On the Complexity of Correcting HTN Domain Models

Songtuan Lin, Pascal Bercher
The Australian National University
{songtuan.lin, pascal.bercher}.anu.edu.au

What If a Planner Acts Different from a Human's Expectation?

The plan offered by a planner differs from the one a user has in mind.

- We want to explain this happens $\implies$ Counter-factual explanations.
- We want to change the planning model so that the planner's behavior is in line with the human's expectation $\implies$ Modeling assistance.

What Are We Going to Do?

Changing the planning model so that the given plan will be solution!

- Computational complexity of deciding whether such changes exist.
- Our investigation is in the context of HTN planning.
- We consider the given plan in different forms, e.g., a partially ordered or a sequential plan.

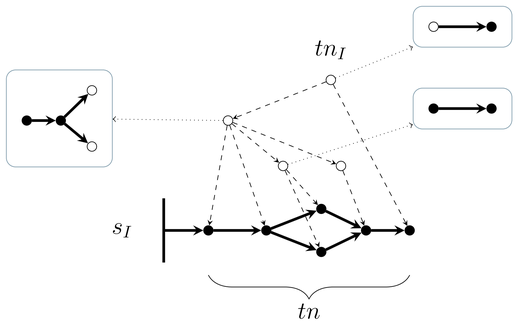

Figure 1: *An example of an HTN planning problem, its solution, and its decomposition. The solution task network tn is a refinement of the initial task network tn_I which contains solely one compound task. Every linearisation of tn is executable in the initial state s_I.*

Solution Criterion of an HTN Planning Problem

A task network tn is a solution *iff*

- It is a refinement of the given initial task network.
- ~~It possesses one executable linearisation~~ (Standard one & NP-complete).
- *Every* linearisation of tn is executable (Hybrid planning & tractable).
  - ▶ We focus on this definition to eliminate the extra hardness source.

What Changes Are Allowed?

We consider four elementary model change operations.

- Adding and removing a primitive task from a method.
- Adding and removing an ordering constraint between primitive tasks in a method.

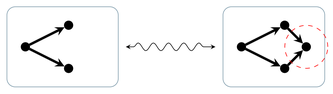

Figure 2: *Adding and deleting actions (primitive tasks).*

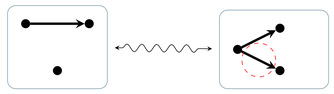

Figure 3: *Adding and deleting ordering constraints.*

Provided Only a Partially Ordered Plan: Formalism

Given an HTN planning problem P and a task network tn, we want to decide whether we can change the methods in P so that tn will be a solution to P.

- Demanding that every linearisation of tn should be executable.
- NP-complete independent of what or how many changes are allowed.

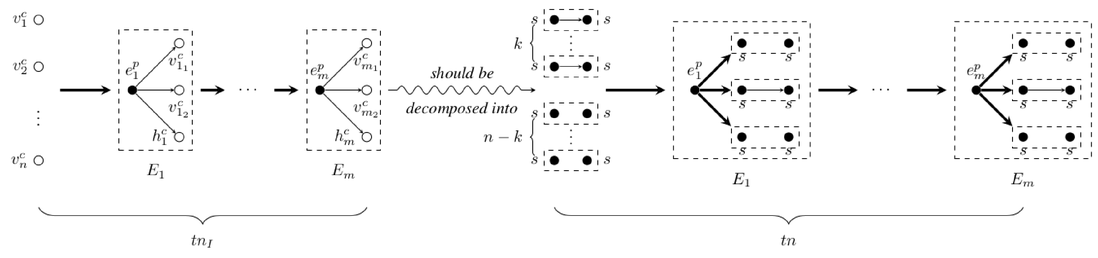

Figure 4: *The reduction from the independent set problem. Each thick arrow represents a set of ordering constraints specifying that the tasks in the lhs are ordered before those in the rhs. Each thin arrow denotes a single ordering constraint. s is a primitive task.*

When Only Deleting Ordering Constraints Is Allowed:Example

Deciding whether a task network tn can be turned into a solution is NP-complete when only deleting ordering constraints is allowed.

- Reduction from the *independent set* problem.
  - ▶ Given a graph $G = (V, E)$ and $k \in \mathbb{N}$, we want to decide whether there exists a $V' \subseteq V$ such that $|V'| = k$ and there are no two vertices in V' that are connected to each other by an edge.
- Constructing the initial task network to encode G and tn to encode the solution criterion of the independent set problem, see Fig.4.
  - ▶ One compound task for each vertex: $v_i \in V \implies v_i^c$ is a compound task.
  - ▶ One primitive task for each edge: $e_i \in E \implies e_i^p$ is a primitive task.
  - ▶ Block E_i encodes that the edge e_i connects the vertices v_{i_1} and v_{i_2}.
- All other changes are redundant even if they are allowed $\implies$ NP-complete independent of what changes are allowed.

Provided a Plan as Well as a Method Sequence: Formalism

Given an HTN planning problem P, a method sequence $\overline{m}$, and a task network tn, we want to decide whether we can change the methods in P so that tn is a solution to P which is refined by $\overline{m}$.

- Eliminating one more hardness source (which decomposition method sequence leads to tn?).
- Demanding that every linearisation of tn should be executable (again).
- NP-complete independent of what or how many changes are allowed as well.
  - ▶ *Special*: Constant time decidable if $\overline{m}$ has one *unique* method.

Provided an action sequence: Formalism

Given an HTN planning problem P and an action sequence π, we want to decide whether we can change the methods in P so that π is a linearisation of a solution to P.

- Demanding that tn only need to have one executable linearisation.
- NP-complete independent of what or how many changes are allowed.

Complexity	Plan	Methods?	Solution
	PO	NO	All
NP-complete	PO	YES	All
	TO	NO	One

Summary: All variants are NP-complete independent of what and how many changes are allowed. The column 'Plan' specifies whether the given plan is partially ordered (PO) or totally ordered (TO). The column 'Solution' specifies whether we demand that all linearisations of tn are executable or at least one is executable.