

Planning and Optimization

C6. SAT Planning: Parallel Encoding

Malte Helmert and Gabriele Röger

Universität Basel

October 12, 2026

Planning and Optimization

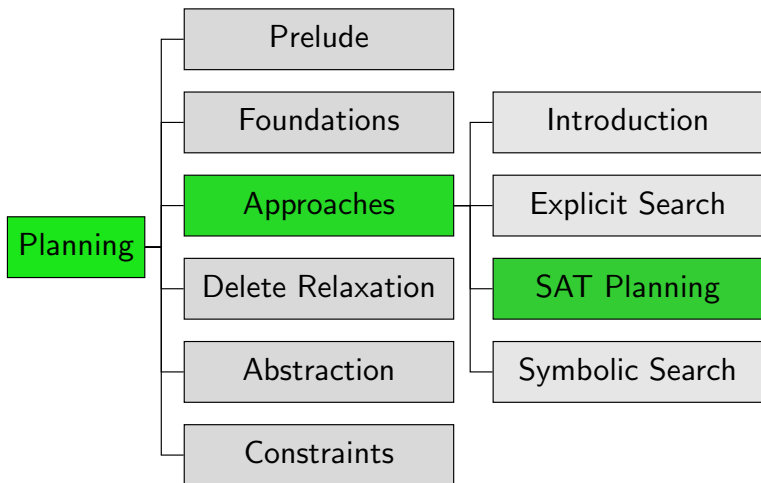
October 12, 2026 — C6. SAT Planning: Parallel Encoding

C6.1 Introduction

C6.2 Adapting the SAT Encoding

C6.3 Summary

Content of the Course



C6.1 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.
- ~> 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 $v \in V$, $0 \leq i \leq T$
encode state after i steps of the plan
- ▶ propositional variables o^i for all $o \in O$, $1 \leq i \leq T$
encode operator(s) applied in i -th step of the plan

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

~> SAT solving runtime usually **exponential in T**

Parallel Plans and Commutativity

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 (commutative, interfere)

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

We say that O' is **commutative** in s if

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

We say that the set O' **interferes** in s if it is not commutative in s .

Parallel Plan Extraction

- ▶ If we can guarantee commutativity, 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 concurrent operators are **commutative**?

C6.2 Adapting the SAT Encoding

Reminder: Sequential SAT Encoding (1)

Sequential SAT Encoding (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:

▶ γ^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 $1 \leq i \leq T, 1 \leq j < k \leq n$

↪ operator exclusion clauses must be adapted

Reminder: Sequential SAT Encoding (2)

Sequential SAT Encoding (2)

precondition clauses:

- ▶ $\neg o^i \vee pre(o)^{i-1}$ for all $1 \leq i \leq T, o \in O$

positive and negative effect clauses:

- ▶ $\neg o^i \vee \neg \alpha^{i-1} \vee v^i$ for all $1 \leq i \leq T, o \in O, v \in V$
- ▶ $\neg o^i \vee \alpha^{i-1} \vee \neg \delta^{i-1} \vee \neg v^i$ for all $1 \leq i \leq T, o \in O, v \in V$

positive and negative frame clauses:

- ▶ $\neg o^i \vee \neg v^{i-1} \vee \delta^{i-1} \vee v^i$ for all $1 \leq i \leq T, o \in O, v \in V$
- ▶ $\neg o^i \vee \alpha^{i-1} \vee v^{i-1} \vee \neg v^i$ for all $1 \leq i \leq T, o \in O, v \in V$

where $\alpha = trig(v, eff(o))$, $\delta = trig(\neg v, eff(o))$.

$\rightsquigarrow$ rewrite clauses as implications

Sequential SAT Encoding (2) Rewritten as Implications

Sequential SAT Encoding (2) Rewritten

precondition clauses:

- ▶ $o^i \rightarrow \text{pre}(o)^{i-1}$ 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 $\alpha = \text{trig}(v, \text{eff}(o))$, $\delta = \text{trig}(\neg v, \text{eff}(o))$.

↪ frame clauses must be adapted

Adapting the Operator Exclusion Clauses: Idea

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

- ▶ **Ideally:** replace with clauses that express “for all states s , the operators selected at time i are **commutative** 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 commutativity 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 commutative 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 Π **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 $\neg o_j^i \vee \neg o_k^i$
for all $1 \leq i \leq T, 1 \leq j < k \leq n$

Solution:

Parallel SAT Formula: Operator Exclusion Clauses

operator exclusion clauses:

- ▶ $\neg o_j^i \vee \neg o_k^i$ 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

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

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

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

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

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

Solution:

Parallel SAT Formula: Frame Clauses

positive and negative frame clauses:

$$\begin{aligned} \blacktriangleright \quad & (v^{i-1} \wedge \neg v^i) \rightarrow ((o_1^i \wedge \delta_{o_1}^{i-1}) \vee \dots \vee (o_n^i \wedge \delta_{o_n}^{i-1})) \\ & \text{for all } 1 \leq i \leq T, v \in V \end{aligned}$$

$$\begin{aligned} \blacktriangleright \quad & (\neg v^{i-1} \wedge v^i) \rightarrow ((o_1^i \wedge \alpha_{o_1}^{i-1}) \vee \dots \vee (o_n^i \wedge \alpha_{o_n}^{i-1})) \\ & \text{for all } 1 \leq i \leq T, v \in V \end{aligned}$$

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

For STRIPS, these are in clause form.

C6.3 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.