

# Planning and Optimization

## B5. Computational Complexity of Planning: Background

Gabriele Röger and Thomas Keller

Universität Basel

October 15, 2018

# Planning and Optimization

October 15, 2018 — B5. Computational Complexity of Planning: Background

## B5.1 Motivation

## B5.2 Background: Complexity Theory

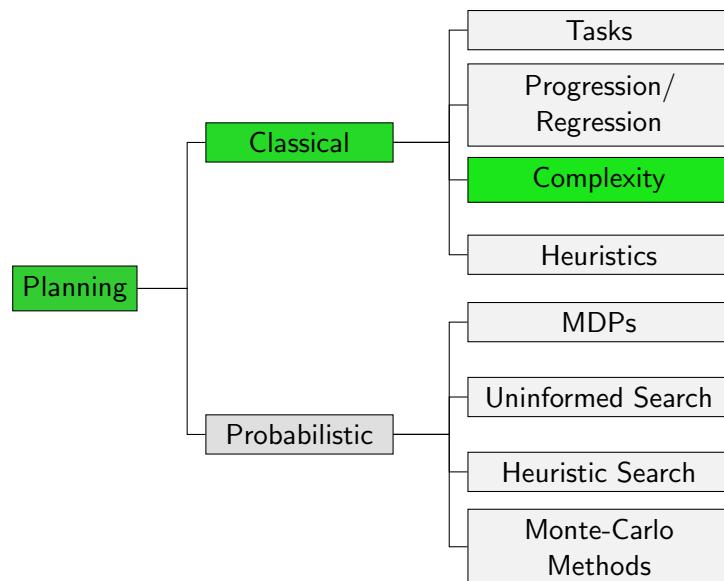
## B5.3 (Bounded-Cost) Plan Existence

## B5.4 PSPACE-Completeness of Planning

## B5.5 More Complexity Results

## B5.6 Summary

## Content of this Course



## B5.1 Motivation

## How Difficult is Planning?

- ▶ Using **progression** and a state-space search algorithm like breadth-first search, planning can be solved in **polynomial time** in the size of the **transition system** (i.e., the number of states).
- ▶ However, the number of states is **exponential** in the number of **state variables**, and hence in general exponential in the size of the input to the planning algorithm.
- ~~ Do non-exponential planning algorithms exist?
- ~~ What is the precise **computational complexity** of planning?

## Why Computational Complexity?

- ▶ **understand** the problem
- ▶ know what is **not** possible
- ▶ find interesting **subproblems** that are easier to solve
- ▶ distinguish **essential features** from **syntactic sugar**
  - ▶ Is STRIPS planning easier than general planning?
  - ▶ Is planning for FDR tasks harder than for propositional tasks?

## B5.2 Background: Complexity Theory

## Reminder: Complexity Theory

### Need to Catch Up?

- ▶ We assume knowledge of complexity theory:
  - ▶ languages and decision problems
  - ▶ Turing machines: NTMs and DTMNs;
  - ▶ polynomial equivalence with other models of computation
  - ▶ complexity classes: P and NP
  - ▶ polynomial reductions
- ▶ If you are not familiar with these topics, we recommend **Chapters C8, E1, E2** of the **Theory of Computer Science** course at <https://dmi.unibas.ch/de/studium/computer-science-informatik/fs18/main-lecture-theory-of-computer-science/>

Note: the **space complexity classes** (DSPACE, NSPACE, PSPACE, NPSPACE) go beyond the content of the prerequisite course.



## Time and Space Complexity Classes

### Definition (DTIME, NTIME, DSPACE, NSPACE)

Let  $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ .

Complexity class  $\text{DTIME}(f)$  contains all languages accepted in time  $f$  by some DTM.

Complexity class  $\text{NTIME}(f)$  contains all languages accepted in time  $f$  by some NTM.

Complexity class  $\text{DSPACE}(f)$  contains all languages accepted in space  $f$  by some DTM.

Complexity class  $\text{NSPACE}(f)$  contains all languages accepted in space  $f$  by some NTM.

## Polynomial Time and Space Classes

Let  $\mathcal{P}$  be the set of polynomials  $p : \mathbb{N}_0 \rightarrow \mathbb{N}_0$  whose coefficients are natural numbers.

### Definition (P, NP, PSPACE, NPSPACE)

$$P = \bigcup_{p \in \mathcal{P}} \text{DTIME}(p)$$

$$NP = \bigcup_{p \in \mathcal{P}} \text{NTIME}(p)$$

$$PSPACE = \bigcup_{p \in \mathcal{P}} \text{DSPACE}(p)$$

$$NPSPACE = \bigcup_{p \in \mathcal{P}} \text{NSPACE}(p)$$

## Polynomial Complexity Class Relationships

### Theorem (Complexity Class Hierarchy)

$$P \subseteq NP \subseteq PSPACE = NPSPACE$$

Proof.

$P \subseteq NP$  and  $PSPACE \subseteq NPSPACE$  are obvious because deterministic Turing machines are a special case of nondeterministic ones.

$NP \subseteq NPSPACE$  holds because a Turing machine can only visit polynomially many tape cells within polynomial time.

$PSPACE = NPSPACE$  is a special case of a classical result known as Savitch's theorem (Savitch 1970). □

## B5.3 (Bounded-Cost) Plan Existence

## The Propositional Planning Problem

### Definition (Plan Existence)

The **plan existence** problem (PLANEx) is the following decision problem:

**GIVEN:** propositional planning task  $\Pi$   
**QUESTION:** Is there a plan for  $\Pi$ ?

~ $\sim$  decision problem analogue of **satisficing planning**

### Definition (Bounded-Cost Plan Existence)

The **bounded-cost plan existence** problem (BCPLANEx) is the following decision problem:

**GIVEN:** propositional planning task  $\Pi$ , cost bound  $K \in \mathbb{N}_0$   
**QUESTION:** Is there a plan for  $\Pi$  with cost at most  $K$ ?

~ $\sim$  decision problem analogue of **optimal planning**

## Plan Existence vs. Bounded-Cost Plan Existence

### Theorem (Reduction from PLANEx to BCPLANEx)

$\text{PLANEx} \leq_p \text{BCPLANEx}$

#### Proof.

Consider a propositional planning task  $\Pi$  with  $n$  state variables. Let  $c_{\max}$  be the maximal cost of all actions of  $\Pi$ .

$\Pi$  is solvable iff there is solution with cost at most  $c_{\max} \cdot (2^n - 1)$  because a solution need not visit any state twice.

~ $\sim$  map instance  $\Pi$  of PLANEx to instance  $\langle \Pi, c_{\max} \cdot (2^n - 1) \rangle$  of BCPLANEx

~ $\sim$  polynomial reduction □

## B5.4 PSPACE-Completeness of Planning

## Membership in PSPACE

### Theorem

$\text{BCPLANEx} \in \text{PSPACE}$

#### Proof.

Show  $\text{BCPLANEx} \in \text{NPSPACE}$  and use Savitch's theorem.

Nondeterministic algorithm:

```
def plan( $\langle V, I, O, \gamma \rangle, K$ ):  

     $s := I$   

     $k := K$   

    loop forever:  

        if  $s \models \gamma$ : accept  

        guess  $o \in O$   

        if  $s \not\models \text{pre}(o)$ : fail  

        if  $\text{cost}(o) > k$ : fail  

         $s := s \llbracket o \rrbracket$   

         $k := k - \text{cost}(o)$ 
```

□

## PSPACE-Hardness

Idea: generic reduction

- For an arbitrary fixed DTM  $M$  with space bound polynomial  $p$  and input  $w$ , generate planning task which is solvable iff  $M$  accepts  $w$  in space  $p(|w|)$ .
- For simplicity, restrict to TMs which never move to the left of the initial head position (no loss of generality).

## Reduction: State Variables

Let  $M = \langle \Sigma, \square, Q, q_0, q_Y, \delta \rangle$  be the fixed DTM, and let  $p$  be its space-bound polynomial.

Given input  $w_1 \dots w_n$ , define relevant tape positions  $X := \{1, \dots, p(n)\}$ .

### State Variables

- state $_q$  for all  $q \in Q$
- head $_i$  for all  $i \in X \cup \{0, p(n) + 1\}$
- content $_{i,a}$  for all  $i \in X$ ,  $a \in \Sigma_{\square}$

## Reduction: Initial State

Let  $M = \langle \Sigma, \square, Q, q_0, q_Y, \delta \rangle$  be the fixed DTM, and let  $p$  be its space-bound polynomial.

Given input  $w_1 \dots w_n$ , define relevant tape positions  $X := \{1, \dots, p(n)\}$ .

### Initial State

Initially true:

- state $_{q_0}$
- head $_1$
- content $_{i,w_i}$  for all  $i \in \{1, \dots, n\}$
- content $_{i,\square}$  for all  $i \in X \setminus \{1, \dots, n\}$

Initially false:

- all others

## Reduction: Operators

Let  $M = \langle \Sigma, \square, Q, q_0, q_Y, \delta \rangle$  be the fixed DTM, and let  $p$  be its space-bound polynomial.

Given input  $w_1 \dots w_n$ , define relevant tape positions  $X := \{1, \dots, p(n)\}$ .

### Operators

One operator for each transition rule  $\delta(q, a) = \langle q', a', d \rangle$  and each cell position  $i \in X$ :

- precondition: state $_q \wedge \text{head}_i \wedge \text{content}_{i,a}$
- effect:  $\neg \text{state}_q \wedge \neg \text{head}_i \wedge \neg \text{content}_{i,a}$   
 $\wedge \text{state}_{q'} \wedge \text{head}_{i+d} \wedge \text{content}_{i,a'}$

## Reduction: Goal

Let  $M = \langle \Sigma, \square, Q, q_0, q_Y, \delta \rangle$  be the fixed DTM, and let  $p$  be its space-bound polynomial.

Given input  $w_1 \dots w_n$ , define **relevant tape positions**  
 $X := \{1, \dots, p(n)\}$ .

Goal  
 $\text{state}_{q_Y}$

## PSPACE-Completeness of STRIPS Plan Existence

**Theorem (PSPACE-Completeness; Bylander, 1994)**

PLANEx and BCPLANEx are PSPACE-complete.

*This is true even if only STRIPS tasks are allowed.*

**Proof.**

Membership for BCPLANEx was already shown.

Hardness for PLANEx follows because we just presented a polynomial reduction from an arbitrary problem in PSPACE to PLANEx. (Note that the reduction only generates STRIPS tasks.)

Membership for PLANEx and hardness for BCPLANEx follow from the polynomial reduction from PLANEx to BCPLANEx.  $\square$

## B5.5 More Complexity Results

## More Complexity Results

In addition to the basic complexity result presented in this chapter, there are many special cases, generalizations, variations and related problems studied in the literature:

- ▶ different **planning formalisms**
  - ▶ e.g., finite-domain representation, nondeterministic effects, partial observability, schematic operators, numerical state variables
- ▶ **syntactic restrictions** of planning tasks
  - ▶ e.g., without preconditions, without conjunctive effects, STRIPS without delete effects
- ▶ **semantic restrictions** of planning task
  - ▶ e.g., restricting variable dependencies ("causal graphs")
- ▶ **particular planning domains**
  - ▶ e.g., Blocksworld, Logistics, FreeCell

## Complexity Results for Different Planning Formalisms

Some results for different planning formalisms:

- ▶ **FDR tasks:**
  - ▶ same complexity as for propositional tasks (“folklore”)
  - ▶ also true for the SAS<sup>+</sup> special case
- ▶ **nondeterministic effects:**
  - ▶ fully observable: EXP-complete (Littman, 1997)
  - ▶ unobservable: EXPSPACE-complete (Haslum & Jonsson, 1999)
  - ▶ partially observable: 2-EXP-complete (Rintanen, 2004)
- ▶ **schematic operators:**
  - ▶ usually adds one exponential level to PLANEX complexity
  - ▶ e.g., classical case EXPSPACE-complete (Erol et al., 1995)
- ▶ **numerical state variables:**
  - ▶ undecidable in most variations (Helmert, 2002)

## B5.6 Summary

## Summary

- ▶ **PSPACE:** decision problems solvable in **polynomial space**
- ▶  $P \subseteq NP \subseteq PSPACE = NPSPACE$ .
- ▶ **Propositional planning is PSPACE-complete.**
- ▶ This is true both for **satisficing** and **optimal** planning.
- ▶ The hardness proof is a polynomial reduction that translates an **arbitrary polynomial-space DTM** into a **STRIPS task**:
  - ▶ DTM configurations are encoded by state variables.
  - ▶ Operators simulate transitions between DTM configurations.
  - ▶ The DTM accepts an input iff there is a plan for the corresponding STRIPS task.
- ▶ This implies that there is **no polynomial algorithm** for classical planning unless  $P = PSPACE$ .
- ▶ It also means that planning is not polynomially reducible to any problem in NP unless  $NP = PSPACE$ .