Hybrid Planning with Preferences Using a Heuristic for Partially Ordered Plans

Pascal Bercher and Susanne Biundo

Institute of Artificial Intelligence

October 4th, 2011
Hybrid Planning with Preferences

Note

- The paper is about solving *hybrid* planning problems with preferences, where **hybrid planning is POCL planning plus hierarchical planning**.
- For the sake of simplicity, this talk focuses on the aspect of POCL planning.

(POCL planning $\equiv$ Partial Order Causal Link Planning)
Problem Representation

Let \( F \) be a set of ground facts. Then, a (classical) planning problem is a tuple \( \mathcal{P} = \langle s_{\text{init}}, s_{\text{goal}}, A \rangle \), where

- \( s_{\text{init}} \in 2^F \) is the initial state,
- \( s_{\text{goal}} \subseteq F \) is the goal description, and
- \( A \) is a set of actions, where each \( a \in A \) has the form \( \langle \text{pre}, \text{add}, \text{del} \rangle \). We call \( \text{pre} \subseteq F \) the precondition, \( \text{add} \subseteq F \) the add list, and \( \text{del} \subseteq F \) the delete list of \( a \).

- An action \( a \) is applicable in a state \( s \in 2^F \) iff \( \text{pre}(a) \subseteq s \). If applied, it leads to \( a(s) = (s \setminus \text{del}(a)) \cup \text{add}(a) \).
- The applicability of an action sequence \( \bar{a} \) is defined straightforward.

An action sequence \( \bar{a} \) is called a solution to \( \mathcal{P} \) if the final state it produces, \( \bar{a}(s_{\text{init}}) \), satisfies the goal description, i.e., \( \bar{a}(s_{\text{init}}) \supseteq s_{\text{goal}} \).
Motivation of Preferences

- It is often too restrictive to have just a goal description that \textit{must} be satisfied. (This might even be impossible!)
- One would rather like to give preferences of the desired goal state thus defining a quality measure on solutions.

Example

- Goal description: $\text{delivered}(p_1) \land \cdots \land \text{delivered}(p_n)$
- Preference 1: $\text{at}(\text{truck}_1, \text{loc}_1)$, value: 5
- Preference 2: $\text{at}(\text{truck}_2, \text{loc}_1)$, value: 15
- Preference 3: $\text{at}(\text{truck}_2, \text{loc}_2)$, value: 5
- Preference 4: $\text{at}(\text{truck}_3, \text{loc}_1)$, value: 10
- Preference 5: $\text{at}(\text{truck}_3, \text{loc}_2)$, value: 5
Planning with Preferences — Definition

Let \( \mathcal{P} = \langle \text{s}_{\text{init}}, \text{s}_{\text{goal}}, \mathcal{A} \rangle \) be a planning problem. Then, \( \mathcal{P}_{\text{ref}} \subseteq \mathcal{F} \times \mathbb{N} \) is a set of weighted preferences.

Let \( \bar{a} \) be a solution to \( \mathcal{P} \).
\( \bar{a} \) satisfies a preference \((f, n) \in \mathcal{P}_{\text{ref}} \) if and only if \( f \in \bar{a}(\text{s}_{\text{init}}) \).

Now, we can define the quality of a solution:

\[
m(\bar{a}) := \sum_{(f, n) \in \mathcal{P}_{\text{ref}}; f \in \bar{a}(\text{s}_{\text{init}})} n
\]
POCL Planning

How to find good solutions?

- We perform search in the space of plans. More precisely: POCL planning.
- We propose a POCL branch-and-bound algorithm and an admissible heuristic to prune plans which will lead to suboptimal solutions.

What is a plan?

A partially ordered (partial) plan is a tuple $P = \langle PS, \prec \rangle$, where

- $PS$ is a set of plan steps from $A$, and
- $\prec \subseteq PS \times PS$ defines the partial order.
Algorithm 1: Plan Space-Based Branch-and-Bound Algorithm

1. $\text{best-quality} := -\infty$
2. $\text{best-solution} := \text{fail}$
3. $\text{Fringe} := \{\langle, \rangle\}$, where $\langle, \rangle$ is the initial plan
4. while Fringe not empty do
   5. $P := \text{remove-best}(\text{Fringe})$
   6. if $P$ is a solution and $m(P) > \text{best-quality}$ then
      7. $\text{best-quality} := m(P)$
      8. $\text{best-solution} := P$
   9. if $h(P) \geq \text{best-quality}$ then
      10. Fringe := Fringe $\cup \{P' \mid P' \text{ is a refinement of } P\}$
11. return $\text{best-solution}$
We want to discard suboptimal solutions:
\[
\text{if } h(P) \geq \text{best-quality then refine further}
\]

Thus, we have to estimate the quality of the partial plan $P$, $m(P)$. To that end, we estimate which of the preferences might hold in the final state produced, given a current partial plan $P$. 
Heuristic — Idea I (What to do)

Basic idea

- Perform a reachability analysis for all preferences. The result is a set of mutex relations: if \((f_1, f_2) \in \mathcal{F} \times \mathcal{F}\) is a detected mutex, then \(f_1\) and \(f_2\) cannot be true at the same time.
- Calculate estimate of \(m\) based on the detected mutexes.
Basic idea

- Find the mutex relations via construction of a planning graph.
  - We want to find the mutexes for different partial plans.
  - Planning graph construction is done once for a planning problem $\mathcal{P}$; i.e. one cannot construct a planning graph for a given partial plan $P$.
    $\rightarrow$ Solution: Encode $P$ within $\mathcal{P}'$!
- Calculate estimate of $m$ by calculating a minimal vertex cover.
Let $\mathcal{P} = \langle s_{\text{init}}, s_{\text{goal}}, A \rangle$ be a planning problem and $P = \langle PS, \prec \rangle$ the current partial plan under consideration.

We want to find a planning problem $\mathcal{P}'$, s.t.

- $\mathcal{P}'$ has a solution if $\mathcal{P}$ has one,
- every solution to $\mathcal{P}'$ is a refinement of $P$.

How to encode it?

- Create additional actions $A'$ for each action $a \in PS$.
- Augment each action $a' \in A'$ with a predicate $occ-a'$.
- Alter the goal description accordingly.
Let \( P = \langle s_{\text{init}}, s_{\text{goal}}, A \rangle \) be a planning problem and \( P = \langle PS, \prec \rangle \) the current partial plan under consideration.

\[
\begin{align*}
a_1 &= \text{drive}(\text{truck}_3, \text{loc}_1, \text{loc}_2) \\
&< \\
a_2 &= \text{deliver}(p_1, \text{loc}_2)
\end{align*}
\]

Then, the problem \( P' \) encoding \( P \) is given by \( \langle s'_{\text{init}}, s'_{\text{goal}}, A' \rangle \) with:

- \( s'_{\text{init}} := s_{\text{init}} \),
- \( A' := \text{delete-relax}(A) \cup \{ \text{enc}(a_1), \text{enc}(a_2) \} \) with
  - \( \text{enc}(a_1) = \langle \text{pre}(a_1) \land \neg \text{occ-a}_1, \text{effects}(a_1) \land \text{occ-a}_1 \rangle \),
  - \( \text{enc}(a_2) = \langle \text{pre}(a_2) \land \neg \text{occ-a}_2 \land \text{occ-a}_1, \text{effects}(a_2) \land \text{occ-a}_2 \rangle \),
- \( s'_{\text{goal}} := s_{\text{goal}} \land \text{occ-a}_1 \land \text{occ-a}_2 \),
Heuristic — Problem Transformation (Example)

Let $\mathcal{P} = \langle s_{\text{init}}, s_{\text{goal}}, A \rangle$ be a planning problem and $P = \langle PS, \prec \rangle$ the current partial plan under consideration.

\[
\begin{align*}
\text{a}_1 &= \text{drive} (\text{truck}_3, \text{loc}_1, \text{loc}_2) < \text{a}_2 = \text{deliver} (p_1, \text{loc}_2)
\end{align*}
\]

Then, the problem $\mathcal{P}'$ encoding $P$ is given by $\langle s'_{\text{init}}, s'_{\text{goal}}, A' \rangle$ with:

- $s'_{\text{init}} := s_{\text{init}}$,
- $A' := \text{delete-relax}(A) \cup \{ \text{enc}(a_1), \text{enc}(a_2) \}$ with
  - $\text{enc}(a_1) = \langle \text{pre}(a_1) \land \neg \text{occ-}a_1, \text{effects}(a_1) \land \text{occ-}a_1 \rangle$,
  - $\text{enc}(a_2) = \langle \text{pre}(a_2) \land \neg \text{occ-}a_2 \land \text{occ-}a_1, \text{effects}(a_2) \land \text{occ-}a_2 \rangle$,
- $s'_{\text{goal}} := s_{\text{goal}} \land \text{occ-}a_1 \land \text{occ-}a_2$,
Heuristic — Problem Transformation (Example)

Let \( P = \langle s_{\text{init}}, s_{\text{goal}}, A \rangle \) be a planning problem and \( P = \langle PS, \prec \rangle \) the current partial plan under consideration.

\[
\begin{align*}
\text{a}_1 &= \text{drive}(\text{truck}_3, \text{loc}_1, \text{loc}_2) \\ \\
\text{a}_2 &= \text{deliver}(p_1, \text{loc}_2)
\end{align*}
\]

Then, the problem \( P' \) encoding \( P \) is given by \( \langle s'_{\text{init}}, s'_{\text{goal}}, A' \rangle \) with:

- \( s'_{\text{init}} := s_{\text{init}} \),
- \( A' := \text{delete-relax}(A) \cup \{ \text{enc}(a_1), \text{enc}(a_2) \} \) with
  - \( \text{enc}(a_1) = \langle \text{pre}(a_1) \land \neg \text{occ-a}_1, \text{effects}(a_1) \land \text{occ-a}_1 \rangle \),
  - \( \text{enc}(a_2) = \langle \text{pre}(a_2) \land \neg \text{occ-a}_2 \land \text{occ-a}_1, \text{effects}(a_2) \land \text{occ-a}_2 \rangle \),
- \( s'_{\text{goal}} := s_{\text{goal}} \land \text{occ-a}_1 \land \text{occ-a}_2 \),
Heuristic — Problem Transformation (Example)

Let $P = \langle s_{init}, s_{goal}, A \rangle$ be a planning problem and $P = \langle PS, \prec \rangle$ the current partial plan under consideration.

Then, the problem $P'$ encoding $P$ is given by $\langle s'_{init}, s'_{goal}, A' \rangle$ with:

- $s'_{init} := s_{init}$,
- $A' := delete-relax(A) \cup \{enc(a_1), enc(a_2)\}$ with
  - $enc(a_1) = \langle pre(a_1) \land \neg occ-a_1, effects(a_1) \land occ-a_1 \rangle$,
  - $enc(a_2) = \langle pre(a_2) \land \neg occ-a_2 \land occ-a_1, effects(a_2) \land occ-a_2 \rangle$,
- $s'_{goal} := s_{goal} \land occ-a_1 \land occ-a_2$, 

\[
\begin{array}{c}
  a_1 = \\
  \text{drive(}truck_3, loc_1, loc_2) \\
  \prec \\
  \text{deliver(}p_1, loc_2)
\end{array}
\]
Calculate an admissible estimate based on the mutex relations.

Example:
Calculate an admissible estimate based on the mutex relations.

Example:
Calculate an admissible estimate based on the mutex relations.

Heuristic Calculation:

Let $b : \mathcal{F} \to \{\top, \bot\}$ be a truth assignment that is consistent with the mutex relations of $\mathcal{F}$. Then,

$$heuristic(P) = heuristic(s'_{init}) := \max_b \left( \sum_{(f,n) \in \mathcal{P}ref, \ b(f) = \top} n \right)$$

where $s'_{init}$ is the initial state of the transformed problem $\mathcal{P}'$. 

Summary

- We introduced the first heuristic for soft goals for POCL planning, which
  - applies to classical planning problems and
  - to hybrid planning problems.

- Approach can easily be extended to handle arbitrary formulas over soft goals.

- Note: basic idea behind this heuristic can be adapted, s.t. any POCL planner can use (almost) any heuristic from classical planning.
Appendix — Problem Transformation (Formally)

Let $\mathcal{P} = \langle s_{\text{init}}, s_{\text{goal}}, A \rangle$ be a planning problem and $P = \langle PS, \prec \rangle$ the current partial plan under consideration.

Let $\mathcal{P}' = \langle s'_{\text{init}}, s'_{\text{goal}}, A' \rangle$ be the transformed problem, s.t.

- $s'_{\text{init}} := s_{\text{init}} \cup \{ \text{not-occ-}a_i \mid a_i \in PS \}$
- $A' := A_1 \cup A_2$ and
  - $A_1 := \text{delete-relax}(A) = \{ \langle \text{pre}, \text{add}, \emptyset \rangle \mid \langle \text{pre}, \text{add}, \text{del} \rangle \in A \}$
  - $A_2 := \text{encode}(P) = \{ \text{enc}(a_i) \mid a_i \in PS, a_i = \langle \text{pre}, \text{add}, \text{del} \rangle \}$, and $\text{enc}(a_i) := \langle \text{pre}', \text{add}', \text{del}' \rangle$ with
    - $\text{pre}' := \{ \text{not-occ-}a_i \} \cup \text{pre} \cup \{ \text{occ-}a_j \mid (a_j, a_i) \in \prec \}$
    - $\text{add}' := \{ \text{occ-}a_i \} \cup \text{add}$
    - $\text{del}' := \{ \text{not-occ-}a_i \} \cup \text{del}$
- $s'_{\text{goal}} := s_{\text{goal}} \cup \{ \text{occ-}a_i \mid a_i \in PS \}$
Appendix — Example: Planning Graph

\[
\begin{align*}
\text{fact layer}_i & \quad \text{action layer}_i & \quad \text{fact layer}_{i+1} \\
\vdots & & \vdots \\
\text{at}(\text{truck}_1, \text{loc}_1) & \quad \text{drive}(\text{truck}_3, \text{loc}_1, \text{loc}_2) & \quad \text{at}(\text{truck}_1, \text{loc}_1) \\
\text{at}(\text{truck}_2, \text{loc}_1) & & \quad \text{at}(\text{truck}_2, \text{loc}_1) \\
\text{at}(\text{truck}_3, \text{loc}_1) & & \quad \text{at}(\text{truck}_3, \text{loc}_1) \\
\vdots & & \vdots \\
\text{pre} & & \text{add} \\
\text{del} & & \text{mutex}
\end{align*}
\]