Preference-Guided Planning: An Active Elicitation Approach

Extended Abstract

Mayukh Das
University of Texas, Dallas
mayukh.das1@utdallas.edu

Phillip Odom
Indiana University, Bloomington
phodom@indiana.edu

Md. Rakibul Islam
Washington State University
mislam@eecs.wsu.edu

Janardhan Rao Doppa
Washington State University
jana@eecs.wsu.edu

Dan Roth
University of Pennsylvania
danroth@seas.upenn.edu

Sriraam Natarajan
University of Texas, Dallas
sriraam.natarajan@utdallas.edu

ACM Reference Format:
Mayukh Das, Phillip Odom, Md. Rakibul Islam, Janardhan Rao Doppa, Dan Roth, and Sriraam Natarajan. 2018. Preference-Guided Planning: An Active Elicitation Approach. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Planning under uncertainty has exploited human (domain) expertise in several different directions [1–6, 9, 10, 13–15]. One key research thrust in this direction is that of specifying preferences as advice to the planner in order to reduce the search over the space of plans. While successful, most of these approaches require that the advice be specified before planning. However, humans tend to give the most obvious advice and more importantly, this advice may not directly benefit the planner. We propose a framework in which the planner actively solicits preferences as needed. More specifically, our proposed planning approach computes the uncertainty in the plan explicitly and then queries the human expert for advice. This approach not only removes the burden on the human expert to provide all the advice upfront but also allows the learning algorithm to focus on the most uncertain regions of the plan space and query accordingly.

We present an algorithm for active preference elicitation in planning called the *preference-guided planner* (PGPlanner) to denote that the agent treats the human advice as *soft preferences* and solicits these preferences as needed. We consider a Hierarchical Task Network (HTN) planner [8] for this task as it allows for seamless natural interaction with humans who solve problems by decomposing them into smaller problems. Hence, HTN planners can facilitate humans in providing knowledge at varying levels of generality. We evaluate our algorithm on several standard domains and a novel Blocks-World domain where we compare against several baselines. Our results show that this collaborative approach allows for more efficient and effective problem solving compared to the standard planning as well as providing all the preferences in advance. It must be mentioned that our framework treats the human input as soft preferences and allows to trade-off between potentially a sub-optimal expert and a complex plan space[1].

[1]For a longer version of the paper, please refer to "https://arxiv.org/abs/1804.07404"

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

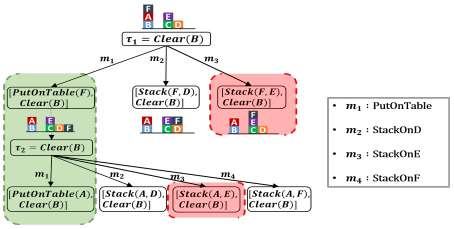

Figure 1: Preference-guided search in Blocks-World. Rectangular nodes are tasks. Admissible methods for decomposing task τ_1, are m_1, m_2, m_3 & an extra one later, m_4. Block configurations indicate current state. Green and red shaded areas denote preferred and non-preferred decompositions.

Algorithm 1 Preference-Guided Planner (PGPlanner)

1: Intialize *Frontier*= $Goal$, $\{\mathfrak{P}\}$, Current Plan $pl = \emptyset$
2: Node $n = (s_n, \tau_n, M_{\tau_n}) \leftarrow$ Pop(*Frontier*)
3: Policy $\pi(M_{\tau_n})$, Uncertainty $\mu(\pi) \leftarrow$ EvaluateNode(n,$\{\mathfrak{P}\}$)
4: **IF** $\mu(\pi) >$ AcceptableUncertainty: $\mathfrak{P}^{(n)} \leftarrow$ Query(s_n,τ_n)
Update policy $\pi \leftarrow$ EvaluateNode(n,$\{\mathfrak{P}\} \cup \mathfrak{P}^{(n)}$)
5: Choose best method $m^* \leftarrow \arg\max_m \pi$: Decompose($\tau_n$, m^*)
6: **IF** τ_n primitive: Update(pl, a) ▷ if action a feasible in s_n
7: Repeat 2 to 6 **IF** $Frontier \neq \emptyset$, **ELSE return** pl

2 PREFERENCE-GUIDED PLANNING

Preference-Guided planning employs preferences to guide the search through the space of possible task decompositions (methods) in an HTN. It actively acquires such preferences in the regions it is most uncertain about, thus, minimizing the set of preferences needed upfront. Preferences are similar to IF-THEN rules, formally defined as, *a tuple* $\mathfrak{P} = (\wedge f_i, \tau_j, M^+_{\tau_j}, M^-_{\tau_j})$, *where* $\wedge f_i$ *encodes applicability condition(s) of preference in the current state,* τ_j *is the task,* $M^+_{\tau_j}$ *is the user's preferred set of methods and* $M^-_{\tau_j}$, *the non-preferred ones.* Preferences may be defined over any level of generality, given by the conditions $\wedge f_i$. Figure 1 illustrates the effect of preferences in a Blocks-World scenario. Moving blocks to table is preferred here ($\mathfrak{P} = (Space(Table), Clear(B), \{PutOnTable\}, \{StackonE\})$) since it makes constructing arbitrary towers easier as the blocks can be positioned quickly. It may apply to multiple levels. Preferences are used to increase $P(M^+)$ and decrease $P(M^-)$ (method probabilities). We demonstrate empirically that such a preference is more useful when solicited than when provided before planning.

While we assume experts' availability throughout the planning process, we aim to rely on him/her only when necessary. PGPlanner, decides when it needs help the most and queries the expert, similar in spirit to stream-based active learning [7]. A query generated by PGPlanner (wrt. HTN node n - abstract container comprising current state s_n, task τ_n and admissible methods M_{τ_n}) is the

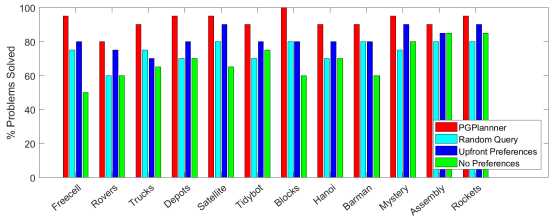

Figure 2: Efficiency (% problems solved in 10 mins)

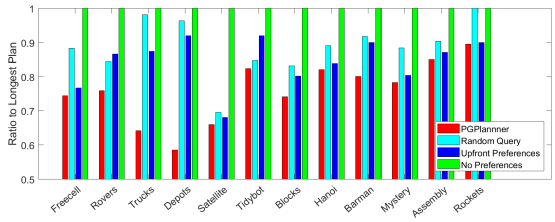

Figure 3: Performance (ratio of avg. plan lengths to longest avg.)

tuple $q_n = (s_n, \tau_n)$. An expert's response to a query is a preference $\mathfrak{P}^{(n)}$. HTN hierarchy allows preferences over any region of the state space that contains s_n, i.e. at any level of generality. In a factored space, this involves choosing the variables in description of s_n.

Problem Setup: *Given* - An HTN problem defining the initial state/goal task(s), access to an expert and query budget. *Objective* - Generate best plan, one that minimizes the total expected cost $J(\pi)$ of generating a plan by finding the most suitable policy π, a distribution over the methods for task τ_n of each HTN node n. Formally, the objective is: $\left[\arg\min_\pi \ \left(J(\pi) = T\mathbb{E}_{n\sim d_\pi} C_\pi(n)\right)\right]$, where π is a distribution over methods, d_π a distribution of HTN nodes reached, T the decomposition depth and $C_\pi(n)$ the expected cost at node n. $C_\pi(n) = \mathbb{E}_{m\sim\pi_n} C(n,m)$ where $C(n,m)$ is the immediate cost of selecting m at node n. The cost is generic and if the planner aims to find the shortest plan, the immediate cost C is the number of primitive actions added to the current plan.

The algorithm: PGPlanner (Alg 1) proceeds via best-first search (with backtracking) through the space of decompositions (methods) to reach a valid plan, maintaining a *Frontier* of HTN nodes and a set of preferences $\{\mathfrak{P}\}$ which may be empty. Each node n in the HTN with task τ_n could potentially decompose in several ways according to the admissible methods (M_{τ_n}). The cost of selecting a method $m \in M_{\tau_n}$ ($C_\pi(n)$ for $\pi(n) = m$) is estimated by rolling out (simulating) decompositions till a predefined depth r and then approximating the utility as the sum of current plan length L_m and distance to goal D_m (post roll-out). The methods are also scored according to the current set of preferences, $A_m = N_m^+(\{\mathfrak{P}\}) - N_m^-(\{\mathfrak{P}\})$, where N_m^+ is the number of rules which prefer method m while N_m^- is the number which *non*-prefer it). The overall cost estimate ($\hat{C}(m)$) is a combination of this preference score and the estimated cost function, $\hat{C}(m) = L_m + D_m + inverse(A_m)$. The *inverse* allows for minimizing cost while maximizing the preference score. $\hat{C}(m)$ is converted into a probability distribution (π) over the methods M_{τ_n} via Boltzmann softmax, $\pi = e^{\hat{C}(m)} / \sum_{m'\in M_{\tau_n}} e^{\hat{C}(m')}$ (EvaluateNode, **line 3**). If the level of uncertainty (entropy in our case $\mu(\pi)$) of this distribution is high, i.e. above AcceptableUncertainty, the expert is queried. The expert may respond to query with a preference $\mathfrak{P}^{(n)}$ which is added to $\{\mathfrak{P}\}$ and the methods are evaluated again (**line 4**). Subsequently, *or if the uncertainty was acceptable previously*, the method with the highest probability (m^*) is chosen to decompose task τ_n (**line 5**) and new HTN nodes are added to *Frontier*. Note, if τ_n is primitive, corresponding action a is added to the current plan and the state s_n is changed if it satisfies the preconditions of a (**line 6**). Backtracking at unfeasible tasks and subsequently choosing the "next best" decomposition method, ensures completeness. It is achieved via simple recursion and the details are excluded for brevity. Essentially, PGPlanner transforms the plan search into a sequential decision making problem where the agent can either query or choose among the available options (methods) and use preference obtained in response to query for policy revision.

3 EXPERIMENTAL EVALUATION

We aim to investigate if PGPlanner generates efficient plans and analyze its efficiency. To this effect, we compare PGPlanner against - (1) *Upfront Preferences* - all preferences specified apriori [12], (2) *Random Query*, and (3) *No Preferences*. PGPlanner is developed by extending JSHOP [11], an HTN planner, for (1) roll-out, (2) active elicitation of preferences, and (3) guided search. We also built an interface that facilitates seamless human-agent interaction. We consider 11 standard planning domains and a novel Blocks-World domain, that employs an apparatus to detect physical block configurations via sensors creating a surrogate real-world environment, for evaluation. The domains were of varying complexity based on number of objects, relations, actions and methods. Most domains, had 20 problems each except a few which had 10.

Results: All experiments were performed with consistent settings (AcceptableUncertainty/entropy= 0.5 & time constraint = 10 mins, accommodating the limited time and attention of the expert). Figure 2 shows PGPlanner outperforming all baselines in terms of efficiency, when we compare the percentage of problems solved given the time constraint. Clearly, active elicitation guides the planner to solutions more efficiently. Also, planning with preferences is almost always better than without them, however, random querying is not an effective elicitation strategy. We investigate the quality of generated plans via the ratio of average plan length of each planning method compared to the longest (*No Pref*). In every domain, we have only used those problems which were solved/completed by all approaches in the given time constraint. We observe (Figure 3) that PGPlanner has the lowest average across all domains thus demonstrating the effectiveness and efficiency.

4 CONCLUSION

Our proposed PGPlanner queries the expert only as needed and reduces the burden on the expert to understand the planning process to suggest useful advice. We empirically validate the efficiency and effectiveness of PGPlanner across several domains and demonstrate that it outperforms the baselines even with fewer preferences. Currently, our planner does not validate the preferences, but rather assumes the user is an expert. We will extend to the planner to recommend improvements to the set of preferences. We will investigate other avenues to obtain preferences including from crowds, transferring across subtasks as well as other domains.

ACKNOWLEDGEMENTS We gratefully acknowledge the support of CwC Program Contract W911NF-15-1-0461 with the US Defense Advanced Research Projects Agency (DARPA) and the Army Research Office (ARO). Any opinions, findings and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, ARO or the US government.

REFERENCES

[1] Mitchell Ai-Chang, John Bresina, Len Charest, Adam Chase, JC-J Hsu, Ari Jonsson, Bob Kanefsky, Paul Morris, Kanna Rajan, Jeffrey Yglesias, et al. 2004. Mapgen: Mixed-initiative planning and scheduling for the mars exploration rover mission. *IEEE Intelligent Systems* 19, 1 (2004), 8–12.

[2] James Allen and George Ferguson. 2002. Human-machine collaborative planning. In *International NASA Workshop on Planning and Scheduling for Space*.

[3] Craig Boutilier, Thomas Dean, and Steve Hanks. 1995. Planning under uncertainty: Structural assumptions and computational leverage. In *European Workshop on Planning*.

[4] Ronen I Brafman and Yuri Chernyavsky. 2005. Planning with Goal Preferences and Constraints.. In *ICAPS*. 182–191.

[5] Thomas Dean, Leslie Pack Kaelbling, Jak Kirman, and Ann Nicholson. 1995. Planning under time constraints in stochastic domains. *Artificial Intelligence* 76, 1 (1995), 35–74.

[6] George Ferguson, James F Allen, and Bradford W Miller. 1996. TRAINS-95: Towards a Mixed-Initiative Planning Assistant.. In *AIPS*.

[7] Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. 1997. Selective Sampling Using the Query by Committee. *Machine Learning* 28 (1997), 133–168.

[8] Malik Ghallab, Dana Nau, and Paolo Traverso. 2004. *Automated planning: theory & practice*. Elsevier.

[9] Yi-Cheng Huang, Bart Selman, Henry Kautz, et al. 1999. Control knowledge in planning: benefits and tradeoffs. In *AAAI/IAAI*. 511–517.

[10] Karen L Myers. 1996. Advisable planning systems. *Advanced Planning Technology* (1996), 206–209.

[11] Dana S. Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, J. William Murdock, Dan Wu, and Fusun Yaman. 2003. SHOP2: An HTN Planning System. *J. Artif. Intell. Res. (JAIR)* 20 (2003), 379–404.

[12] Shirin Sohrabi, Jorge A Baier, and Sheila A McIlraith. 2009. HTN planning with preferences. In *IJCAI*.

[13] Shirin Sohrabi and Sheila A McIlraith. 2008. On planning with preferences in HTN. In *(NMR)*.

[14] Kartik Talamadupula, Gordon Briggs, Matthias Scheutz, and Subbarao Kambhampati. 2013. Architectural mechanisms for handling human instructions in open-world mixed-initiative team tasks. *Advances in Cognitive Systems (ACS)* 6 (2013).

[15] Sek-Wah Tan and Judea Pearl. 1994. Specification and evaluation of preferences for planning under uncertainty. In *KR*.