On Domain-Independent Heuristics for Planning with Qualitative Preferences

Jorge Baier   Sheila McIlraith

January 8th, 2007

Motivation

Classical Planning

- Plan must satisfy (final-state) goals.

Planning with Qualitative Temporally Extended Preferences (QTEPs)

- Qualitative language to specify *preferred* plans.
  E.g., Plans such that: **eventually**(*eat*(*tandooriChicken*))
  are preferred to
  those such that: **eventually**(*eat*(*spaghetti*)).
- Language allows temporally extended properties.
- We want a **most-preferred plan for the goal**.

Motivation

Classical Planning

- Plan must satisfy (final-state) goals.

➦ Fastest state-of-the-art planners use lookahead heuristics.

Planning with Qualitative Temporally Extended Preferences (QTEPs)

- Qualitative language to specify *preferred* plans.
  E.g., Plans such that: **eventually**(*eat*(*tandooriChicken*))
  are preferred to
  those such that: **eventually**(*eat*(*spaghetti*)).
- Language allows temporally extended properties.
- We want a **most-preferred plan for the goal**.

Motivation

Classical Planning

- Plan must satisfy (final-state) goals.

➦ Fastest state-of-the-art planners use lookahead heuristics.

Planning with Qualitative Temporally Extended Preferences (QTEPs)

- Qualitative language to specify *preferred* plans.
  E.g., Plans such that: **eventually**(*eat*(*tandooriChicken*))
  are preferred to
  those such that: **eventually**(*eat*(*spaghetti*)).
- Language allows temporally extended properties.
- We want a **most-preferred plan for the goal**.

➦ Current planners for qualitative preferences don't use lookahead heuristics.

We propose a **heuristic planner for QTEPs**

Outline of the talk

- Background
  - LPP and planning
- Problem Simplification
- Heuristics for QTEP planning
- Algorithm
- Implementation of HPlan-QP & Experimental Results
- Conclusions

Outline of the talk

- Background
  - LPP and planning
- Problem Simplification
- Heuristics for QTEP planning
- Algorithm
- Implementation of HPlan-QP & Experimental Results
- Conclusions

LPP: a language for preferences

We consider LPP [Bienvenu *et al.*, 2006], a rich language for temporally extended preferences.

Main element APFs:

Atomic Preference Formulae (APFs)

Used to express preferences over alternative properties. Form:

$$\varphi_0[v_0] \gg \varphi_1[v_1] \gg \ldots \gg \varphi_n[v_n],$$

where $v_1 < v_2 < \cdots < v_n \in \mathcal{V}$, and $\mathcal{V}$ is a totally **ordered qualitative** finite **set**, and φ_i is an formulae of a **linear temporal logic** (LTL).

Examples of APFs

Let $\mathcal{V} = \{best, great, good, ok, bad\}$. Examples of APFs:

$$\begin{aligned} P_{food} \stackrel{\text{def}}{=} & \textbf{eventually}(\textbf{occ}(eat(pizza)))[best] \gg \\ & \textbf{eventually}(\textbf{occ}(eat(spag)))[great] \gg \\ & \textbf{eventually}(\textbf{occ}(eat(crêpes)))[good] \gg \\ & \textbf{eventually}(\textbf{occ}(eat(taoChicken)))[ok] \end{aligned}$$

$$\begin{aligned} P_{home} \stackrel{\text{def}}{=} & \textbf{always}(at(home)) \wedge \forall x\, \neg\textbf{eventually}(\textbf{occ}(cook(x)))[best] \gg \\ & \textbf{always}(at(home)) \wedge \exists x\, \textbf{eventually}(\textbf{occ}(cook(x)))[good] \end{aligned}$$

Preference aggregation LPP

LPP allows combining preferences through *general preference formulae* (GPFs).
If γ is an LTL formula, and Ψ_1 and Ψ_2 are APFs:

GPF	**Informal semantics**
$\gamma : \Psi_1,$	If γ holds in the plan, preferences given by Ψ_1
$\Psi_1 \,\&\, \Psi_2$	Prefer to satisfy both Ψ_1 and Ψ_2
$\Psi_1 \mid \Psi_2$	Indifferent between Ψ_1 and Ψ_2

Examples:

$$P_{home} \,\&\, P_{food} \qquad IsSnowing : P_{home}$$

Semantics of LPP

The semantics of LPP are defined in the situation calculus [Bienvenu *et al.*, 2006].

The w function is such that if s_1 and s_2 are situations and Ψ is an GPF,

$$w_{s_1}(\Psi) < w_{s_2}(\Psi) \text{ iff } s_1 \text{ is } \textit{preferred to } s_2 \text{ with respect to } \Psi.$$

Planning and Planning with Preferences

Definition ((Classical) Planning)

Given a Situation Calculus theory of action $\mathcal{D}$ and a goal formula G, find a situation S such that:

$$D \models G(S)$$

Definition (Preference-Based Planning)

Given a Situation Calculus theory of action $\mathcal{D}$, a goal formula G, and a GPF Ψ find an S such that:

$$D \models G(S) \wedge \neg \exists s' \, [G(s') \wedge w_{s'}(\Psi) < w_S(\Psi)]$$

State of the art

Best Classical Planners:

- Use some form search
- **Guided** by **heuristics** measuring **progress** towards achieving the goal.

Planners for QTEPs:

- Use some form search
- Guiding **not based on progress** towards achieving preferences

State of the art

Best Classical Planners:

- Use some form search
- **Guided** by **heuristics** measuring **progress** towards achieving the goal.

Planners for QTEPs:

- Use some form search
- Guiding **not based on progress** towards achieving preferences

Our goal: apply heuristics for efficient QTEP planning

The Challenge

Efficient classical planners use **heuristics**.

- Designed for single goals
- Designed for final-state goals

In planning with LPP preferences:

- GPFs composed by several properties, interacting in complex ways.
- Properties are temporal.

We need to solve **two problems**:

- **Identify the properties** that characterize preferred plans
- Guide search with a **single heuristic function**

Outline of the talk

- Background
  - LPP and planning
- Problem Simplification
- Heuristics for QTEP planning
- Algorithm
- Implementation of HPLAN-QP & Experimental Results
- Conclusions

First step: simplification

We simplify the planning problem, generating a new one such that:

- All GPFs are replaced by APFs $\Rightarrow$ **reduce interaction** among BDFs.
- **Replace temporal** prefs. by equivalent **non-temporal** prefs.

Simplifying GPFs

We prove that:

Theorem

Let Ψ be an arbitrary GPF over the set of preference values $\mathcal{V}$, then it is possible to construct an **equivalent APF** *ϕ_Ψ, over $\mathcal{V}$.*

This means that all our preferences look like:

$$\varphi_0[v_0] \gg \varphi_1[v_1] \gg \ldots \gg \varphi_n[v_n],$$

However, still the φ_i's is temporal.

Simplifying temporal formulae

In previous work [Baier and McIlraith, 2006], we proved that:

Theorem

Let P be a planning problem, and φ be a first-order LTL formulae. P can be extended with a new additional predicate, Sat_φ, that is true in the **final sate** *iff φ_i is true.*

This means that now our preferences now look like:

$$\varphi_0[v_0] \gg \varphi_1[v_1] \gg \ldots \gg \varphi_n[v_n],$$

Where the φ_i's are all **non**-temporal.

Outline of the talk

- Background
  - LPP and planning
- Problem Simplification
- Heuristics for QTEP planning
- Algorithm
- Implementation of HPlan-QP & Experimental Results
- Conclusions

Heuristic functions for guiding search

We always want to **achieve** *our goal*

Heuristic functions for guiding search

We always want to **achieve** *our goal*

Goal distance (G)

A distance-to-the-goals function computed from the expanded **relaxed graph**. In our implementation, is the additive heuristic by [Bonet and Geffner, 2001] adapted for ADL operators.

Heuristic functions for guiding search

We always want to **achieve** *our goal*

Goal distance (G)

A distance-to-the-goals function computed from the expanded **relaxed graph**. In our implementation, is the additive heuristic by [Bonet and Geffner, 2001] adapted for ADL operators.

Guide search **towards** *preferred properties*

Heuristic functions for guiding search

We always want to **achieve** *our goal*

Goal distance (G)

A distance-to-the-goals function computed from the expanded **relaxed graph**. In our implementation, is the additive heuristic by [Bonet and Geffner, 2001] adapted for ADL operators.

Guide search **towards** *preferred properties*

Preference distance function (**P**)

A distance-to-the-preferences function computed from the expanded **relaxed graph**. If the preference is

$$\varphi_0[v_0] \gg \varphi_1[v_1] \gg ... \gg \varphi_n[v_n],$$

Then $\mathbf{P} = (p_0, \ldots, p_n)$, where p_i is estimates how hard it is to achieve φ_i.

Heuristics for pruning

if found plan with weight W, **don't extend** *plans that* **won't reach a better weight**

Best Relaxed Metric (B)

- An *estimation* of the best metric weight that plan that traverses the current state can achieve
- Corresponds to the best weight in the relaxed worlds.

Putting pieces together: adding the goal

Still unanswered: **When is s_1 better than s_2?**

- Let G_1 and G_2 be the value of the goal distance function for s_1 and s_2.

Strategy	**Check whether**	**If tied, check whether**
goal-value	$G_1 < G_2$	Is s_1's best weighted preferred property easier than that of s_2?
goal-easy	$G_1 < G_2$	Is s_1's easiest preferred property easier than that of s_2?

value-goal and **easy-goal** do the tests in reverse order.

Outline of the talk

- Background
  - LPP and planning
- Problem Simplification
- Heuristics for QTEP planning
- Algorithm
- Implementation of HPLAN-QP & Experimental Results
- Conclusions

The Algorithm

Input: goal; APF preference; a bound for the plan *k*
Output: sequence plans for goal with incrementally better weight

Perform **best-first** search, where:

- States are ordered using one of the strategies proposed.

The Algorithm

Input: goal; APF preference; a bound for the plan k
Output: sequence plans for goal with incrementally better weight

Perform **best-first** search, where:

- States are ordered using one of the strategies proposed.
- If best plan found has weight W, then prune states whose B function value is worse than W.
- Prune plans whose length exceed k.
- Output a plan when its weight is the best found so far.
- Execute until the search space is empty.

This is a **heuristic**, **incremental** planner for QTEPs.

Properties

Definition (*k*-optimal)

A planning algorithm is *k*-optimal, if it eventually returns the best-weighted plan among all those of length bounded by k.

Theorem

Our proposed algorithm is k-optimal.

Observation

This theorem does not mean that the *first* plan that is output

Implementation of HPLAN-QP

- Preprocessor:
  - Parses a domain with atomic preferences (in an extended PDDL3!)
  - Performs the temporal simplification.
  - Generates TLPlan files.
- Modified TLPlan:
  - Compute heuristic estimates using relaxed graphs
  - Handle efficiently the automata updates.

Experimental Evaluation & Comparison

- We compared our planner to the PPLAN planner [Bienvenu *et al.*, 2006].
- Characteristics of PPLAN:
  - Best-first search, admissible heuristics.
  - *k*-optimal; *first* plan is optimal.
  - Not optimized for speed

Experimental Evaluation & Comparison

- We compared our planner to the PPLAN planner [Bienvenu *et al.*, 2006].
- Characteristics of PPLAN:
  - Best-first search, admissible heuristics.
  - *k*-optimal; *first* plan is optimal.
  - Not optimized for speed
- Examples performed over a *dinner* domain.

Summary of Results, dinner domain

Table: Number of **expanded nodes**.

Prob#	PPLAN	goal-easy	goal-value	easy-goal	value-goal
1	7	3	3	3	3
7	29	34	20	27	8
8	42	12	12	4	4
9	55	13	13	4	4
11*	57	107	45	102	5
12	92	33	33	6	6
13	171	11617	11617	24	24
14	194	4	4	4	4
16	313	58	58	8	8
17	13787	12	12	7562	7
19*	>20000	3	3	3	3
21	>20000	71	71	8	8
22*	>20000	85	30	7	145
23*	>20000	4	4	4	6
24*	>20000	49	22	7	8

*: Best value BDF preference cannot be achieved

Summary & Conclusions

- We have proposed a **heuristic** algorithm for **QTEPs**
- Key enablers:
  - Simplification of preference formulae.
  - Transformation of temporal preferences into non-temporal ones.
- We have **implemented** this algorithm TLPlan.
- The algorithm shows **better performance** than existing planners.

Related Work

Languages and planners for QTEP

- [Delgrande *et al.*, 2004]: Temporally extended preference language
- [Son and Pontelli, 2004]: Using Answer Set Programming.
- [Bienvenu *et al.*, 2006]: Optimal Best-First Planning

2006 Planning Competition (Quantitative)

- Final-state preferences: *Yochan*$^{\mathcal{PS}}$ [Benton *et al.*, 2006].
- Temporally extended preferences: SGPlan$_5$ [Hsu *et al.*, 2007], MIPS-XXL [Edelkamp, 2006], MIPS-BDD [Edelkamp *et al.*, 2006], HPlan-P [Baier *et al.*, 2007].

Putting pieces together

Let P_1 and P_2 be the preference vectors of states s_1 and s_2.

When is s_1 *better than* s_2?

Putting pieces together

Let P_1 and P_2 be the preference vectors of states s_1 and s_2.

When is s_1 *better than* s_2?

Regarding preferences, we have defined **two criteria**:

$\mathbf{P}_1 <_{\text{VALUE}} \mathbf{P}_2$ Best-weighted BDF preference of $\mathbf{P}_1$ estimated easier than that of $\mathbf{P}_2$.

$\mathbf{P}_1 <_{\text{EASY}} \mathbf{P}_2$ means that either $\mathbf{P}_1$ contains a preference formula that has been estimated to be easier than all those in $\mathbf{P}_2$.

Putting pieces together: adding the goal

When is s_1 *better than* s_2?

Now we **consider the goal**:

- Let G_1 and G_2 be the value of the goal distance function for s_1 and s_2.
- The following strategies guide search towards the preferences *and* the goal.

Strategy	Check whether	If tied, check whether
goal-value	$G_1 < G_2$	$\mathbf{P}_1 <_{\text{VALUE}} \mathbf{P}_2$
goal-easy	$G_1 < G_2$	$\mathbf{P}_1 <_{\text{EASY}} \mathbf{P}_2$
value-goal	$\mathbf{P}_1 <_{\text{VALUE}} \mathbf{P}_2$	$G_1 < G_2$
easy-goal	$\mathbf{P}_1 <_{\text{EASY}} \mathbf{P}_2$	$G_1 < G_2$

References I

Jorge A. Baier and Sheila A. McIlraith.

Planning with first-order temporally extended goals using heuristic search.

In *Proc. of the 21st National Conference on Artificial Intelligence (AAAI-06)*, pages 788–795, Boston, MA, 2006.

J. Baier, F. Bacchus, and S. McIlraith.

A heuristic search approach to planning with temporally extended preferences.

In *Proc. of the 20th Int'l Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, January 2007.

To appear.

J. Benton, Subbarao Kambhampati, and Minh B. Do.

YochanPS: PDDL3 simple preferences and partial satisfaction planning.

In *5th International Planning Competition Booklet (IPC-2006)*, pages 54–57, Lake District, England, July 2006.

References II

Meghyn Bienvenu, Christian Fritz, and Sheila McIlraith.
Planning with qualitative temporal preferences.
In *Proc. of the 10th Int'l Conference on Knowledge Representation and Reasoning (KR-06)*, pages 134–144, Lake District, England, 2006.

Blai Bonet and Hector Geffner.
Planning as heuristic search.
Artificial Intelligence, 129(1-2):5–33, 2001.

James P. Delgrande, Torsten Schaub, and Hans Tompits.
Domain-specific preferences for causal reasoning and planning.
In *Proc. of the 14th Int'l Conference on Automated Planning and Scheduling (ICAPS-04)*, pages 63–72, Whistler, Canada, June 2004.

References III

Stefan Edelkamp, Shahid Jabbar, and Mohammed Naizih.
Large-scale optimal PDDL3 planning with MIPS-XXL.
In *5th International Planning Competition Booklet (IPC-2006)*, pages 28–30, Lake District, England, July 2006.

Stefan Edelkamp.
Optimal symbolic PDDL3 planning with MIPS-BDD.
In *5th International Planning Competition Booklet (IPC-2006)*, pages 31–33, Lake District, England, July 2006.

Chih-Wei Hsu, Benjamin Wah, Ruoyun Huang, and Yixin Chen.
Constraint partitioning for solving planning problems with trajectory constraints and goal preferences.
In *Proc. of the 20th Int'l Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, January 2007.
To appear.

References IV

Tran Cao Son and Enrico Pontelli.
Planning with preferences using logic programming.
In V. Lifschitz and I. Niemela, editors, *Proc. of the 7th Int'l Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-04)*, number 2923 in LNCS, pages 247–260. Springer, 2004.