Solving Goal Utility Dependencies* and Simple Preferences in Partial Satisfaction Planning

J. Benton
Computer Sci. & Eng. Dept.
Arizona State University
Tempe, AZ 85287
j.benton@asu.edu

Abstract

This doctoral research focuses on the analysis and development of techniques for solving domain-independent partial satisfaction planning (PSP) problems and planning problems with preferences. Recently, these areas have gained the attention of the planning community. This has been underscored by the recent introduction of preferences to the Fifth International Planning Competition (IPC5). This extended abstract outlines contributions made to the area of PSP and shows that planning with "simple preferences", as defined in the IPC5, is can be compiled to PSP. Afterwards, it outlines future steps to be taken for advancing this line of research.

Introduction

In many real world problems, users prefer some goals over others. In this sense, they have preferences among specified "soft" goals. For instance, a user may prefer brown flour over white flour but if white flour is all that is available, the user will accept it. Goal preferences like this may also be balanced with the cost of achieving the goal. For example, if brown flour costs more than some measurable utility it brings, and white flour costs less than its utility, then white flour is the obvious choice despite the preference for brown flour. This type of problem is called a partial satisfaction planning (PSP) problem (van den Briel *et al.* 2004) and provides a starting point for this work. Work on this type of planning has been given recent attention (van den Briel *et al.* 2004; Do & Kambhampati 2004; Smith 2004; Brafman & Chernyavsky 2005; Nigenda & Kambhampati 2005; Benton, Do, & Kambhampati 2005). The primary focus of this doctoral research is to extend the expressiveness of this type of planning by using the current state of the art planning graph heuristics as a means for solving these problems.

The most recent work on this doctoral thesis has focused on extending PSP into handling goals with utility dependencies. That is, some goals are worth much more or less in conjunction with other goals. For instance, having both a left and right shoe is worth much more than having just one or the other and having two books on the same subject is worth less than the sum of having either book independently.[1] Another recent extension is the ability to handle "simple preferences" as defined by PDDL3 in the 5^{th} International Planning Competition. It turns out that planning problems with preferences defined in this way are very similar to PSP problems.

The rest of this extended abstract is organized as follows. First, we motivate the need for representing and handling goal utility dependencies in PSP and provide a framework of representing them using the General Additive Independence (GAI) model (Bacchus & Grove 1995) and give an outline of heuristic methods for handling them. To show the effectiveness of our framework, we provide empirical results on some benchmark planning domains. We then briefly outline the method of generating a PSP problem from "simple preferences" defined in PDDL3. Afterwards, we discuss future work.

Goal Utility Dependency

Classical planning problems define each goal as a member of a conjunctive set that must be satisfied at a plan's end. In partial satisfaction planning (PSP) we relax the constraint of ending a plan with every goal satisfied. Instead we define soft goals and provide each with a numeric utility value. This allows the planner to solve for a subset of the goals. We also attribute to each action a numeric cost. The planner then aims to find a plan with the best *net benefit*, where net benefit is defined as the difference between the satisfied goal utility and the action costs.

The process of finding plans in PSP is complicated by two types of dependencies between goals: (i) A set of goals may have *cost dependencies* in that there are dependencies among the plans to achieve them (making the cost of achieving them together significantly more or less than the total cost of achieving them in isolation) (ii) A set of goals may have *utility dependencies* in that achieving the goals together may lead to significantly different utility than the sum of

*Joint work with Minh Do, Palo Alto Research Center, Palo Alto, CA and Subbarao Kambhampati, Arizona State University, Tempe, AZ

[1]These are examples of mutual dependency. There is also the idea of conditional dependency, in which the utility of having one item is conditional on whether we have the other item. The difference is subtle, but the general idea is that conditional dependency is based upon an "if" relationship rather than an "and" relationship. Also note that the "and" relationship is more general and can be used to represent an "if" relationship by listing possible goal combinations.

achieving individual goals. Part of this dissertation work is on investigating heuristic approaches to handle both utility and cost dependencies together in PSP.

We have developed an approach for representing these utility dependencies between planning goals using the *Generalized Additive Independence (GAI)* model (Bacchus & Grove 1995) and a planning algorithm based on forward search that solves this extended PSP problem. The algorithm is based on the forward heuristic search described in the *Sapa*$^{\mathcal{PS}}$ planner (van den Briel *et al.* 2004). The main innovation is our heuristic, which is able to take into account both goal utility and goal achievement cost dependencies.

Problem Formulation & Heuristics

A classical planning problem is a 4-tuple $\langle F, I, G, A\rangle$ where: F is a set of predicate symbols representing state facts; I is the initial state, completely defined by predicates in F; G is a goal state, which is partially defined by a set of predicates in F; A is a set of actions with $a \in A$ is defined by pre and post-conditions $Precond(a), Effect(a) \subseteq F$. The plan is a sequence of actions in A such that, when executed from I, will achieve all goals $g \in G$. In PSP (Smith 2004; van den Briel *et al.* 2004), goals $g \in G$ have utility values $u_g \geq 0$, representing how much each goal is worth to a user, and each action $a \in A$ has an associated positive execution cost c_a. Moreover, not all goals in G need to be achieved. Let P be the lowest cost plan that achieves a subset $G' \subseteq G$ of those goals. The objective is to maximize the tradeoff between total utility $U(G')$ of G' and total cost of actions $a \in P$.

Work on PSP until now assumed that goals have no utility dependencies and thus their utilities are additive: $U(G') = \Sigma_{g \in G'} u_g$. To represent the goal utility dependencies we adopt the *Generalized Additive Independence(GAI)* model (Bacchus & Grove 1995). We named the PSP problem with utility dependencies represented by GAI model *PSP*$^{\mathcal{UD}}$. We chose this model because it is simple, intuitive and expressive. It also is more general than other commonly used models such as CP-Net (Brafman & Chernyavsky 2005) or UCP-Net (Boutilier *et al.* 2001). Because of this, representing goals specified using GAI may result in a problem size increase in comparison with these other modeling methods. However, its generality allows problem specification to be more straightforward for the user (i.e. there are no "inferred" utility values). A cost propagation process is used on the planning graph to estimate the achievement cost for each individual goal. After the propagation process is done we have an estimated cost $c(g)$ for each goal $g \in G$. As shown in (Do & Kambhampati 2001), if we use *max* propagation, then $c(g)$ will underestimate the cost to achieve g while there is no such guarantee for *sum* propagation.

The max family of heuristics tend to perform badly in practice. Therefore, we use an alternative approach of utilizing the relaxed plan employed by *Sapa*$^{\mathcal{PS}}$ for PSP[2]. For each state S explored in a progression planner, after building the relaxed planning graph and doing forward cost propagation on the graph, we extract a relaxed plan RP to support a subset of goals $G' \subseteq G$. Let $RP(G')$ be the relaxed plan with highest net-benefit value among those achieving $G' \subseteq G$, the relaxed plan heuristic for *PSP*$^{\mathcal{UD}}$ is:

$$h_{relax} = max_{G' \subseteq G}\,(u(G') - \sum_{a \in RP(G')} c_a) \qquad (1)$$

To capture the mutual cost dependencies between the goal achievement costs (i.e. cost dependencies), we find the set of actions shared between different partial plans achieving different goals. This allows the generation of $GS(a)$ which specifies the set of goals for which the action a contributes.

Given the utility dependencies represented by GAI local functions f^u and the goal achievement cost dependencies represented by goal supporting action set $GS(a)$, we set up an ILP encoding for h_{relax}. The purpose of this encoding is to capture the set of goals $G' \subseteq G$ that gives the maximum tradeoff between utility of G' and the cost of actions in the relaxed plan supporting G'.

[2]Variants of this approach are also used in several other PSP planners such as *AltAlt*^{ps} (van den Briel *et al.* 2004; Nigenda & Kambhampati 2005) or the orienteering planner (Smith 2004).

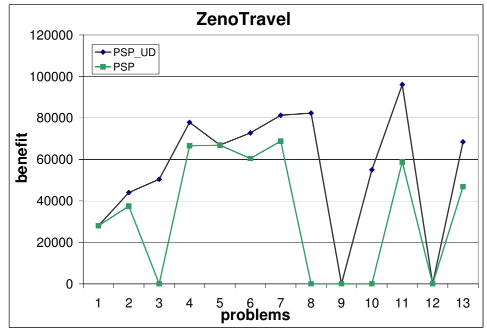

Figure 1: *Sapa*$^{\mathcal{PS}}_{\mathcal{UD}}$ and *Sapa*$^{\mathcal{PS}}$ in ZenoTravel domain.

Results

We have implemented the heuristic search algorithm for PSP problems discussed in this paper on top of the *Sapa*$^{\mathcal{PS}}$ planner. We call the new planner *Sapa*$^{\mathcal{PS}}_{\mathcal{UD}}$ and tested it on two sets of random *ZenoTravel* and *Satellite* problems. These problems were generated on top of the problem sets used in the Third International Planning Competition (Long & Fox 2003).

All tests were run using a Pentium IV 2.66GHz with 1GB RAM and a 1200 second time limit. Because A^*_{PSP} continuously finds better solutions given more time (or the termination node is found), the results reported in this section represent the plan with the highest benefit value found within the time limit. For solving the ILP encoding, we use the C version of `lp_solve ver5.5` software, a free solver, with a Java wrapper.

While *Sapa*$^{\mathcal{PS}}_{\mathcal{UD}}$ is sensitive to both cost and utility dependencies, *Sapa*$^{\mathcal{PS}}$ only accounts for cost dependencies. The empirical evaluation is designed to test whether *Sapa*$^{\mathcal{PS}}_{\mathcal{UD}}$ is able to solve the *PSP*$^{\mathcal{UD}}$ problems more effectively (i.e. with higher net benefit). Figure 1 and 2 show the comparison between those two planners.

PDDL3 "Simple Preferences" to PSP

The Fifth International Planning Competition defined preferences as a new language feature for PDDL3(Gerevini &

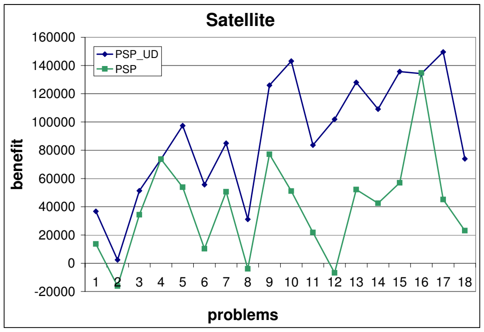

Figure 2: $Sapa^{\mathcal{PS}}_{\mathcal{UD}}$ and $Sapa^{\mathcal{PS}}$ in Satellite domain.

Long 2005). This feature allows domain modelers to express soft constraints on action conditions and goals. Each preference is given a name and an associated violation count. This count can then be used as part of a metric specifying how to measure the quality of the resulting plan.

In the planning competition, the "simple preferences" category of domains specifies preferences and plan metrics in a manner that allows problems to be converted from PDDL3 to PSP. The domains in this category define preferences on actions as well as goals. An example is the *drive* action of the trucks domain:

```
(:action drive
 :parameters
   (?t - truck ?from ?to - place)
 :precondition (and
   (at ?t ?from) (connected ?from ?to)
   (preference p-drive (and
    (ready-to-load goods1 ?from level0)
    (ready-to-load goods2 ?from level0)
    (ready-to-load goods3 ?from level0))
     ))
 :effect (and (not (at ?t ?from))
              (at ?t ?to)))
```

A plan metric assigns a weight to this preference in the following manner:

```
(:metric (+ (* 10 (is-violated p-drive))
))
```

A domain specified in this way can be compiled into a PSP problem (Benton, Kambhampati, & Do 2006). This is done by generating an action for each preference combination on the original action. The cost of executing the action is equal to the cost of not satisfying the preferences excluded from the action definition. Preferences on goals are handled similarly except actions provide a "has preference" goal with a utility that matches the cost of not having the preference.

Conclusion & Future Work

In this extended abstract, we discussed a framework of solving partial satisfaction planning (PSP) problems with utility dependencies and a way to handle IPC5 problems with "simple preferences" by compiling them to PSP problems. The former methods show that there exists expressive power in combining problem heuristics with declarative formulations. In this case, re-formulating the relaxed plan as an ILP allows us to impose more constraints that cannot easily be handled procedurally. Though there exists additional computational cost for generating the new heuristics, this is offset by the extra guidance achieved toward better quality plans. This line of reasoning is especially important to consider as the planning community's concerns have begun to focus on finding plans of quality as evidenced with the Fifth International Planning Competition (Gerevini & Long 2005).

For the future, this dissertation work will extend the heuristic and search architectures used to solve utility dependencies for dealing with trajectory preferences and constraints in PDDL3 (Gerevini & Long 2005). The idea is to include in the ILP formulation information about the time points on which actions are executed in the relaxed plan so that we may find good estimates of the best action orderings. We also plan on extending this work to take more negative information into account, following the example of AltWlt (Nigenda & Kambhampati 2005).

References

Bacchus, F., and Grove, A. 1995. Graphical models for preference and utility. In *Proc. of UAI-95*.

Benton, J.; Do, M. B.; and Kambhampati, S. 2005. Oversubscription planning with numeric goals. In *IJCAI*, 1207–1213.

Benton, J.; Kambhampati, S.; and Do, M. B. 2006. Yochan$\sqrt{}^{\int}$: Pddl3 simple preferences as partial satisfaction planning. In *Fifth International Planning Competition Extended Abstract Booklet*.

Boutilier, C.; Brafman, R.; Hoos, H.; ; and Poole, D. 2001. Reasoning with conditional ceteris paribus preference statements. In *Proc. of UAI-2001*.

Brafman, R. I., and Chernyavsky, Y. 2005. Planning with goal preferences and constraints. In *ICAPS*.

Do, M., and Kambhampati, S. 2001. Sapa: A domain-independent heuristic metric temporal planner. In *Proc. of ECP-01*.

Do, M., and Kambhampati, S. 2004. Partial satisfaction (over-subscription) planning as heuristic search. In *Knowledge Based Computer Systems*.

Gerevini, A., and Long, D. 2005. Plan constraints and preferences in PDDL3: The language of the fifth international planning competition. Technical report, University of Brescia, Italy.

Long, D., and Fox, M. 2003. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research*.

Nigenda, R. S., and Kambhampati, S. 2005. Planning graph heuristics for selecting objectives in over-subscription planning problems. In *ICAPS*.

Smith, D. 2004. Choosing objectives in over-subscription planning. In *Proc. of ICAPS-04*.

van den Briel, M.; Sanchez, R.; Do, M. B.; and Kambhampati, S. 2004. Effective approaches for partial satisfaction (over-subscription) planning. In *Proc. of AAAI-04*.