Planning with Partial Preference Models

Tuan A. Nguyen * and **Minh B. Do** † and **Subbarao Kambhampati** * and **Biplav Srivastava** ‡

* Dept. of Computer Science & Engineering, Arizona State University, Tempe, AZ 85287, USA. Email: {tanguye1,rao}@asu.edu

† Embedded Reasoning Area, Palo Alto Research Center, Palo Alto, CA 94304, USA. Email: minh.do@parc.com

‡ IBM India Research Laboratory, New Delhi and Bangalore, India. Email: sbiplav@in.ibm.com

Abstract

In many real-world planning scenarios, the users are interested in optimizing multiple objectives (such as makespan and execution cost), but are unable to express their exact tradeoff between those objectives. When a planner encounters such partial preference models, rather than look for a single optimal plan, it needs to present the pareto set of plans and let the user choose from them. This idea of presenting the full pareto set is fraught with both computational and user-interface challenges. To make it practical, we propose the approach of finding a *representative subset* of the pareto set. We measure the quality of this representative set using the *Integrated Convex Preference* (ICP) model, originally developed in the OR community. We implement several heuristic approaches based on the Metric-LPG planner to find a good solution set according to this measure. We present empirical results demonstrating the promise of our approach.

1 Introduction

In many real world planning scenarios, the user's preferences on desired plans are either unknown or at best partially specified (c.f. [Kambhampati, 2008]). In such cases, the planner's job changes from finding a single optimal plan to finding a set of representative solutions ("options") and present them to the user (with the hope that the user will find one of them desirable). Most work in automated planning ignores this reality, and assumes instead that the user's preferences on the plan quality are expressed in terms of a completely specified objective function.

One exception to this general trend is our own previous work [Srivastava *et al.*, 2007] which focused on diverse plan generation in the extreme case when there is absolutely no knowledge about the underlying preference model. In this case, the best strategy is to generate a set of plans that are maximally different (distant) from each other. The critical challenges turn out to be: (1) to develop meaningful distance measures between plans, and (2) to use them in the plan synthesis process. However, not all applications fall into this extreme case. More often than not, partial information about user preferences—especially, the attributes of the solutions that matter (e.g., flying time, ticket price, number of stops when buying airline ticket) — are known, although the exact trade-offs between them are not fully known. One approach of course would be to ignore this partial information and still search for plans that are maximally diverse according to a distance function. The problem is that the resulting set of plans, while diverse, may not differ in the attributes that matter. For example, two travel plans–one on United and the other on Delta Airlines–may be considered "distant" from each other, but may be equivalent if the user only cares about the makespan and cost.

In such cases with partially expressed preferences, the favored approach in operations research is to compute the *pareto optimal* set of non-dominated solutions. There are two problems with this approach—one computational, and other comprehensional. The computational problem is that the pareto set is often too large. While enumerating the entire pareto set is known to be already hard even in simpler combinatorial problems like scheduling [Carlyle *et al.*, 2003], it is even harder in planning since synthesis of a single feasible plan is often quite costly already. Coming to the second problem, even if we can afford to compute the full pareto set, it is unclear that users will be able to inspect such a large set of plans to identify the ones they prefer.

What is clearly needed is the ability to compute a small representative subset of plans from the pareto set and present them to the user. An immediate challenge is formalizing what it means for a subset of plans to be "*representative*" of the pareto set. Several obvious ideas, such as picking plans on the pareto set that are maximally distant from each other, turn out not to be robust. Instead, we adapt the idea of *Integrated Preference Function* (IPF) [Carlyle *et al.*, 2003] (and its special case *Integrated Convex Preference* (ICP)), that was developed in the Operations Research (OR) community in the context of multi-criteria scheduling. Given a set of quality criteria whose convex combination defines the unknown objective function, and a probability distribution[1] on the values the weights in the convex combination can take, ICP is able to associate a robust measure of representativeness for any set of solution plans. Armed with this, we can then formulate the problem of planning with partial preference models as one of finding a bounded set of solutions that have the best ICP measure.

[1] Even if we do not have any special knowledge about this probability distribution, we can always start by initializing it to be uniform, and gradually improve it based on interaction with the user.

We present a spectrum of approaches for solving this problem efficiently. We implement these approaches on top of Metric-LPG [Gerevini *et al.*, 2008]. Our empirical evaluation compares the approaches both among themselves as well as against existing methods for generating diverse plans that ignore the partial preference information (c.f. [Srivastava *et al.*, 2007]). Our results demonstrate the promise of our proposed solutions.

2 Problem Formulation

While our work is applicable to any general planning scenarios, to make our discussion concrete, we will concentrate on metric temporal planning where each action $a \in A$ has a duration d_a and execution cost c_a. The planner needs to find a plan $p = \{a_1 \ldots a_n\}$, which is a sequence of actions that is executable and achieves all goals. The two most common plan quality measures are: *makespan*, which is the total execution time of p; and *plan cost*, which is the total execution cost of all actions in p. In most real-world applications, these two criteria compete with each other: shorter plans usually have higher cost and vice versa. Throughout the paper, we will use the following assumptions:

- The desired objective function involves minimizing both components: $time(p)$ measures the makespan of the plan p and $cost(p)$ measures its execution cost.
- The quality of a plan p is a convex combination: $f(p, w) = w \times time(p) + (1 - w) \times cost(p)$, where weight $w \in [0, 1]$ represents the trade-off between the two competing objective functions.
- The belief distribution of w over the range $[0, 1]$ is known. If the user does not provide any information or we have not learnt anything about the preference on the trade-off between *time* and *cost* of the plan, then the planner can assume a uniform distribution (and improve it later using techniques such as preference elicitation).

Given that the exact value of w is unknown, we cannot find a single optimal plan. Therefore, the best strategy is to find a representative set of non-dominated plans[2] minimizing the expected value of $f(p, w)$ with regard to the given distribution of w over $[0, 1]$.

Example: Figure 1 shows our running example in which there are a total of 7 plans with their $time(p)$ and $cost(p)$ values as follows: $p_1 = \{4, 25\}$, $p_2 = \{6, 22\}$, $p_3 = \{7, 15\}$, $p_4 = \{8, 20\}$, $p_5 = \{10, 12\}$, $p_6 = \{11, 14\}$, and $p_7 = \{12, 5\}$. Among these 7 plans, 5 of them belong to a pareto optimal set of non-dominated plans: $\mathcal{P}_p = \{p_1, p_2, p_3, p_5, p_7\}$. The other two plans are dominated by some plans in $\mathcal{P}_p$: p_4 is dominated by p_3 and p_6 is dominated by p_5. Plans in the $\mathcal{P}_p$ are depicted in solid dots in Figure 1.

3 Integrated Convex Preference (ICP)

The *Integrated Preference Function* (IPF) [Carlyle *et al.*, 2003] has been used to measure the quality of a solution set in a wide range of multi-objective optimization problems. In this section, we will first discuss IPF in its general form and then its special case called Integrated Convex Preference

[2]A plan p_1 is dominated by p_2 if $time(p_1) \geq time(p_2)$ and $cost(p_1) \geq cost(p_2)$ and at least one of the inequalities is strict.

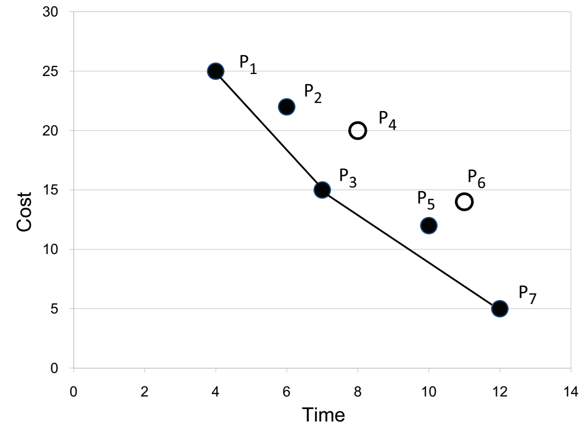

Figure 1: Solid dots represents plans in the pareto set $(p_1, p_2, p_3, p_5, p_7)$. Connected dots represent plans in the lower convex hull (p_1, p_3, p_7) giving optimal ICP value for any distribution on trade-off between *cost* and *time*.

(ICP), which is applicable to our problem formulation discussed in the previous section.

The IPF measure assumes that the user preference model is represented by two factors: (1) a probability distribution $h(\alpha)$ of parameter vector α such that $\int_\alpha h(\alpha)\, d\alpha = 1$ (in the absence of any special information about the distribution, $h(\alpha)$ can be assumed to be uniform), and (2) a function $f(p, \alpha) : \mathcal{S} \rightarrow \mathbb{R}$ (where $\mathcal{S}$ is the solution space) combines different objective functions into a single real-valued quality measure for solution p. The IPF value of solution set $\mathcal{P} \subseteq \mathcal{S}$ is defined as:

$$IPF(\mathcal{P}) = \int_\alpha h(\alpha) f(p_\alpha, \alpha)\, d\alpha \qquad (1)$$

with $p_\alpha = \underset{p \in \mathcal{P}}{\operatorname{argmin}}\, f(p, \alpha)$ is the best solution according to $f(p, \alpha)$ for each given α value. Let p_α^{-1} be its inverse function specifying a range of α values for which p is an optimal solution according to $f(p, \alpha)$, as p_α is piecewise constant, the $IPF(\mathcal{P})$ value can be computed as:

$$IPF(\mathcal{P}) = \sum_{p \in \mathcal{P}} \left[\int_{\alpha \in p_\alpha^{-1}} h(\alpha) f(p, \alpha)\, d\alpha \right] \qquad (2)$$

let $\mathcal{P}^* = \{p \in \mathcal{P} :\ p_\alpha^{-1} \neq \emptyset\}$ then we have:

$$IPF(\mathcal{P}) = IPF(P^*) = \sum_{p \in P^*} \left[\int_{\alpha \in p_\alpha^{-1}} h(\alpha) f(p, \alpha)\, d\alpha \right] \qquad (3)$$

The $IPF(\mathcal{P})$ now can be interpreted as the expected utility value of the *best solution* of $\mathcal{P}$ using probability distribution $h(\alpha)$ on the trade-off value α and the utility function $f(p, \alpha)$ measuring the quality of the *optimal* solution for each given α value, represented by the $\mathcal{P}^*$ set. Therefore, the set $\mathcal{P}^*$ of solutions (known as *lower convex hull* of $\mathcal{P}$) with the minimal IPF value is most likely to contain the desired solutions that the user wants and in essense a good representative of $\mathcal{P}_p$. In our running example, $\mathcal{P}^* = \{p_1, p_3, p_7\}$ and is highlighted by connected dots in Figure 1. the *best solution* of $\mathcal{P}$ using: (1) the user belief probability distribution $h(\alpha)$ on the value of α and (2) the utility function $f(p, \alpha)$ measuring the quality of the *optimal* solution for each given α value, represented by the $\mathcal{P}^*$ set. Therefore, the set $\mathcal{P}^*$ of solutions with the minimal IPF value is most likely to contain the desired solutions that the user wants and in essense a good representative of $\mathcal{P}$. Note that if $\mathcal{P}_p \subseteq \mathcal{P}$ is the pareto optimal solution set, then $\mathcal{P}^* \subseteq \mathcal{P}_p$ (we avoid the proof due to space limit). In our running example, $\mathcal{P}^* = \{p_1, p_3, p_7\}$ and is highlighted by connected dots in Figure 1.

IPF for Temporal Planning: The user preference model in our target domain of temporal planning is a convex combination of the *time* and *cost* quality measures. The IPF measure now is called *Integrated Convex Preference* (ICP). Given a set of plans $\mathcal{P}^*$, let $t_p = time(p)$ and $c_p = cost(p)$ be the makespan and total execution cost of plan $p \in \mathcal{P}^*$, the ICP value of $\mathcal{P}^*$ with regard to the objective function $f(p, w) = w \times t_p + (1 - w) \times c_p$ and the parameter vector $\alpha = (w, 1 - w)$ ($w \in [0, 1]$) is defined as:

$$ICP(\mathcal{P}^*) = \sum_{i=1}^{k} \int_{w_{i-1}}^{w_i} h(w)(w \times t_{p_i} + (1-w) \times c_{p_i})dw \quad (4)$$

where $w_0 = 0$, $w_k = 1$ and $p_i = \underset{p \in \mathcal{P}^*}{\operatorname{argmin}} f(p, w)\ \forall w \in [w_{i-1}, w_i]$. In other words, we divide $[0, 1]$ into non-overlapping regions such that in each region (w_{i-1}, w_i) there is a single solution $p_i \in \mathcal{P}^*$ that has better $f(p_i, w)$ value than all other solutions in $\mathcal{P}^*$.

We select the IPF/ICP measure to evaluate our solution set due to its several nice properties:

- If $\mathcal{P}_1, \mathcal{P}_2 \subseteq \mathcal{P}$ and $ICP(\mathcal{P}_1) < ICP(\mathcal{P}_2)$ then $\mathcal{P}_1$ is *probabilistically* better than $\mathcal{P}_2$ in the sense that for any given w, let $p_1 = \underset{p \in \mathcal{P}_1}{\operatorname{argmin}} f(p, w)$ and $p_2 = \underset{p \in \mathcal{P}_2}{\operatorname{argmin}} f(p, w)$, then the probability of $f(p_1, w) < f(p_2, w)$ is higher than the probability of $f(p_1, w) > f(p_2, w)$.
- If $\mathcal{P}_1$ is *obviously* better than $\mathcal{P}_2$, then the ICP measure agrees with the assertion. More formally: if $\forall p_2 \in \mathcal{P}_2, \exists p_1 \in \mathcal{P}_1$ such that p_2 is dominated by p_1, then $ICP(\mathcal{P}_1) < ICP(\mathcal{P}_2)$.

Empirically, extensive experimental results on scheduling problems in [Fowler *et al.*, 2005] have shown that ICP measure *"evaluates the solution quality of approximation robustly (i.e., similar to visual comparison results) while other alternative measures can misjudge the solution quality"*.

4 Finding Representative Plans Using ICP

Using ICP as the basis, we can now formally state the specific problem we aim to tackle: *Given a planning problem whose solution quality is determined by an objective function that is a convex combination of a set of criteria, and $h(\alpha)$ is the estimated distribution of α, the parameter vector, that combines these criteria, our objective is to find a a set of solution plans $\mathcal{P}_s$ where $|\mathcal{P}_s| \leq k$ and $ICP(\mathcal{P}_s)$ is the lowest.*

Notice that we restrict the size of the solution set returned. This is important since the ICP measure is a monotonically non-increasing function of the solution set (specifically, given two solution sets $\mathcal{P}_1$ and $\mathcal{P}_2$ such that the latter is a superset of the former, it is easy to see that $ICP(\mathcal{P}_2) \leq ICP(\mathcal{P}_1)$).

In the rest of this section, we will consider three approximate methods for tackling this problem.

4.1 Sampling Weight Values

Given that the distribution of trade-off value w is known, the straightforward way to find a set of representative solutions is to first sample a set of k values for w: $\{w_1, w_2, ..., w_k\}$ based on the distribution $h(w)$. For each value w_i, we can find an (optimal) plan p_i minimizing the value of the overall objective function $f(p, w_i) = w_i \times t_p + (1 - w_i) \times c_p$. The final set of solution $\mathcal{P} = \{p_1, p_2, ...p_k\}$ is then filtered to remove duplicate and dominated solutions resulting the plans making up the lower-convex hull. The final set can then be returned to the user. While intuitive and easy to implement, this sampling-based approach has several potential flaws that can limit the quality of its resulting plan set.

First, given that k solution plans are searched sequentially and independently of each other, even if the plan p_i found for each w_i is optimal, the final solution set $\mathcal{P} = \{p_1, p_2...p_k\}$ may not even be the optimal set of k solutions with regard to the ICP measure. More specifically, for a given set of solutions $\mathcal{P}$, some tradeoff value w, and two non-dominated plans p, q such that $f(p, w) < f(q, w)$, it is possible that $ICP(\mathcal{P} \cup \{p\}) > ICP(\mathcal{P} \cup \{q\})$. In our running example, let $\mathcal{P} = \{p_2, p_5\}$ and $w = 0.8$ then $f(p_1, w) = 0.8 \times 4 + 0.2 \times 25 = 8.2 < f(p_7, w) = 0.8 \times 12 + 0.2 \times 5 = 10.6$. Thus, the planner will select p_1 to add to $\mathcal{P}$ because it looks locally better given the weight $w = 0.8$. However, $ICP(\{p_1, p_2, p_5\}) \approx 10.05 > ICP(\{p_2, p_5, p_7\}) \approx 7.71$ so indeed by taking previous set into consideration then p_7 is a much better choice than p_1.

Second, the trade-off values w are sampled based on a given distribution, and independently of the particular planning problem being solved. As there is no relation between the sampled w values and the solution space of a given planning problem, sampling approach may return very few solutions even if we sample a large number of weight values w. In our example, if all w samples have values $w \leq 0.67$ then the optimal solution returned for any of them will always be p_7. However, we know that $\mathcal{P}^* = \{p_1, p_3, p_7\}$ is the optimal set according to the ICP measure. Indeed, if $w \leq 0.769$ then the sampling approach can only find the set $\{p_7\}$ or $\{p_3, p_7\}$ and still not be able to find the optimal set $\mathcal{P}^*$.

4.2 ICP Sequential Approach

Given the potential drawbacks of the sampling approach outlined above, we also pursued an alternative approach that takes into account the ICP measure more actively. Specifically, we incrementally build the solution set $\mathcal{P}$ by finding a solution p such that $\mathcal{P} \cup \{p\}$ has the lowest ICP value. We can start with an empty solution set $\mathcal{P} = \emptyset$, then at each step try to find a new plan p such that $\mathcal{P} \cup \{p\}$ has the lowest ICP value.

While this approach directly takes the ICP measure into consideration at each step of finding a new plan, and avoids the drawbacks of the sampling-based approach, it also has its own share of potential flaws. Given that the set is built incrementally, the earlier steps where the first "seed" solutions are

Algorithm 1: Incrementally find solution set $\mathcal{P}$

```
1  Input: A planning problem with a solution space S;
   maximum number of plans required k; number of
   sampled trade-off values k_0 (0 < k_0 < k); time bound t;
2  Output: A plan set P (|P| ≤ k);
3  begin
4      W ← sample k_0 values for w;
5      P ← find optimal plans in S for each w ∈ W;
6      while |P| < k and search_time < t do
7          Search for p s.t. ICP(P ∪ {p}) < ICP(P)
8          P ← P ∪ {p}
9      Return P
10 end
```

found are very important. The closer the seed solutions are to the global lower convex hull, the better the improvement in the ICP value. In our example (Figure 1), if the first plan found is p_2 then the subsequent plans found to best extend $\{p_2\}$ can be p_5 and thus the final set does not come close to the optimal set $\mathcal{P}^* = \{p_1, p_3, p_7\}$.

4.3 Hybrid Approach

In this approach, we aim to combine the strengths of both the sampling and ICP-sequential approaches. Specifically, we use sampling to find several plans optimizing for different weights. The plans are then used to seed the subsequent ICP-sequential runs. By seeding the hybrid approach with good quality plan set scattered across the pareto optimal set, we hope to gradually expand the initial set to a final set with a much better overall ICP value. Algorithm 1 shows the pseudo-code for the hybrid approach. We first independently sample the set of k_0 values (with k_0 pre-determined) of w given the distribution on w (step 4). We then run a heuristic planner multiple times to find an optimal (or good quality) solution for each trade-off value w (step 5). We then collect the plans found and seed the subsequent runs when we incrementally update the initial plan set with plans that lower the overall ICP value (steps 6-8). The algorithm terminates and returns the latest plan set (step 9) if k plans are found or the time bound exceeds.

4.4 Making LPG Search Sensitive to ICP

We use a modified version of the Metric-LPG planner [Gerevini *et al.*, 2008] as the base planner in implementing our algorithms. Selection of Metric-LPG was motivated by its very flexible local-search framework that has been extended to handle various objective functions, and the fact that it can be made to search for single or multiple solutions. Specifically, for the sampling-based approach, we first sample the w values based on a given distribution. For each w value, we set the metric function in the domain file to: $w \times makespan + (1 - w) \times totalcost$, and run the original LPG in the quality mode to heuristically find the best solution for that metric function. The final solution set is filtered to remove any duplicate solutions, and returned to the user.

For the ICP-sequential and hybrid approach, we can not use the original LPG implementation as is and need to modify the neighborhood evaluation function in LPG to take into account the ICP measure and the current plan set $\mathcal{P}$. For the rest of this section, we will explain this procedure in detail.

Background: Metric-LPG uses local search to find plans within the space of *numerical action graphs* (NA-graph). This leveled graph consists of a sequence of interleaved proposition and action layers. The proposition layers consist of a set of propositional and numerical nodes, while each action layer consists of at most one action node, and a number of no-op links. An NA-graph G represents a valid plan if all actions' preconditions are supported by some actions appearing in the earlier level in G. The search neighborhood for each local-search step is defined by a set of graph modifications to fix some remaining inconsistencies (unsupported preconditions) p at a particular level l. This can be done by either inserting a new action a supporting p or removing from the graph the action a that p is a precondition of (which can introduce new inconsistencies).

Each local move creates a new NA-graph G', which is evaluated as a weighted combination of two factors: $SearchCost(G')$ and $ExecCost(G')$. Here, $SearchCost(G')$ is the amount of search effort to resolve inconsistencies newly introduced by inserting or removing action a; it is measured by the number of actions in a relaxed plan R resolving all such inconsistencies. The total cost $ExecCost(G')$, which is a default function to measure plan quality, is measured by the total *action execution costs* of all actions in R. The two weight adjustment values α and β are used to steer the search toward either finding a solution quickly (higher α value) or better solution quality (higher β value). LPG then selects the local move leading to the smallest $E(G')$ value.

Adjusting the evaluation function $E(G')$ for finding set of plans with low ICP measure: To guide LPG towards optimizing our ICP-sensitive objective function instead of the original minimizing cost objective function, we need to replace the default plan quality measure $ExecCost(G')$ with a new measure $ICPEst(G')$. Specifically, we adjust the function for evaluating each new NA-graph generated by local moves at each step to be a combination of $SearchCost(G')$ and $ICPEst(G')$. Given the set of found plans $\mathcal{P} = \{p_1, p_2, ..., p_n\}$, $ICPEst(G')$ guides LPG's search toward a plan p generated from G' such that the resulting set $\mathcal{P} \cup \{p\}$ has a minimum ICP value: $p = \underset{p}{\operatorname{argmin}}\, ICP(\mathcal{P} \cup \{p\})$. Thus, $ICPEst(G')$ estimates the expected total ICP value if the best plan p found by expanding G' is added to the current found plan set $\mathcal{P}$. Like the original Metric-LPG, p is estimated by $p_R = G' \bigcup R$ where R is the relaxed plan resolving inconsistencies in G' caused by inserting or removing a. The $ICPEst(G')$ for a given NA-graph G' is calculated as: $ICPEst(G') = ICP(\mathcal{P} \cup p_R)$ with the ICP measure as described in Equation 4. Notice here that while $\mathcal{P}$ is the set of valid plans, p_R is not. It is an invalid plan represented by a NA-graph containing some unsupported preconditions. However, Equation 4 is still applicable as long as we can measure the time and cost dimensions of p_R. To measure the makespan of p_R, we estimate the time points at which unsupported facts in G' would be supported in $p_R = G' \cup R$ and propagate them over actions in G' to its last level. We then take the earliest time point at which all facts at the last level appear to measure the makespan of p_R. For the cost measure,

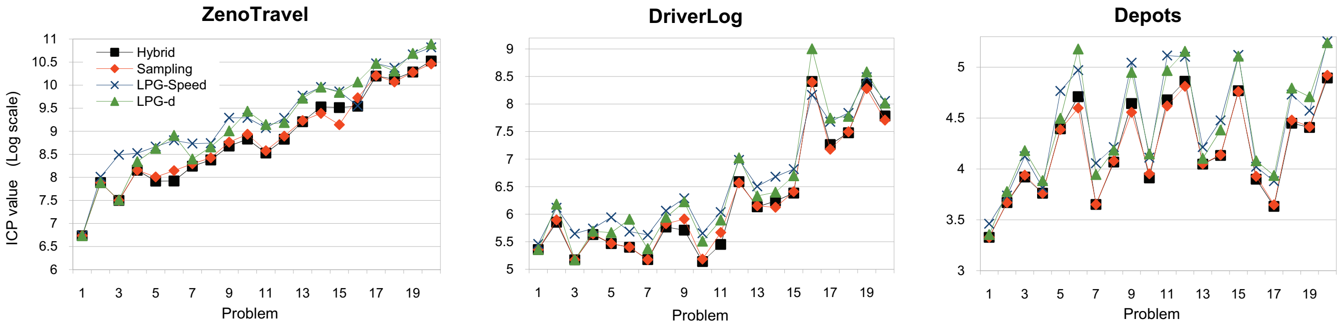

Figure 2: Results for the *ZenoTravel*, *Depot*, and *DriverLog* domains comparing the sampling-based, Hybrid, and baseline LPG approaches on the overall ICP value (log scale).

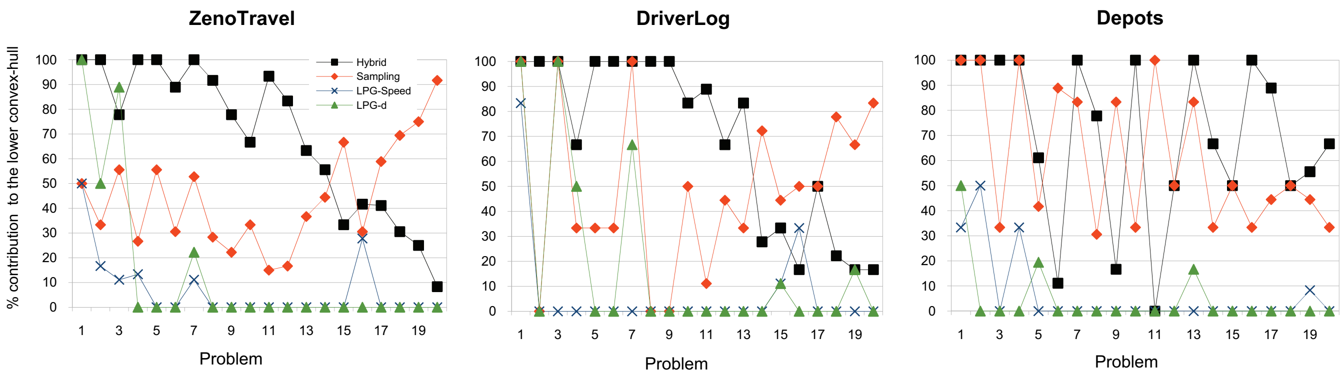

Figure 3: Results for the *ZenoTravel*, *Depot*, and *DriverLog* domains comparing the sampling-based, Hybrid, and baseline LPG approaches on the percentage of plans contributed to the overall lower convex hull.

we just sum the individual costs of all actions in p_R.

At each step of LPG's local search framework, combining $ICPEst(G')$ with $SearchCost(G')$ gives us an evaluation function that fits right into the original Metric-LPG framework and prefers a NA-graph G' in the neighborhood of G that gives the best trade-off between the estimated effort to repair and the estimated decrease in quality of the next resulting plan set.

5 Experimental Results

We have implemented several approaches based on our algorithms discussed in Section 4: Sampling (Section 4.1), ICP-sequential (Section 4.2) and Hybrid that combines both (Section 4.3). To evaluate the utility of taking partial preferences into account, we compare our results against the naive approaches that generate multiple plans without explicitly taking into account the partial preference model. Specifically, we run the default LPG planner with different random seeds to find multiple non-dominated plans. The LPG planner was run with both *speed* setting, which finds plans quickly, and *diverse* setting, which takes longer time to find better set of diverse plans [Srivastava *et al.*, 2007].

We test all implementations against a set of 20 problems in each of several benchmark temporal planning domains used in the previous International Planning Competitions (IPC): *ZenoTravel*, *DriverLog*, and *Depot*. The only modification to the original benchmark set is the added action costs. The descriptions of these domains can be found at the IPC website (*ipc.icaps-conference.org*). The experiments were conducted Intel Core2 Duo machine with 3.16GHz CPU and 4Gb RAM. For all approaches, we search for a maximum of 10 plans within the 10-minute time limit for each problem, and the resulting plan set is used to compute the ICP value. We assume no special knowledge of the distribution of the weight vectors, and thus set $h(\alpha)$ to be uniform in computing ICP. As LPG is a stochastic local search planner, we run it three times for each problem and average the results. In 77% and 70% of 60 problems in the three tested domains for Hybrid and Sampling approaches respectively, the standard deviation of ICP values of plan sets are at most 5% of the average values. This indicates that ICP values of plan set in different runs are quite stable. As the Hybrid approach is an improved version of ICP-sequential and gives better results in almost all tested problems, we omit ICP-sequential in discussions below.

For the uniform distribution on the trade-off value w between *makespan* and *plan cost*, Figure 2 shows the comparison between different approaches based on the overall ICP value of the solution set (smaller is better). In all three domains, the results in log-scale show that both Hybrid and Sampling approaches perform significantly better than the two baseline approaches of using LPG. Overall, among 20 tested problems for each of the ZenoTravel, DriverLog, and Depots domains, the Hybrid approach is better than LPG-speed in all 60 problems and better than LPG-d in 19/20, 18/20, and 20/20 problems respectively. The Sampling approach is better than LPG-speed in 19/20, 20/20 and 20/20 and is better than LPG-d in 18/20, 18/20, and 20/20 problems respectively. The Hybrid approach also dominates Sampling in the majority of problems; it is better in 15/20, 11/20, and 11/20 problems in ZenoTravel, DriverLog, and Depots domains (both approaches return plan sets with equal ICP values for 3 problems in each of the DriveLog and Depots domains).

We also compare the relative total number of plans in the lower convex-hull found by each approach. Given that this is the set that should be returned to the user (to select one from), the higher number is the better. To measure the relative performance of different approaches with respect to this measurement, we first create the set S combining the plans returned by all different approaches. We then compute the set $S_{lch} \subseteq S$ of plans in the lower convex hull among all plans in S. We measure the percentages of plans in S_{lch} that are actually returned by each of our tested approaches and Figure 3 shows the comparison between them. Overall, almost all plans on the lower convex hull are contributed by either Hybrid or Sampling approaches in all tested problems. Between Hybrid and Sampling approaches, Hybrid is again in general better for this criterion. It returns more plans in the lower convex hull than Sampling in 15/20, 10/20 (and another 4 equals), 11/20 (and another 6 equals) problems for ZenoTravel, DriverLog, and Depots respectively.

The results above show that naive approaches which ignore the user's partial preferences and focus on generating diverse plans do not work well. They also show that the hybrid method, which is sensitive to the ICP during search, tends to generate more plans in the lower convex hull than the sampling method. This advantage is reduced to some extent for some of the larger problems in ZenoTravel and DriverLog, mostly because the sampling phase of the hybrid approach takes too long, leaving too little time for the sequential phase that is aimed at improving the seed set. To address this, in future we propose to consider a more dynamic and problem-dependent ways of splitting time between the sampling and sequential phases of the hybrid approach.

6 Related Work

Currently there are very few research efforts in the planning literature that explicitly consider incompletely specified user preferences during planning. The usual approach for handling multiple objectives is to assume that a specific way of combining the objectives is available [Refanidis and Vlahavas, 2003; Do and Kambhampati, 2002]. Brafman & Chernyavsky(2005) discuss a CSP-based planner that finds the pareto optimal set of plans given the qualitative preferences on goals. There is no action cost and makespan measurements such as in our problem setting. Other relevant work includes [Bryce *et al.*, 2007], in which the authors devise a variant of LAO* algorithm to search for a conditional plan with multiple execution options for each observation branch that are non-dominated with respect to objectives like probability and cost to reach the goal.

7 Conclusion

In this paper, we considered an approach of finding a representative subset of the pareto set utilizing the belief distribution of the trade-offs between conflicting objective functions (e.g., plan makespan and execution cost). We measured the quality of this representative solution set using the *Integrated Convex Preference* (ICP) model and presented several heuristic approaches based on the Metric-LPG planner [Gerevini *et al.*, 2008] to find a good solution set according to this measure. We showed empirically that taking partial preferences into account does improve the quality of the plan set returned to the users.

While a planning agent may well start with a partial preference model, in the long run, we would like the agent to be able to improve the preference model through repeated interactions with the user. Accordingly, we are currently working on adding a learning component to our framework. In our context, the learning will involve improving the estimate of $h(\alpha)$ based on the feedback about the specific plan that the user selects from the set returned by the system.

Acknowledgements: We thank Alfonso Gerevini and his group for the LPG source code as well as clarifications. We also thank Menkes van den Briel for drawing our attention to ICP measure initially. Tuan Nguyen's research is supported by a Science Foundation of Arizona fellowship. Kambhampati's research is supported in part by ONR grants N00014-09-1-0017 and N00014-07-1-1049, and the DARPA Integrated Learning Program (through a sub-contract from Lockheed Martin).

References

[Brafman and Chernyavsky, 2005] Ronen I. Brafman and Yuri Chernyavsky. Planning with goal preferences and constraints. In *Proceeding of ICAPS-05*, 2005.

[Bryce *et al.*, 2007] D. Bryce, W. Cushing, and S. Kambhampati. Model-lite planning: Diverse multi-option plans & dynamic objective functions. In *ICAPS 2007 Workshop on Planning and Plan Execution for Real World Systems*, 2007.

[Carlyle *et al.*, 2003] W. M. Carlyle, J. W. Fowler, and B. Kim E. S. Gel. Quantitative comparison of approximate solution sets for bi-criteria optimization problems. *Decision Sciences*, 34(1), 2003.

[Do and Kambhampati, 2002] Minh B. Do and Subbarao Kambhampati. Sapa: A scalable multi-objective metric temporal planer. *Journal of Artificial Intelligence Research*, 20:155–194, 2002.

[Fowler *et al.*, 2005] J. W. Fowler, B. Kim, W. M. Carlyle, and S. Horng E. S. Gel. Evaluating solution sets of a posteriori solution techniques for bi-criteria combinatorial optimization problems. *Journal of Scheduling*, 8, 2005.

[Gerevini *et al.*, 2008] Alfonso Gerevini, Alessandro Saetti, and Ivan Serina. An approach to efficient planning with numerical fluents and multi-criteria plan quality. *Artificial Intelligence Journal*, 172(8-9), 2008.

[Kambhampati, 2008] Subbarao Kambhampati: Model-lite Planning for the Web Age Masses: The Challenges of Planning with Incomplete and Evolving Domain Models. AAAI 2007.

[Refanidis and Vlahavas, 2003] Ioanis Refanidis and Ioannis Vlahavas. Multiobjective heuristic state-space planning. *Artificial Intelligence Journal*, 145:1–32, 2003.

[Srivastava *et al.*, 2007] B. Srivastava, S. Kambhampati, T.A. Nguyen, M.B. Do, A. Gerevini, and I. Serina. Domain-independent approaches for finding diverse plans. In *IJCAI*, 2007.