#AI4Planning

Integrating Logic and Constraint Reasoning in a Timeline-based Planner

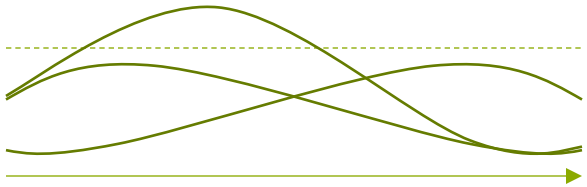
Riccardo De Benedictis and Amedeo Cesta

Institute of Cognitive Science and Technology
National Research Council
Rome – Italy

25/09/2015

The Timeline-based Approach

- A set of relevant *features* which need to be controlled to obtain a desired temporal behavior
- **Physical** or **logical** subsystems which are relevant to a given planning context
- The planner/scheduler plays the role of the *controller* for these entities



riccardo.debenedictis@istc.cnr.it

The ILOC Reasoning Environment – Objects and Constraints

- An object-oriented environment for the definition of **objects** and **constraints** among them
- Every object is an instance of a specific **type**
  - **Primitive types** (e.g., bools, ints, reals, etc.)
  - **Complex types** (e.g., robots, trucks, locations, etc.)
- **Dot notation** for addressing objects and enforcing constraints:

  <object>.<property>

- Constraints

$$[\![l.x \leq r.x]\!]$$
$$\neg[\![l.x \leq 5]\!]$$
$$[\![l.x \geq 5]\!] \wedge [\![l.x \leq 10]\!]$$
$$[\![l.x \leq 5]\!] \vee [\![l.x \geq 10]\!]$$
$$[\![l.x \geq 10]\!] \rightarrow [\![l.y \geq 10]\!]$$
$$\exists l \in Locations: l.x \geq 10$$
$$\forall l \in Locations: l.x \leq 100$$

riccardo.debenedictis@istc.cnr.it

The ILOC Reasoning Environment - Rules

- First-order Horn clauses
  - At most one positive literal (**head** of the clause)
  - Any number of negative literals (**body** of the clause)

$$Head \Leftarrow Body$$

- The body contains calls to **predicates** (sub-goals) and **constraints** (in any logical combination)
- No constraints in the head of the clause
- Rules having the same head are disjunctive

- First-order resolution
  - Not ordered sub-goaling

riccardo.debenedictis@istc.cnr.it

Timeline-based Planning within ILoC

- Create timeline complex types
  - StateVariable, ReusableResource, ConsumableResource, etc.
  - extendable through inheritance

- Endow predicates with numerical parameters representing **time**:
  - starting time
  - ending time
  - duration (duration = end - start)

- Endow predicates with a **scope** parameter
  - denotes **on which object** (e.g., timeline) the formula will appear

- Extend resolution for managing objects' **inconsistencies**
  - add further "implicit" constraints on the formula according to the **scope's type**
  - i.e., we provide **scheduling** capabilities to timelines

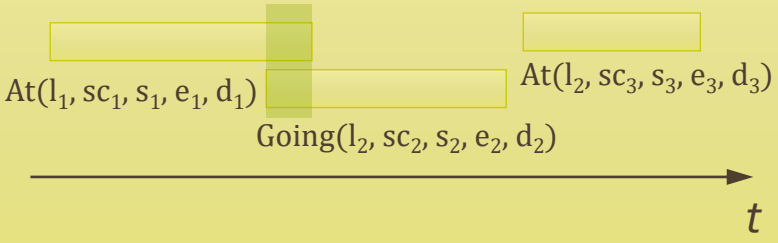
riccardo.debenedictis@istc.cnr.it

Timeline-based Planning within ILoC

- Create timeline complex types

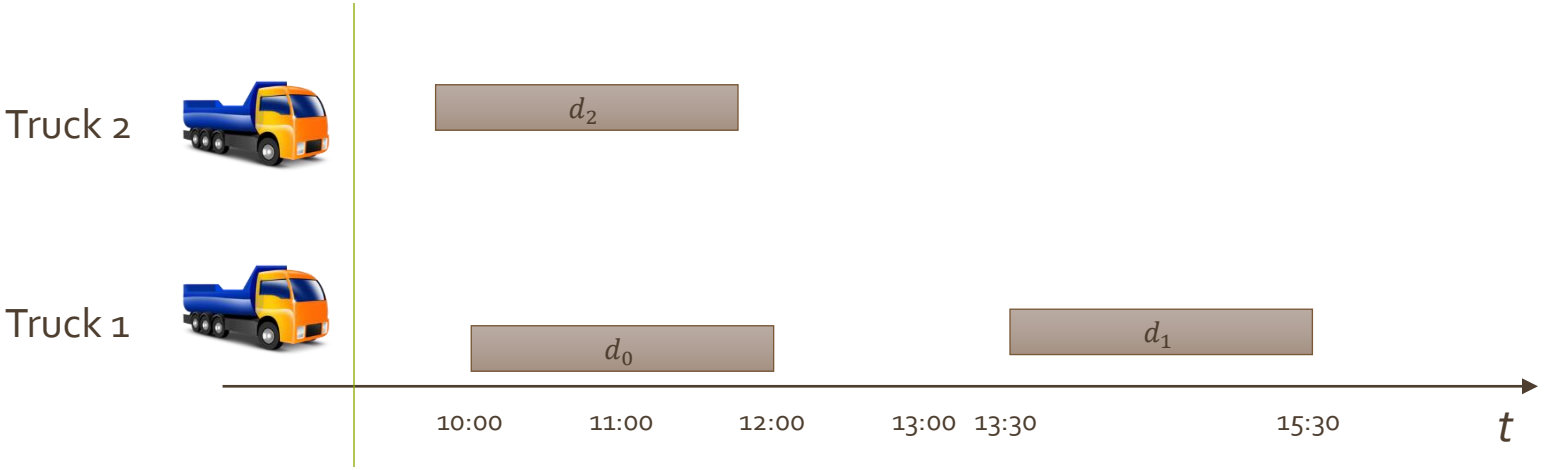
```
class Robot extends StateVariable {
   …
}

Going(Location l, Robot scope, real start, real end, real duration) := …

At(Location l, Robot scope, real start, real end, real duration) := …
```

Robot **extends** StateVariable



$$[\![e_1 \leq s_2]\!] \vee [\![e_2 \leq s_1]\!] \vee [\![sc_1 \neq sc_2]\!]$$

Scope Variables and Execution Uncertainty

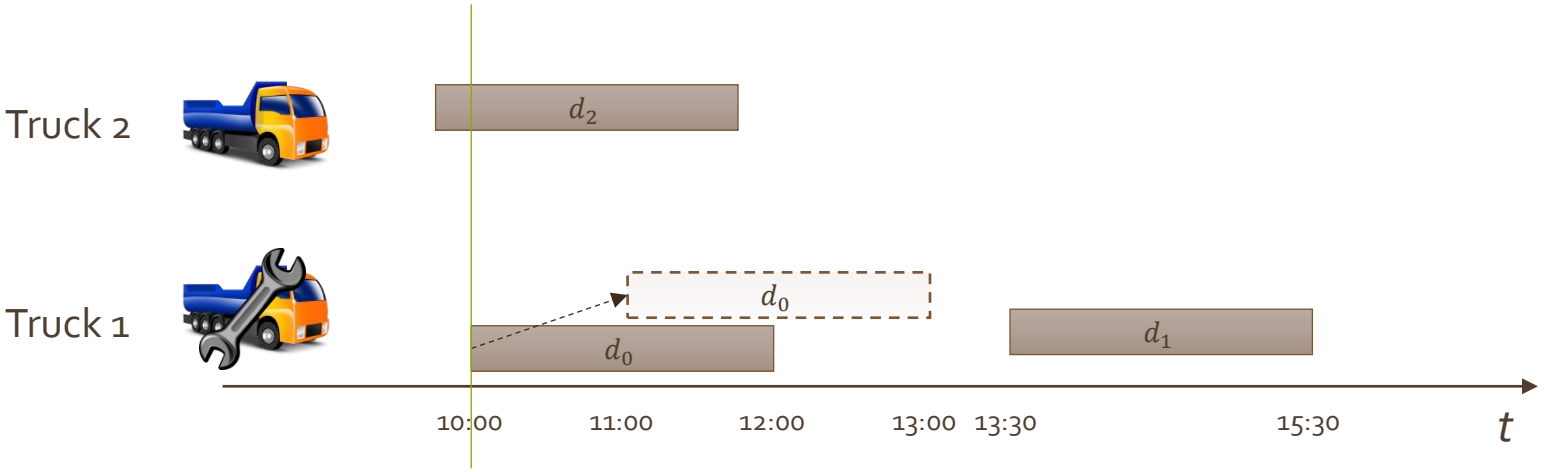


$[\![d_0.end \leq 17{:}00]\!]$
$[\![d_0.duration \geq 2{:}00]\!]$

$[\![d_1.start = 13{:}30]\!]$

riccardo.debenedictis@istc.cnr.it

Scope Variables and Execution Uncertainty



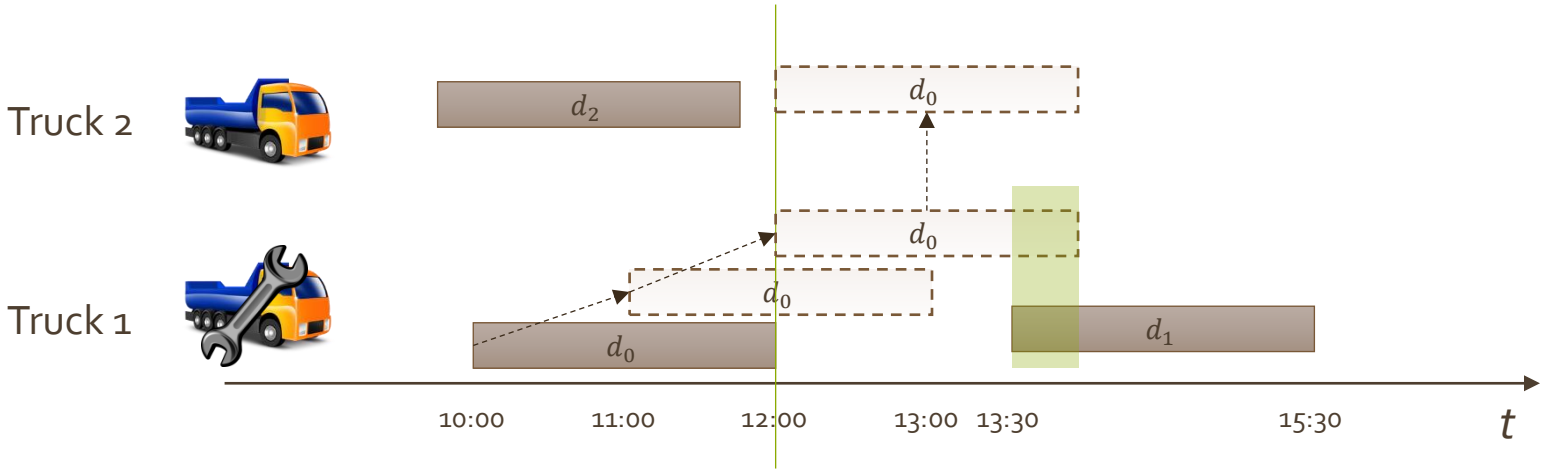
$[\![d_0.end \leq 17{:}00]\!]$
$[\![d_0.duration \geq 2{:}00]\!]$

$[\![d_1.start = 13{:}30]\!]$

$[\![d_0.start \geq 11{:}00]\!]$

riccardo.debenedictis@istc.cnr.it

Scope Variables and Execution Uncertainty



$[\![d_0.end \leq 17{:}00]\!]$
$[\![d_0.duration \geq 2{:}00]\!]$

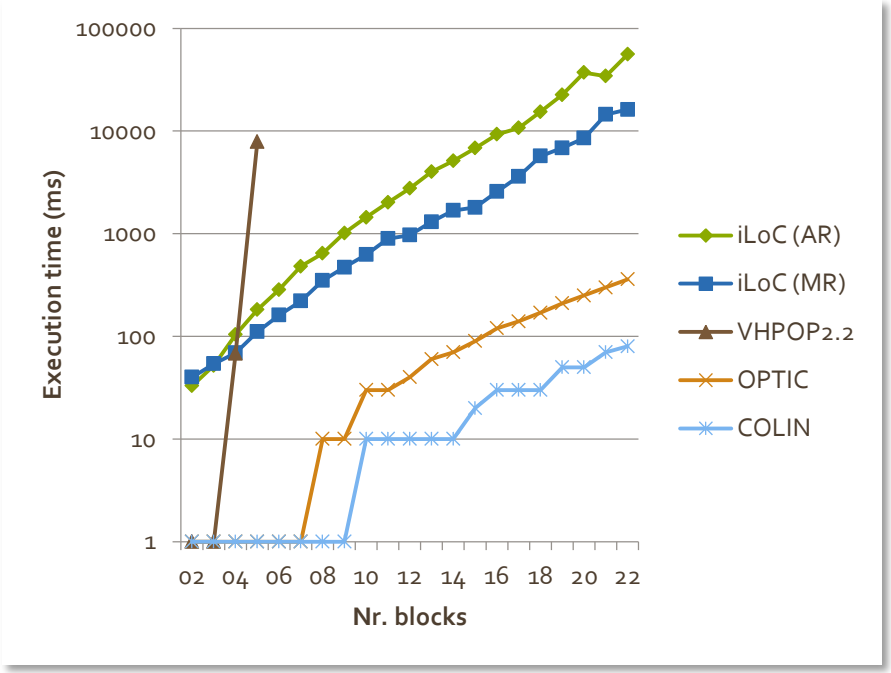
$[\![d_1.start = 13{:}30]\!]$

$[\![d_0.start \geq 11{:}00]\!]$
$[\![d_0.start \geq 12{:}00]\!]$

riccardo.debenedictis@istc.cnr.it

Conclusions

- An uniform schema for
  - Logic Programming (LP)
    - not-ordered subgoaling
  - Constraint Programming (CP)
    - similar to CLP
  - Timeline-reasoning
    - e.g., Scheduling



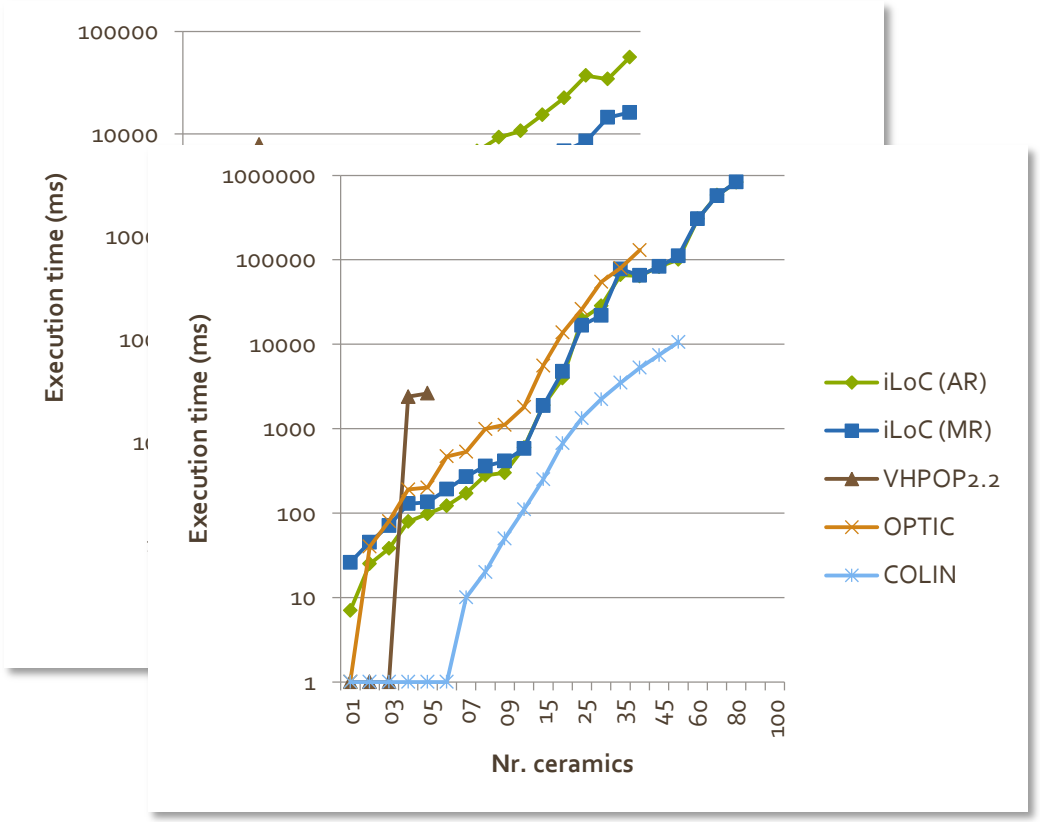
- Heuristics
  - Static and Dynamic

- Different planners
  - VHPOP (partial order approach)
  - OPTIC (TRPG heuristic)
  - COLIN (TRPG heuristic)
  - CPT, TPSHE, ITSAT, LPG, etc… will be added soon!!! :D

riccardo.debenedictis@istc.cnr.it

Conclusions

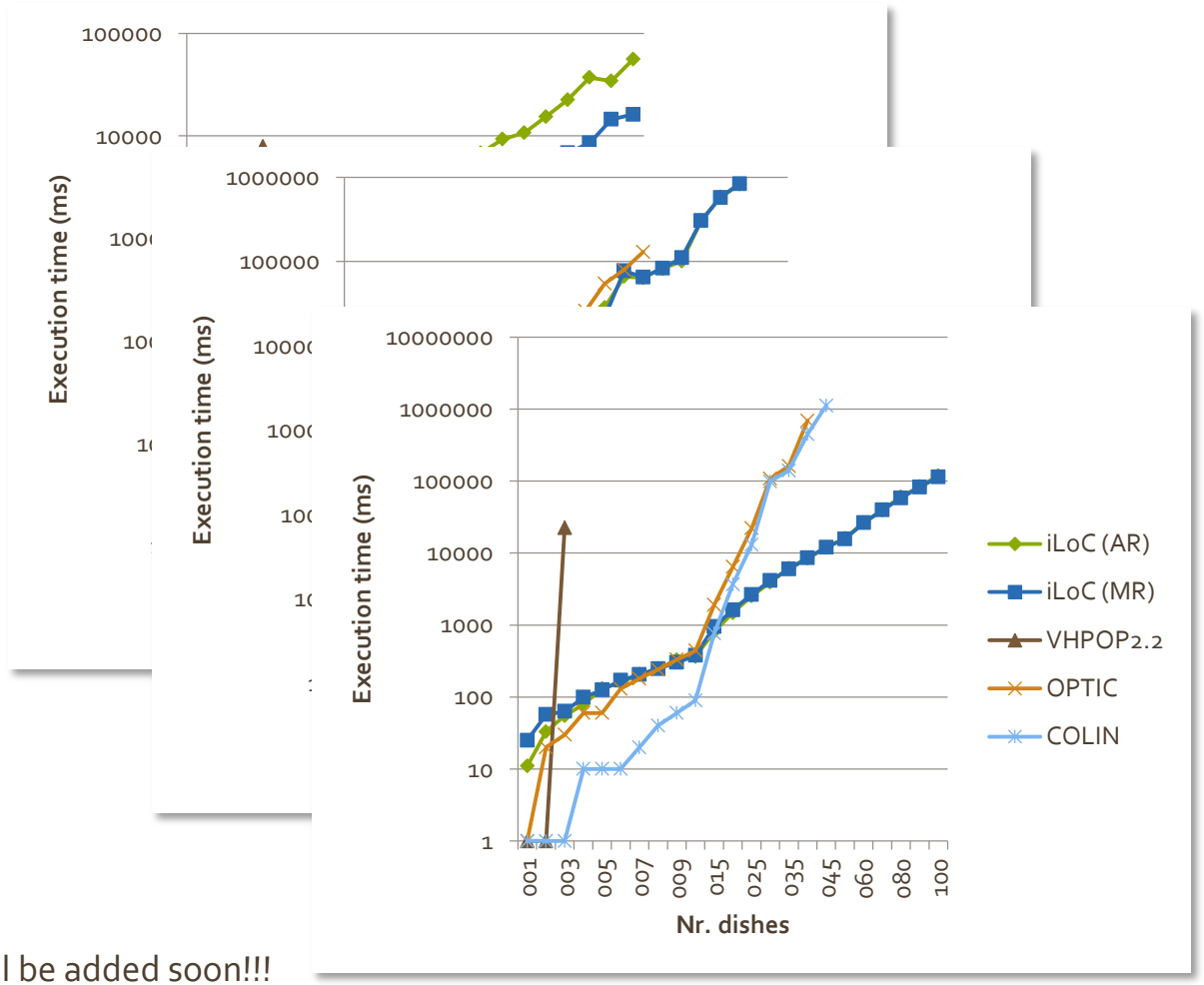
- An uniform schema for
  - Logic Programming (LP)
    - not-ordered subgoaling
  - Constraint Programming (CP)
    - similar to CLP
  - Timeline-reasoning
    - e.g., Scheduling
- Heuristics
  - Static and Dynamic
- Different planners
  - VHPOP (partial order approach)
  - OPTIC (TRPG heuristic)
  - COLIN (TRPG heuristic)
  - CPT, TPSHE, ITSAT, LPG, etc… will be added soon!!! :D



riccardo.debenedictis@istc.cnr.it

Conclusions

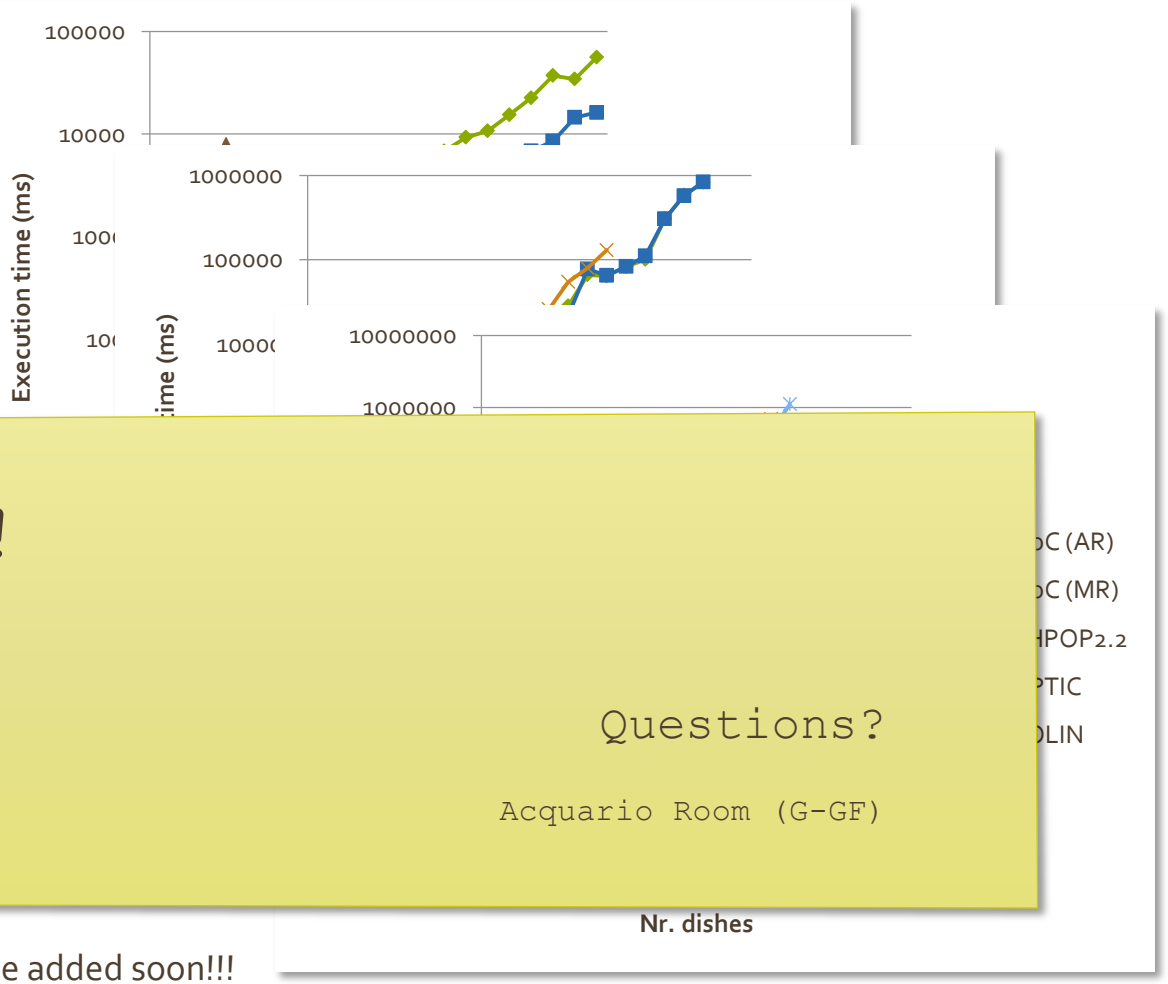
- An uniform schema for
  - Logic Programming (LP)
    - not-ordered subgoaling
  - Constraint Programming (CP)
    - similar to CLP
  - Timeline-reasoning
    - e.g., Scheduling
- Heuristics
  - Static and Dynamic
- Different planners
  - VHPOP (partial order approach)
  - OPTIC (TRPG heuristic)
  - COLIN (TRPG heuristic)
  - CPT, TPSHE, ITSAT, LPG, etc… will be added soon!!! :D



riccardo.debenedictis@istc.cnr.it

Conclusions

- An uniform schema for
  - Logic Programming (LP)
    - not-ordered subgoaling
  - Constraint Programming (CP)
    - similar to CLP
  - Timeline-reasoning
    - e.g., Scheduling
- H
  -
- D
  -
  -
  - COLIN (TRPG heuristic)
  - CPT, TPSHE, ITSAT, LPG, etc… will be added soon!!! :D



Thank you!

Questions?

Acquario Room (G-GF)

riccardo.debenedictis@istc.cnr.it