Learning Hierarchical Policies by Iteratively Reducing the Width of Sketch Rules

Dominik Drexler\textsuperscript{1}  
Jendrik Seipp\textsuperscript{1}  
Hector Geffner\textsuperscript{2,1}

September 8, 2023 at KR conference

\textsuperscript{1}Linköping University, Linköping, Sweden,  
\textsuperscript{2}RWTH Aachen University, Aachen, Germany
Classical Planning

• **Input:**
  1. **Domain** $D$:
     - Set of predicates
     - Set of action schemas
  2. **Instance** $I$:
     - Set of objects
     - Set of ground atoms for the initial state $s_0$ and goal states $G$

• **Output:**
  - A plan, i.e., sequence of ground actions from $s_0$ to $s \in G$

**Example (Delivery)**
Generalized Planning

- **Input**: Class of classical planning problems $Q$ over common domain $D$
- **Output**: An algorithm $A$ that solves any $P \in Q$ in polynomial time w.r.t. input size

**Example (Delivery)**

Input: $Q_{Delivery}$ consists of all problems of delivering packages, 1-by-1, in a grid.

Output: $A$ is a hierarchical policy

- Note: has no solution for **intractable** classes (plan existence NP-hard) unless $P = NP$
Hierarchical policies involve the execution of sub-policies for achieving subgoals

Subgoals are important in planning where they are exploited as landmarks

Subgoals are important in RL where they appear as intrinsic rewards

The main challenge in learning hierarchical policies is how to define a hierarchy of sub-policies for achieving subgoals

We present a width-based characterization of hierarchical policies and how to learn them
Features $\Phi$

- $G$: all packages delivered?
- $H$: holding a package?
- $u$: number of undelivered packages
- $p$: distance to nearest package
- $t$: distance to target cell

Hierarchical Policy $\Pi_2$

$n_0: \{\neg G\} \mapsto \{G\}$

$n_1: \{u > 0\} \mapsto \{u\downarrow\}$

$n_2: \{\neg H, u > 0\} \mapsto \{H\}$

$n_3: \{H, u > 0\} \mapsto \{\neg H, u\downarrow\}$

$n_4: \{\neg H, p > 0\} \mapsto \{p\downarrow\}$

$n_5: \{\neg H, p = 0\} \mapsto \{H, p\}$

$n_6: \{H, t > 0\} \mapsto \{t\downarrow\}$

$n_7: \{H, t = 0\} \mapsto \{\neg H\}$
Planning Width [Lipovetzky and Geffner, 2012]

- Background theory of width
  - **Width** $w(P)$ measures the difficulty of a planning problem $P$
  - **Thm:** if $w(P) = k$ then $IW(k)$ solves $P$ optimally with resources $O(\exp(k))$

- Width in practice
  - Achieving a single goal atom: width is often small (1 or 2)
  - Achieving conjunctive goals: $SIW(k)$ calls $IW(k)$ to **achieve one goal atom at a time**

- Extensions
  - **Policy sketches** is a language that allows to define richer decompositions
Policy Sketches [Bonet and Geffner, 2021]

- A sketch $R$ is a set of rules of form $C \mapsto E$ over Boolean and numerical features $\Phi$ with sets of feature conditions $C$ and effects $E$
- **Sketch width** is max width of subproblems from class of problems $Q$:
  
  $$w_R(Q) = \max_{P \in Q, s \in S_R(P)} w(P[s, \bigcup_{r \in R} G_r(s)])$$
- **Thm**: if $w_R(Q) = k$ then $\text{SIW}_R(k)$ solves $P \in Q$ with resources $O(\exp(k))$

**Example (Delivery; 2-width sketch)**

$\{u > 0\} \mapsto \{u \downarrow\}$; Decrease # undelivered packages

**Example (Delivery; 1-width sketch)**

$\{\neg H, u > 0\} \mapsto \{H\}$; Get hold of undelivered package

$\{H, u > 0\} \mapsto \{\neg H, u \downarrow\}$; Deliver package
Hierarchical Policies: Formulation

- A hierarchical policy $\Pi$ for a class of problems $Q$ is a single rooted tree where every node $n$ has a sketch rule $r(n)$ with features over $Q$.

Example (Hierarchical policy $\Pi_2$ for Delivery)

- $n_0: \{\neg G\} \mapsto \{G\}$
- $n_1: \{u > 0\} \mapsto \{u\downarrow\}$
- $n_2: \{\neg H, u > 0\} \mapsto \{H\}$
- $n_3: \{H, u > 0\} \mapsto \{\neg H, u\downarrow\}$
- $n_4: \{\neg H, p > 0\} \mapsto \{p\downarrow\}$
- $n_5: \{\neg H, p = 0\} \mapsto \{H, p?\}$
- $n_6: \{H, t > 0\} \mapsto \{t\downarrow\}$
- $n_7: \{H, t = 0\} \mapsto \{\neg H\}$
Valid Hierarchical Policies

- **Valid hierarchical policy** recursively decomposes the target class of problems $Q$ into easier (smaller width) classes of subproblems $Q'$.
- The decomposition has constraints depending on three types of a node $n$:
  1. **Root node** $n$:
     - The rule $r(n)$ is $\{\neg G\} \mapsto \{G\}$ where $G$ is true only in the goal of any $P \in Q$.
     - The class of subproblems $Q_n = Q$.
  2. **Inner node** $n$:
     - The rules $r(n')$ of the children $n'$ of $n$ define a sketch $R$ whose **sketch width** $w_R(Q_n)$ is strictly less than the **width** $w(Q_n)$ of class $Q_n$.
     - The class of subproblems $Q'_n$ is derived from $R$ and $Q_n$.
  3. **Leaf node** $n$:
     - The **width** $w(Q_n)$ of class $Q_n$ is **zero** meaning that each $P \in Q_n$ is solvable by executing a single action.
Learning Hierarchical Policies

- Input:
  - Set of small training instances: $\mathcal{P} \subset \mathcal{Q}$
  - Width parameter: $k$
  - Maximum number of rules per learned sketch: $m$

- Initially, the hierarchical policy $\Pi_k$ contains a single root node $n_0$ with $\mathcal{Q}_{n_0} = \mathcal{P}$

- Iteratively refine leaf nodes $n$ with width $w(\mathcal{Q}_n) > 0$ as follows
  - Find sketch $R$ decomposing $\mathcal{Q}_n$ with width $w_R(\mathcal{Q}_n) = w(\mathcal{Q}_n) - 1$
  - Compute set of subproblems $\mathcal{Q}_{n'}$ for each child $n'$ with rule $r(n')$ from $R$

- We implemented the main operation of learning a sketch in ASP with Clingo [Gebser et al., 2019]
Learned Valid Hierarchical Policy $\Pi_2$ for $Q_{\text{Miconic}}$

Features $\Phi$

- $G$: all people served?
- $w$: # waiting people that are boardable
- $d$: # people unboardable at destination
- $b$: # boarded people
- $p$: # served people

Hierarchical Policy $\Pi_2$

- $n_0$: $\{\neg G\} \mapsto \{G\}$
- $n_1$: $\emptyset \mapsto \{p\uparrow\}$
- $n_2$: $\{b = 0\} \mapsto \{b\uparrow\}$
- $n_3$: $\{b > 0\} \mapsto \{b\downarrow\}$
- $n_4$: $\{w = 0\} \mapsto \{b?, w\uparrow\}$
- $n_5$: $\{w > 0\} \mapsto \{b\uparrow, w?\}$
- $n_6$: $\{d = 0\} \mapsto \{b?, d\uparrow\}$
- $n_7$: $\{d > 0\} \mapsto \{b\downarrow, d\downarrow\}$
Hierarchical Execution

Example (Delivery)

State $s_0$

$\{\neg G, \neg H, u = 1, p = 0, t = 1\}$

- $s_0$ $n_0: \{\neg G\} \Rightarrow \{G\}$
- $n_1: \{u > 0\} \Rightarrow \{u\downarrow\}$
- $n_2: \{\neg H, u > 0\} \Rightarrow \{H\}$
- $n_3: \{H, u > 0\} \Rightarrow \{\neg H, u\downarrow\}$
- $n_4: \{\neg H, p > 0\} \Rightarrow \{p\downarrow\}$
- $n_5: \{\neg H, p = 0\} \Rightarrow \{H, p?\}$
- $n_6: \{H, t > 0\} \Rightarrow \{t\downarrow\}$
- $n_7: \{H, t = 0\} \Rightarrow \{\neg H\}$
Hierarchical Execution

Example (Delivery)

State \( s_0 \)
\( \{¬G, ¬H, u = 1, p = 0, t = 1\} \)

\( n_0: \{¬G\} \mapsto \{G\} \)

\( n_1: \{u > 0\} \mapsto \{u↓\} \)

\( n_2: \{¬H, u > 0\} \mapsto \{H\} \)

\( n_3: \{H, u > 0\} \mapsto \{¬H, u↓\} \)

\( n_4: \{¬H, p > 0\} \mapsto \{p↓\} \)

\( n_5: \{¬H, p = 0\} \mapsto \{H, p?\} \)

\( n_6: \{H, t > 0\} \mapsto \{t↓\} \)

\( n_7: \{H, t = 0\} \mapsto \{¬H\} \)
Hierarchical Execution

Example (Delivery)

State $s_0$

$\{\neg G, \neg H, u = 1, p = 0, t = 1\}$

Diagram:

- $s_0$: $n_0: \{\neg G\} \rightarrow \{G\}$
- $s_0$: $n_1: \{u > 0\} \rightarrow \{u\downarrow\}$
- $s_0$: $n_2: \{\neg H, u > 0\} \rightarrow \{H\}$
- $n_4: \{\neg H, p > 0\} \rightarrow \{p\downarrow\}$
- $n_5: \{\neg H, p = 0\} \rightarrow \{H, p?\}$
- $n_6: \{H, t > 0\} \rightarrow \{t\downarrow\}$
- $n_7: \{H, t = 0\} \rightarrow \{\neg H\}$
Hierarchical Execution

Example (Delivery)

State $s_0$
$\{\neg G, \neg H, u = 1, p = 0, t = 1\}$

$\rightarrow$

$\begin{align*}
  s_0 & \Rightarrow n_0: \{\neg G\} \mapsto \{G\} \\
  s_0 & \Rightarrow n_1: \{u > 0\} \mapsto \{u\downarrow\} \\
  s_0 & \Rightarrow n_2: \{\neg H, u > 0\} \mapsto \{H\} \\
  s_0 & \Rightarrow n_3: \{H, u > 0\} \mapsto \{\neg H, u\downarrow\} \\
  s_0 & \Rightarrow n_4: \{\neg H, p > 0\} \mapsto \{p\downarrow\} \\
  s_0 & \Rightarrow n_5: \{\neg H, p = 0\} \mapsto \{H, p?\} \\
  s_0 & \Rightarrow n_6: \{H, t > 0\} \mapsto \{t\downarrow\} \\
  s_0 & \Rightarrow n_7: \{H, t = 0\} \mapsto \{\neg H\}
\end{align*}$
Hierarchical Execution

Example (Delivery)

State $s_0$
\{\neg G, \neg H, u = 1, p = 0, t = 1\}

State $s_1$
\{\neg G, H, u = 1, p = \infty, t = 1\}

$n_0: \neg G \mapsto \{G\}$
$n_1: \{u > 0\} \mapsto \{u\downarrow\}$
$n_2: \neg H, u > 0 \mapsto \{H\}$

$n_3: \{H, u > 0\} \mapsto \{\neg H, u\downarrow\}$
$n_4: \neg H, p > 0 \mapsto \{p\downarrow\}$

$n_5: \neg H, p = 0 \mapsto \{H, p\}$

$n_6: \{H, t > 0\} \mapsto \{t\downarrow\}$

$n_7: \{H, t = 0\} \mapsto \{\neg H\}$

pick(P)
Hierarchical Execution

Example (Delivery)

\[
\begin{align*}
\text{State } s_0 & \quad \{\neg G, \neg H, u = 1, p = 0, t = 1\} \\
\text{State } s_1 & \quad \{\neg G, H, u = 1, p = \infty, t = 1\}
\end{align*}
\]
Hierarchical Execution

Example (Delivery)

State $s_0$
\{\neg G, \neg H, u = 1, p = 0, t = 1\}

State $s_1$
\{\neg G, H, u = 1, p = \infty, t = 1\}

Example:

- $n_0$: \{\neg G\} $\rightarrow$ \{G\}
- $n_1$: \{u > 0\} $\rightarrow$ \{u↓\}
- $n_2$: \{\neg H, u > 0\} $\rightarrow$ \{H\}
- $n_3$: \{H, u > 0\} $\rightarrow$ \{\neg H, u↓\}
- $n_4$: \{\neg H, p > 0\} $\rightarrow$ \{p↓\}
- $n_5$: \{\neg H, p = 0\} $\rightarrow$ \{H, p?\}
- $n_6$: \{H, t > 0\} $\rightarrow$ \{t↓\}
- $n_7$: \{H, t = 0\} $\rightarrow$ \{\neg H\}
Hierarchical Execution

Example (Delivery)

State $s_0$
\{\neg G, \neg H, u = 1, p = 0, t = 1\}

State $s_1$
\{\neg G, H, u = 1, p = \infty, t = 1\}

$n_0$: \{\neg G\} $\mapsto$ \{G\}

$n_1$: \{u > 0\} $\mapsto$ \{u\}\downarrow

$n_2$: \{\neg H, u > 0\} $\mapsto$ \{H\}

$n_4$: \{\neg H, p > 0\} $\mapsto$ \{p\}\downarrow

$n_5$: \{\neg H, p = 0\} $\mapsto$ \{H, p?\}

$n_6$: \{H, t > 0\} $\mapsto$ \{t\}\downarrow

$n_7$: \{H, t = 0\} $\mapsto$ \{\neg H\}
Hierarchical Execution

Example (Delivery)

State $s_0$
\{\neg G, \neg H, u = 1, p = 0, t = 1\}

State $s_1$
\{\neg G, H, u = 1, p = \infty, t = 1\}

$n_0$: $\{\neg G\} \mapsto \{G\}$

$n_1$: $\{u > 0\} \mapsto \{u\downarrow\}$

$n_2$: $\{\neg H, u > 0\} \mapsto \{H\}$

$n_3$: $\{H, u > 0\} \mapsto \{\neg H, u\downarrow\}$

$n_4$: $\{\neg H, p > 0\} \mapsto \{p\downarrow\}$

$n_5$: $\{\neg H, p = 0\} \mapsto \{H, p?\}$

$n_6$: $\{H, t > 0\} \mapsto \{t\downarrow\}$

$n_7$: $\{H, t = 0\} \mapsto \{\neg H\}$
Hierarchical Execution

Example (Delivery)

\begin{itemize}
  \item State $s_0$: \{\neg G, \neg H, u = 1, p = 0, t = 1\}
  \item State $s_1$: \{\neg G, H, u = 1, p = \infty, t = 1\}
  \item State $s_2$: \{\neg G, H, u = 1, p = \infty, t = 0\}
\end{itemize}
Hierarchical Execution

Example (Delivery)

State $s_0$
\{\neg G, \neg H, u = 1, p = 0, t = 1\}

State $s_1$
\{\neg G, H, u = 1, p = \infty, t = 1\}

State $s_2$
\{\neg G, H, u = 1, p = \infty, t = 0\}

$n_0$: \{\neg G\} $\mapsto$ \{G\}

$n_1$: \{u > 0\} $\mapsto$ \{u\downarrow\}

$n_2$: \{\neg H, u > 0\} $\mapsto$ \{H\}

$n_3$: \{H, u > 0\} $\mapsto$ \{\neg H, u\downarrow\}

$n_4$: \{\neg H, p > 0\} $\mapsto$ \{p\downarrow\}

$n_5$: \{\neg H, p = 0\} $\mapsto$ \{H, p?\}

$n_6$: \{H, t > 0\} $\mapsto$ \{t\downarrow\}

$n_7$: \{H, t = 0\} $\mapsto$ \{\neg H\}
Hierarchical Execution

Example (Delivery)

\[
\begin{align*}
\text{State } s_0 & : \{\neg G, \neg H, u = 1, p = 0, t = 1\} \\
\text{State } s_1 & : \{\neg G, H, u = 1, p = \infty, t = 1\} \\
\text{State } s_2 & : \{\neg G, H, u = 1, p = \infty, t = 0\}
\end{align*}
\]

\[
\begin{align*}
\text{n}_0 : & \{\neg G\} \mapsto \{G\} \\
\text{n}_1 : & \{u > 0\} \mapsto \{u \downarrow\} \\
\text{n}_2 : & \{\neg H, u > 0\} \mapsto \{H\} \\
\text{n}_3 : & \{H, u > 0\} \mapsto \{\neg H, u \downarrow\} \\
\text{n}_4 : & \{\neg H, p > 0\} \mapsto \{p \downarrow\} \\
\text{n}_5 : & \{\neg H, p = 0\} \mapsto \{H, p?\} \\
\text{n}_6 : & \{H, t > 0\} \mapsto \{t \downarrow\} \\
\text{n}_7 : & \{H, t = 0\} \mapsto \{\neg H\}
\end{align*}
\]
Hierarchical Execution

Example (Delivery)

State $s_0$

$\{\neg G, \neg H, u = 1, p = 0, t = 1\}$

State $s_1$

$\{\neg G, H, u = 1, p = \infty, t = 1\}$

State $s_2$

$\{\neg G, H, u = 1, p = \infty, t = 0\}$

$n_0$: $\{\neg G\} \rightarrow \{G\}$

$n_1$: $\{u > 0\} \rightarrow \{u \downarrow\}$

$n_2$: $\{\neg H, u > 0\} \rightarrow \{H\}$

$n_3$: $\{H, u > 0\} \rightarrow \{\neg H, u \downarrow\}$

$n_4$: $\{\neg H, p > 0\} \rightarrow \{p \downarrow\}$

$n_5$: $\{\neg H, p = 0\} \rightarrow \{H, p?\}$

$n_6$: $\{H, t > 0\} \rightarrow \{t \downarrow\}$

$n_7$: $\{H, t = 0\} \rightarrow \{\neg H\}$

$n_8$: $\{\neg G, \neg H, u = 1, p = 0, t = 1\}$

$n_9$: $\{\neg G, H, u = 1, p = \infty, t = 1\}$

$n_{10}$: $\{\neg G, H, u = 1, p = \infty, t = 0\}$

$n_{11}$: $\{\neg G\} \rightarrow \{G\}$

$n_{12}$: $\{u > 0\} \rightarrow \{u \downarrow\}$

$n_{13}$: $\{\neg H, u > 0\} \rightarrow \{H\}$

$n_{14}$: $\{H, u > 0\} \rightarrow \{\neg H, u \downarrow\}$

$n_{15}$: $\{\neg H, p > 0\} \rightarrow \{p \downarrow\}$

$n_{16}$: $\{\neg H, p = 0\} \rightarrow \{H, p?\}$

$n_{17}$: $\{H, t > 0\} \rightarrow \{t \downarrow\}$

$n_{18}$: $\{H, t = 0\} \rightarrow \{\neg H\}$
Hierarchical Execution

Example (Delivery)

\[
\begin{align*}
\text{State } s_0 &: \{\neg G, \neg H, u = 1, p = 0, t = 1\} \\
\text{State } s_1 &: \{\neg G, H, u = 1, p = \infty, t = 1\} \\
\text{State } s_2 &: \{\neg G, H, u = 1, p = \infty, t = 0\} \\
\text{State } s_3 &: \{G, \neg H, u = 0, p = \infty, t = 0\}
\end{align*}
\]
Hierarchical Execution

Example (Delivery)

State $s_0$

$\{\neg G, \neg H, u = 1, p = 0, t = 1\}$

$n_0$: $\{\neg G\} \rightarrow \{G\}$

$n_1$: $\{u > 0\} \rightarrow \{u\downarrow\}$

$n_2$: $\{\neg H, u > 0\} \rightarrow \{H\}$

$n_3$: $\{H, u > 0\} \rightarrow \{\neg H, u\downarrow\}$

$n_4$: $\{\neg H, p > 0\} \rightarrow \{p\downarrow\}$

$n_5$: $\{\neg H, p = 0\} \rightarrow \{H, p?\}$

$n_6$: $\{H, t > 0\} \rightarrow \{t\downarrow\}$

$n_7$: $\{H, t = 0\} \rightarrow \{\neg H\}$

State $s_1$

$\{\neg G, H, u = 1, p = \infty, t = 1\}$

State $s_2$

$\{\neg G, H, u = 1, p = \infty, t = 0\}$

State $s_3$

$\{G, \neg H, u = 0, p = \infty, t = 0\}$
Hierarchical Execution

Example (Delivery)

State $s_0$

$\{ \neg G, \neg H, u = 1, p = 0, t = 1 \}$

State $s_1$

$\{ \neg G, H, u = 1, p = \infty, t = 1 \}$

State $s_2$

$\{ \neg G, H, u = 1, p = \infty, t = 0 \}$

State $s_3$

$\{ G, \neg H, u = 0, p = \infty, t = 0 \}$

$n_0$: $\{ \neg G \} \leftrightarrow \{ G \}$

$n_1$: $\{ u > 0 \} \leftrightarrow \{ u \downarrow \}$

$n_2$: $\{ \neg H, u > 0 \} \leftrightarrow \{ H \}$

$n_3$: $\{ H, u > 0 \} \leftrightarrow \{ \neg H, u \downarrow \}$

$n_4$: $\{ \neg H, p > 0 \} \leftrightarrow \{ p \downarrow \}$

$n_5$: $\{ \neg H, p = 0 \} \leftrightarrow \{ H, p \}$

$n_6$: $\{ H, t > 0 \} \leftrightarrow \{ t \downarrow \}$

$n_7$: $\{ H, t = 0 \} \leftrightarrow \{ \neg H \}$
Hierarchical Execution

Example (Delivery)

State $s_0$
\{\neg G, \neg H, u = 1, p = 0, t = 1\}

State $s_1$
\{\neg G, H, u = 1, p = \infty, t = 1\}

State $s_2$
\{\neg G, H, u = 1, p = \infty, t = 0\}

State $s_3$
\{G, \neg H, u = 0, p = \infty, t = 0\}

$n_0$: \{\neg G\} \mapsto \{G\}

$n_1$: \{u > 0\} \mapsto \{u\downarrow\}

$n_2$: \{\neg H, u > 0\} \mapsto \{H\}

$n_3$: \{H, u > 0\} \mapsto \{\neg H, u\downarrow\}

$n_4$: \{\neg H, p > 0\} \mapsto \{p\downarrow\}

$n_5$: \{\neg H, p = 0\} \mapsto \{H, p?\}

$n_6$: \{H, t > 0\} \mapsto \{t\downarrow\}

$n_7$: \{H, t = 0\} \mapsto \{\neg H\}
## Experiments: Planning

<table>
<thead>
<tr>
<th>Domain</th>
<th>Coverage</th>
<th>Time (sec)</th>
<th>Coverage</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blocks-clear (30)</td>
<td>30</td>
<td>32</td>
<td>30</td>
<td>29</td>
</tr>
<tr>
<td>Blocks-on (30)</td>
<td>30</td>
<td>23</td>
<td>30</td>
<td>23</td>
</tr>
<tr>
<td>Delivery (30)</td>
<td>4</td>
<td>999</td>
<td>30</td>
<td>22</td>
</tr>
<tr>
<td>Gripper (30)</td>
<td>30</td>
<td>2</td>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>Miconic (30)</td>
<td>30</td>
<td>7</td>
<td>30</td>
<td>7</td>
</tr>
<tr>
<td>Reward (30)</td>
<td>30</td>
<td>381</td>
<td>30</td>
<td>39</td>
</tr>
<tr>
<td>Spanner (30)</td>
<td>0</td>
<td>–</td>
<td>30</td>
<td>11</td>
</tr>
<tr>
<td>Visitall (30)</td>
<td>29</td>
<td>189</td>
<td>30</td>
<td>783</td>
</tr>
</tbody>
</table>

# Solved domains 5 8

**Table 1:** Satisficing planning with resource limits 8 GB memory and 30 minutes time.
Summary

- Hierarchical policies are important in planning and RL
- There are no principled methods in generalized planning for learning them
- New width-based formulation: hierarchical policy is a tree with sketch rule $r(n)$ and classes of subproblems $Q(n)$ for each node $n$ where
  - $Q(\text{root}) = Q_{\text{target}}$
  - $\text{width}(Q(n)) < \text{width}(Q(\text{parent}(n)))$
  - $\text{width}(Q(\text{leaf})) = 0$
- Method for learning hierarchical policies with no supervision from small instances
  - Based on ASP/Clingo
  - Uses pool of $C_3$ features
  - Interesting hierarchical policies obtained for number of benchmarks
