

Towards Quicker Probabilistic Recognition with Multiple Goal Heuristic Search

Richard G. Freedman Yi Ren Fung* Roman Ganchin* Shlomo Zilberstein

College of Information and Computer Sciences
University of Massachusetts, Amherst, MA 01003, USA
{freedman, rganchin, shlomo}@cs.umass.edu, yfung@umass.edu

Abstract

Referred to as an approach for either plan or goal recognition, the original method proposed by Ramírez and Geffner introduced a domain-based approach that did not need a library containing specific plan instances. This introduced a more generalizable means of representing tasks to be recognized, but was also very slow due to its need to run simulations via multiple executions of an off-the-shelf classical planner. Several variations have since been proposed for quicker recognition, but each one uses a drastically different approach that must sacrifice other qualities useful for processing the recognition results in more complex systems. We present work in progress that takes advantage of the shared state space between planner executions to perform multiple goal heuristic search. This single execution of a planner will potentially speed up the recognition process using the original method, which also maintains the sacrificed properties and improves some of the assumptions made by Ramírez and Geffner.

1 Introduction

Early research in plan recognition (Kautz 1991; Charniak and Goldman 1991) identified which plan was being executed given a description of what an agent was doing and a *plan library* containing a set of action sequences (plans) that solve a collection of predefined tasks. This ‘most-likely match’ formulation was later extended to plan recognition with hierarchical task network representations through an analogy with natural language parsing (Geib and Steedman 2007; Freedman, Jung, and Zilberstein 2014), which was similarly used for plan recognition over exploratory grammars (Amir and Gal 2013; Mirsky and Gal 2016; Mirsky, Gal, and Shieber 2017). However, the formulation developed for plan recognition with classical planning representations such as PDDL and STRIPS instead used a *plan domain* that generalized the plan library (Ramírez and Geffner 2009); a specified set of plans is not necessary for each task because the domain definition represents the set of all plans for each task. Related research has since labeled this work as goal recognition rather than plan recognition because it identifies the goal conditions/task description only—no specific plan is returned due to the lack of a

match in a plan library (Sohrabi, Riabov, and Udrea 2016; E-Martín, R-Moreno, and Smith 2015; Pereira, Oren, and Meneguzzi 2017).

The method Ramírez and Geffner initially proposed uses the domain definition to run any off-the-shelf classical planner and solve each recognizable goal in the list, both with and without requiring the observed actions to be in the solution. A probabilistic variation was proposed the following year (Ramírez and Geffner 2010), introducing ranking and a mathematical description of the recognition process for classical planning. However, both versions require running the chosen planner for each of these possible goals, which makes the process very slow because it can only solve one goal at a time.

Methods for speeding up the recognition process were inspired by this issue, but made modifications to the approach that introduce sacrifices to the mathematical intuition and accuracy in some cases. E-Martín, R-Moreno, and Smith propagate cost information through a planning graph and use the values of the nodes representing the goal conditions to approximate the least-costly plan (2015). Their results were typically faster, but were less accurate in some cases and actually slower in a few domains. Pereira, Oren, and Meneguzzi alternatively abstracted the problem from simulating the entire planning process to just simulation of the landmarks that must be accomplished no matter which plan is used to reach a satisfying goal state (2017). Their results performed far faster in every case, but the method is not yet able to provide the probabilistic distribution due to the use of a relative heuristic rather than actual plan cost. Furthermore, there are far fewer landmarks than states so that applications that need in-progress recognition (Freedman and Zilberstein 2017) will have the short-sighted bias of predicting the goals whose optimal plans are closest to completion. This is due to the partial observation sequences exclusively containing actions from the beginning of the execution sequence. Vered and Kaminka (2017) begin to address this for on-line recognition of motion planning using two heuristics that decide (1) whether to recompute as each new observation is made and (2) when to remove goals that are less likely from the list of possible goals. However, this approach also lacks a probabilistic distribution over the possible goals. A variation strictly designed for GRIDWORLD-style path planning domains found approximate values that can be precomputed

*These authors contributed equally to this work.

once and showed that, based on the strongest assumption made by Ramírez and Geffner, the observation sequence is not necessary if the current state and waypoints of interest are all known in advance (Masters and Sardina 2017).

Instead of using shortcuts to speed up the traditional recognition approach, we will take advantage of the consistent search space and research on multiple goal heuristic search (Davidov and Markovitch 2006) to avoid iteratively running the off-the-shelf planner. In particular, the explored regions of the space are expected to overlap between iterations so that many nodes will no longer be redundantly expanded. We will provide background on the Ramírez and Geffner method and multiple goal heuristic search in Section 2 and then discuss how the two work together in Section 3. Section 4 discusses advantages of the integrated approach including the potential speed-up and increase in accuracy due to an assumption the original method makes. As this is work in progress, we then conclude with our plans for future research using this proposed method in Section 5.

2 Background

Recognition with Classical Planners

The classical planning problem has tuple representation $\mathcal{P} = \langle F, I, A, G \rangle$ where F is the set of all fluents that describe the world, $I \subseteq F$ is the initial state’s true fluents, A is the set of actions that may alter the state’s fluents, and $G \subseteq F$ is the set of goal conditions that must be true in a state for the problem to be solved. Each action $a \in A$ is composed of precondition fluents $pre(a) \subseteq F$ that determine whether an action is applicable and effects $del(a), add(a) \subseteq F$ that set fluents false or true, respectively. Because full observability and deterministic actions are assumed, it is possible to solve \mathcal{P} using traditional search methods over a graph called the *state space* whose nodes represent every possible state of the world $S = 2^F$ and edges represent applicable actions whose effects transition between different states $a \in A : s_1 \in S \mapsto s_2 \in S$ if $pre(a) \subseteq s_1$ and $s_2 = (s_1 - del(a)) \cup add(a)$. The solution is a sequence of actions, called a plan π , that is equivalent to a path in the state space from $I \in S$ to some goal state $g \in S$ such that $G \subseteq g$. π is optimal if its total path cost $c(\pi)$ (the sum of the action costs) is the lowest possible cost that solves \mathcal{P} , but any π that solves \mathcal{P} is satisficing; this set is called Π_G .

On the other hand, a plan recognition problem defined by Ramírez and Geffner keeps the domain $\mathcal{D} = \langle F, I, A \rangle$ within the tuple $\mathcal{T} = \langle \mathcal{D}, O, \mathcal{G} \rangle$ where \mathcal{G} is the set of all possible goal conditions that an agent may choose to accomplish and O is a (possibly partial) sequence of observed actions that the agent has already taken towards its particular goal conditions $\gamma \in \mathcal{G}$. Thus the task is to identify γ given O and our domain \mathcal{D} (Ramírez and Geffner 2009), which was later extended to a probability distribution over \mathcal{G} given O (Ramírez and Geffner 2010). Because one can simply take the most probable goal conditions to be γ , we will use the latter approach for the remainder of this manuscript.

Under the assumption that an agent is as rational as possible, the observed agent will ideally perform actions that will accomplish γ with the lowest total plan cost possible.

Thus, if we were to strictly consider the set of plans that include the observed actions in order (but not necessarily consecutively), then the optimal plan(s) that provide a path from I to a state satisfying goal conditions in \mathcal{G} are the most likely plans that the agent will take. It thus follows that the goal conditions of these plans are the most likely ones that the agent intends to accomplish. We believe this is also the reason for the discrepancy between whether the method performs plan recognition or goal recognition, but we will neither take a side nor pursue this argument further.

More formally, the probabilistic reasoning for computing the likelihood over \mathcal{G} given O is derived via Bayes’s Rule:

$$P(\gamma \in \mathcal{G} | O) \propto P(O | \gamma) \cdot P(\gamma)$$

where the prior $P(\gamma)$ is assumed to be uniform and the likelihood is computed based on the plans that satisfy γ and O :

$$P(O | \gamma) = \sum_{\pi \in \Pi_\gamma} P(O, \pi | \gamma) = \sum_{\pi \in \Pi_\gamma} P(O | \pi, \gamma) \cdot P(\pi | \gamma).$$

The former term of the product is binary because each plan π either contains O as a subsequence or does not, which simplifies the above equality by summing over the subset of plans $\Pi_{\gamma+O} \subseteq \Pi_\gamma$. We explain the significance of $\gamma + O$ in the next paragraph, but we conclude the probability derivation as the latter term of the product is considered to be proportional to the exponential decay of the cost - this captures the notion of the rationality assumption for choosing a plan:

$$P(\pi | \gamma) \propto e^{-\beta c(\pi)}$$

for some constant β . Hence

$$P(O | \gamma) \propto \sum_{\pi \in \Pi_{\gamma+O}} e^{-\beta c(\pi)}.$$

To avoid a potentially infinite sum, Ramírez and Geffner lastly assume that, due to the rapid rate at which exponential decay decreases, it is sufficient to approximate the sum over $\Pi_{\gamma+O}$ with just the greatest term. This is conveniently the term associated with the optimal plan that both satisfies γ and follows the observation sequence O . By making a few changes to domain \mathcal{D} , we are able to customize it to account for following O :

- $F' = F \cup \{p_0, p_1, \dots, p_{|O|}\}$
- $I' = I \cup p_0$
- For each $a \in A$, observed action $a' \in A'$ has $pre(a') = pre(a)$, $add(a') = add(a) \cup \{p_{i-1} \rightarrow p_i \mid a \text{ is the } i^{\text{th}} \text{ action in } O\}$, $del(a') = del(a) \cup \{p_{i-1} \mid a \text{ is the } i^{\text{th}} \text{ action in } O\}$. This conditional effect is not directly applied to the preconditions because the action may still be performed without the purpose of matching an observation.

Then we may also define modified goal condition $\gamma + O = \gamma \cup \{p_{|O|}\}$ that signifies both achieving the goal conditions and following all the observations in the sequence. Hence, for each $\gamma \in \mathcal{G}$, we may define a classical planning problem $\langle F', I', A', \gamma + O \rangle$ whose optimal solution can compute the

above probability’s approximation. To normalize for the proportionality of $P(O|\gamma)$, a Boltzmann distribution with alternative state \bar{O} allows us to compute $P(O|\gamma)+P(\bar{O}|\gamma) = 1$. As \bar{O} implies that the observations were not followed, we can similarly approximate $P(\bar{O}|\gamma)$ using the optimal solution from the planning problem $\langle F', I', A', \gamma + \bar{O} \rangle$ where $\gamma + \bar{O} = \gamma \cup \{-p|_O\}$. Therefore, running any off-the-shelf classical planner with each new planning problem will yield all the numbers needed to approximate the probability distribution $P(\gamma \in \mathcal{G} | O)$ for recognition.

Multiple Goal Heuristic Search

Classical planning problems typically aim to find one solution for each planning problem because the single plan is sufficient for completing the task. Likewise, the majority of problems in computer science-related research need just one solution per problem. For example, machine learning models typically need one optimal parameter configuration. However, there are problems such as searching for website pages that require finding multiple solutions. Crawling the web should ideally find multiple websites that satisfy a queried topic.

Research on multiple goal heuristic search (Davidov and Markovitch 2006) was inspired by this application under the realistic constraint that there are not enough resources to search through the entire space, whether it be limited time, memory storage, or something more problem-specific. It is an anytime or contract algorithm that exchanges optimality for finding as many goals as possible before the specified resource limit is exhausted. Some degree of optimality is still implicit because it costs more resources to find less optimal results, but there can be trade-offs when there is one optimal solution in one region of the search space compared to many less optimal solutions in another region of the search space.

More formally, a search problem is represented as tuple $\langle N, E, I, G \rangle$ where N is the set of nodes that compose the search space, E is the set of edges that join adjacent nodes in the search space, $I \in N$ is the initial state from which the search begins, and $G \subseteq N$ is the set of all goal nodes in the search space that will solve the problem. The traditional search process iteratively explores some state along a horizon/frontier $H \subseteq N$ and then places all its adjacent nodes that are not yet explored into H . The loop typically stops once the chosen node from H is one of the desired goals G . The data structure chosen to represent H often determines how to navigate the search space; using a priority queue for F generates a *heuristic search* where the priority values are computed via a heuristic function $h : N \rightarrow \mathbb{R}^{\geq 0}$ that approximates the remaining cost from some node $n \in N$ to some goal node $g \in G$. In heuristic search, selecting n from H is interpreted as n being the closest node within H to a goal node, possibly a goal node itself.

Multiple goal heuristic search adjusts this standard procedure by terminating the loop when the specified amount of resources R is exhausted instead of after selecting a goal node from H to explore. The proposed heuristic function that ignores resource constraints is the *progress heuristic* that accounts for the subset of goal nodes towards which

each node in the frontier is relatively progressing. For each node $n \in N$, this set of goal nodes is

$$G_p(n) = \left\{ g \in G \mid h_d(n, g) = \min_{n' \in F} h_d(n', g) \right\}$$

where $h_d(n \in N, g \in G)$ is the heuristic distance function to a specific goal node. Then the average distance between n and all goal nodes in $G_p(n)$ is

$$D_p(n) = |G_p(n)|^{-1} \cdot \sum_{g \in G_p(n)} h_d(n, g)$$

so that the progress heuristic is $h_{progress}(n) = D_p(n) \cdot |G_p(n)|^{-1}$, which increases the priority value of node n in H when $D_p(n)$ is lesser and/or $G_p(n)$ is greater. That is, nodes with a *lesser average distance to more goal nodes are preferred* in the exploration step of search.

In the case of web search where the distance between web pages may not be easily estimated, the heuristic function can be substituted with the *marginal utility* function that represents how many goal states are expected to be found from a search starting at node $n \in N$ with the remaining unused resources. If there was perfect information about the entire search space, then the marginal utility would simply be:

$$MU(n, r \leq R) = \frac{\max_{T \in T(n, r)} |T_g(n) \cap T|}{r}$$

where $T_g(n)$ is the set of all goal nodes reachable from n in the search space and $T(n, r)$ is the set of all search trees generated from initial node n using no more than r resources. In this case, R is typically the number of nodes that can be explored during the search. Due to the lack of such perfect information without already performing the search, partial values at depth $d \leq D$ for the number of nodes visited $N_d(n)$ from each intermediate node n and the number of goal nodes found $G_d(n)$ are maintained in a table. These table entries are updated as the search continues, and the ratio $N_d(n) \cdot G_d(n)^{-1}$ can approximate the marginal utility for unexplored nodes in H when they are ‘similar’ either as siblings in a search tree/graph or via a metric between nodes.

3 Recognition with Multiple Goal Heuristic Search

When running an off-the-shelf classical planner for each modified planning problem generated as part of Ramírez and Geffner’s method (2010), we note that F' , I' , and A' are all kept constant — only the goal conditions change between each planner execution. Therefore, the state space and initial state do not change between planning problems so that the independent searches start identically. We thus rewrite all $2 \cdot |\mathcal{G}|$ planning problems into a single problem for multiple goal heuristic search: $\langle S', E_{A'}, I', G_{\mathcal{G}} \rangle$ where $S' = 2^{F'}$ is the set of all states, $E_{A'}$ is the set of all transition edges in the state space formed by the actions in A' , and $G_{\mathcal{G}} = \{\gamma + O \mid \gamma \in \mathcal{G}\} \cup \{\gamma + \bar{O} \mid \gamma \in \mathcal{G}\}$ is the set of all goal conditions.

Modifications for PDDL

As the search space for web crawling can be different from the state space composed by a PDDL representation, some modifications must be made to multiple goal heuristic search. Because the search space is derived from the state space, we will use node and state interchangeably.

The simplest change is the goal node check. Instead of comparing the explored node to all specified goal states in a list, we check whether the current state satisfies any one of the specified goal conditions. Although it is possible to enumerate all $2^{|F'-G|}$ goal states for each set of goal conditions $G \in G_G$, such a list would be quite large. The time complexity is also a linear search over the set of goal conditions $O\left(\sum_{G \in G_G} |G|\right)$ compared to a binary search (all states can be uniquely mapped to $|F'|$ -digit binary numbers) over the set of all goal states satisfying at least one set of goal conditions $O\left(\log\left(\sum_{G \in G_G} 2^{|F'-G|}\right)\right)$ where it is often the case that there are far fewer goal conditions than fluents $|G| \ll |F'|$.

The other differences are fortunately acknowledged in Section 4.5 of Davidov and Markovitch’s paper (2006), titled ‘Additional Considerations’. These were provided as alterations for search spaces with certain properties not present in their applications, and several of them are found in our case. In particular, we take advantage of the fact that fluents F' serve as features over PDDL-represented states so that we may apply the Hamming distance between binary representations of states to measure their similarity.

For example, we need to find at least one goal state that satisfies each $G \in G_G$ in order to complete all the computations for the probabilistic recognition algorithm, but the progress heuristic and marginal utility prefer to continue searching in regions of the search space where the already found goal states exist. Removing the found goal states from the goal list cannot be done due to the above change to goal condition checking. However, their proposed modified heuristic function $h'(n) = h(n) (1 + c_1 e^{-c_2 d(n)})$ is applicable where $d(n \in S')$ is the minimal Hamming distance from n to the visited goal states (we instead consider the goal conditions that visited goal states satisfy) and $c_1, c_2 \in \mathbb{R}^{\geq 0}$ are parameters. This increases the heuristic value of states closer to the found goals so that more exploration through other regions of the state space is enforced until one goal state per set of conditions is found.

Likewise, many actions are reversible in PDDL-represented domains so that the search space is often a well-connected bidirectional graph rather than a tree-like structure. This means that many nodes in H may lead to the same goal states such that exploring all of them is redundant and a waste of the limited resources. Davidov and Markovitch (2006) address this using the Hamming distance between states with lowest priority value in H and recently expanded states as a tie-breaker. States with greater distances are preferred because they are more likely to be associated with a different region of the search space and thus find novel goal states. Enforcing diversity in tie-breaking is an effective strategy for quickly finding a goal state in single-goal heuris-

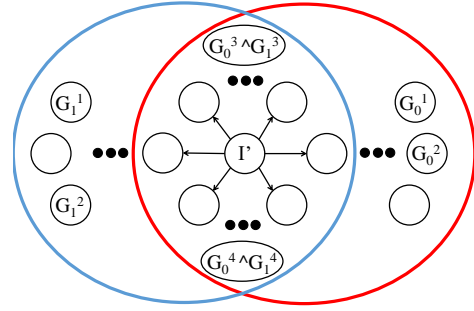


Figure 1: A search space where multiple goal heuristic search does not significantly reduce the number of explored nodes. The sparsity of the goal conditions’ states (G_i^k is the k^{th} goal state satisfying conditions G_i) and their displacements in the space make it as effective as running independent searches for each goal (red for G_0 and blue for G_1).

tic search methods such as A* without wasting as many resources (Asai and Fukunaga 2017), which further supports this strategy for quickly finding a variety of different goals.

4 Expected Benefits of Approach

Improved Runtime

We hypothesize that this reformulation will provide several benefits to the original recognition algorithm. The greatest contribution is quickening the algorithm because calling the planner multiple times not only increases overhead, but restarting the search within the same state space is expected to repeat exploration and expansion of many states. This repetition is most likely to occur with states whose distance to I' is smaller, but can also happen if there is overlap between different goal conditions. In the best case, one of the goal conditions γ cannot be satisfied with or without the observations so that there will be no solution for $\gamma + O$ or $\gamma + \bar{O}$. The off-the-shelf planner will identify this case when the entire subset of the state space reachable from I' is exhausted because no goal states were found, but all the other goal states for the remaining problems would have already been expanded for a complete overlap. In the worst case, the goal conditions correspond to sets of states in vastly different regions of the state space so that there is little overlap of the explored state space between each individual planning problem, illustrated in Figure 1. However, the average case will contain a reasonable overlap between each individual search’s visited regions of the state space because the goal states are not often at opposite regions of the state space.

Improved Computational Accuracy

The second hypothesized benefit takes advantage of the fact that a single set of goal conditions G corresponds to $2^{|F'-G|}$ goal states in the state space. This enables multiple goal heuristic search to find multiple goal states for the same goal condition, which gives us *multiple plans* that solve a single one of the planning problems. This is beneficial for the computational accuracy of the probability $P(O|\gamma)$ be-

cause, as mentioned in Section 2, the method assumes that only the most optimal plan’s cost matters. While their results imply that this assumption may be sufficient for ranking purposes in recognition, other works that use the probabilities for computation such as necessities in interaction (Freedman and Zilberstein 2017) will benefit from having a more accurate value. Clearly the assumption is also necessary for practical purposes because running an off-the-shelf planner $2 \cdot |\mathcal{G}|$ times is already time intensive, and obtaining additional plans from more planner executions (a satisficing planner is needed to avoid getting just the optimal plans) only increases the time requirement.

While the most accurate approximation would come from a method such as K^* search (Aljazzar and Leue 2011) that returns the k shortest paths (k most optimal plans), it would need to be executed for each goal independently and again sacrifice the speed-up. Sohrabi, Riabov, and Udrea (2016) used the K^* and LPG-d diversity planner (Nguyen et al. 2012) to sample many plans for a similar purpose in their variation that handles noisy observations. Multiple goal heuristic search will also find a variety of plans that satisfy the goal conditions. Employing a tree search version with sufficient resources will find every path to all the goal states, but the set of all plans will be infinite if a goal state is part of a cycle within the state space. Thus we must use a graph search version, which can still identify a single plan to reach each goal state and contribute up to $\sum_{G \in G_G} 2^{|F' - G|}$ plans to the computations.

Despite the rapid exponential decay of the probability mass contribution from much less optimal plans, Ramírez and Geffner do not mention that search trees usually grow exponentially at each depth. Hence there are more instances of plans with a specific less optimal cost, and there are often multiple optimal plans as well.

Lemma 1. *For all $d, i \in \mathbb{R}^{\geq 0}$, let Π_d be the set of all plans that solve some planning problem $\mathcal{P} = \langle F', I', A', G \in G_G \rangle$ with cost d greater than the optimal plan’s cost, and let Π_{d+i} be the set of all plans that solve \mathcal{P} with cost $d + i$ greater than the optimal plan’s cost. If the number of plans increases by a factor of $e^{\beta i}$ from Π_d to Π_{d+i} , then Π_{d+i} ’s contribution to the probability mass of $P(O|\gamma)$ will be as much or greater than Π_d ’s.*

Proof. Let $d \in \mathbb{R}^{\geq 0}$ and Π_d be the set of all plans that solve some planning problem $\mathcal{P} = \langle F', I', A', G \in G_G \rangle$ with cost d greater than the optimal plan’s cost. Then $P(\pi \in \Pi_d | G) = Z^{-1} e^{-\beta c(\pi)} = Z^{-1} e^{-\beta(c(\pi^*) + d)}$ where π^* is an optimal plan that solves \mathcal{P} and Z is a normalizing constant. Then setting $\sum_{\pi \in \Pi_d} P(\pi | G) = |\Pi_d| Z^{-1} e^{-\beta(c(\pi^*) + d)} \leq |\Pi_{d+i}| Z^{-1} e^{-\beta(c(\pi^*) + d + i)} = \sum_{\pi \in \Pi_{d+i}} P(\pi | G)$ for any $i \in \mathbb{R}^{\geq 0}$ implies that

$$e^{\beta i} \leq \frac{|\Pi_{d+i}|}{|\Pi_d|}.$$

Thus increasing the number of plans by a factor of $e^{\beta i}$ ensures that the set of plans with cost $d + i$ more than the optimal plan’s contribution to the probability mass of $P(O|\gamma)$ will be greater. \square

In the works that use these computations, β is always set to 1 so that the rate increases about 3 times per extra unit of cost. While this means that additional plans will contribute more to significantly alter the unnormalized likelihood value, experimentation will be necessary to confirm whether this alters the probabilities after normalization. If these extra plans affect all the likelihood computations evenly, then it is possible that the normalized probabilities will still be similar to the approximation version.

5 Discussion

Recognition using off-the-shelf classical planners allows the flexibility of not needing a library of predefined plans, but is consequently much slower due to the need to simulate the plans. We propose using multiple goal heuristic search in order to reduce the expected runtime by taking advantage of the fact that each individual execution of the planner has the same state space and initial state. This means that many of the same states are revisited with each planner call, and this redundancy can be removed by searching for all the goal states at once. This also allows us to potentially find multiple plans for the same goal condition, which may improve the accuracy of the probabilistic computations that currently assume that only the cost of the optimal plan matters. At the moment, this approach is being implemented for upcoming experiments that will test our hypotheses of these benefits.

Planned Experiments

We are currently implementing the algorithm for recognition using multiple goal heuristic search and plan to first test the hypothesis regarding expected speed-up. Because the Ramírez and Geffner method and ours both perform a form of heuristic search, we can compare speed with respect to the number of states that are generated and expanded during the searches. However, the other approaches for fast recognition that use cost propagation over the planning graph (E-Martín, R-Moreno, and Smith 2015) or landmark-based heuristics (Pereira, Oren, and Meneguzzi 2017) cannot be measured by the number of nodes expanded. Thus, we will need to perform the overall comparison via clocked runtime for the standard benchmarks developed from domains at past iterations of the International Planning Competition. Our approach should find additional goal states that would not be found during the individual planner executions; so it may be more fair to try a variation for this experiment that removes a goal condition from G_G when one of its corresponding states is selected.

On the other hand, it will be necessary to find all these different goal states in order to run experiments that test the hypothesis regarding expected accuracy improvements. These experiments will simply compare the approximated probability $P(\gamma \in \mathcal{G})$ using only the optimal plan’s cost per goal condition in G_G against all the plans’ costs found per goal condition in G_G . We will also investigate how the resource limit parameter for multiple goal heuristic search impacts the runtime and approximation; there may be some trade-offs between them.

Future Work

While faster recognition algorithms are generally ideal for any application, one of our key motivations is adaptive interaction. Initial work for in-progress recognition based on the Ramírez and Geffner approach was used to perform simple responsive interaction, but it was so slow that it was impossible to use with an actual human because it is unreasonable to expect a person to wait nearly thirty minutes for an interactive response to a single action (Freedman and Zilberstein 2017). Thus, if our hypotheses hold, we plan to use this quicker method in order to begin testing simple interactions with humans. We will also further study the applicability of multiple goal heuristic search to other recognition and planning problems.

Acknowledgements

We thank the anonymous reviewers for their feedback. This work was supported in part by the National Science Foundation under grant IIS-1405550.

References

- Aljazzar, H., and Leue, S. 2011. K^* : A heuristic search algorithm for finding the k shortest paths. *Artificial Intelligence* 175(18):2129–2154.
- Amir, O., and Gal, Y. K. 2013. Plan recognition and visualization in exploratory learning environments. *ACM Transactions on Interactive Intelligent Systems* 3(3):16:1–16:23.
- Asai, M., and Fukunaga, A. 2017. Tie-breaking strategies for cost-optimal best first search. *Journal of Artificial Intelligence Research* 58:67–121.
- Charniak, E., and Goldman, R. 1991. Probabilistic abduction for plan recognition. Technical Report CS-91-12, Brown University, Providence, RI, USA.
- Davidov, D., and Markovitch, S. 2006. Multiple-goal heuristic search. *Journal of Artificial Intelligence Research* 26:417–451.
- E-Martín, Y.; R-Moreno, M. D.; and Smith, D. E. 2015. A fast goal recognition technique based on interaction estimates. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, 761–768.
- Freedman, R. G., and Zilberstein, S. 2017. Integration of planning with recognition for responsive interaction using classical planners. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 4581–4588.
- Freedman, R. G.; Jung, H.-T.; and Zilberstein, S. 2014. Plan and activity recognition from a topic modeling perspective. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling*, 360–364.
- Geib, C. W., and Steedman, M. 2007. On natural language processing and plan recognition. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 1612–1617.
- Kautz, H. A. 1991. A formal theory of plan recognition and its implementation. In Allen, J. F.; Kautz, H. A.; Pelavin, R. N.; and Tenenber, J. D., eds., *Reasoning About Plans*. San Francisco, CA, USA: Morgan Kaufmann. 69–125.
- Masters, P., and Sardina, S. 2017. Cost-based goal recognition for path-planning. In *Proceedings of the Sixteenth Conference on Autonomous Agents and Multiagent Systems*, 750–758. São Paulo, Brazil: International Foundation for Autonomous Agents and Multiagent Systems.
- Mirsky, R., and Gal, Y. K. 2016. SLIM: semi-lazy inference mechanism for plan recognition. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 394–400.
- Mirsky, R.; Gal, Y. K.; and Shieber, S. M. 2017. CRADLE: an online plan recognition algorithm for exploratory domains. *ACM Transactions on Intelligent Systems and Technology* 8(3):45:1–45:22.
- Nguyen, T. A.; Do, M.; Gerevini, A. E.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. *Artificial Intelligence* 190:1–31.
- Pereira, R. F.; Oren, N.; and Meneguzzi, F. 2017. Landmark-based heuristics for goal recognition. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 3622–3628.
- Ramírez, M., and Geffner, H. 2009. Plan recognition as planning. In *Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence*, 1778–1783.
- Ramírez, M., and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1121–1126.
- Sohrabi, S.; Riabov, A. V.; and Udrea, O. 2016. Plan recognition as planning revisited. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 3258–3264.
- Vered, M., and Kaminka, G. A. 2017. Heuristic online goal recognition in continuous domains. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 4447–4454.