Towards Model-lite Planning: Extending Sapa for Incomplete Metric Model

Tuan Nguyen and Subbarao Kambhampati
Dept. of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287, USA
{tuan.anh.nguyen, rao}@asu.edu

Biplav Srivastava
IBM India Research Laboratory
New Delhi and Bangalore, India
sbiplav@in.ibm.com

Abstract

Most of current works on planning community assume the completeness of the physical dynamics or the user preference model. Unfortunately, many domains in the real-world are complicated enough so that modeling simply can not be done completely. As a result, there has been some initial efforts to shift the research in the field towards dealing with domains with approximate or even shallow models (Kambhampati 2007). In this paper, we take that view point in the context of Sapa (Do and Kambhampati 2004) planner and consider the scenario in which the exact trade-off between objective functions *time* and *cost* of the plan is unknown to the planner. We propose an approach using heuristic search to find a set of plans which shows the usefulness in supporting the user's decision.

Introduction

Current research in planning community concentrates to find a plan possibly optimizing some particular metric function. However, in many planning situations, the quality of a plan is determined by various objective functions, for instance some quantitative features such as time, cost, fuel, or qualitative features such as preferring to visit some place during a travel plan. Combining all objective functions the user is interested in, as most of the works do, gives a simple approach which will lessen the hardness of solving the problem, but is not natural since the user preference is complicated and incomplete, i.e. they do not know exactly how to combine all of their interested objectives of the plans into one single value function.

To address planning problems optimizing various criteria, methods to deal with them need to be considered. The majority of solution approaches in multi-objective optimization can be divided into three categories. In *a priori*, the users preference information is assumed to be obtained before solving the problem. The second approaches is *interactive*, in which the system and user interact with each other during the solving process, and based on the critics from the users, the system will eventually give a solution the user wants. In the third approach, called *a posteriori*, the complete preference information is not considered in advance, but the incomplete trade-off information among various kinds of preference is considered so that the users can select one solution among the set found by the system.

In this work, we will tackle planning problems with incomplete user preference model in the context of Sapa planner (Do and Kambhampati 2004), where the user is assumed to be interested in *time* and *cost* of the plans. However, unlike Sapa, which can be considered as using *a priori* approach with the pre-defined trade-off α between two objectives, we will assume the distribution of α. The motivation of our approach is that normally, quantitative features as *time* and *cost* are considered linearly trade-off by the users, but extracting the distribution of trade-off value is more feasible, for instance by statistic on people used the system, than fixing the trade-off value.

The paper is organized as follows: the next section discusses about *Integrated Preference Function* (Carlyle *et al.* 2003) which will be used as the measure to evaluate our set of plans, followed by its special case *Integrated Convex Preference*. We then formalize our problem description and the assumption for our work, and the solution approach using hill-climbing method. After the proposed model for testing the approach, we conclude the work with some extensions for the future.

Integrated Preference Function (IPF)

In order to support the user using *a posteriori* approach, we need a measure to determine the quality of a set of solutions. One of the characteristic for a good solution set X is that it contains a desired solution satisfying the user hidden preference model, or approximates X as close as possible, so that the user eventually can select her desired solution x^* or one very close to it. *Integrated Preference Function* (IPF) (Carlyle *et al.* 2003) aiming to measure the expected value function of a solution set has been shown to evaluate robustly how well a set of solutions approximates the true set of ideal solutions (Fowler *et al.* 2005).

Formalization of IPF measure

The general form of IPF measure assumes that the user preference model can be represented with two factors: (1) a distribution of parameter vector α, and (2) a combination of various objective functions into a scalar value $g(x; \alpha)$. An example of a convex combination value function of a solution x with two objective functions f_1 and f_2 is $g(x; \alpha) = \alpha_1 f_1 + \alpha_2 f_2$, in which $\alpha = (\alpha_1, \alpha_2)$ and $\alpha_1 + \alpha_2 = 1$.

Formally, given a set of solutions X and a distribution function of parameter vector α over its domain A: $h : A \rightarrow$

R^+ such that $\int_{\alpha \in A} h(\alpha)\, d\alpha = 1$, the IPF measure of X is defined as follows:

$$IPF(X) = \int_{\alpha \in A} h(\alpha) g(x_g(\alpha); \alpha)\, d\alpha \qquad (1)$$

in which $x_g(\alpha)$ is the optimal solution with respect to the value function g and parameter vector α.

For each parameter vector α, let $x_g(\alpha) = argmin_{x \in X} g(x; \alpha)$, and $x_g^{-1}(\alpha)$ be its inverse function. As $x_g(\alpha)$ is a piecewise constant over the domain A, the value IPF(X) can be computed by decomposing it into several value functions on each portion $x_g^{-1}(\alpha)$ of A on which a particular solution x is optimal:

$$IPF(X) = \sum_{x \in X} [\int_{\alpha \in x_g^{-1}(\alpha)} h(\alpha) g(x; \alpha)\, d\alpha] \qquad (2)$$

This equation can be interpreted as the expected value of X with respect to the value function $g(x, \alpha)$. Therefore, the set of solutions with minimum IPF value contains the desired solutions that the user wants in infinite numbers of transactions.

Integrated Convex Preference (ICP)

Extracting the correct preference model is very difficult, so one of the most frequently used model is the convex combination of objective functions, and in this case the *Integrated Preference Function* is called *Integrated Convex Preference* (ICP). In this section, we consider this special form of value function in bi-objective optimization problems, the exact computation of ICP value and its geometric meaning.

Consider a set of solutions $X = \{x_1, x_2, ..., x_n\}$, and let f_{1i}, f_{2i} be two objective functions of each solution $x_i \in X$. The convex combination value function of a solution x_i with parameter vector $\alpha = (w, 1-w)$ is defined as:

$$g(x_i, w) \stackrel{def}{=} g(x_i, \alpha) = w f_{1i} + (1-w) f_{2i} (w \in [0, 1]) \quad (3)$$

Comparing solutions in X to each other with respect to the value function g, we have the following properties.

Property 1: Given $x_i, x_j \in X (i \neq j)$, if x_i is *dominated* by x_j [1] then $g(x_i, w) \leq g(x_j, w)$.

Property 2: Given $x_i, x_j \in X (i \neq j)$ such that $f_{1i} < f_{1j}$, $f_{2i} > f_{2j}$. Then $\forall x_k \in X$ such that $f_{1i} < f_{1k} < f_{1j}$, $f_{2i} < f_{2k} < f_{2j}$ and $(f_{2i} - f_{2j}) f_{1k} + (f_{1j} - f_{1i}) f_{2k} > 0$: $g(x_i, w) < g(x_k, w) \vee g(x_j, w) < g(x_k, w)$, $\forall w \in [0, 1]$.

Figure 1 illustrates the meaning of this property. Solution x_k (mentioned in property 2) inside the upper right triangle with the hypotenuse created by (f_{1i}, f_{2i}) and (f_{1j}, f_{2j}) is worse than either x_i or x_j in term of value function g because for any $w \in [0, 1]$, we can draw a line crossing either x_i or x_j parallel with but below the line crossing x_k with the same slope $\frac{-w}{1-w}$.

According to these properties, only extreme point solutions in X, which can be found with well-known convex-hull algorithm, contribute to the computation of ICP(X) and should be considered by the user. The exact computation of ICP(X) includes three steps:

- Find the set of extreme point solution in X, called $E_X = \{x_{e_1}, x_{e_2}, ..., x_{e_m}\}$.
- For each $x_{e_i} \in E_X$, find the portion $[w_{i-1}, w_i]$ $(0 \leq w_{i-1} < w_i \leq 1)$ on which x_{e_i} is optimal using linear inequalities:
  $(w f_{1e_i} + (1-w) f_{2e_i}) - (w f_{1e_j} + (1-w) f_{2e_j}) \leq 0$ $\forall x_{e_j} \in K(x_{e_i})$
  where $K(x_{e_i})$ is set of adjacent extreme point solutions of x_{e_i}. The two linear inequalities give the lower and upper bound for the value w, except for the two tail solutions on the convex-hull which have 0 and 1 as one of the bound.
- The exact value of ICP(X) is computed based on the equation (2). For instance, the exact formula of ICP(X) with uniform distribution function is:
  IPF(X) = $\sum_{i=1}^{m} [\int_{w_{i-1}}^{w_i} g(x_{e_i}; w)\, dw] = \int_0^{w_1} (w f_{1e_1} + (1-w) f_{2e_1})\, dw + \int_{w_1}^{w_2} (w f_{1e_2} + (1-w) f_{2e_2})\, dw + ... + \int_{w_{m-1}}^{1} (w f_{1e_m} + (1-w) f_{2e_m})\, dw$

Figure 2 shows an example with $X = \{x_1, x_2, ..., x_7\}$ and $E_X = \{x_1, x_2, x_3\}$. Each extreme solution x_i $(i = 1, 2, 3)$ is optimal on the range $[w_{i-1}, w_i]$.

The following property illustrates the relationship between two set of solutions with respect to ICP value.

Property 3: Given $S_1, S_2 \subseteq X$. If $\forall x \in S_1, \exists y \in S_2$ such that x is dominated by y, then ICP(S_2) $\leq$ ICP(S_1).

The property implies that ICP(E_X) $\leq S, \forall S \subseteq X$. As an example, let $S_1 = \{x_4, x_5, x_7\}$ and $S_2 = \{x_1, x_4, x_5, x_6\}$ in Figure 2, we have: $ICP(E_X) < ICP(S_2) < ICP(S_1)$.

As a consequence of this property, the following corollary can be considered the basis for the belief of the success for our hill-climbing approach to find a good set of plans in the next section.

Corollary 1: If x is a solution not dominated by any existing solution in X, then adding x into X never increases the value of ICP(X). In other words, ICP(X) is monotonically nonincreasing over increasing sequences of solution set.

Problem description and assumption

In this paper, we are interested in supporting the user to select their desired plan using *a posteriori* approach, based on the following assumptions:

- The user concerns about two objective functions *time* and *cost* of executing a plan.

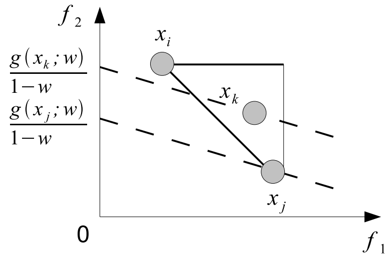

Figure 1: Either x_i or x_j is better than x_k for all trade-off values between two objectives.

[1] x_i is *dominated* by x_j iff $f_{1i} \leq f_{1j}$, $f_{2i} \leq f_{2j}$, and $(f_{1i}, f_{2i}) \neq (f_{1j}, f_{2j})$.

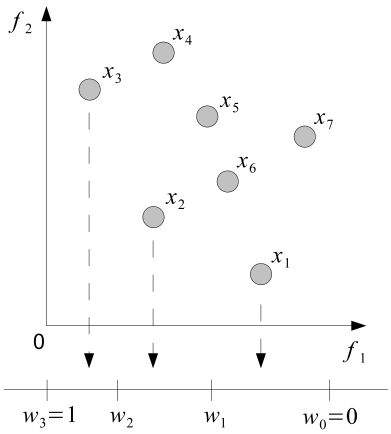

Figure 2: Extreme solutions x_1, x_2, x_3 of the set $X = \{x_1, ..., x_7\}$ are optimal on their own range $[w_{i-1}, w_i]$, and contribute to the computation of ICP(X).

- The user will evaluate a plan x using a convex combination value function of *time* and *cost*: $g(x; w) = wtime(x) + (1-w)cost(x)$, where $w \in [0, 1]$ is the fixed value of a particular user but is unknown to the system.
- The system is given a distribution of w over the range $[0, 1]$. For instance, if the user does not have any preference on the trade-off between *time* and *cost* of the plan, the distribution is the uniform function.

Given the trade-off preference between two objectives, or a particular w value, the best plan is one in the extreme plan set dominating all others on the portion $[a, b] \ni w$. Therefore, the ideal set of plans returned to the user must be the set of extreme plans E_X, whose ICP value is minimal (according to property 3). However, finding exactly E_X is too costly, and we will seek an approach to approximate it with a set of plans having as low ICP value as possible.

Solution Approach

We propose an approach working on Sapa planner to find our desired set of plans. Unlike Sapa, however, we will not incorporate each search node with one scalar function corresponding to a particular w value, but a set of $(time, cost)$ vectors, each of which represents a path from initial node to a goal node. This extension on the search of Sapa leads us to considering Multi-objective A* (Stewart and White 1991) which can find the set of all *Pareto-optimal* [2] plans with admissible heuristic. Our approach will control the search of Multi-objective A* so that we will stop with a desired set of plans, i.e. the one with as low ICP value as possible.

Multi-objective A*

Unlike original A* search algorithm, in Multi-objective A*, each node n in the search tree is associated with the following information:

- $G(n)$: set of non-dominated $(time, cost)$ vectors, each of which represents time and cost of the path from initial node to node n.
- $H(n)$: set of $(time, cost)$ vectors, each of which represents estimated time and cost of the path from node n to one of the nodes satisfying goal condition.
- $F(n)$: set of $(time, cost)$ vectors, each of which is the summation of one vector from $G(n)$ and one from $H(n)$.

Note that as we are comparing plans with respect to two objectives, there may be more than one path from initial node to some node n and none of them is dominated by another. Also, many paths may exist from current node n to goal nodes. These are why we need to keep in $G(n)$ and $H(n)$ set of $(time, cost)$ vectors.
The main steps of Multi-objective A* is summarized below:

1. $X = \emptyset$ (set of solution plans)
2. Initialize the $OPEN$ list with the node s_0 corresponding to the initial state.
3. Find the set of nodes $ND \subseteq OPEN$ such that there exists $v \in F(n)$ non-dominated by $(time, cost)$ vector of any solution path found and any node in $OPEN$.
4. If $ND = \emptyset$ then terminate.
   Otherwise:
   (a) Choose a node $n \in ND$, taking goals first, if any.
   (b) Move n from $OPEN$ to $CLOSED$.
5. If n is a goal node:
   - Remove any solution path found so far dominated by any new path from s_0 to n.
   - Add into X all solution paths from s_0 to n.
   - Go to step 3.

   Otherwise:
   (a) Expand n.
   (b) For each successor node n' of n: compute its non-dominated $G(n')$, $H(n')$, $F(n')$ and move it to $OPEN$.
6. Go to step 3.

Incorporating ICP measure into Multi-objective A*

Given that Multi-objective A* eventually finds the whole set of Pareto-optimal plans if admissible heuristic is used, and due to its costly computation, we propose a hill-climbing approach to put a heuristic function on top of it to prioritize nodes to be selected. We will start the set X with two plans: one with the least time and one with the least cost, called x_t and x_c respectively, and the step 3 of Multi-objective A* will be modified to select non-dominated nodes such that we can decrease the ICP(X) value. And when the selected node n is a goal node not dominated by any solution in X, the corresponding solution plan will be added into X. The modified algorithm is shown below:

1. Invoke Sapa to find plans with the least time (x_t) and with the least cost (x_c). Initialize X with these two plans: $X = \{x_t, x_c\}$
2. Initialize the $OPEN$ list with the node s_0 corresponding to the initial state.

[2] A plan is Pareto-optimal if it is not dominated by any other plan.

3. Find the set of nodes $ND \subseteq OPEN$ such that there exists $v \in F(n)$ satisfying the following conditions:
   - v is non-dominated by $(time, cost)$ vector of any solution path found and any node in $OPEN$.
   - $ICP(X + x_v) < ICP(X)$, where x_v is a plan whose $(time, cost)$ vector is equal to v.
4. If $ND = \emptyset$ then terminate.
   Otherwise:
   (a) Choose a node $n \in ND$, taking goals first, if any.
   (b) Move n from $OPEN$ to $CLOSED$.
5. If n is a goal node:
   - Remove any solution path found so far dominated by any new path from s_0 to n.
   - Add into X all solution paths from s_0 to n.
   - Go to step 3.

   Otherwise:
   (a) Expand n.
   (b) For each successor node n' of n: compute its non-dominated $G(n')$, $H(n')$, $F(n')$ and move it to $OPEN$.
6. Go to step 3.

To find x_t and x_c, we simply invoke Sapa with the scalar functions $f(x) = time(x)$ and $f(x) = cost(x)$ respectively. As one of the objectives of x_t, x_c is minimal, they are two Pareto-optimal plans. More interestingly, starting the search with these two plans helps us to narrow down the feasible plan space significantly. Figure 3 illustrates the feasible solution region after x_t and x_c are found. It also show that in this region, $ICP(\{x_t, x_3, x_4, x_5, x_c\}) < ICP(\{x_t, x_6, x_7, x_c\})$ according to property 3, and some dominated plans based on property 1 and 2.

Experiment Results

Given that the distribution of parameter w is known, and Sapa finds plans using the single scalar function $f(x) = wtime(x) + (1 - w)cost(x)$ with the fixed value of w, a natural approach raises to retrieve a set of plans taking into account the user preference model. The approach is based on sampling method and is described as follows:

- Generate N values $w_1, w_2, ..., w_N$ based on the distribution.
- For each w_i, invoke Sapa to find a plan with the objective function $f(x) = w_i time(x) + (1 - w_i)cost(x)$. Let the resulting plans be $P = \{p_1, p_2, ..., p_N\}$.
- Remove any dominated plans from P, and let the resulting set be P'.

Let t_1 minute(s) be the time Sapa with sampling runs to find P'.

Now assuming that our approach runs and find a set of plans $Q = \{q_1, q_2, ..., q_M\}$, and it takes t_2 minute(s). We will compare our method with Sapa with sampling in two dimensions:

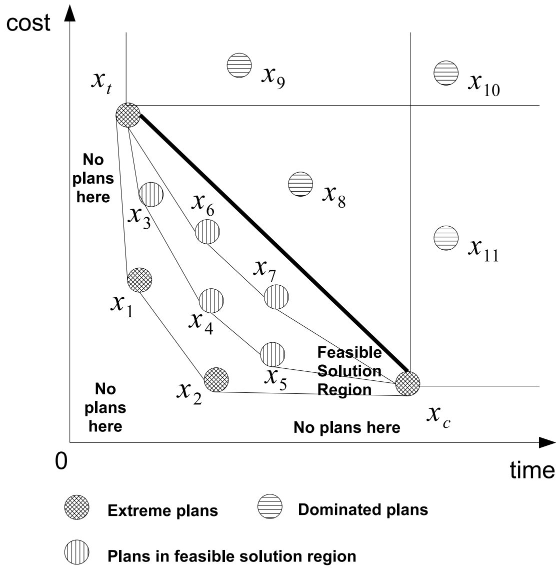

Figure 3: Feasible solution region of the problem. $\{x_t, x_3, x_4, x_5, x_c\}$ approximates the true set of extreme plans more accurately than $\{x_t, x_6, x_7, x_c\}$ does. x_8 is worse than either x_t or x_c based on property 2. And while x_9, x_{11} are dominated by x_t, x_c respectively, x_{10} is dominated by both of them.

- *How well the two approaches satisfy the user*: we simulate T transactions between the system and user. For each transaction, we generate a w value based on the distribution. Sapa with sampling and our approach will return plans p, q respectively in P' and Q optimal corresponding to the value w. We can check which approach is better by comparing two values $wtime(p) + (1 - w)cost(p)$ and $wtime(q) + (1 - w)cost(q)$.
- *The average time to find one plan in the set*: $\frac{t_1}{|P'|}$ v.s $\frac{t_2}{|Q|}$.

Using experimental results, we would like to illustrate the followings: (1) Sapa with sampling is worse than our approach in supporting the user with quality plan, and (2) the average time to find one plan of our approach is faster.

Conclusion and Future works

In this paper, we have proposed an approach based on heuristic search to find a set of plans supporting user with incomplete preference model. We extend Sapa to work on the preference model that is a convex combination of *time* and *cost* of executing a plan, one of the most frequently used preference model. By putting a heuristic measured by ICP value on top of Multi-objective A*, we illustrated that our approach most of the time generates more quality plan to the user and faster as well than Sapa with sampling, a method simply using Sapa without improvement to find set of plans.

For future works, we will continue with the following directions:

- Incorporate qualitative preference into the model, for instance the user prefer to visit some place in a travel domain.
- From one view point, our work can be consider as an extension of Over-Subscription Planning which aims to find

plan with the best net-benefit defined as: $net - benefit = utility - cost$. In our current work, the utility is fixed to the utility of all goals, but unlike previous works on this area, we consider $cost$ as not only the *cost* but also the *time* a user needs to pay for executing actions. Therefore, a natural extension is that all goals now can not be achieved. In this case, *utility* becomes an objective function of the plan.

References

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.

W. Matthew Carlyle, John W. Fowler, Esma S. Gel, and Bosun Kim. Quantitative comparison of approximate solution sets for bi-criteria optimization problems. *Decision Sciences*, 34(1), 2003.

M.B. Do and S. Kambhampati. Sapa: A scalable multi-objective metric temporal planner. *Journal Artificial Intelligence Research*, 20:155–194, 2004.

John W. Fowler, Bosun Kim, W. Matthew Carlyle, Esma S. Gel, , and Shwu-Min Horng. Evaluating solution sets of a posteriori solution techniques for bi-criteria combinatorial optimization problems. *Journal of Scheduling*, 8, 2005.

S. Kambhampati. Model-lite planning for the web age masses: The challenges of planning with incomplete and evolving domain theories. In *AAAI*, 2007.

I. Refanidis and I. Vlahavas. Multiobjective heuristic state-space planning. *Artificial Intelligence*, 145, 2003.

B. Srivastava, S. Kambhampati, T.A. Nguyen, M.B. Do, A. Gerevini, and I. Serina. Domain-independent approaches for finding diverse plans. In *International Joint Conference on Artificial Intelligence*, 2007.

S. Stewart and Chelsea C. White. Multi-objective a*. *Journal of the ACM*, 38, 1991.

Paolo Viappiani, B. Faltings, and P. Pu. Preference-based search using example-critiquing with suggestions. *Journal of Artificial Intelligence Research*, 27, 2006.

S. Yoon and S. Kambhampati. Towards model-lite planning: A proposal for learning & planning with incomplete domain models. In *ICAPS 2007 Workshop on AI Planning and Learning*, 2007.