On the Computational Complexity of Plan Verification, (Bounded) Plan-Optimality Verification, and Bounded Plan Existence

Songtuan Lin[1] Conny Olz[2] Malte Helmert[3]
Pascal Bercher[1]

[1]The Australian National University
[2]Ulm University
[3]University of Basel

Feb 2024

Introduction

Objective

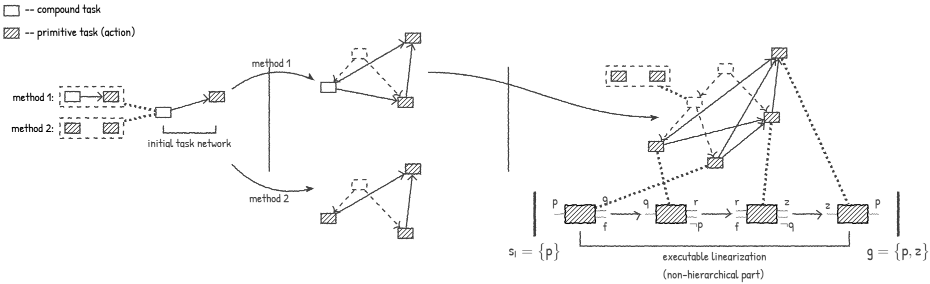
We want to study the computational complexity of several problems centered at the bounded plan existence problem.

Problems we are going to study:

- Plan Verification
- Bounded Plan Existence
- (Bounded) Plan Optimality Verification

Background

- Classical (non-hierarchical) planning – Find a sequence of actions (i.e., a plan) that achieves a goal

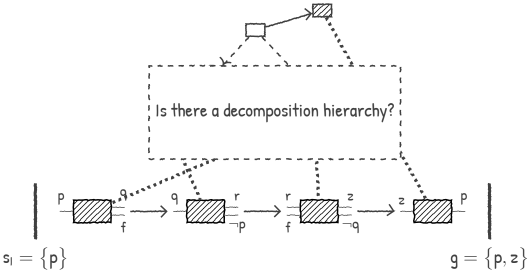


- Hierarchical planning – Keep decomposing compound tasks until a primitive action sequence is obtained

Plan Verification (Complexity Ranges from P to NEXPTIME)

Plan Verification

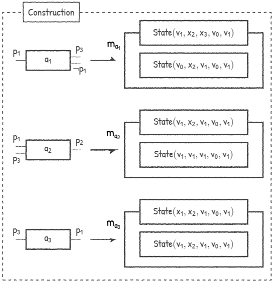
The plan verification problem is to decide whether a plan is a solution to a planning problem.

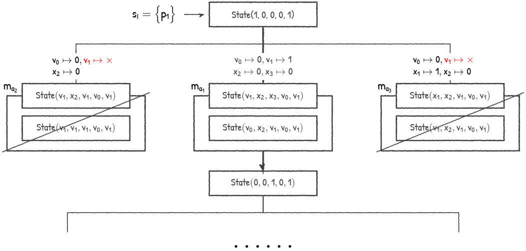


- It is trivial for classical planning with both the grounded and lifted settings
- It is NP-complete for grounded hierarchical planning
- For lifted hierarchical planning, it is PSPACE-hard (but in NEXPTIME)

Encoding a Classical Planning Problem as a Plan Verification Problem

- For hierarchical planning with the lifted setting, plan verification is PSPACE-hard (but in NEXPTIME)
- Decomposition hierarchies can simulate state transitions:

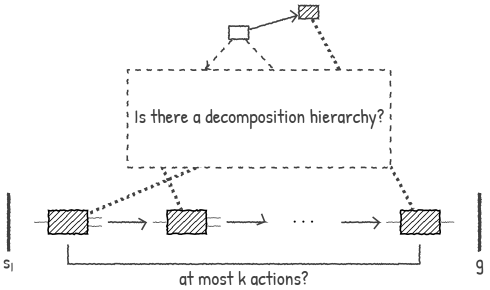




Bounded Plan Existence (Complexity Ranges from NP-complete to NEXPTIME-complete)

Bounded Plan Existence

The bounded plan existence problem is to decide whether there exists a solution to a planning problem of length up to a given bound.



- Its complexity coincides with the complexity of the plan verification problem
- We can guess and verify a plan up to length k

Upper Bound for the Bounded Plan Existence Problem

$$T_V^N(||\Pi|| + ||\pi||) + T_G^N(||k||)$$

- T_V^N is a function denoting the runtime for non-deterministically verifying whether the plan π is a solution to the planning problem Π, with respect to the encoding size of Π and π.
- T_G^N is a function denoting the runtime for guessing non-deterministically a plan of length up to k, with respect to the encoding size of k.

Property Pretaining to the Bounded Plan Existence Problem

For any planning formalism, the bounded plan existence problem is in NEXPTIME if the following two conditions hold:

- Each action can be encoded in polynomially many bits wrt. the encoding size of the planning problem (i.e., $T^N_G(||k||$ is exponential to $||k||$)).
- The plan verification problem under the respective formalism is in NP (i.e., $T^N_V(||\Pi|| + ||\pi||)$ is polynomial to $||\Pi|| + ||\pi||$ and hence exponential to $||k||$).

The following results thus follow:

- The bounded plan existence problem in grounded and lifted classical planning and grounded HTN planning is in NEXPTIME.

Complexity Results

- The bounded plan existence problem in grounded and lifted classical planning and grounded HTN planning is in NEXPTIME.
  - It is actually PSPACE-complete for grounded classical planning and NEXPTIME-complete for lifted classical planning and grounded HTN planning.
- For lifted HTN planning, the problem is NEXPTIME-hard and in 2NEXPTIME.

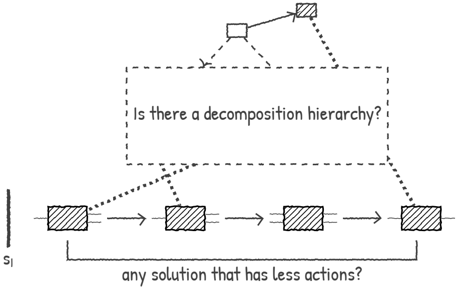
If the bound is encoded in unary, the following results hold:

- The bounded plan existence problem for (both grounded and lifted) classical planning and grounded HTN planning is NP-complete.
- It is PSPACE-hard and in NEXPTIME for lifted HTN planning.

Optimality Verification (The Complement of Bounded Plan Existence)

(Bounded) Optimality Verification

The (bounded) optimality verification problem is to decide whether a plan is a solution to a planning problem and its length is not far from the length of an optimal solution by a given bound.



- It is the complement of the bounded plan existence problem with the bound given in unary

Summary

	Plan Verification	k-length Plan Existence		Plan Optimality Verification	Bounded Plan Optimality Verification	
		k in binary	k in unary		plan given	only plan length given
				Classical Planning		
ground	In P	PSPACE-complete	NP-complete	coNP-complete	coNP-complete	PSPACE-complete
lifted	In P	NEXPTIME-complete	NP-complete	coNP-complete	coNP-complete	coNEXPTIME-complete
				Hierarchical Planning		
ground	NP-complete	NEXPTIME-complete	NP-complete	coNP-complete	coNP-complete	coNEXPTIME-complete
lifted	PSPACE-hard	NEXPTIME-hard	PSPACE-hard	PSPACE-hard	PSPACE-hard	coNEXPTIME-hard
	In NEXPTIME	In 2NEXPTIME	In NEXPTIME	In coNEXPTIME	In coNEXPTIME	In co2NEXPTIME