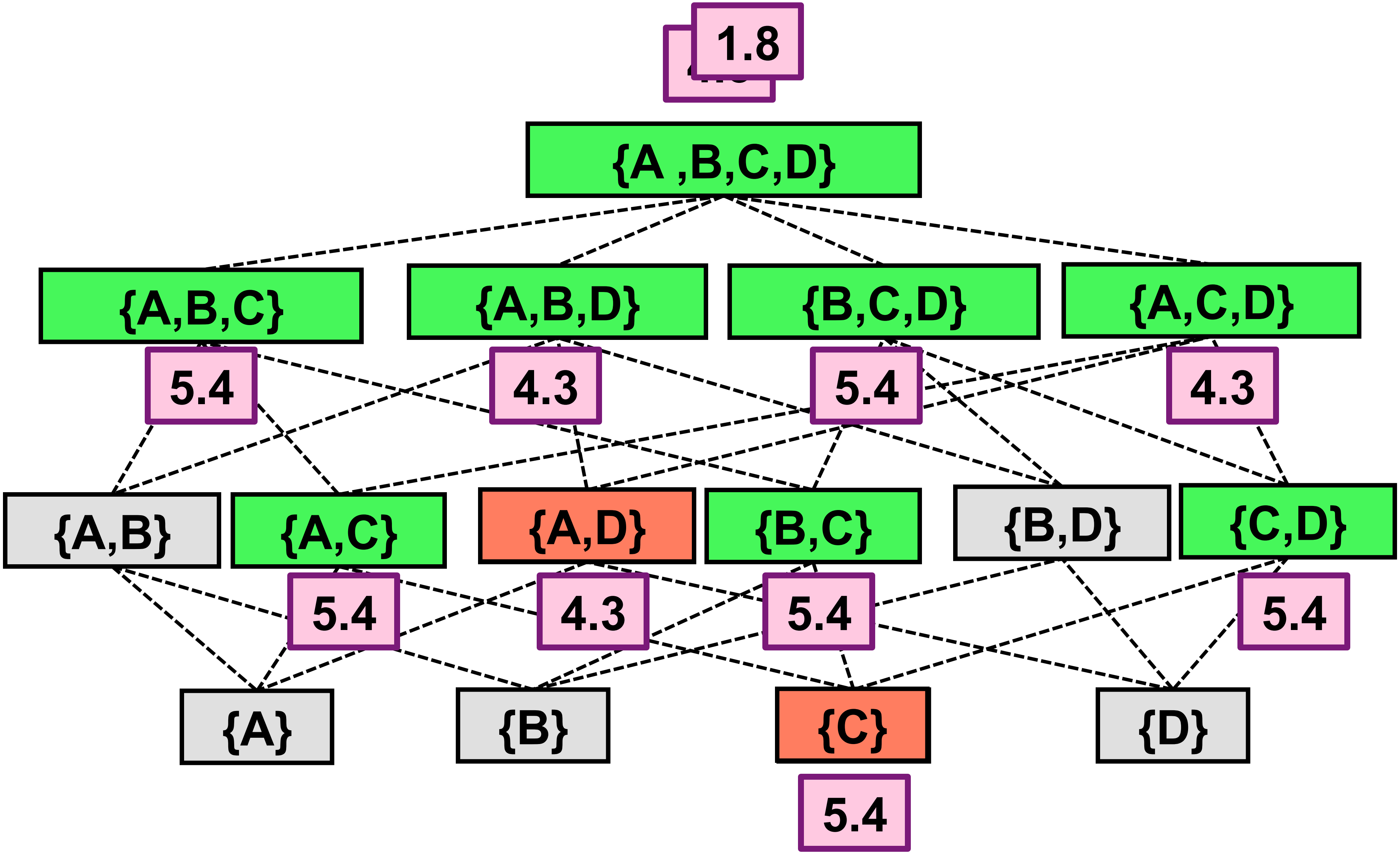
Configurations



Configurations



Configurations



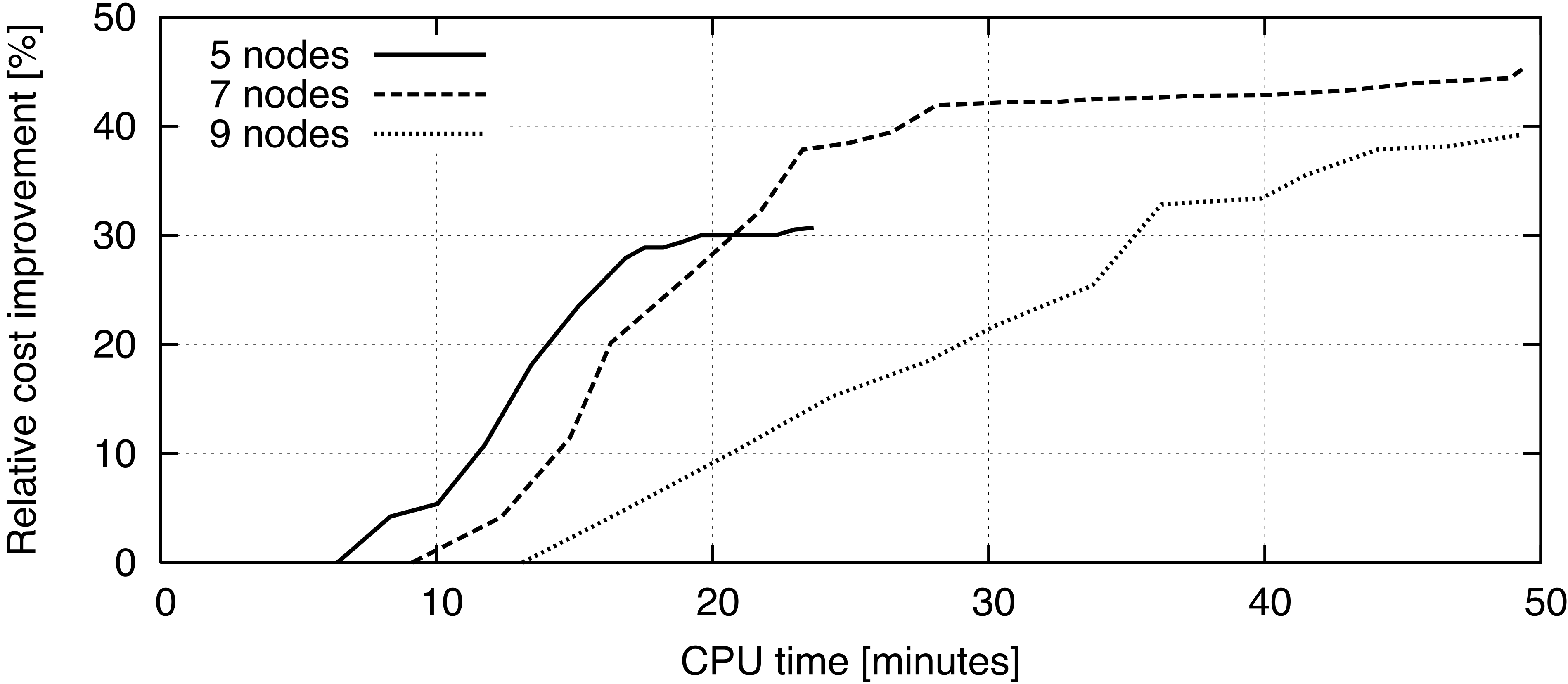
Optimization

- **Construct solutions for additional configurations (heuristic considers node failure probabilities)**
    - **Trade-offs: control quality, design time**

- **Mapping realization (ILP formulation)**
  - **Task migration (time constraint, overhead)**
  - **Store tasks on nodes (memory constraint)**

- **Cost function to minimize:** $\sum_{C} p_C \cdot J_C$
  - p_C: **Probability of reaching configuration** C

Experiments



Conclusions

- **Faults lead to different configurations**
  - **Not practical to design a customized solution to each configuration**

- **Synthesize solutions to a subset of all configurations in order to achieve a level of fault tolerance given by the available sensor/actuator interfaces and capacity of the platform**

- **Optimization method for control-quality improvements in the configurations**