Cost-Optimal Factored Planning: Promises and Pitfalls

Éric Fabre[1], Patrik Haslum[2], Loïg Jezequel[3], Sylvie Thiébaux[2]

[1]INRIA Rennes - Bretagne Atlantique

[2]Australian National University & NICTA

[3]ENS Cachan Bretagne

Toronto, May 15, 2010

Introduction

Our goal: cost-optimal factored planning

cost-optimal: find the best plan

factored: find it quickly

Idea

- Split a planning problem P in several subproblems P_k;
- Update valid plans and their costs in P_k in order to ensure that locally optimal plans are part of globally optimal plans (in P);

Our approach

- Represent planning problems as weighted automata;
- use a message passing algorithm in networks of automata to find solutions.

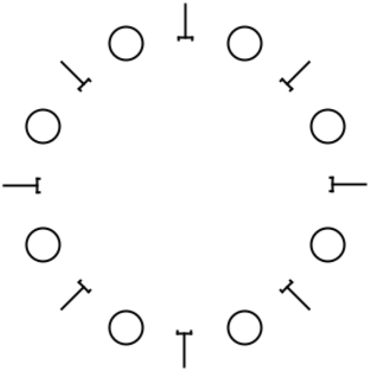
Outline

1. Introductory example: philosophers from IPC4
2. Cost-optimal factored planning in networks of weighted automata
3. Experimental results and remarks on complexity

Outline

1. Introductory example: philosophers from IPC4
2. Cost-optimal factored planning in networks of weighted automata
3. Experimental results and remarks on complexity

Philosophers: the problem (re-encoding of IPC4's)



Configuration

an alternating cycle of philosophers and forks

Actions availables for philosophers

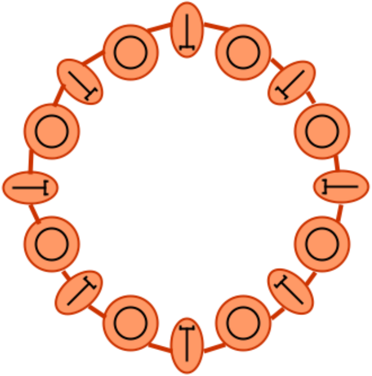
only in this order:

1. take left fork
2. take right fork
3. release left fork
4. release right fork

Goal

find deadlock

Philosophers: factored approach



Components

philosophers and forks

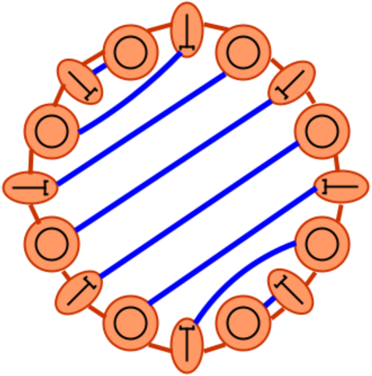
Interactions

shared actions

Needed

interaction graph which is a tree

Philosophers: from cycle to tree



Idea

merge components

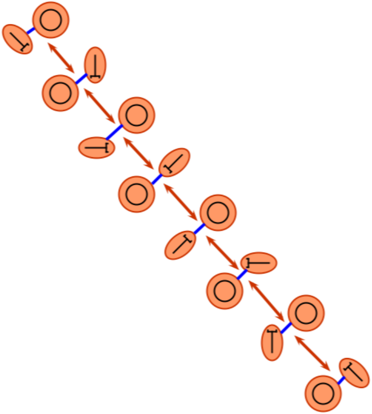
In this case

philosopher with opposed fork

Result

the new interaction graph is a tree (actually a line)

Philosophers: from cycle to tree



Idea

merge components

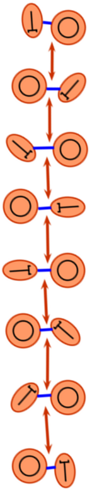
In this case

philosopher with opposed fork

Result

the new interaction graph is a tree (actually a line)

Philosophers: factored solving



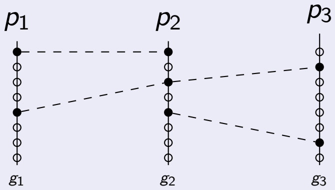
$\Rightarrow$

$$P_0 - P_1 - P_2 - P_3 - P_4 - P_5 - P_6 - P_7$$

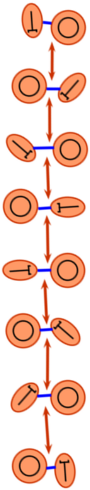
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



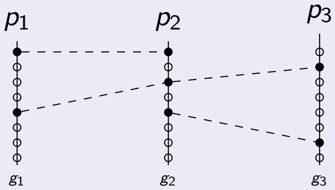
$\Rightarrow$

$$P_0$$
$$\downarrow$$
$$P'_1 = P_0 \times P_1$$
$$|$$
$$P_2$$
$$|$$
$$P_3$$
$$|$$
$$P_4$$
$$|$$
$$P_5$$
$$|$$
$$P_6$$
$$|$$
$$P_7$$

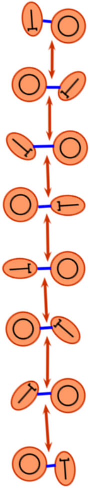
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



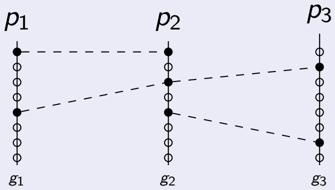
$\Rightarrow$

$$
\begin{array}{c}
P_0 \\
| \\
P'_1 = P_0 \times P_1 \\
\downarrow \\
P'_2 = P'_1 \times P_2 \\
| \\
P_3 \\
| \\
P_4 \\
| \\
P_5 \\
| \\
P_6 \\
| \\
P_7
\end{array}
$$

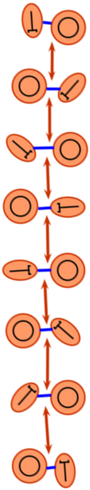
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



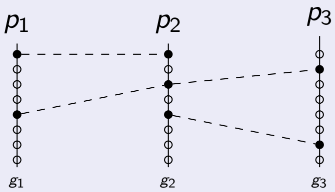
$\Rightarrow$

$$
\begin{array}{c}
P_0 \\
| \\
P'_1 = P_0 \times P_1 \\
| \\
P'_2 = P'_1 \times P_2 \\
\downarrow \\
P'_3 = P'_2 \times P_3 \\
| \\
P_4 \\
| \\
P_5 \\
| \\
P_6 \\
| \\
P_7
\end{array}
$$

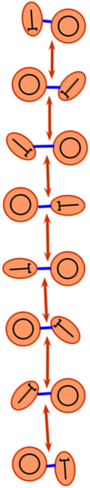
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



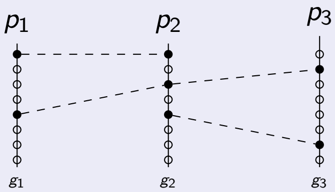
$\Rightarrow$

$$
\begin{array}{c}
P_0 \\
| \\
P'_1 = P_0 \times P_1 \\
| \\
P'_2 = P'_1 \times P_2 \\
| \\
P'_3 = P'_2 \times P_3 \\
\downarrow \\
P'_4 = P'_3 \times P_4 \\
| \\
P_5 \\
| \\
P_6 \\
| \\
P_7
\end{array}
$$

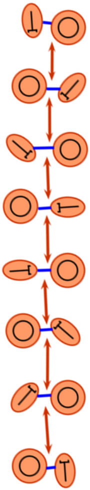
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



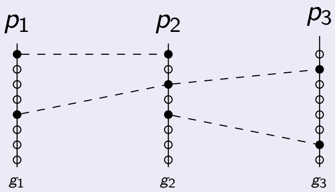
$\Rightarrow$

$$
\begin{array}{c}
P_0 \\
| \\
P'_1 = P_0 \times P_1 \\
| \\
P'_2 = P'_1 \times P_2 \\
| \\
P'_3 = P'_2 \times P_3 \\
| \\
P'_4 = P'_3 \times P_4 \\
\downarrow \\
P'_5 = P'_4 \times P_5 \\
| \\
P_6 \\
| \\
P_7
\end{array}
$$

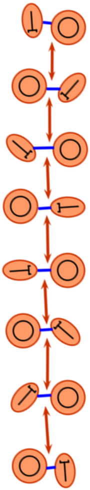
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



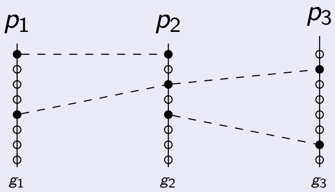
$\Rightarrow$

$$
\begin{array}{c}
P_0 \\
| \\
P_1' = P_0 \times P_1 \\
| \\
P_2' = P_1' \times P_2 \\
| \\
P_3' = P_2' \times P_3 \\
| \\
P_4' = P_3' \times P_4 \\
| \\
P_5' = P_4' \times P_5 \\
\downarrow \\
P_6' = P_5' \times P_6 \\
| \\
P_7
\end{array}
$$

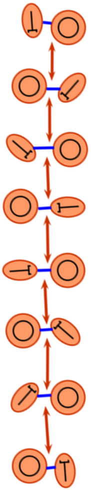
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



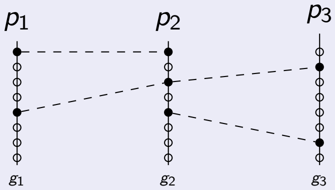
$\Rightarrow$

$$
\begin{aligned}
&P_0\\
P'_1 &= P_0 \times P_1\\
P'_2 &= P'_1 \times P_2\\
P'_3 &= P'_2 \times P_3\\
P'_4 &= P'_3 \times P_4\\
P'_5 &= P'_4 \times P_5\\
P'_6 &= P'_5 \times P_6\\
P'_7 &= P'_6 \times P_7
\end{aligned}
$$

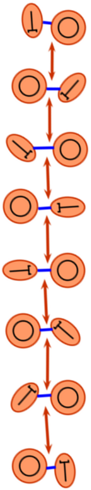
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



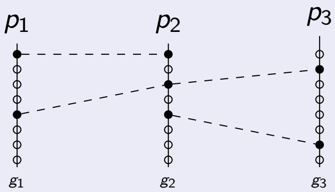
$\Rightarrow$

$$
\begin{array}{c}
P_0 \\
| \\
P'_1 \\
| \\
P'_2 \\
| \\
P'_3 \\
| \\
P'_4 \\
| \\
P'_5 \\
| \\
P''_6 = P'_7 \times P'_6 \\
\uparrow \\
P'_7
\end{array}
$$

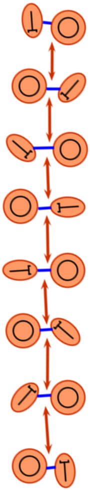
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



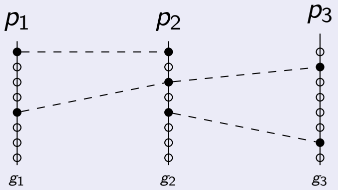
$\Rightarrow$

$$\begin{array}{c}
P_0 \\ | \\ P'_1 \\ | \\ P'_2 \\ | \\ P'_3 \\ | \\ P'_4 \\ | \\ P''_5 = P''_6 \times P'_5 \\ \uparrow \\ P''_6 = P'_7 \times P'_6 \\ | \\ P'_7
\end{array}$$

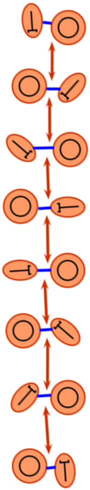
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



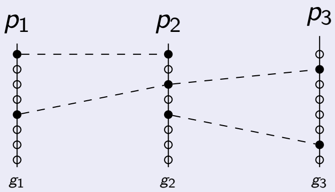
$\Rightarrow$

$$
\begin{array}{c}
P_0 \\
| \\
P'_1 \\
| \\
P'_2 \\
| \\
P'_3 \\
| \\
P''_4 = P''_5 \times P'_4 \\
\uparrow \\
P''_5 = P''_6 \times P'_5 \\
| \\
P''_6 = P'_7 \times P'_6 \\
| \\
P'_7
\end{array}
$$

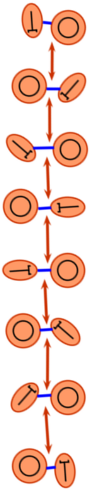
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



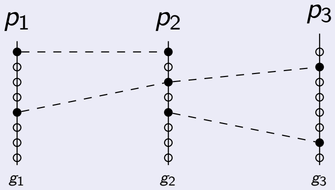
$\Rightarrow$

$$
\begin{array}{rcl}
 & P_0 & \\
 & | & \\
 & P'_1 & \\
 & | & \\
 & P'_2 & \\
 & | & \\
P''_3 = & P''_4 \times P'_3 & \\
 & \uparrow & \\
P''_4 = & P''_5 \times P'_4 & \\
 & | & \\
P''_5 = & P''_6 \times P'_5 & \\
 & | & \\
P''_6 = & P'_7 \times P'_6 & \\
 & | & \\
 & P'_7 &
\end{array}
$$

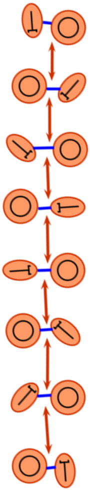
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



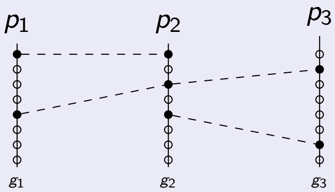
$\Rightarrow$

$$
\begin{aligned}
&P_0\\
&P'_1\\
P''_2 &= P''_3 \times P'_2\\
P''_3 &= P''_4 \times P'_3\\
P''_4 &= P''_5 \times P'_4\\
P''_5 &= P''_6 \times P'_5\\
P''_6 &= P'_7 \times P'_6\\
&P'_7
\end{aligned}
$$

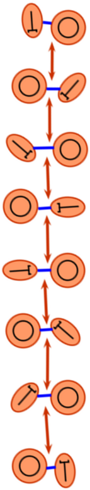
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



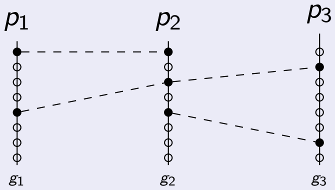
$\Rightarrow$

$$
\begin{array}{c}
P_0 \\
| \\
P''_1 = P''_2 \times P'_1 \\
\uparrow \\
P''_2 = P''_3 \times P'_2 \\
| \\
P''_3 = P''_4 \times P'_3 \\
| \\
P''_4 = P''_5 \times P'_4 \\
| \\
P''_5 = P''_6 \times P'_5 \\
| \\
P''_6 = P'_7 \times P'_6 \\
| \\
P'_7
\end{array}
$$

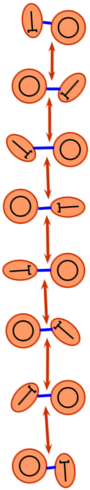
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving

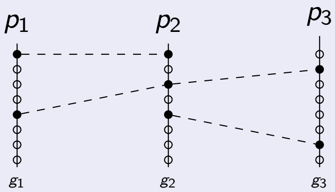


$$\Rightarrow \quad \begin{aligned}
P'_0 &= P''_1 \times P_0 \\
P''_1 &= P''_2 \times P'_1 \\
P''_2 &= P''_3 \times P'_2 \\
P''_3 &= P''_4 \times P'_3 \\
P''_4 &= P''_5 \times P'_4 \\
P''_5 &= P''_6 \times P'_5 \\
P''_6 &= P'_7 \times P'_6 \\
& \quad P'_7
\end{aligned}$$

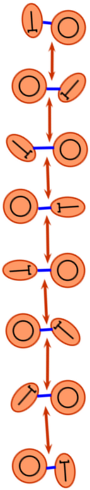
Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



Philosophers: factored solving



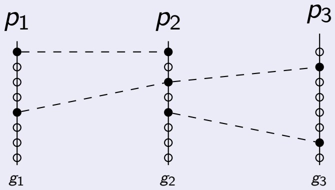
$\Rightarrow$

$$P'_0 - P''_1 - P''_2 - P''_3 - P''_4 - P''_5 - P''_6 - P'_7$$

Principle

1. Each component is replaced by a representation of its local plans (to a local goal),
2. using knowledge from neighbors these sets of local plans are updated in order to be compatible with the other sets.

Compatibility of plans



The algorithm: previous works

Constraint satisfaction [Dechter03]

- Product to ensure constraint satisfaction between neighbours
- Projection to limit size of objects

Factored planning [Brafman&Domshlak06] [Brafman&Domshlak08]

- Message = set of plans;
- not polynomially bounded in general (different from constraint satisfaction)
- Restriction of message size (enforced bound)
- Does not allow cost-optimal planning

Outline

1. Introductory example: philosophers from IPC4
2. Cost-optimal factored planning in networks of weighted automata
3. Experimental results and remarks on complexity

The algorithm: our approach

Theorem: generalization

Any representation of plans (with notion of product/projection) may be used in the MPA if the following holds:

$$\mathcal{P}_\nu(S_1 \times S_2) = \mathcal{P}_\nu(S_1) \times \mathcal{P}_\nu(S_2)$$

set of plans with costs as weighted automata

- Automata allow to represent all plans with finite objects;
- representing all plans allows to perform cost-optimal planning;
- finite weighted automata represent all plans with their costs.

Overview of the approach

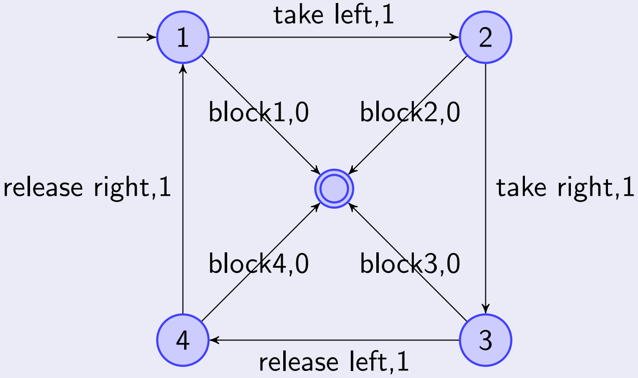
Product: responsible for compatibility of local plans

Projection: reduce size of objects and responsible for cost-optimization

The algorithm: component representation

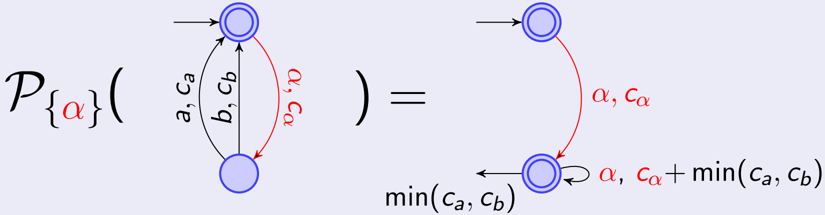
Think of domain transition graph.

Example: a philosopher

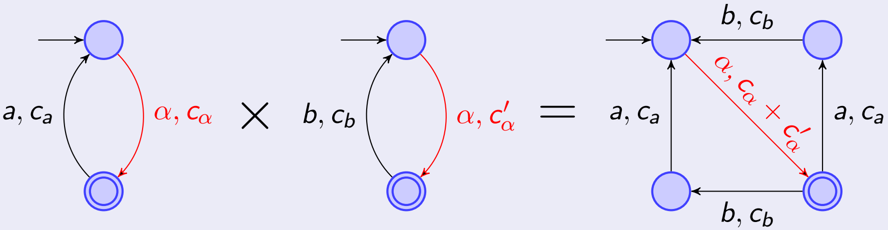


The algorithm: operations

Projection (responsible for optimization)



Product (responsible for synchronization of local plans)



The algorithm: guarantees

After the MPA, if the interaction graph was a tree, some properties are ensured on updated components:

local plans are part of a global plan:

any local plan can be extended into a global plan and the projection of any global plan on a component is present in it as one of its local plans.

cost-optimal local plans are part of a cost-optimal global plan:

- any cost-optimal local plan can be extended into a cost-optimal global plan;
- the projection of any cost-optimal global plan on a component is a cost-optimal local plan in this component.

Outline

Benchmarks

Chosen problems

Taken from IPCs, but:

- a lot of problems were not well suited to factored planning (huge tree width);
- others needed re-encoding (they were centralized encoding of distributed problems).

Call:

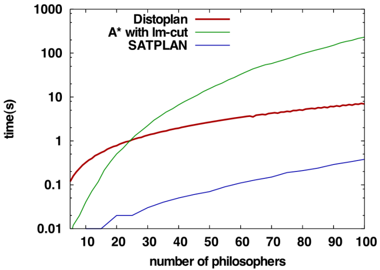
we are still looking for other benchmarks!

Factoring

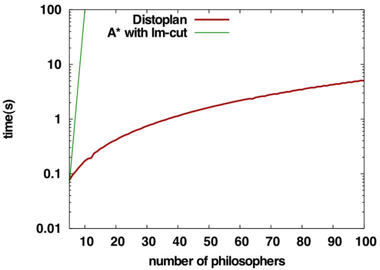
"by hand"

Philosophers (IPC4, alternative encoding)

Deadlockable:

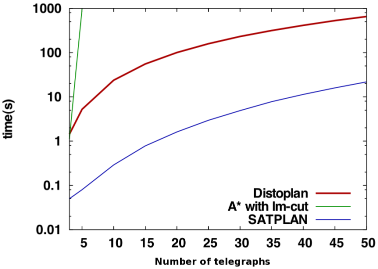


Deadlock free:

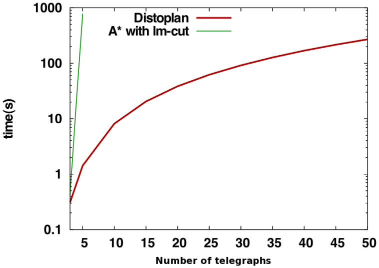


Optical telegraph (IPC4, alternative encoding)

Deadlockable:



Deadlock free:



Time complexity

- MPA sends a polynomial (in number of components) number of messages;
- each one is processed in polynomial time (in the size of the message and the receiving component);
- without additional restrictions, the size of messages can grow beyond polynomial.

Under similar condition as [Brafman&Domshlak08] we ensure polynomial time complexity (in number of components)

Sufficient condition for polynomial size of messages

Number of shared operators in any locally valid plan is bounded by a constant.

This condition is not necessary: there is problems where it does not hold, but message sizes are polynomially bounded (philosophers for example)

Conclusion

Conclusion

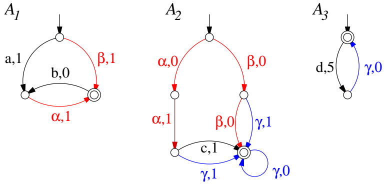
We presented an algorithm for cost-optimal factored planning, which:

- computes partially ordered plans;
- computes all the plans (more general than previous approaches);
- has polynomial time-complexity (under similar restrictions as previous approaches).

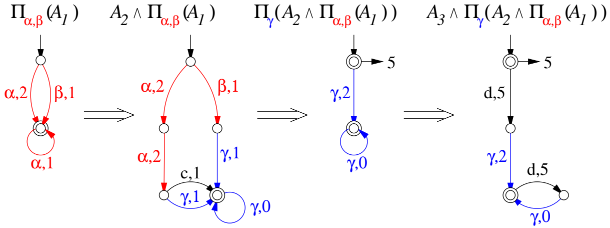
We implemented it and tested it on some benchmarks:

- results comparable to other up to date planners on solvable instances of problems;
- capability to detect non-solvability;
- however only on few problems: our results depend strongly of problems structure.

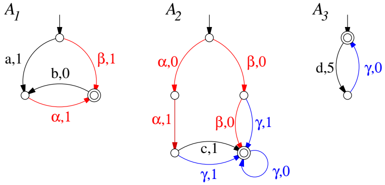
Sample execution of the MPA



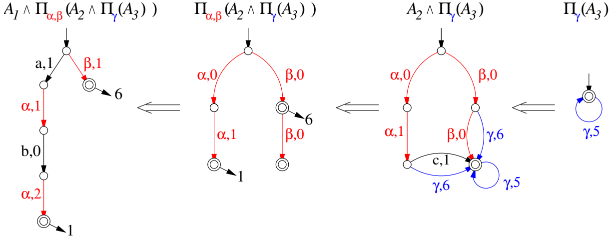
Messages from left to right:



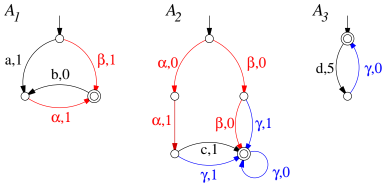
Sample execution of the MPA



Messages from right to left:



Sample execution of the MPA



Reduced components and optimal plans: