Arguments in Favor of Allowing a Modeler to Constrain Action Repetitions

Pascal Lauer[1,2]

[1]Australian National University, Australia
[2]Saarland Informatics Campus, Germany

Pipeline To Find Plans Fast

refine ↻

task ⟶ encoding ⟶ planner ⟶ plan

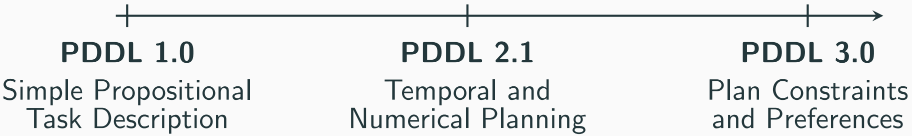
Pipeline To Find Plans Fast

refine

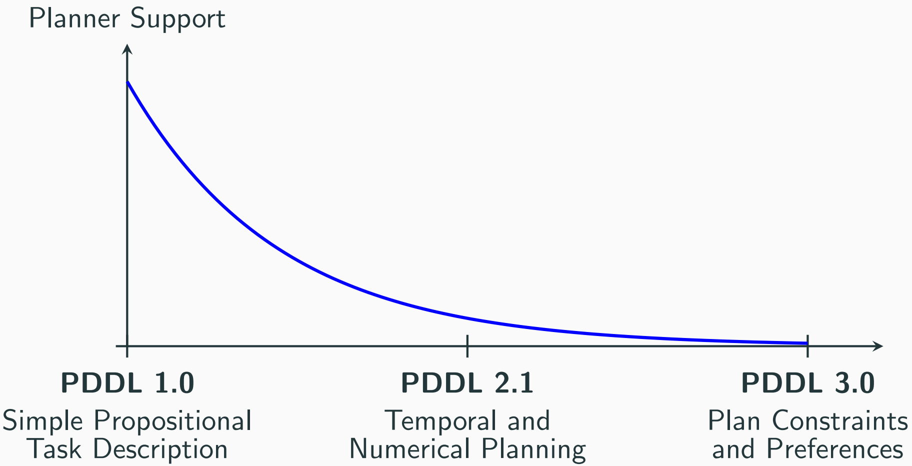
task ⟶ encoding ⟶ planner ⟶ plan

Why can't I just be given a better encoding?

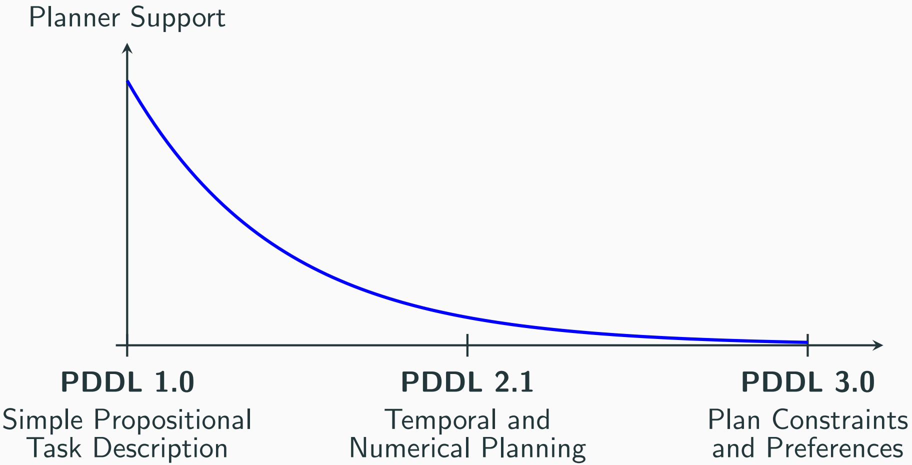
PDDL Feature Scale



PDDL Feature Scale vs Feature Support

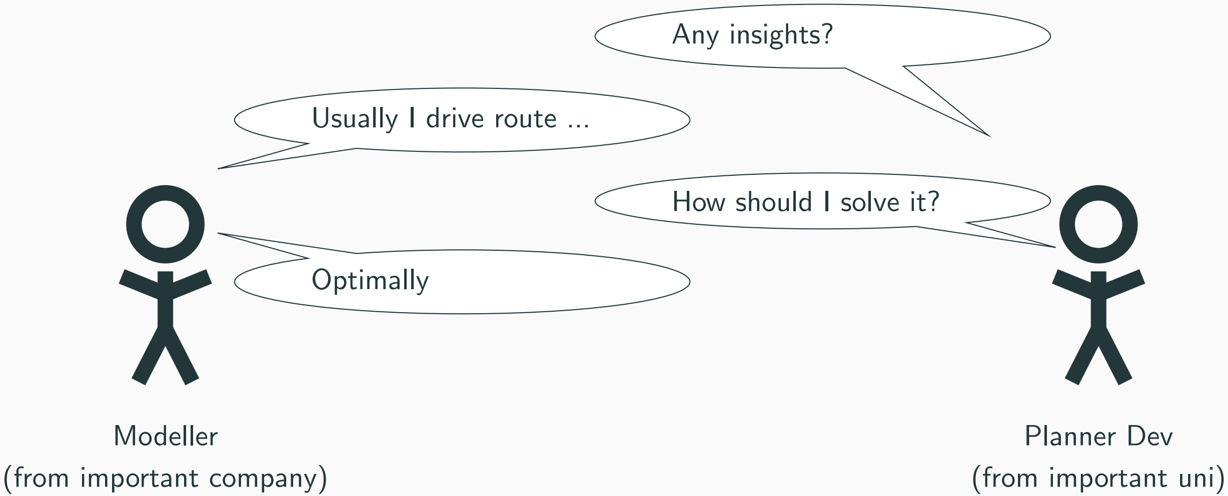


PDDL Feature Scale vs Feature Support



Observation: For solver support we want simple descriptions.

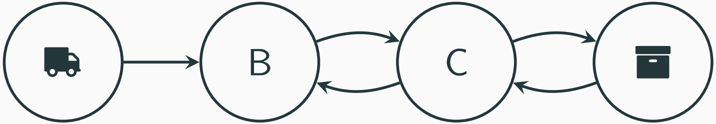
So just don't be lazy right?



One simple constraint:

A bound on action repetitions!

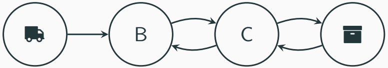
Transport Domain



Truck t_1 at A

Package p_1 at D

Peliminaries



Assume Ground STRIPS representation

- Facts $\{at(pkg, D),\ at(tr, A),\ in(pkg, tr), ...\}$
- Actions with $a = (pre(a), add(a), del(a))$ to $load_{tr,pkg,l}$, $pickup_{tr,pkg,l}$, $drive_{tr}$
- Initial State $\{at(tr, A), at(pkg, D)\}$
- Goal $\{at(pkg, B)\}$

Complexity question: Is there a plan for a given task?

Bounding Action Repetitions

Definition (Bound Map)

A plan π respects a bound map $b : A \to \mathbb{N} \cup \{\infty\}$ iff for every action $a \in A$ the number of occurrences of a in π does not exceed $b(a)$.

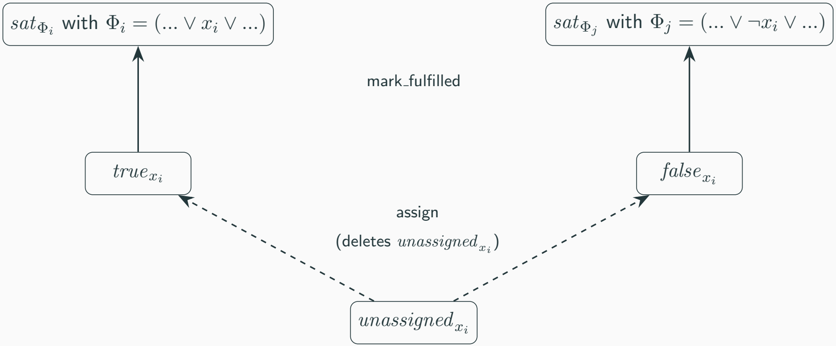
Theorem

Deciding whether there is a plan for a classical planning task that respect bound b, where each $b(a)$ is at most some constant $k \in \mathbb{N}^+$, is NP-complete.

Proof

Hardness (for $k = 1$):

Reduction from 3-SAT with formula $\Phi = \Phi_1 \wedge \cdots \wedge \Phi_n$ (Bylander, 1994)



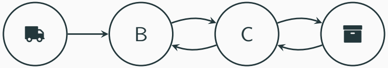
Membership:

Plan length bounded by $\sum_{a \in A} b(a)$ (polynomial) $\Rightarrow$ Guess and check

Bounding Plans vs Bounding Actions

Why not just bound the plan length then? (Remember $\sum_{a \in A} b(a)$)

- Remember:

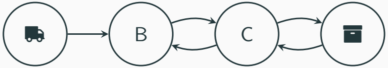


  Anybody can see that a package can be delivered directly without first placing it at your neighbor's door.

Bounding Plans vs Bounding Actions

Why not just bound the plan length then? (Remember $\sum_{a \in A} b(a)$)

- Remember:



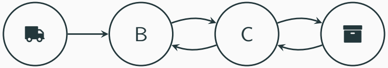
  Anybody can see that a package can be delivered directly without first placing it at your neighbor's door.

- I doubt that the bound on total plan length is obvious to the average human being.

Bounding Plans vs Bounding Actions

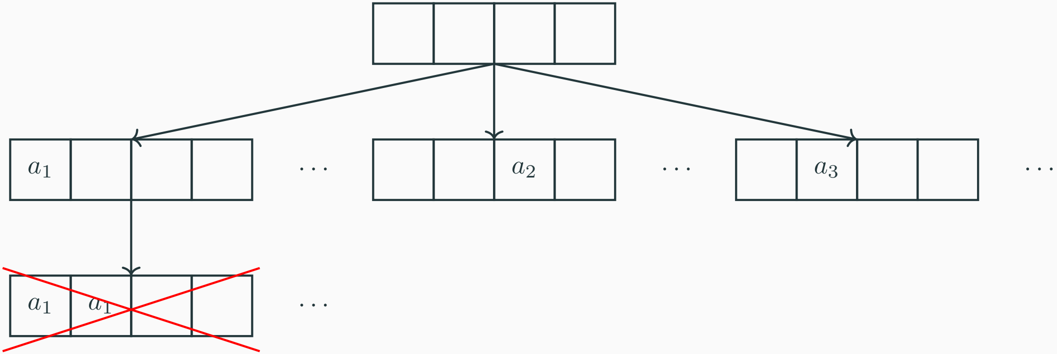
Why not just bound the plan length then? (Remember $\sum_{a \in A} b(a)$)

- Remember:

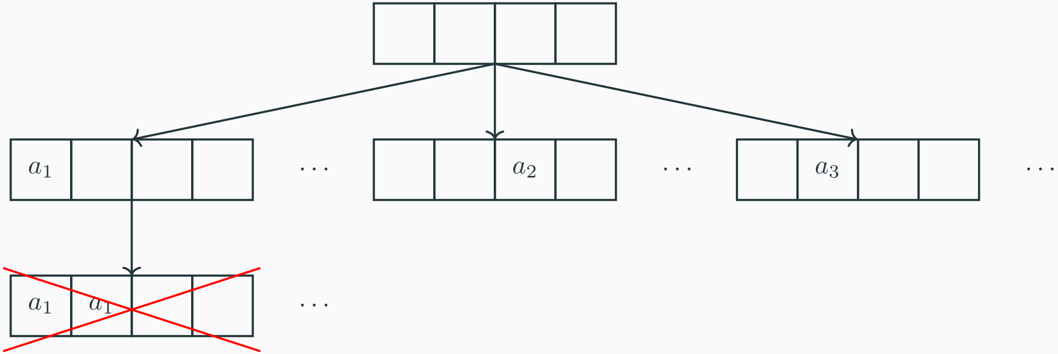


  Anybody can see that a package can be delivered directly without first placing it at your neighbor's door.
- I doubt that the bound on total plan length is obvious to the average human being.
- At least for constraint solvers this helps **a lot** more.

High Level Constraint Search (SAT, ILP, ...)



High Level Constraint Search (SAT, ILP, ...)



(The total plan length would not prune this node.)

But what if you can not bound action repetitions?

Simple Actions

Definition

An action $a \in A$ is *simple* if it has exactly one fact $f \in F$ in its precondition and delete list, i.e., $\{f\} = pre(a) = del(a)$, and every add effect $f' \in add(a)$ is poly-mutex[1] with f.

E.g.:

$Drive(t, from, to)$: pre: $at(t, from)$

add: $at(t, to)$

del: $at(t, from)$

[1] A mutex relation we can determine in polynomial time.

Simple Actions

Definition

An action $a \in A$ is *simple* if it has exactly one fact $f \in F$ in its precondition and delete list, i.e., $\{f\} = pre(a) = del(a)$, and every add effect $f' \in add(a)$ is poly-mutex[1] with f or never deleted.

E.g.:

$$Drive(t, from, to): \quad \text{pre: } at(t, from)$$
$$\text{add: } at(t, to)$$
$$\text{del: } at(t, from)$$

Allows tracking that you have been at some position at some point, e.g.: TSP or Visitall

[1] A mutex relation we can determine in polynomial time.

Simple Actions

Theorem

Deciding plan existence for a planning task with simple actions is NP-complete.

Simple Actions

Theorem

Deciding plan existence for a planning task with simple actions is NP-complete.

Theorem

Deciding whether there is a plan for a classical planning where all non-simple actions a occur at most $k \in \mathbb{N}^+$ times is NP-complete.

Preliminary Experiments

(Automatically) encode model with simple actions into Minizinc. (Visitall, TSP, ...)

- If actions are bounded to be repeated at most 1 times:
  Solves all tasks ≤ 1 minute optimally

Preliminary Experiments

(Automatically) encode model with simple actions into Minizinc. (Visitall, TSP, ...)

- If actions are bounded to be repeated at most 1 times:
  Solves all tasks ≤ 1 minute optimally
- Isn't the case if unbounded.

Preliminary Experiments

(Automatically) encode model with simple actions into Minizinc. (Visitall, TSP, ...)

- If actions are bounded to be repeated at most 1 times:
  Solves all tasks ≤ 1 minute optimally
- Isn't the case if unbounded.
- Isn't the case for state of the art planner (like Scorpion).

Preliminary Experiments

(Automatically) encode model with simple actions into Minizinc. (Visitall, TSP, ...)

- If actions are bounded to be repeated at most 1 times:
  Solves all tasks ≤ 1 minute optimally
- Isn't the case if unbounded.
- Isn't the case for state of the art planner (like Scorpion).
- Includes all IPC tasks for Visitall.

Thank You :)