A Fast Goal Recognition Technique Based on Interaction Estimates

Yolanda E-Martín[1,2] and **María D. R-Moreno**[1] and **David E. Smith**[3]
[1] Departamento de Automática. Universidad de Alcalá. Ctra Madrid-Barcelona,
Km. 33,6 28871 Alcalá de Henares (Madrid), Spain.
{yolanda,mdolores}@aut.uah.es
[2] Universities Space Research Association. 615 National Ave, Suite 220, Mountain View, CA 94043
[3] Intelligent Systems Division. NASA Ames Research Center. Moffett Field, CA 94035-1000
david.smith@nasa.gov

Abstract

Goal Recognition is the task of inferring an actor's goals given some or all of the actor's observed actions. There is considerable interest in Goal Recognition for use in intelligent personal assistants, smart environments, intelligent tutoring systems, and monitoring user's needs. In much of this work, the actor's observed actions are compared against a generated library of plans. Recent work by Ramírez and Geffner makes use of AI planning to determine how closely a sequence of observed actions matches plans for each possible goal. For each goal, this is done by comparing the cost of a plan for that goal with the cost of a plan for that goal that includes the observed actions. This approach yields useful rankings, but is impractical for real-time goal recognition in large domains because of the computational expense of constructing plans for each possible goal. In this paper, we introduce an approach that propagates cost and interaction information in a plan graph, and uses this information to estimate goal probabilities. We show that this approach is much faster, but still yields high quality results.

1 Introduction

Goal recognition aims to infer an actor's goals from some or all of the actor's observed actions. It is useful in many areas such as intelligent personal assistants [Weber and Pollack, 2008], smart environments [Wu *et al.*, 2007], monitoring user's needs [Pollack *et al.*, 2003; Kautz *et al.*, 2002], and for intelligent tutoring systems [Brown *et al.*, 1977]. Several different techniques have been used to solve plan recognition problems: hand-coded action taxonomies [Kautz and Allen, 1986], probabilistic belief networks [Huber *et al.*, 1994], consistency graphs [Lesh and Etzioni, 1995], Bayesian inference [Albrecht *et al.*, 1997; Bui, 2003], machine learning [Bauer, 1998], parsing algorithms [Geib and Goldman, 2009], and more recently AI planning. In particular, Jigui and Minghao [2007] developed Incremental Planning Recognition (IPR) that is based on reachability in a plan graph. Ramírez and Geffner [2009] developed an approach that identifies goals where the observed actions are compatible with an optimal plan for one or more of those goals. To identify those plans, they used optimal and satisficing planners, as well as a heuristic estimator [Keyder and Geffner, 2007], which approximates the solution by computing a relaxed plan. The limitation of this work is the assumption that agents act optimally – suboptimal plans compatible with the given observations are not considered. To remove this limitation, they developed an approach for estimating the probability of each possible goal, given the observations [2010; 2012]. The likelihood of a goal given a sequence of observations is computed using the cost difference between achieving the goal complying with the observations, and not complying with them. This cost is computed by means of two calls to a planner for each possible goal.

A significant drawback to the Ramírez approach is the computational cost of calling a planner twice for each possible goal. This makes the approach impractical for real-time goal recognition, such as for a robot observing a human and trying to assist or avoid conflicts. In this paper we present an approach that can quickly provide a probability distribution over the possible goals. This approach makes use of the theoretical framework of Ramírez, but instead of invoking a planner for each goal, it computes cost estimates using a plan graph. These cost estimates are more accurate than usual because we use interaction [Bryce and Smith, 2006]. Moreover, we can prune the cost-plan graph considering the observed actions sequence. Consequently, we can quickly compute cost estimates for goals with and without the observations, and thus infer a probability distribution over those goals. We show that this approach is much faster, but still yields high quality results.

In the next two sections we review the basic notions of planning and goal recognition from the perspective of planning. In Section 4 we develop our fast technique for approximate goal recognition, which includes the propagation of cost and interaction information through a plan graph, and a plan graph pruning technique from IPR. In Section 5 we present an empirical study, and in Section 6 we discuss future work.

2 Planning Background

Automated planning is the problem of choosing and organizing a sequence of actions that when applied in a given initial state results in a goal state. Formally, a STRIPS planning problem is defined as a tuple $\Pi =< P, O, I, G >$ where P is a finite set of propositional state variables; O is a set of operators, each having the form $< prec(O), add(O), del(O) >$ where $prec(O) \subseteq P$ is the set of propositions that must be satisfied before the operator can be executed; $add(O) \subseteq P$ is the set of positive propositions that become true when the operator is applied; $del(O) \subseteq P$ is the set of propositions that become false when the operator is applied; $I \subseteq P$ is the initial state; $G \subseteq P$ is the goal state. A solution or *plan* for a planning problem is a sequence of actions $\pi = (o_1, ..., o_n)$, which represents the path to reach G starting from I.

Each action $a \in O$ has an associate cost $Cost_a > 0$. Thus, a plan solution cost is the sum of the cost of the operators in the plan, i.e., $Cost(\pi) = \sum_{a \in \pi} Cost_a$. The optimal plan cost, $Cost^*(\pi)$, is the minimum cost among all the possible plans.

3 Goal Recognition Background

Ramírez (2010; 2012), defines a goal recognition problem to be a tuple $T = \langle P, \mathcal{G}, O, Pr \rangle$ where P is a planning domain and initial conditions, $\mathcal{G}$ is a set of possible goals or hypotheses, O is the observed action sequence $O = o_1, ..., o_n$, and Pr is the prior probability distribution over the goals in $\mathcal{G}$. The solution to a plan recognition problem is a probability distribution over the set of goals $G \in \mathcal{G}$ giving the relative likelihood of each goal. These posterior goal probabilities $P(G|O)$ can be characterized using Bayes Rule as:

$$Pr(G|O) = \alpha \, Pr(O|G) \, Pr(G) \tag{1}$$

where α is a normalizing constant, $Pr(G)$ is the prior distribution over $G \in \mathcal{G}$, and $Pr(O|G)$ is the likelihood of observing O when the goal is G. Ramírez goes on to characterize the likelihood $Pr(O|G)$ in terms of cost differences for achieving G under two conditions: complying with the observations O, and not complying with the observations O. More precisely, Ramírez characterizes the likelihood, $Pr(O|G)$, in terms of a Boltzman distribution:

$$Pr(O|G) = \frac{e^{[-\beta \, \Delta(G,O)]}}{1 + e^{[-\beta \, \Delta(G,O)]}} \tag{2}$$

where β is a positive constant and $\Delta(G, O)$ is the cost difference between achieving the goal with and without the observations:

$$\Delta(G, O) = Cost(G|O) - Cost(G|\overline{O}) \tag{3}$$

Putting equations (1) and (2) together yields:

$$Pr(G|O) = \alpha \, \frac{e^{[-\beta \, \Delta(G,O)]}}{1 + e^{[-\beta \, \Delta(G,O)]}} \, Pr(G) \tag{4}$$

By computing $\Delta(G, O)$ for each possible goal, equation 4 can be used to compute a probability distribution over those goals. The two costs necessary to compute Δ can be found by optimally solving the two planning problems $G|O$ and $G|\overline{O}$. Ramírez shows how the constraints O and $\overline{O}$ can be compiled into the goals, conditions and effects of the planning problem so that a standard planner can be used to find plans for $G|O$ and $G|\overline{O}$.

To illustrate the Ramírez approach, consider the example shown in Figure 1, where an agent can move *up*, *left*, and *right* at cost 1. It has two possible goals, G_1 and G_2, and $O = (o_1)$ as the observed sequence. For goal G_1, $Cost(G_1|O) = 3$, and $Cost(G_1|\overline{O}) = Cost(G_1) = 3$. (The costs are the same since o_1 is on an optimal path to G_1 and there is another optimal path that reaches G_1 but does not include o_1.) Hence, $\Delta(G_1, O) = 0$, and $Pr(G_1|O) = 0.5$. In contrast, $Cost(G_2|O) = 2$ and $Cost(G_2|\overline{O}) = 4$, since avoiding o_1 requires a suboptimal plan for G_2. This results in $\Delta(G_2, O) = -2$, and $Pr(G_2|O) = 0.88$. This means that G_2 is more likely to occur than G_1.

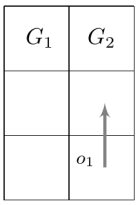

Figure 1: A grid examples for goals G_1 and G_2

This result is somewhat counterintuitive. The fact that the sequence of observed actions consists exclusively of optimal action landmarks for G_2 causes $Cost(G_2|\overline{O})$ to differ from $Cost(G_2)$ yielding a higher probability for G_2. However, In most cases, $Cost(G|\overline{O}) = Cost(G)$ because there are multiple possible paths to a goal. In order for the two costs to be different, the observation sequence would need to consist exclusively of optimal landmarks (like the case for G_2 above). For example, if we were to shift G_1 and G_2 left one column there would now be multiple optimal paths to G_2 so $Cost(G_2|\overline{O})$ would equal $Cost(G_2)$. This would result in both $\Delta(G_1, O) = 0$ and $\Delta(G_2, O) = 0$ yielding the same probability for both goals.

4 Fast Goal Recognition

The major drawback to the Ramírez approach is the computational expense of finding two plans for every possible goal. Moreover, the constraints O and $\overline{O}$ make the planning problems more difficult to solve. As a result, even for relatively simple Logistics and Blocks World problems it can take 15-30 minutes to find all the plans necessary for this computation. This makes it impractical to use this approach for any sort of real-time goal recognition. To address this problem, we developed a heuristic approach that combines two principal ideas: 1) cost and interaction estimates using a plan graph, and 2) the pruning technique of IPR. As with Ramírez, we assume that we are given $T = \langle P, \mathcal{G}, O, Pr \rangle$, which includes the problem P (planning domain and initial conditions), the set of possible goals G, a set of observations O, and a prior

distribution over the possible goals G. It is also assumed that the sequence of observed actions may be incomplete, but is accurate (not noisy).

4.1 Plan Graph Cost Estimation

Simple propagation of cost estimates in a plan graph is a technique that has been used in a number of planning systems to do heuristic cost estimation (e.g: [Do and Kambhampati, 2002]). Unfortunately, the cost information computed this way is often inaccurate because it does not consider the interaction between different actions and subgoals.

Cost Interaction is a quantity that represents how more or less costly it is that two propositions or actions are established together instead of independently. Formally, the optimal Interaction, I^*, is an n-ary interaction relationships among propositions and among actions (p_0 to p_n) in the plan graph, and it is defined as:

$$I^*(p_0, ..., p_n) = cost^*(p_0 \wedge ... \wedge p_n) - (cost^*(p_0) + ... + cost^*(p_n)) \quad (5)$$

where the term $cost^*(p_0 \wedge ... \wedge p_n)$ is the minimum cost among all the possible plans that achieve all the members in the set. Computing I^* would be computationally prohibitive. As a result, we limit the calculation of these values to pairs of propositions and pairs of actions in each level of a plan graph – in other words, binary interaction:

$$I^*(p, q) = cost^*(p \wedge q) - (cost^*(p) + cost^*(q)) \quad (6)$$

A value $I < 0$ means that two propositions or actions are synergistic – that is, the cost of establishing both is less than the sum of the costs of establishing the two independently. When $I = 0$ the two propositions or actions are independent. When $I > 0$ the two propositions or actions interfere with each other, so it is harder to achieve them both than to achieve them independently. The extreme value $I = \infty$, indicates that the two propositions or actions are mutually exclusive. One can therefore think of interaction as a more nuanced generalization of the notion of mutual exclusion.

The computation of cost and interaction information begins at level zero of a plan graph and proceeds sequentially to higher levels. For level zero we assume 1) the cost for propositions at this level is 0 because the initial state is given, and 2) the interaction between each pair of propositions is 0, that is, the propositions are independent. With these assumptions, we start the propagation by computing the cost of the actions at the first level of the plan graph. In general, for an action a at level l with a set of preconditions $\mathcal{P}_a$, the cost is approximated as:

$$cost^*(a) = cost^*(\mathcal{P}_a) \approx \sum_{x_i \in \mathcal{P}_a} cost(x_i) + \sum_{\substack{(x_i, x_j) \in \mathcal{P}_a \\ j > i}} I(x_i, x_j) \quad (7)$$

The next step is to compute the interaction between two actions. The interaction between two actions a and b at level l is:

$$I^*(a, b) = \begin{cases} \infty & \text{if } a \text{ and } b \text{ are mutex by inconsistent effects or interference} \\ Cost^*(a \wedge b) - Cost^*(a) - Cost^*(b) & \text{otherwise} \end{cases}$$

where $cost^*(a \wedge b)$ is defined to be $cost(\mathcal{P}_a \cup \mathcal{P}_b)$, which is approximated as in (7) by:

$$cost(\mathcal{P}_a \cup \mathcal{P}_b) \approx \sum_{x_i \in \mathcal{P}_a \cup \mathcal{P}_b} cost(x_i) + \sum_{\substack{(x_i, x_j) \in \mathcal{P}_a \cup \mathcal{P}_b \\ j > i}} I(x_i, x_j)$$

If the actions are mutex by inconsistent effects, or interference, then the interaction is ∞. Otherwise, the interaction above simplifies to:

$$I(a, b) \approx \sum_{\substack{x_i \in \mathcal{P}_a - \mathcal{P}_b \\ x_j \in \mathcal{P}_b - \mathcal{P}_a}} I(x_i, x_j) - \Bigg[\sum_{x_i \in \mathcal{P}_a \cap \mathcal{P}_b} cost(x_i) + \sum_{\substack{(x_i, x_j) \in \mathcal{P}_a \cap \mathcal{P}_b \\ j > i}} I(x_i, x_j) \Bigg]$$

For a proposition x at level l, achieved by the actions $\mathcal{A}_x$ at the preceding level, the cost is calculated in the same way as for traditional plan graph cost estimation:

$$Cost^*(x) = \min_{a \in \mathcal{A}(x)} [Cost(a) + Cost_a] \quad (8)$$

where $\mathcal{A}_x$ is the set of actions that support x.

Finally, in order to compute the interaction between two propositions at a level l, we need to consider all possible ways of achieving those propositions at the previous level. That is, all the actions that achieve the pair of propositions and the interaction between them. Suppose that $\mathcal{A}_x$ and $\mathcal{A}_y$ are the sets of actions that achieve the propositions x and y at level l. The interaction between x and y is then:

$$\begin{aligned} I^*(x, y) &= cost^*(x \wedge y) - cost^*(x) - cost^*(y) \\ &= \min_{\substack{a \in \mathcal{A}_x \\ b \in \mathcal{A}_y}} \Big\{ cost^*(a \wedge b) \Big\} - cost^*(x) - cost^*(y) \\ &\approx \min \left\{ \begin{array}{l} \min\limits_{a \in \mathcal{A}_x \cap \mathcal{A}_y} cost(a) + Cost_a \\ \\ \min\limits_{\substack{a \in \mathcal{A}_x - \mathcal{A}_y \\ b \in \mathcal{A}_y - \mathcal{A}_x}} \left[\begin{array}{l} cost(a) + Cost_a + \\ cost(b) + Cost_b + \\ I(a, b) \end{array} \right] \end{array} \right\} \\ &\quad - cost(x) - cost(y) \end{aligned} \quad (9)$$

Using equations (7), (8), and (9), a cost-plan graph is built until quiescence. On completion, each possible goal proposition has an estimated cost of achievement, and there is an interaction estimation between each pair of goal propositions. Using this information, one can estimate the cost of achieving a conjunctive goal $G = g_1, ..., g_n$ as:

$$Cost(G) \approx \sum_{i=1}^{n} \left[Cost(g_i) + \sum_{j<i} I(g_i, g_j) \right] \quad (10)$$

Using this information, we can estimate the cost of a plan for each possible goal G. While this allows us to estimate $Cost(G)$, what we need for goal recognition is to compute $\Delta(G, O)$, which requires $Cost(G|O)$ and $Cost(G|\overline{O})$. As discussed earlier, unless O is a subsequence of every optimal plan for G, $Cost(G|\overline{O}) = Cost(G)$. Even when this is not the case, these two costs tend to be similar except in a few rare cases. (This is confirmed in our experiments; for most problems there are multiple distinct optimal plans for each goal.) As a result, we approximate $Cost(G|\overline{O})$ by $Cost(G)$, which can be estimated as shown above. To estimate $Cost(G|O)$ we modify the plan graph as described in the next section, and repropagate cost and interaction information.

4.2 Incremental Plan Recognition

Jigui and Minghao [2007] developed a framework for plan recognition that narrows the set of possible goals by incrementally pruning a plan graph as actions are observed. The approach consists of building a plan graph to determine which actions and which propositions are true (1), false (-1), or unknown (0) given the observations. For level zero, since it is assumed that the initial state is true, every proposition has value 1. In addition, when an action is observed at a level it gets value 1. The process incrementally builds a plan graph and updates it level by level. The values of propositions and actions are updated according to the following rules:

1. An action in the plan graph gets value -1 when any of its preconditions or any of its effects is -1.
2. An action in the plan graph gets value 1 when it is the sole producer of an effect that has value 1, `noop` included.
3. A proposition in the plan graph gets value -1 when all of its consumers or all of its producers are -1, `noop` included.
4. A proposition in the plan graph gets value 1 when any of its consumers or any of its producers is 1, `noop` included.

The process results in a plan graph where each proposition and each action is labeled as 1, -1, or 0. Those propositions and actions identified as -1 can be ignored for plan recognition purposes, meaning that these are pruned from the resulting plan graph. To illustrate this propagation and pruning technique, consider a simple problem with three operators:

$$\begin{aligned} A &: y \rightarrow z \\ B &: y \rightarrow, \neg y, t \\ C &: t \rightarrow k, \neg t \end{aligned} \quad (11)$$

Suppose that the sequence of observed actions is A at level 0, and C at level 2. Figure 2 shows the plan graph for this problem. The numbers above the propositions and actions are the values for each proposition and action computed using the above propagation rules. As a result of the propagation, z must be true (has value 1) at level 1 because action A was observed. As a result, since A and B are mutually exclusive, action B and its effects t and $\neg y$ are false (have value -1) at level 0. As a consequence of t being false, C is also false at level 1 along with its effects k and $\neg t$. At level 2, k and $\neg t$ must be true because action C was observed. (This results in t being true at level 1.) Since A and C, and B and C are mutually exclusive, A, B, $\neg y$, and t are not possible (have value -1) at level 2. Action B is unknown (has value 0) at level 1 since there is not enough information to determine whether it is true or false. The proposition y is true at level 0 since it is assumed that the initial state is true, and is unknown at level 1 because there is not enough information to determine whether it is true or false. However, it is false at level 2 due to the mutual exclusion between C and noop-y. Proposition z is true at every level since there are no operators in the domain that delete it.

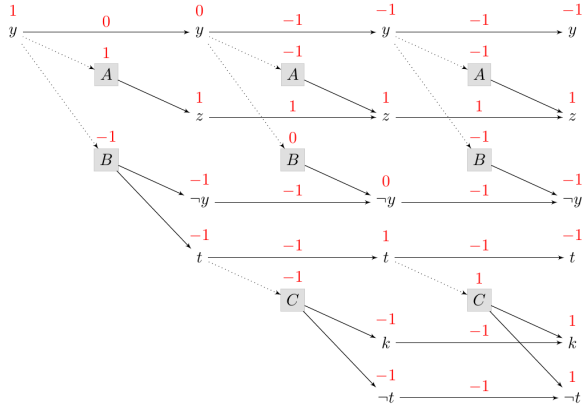

Figure 2: A plan graph with status values of propositions and actions

4.3 Relaxing the Time Step Assumption

We have modified the IPR pruning technique in order to relax the assumption of knowing the time step of each action in the observed sequence. Like Ramírez and Geffner [2010], we assume that the sequence of actions is sequential. Initially, we assign an *earliest time step* (*ets*) i to each action o in the observed sequence. The *ets* is given by the order of each action in the observed sequence. That is, given $(o_0, o_1, ..., o_i)$, the *ets* for each action is: *ets*(o_0)=0, *ets*(o_1)=1, *ets*(o_2)=2, etc. When the pruning process starts, we establish that an observed action o is possible to be observed at the assigned level i if all its preconditions are true (value 1) and/or unknown (value 0), and they are not mutually exclusive at level $i - 1$. Otherwise, the action cannot be executed at that level, which results in an update of the *ets* of each remaining action in the observed sequence. For instance, considering the initial sequence where *ets*(o_0)=0, *ets*(o_1)=1, *ets*(o_2)=2, and o_0 can be executed at level 0. If o_1 cannot be executed at level 1, then *ets*(o_1)=2 and *ets*(o_2)=3. If necessary, the cost-plan graph will be expanded until an *ets* is assigned to each observed action in the sequence. To illustrate this method, consider the example from Figure 2. Let us suppose the sequence of observed

actions is A and C, with initial *ets* 0 and 1 respectively. As a result of the propagation, z must be true (have value 1) at level 1 because action A was observed. As a result, since A and B are mutually exclusive, action B and its effects t and $\neg y$ are false (have value -1) at level 0. Action C is initially assumed to be at level 1, but this cannot be the case because its precondition t is false at level 0. Therefore, the *ets* for C is updated to 2. The result of this updating is that each observed action is assumed to occur at the earliest possible time consistent with both the observation sequence and the constraints found in constructing the plan graph, using the interaction information.

4.4 Computing Goal Probabilities

With the plan graph cost estimation technique described in Section 4.1, and the observation pruning technique described in Section 4.2 and Section 4.3, we can now put these pieces together to allow fast estimation of cost differences $\Delta(G, O)$, giving us probability estimates for the possible goals $G \in \mathcal{G}$. The steps are:

1. Build a plan graph for the problem P (domain plus initial conditions) and propagate cost and interaction information through this plan graph according to the technique described in Section 4.1.
2. For each (possibly conjunctive) goal $G \in \mathcal{G}$ estimate the $Cost(G)$ from the plan graph using equation (10).
3. Prune the plan graph, based on the observed actions O, using the technique described in Section 4.2 and Section 4.3.
4. Compute new cost and interaction estimates for this pruned plan graph, considering only those propositions and actions labeled 0, or 1.
5. For each (possibly conjunctive) goal $G \in \mathcal{G}$ estimate the $Cost(G|O)$ from the cost and interaction estimates in the pruned plan graph, again using equation (10). The pruned cost-plan graph may discard propositions or/and actions in the cost-plan graph necessary to reach the goal. This constraint provides a way to discriminate possible goals. However, it may imply that 1) the real goal is discarded, 2) the calculated costs are less accurate. Therefore, computation of $Cost(G|O)$ has been developed under two strategies:
   (a) $Cost(G|O)$ is computed using the pruned cost-plan graph.
   (b) $Cost(G|O)$ is computed after the pruned cost-plan graph is expanded to quiescence again. This will reintroduce any pruned goals that are still possible given the observations.
6. For each goal $G \in \mathcal{G}$, compute $\Delta(G, O)$, and using equation (4) compute the probability $Pr(G|O)$ for the goal given the observations.

To illustrate this computation, consider again the actions A, B, and C from equation (11), and the plan graph shown in Figure 2. Suppose that A, B, and C have costs 2, 1, and 3 respectively, and that the possible goals are $g_1 = \{z, k\}$ and $g_2 = \{z, t\}$. Propagating cost and interaction information through the plan graph, we get $Cost(t) = 1$, $Cost(z) = 2$, $Cost(k) = 4$, and interaction values $I(k, t) = \infty$ and $I(k, z) = 0$ at level 3. Now consider the hypothesis $g_1 = \{z, k\}$; in order to compute $Cost(k \wedge z)$, we use the cost and interaction information propagated through the plan graph. In order to compute $Cost(k \wedge z|O)$, the cost and interaction information is propagated again only in those actions with status 1 and 0. In our example, these costs are:

$$Cost(k \wedge z) \approx Cost(z) + Cost(k) + I(k, z) = 2 + 4 + 0 = 6$$

$$Cost(k \wedge z|O) \approx Cost(k) + Cost(z) + I(k, z) = 4 + 2 - 1 = 5$$

Thus, the cost difference is:

$$\Delta(g_1, O) = Cost(g_1|O) - Cost(g_1) = 5 - 6 = -1$$

As a result:

$$Pr(O|g_1) = \frac{e^{-(-1)}}{1 + e^{-(-1)}} = 0.73$$

For the hypothesis $g_2 = \{z, t\}$, the plan graph dismisses this hypothesis as a solution because once the plan graph is pruned, propositions *t* and *z* are labeled as -1. Therefore:

$$Cost(k \wedge t|O) \approx Cost(k) + Cost(t) + I(k, t) = \infty$$

So:

$$Pr(O|g_2) = \frac{e^{-\infty}}{1 + e^{-\infty}} = \frac{0}{1} = 0$$

If we expand the pruned cost-plan graph until quiescence again, the solution is still the same because A and B are permanently mutually exclusive.

Assuming uniform priors, $Pr(G)$, after normalizing the probabilities, we get that $Pr(g_1|O) = 1$ and $Pr(g_2|O) = 0$, so the goal g_1 is certain in this simple example, given the observations of actions A and C.

5 Experimental Results

We have conducted an experimental evaluation on planning domains used by Ramírez and Geffner: BlocksWord, Intrusion, Kitchen, and Logistics. Each domain has 15 problems. The hypotheses set and actual goal for each problem were chosen at random with the priors on the goal sets assumed to be uniform. For each problem in each of the domains, we ran the LAMA planner [Richter *et al.*, 2008] to solve the problem for the actual goal. The set of observed actions for each recognition problem was taken to be a subset of this plan solution, ranging from 100% of the actions, down to 10% of the actions. The experiments were conducted on an Intel Xeon CPU E5-1650 processor running at 3.20GHz with 32 GB of RAM.

Ramírez evaluates his technique using an optimal planner HSP^*_f [Haslum, 2008], and LAMA, a satisficing planner that

Table 1: Summarized Evaluation Random Observations

Domain		Blocks					Campus					Easy Grid					Intrusion				
Approach	%O	100	70	50	30	10	100	70	50	30	10	100	70	50	30	10	100	70	50	30	10
HSP^*_fu	T	704.77	579.04	508.12	479.16	479.16	1.78	1.1	0.55	0.28	0.28	224.67	144.49	79.82	41.47	37.42	588.34	414.29	214.19	4.25	4.22
	Q	1	1	1	1	1	1	1	1	1	1	1	0.2	0.66	0.86	0.73	1	1	1	1	1
	S	1.06	1.13	4.06	11.46	11.46	1	1	1	1	1	1	1.06	1.93	3.6	3.93	1	1	1.06	4.46	4.6
gHSP$^*_f$u	T	558.08	419.45	379.81	357.94	357.94	1.22	0.8	0.5	0.32	0.32	114.12	79.76	52.92	39.2	38.8	447.41	281.12	151.37	3.58	3.55
	Q	1	1	1	1	1	1	1	1	1	1	1	0.2	0.8	0.93	0.8	1	1	1	1	1
	S	1.06	1.13	4.06	11.46	11.46	1	1	1	1	1	1	1.2	2.26	3.86	4.2	1	1	1.06	4.46	4.6
LAMA	T	1603.24	1522.96	1260.6	1077.62	1082.15	1.75	1.38	0.8	0.62	0.52	13.93	13.19	11.66	10.38	10.36	92.85	89.01	64.97	17.57	17.48
	Q	0.33	0.8	0.8	0.93	1	1	1	1	1	0.4	0.53	0.6	0.66	0.73	0.93	1	1	1	1	1
	S	1	1.13	3.86	10.4	10.93	1	1	1	1	1	2	1.93	1.8	2.53	2.53	0.93	1	1.06	4.46	4.6
	Q_{20}	1	1	1	1	1	1	1	1	1	1	0.6	0.6	0.66	0.73	0.93	0.93	1	1	1	1
	Q_{50}	1	1	1	1	1	1	1	1	1	1	0.8	0.6	0.86	0.86	0.93	0.93	1	1	1	1
	d	0.24	0.316	0.192	0.048	0.068	$3.66{\cdot}10^{-6}$	$8{\cdot}10^{-5}$	$1.31{\cdot}10^{-3}$	0.054	$3.66{\cdot}10^{-6}$	0.021	0.016	0.011	$7.54{\cdot}10^{-3}$	0.018	1.739	1.12	0.095	$7{\cdot}10^{-6}$	$7{\cdot}10^{-6}$
LAMA$_G$	T	849.08	840.76	814.95	803.04	809.01	0.65	0.55	0.45	0.42	0.37	10.43	9.38	8.47	7.45	7.54	3.32	2.63	2.21	2.08	2.08
	Q	0.93	0.8	0.73	0.66	0.46	1	1	1	0.8	0.66	0.26	0.53	0.46	0.6	0.66	0	0.4	1	1	1
	S	1	1.2	3	6.2	4.2	1	1	1	1.06	1.06	1.66	1.66	1.73	2.13	2.13	0	0.4	1.13	4.46	4.6
	Q_{20}	0.93	1	1	1	1	1	1	1	0.8	0.66	0.53	0.6	0.66	0.73	0.8	0	0.4	1	1	1
	Q_{50}	0.93	1	1	1	1	1	1	1	1	1	0.8	0.6	0.86	0.86	0.8	0	0.4	1	1	1
	d	0.068	0.34	0.404	0.322	0.341	$2.7{\cdot}10^{-6}$	$4.8{\cdot}10^{-4}$	$7{\cdot}10^{-3}$	0.138	0.155	0.021	0.016	0.011	$6.74{\cdot}10^{-3}$	0.018	1.86	1.12	0.108	$7{\cdot}10^{-6}$	$7{\cdot}10^{-6}$
h^s_a	T	1.04	0.89	0.81	0.81	0.8	0.04	0.04	0.03	0.03	0.03	0.43	0.32	0.28	0.25	0.23	0.7	0.46	0.39	0.35	0.36
	Q	1	1	1	1	1	1	1	1	1	1	0.86	0.86	0.86	0.86	0.86	1	1	1	1	1
	S	20.26	20.26	20.26	20.26	20.26	2	2	2	2	2	5.33	5.33	5.33	5.33	5.33	16.66	16.66	16.66	16.66	16.66
	Q_{20}	1	1	1	1	1	1	1	1	1	1	0.86	0.86	0.86	0.86	0.86	1	1	1	1	1
	Q_{50}	1	1	1	1	1	1	1	1	1	1	0.86	0.86	0.86	0.86	0.86	1	1	1	1	1
	d	0.092	0.087	0.062	0.035	0.035	0.19	0.19	0.26	0.19	0.19	0.05	0.05	0.05	0.03	0.03	0.123	0.124	0.119	0.075	0.073
GR$_I$	T	1.007	1.029	1.265	1.452	1.452	0.27	0.29	0.35	0.39	0.39	108.49	110.11	97.24	98.8	95.21	0.885	0.493	0.212	0.209	0.21
	Q	1	0.66	0.4	0.13	0.13	1	1	0.93	0.93	0.93	1	0.13	0.6	0.86	0.66	1	1	0.93	0.93	0.93
	S	1.06	0.8	1.06	1.73	1.73	1	1	0.93	1.13	1.13	1	1.4	1.93	2.33	2.06	1	1	1	4.4	4.53
	Q_{20}	1	0.66	0.53	0.46	0.46	1	1	0.93	0.93	0.93	1	0.13	0.6	0.86	0.66	1	1	1	0.93	0.93
	Q_{50}	1	0.73	0.73	0.8	0.8	1	1	0.93	0.93	0.93	1	0.13	0.6	0.93	0.66	1	1	1	1	1
	d	0.149	0.962	0.751	0.336	0.336	$7.2{\cdot}10^{-7}$	0.12	0.108	0.038	$7.2{\cdot}10^{-7}$	0.001	0.082	0.149	0.084	0.001	$3.2{\cdot}10^{-4}$	0.055	0.872	0.241	0.241
GR$_I$E	T	9.936	8.211	4.542	3.696	3.687	0.39	0.44	0.43	0.43	0.43	124.15	124.93	110.36	115.47	100.46	1.293	1.191	0.996	0.743	0.738
	Q	0.46	0.53	0.46	0.13	0.13	1	1	0.93	0.93	0.93	1	0.13	0.6	0.86	0.66	1	1	0.93	0.93	0.93
	S	1.26	1.13	1.86	1.73	1.73	1	1	0.93	1.13	1.13	1	1.4	1.86	2.06	2.13	1	1	1	4.4	4.53
	Q_{20}	0.46	0.73	0.66	0.4	0.4	1	1	0.93	0.93	0.93	1	0.13	0.6	0.86	0.66	1	1	1	0.93	0.93
	Q_{50}	0.46	0.8	0.86	0.8	0.8	1	1	0.93	0.93	0.93	1	0.13	0.6	0.86	0.66	1	1	1	1	1
	d	1.094	1.025	0.742	0.358	0.358	$5.6{\cdot}10^{-7}$	$4.37{\cdot}10^{-5}$	0.017	0.038	$5.6{\cdot}10^{-7}$	0.001	0.082	0.152	0.077	0.001	$3.2{\cdot}10^{-4}$	0.055	0.872	0.241	0.241
GR$_{\neg I}$	T	0.761	0.609	0.643	0.782	0.783	0.19	0.2	0.23	0.25	0.25	33.15	33.6	33.29	35.52	35	0.886	0.491	0.196	0.192	0.193
	Q	1	0.46	0.46	0.46	0.46	1	0.33	0.53	0.93	0.93	0.93	0.13	0.53	0.73	0.6	1	1	1	1	0.93
	S	1	0.53	1.2	2.06	2.06	1.2	0.33	0.73	1.2	1.2	1.2	1.66	2.53	2.26	2.33	1	1.13	1.13	4.06	3.93
	Q_{20}	1	0.46	0.6	0.6	0.6	1	0.33	0.53	0.93	0.93	0.93	0.13	0.53	0.73	0.6	1	1	1	1	0.93
	Q_{50}	1	0.46	0.6	0.73	0.73	1	0.33	0.53	0.93	0.93	0.93	0.13	0.53	0.8	0.6	1	1	1	1	1
	d	0.088	1.084	1.036	1.004	1.004	$7.2{\cdot}10^{-7}$	0.08	0.093	0.133	$7.2{\cdot}10^{-7}$	0.013	0.068	0.148	0.084	0.013	0.151	0.764	1.25	0.311	0.285

is used in two modes: as a greedy planner that stops when it finds the first plan (LAMA$_G$), and as a planner that returns the best plan found in a given time limit (LAMA). For purposes of this test, Ramírez technique is also evaluated using the heuristic estimator h^s_a, which was used by Ramírez and Geffner (2009). Like our technique, this requires no search since the cost is given by a heuristic estimator. We compare our goal recognition technique, GR, against Ramírez's technique for those three planners and h^s_a, on the aforementioned domains, using a range of time limits from 5 seconds up to 1800 seconds. We present two variations of our technique, with and without extension of the plan graph after pruning:

- GR$_I$: the propagation of cost information through the plan graph considers interaction information.
- GR$_I$E: same as above, but the pruned cost-plan graph is expanded until quiescence.
- GR$_{\neg I}$: the propagation of cost information through the plan graph does not consider interaction information.

Table 1 and 2 show the results. For each planner, each column shows average performance over the 15 problems in each domain. The first row in the table represents the optimal solution where HSP$^*_f$ (HSP$^*_f$u) computes $\Delta(G, O)$ using equation (3). The second row in the table represents the optimal solution where HSP$^*_f$ (gHSP$^*_f$u) computes $\Delta(G, O)$ using $Cost(G)$ instead of $Cost(G|\overline{O})$ (Equation (3)). Both approaches were allowed to run for an unlimited amount of time. The other rows represent different measures of quality and performance:

- T shows the average time in seconds taken for solving the problems.
- Q shows the fraction of times the actual goal was among the goals found to be the most likely.
- S shows the *spread*, that is, the average number of goals in $\mathcal{G}$ that were found to be the most likely.
- Q_{20} and Q_{50} show the fraction of times the actual goal is in the top 20% and top 50% of the ranked goals. Although Q might be less than 1 for some problem, Q_{20} or Q_{50} might be 1, indicating that the actual goal was *close* to the top.
- d is the mean distance between the probability scores produced for all the goal candidates, and the probability scores produced by gHSP$^*_f$u. More precisely, if the set of possible goals is $\{G_1, ..., G_n\}$, a method produces probabilities $\{e_1, ..., e_n\}$ for those goals, and gHSP$^*_f$u produces $\{p_1, ..., p_n\}$, d is defined as:

$$d = 1/n \sum_{i=1}^{n} |e_i - p_i| \quad (12)$$

In all the domains except Kitchen and Logistics, HSP$^*_f$u finds the actual goal with highest probability ($Q = 1$). In Logistics, the value of Q degrades with lower percentages of observed actions. The *spread* increases as the percentage of observed actions decreases, because there is not enough information to distinguish among different possible goals. Results for HSP$^*_f$u are the same except for the Kitchen domain where gHSP$^*_f$u actually finds higher quality solutions. We note, however, that gHSP$^*_f$u is significantly faster than HSP$^*_f$u because it is generally easier to plan for G than for $G|\overline{O}$.

LAMA and LAMA$_G$ planners solve all the problems for the BlocksWord domain within 1800 seconds, which is actu-

ally slower than HSP^*_f. However, the quality Q is high, which means that the actual goal is among the most likely goals for most problems. In the Campus domain, LAMA solves all the problems within 1.5 seconds, and generates high quality solutions. $LAMA_G$ solves all the problems in less than a second, but the quality of the solution decreases as the percentage of observed actions drops. In the Grid domain, LAMA solves all the problems within the 13 seconds. The quality of the solution is low when the percentage of observed actions is high, and it improves as the percentage of observed actions drops. $LAMA_G$ solves all the problems in the Grid domain within 10 seconds, but the quality of the solution is lower than for LAMA. In the Intrusion domain, LAMA solves all the problems producing quality solutions within 90 seconds, when the percentage of observed actions is high. The computational time decreases to 18 seconds as the percentage of observed actions drops. $LAMA_G$ solves all the problems in the Intrusion domain within 3 seconds, but the quality of the solutions is very low when the percentage of observed actions is high. In the Kitchen domain, the quality of solutions given by LAMA and $LAMA_G$ varies depending on the percentage of observations. However, $LAMA_G$ is faster than LAMA. In the Logistics domains, LAMA solves all the problems, and produces quality solutions within 42 seconds as long as the percentage of observed actions is high. The computation time decreases to around 11 seconds as the percentage of observed actions drops. For $LAMA_G$ the problems are solved within 5 seconds. When the percentage of observed actions is high, the quality of the solution is good. (The quality degrades as the percentage of observed actions drops.)

The h^s_a heuristic solves all the problems within a second. In all the domain except Logistics, the quality Q of the solution is 1, which means that the actual goal is among the most likely goals for all the problems. However, the *spread* is very high, which means that the approach does not discriminate among the possible goals very well. In Logistics, the quality Q of the solution is very low, although in most cases, the actual goal appears in the top 20% of the ranked goals (Q_{20}).

GR_I solves all the problems, except those in the Grid domain, within 2 seconds, $GR_I E$ solves them within 10 seconds. In general, the three approaches quickly provide high quality solutions when the percentage of observed actions is high, although this degrades a bit as the percentage of observed actions drops. Considering the expansion of the pruned cost-plan graph, we expected $GR_I E$ to dominate GR_I in terms of goal recognition accuracy. Surprisingly, this does not appear to be the case in the domains studied. In some cases, $GR_I E$ shows a small improvement, but in others the quality of the solutions drops. Our hypothesis is that the pruned goals are sufficiently unlikely that reintroducing them does not significantly impact the resulting probability distribution. Considering the use of interaction estimates, GR_I gets more high quality solutions than $GR_{\neg I}$, except for some cases of BlocksWord and Logistics domains for 30% and 10% of observed actions. In the Grid domain, GR_I solves all the problem within 100 seconds and $GR_I E$ within 130 seconds, which is slower than HSP^*_f. However, for all the percentages of observed actions, except for 70%, both approaches provide almost the same quality solution as that provided by LAMA. $GR_{\neg I}$ solves all the problems in the Grid domain within the 35 seconds, but the quality is a bit lower than GR_I.

Table 2: Summarized Evaluation Random Observations

Domain		Kitchen					Logistics				
Approach	%O	100	70	50	30	10	100	70	50	30	10
HSP^*_fu	T	693.74	243.52	61.03	33.28	33.3	46.06	40.89	18.94	9.18	9.18
	Q	1	1	0.86	0.86	0.86	1	0.93	0.66	0.8	0.8
	S	1	1	1.2	1.26	1.26	1	1.13	2.26	3.6	3.6
gHSP$^*_f$u	T	480.51	171.08	49.62	37.93	37.92	36.26	32.46	14.08	7.04	7.04
	Q	1	1	1	1	1	1	0.93	0.66	0.8	0.8
	S	1	1	1.33	1.4	1.4	1	1.13	2.26	3.6	3.6
LAMA	T	26.33	26.2	20.45	20.3	20.3	45.17	41.71	26.53	11.38	11.39
	Q	0.73	1	1	1	1	1	0.93	0.66	0.8	0.8
	S	0.73	1	1.33	1.4	1.4	1	1.13	2.26	3.6	3.6
	Q_{20}	0.73	1	1	1	1	1	1	0.93	1	1
	Q_{50}	0.73	1	1	1	1	1	1	1	1	1
	d	0.358	3×10^{-3}	0.016	0.012	0.012	0.019	0.673	1.051	0.956	0.956
$LAMA_G$	T	0.42	0.36	0.33	0.32	0.32	5.47	4.95	4.5	4.33	4.36
	Q	0.73	1	1	1	1	1	0.8	0.4	0.46	0.46
	S	0.73	1	1.33	1.4	1.4	1	1.2	1.8	2.93	2.93
	Q_{20}	0.73	1	1	1	1	1	1	0.66	0.6	0.6
	Q_{50}	0.73	1	1	1	1	1	1	0.93	0.86	0.86
	d	0.358	3×10^{-3}	0.016	0.012	0.012	0.019	0.675	1.179	1.063	1.063
h^s_a	T	0.066	0.049	0.044	0.045	0.046	0.61	0.55	0.51	0.51	0.51
	Q	1	1	1	1	1	0.13	0.06	0.13	0.06	0.06
	S	3	3	3	3	3	2.8	1.86	1.93	1	1
	Q_{20}	1	1	1	1	1	0.93	0.93	0.86	0.86	0.86
	Q_{50}	1	1	1	1	1	0.93	0.93	0.86	0.86	0.86
	d	0.443	0.436	0.284	0.226	0.226	0.123	0.117	0.099	0.074	0.074
GR_I	T	0.261	0.195	0.140	0.135	0.135	0.886	1.015	1.19	1.266	1.271
	Q	1	1	1	1	1	1	0.86	0.53	0.6	0.6
	S	1	1	1.2	1.26	1.26	1	1.26	1.6	2.46	2.46
	Q_{20}	1	1	1	1	1	1	0.93	0.66	0.73	0.73
	Q_{50}	1	1	1	1	1	1	0.93	0.8	0.86	0.86
	d	3.98×10^{-4}	0.036	0.107	0.192	0.192	0.264	0.786	1.163	0.718	0.718
GR_IE	T	0.287	0.266	0.230	0.193	0.193	7.535	4.24	3.016	2.842	2.834
	Q	1	1	1	1	1	0.86	0.66	0.66	0.6	0.6
	S	1	1	1.2	1.26	1.26	1.13	1.4	1.8	2.8	2.8
	Q_{20}	1	1	1	1	1	0.86	0.8	0.8	0.73	0.73
	Q_{50}	1	1	1	1	1	0.93	1	0.86	0.86	0.86
	d	3.98×10^{-4}	0.036	0.107	0.192	0.192	0.387	0.943	1.107	0.771	0.771
$GR_{\neg I}$	T	0.258	0.191	0.136	0.128	0.129	0.806	0.405	0.448	0.508	0.51
	Q	1	1	1	1	1	1	0.4	0.46	0.6	0.6
	S	1	1	1.33	1.4	1.4	1	1	2.13	2.93	2.93
	Q_{20}	1	1	1	1	1	1	0.46	0.6	0.66	0.66
	Q_{50}	1	1	1	1	1	1	0.53	0.73	0.8	0.8
	d	2×10^{-6}	0.038	5×10^{-6}	0.022	0.022	0.011	1.196	1.29	0.899	0.899

We have conducted a second test where the set of observed actions for each recognition problem was considered to be the prefix of the plan solution, ranging from 100% of the actions, down to 10% of the actions. (Priors on the goal sets are also assumed to be uniform.) The reason for this test is to model incremental goal recognition as actions are observed. This produces essentially the same results as the previous test for all the different techniques. We have also tested the approaches in Table 1 on test problems where the observation sequences were generated using an optimal planner (rather than LAMA). This also makes no difference in the results.

6 Conclusions and Future Work

In this paper we presented a fast technique for goal recognition. This technique computes cost estimates using a plan graph, and uses this information to infer probability estimates for the possible goals. This technique provides fast, high quality solutions when the percentage of observed actions is relatively high, and degrades as the percentage of observed actions decreases. In many domains, this approach yields results two orders of magnitude faster than the Ramírez approach using HSP^*_f, LAMA, or greedy LAMA on the more difficult domains. The solutions are comparable or better in quality than those produced by greedy LAMA.

The ranking that we produce could be used to select the highest probability goals, and then feed this set to the Ramírez approach to provide further refinement to the probabilities. While this does reduce the amount of computation time require by the Ramírez approach, in our experiments the rank-

ing occasionally improved, and the computation time was still large for the more difficult domains and problems. When the top goals are fed to the Ramírez approach with the LAMA planner, the overall ranking occasionally improves for higher time limits. For low time limits, it doesn't improve or decreases the solution quality.

It appears possible to extend our approach to deal with temporal planning problems involving durative actions and concurrency. It is not clear whether the compilation technique of Ramírez would extend to temporal problems, but even if this were possible, the overhead of solving two temporal planning problems for each possible goal is likely to be prohibitive.

Acknowledgments

This work was funded by the Junta de Comunidades de Castilla-La Mancha project PEII-2014-015A, Universities Space Research Association, and NASA Ames Research Center.

References

[Albrecht *et al.*, 1997] D. Albrecht, I. Zukerman, A. Nicholson, and A. Bud. Towards a bayesian model for keyhole plan recognition in large domains. *In Proceedings of UM*, 1997. Chia Laguna, Sardinia.

[Bauer, 1998] M. Bauer. Acquisition of abstract plan descriptions for plan recognition. *In Proceedings of AAAI*, 1998. Madison, WI, USA.

[Brown *et al.*, 1977] J. S. Brown, R. Burton, and C Hausmann. Representing and using procedural bugs for educational purposes. *In Proceedings of ACM*, 1977. Amsterdam, The Netherlands.

[Bryce and Smith, 2006] D. Bryce and D. E. Smith. Using interaction to compute better probability estimates in plan graphs. *In ICAPS-06 Workshop on Planning Under Uncertainty and Execution Control for Autonomous Systems*, 2006. Lake District, United Kingdom.

[Bui, 2003] H. Bui. A general model for online probabilistic plan recognition. *In Proceedings of IJCAI*, 2003. Acapulco, Mexico.

[Do and Kambhampati, 2002] M. Do and R. Kambhampati. Planning graph-based heuristics for cost-sensitive temporal planning. *In Proceedings of AIPS*, 2002. Toulouse, France.

[Geib and Goldman, 2009] C. W. Geib and R. P. Goldman. A probabilistic plan recognition algorithm based on plan tree grammar. *In Artificial Intelligence*, 84(1-2):57—112, 2009.

[Haslum, 2008] P. Haslum. Additive and reversed relaxed reachability heuristics revisited. *In Proceedings of the 2008 IPC*, 2008. Sydney, Australia.

[Huber *et al.*, 1994] M. J. Huber, E. H. Durfee, and M. P. Wellman. The automated mapping of plans for plan recognition. *In Proceedings of UAI*, 1994. Seattle, WA, USA.

[Jigui and Minghao, 2007] S. Jigui and Y. Minghao. Recognizing the agent's goals incrementally: planning graph as a basis. *In Frontiers of Computer Science in China*, 1(1):26—36, 2007.

[Kautz and Allen, 1986] H. Kautz and J. F. Allen. Generalized plan recognition. *In Proceedings of AAAI*, 1986. Philadelphia, PA, USA.

[Kautz *et al.*, 2002] H. Kautz, L. Arnstein, G. Borriello, O. Etzioni, and D. Fox. An overview of the assisted cognition project. *In Proceedings of AAAI Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, 2002. Edmonton, Alberta, Canada.

[Keyder and Geffner, 2007] E. Keyder and H. Geffner. Heuristics for planning with action costs. In *Proceedings of the Spanish Artificial Intelligence Conference*, Salamanca, Spain, 2007.

[Lesh and Etzioni, 1995] N. Lesh and O. Etzioni. A sound and fast recognizer. *In Proceedings of IJCAI*, 1995. Montreal, Canada.

[Pollack *et al.*, 2003] M.E. Pollack, L. Brownb, D. Colbryc, C. E. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan, and I. Tsamardinos. Autominder: an intelligent cognitive orthotic system for people with memory impairment. *In Robotics and Autonomous Systems*, 44(3-4):273—282, 2003.

[Ramírez and Geffner, 2009] M. Ramírez and H. Geffner. Plan recognition as planning. *In Proceedings of IJCAI*, 2009. Pasadena, CA, USA.

[Ramírez and Geffner, 2010] M. Ramírez and H. Geffner. Probabilistic plan recognition using off-the-shelf classical planners. *In Proceedings of AAAI*, 2010. Atlanta, GA, USA.

[Ramírez, 2012] M. Ramírez. *Plan Recognition as Planning*. PhD thesis, Universitat Pompeu Fabra, 2012. Barcelona, Spain.

[Richter *et al.*, 2008] S. Richter, M. Helmert, and M Westphal. Landmarks revisited. *In Proceedings of AAAI*, 2008. Chicago, IL, USA.

[Weber and Pollack, 2008] J. S. Weber and M. E. Pollack. Evaluating user preferences for adaptive reminding. *In Proceedings of CHI EA*, 2008. Florence, Italy.

[Wu *et al.*, 2007] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. M. Rehg. A scalable approach to activity recognition based on object use. *In Proceedings of ICCV*, 2007. Rio de Janeiro, Brazil.