# Planning and Optimization

## C5. SAT Planning: Parallel Encoding

Malte Helmert and Gabriele Röger

Universität Basel

October 11, 2023



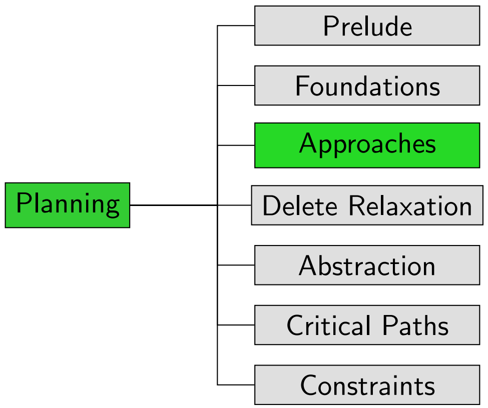
# Planning and Optimization

October 11, 2023 — C5. SAT Planning: Parallel Encoding





# Content of this Course








# C5.1 Introduction

# Efficiency of SAT Planning

- All other things being equal, the most important aspect for efficient SAT solving is the number of propositional variables in the input formula.
- For sufficiently difficult inputs, runtime scales exponentially in the number of variables.

$\rightsquigarrow$ Can we make SAT planning more efficient by using fewer variables?





# Number of Variables

Reminder:

- given propositional planning task $\Pi = \langle V, I, O, \gamma \rangle$
- given horizon $T \in \mathbb{N}_0$

**Variables of the SAT Formula**

- propositional variables $v^i$ for all $v \in V$, $0 \le i \le T$ encode state after $i$ steps of the plan
- propositional variables $o^i$ for all $o \in O$, $1 \le i \le T$ encode operator(s) applied in $i$-th step of the plan

$\rightsquigarrow$ $|V| \cdot (T + 1) + |O| \cdot T$ variables

$\rightsquigarrow$ SAT solving runtime usually exponential in $T$





# Parallel Plans and Interference

Can we get away with shorter horizons?

Idea:

- allow parallel plans in the SAT encoding: multiple operators can be applied in the same step if they do not interfere

**Definition (Interference)**

Let $O' = \{o_1, \dots, o_n\}$ be a set of operators applicable in state $s$.

We say that $O'$ is interference-free in $s$ if

- for all permutations $\pi$ of $O'$, $s[\![\pi]\!]$ is defined, and
- for all permutations $\pi$, $\pi'$ of $O'$, $s[\![\pi]\!] = s[\![\pi']\!]$.

We say that $O'$ interfere in $s$ if they are not interference-free in $s$.





# Parallel Plan Extraction

- If we can rule out interference, we can allow multiple operators at the same time in the SAT encoding.
- A parallel plan (with multiple $o^i$ used for the same $i$) extracted from the SAT formula can then be converted into a "regular" plan by ordering the operators within each time step arbitrarily.

## Challenges for Parallel SAT Encodings

Two challenges remain:
- our current SAT encoding does not allow concurrent operators
- how do we ensure that our plans are interference-free?





# C5.2 Adapting the SAT Encoding





## Reminder: Sequential SAT Encoding (1)

**Sequential SAT Formula (1)**

initial state clauses:
- $v^0$ for all $v \in V$ with $I(v) = \mathbf{T}$
- $\neg v^0$ for all $v \in V$ with $I(v) = \mathbf{F}$

goal clauses:
- $\gamma^T$

operator selection clauses:
- $o_1^i \vee \dots \vee o_n^i$ for all $1 \leq i \leq T$

operator exclusion clauses:
- $\neg o_j^i \vee \neg o_k^i$ for all $1 \leq i \leq T$, $1 \leq j < k \leq n$

$\rightsquigarrow$ operator exclusion clauses must be adapted





## Sequential SAT Encoding (2)

**Sequential SAT Formula (2)**

precondition clauses:
- $o^i \rightarrow pre(o)^{i-1}$ for all $1 \leq i \leq T$, $o \in O$

positive and negative effect clauses:
- $(o^i \wedge \alpha^{i-1}) \rightarrow v^i$ for all $1 \leq i \leq T$, $o \in O$, $v \in V$
- $(o^i \wedge \delta^{i-1} \wedge \neg\alpha^{i-1}) \rightarrow \neg v^i$ for all $1 \leq i \leq T$, $o \in O$, $v \in V$

positive and negative frame clauses:
- $(o^i \wedge v^{i-1} \wedge \neg v^i) \rightarrow \delta^{i-1}$ for all $1 \leq i \leq T$, $o \in O$, $v \in V$
- $(o^i \wedge \neg v^{i-1} \wedge v^i) \rightarrow \alpha^{i-1}$ for all $1 \leq i \leq T$, $o \in O$, $v \in V$

where $\alpha = \mathit{effcond}(v, \mathit{eff}(o))$, $\delta = \mathit{effcond}(\neg v, \mathit{eff}(o))$.

$\rightsquigarrow$ frame clauses must be adapted

## Adapting the Operator Exclusion Clauses: Idea

Reminder: operator exclusion clauses $\neg o^i_j \vee \neg o^i_k$
for all $1 \leq i \leq T$, $1 \leq j < k \leq n$

- Ideally: replace with clauses that express "for all states $s$, the operators selected at time $i$ are interference-free in $s$"
- but: testing if a given set of operators interferes in any state is itself an NP-complete problem

⇝ use something less heavy: a sufficient condition for interference-freeness that can be expressed at the level of pairs of operators





## Conflicting Operators

- Intuitively, two operators conflict if
  - one can disable the precondition of the other,
  - one can override an effect of the other, or
  - one can enable or disable an effect condition of the other.
- If no two operators in a set $O'$ conflict, then $O'$ is interference-free in all states.
- This is still difficult to test, so we restrict attention to the STRIPS case in the following.

**Definition (Conflicting STRIPS Operator)**

Operators $o$ and $o'$ of a STRIPS task $\Pi$ conflict if
- $o$ deletes a precondition of $o'$ or vice versa, or
- $o$ deletes an add effect of $o'$ or vice versa.





## Adapting the Operator Exclusion Clauses: Solution

Reminder: operator exclusion clauses $\neg o^i_j \vee \neg o^i_k$
for all $1 \leq i \leq T$, $1 \leq j < k \leq n$

Solution:

**Parallel SAT Formula: Operator Exclusion Clauses**

operator exclusion clauses:
- $\neg o^i_j \vee \neg o^i_k$ for all $1 \leq i \leq T$, $1 \leq j < k \leq n$ such that $o_j$ and $o_k$ conflict





## Adapting the Frame Clauses: Idea

Reminder: frame clauses
$(o^i \wedge v^{i-1} \wedge \neg v^i) \rightarrow \delta^{i-1}$ for all $1 \leq i \leq T$, $o \in O$, $v \in V$
$(o^i \wedge \neg v^{i-1} \wedge v^i) \rightarrow \alpha^{i-1}$ for all $1 \leq i \leq T$, $o \in O$, $v \in V$

What is the problem?

- These clauses express that if $o$ is applied at time $i$ and the value of $v$ changes, then $o$ caused the change.
- This is no longer true if we want to be able to apply two operators concurrently.

⇝ Instead, say "If the value of $v$ changes, then some operator must have caused the change."

## Adapting the Frame Clauses: Solution

Reminder: frame clauses

$(o^i \wedge v^{i-1} \wedge \neg v^i) \rightarrow \delta^{i-1}$ for all $1 \leq i \leq T$, $o \in O$, $v \in V$

$(o^i \wedge \neg v^{i-1} \wedge v^i) \rightarrow \alpha^{i-1}$ for all $1 \leq i \leq T$, $o \in O$, $v \in V$

Solution:

**Parallel SAT Formula: Frame Clauses**

positive and negative frame clauses:

- $(v^{i-1} \wedge \neg v^i) \rightarrow ((o_1^i \wedge \delta_{o_1}^{i-1}) \vee \dots \vee (o_n^i \wedge \delta_{o_n}^{i-1}))$
  for all $1 \leq i \leq T$, $v \in V$
- $(\neg v^{i-1} \wedge v^i) \rightarrow ((o_1^i \wedge \alpha_{o_1}^{i-1}) \vee \dots \vee (o_n^i \wedge \alpha_{o_n}^{i-1}))$
  for all $1 \leq i \leq T$, $v \in V$

where $\alpha_o = \mathit{effcond}(v, \mathit{eff}(o))$, $\delta_o = \mathit{effcond}(\neg v, \mathit{eff}(o))$,
$O = \{o_1, \dots, o_n\}$.

For STRIPS, these are in clause form.





# C5.3 Summary





## Summary

- As a rule of thumb, SAT solvers generally perform better on formulas with fewer variables.
- Parallel encodings reduce the number of variables by shortening the horizon needed to solve a planning task.
- Parallel encodings replace the constraint that operators are not applied concurrently by the constraint that conflicting operators are not applied concurrently.
- To make parallelism possible, the frame clauses also need to be adapted.