Hybrid Planning with Preferences Using a Heuristic for Partially Ordered Plans

Pascal Bercher and Susanne Biundo

Institute of Artificial Intelligence

October 4th, 2011

ulm university universität uulm

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 == Partial Order Causal Link Planning)

Problem Representation

Let $\mathcal{F}$ 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

- $s_{\text{init}} \in 2^{\mathcal{F}}$ is the initial state,
- $s_{\text{goal}} \subseteq \mathcal{F}$ is the goal description, and
- A is a set of actions, where each $a \in A$ has the form $\langle pre, add, del \rangle$. We call $pre \subseteq \mathcal{F}$ the precondition, $add \subseteq \mathcal{F}$ the add list, and $del \subseteq \mathcal{F}$ the delete list of a.
  - An action a is applicable in a state $s \in 2^{\mathcal{F}}$ iff $pre(a) \subseteq s$. If applied, it leads to $a(s) = (s \setminus del(a)) \cup add(a)$.
  - The applicability of an action sequence $\bar{a}$ is defined straight forward.

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., $\bar{a}(s_{\text{init}}) \supseteq s_{\text{goal}}$.

Planning with Preferences — Motivation

Motivation of Preferences

- It is often too restrictive to have just a goal description that *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: $\mathit{delivered}(p_1) \land \cdots \land \mathit{delivered}(p_n)$
- Preference 1: $\mathit{at}(\mathit{truck}_1, \mathit{loc}_1)$, value: 5
- Preference 2: $\mathit{at}(\mathit{truck}_2, \mathit{loc}_1)$, value: 15
- Preference 3: $\mathit{at}(\mathit{truck}_2, \mathit{loc}_2)$, value: 5
- Preference 4: $\mathit{at}(\mathit{truck}_3, \mathit{loc}_1)$, value: 10
- Preference 5: $\mathit{at}(\mathit{truck}_3, \mathit{loc}_2)$, value: 5

Planning with Preferences — Definition

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

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

Now, we can define the quality of a solution:

$$m(\bar{a}) := \sum_{\substack{(f,n) \in \mathcal{P}ref \\ f \in \bar{a}(s_{\text{init}})}} n$$

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.

Search algorithm

Algorithm 1: Plan Space-Based Branch-and-Bound Algorithm

`best-quality` $:= -\infty$
`best-solution` $:=$ *fail*
`Fringe` $:= \{\langle,\rangle\}$, where $\langle,\rangle$ is the initial plan
while *Fringe not empty* **do**
    $P :=$ remove-best(`Fringe`)
    if *P is a solution* **and** $m(P) >$ *best-quality* **then**
        `best-quality` $:= m(P)$
        `best-solution` $:= P$
    if $h(P) \geq$ *best-quality* **then**
        `Fringe` $:=$ `Fringe` $\cup \{P' \mid P'$ *is a refinement of* $P\}$
return *best-solution*

Discarding Suboptimal Solutions

We want to discard suboptimal solutions:
if $h(P) \geq \texttt{best-quality}$ **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.

Heuristic — Idea II (How to do it)

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.
    $\longrightarrow$ Solution: Encode P within $\mathcal{P}'$!
- Calculate estimate of m by calculating a minimal vertex cover.

Heuristic — Problem Transformation (Idea)

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.

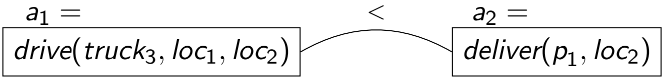
- $\mathcal{P}'$ has a solution if $\mathcal{P}$ has one,
- every solution to $\mathcal{P}'$ is a refinement of P.

How to encode it?

- Create additional actions A' for each action $a \in PS$.
- Augment each action $a' \in A'$ with a predicate $occ\text{-}a'$.
- Alter the goal description accordingly.

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.



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

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

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.

$$a_1 = \boxed{drive(truck_3, loc_1, loc_2)} \quad < \quad a_2 = \boxed{deliver(p_1, loc_2)}$$

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

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

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.

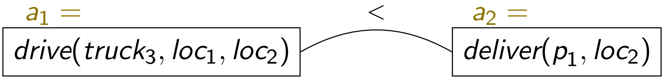
$a_1 =$ $drive(truck_3, loc_1, loc_2)$ $<$ $a_2 =$ $deliver(p_1, loc_2)$

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

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

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.



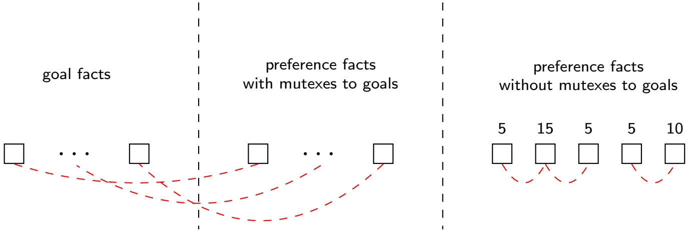
Then, the problem $\mathcal{P}'$ encoding P is given by $\langle s'_{\text{init}}, s'_{\text{goal}}, A' \rangle$ with:

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

Heuristic — Calculation (Example)

Calculate an admissible estimate based on the mutex relations.

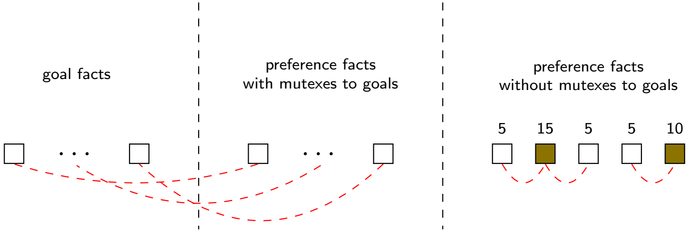
Example:



Heuristic — Calculation (Example)

Calculate an admissible estimate based on the mutex relations.

Example:



Heuristic — Calculation (Formally)

Calculate an admissible estimate based on the mutex relations.

Heuristic Calculation:

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

$$heuristic(P) = heuristic(s'_{\text{init}}) := \max_{b} \Big(\sum_{\substack{(f,n) \in \mathcal{P}ref, \\ b(f) = \top}} n \Big)$$

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.

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

Appendix — Example: Planning Graph