MIP formulations for delete-free AI planning

Domenico Salvagnin
Matteo Zanella

DEI, University of Padova

Minneapolis, 2025

Classical AI Planning*

- Finite set P of boolean variables (facts)
- Initial state I (list of facts true at the beginning)
- Goal state G (which facts we want to be true)
- Finite set A of actions. Each action a has:
  - Nonnegative cost $cost(a) \geq 0$
  - Precondition $pre(a) \subseteq P$
  - Add effects $add(a) \subseteq P$
  - Delete effects $del(a) \subseteq P$
- We want to find the plan (sequence of actions) of minimum total cost to reach a goal state

* *STRIPS formalism*

Classical AI Planning

- This is basically a shortest path on an (exponential) state space
- Usually solved with the A* algorithm
- A* needs (admissible) heuristics *[i.e., lower bounds]*
- One of the most studied relaxation is the so called delete-free relaxation of a planning task (h^+)
- Still not polynomial, but at least “just” NP-hard
- Can use MIP technology for it! :)

Delete-free Planning Tasks

- Each action is applied at most once
- Length of optimal plan always at most min(|P|,|A|)
- Feasibility can be tested in polynomial time
  - Basically a reachability test
- Finding feasible plans is trivial
  - Any random dive will do
- Wlog we can assume that I = ∅

Basic MIP model

$$\min \sum_{a \in A} cost(a) x_a$$

minimize total cost

$$\sum_{a \in A : p \in add(a)} x_{a,p} = x_p \quad \forall p \in P$$

Fact achieved iff an action is its first achiever

$$\sum_{a \in A : p \in pre(a), q \in add(a)} x_{a,q} \le x_p \quad \forall p, q \in P$$

Action can be a first achiever only if its preconditions are met

$$x_{a,p} \le x_a \quad \forall a \in A, \forall p \in add(a)$$

Action can be a first achiever only if used

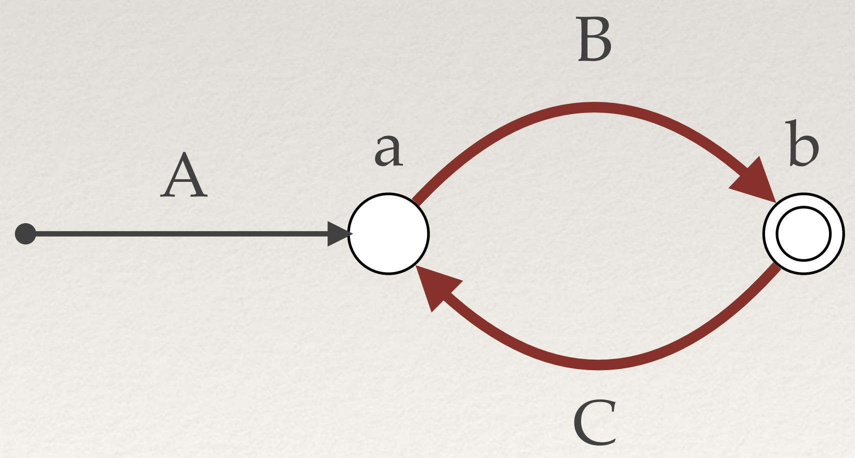
$$x_p = 1 \quad \forall p \in G$$

Each goal fact is achieved

Rankooh & Rintanen "Efficient Computation and Informative Estimation of h+ by Integer and Linear Programming", 2022

Basic MIP model

- The basic set of constraints does not give a complete MIP formulation
- We are missing causal acyclicity



Timestamps

- Assign an integer timestamp $t_p \in [0, |P|]$ to each fact
- Any precondition of the first achiever of p must have a timestamp smaller than the timestamp of p

$$t_p + 1 \leq t_q + |P|(1 - x_{a,q}) \quad \forall a \in A, p \in pre(a), q \in add(a)$$

- Quite compact, but LP relaxation is weak

Imai & Fukunaga "On a practical, integer-linear programming model for delete-free tasks and its use as a heuristic for cost-optimal planning", 2015

Vertex Elimination

- Consider the causal graph G_Π of the delete free planning task Π
  - Each fact is a node
  - For each action a, we have the set of arcs (p,q) for every p in pre(a) and q in add(a)
- Pick any elimination ordering O of G_Π and consider the corresponding vertex elimination graph G_Π^*, and let Δ be the set of all the triples (p,q,r) added during the elimination process

Rankooh & Rintanen "Efficient Computation and Informative Estimation of h+ by Integer and Linear Programming", 2022

Vertex Elimination

- Then we can add new binary variables e_{pq} for all (p,q) in the edge set E* of G^*_Π and constraints:

$$
\begin{aligned}
x_{a,q} &\leq e_{pq} && \forall a \in A, p \in add(a), q \in add(a) \\
e_{p,q} + e_{q,p} &\leq 1 && \forall (p,q) \in E^* \\
e_{p,q} + e_{q,r} - 1 &\leq e_{p,r} && \forall (p,q,r) \in \Delta
\end{aligned}
$$

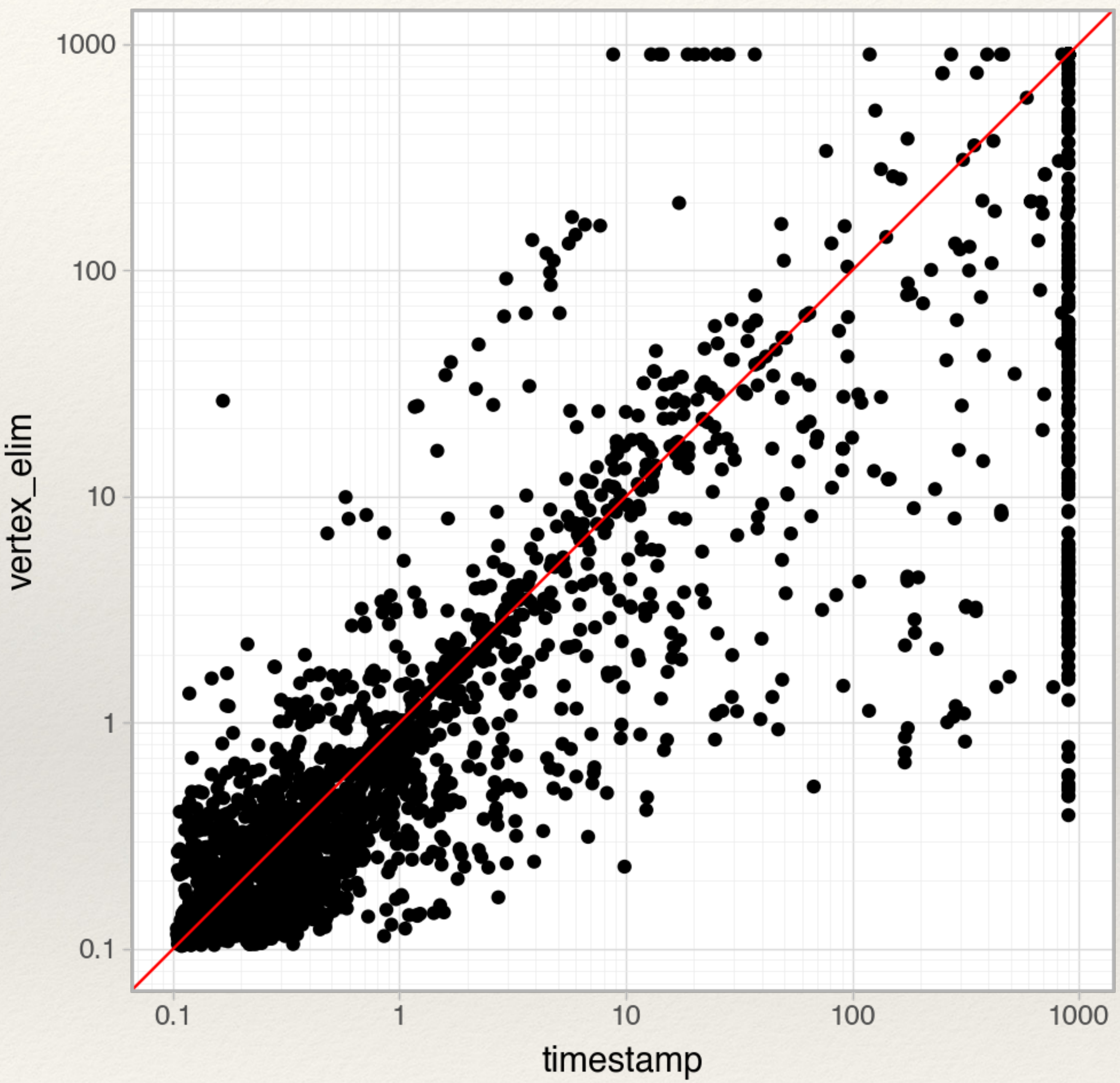
- Can grow quite large in practice (but still polynomial)
- Its LP relaxation is quite stronger

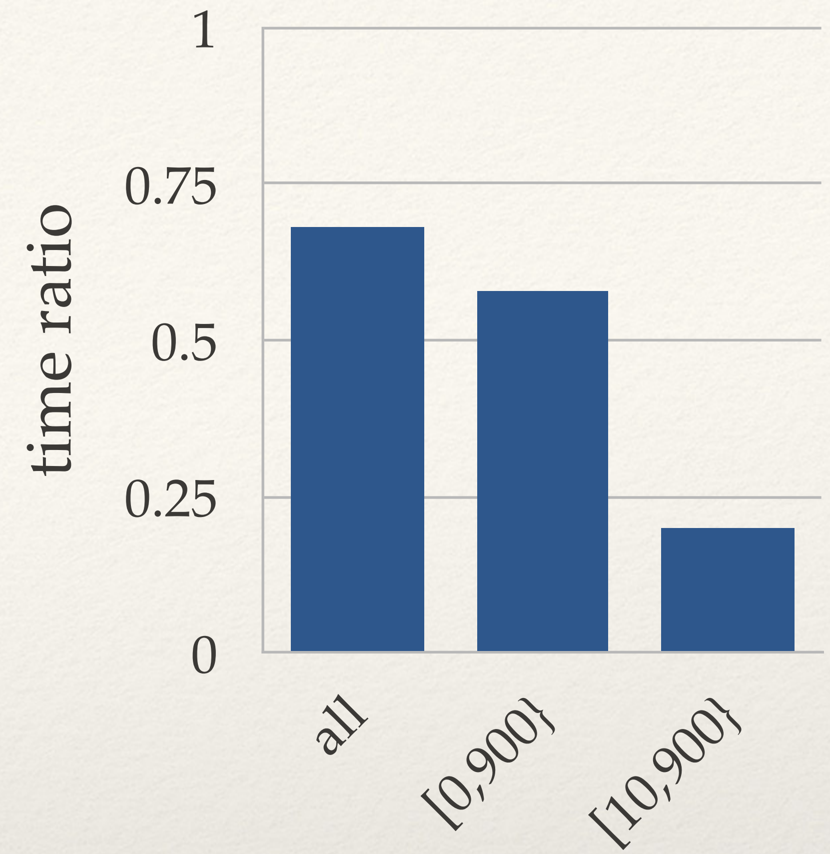
Rankooh & Rintanen "Efficient Computation and Informative Estimation of h+ by Integer and Linear Programming", 2022

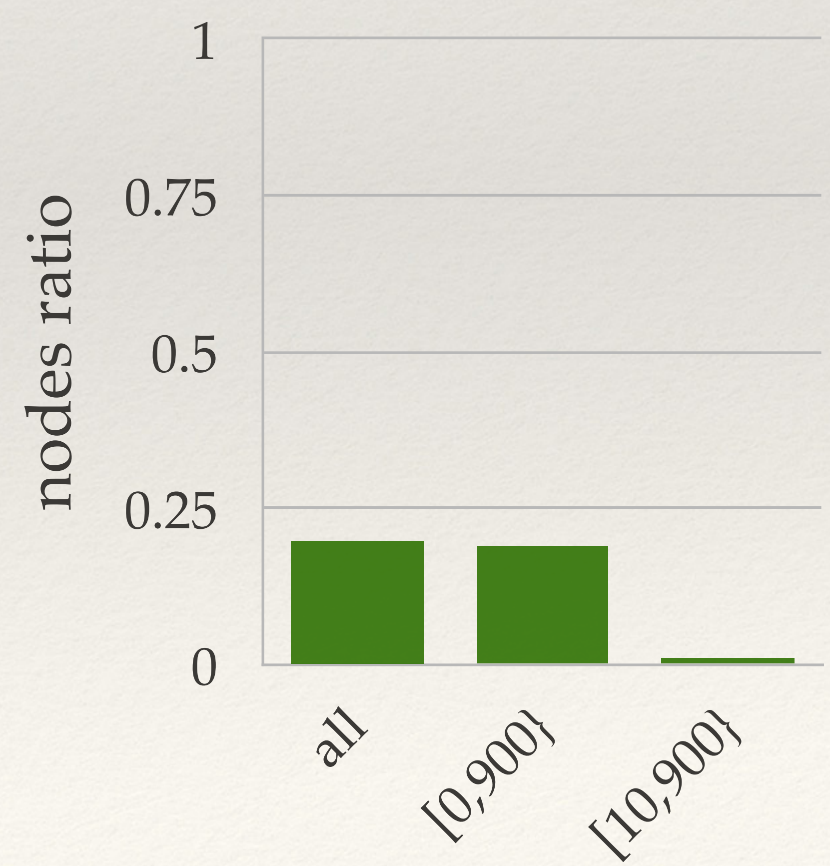
Preprocessing

- Long list of known specific reductions from literature:
  - Landmark-based reductions
  - First-achievers filtering
  - Relevance analysis
  - Removal of dominated actions

Imai & Fukunaga "On a practical, integer-linear programming model for delete-free tasks and its use as a heuristic for cost-optimal planning", 2015



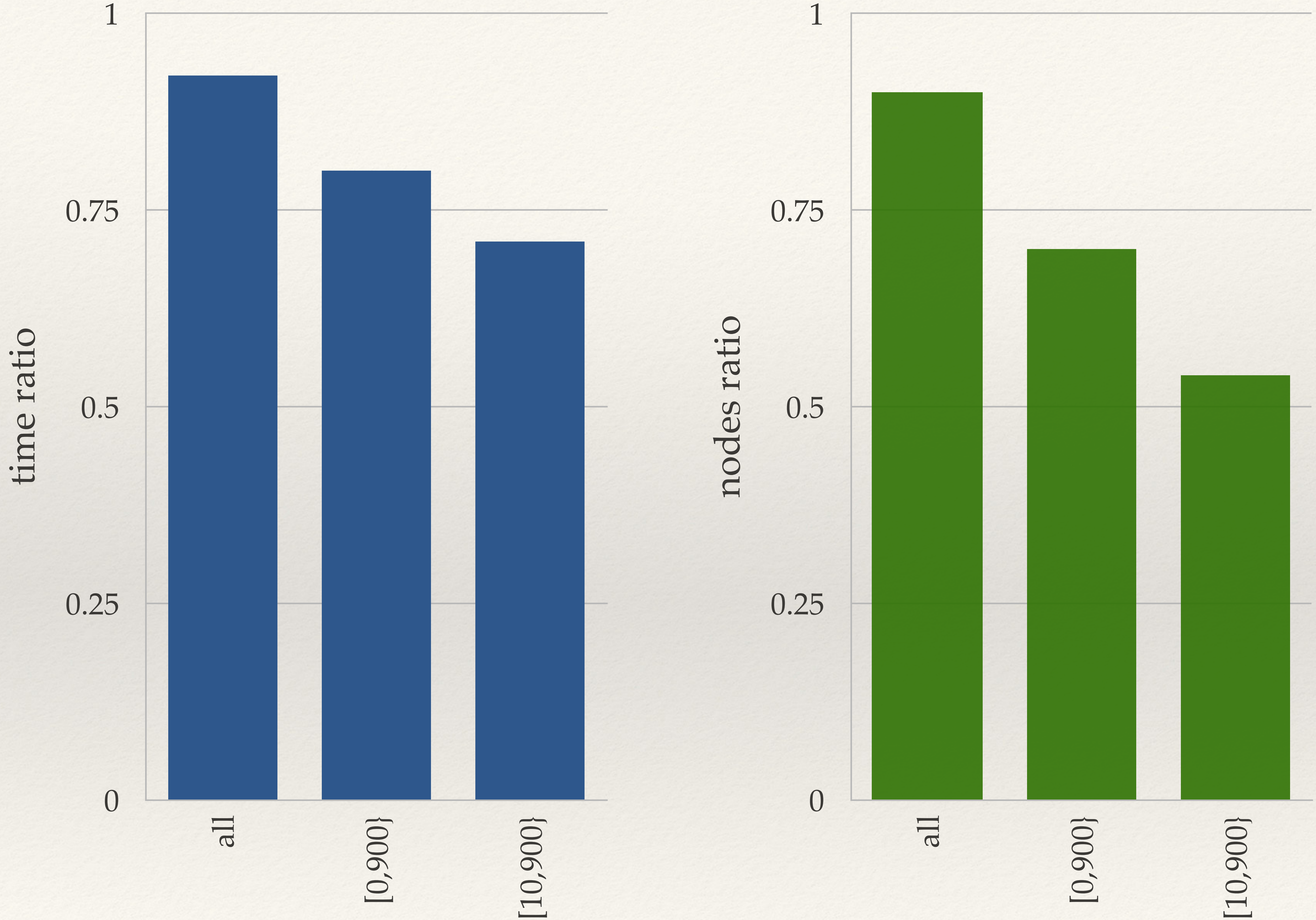




4 threads - 900s timelimit

(Primal) Heuristics

- Both formulations sometimes struggle in finding good feasible solutions (sometimes even the first…)
- On the other hand finding feasible solutions for delete-free planning tasks is easy…
- So we implemented some quick greedy heuristics to provide MIP starts for our models, based on h_{ADD}
  - At each step, evaluate the applicable actions by computing the h_{ADD} value of the state we would reach
  - Peak the best, breaking ties randomly



Ratios w.r.t. vertex_elimination
4 threads - 900s timelimit

A TSP approach?

- Still not satisfied by the current models
  - One is too weak, the other too heavy
- Can we deal with causal acyclicity in a different way?
  - State of the art for TSP does exactly that, via SECs
- Let's try to do the same:
  - Keep only the base model
  - Add lazy constraints on the fly to enforce acyclicity

Subtour Elimination Constraints

- Each integer solution x is associated with a causal graph G_x (encoded by the variables $x_{a,p}$), which is a subgraph of G_Π
- Any cycle C in G_x gives a violated SEC of the form:

$$\sum_{(p,q)\in C} x_{a,p} \leq |C| - 1$$

- Can be separated in linear time with a graph visit

Landmark Constraints

- A disjunctive landmark L is a set of actions such that at least one must be present in any feasible plan

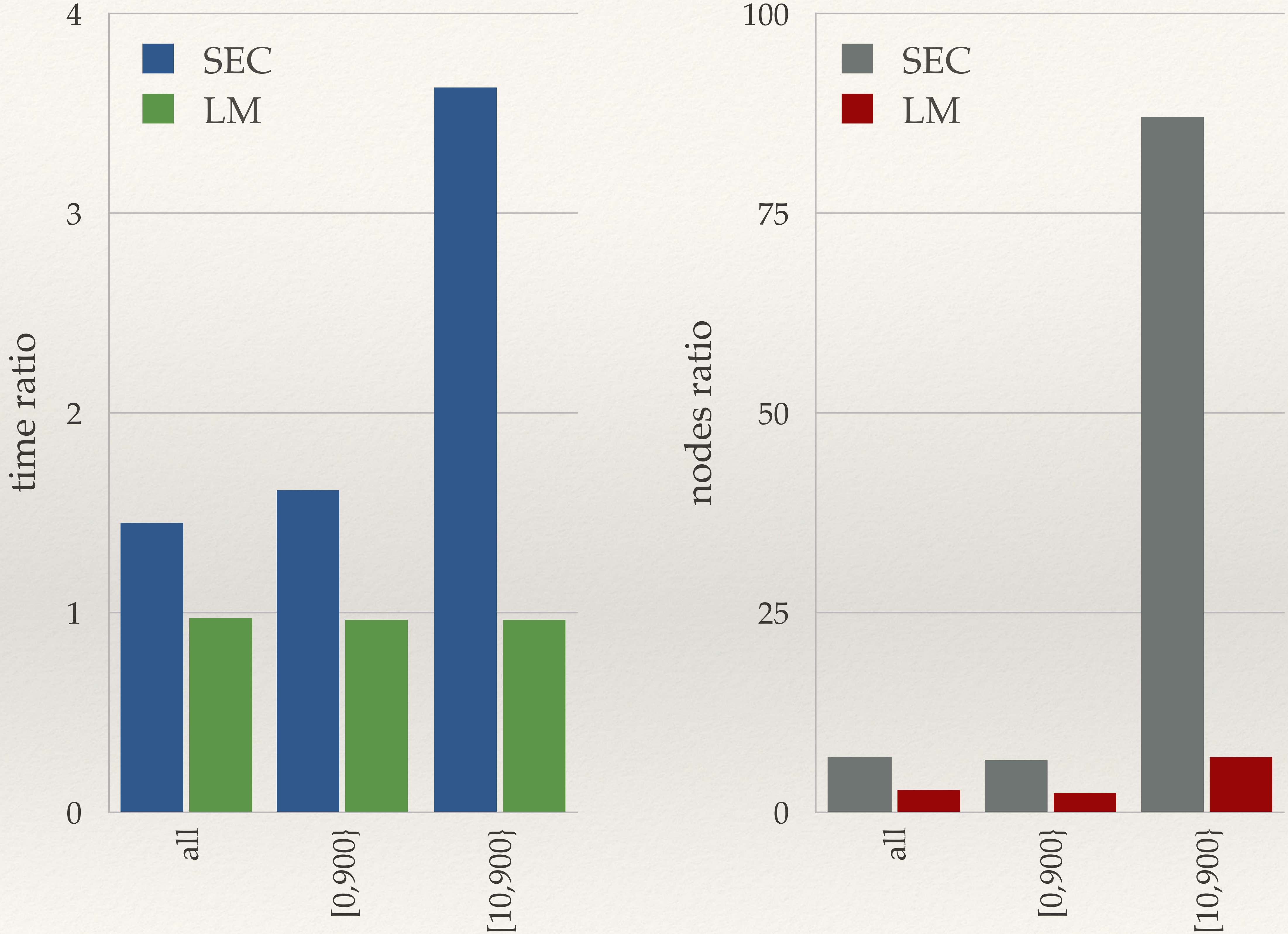
$$\sum_{a \in A} x_a \geq 1$$

- Delete-free planning is equivalent to solving a hitting set problem over all its landmarks

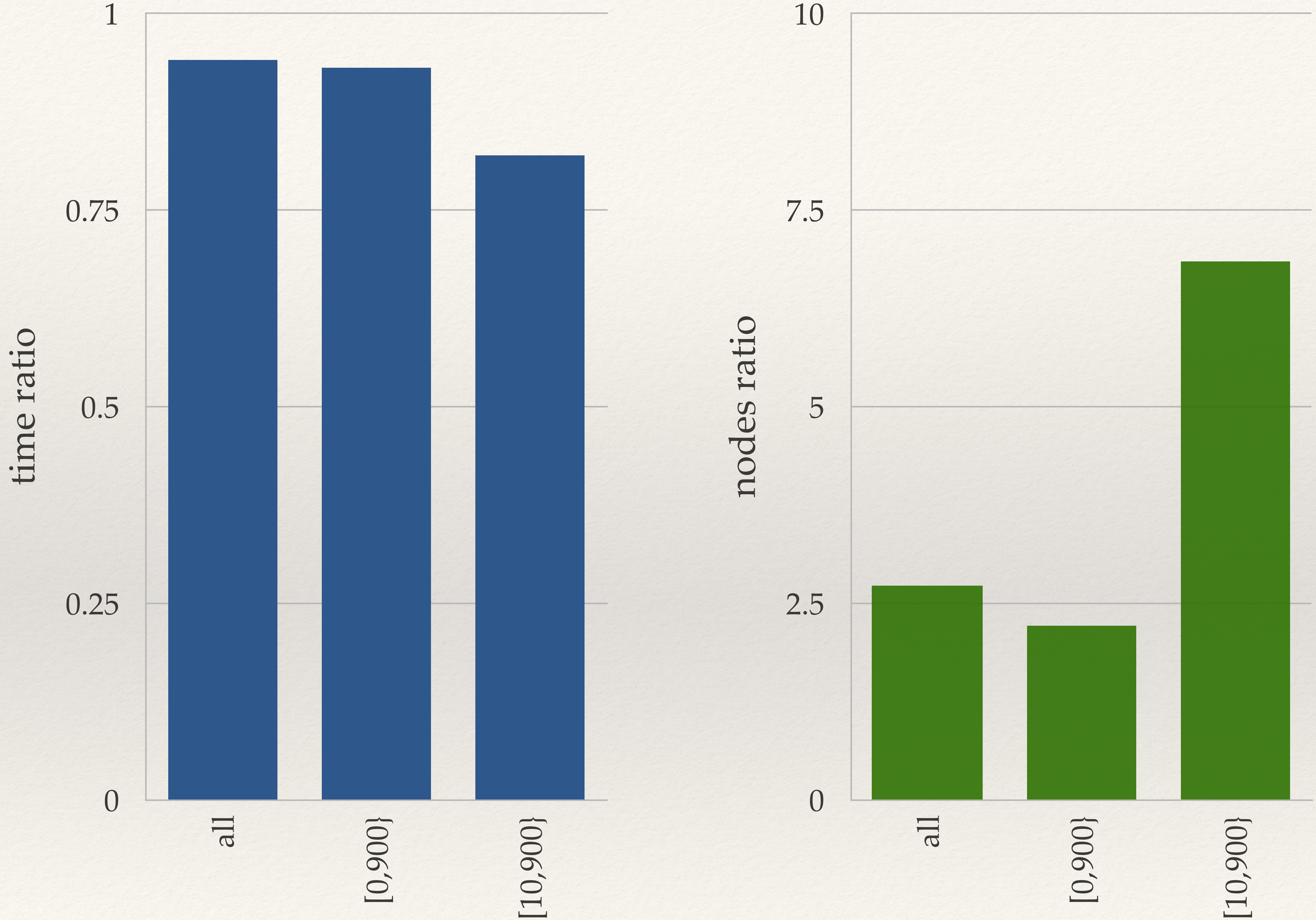
- Landmark constraints can be used as lazy constraintsto break cycles

- Can be separated in linear time via a simple combinatorial algorithm!

Bonet & Castillo "A Complete Algorithm for Generating Landmarks", 2011
Haslum, Slaney & Thiebaux "Minimal Landmarks for Optimal Delete-Free Planning", 2012



Ratios w.r.t. vertex_elim + ws
4 threads - 900s timelimit



Ratios w.r.t. vertex_elim + ws
4 threads - 900s timelimit

What about fractional solutions?

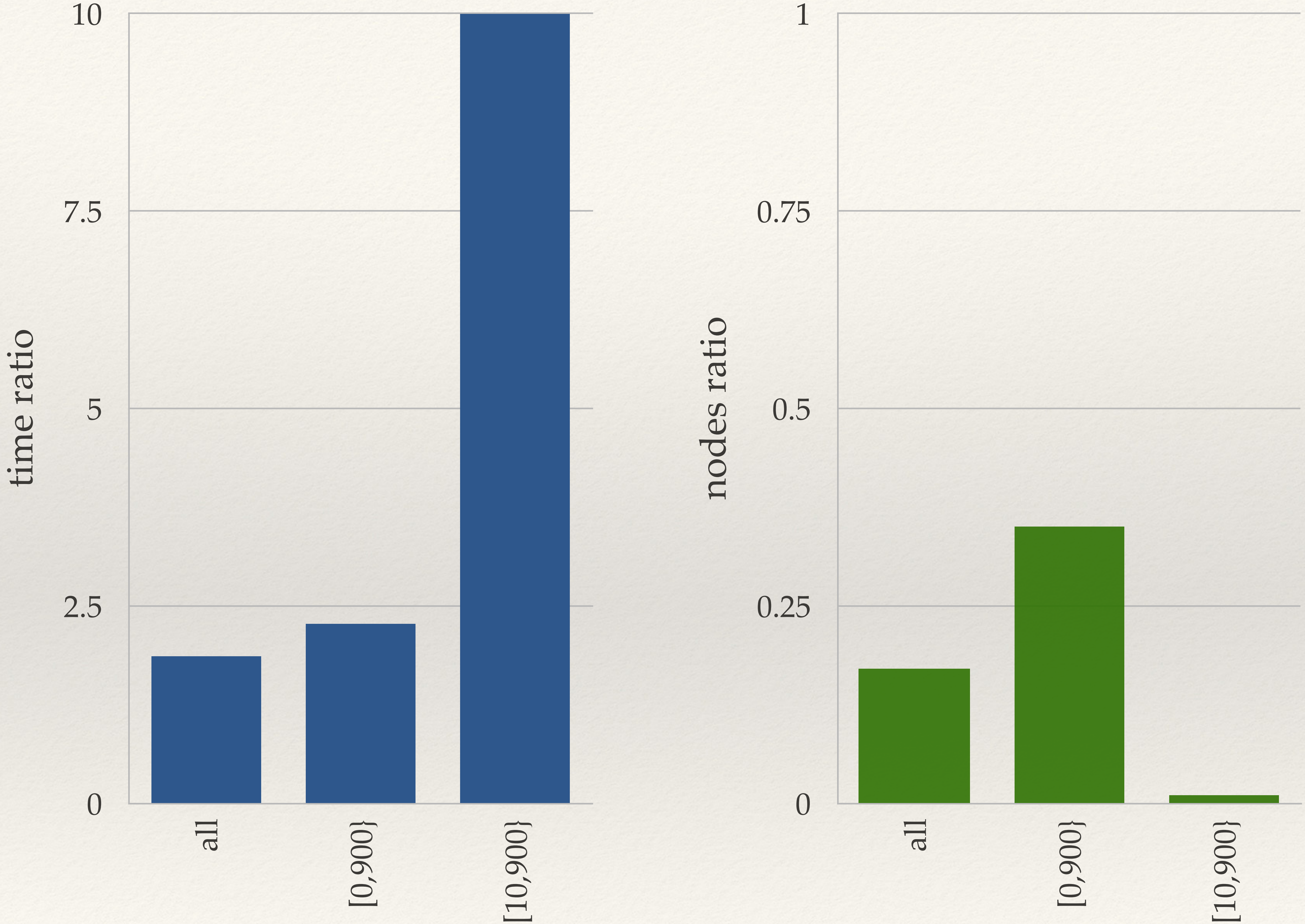
SEC:

- A fractional solution corresponds to a weighted causal graph (fractional weights)
- A violated SEC corresponds to a cycle of weight > |C|-1
- After some manipulation can be expressed as a mininum weight cycle problem
- Can be solved in polynomial time with a combinatorial algorithm based on shortest paths

What about fractional solutions?

Landmarks:

- Could not find a polynomial exact separation procedure so far (but this is very preliminary)
- For the moment, we resort to a MIP formulation based on the definition of landmarks from cut-sets
  - Given a partition $(S, P \setminus S)$ of the facts such that the goal G is in $P \setminus S$, the labels of the causal graph crossing the cut form by definition a landmark



Ratios w.r.t. vertex_elim + ws + lazy
4 threads - 900s timelimit

What went wrong?

- Results very preliminary :-(
- Landmark separation not very efficient (but numbers with SECs are not qualitatively different)
- Root cutloop takes forever (and kills parallelism):
  - Warm start landmarks cuts via some quick heuristic (like LM-cut)?
  - Separate more cuts per iteration?
  - Stabilize cutloop with in-out strategies?

Conclusions

- AI planning is a nice application MIP technology can contribute to
- We could improve (a bit) over state of the art for delete-free formulations with standard techniques in our community
- Still much to be done, in particular for separating fractional solutions (and what about branching?)