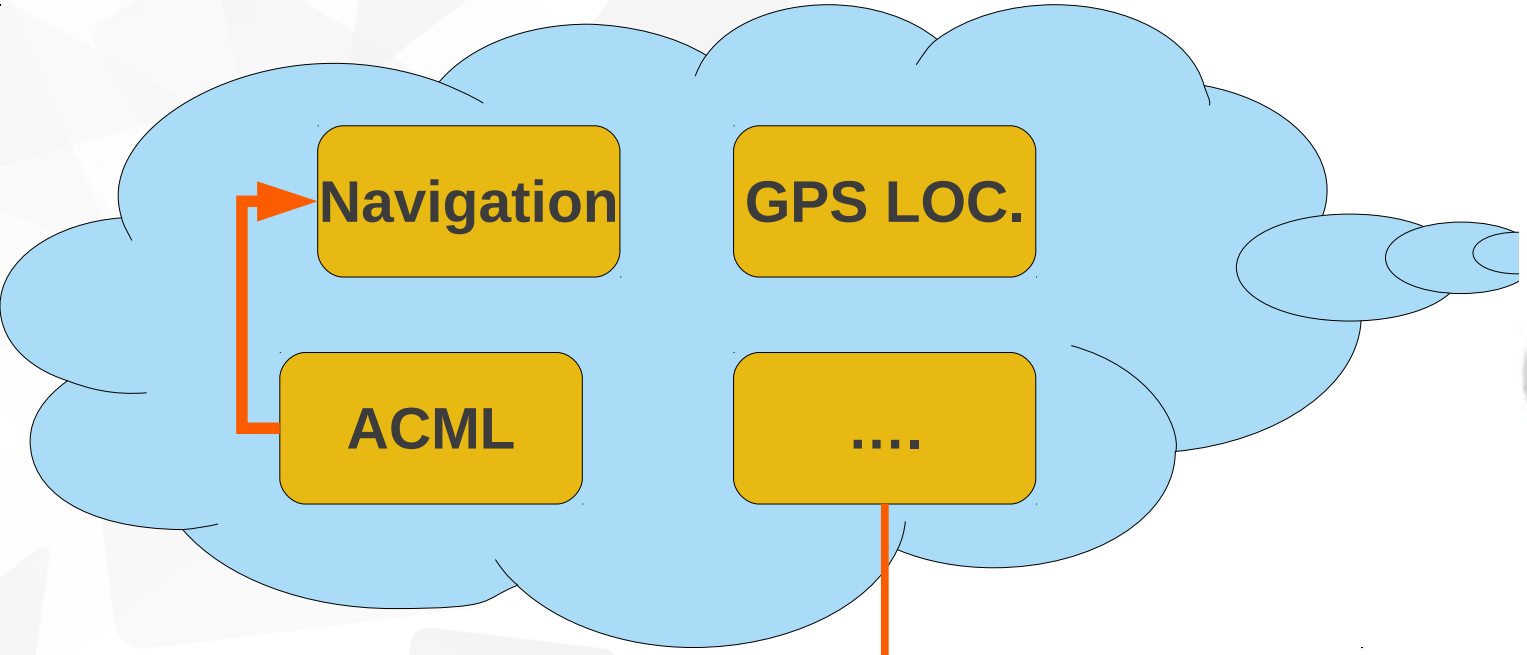
Closed Loop Configuration Planning with Time and Resources

M. Di Rocco F. Pecora A. Saffiotti

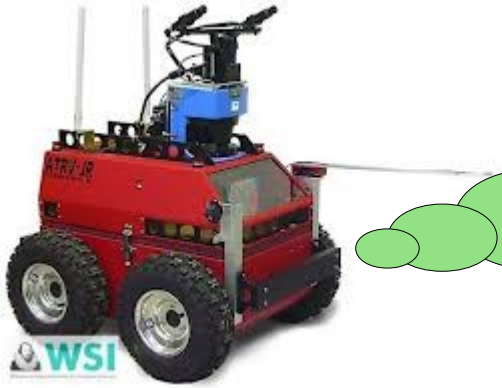
COGNITIVE ROBOTIC SYSTEMS

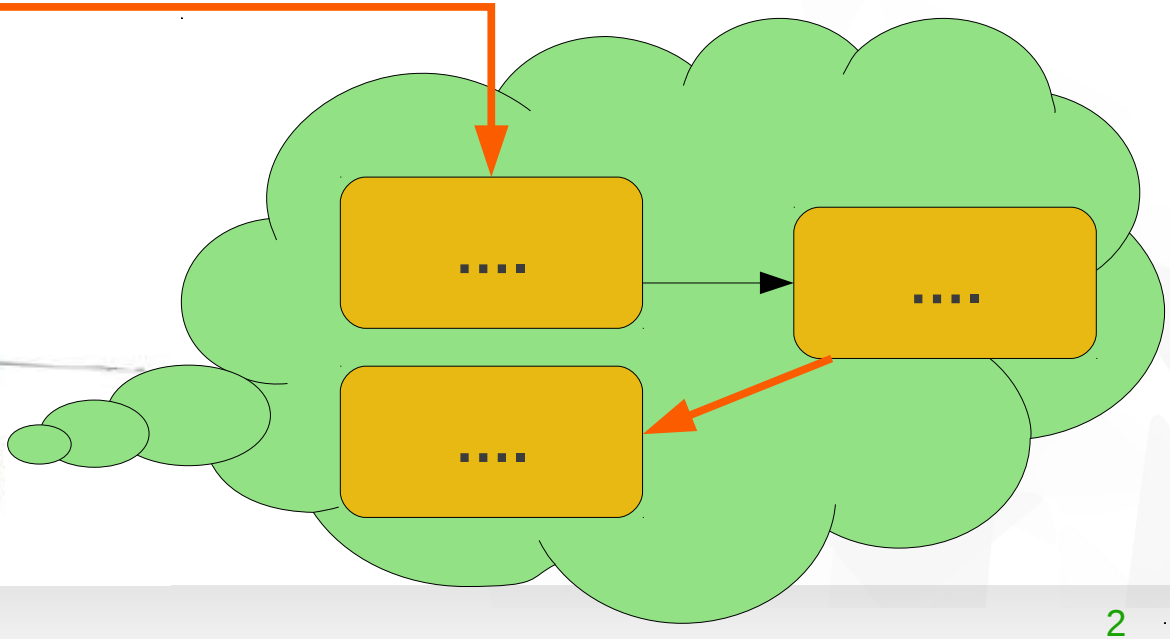
Center for Applied Autonomous Sensor Systems (AASS)
Örebro University, SWEDEN

Configuration... multi robot scenario



Robots run **interacting software modules**





A simple example

Propagating consequences...

40 mins delay led to 43H trip!

Planning...

Each subsystem can:

- Interact with the environment (**causal constraints**), e.g. light on
- Take time into consideration(**temporal constraints**), e.g. deadlines, rendez-vous
- Exploit resources (**resource constraints**) e.g. a pan-tilt camera, CPU usage
- Exchange information (**information dependencies**) e.g. motion controller must know position of robot

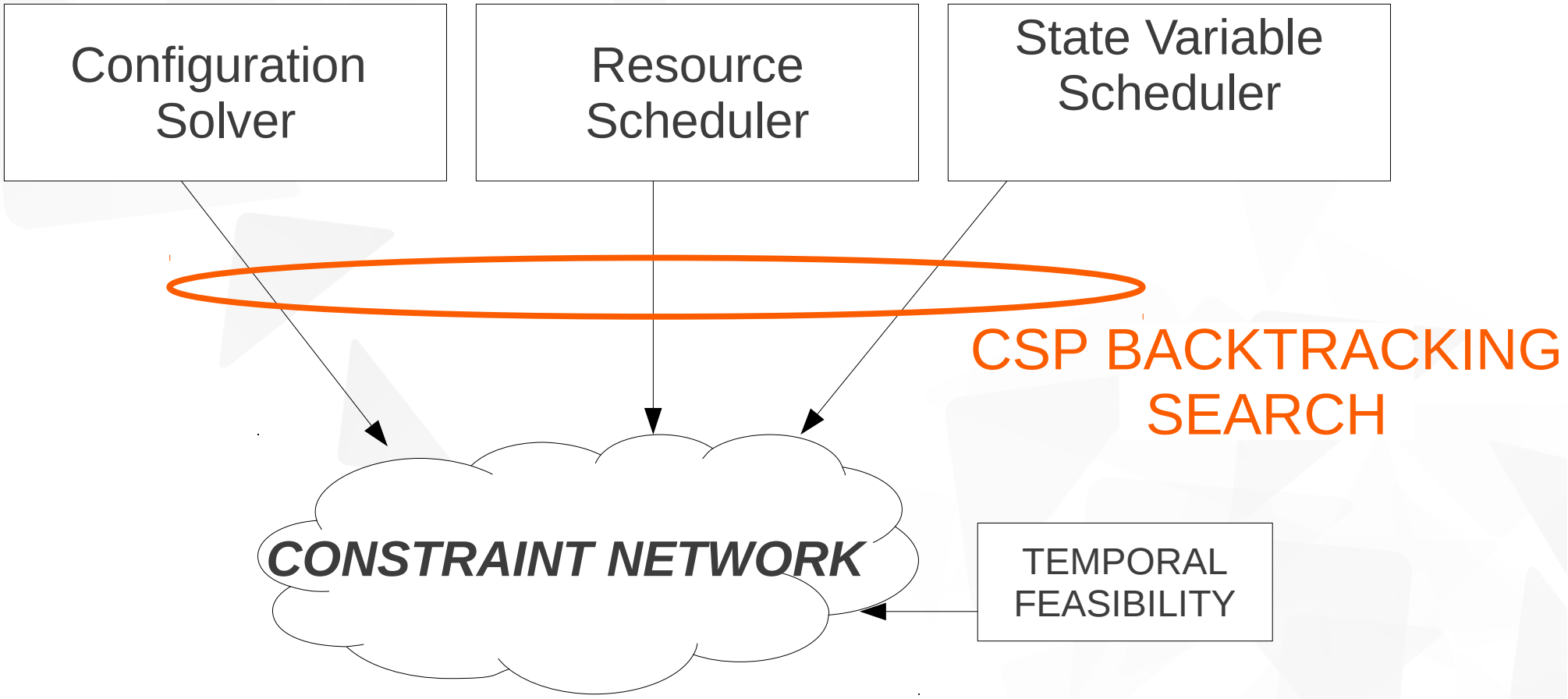
Closed loop

Most fundamental features:

- Adapting to the state of the environment
- Dynamic goals
- React to contingencies at adequate levels

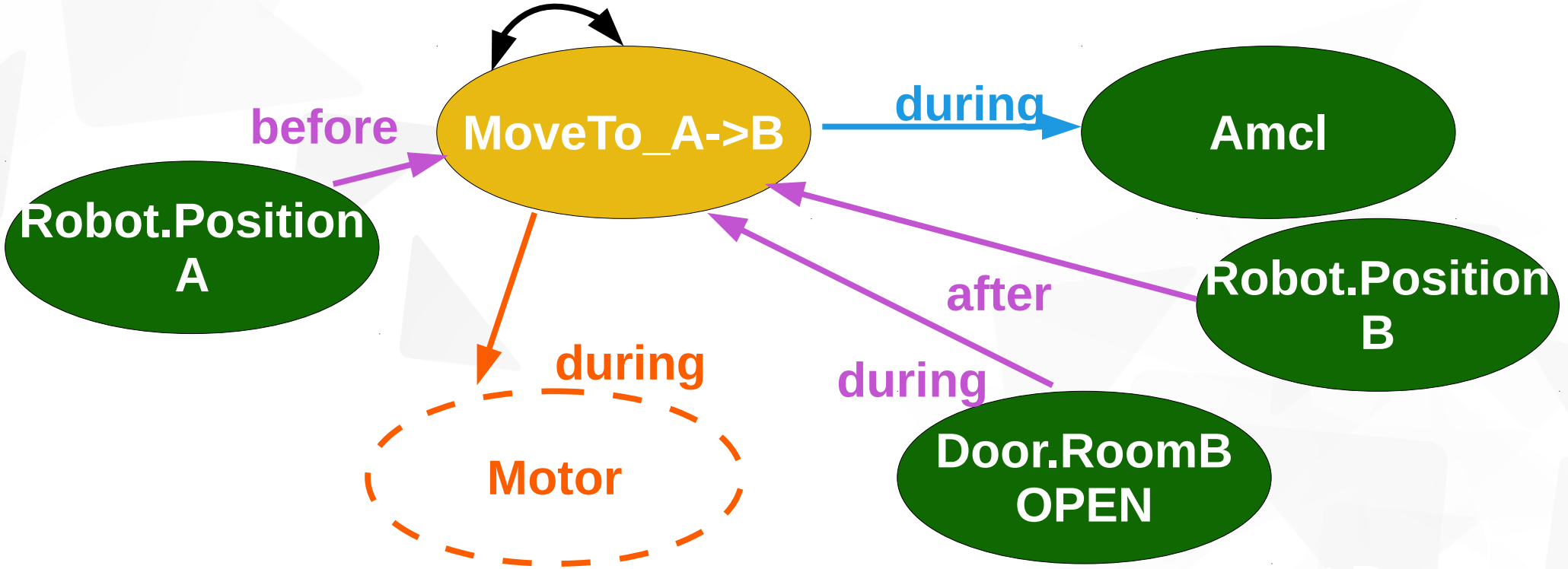
Architecture

Several solvers reason about the same constraint network



Activities

Activities **exchange information**, consume **resources** and require **causal requirements**



Configuration process

Activities need requisites about:

- Information
- Environment (EV)

Requisites can be specified through recipes

Operator

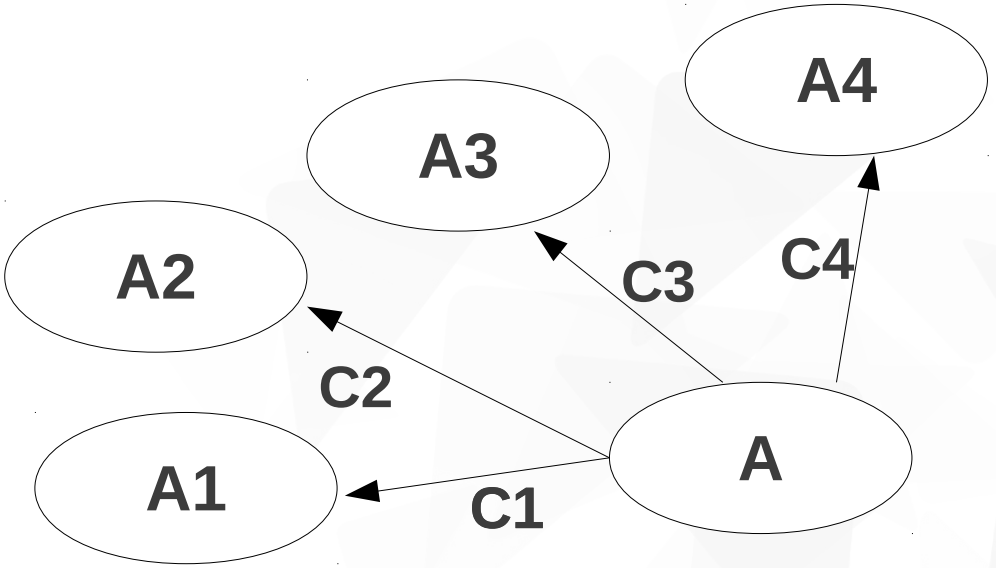
HEAD: (*MoveTo_entrance->kitchen*,*ON*, …, *...*,*NULL*)

A:

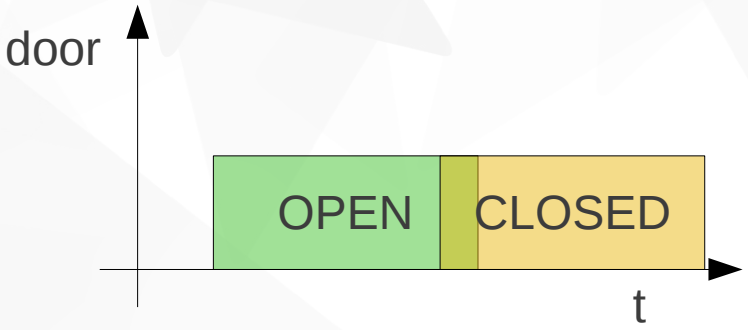
A1	(*...*, *...*, *...*,*....*, *Robot.position*)
A2	(*...*, *...*, *...*,*....*, *Range_data*)
A3	(*Robot.Location*, *entrance*, *...*,*....*, *...*)
A4	(*Robot.Location*, *kitchen*, *...*,*....*, *…*)

C:

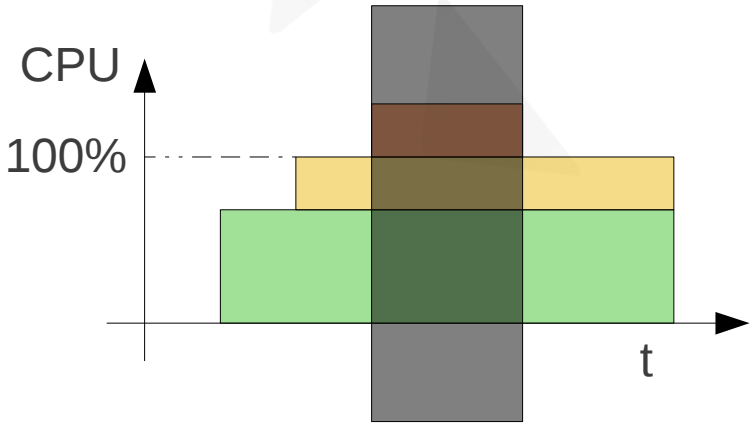
C1	*During*
C2	*During*
C3	*After*
C4	*Before*



Conflicts...

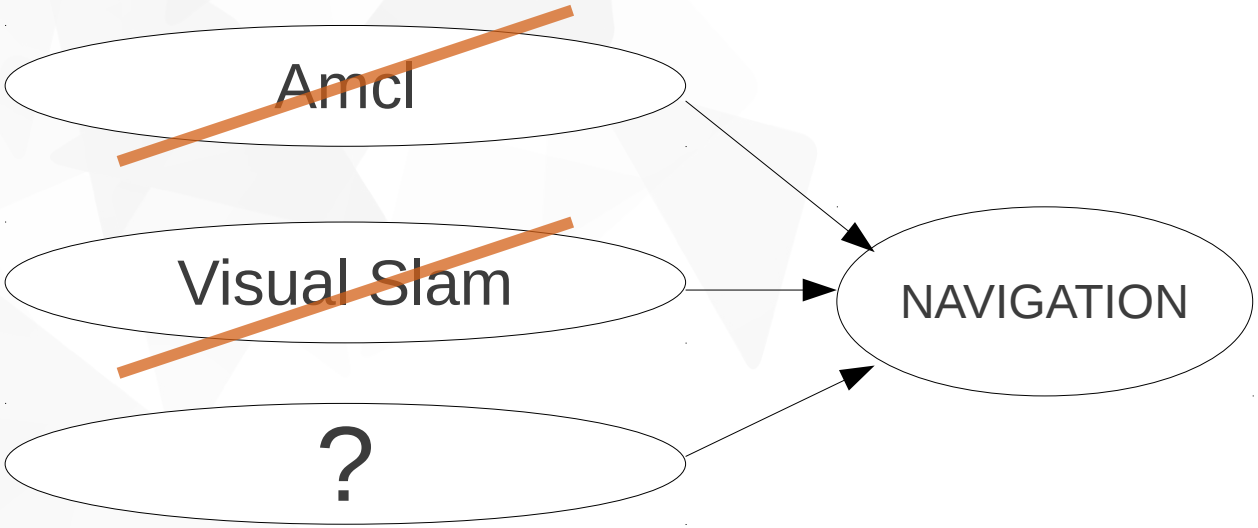


STATE VARIABLE INCONSISTENCIES

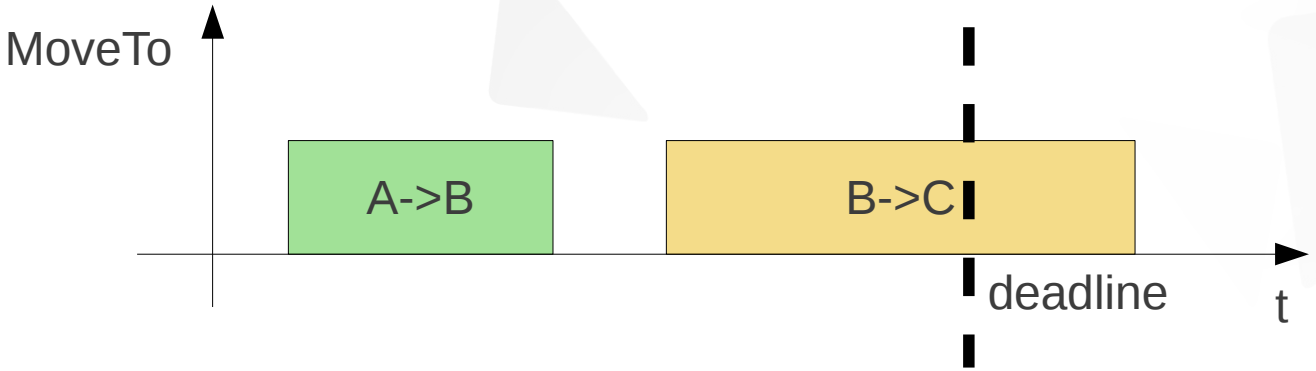


RESOURCE OVERUSE

Conflicts...



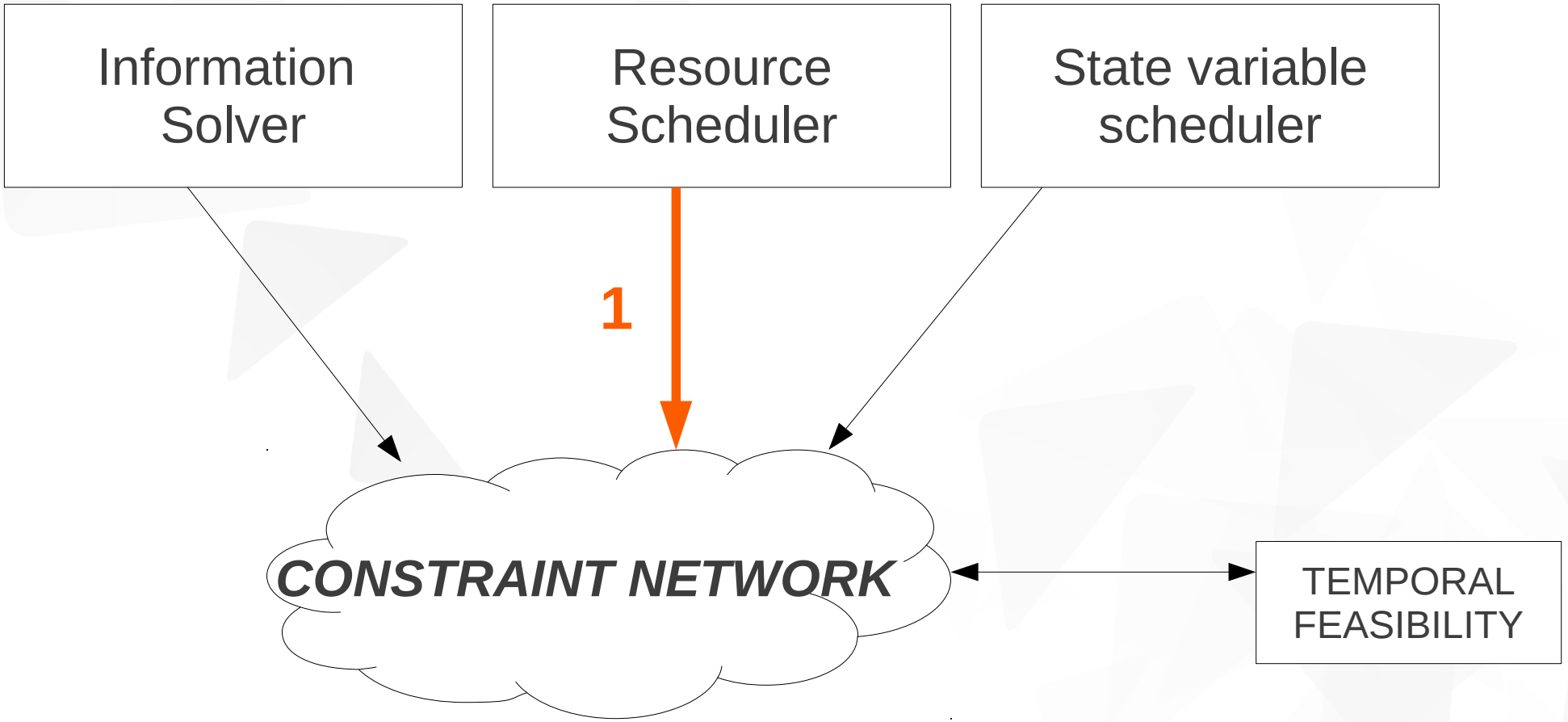
INFORMATION INCONSISTENCIES



TEMPORAL INCONSISTENCIES

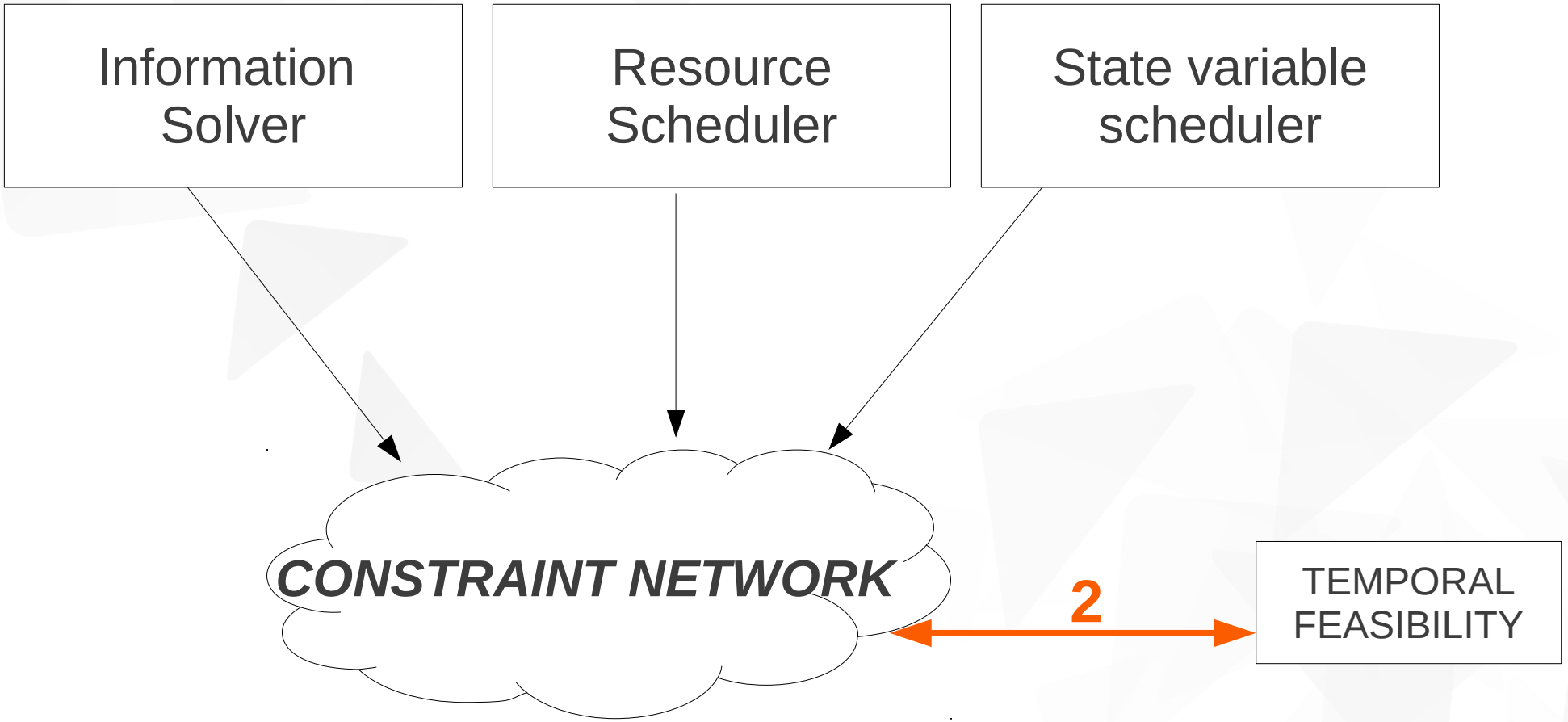
Different solvers...

Backtracking techniques handle conflicts:



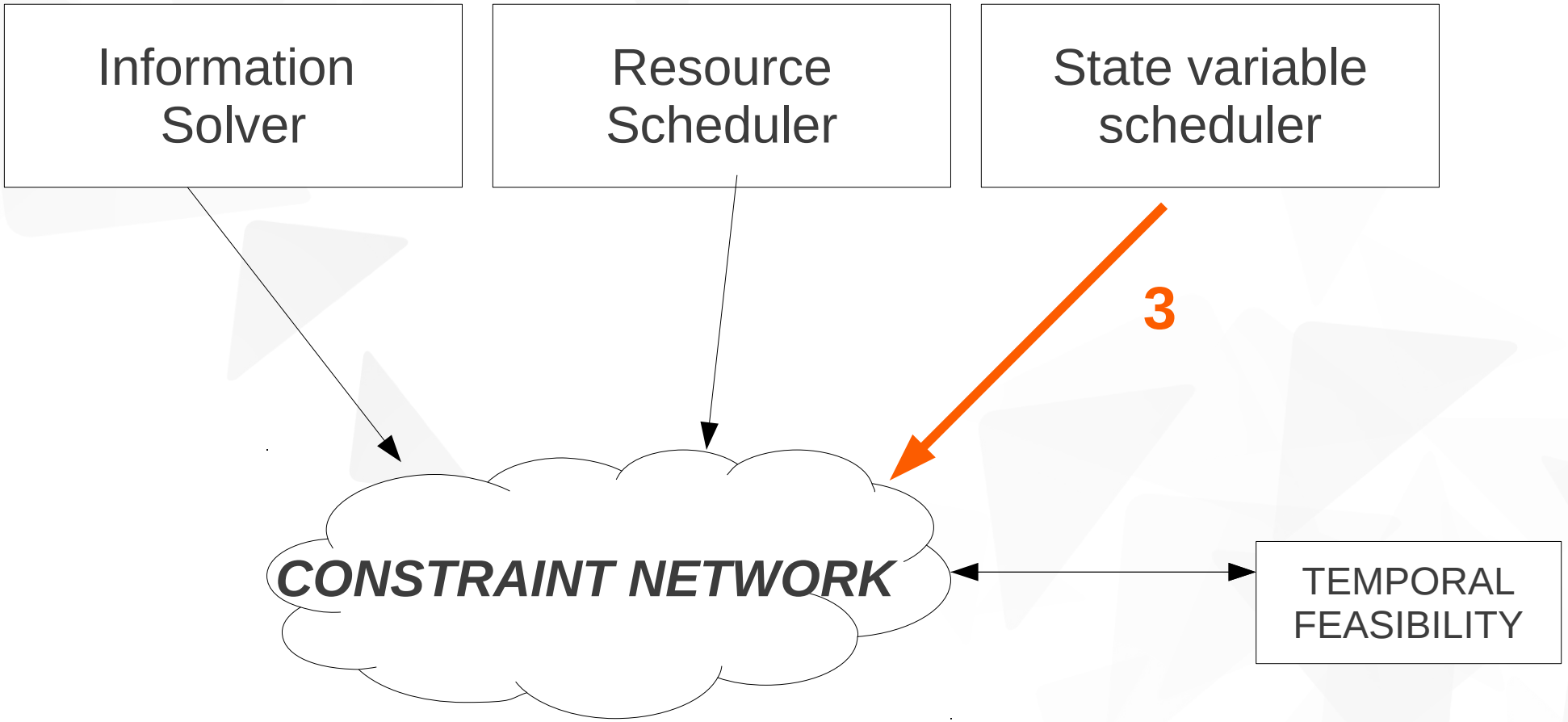
Different solvers...

Backtracking techniques handle conflicts:



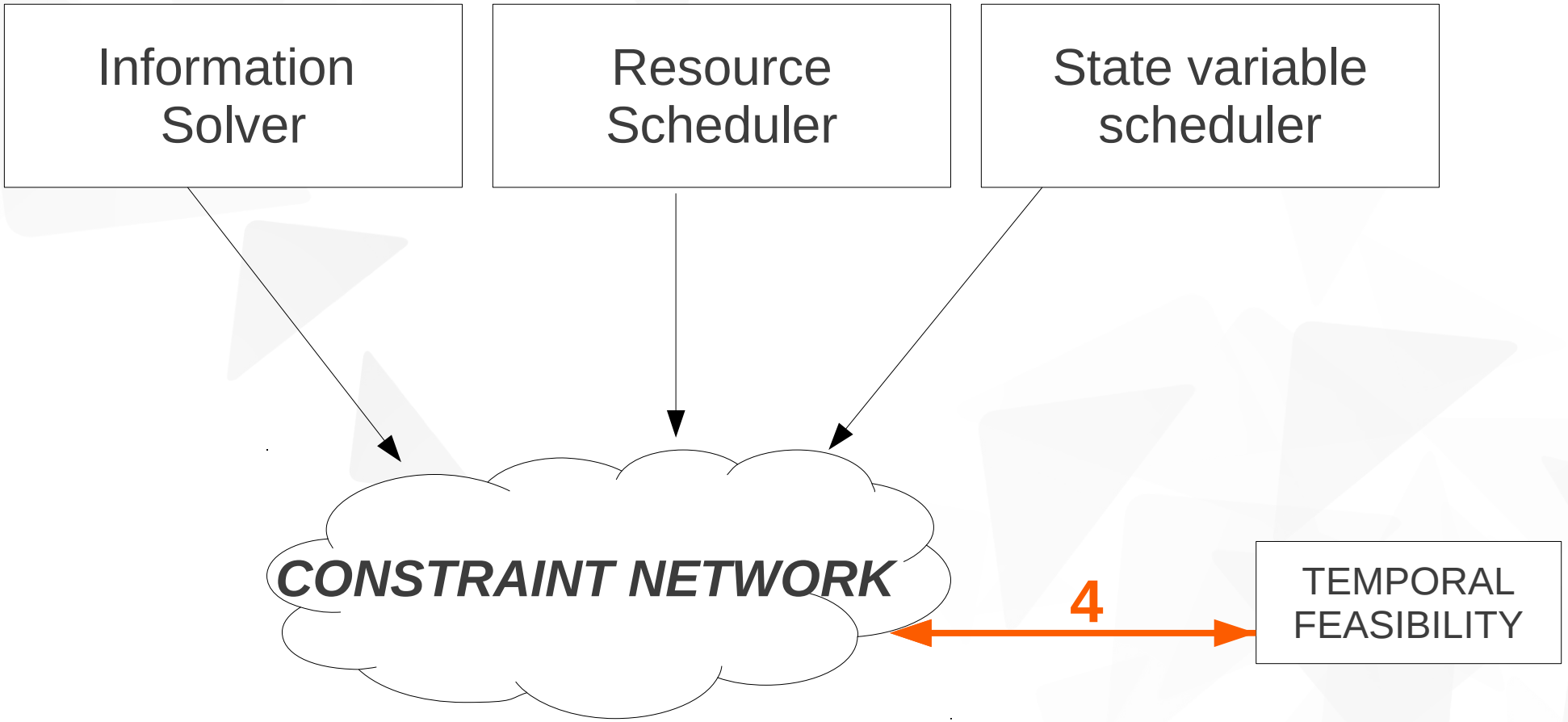
Different solvers...

Backtracking techniques handle conflicts:



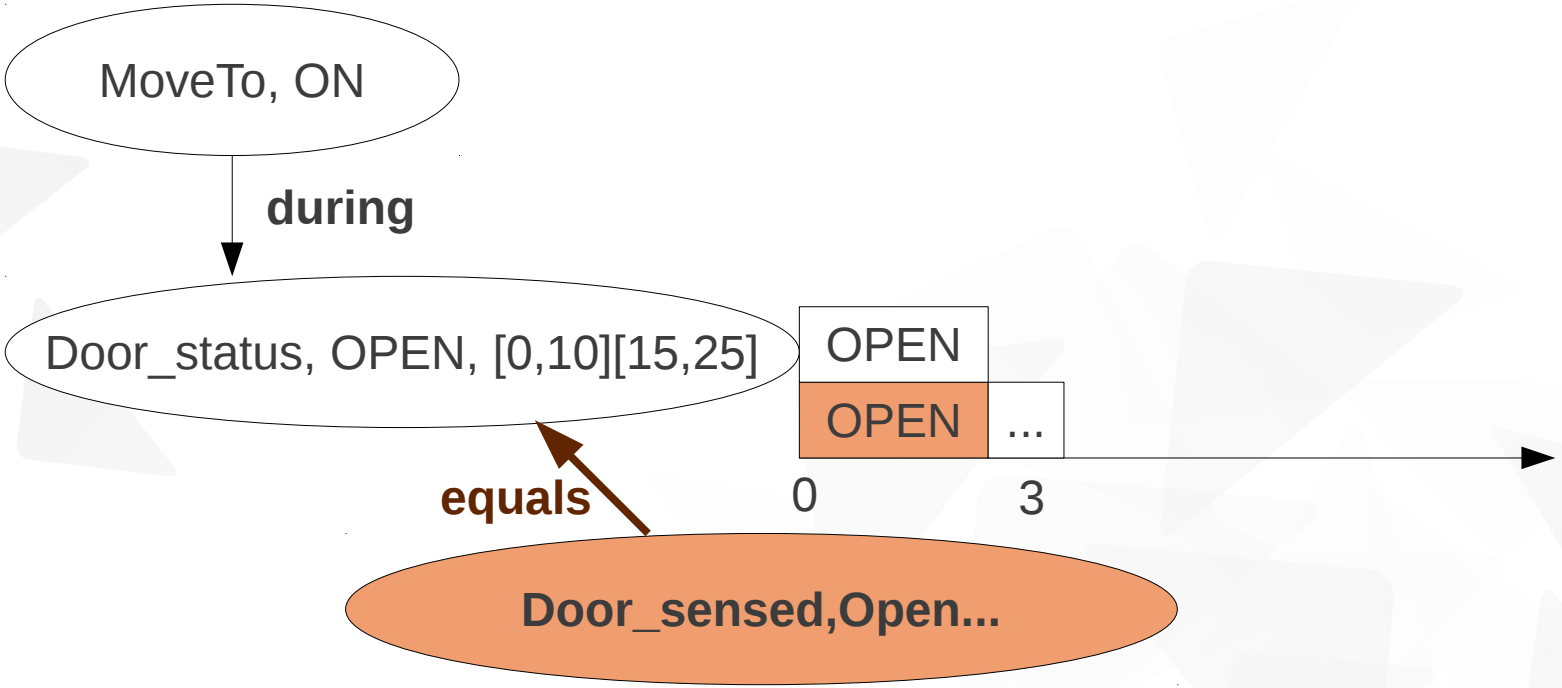
Different solvers...

Backtracking techniques handle conflicts:



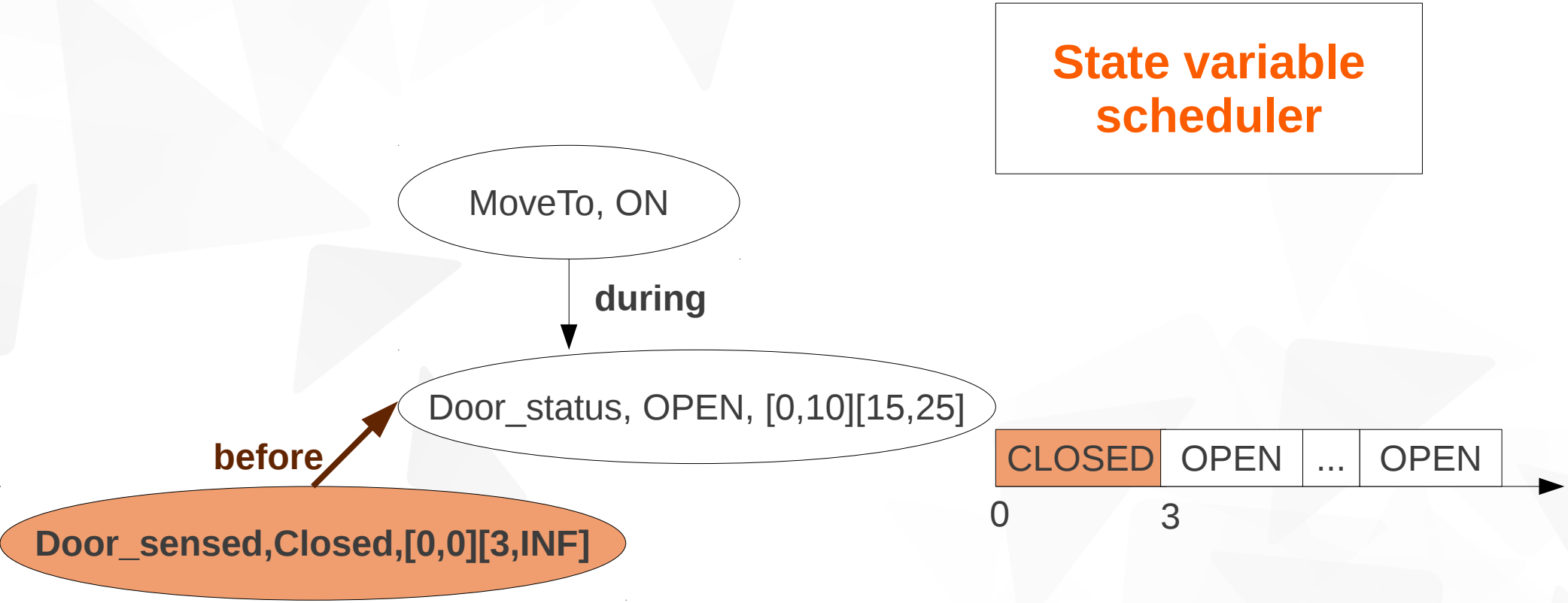
Execution observer – How?

Perception is **integrated** into the same **constraint network**

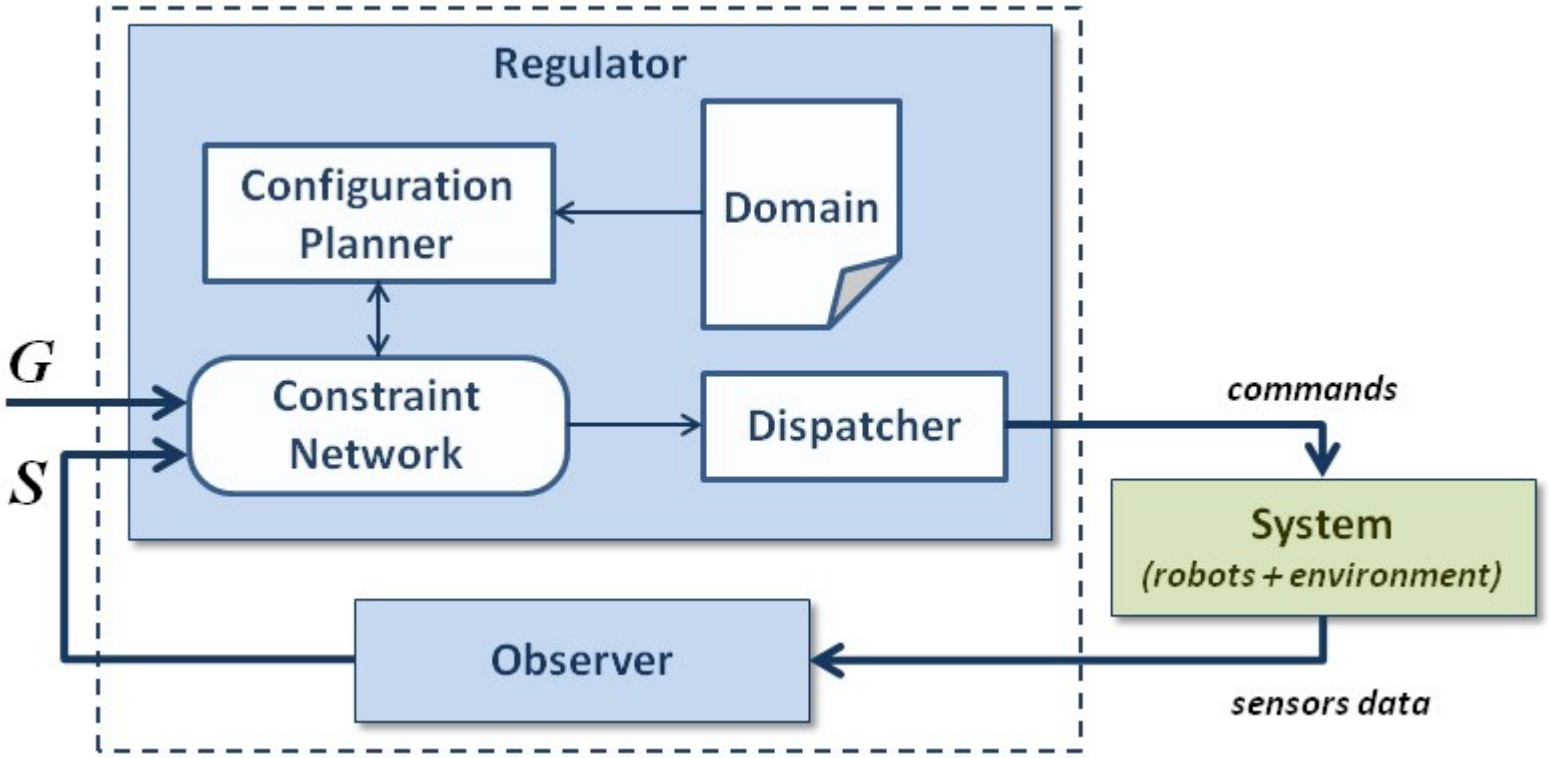


Execution observer – How?

Perception is **integrated** into the same **constraint network**



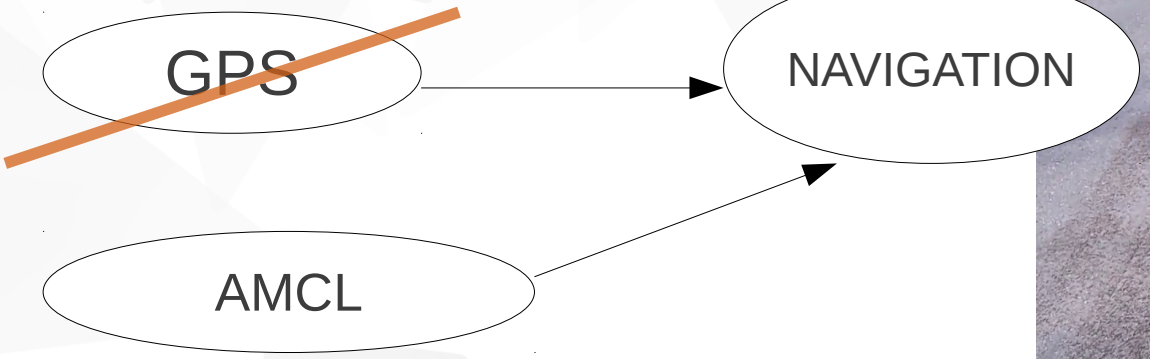
Closed-loop architecture

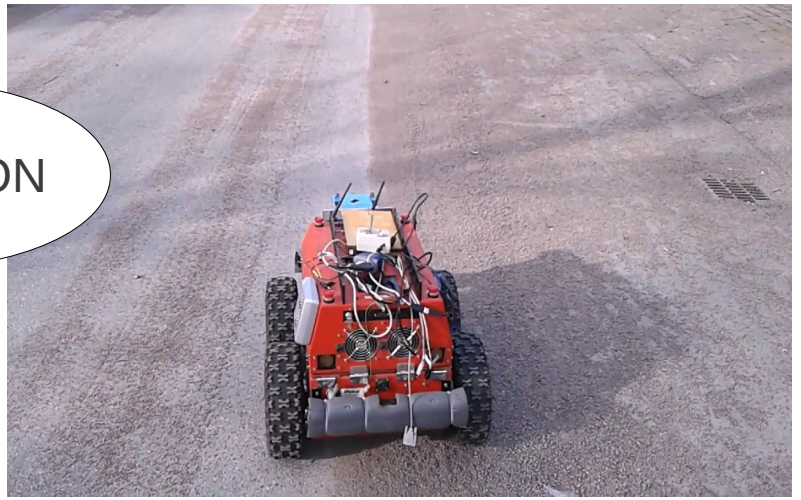


It is possible to describe the plan progress in terms of moves from the goal

Conflicts at execution time

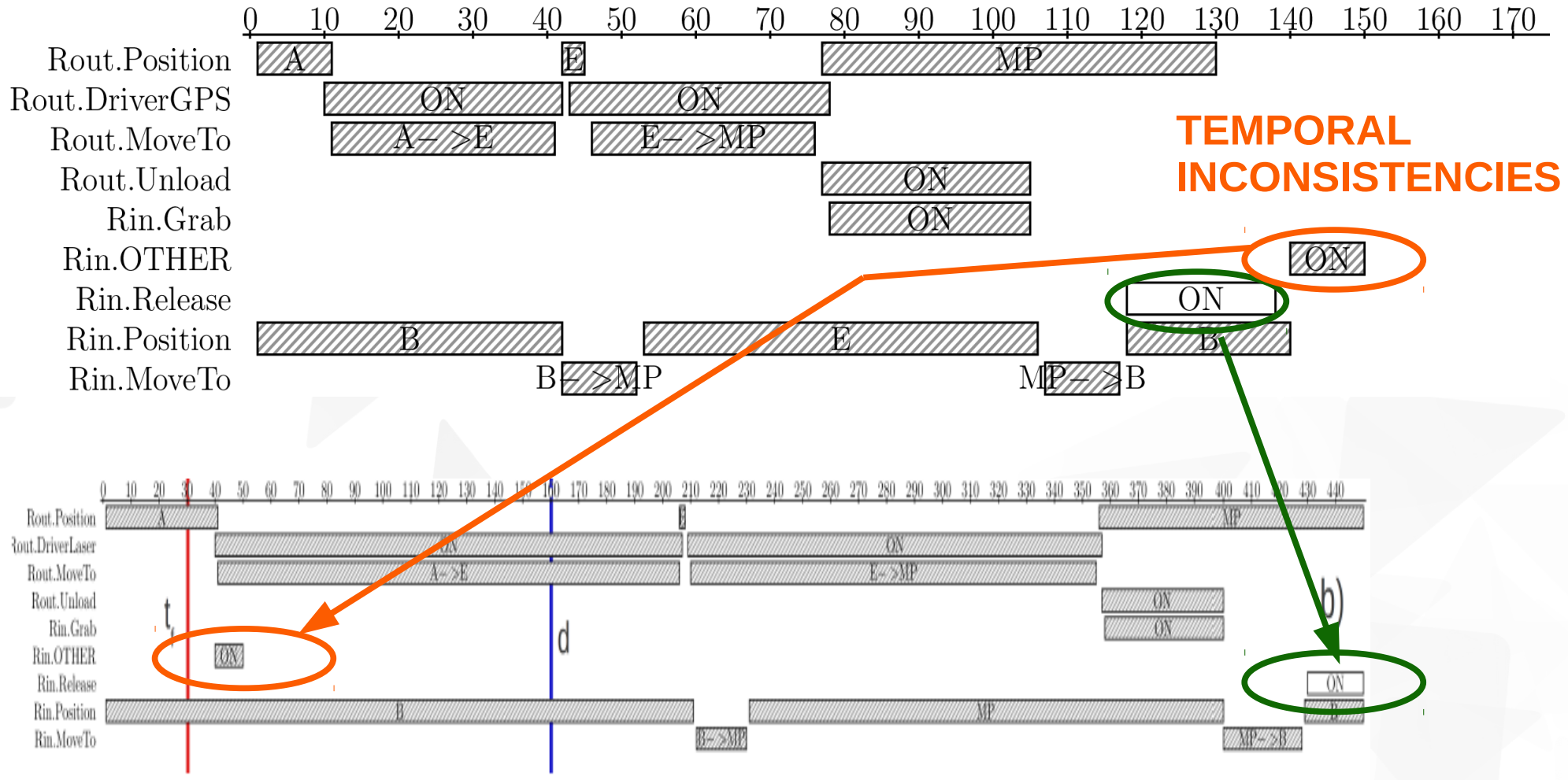
Conflicts are handled by multiple solvers





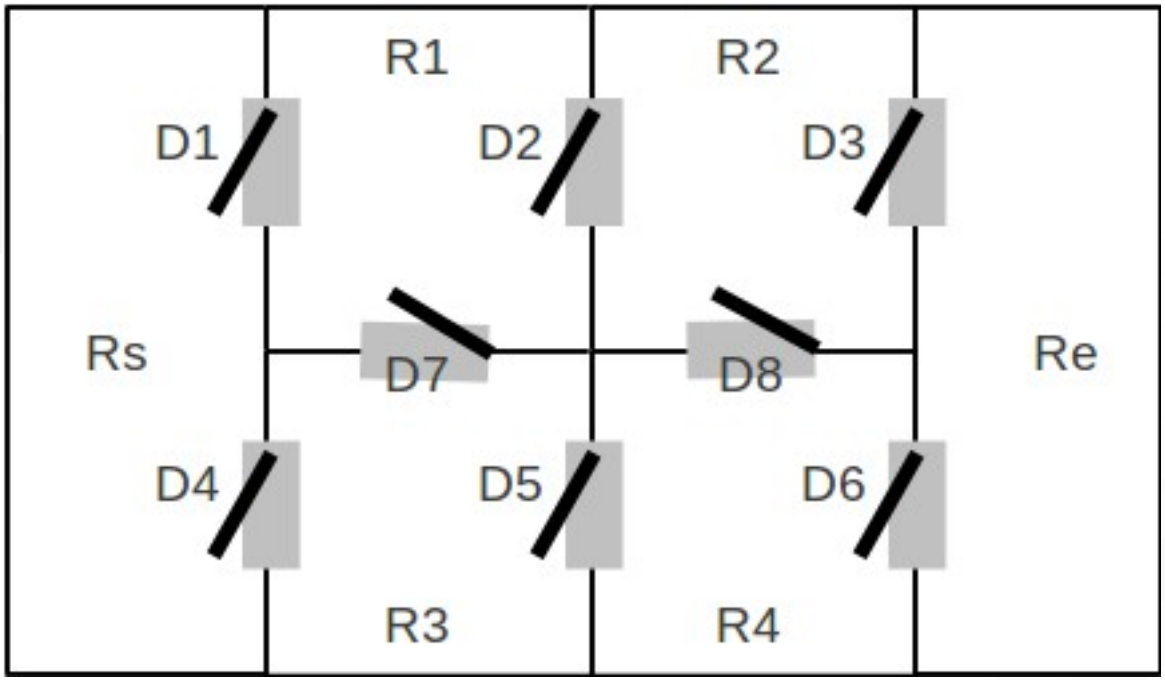
INFORMATION INCONSISTENCIES

Conflicts at execution time



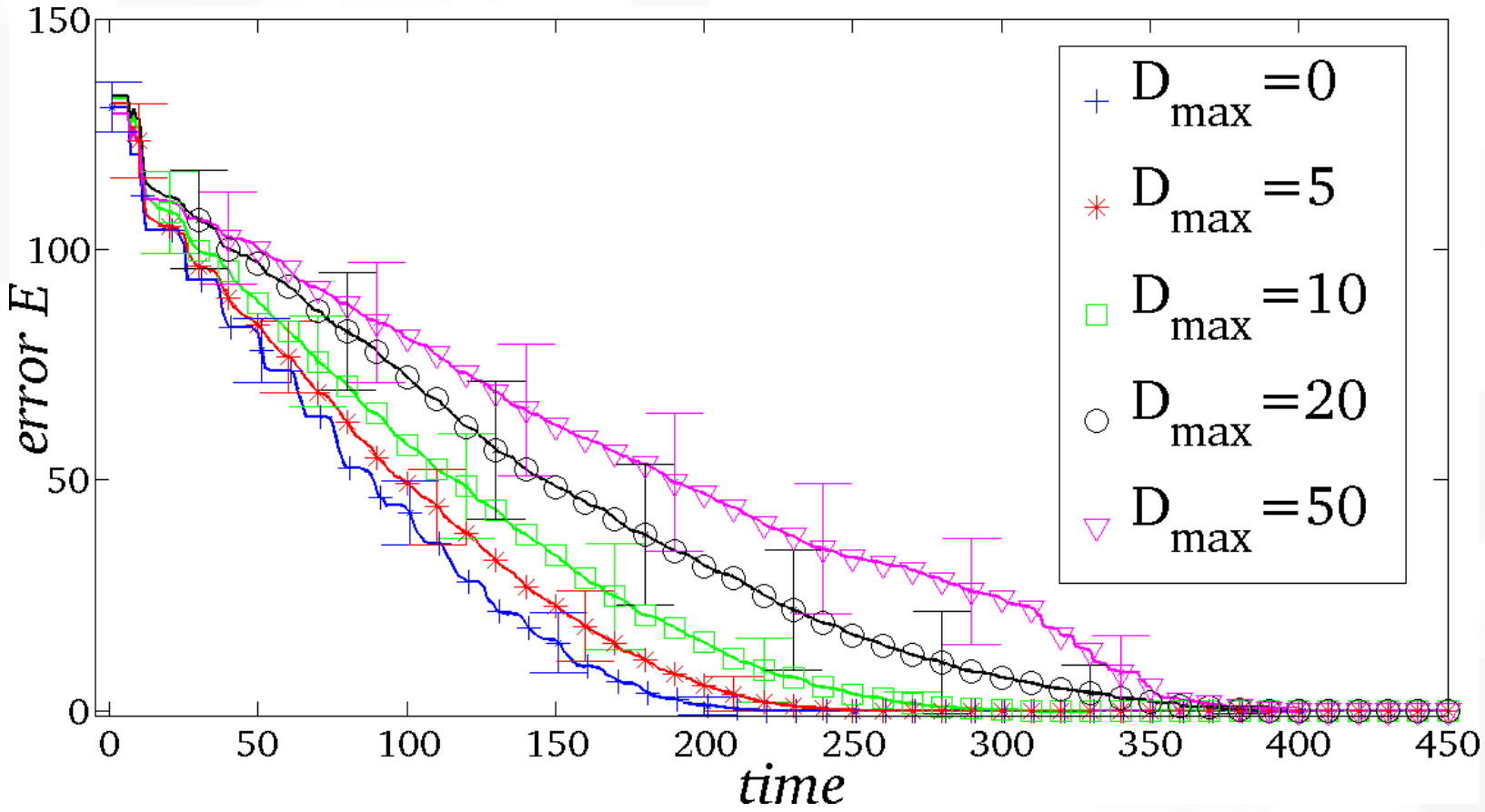
Robustness...

3 robots must transport goods from Rs to Re. Doors are actuated and rooms have a limited capacity.



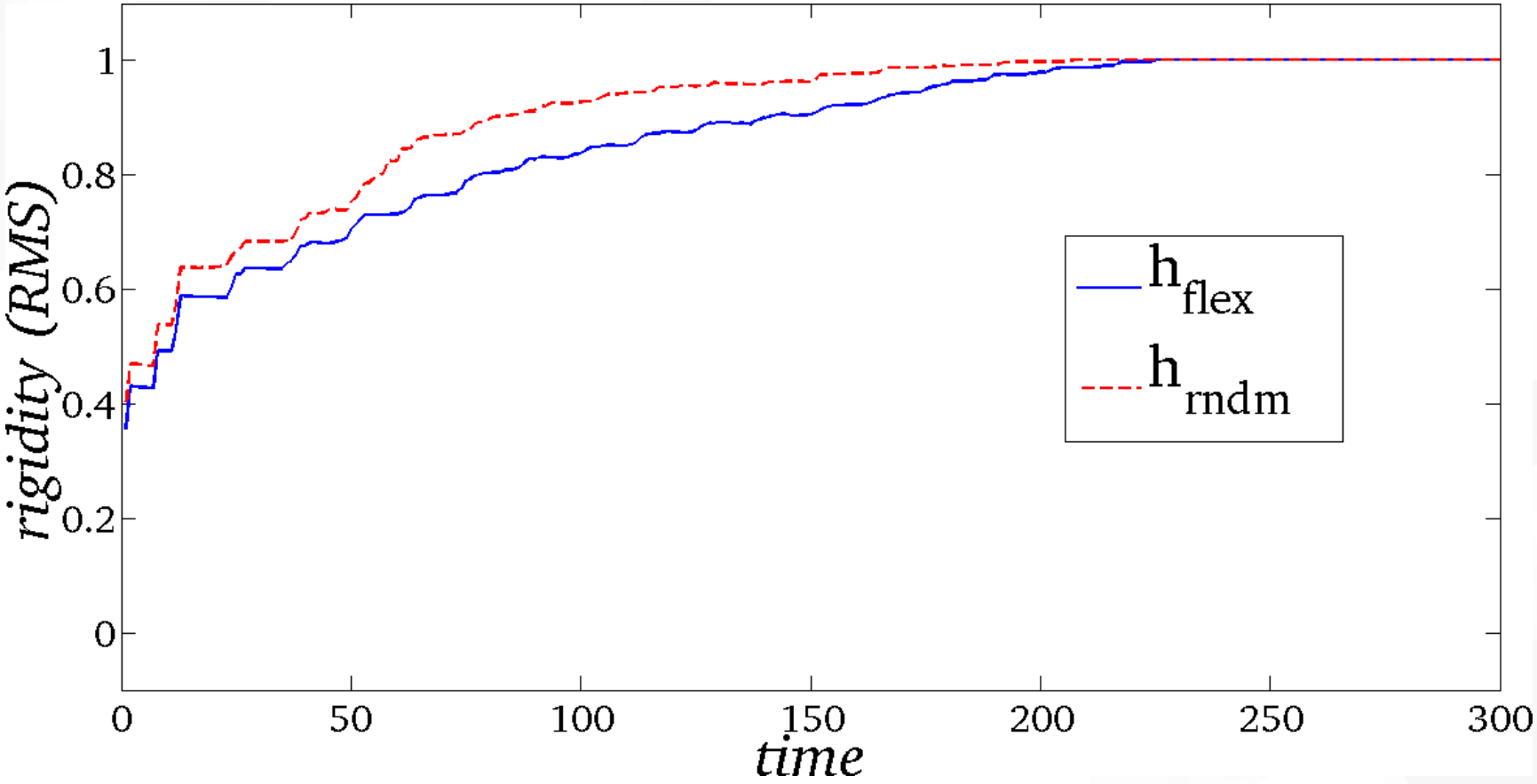
Delays

Activities are perturbed by random delays



Resource collapse

Randomly collapse resources are present at plan execution



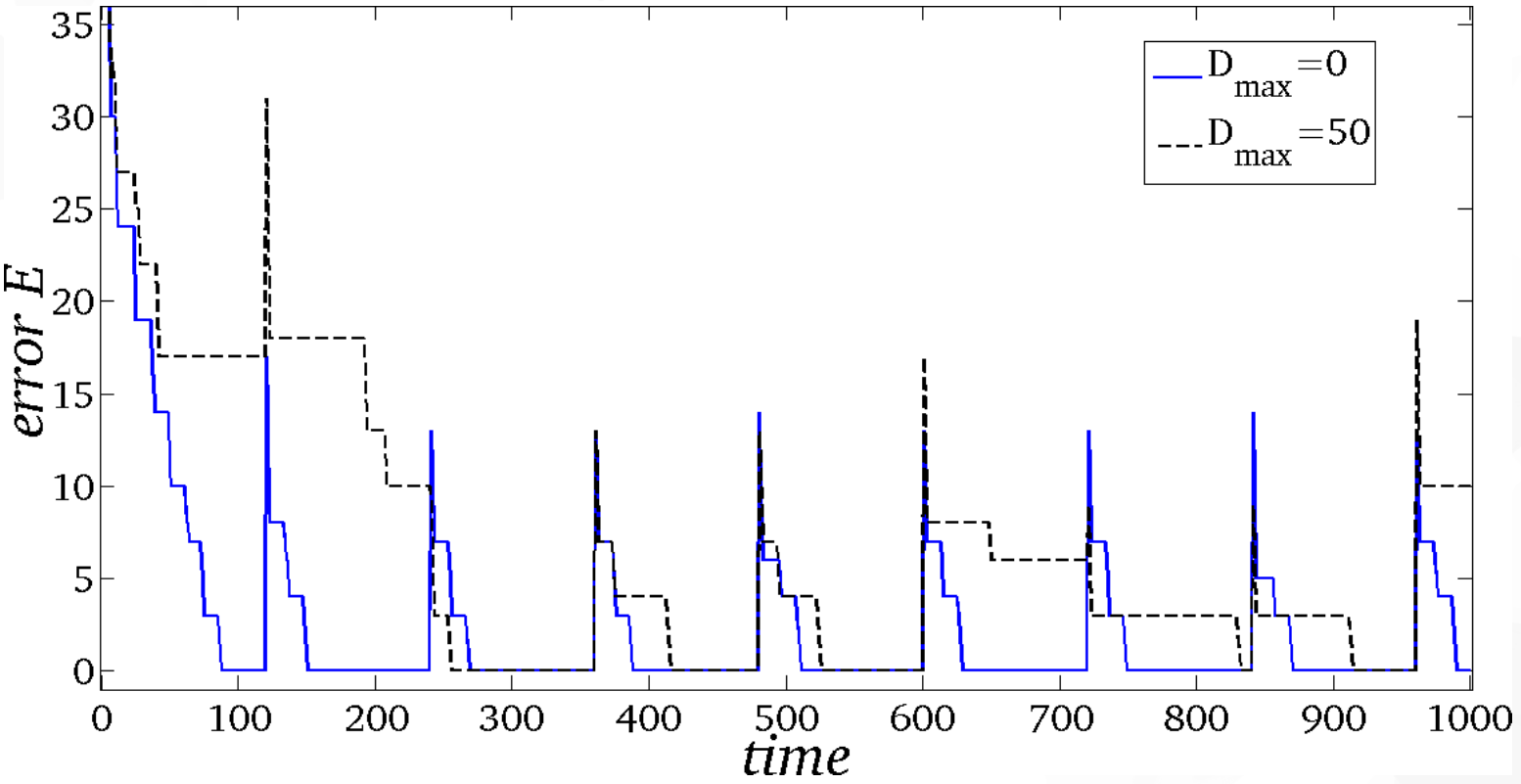
Exogenous events

Specifically, at regular intervals T_{door}, each door can be closed with probability p

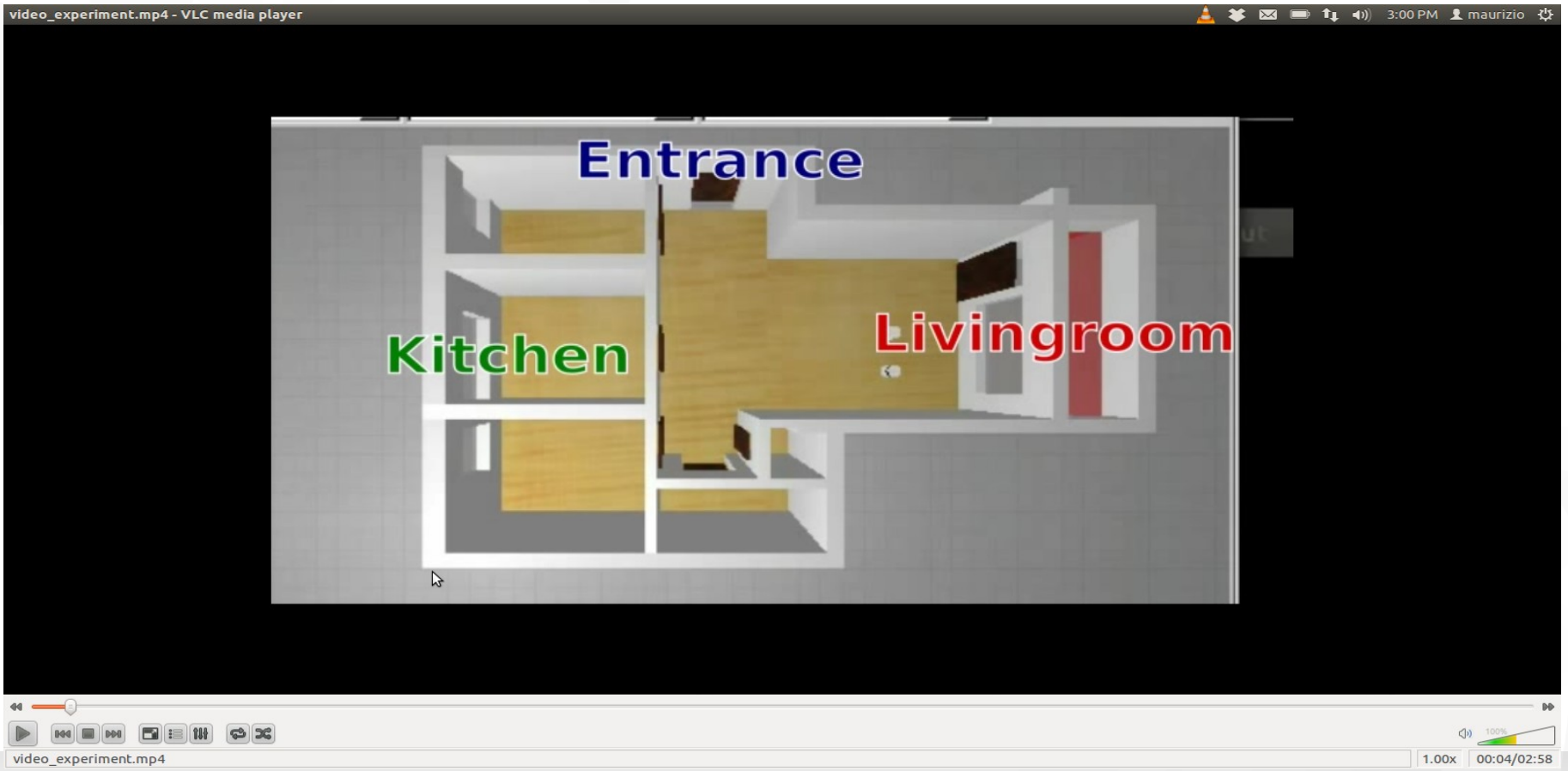
p_{close}	0	0.2	0.4	0.6	1.0
Success rate	1	0.76	0.47	0.3	0.03

Tracking

Rooms are dynamically put under surveillance (two robots must be in the same room to declare a room surveilled)



Two robots inside...



Future work

Which solver to choose depending on the situation (i.e.variable / value ordering heuristics)

Ongoing application in fully integrated robotic systems (Rubicon, Robot-ERA, Race)

Contact:
maurizio.di-rocco@aass.oru.se