Probabilistic Hierarchical Planning over MDPs

(Extended Abstract)

Yuqing Tang
Graduate Center
City University of New York
New York, USA
ytang@cs.gc.cuny.edu

Felipe Meneguzzi
Katia Sycara
Robotics Institute
Carnegie Mellon University
Pittsburgh, USA
meneguzz@cs.cmu.edu
katia@cs.cmu.edu

Simon Parsons
Brooklyn College
City University of New York
New York, USA
parsons@sci.brooklyn.cuny.edu

ABSTRACT

In this paper, we propose a new approach to using probabilistic hierarchical task networks (HTNs) as an effective method for agents to plan in conditions in which their problem-solving knowledge is uncertain, and the environment is non-deterministic. In such situations it is natural to model the environment as a Markov decision process (MDP). We show that using Earley graphs, it is possible to bridge the gap between HTNs and MDPs. We prove that the size of the Earley graph created for given HTNs is bounded by the total number of tasks in the HTNs and show that from the Earley graph we can then construct a plan for a given task that has the maximum expected value when it is executed in an MDP environment.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence

General Terms

Algorithms, Design

Keywords

Planning (single and multi-agent)

1. INTRODUCTION

Although the complexities of planning in the real-world are better captured by *stochastic* formalisms such as Markov Decision Processes (MDPs), domain specification using these formalisms is a very complex task for all but trivial scenarios. By contrast, classical planning formalisms are more intuitive to non-experts where one particular formalism, Hierarchical Task Networks (HTNs) being the formalism of choice for planning in *deterministic* domains. In this paper, we propose a method to bridge the gap between HTNs and MDPs by performing maximum expected utility (MEU) planning on an HTN domain specified in terms of a hierarchy of tasks induced by a library of *methods*. To accomplish this, we look at the HTN methods as if they were the rules of a context-free grammar and apply our own modified version of an Earley parser [3] to generate a data structure known as *Earley state chart* [4]. Earley parsing is a dynamic programming technique widely used in the efficient processing of natural language that has been adapted to parse sentences probabilistically in order to cope with the ambiguity inherent to human languages. The semantic representation in the Earley state chart naturally leads to a probabilistic semantics, as well as algorithms for probabilistic context free grammar parsing. This class of algorithm performs a parallel top-down search over all possible grammar parses for a given input sentence, and its complexity is bounded by $O(N^3)$ on the number of input words [3].

Our adaptation of Earley parsing for probabilistic HTN planning was inspired by earlier efforts relating task decomposition to grammar parsing [1].In constructing our modified Earley graph, we take into consideration the preconditions of tasks and the effects of actions to make sure that the generated plans follow the constraints imposed by the HTN domain specification. While earlier work relates planning and parsing only for deterministic domains, we extend this concept into probabilistic domains by annotating probabilities in the HTN methods, allowing us to calculate the probabilities of generating plans in the domain. Furthermore, we allow a user to specify rewards for specific states in the HTN specification in the same way as goal states are specified in classical planning, allowing us to use the Earley graph to calculate the expected utilities of these plans, and ultimately allowing us to perform MEU planning conforming with HTN constraints.

2. FROM METHODS TO EARLEY GRAPHS

The core of our approach consists of adapting the Earley parsing approach of [4] to accommodate the components of states (of preconditions and effects), and task decompositions. The approach keeps track of the decomposition procedure for the set of all possible execution trajectories using the methods from an HTN domain. This is done by modifying the concept of Earley states to include the information of states and actions in addition to the task decompositions. To avoid the naming conflict with the state space of a planning domain, we call these modified states *Earley nodes*.

DEFINITION 1. *Let $m = \langle t, \mathcal{H} \rangle$ be an HTN method, t be a task and $\mathcal{H} = \langle T, C \rangle$ be an HTN with tasks T and constraints C. From m we generate $|T|$ Earley nodes. Each Earley Node EN is of the form $EN_{m,t_i} = \langle m, t_i \rangle$ where $t_i \in network(m)$. For notational convenience, we denote m by $method(EN_{m,t_i})$, t_i by $current(EN_{m,t_i})$, and $task(m)$ by $root(EN_{m,t_i})$.*

DEFINITION 2. *An Earley graph for a method library M is a graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ where*

- *$\mathcal{N} = \{EN\}$ is a set of Earley nodes; and*

Cite as: Probabilistic Hierarchical Planning over MDPs (Extended Abstract), Y. Tang, F. Meneguzzi, S. Parsons and K. Sycara,, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 1143-1144.

- $\mathcal{E}$ *is the set of Earley links of three types:*
  - *A* predicting link $\langle EN_{m,t_i}, EN_{m',t'_{start}}\rangle$ *where* $task(m') = t_i$ *and* $t'_{start} = start(m')$ *is the starting task of* m' *which precedes all the other tasks in* m'. $EN_{m',t'_{start}}$ *is called* a predicting node.
  - *A* scanning link $\langle EN_{m,a}, EN_{m,t_i}\rangle$ *where* a *is a primitive task in* m, *and* $t_i = next(m, a)$ *is a task immediate succeeding* a *in* m. $EN_{m,a}$ *is called* a scanning node.
  - *A* completing link $\langle EN_{m',t'_{end}}, EN_{m,t_i}\rangle$ *where* $t'_{end} = end(m')$ *is the ending task of* m', *and* $t_i = next(m, task(m'))$ *is an immediate task succeeding* $task(m')$ *in* m. $EN_{m',t'_{end}}$ *is called* a completing node.

A predicting link $\langle EN_{m,t_i}, EN_{m',t'_{start}}\rangle$ marks a possible decomposition of a task t_i; a completing link $\langle EN_{m',t'_{end}}, EN_{m,t_i}\rangle$ marks a possible completion of a task in m resulting in the investigation of the next task t_i in m; a scanning link marks an execution of a primitive task a resulting in the investigation of the next task t_i in m.

In an Earley graph, a path from $EN_{m,t_{start}}$ to $EN_{m,t_{end}}$ corresponds to a decomposition of $task(m)$ and an execution trajectory of $task(m)$ according to the methods in the library $\mathcal{M}$ if the traversal of the paths are carefully managed to ensure that 1) the task decompositions corresponding to the path are valid, and 2) the preconditions of the methods and primitive tasks in the path are met. The first condition is to avoid the mismatch of a completing node into a parent method which doesn't invoke such a method. The Earley graph enables us to do this kind of dynamic programming with complexity bounded by the size of the Earley graph. After the relaxation, we can assign probability to the predicting and completing links to model the uncertainty in which decompositions can be valid.

3. INTEGRATING HTNS AND MDPS

In a classic MDP problem, the solution of an MDPs is a *policy*, which indicates the best action to take in each state. Thus, an MDP policy is a total function mapping states into actions, so a policy π is represented as a function $\pi : S \rightarrow A$. Information on the rewards of states makes it possible to compute the value of a a given state under a particular policy π – it is the expected value of carrying out the policy from that state, given some *discount factor* γ. While in the literature, other solution concepts have been proposed (such as decision trees [2]), we focus on the concept of probabilistic hierarchical planning, therefore we will adopt the task decomposition solution concept of HTN planning while obtaining the maximum expected rewards for this task decomposition.

3.1 Semantics of an HTN Earley Graph

The probabilities assigned to the Earley links are about the uncertainty in decomposing tasks. The predicting link stores the subjective knowledge on how probable it is that a method can be used to successfully decompose a task, so it is assigned number $Pr(m|t)$. A scanning link is assigned probability 1 because in terms of task decomposition, encountering a primitive task in the task network means that we will move to the next task of the encounter task with probability 1. Thus, the probability of a path $\langle EN_0, EN_1, \ldots, EN_N\rangle$ extracted by our technique is

$$Pr(\langle EN_0, EN_1, \ldots, EN_N\rangle) = Pr(EN_0|EN_1)\times\ldots\times Pr(EN_{n-1}|EN_n)$$

This is the probability of a pure task network decomposition which models the uncertainty of how computer program or a human expert uses a library of methods to achieve a task corresponding to $root(EN_0)$ assuming that the method choices for any two tasks are independent.

3.2 Utility of Earley Paths

Given a decomposition-execution path de, the value of this path is the sum of all the rewards encountered $V(de) = \Sigma_{a_j \in de} R(s_j)$. The expected value of a decomposition path is

$$V(path) = \Sigma_{de \in DE(path)} \left(V(de) \cdot Pr(de)\right)$$

Similar to the MDP value computation, the expected value of a path can be computed iteratively with the Earley graph. Let $sub^{path}(s, EN)$ be the subpath of $path$ starting from $\langle s, EN\rangle$, we define $V^{path}(s|EN) = V(sub^{path}(s, EN))$ and $V^{path}(EN) = \Sigma_s V^{path}(s|EN)$. Related to a decomposition $path = \langle EN_0, \ldots, EN_n\rangle$, we define the value of the fully complete Earley node EN_n to be

$$V^{path}(s|EN_n) = R(s).$$

If EN_{i+1} is a predicting or completing node, we define

$$V^{path}(s|EN_i) = Pr(EN_{i+1}|EN_i) \cdot V(s|EN_i)$$

If EN_{i+1} is a scanning node, we define

$$V^{path}(s|EN_i) = Pr(EN_{i+1}|EN_i) \cdot \Big(R(s) + \Sigma_{s'} Pr(s'|s, a)\, V^{path}(s'|EN_{i+1})\Big)$$

We can then traverse the Earley graph for paths corresponding to valid task decompositions with a stack tracking the start and completion of methods. Using dynamic programming, the traversal can be focused towards paths with maximum expected utilities.

4. CONCLUSIONS AND FUTURE WORK

Our ultimate goal here is not only to perform probabilistic hierarchical planning for an uncertain environment, but also to utilize the approach for multiagent system control. A system of cooperative agents could thus communicate to share the same set of task networks while working in the same environment with the same characteristics of uncertainty. As every agent can construct the same Earley graph structure from the task network library, we will be able to incrementally adapt to the environment and revise their task decomposition probabilities. Thus, the multiagent system can converge to a set of cooperative behaviors prescribed by the shared set of task networks. The resulting system allows us to specify its group behaviors in a way that is close to how humans perform problem solving while accommodating uncertainty both in the knowledge of problem solving and the in the environment.

Acknowledgement: This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

5. REFERENCES

[1] A. Barrett and D. S. Weld. Task-decomposition via plan parsing. In *AAAI'94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 2)*, pages 1117–1122, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence.

[2] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[3] J. Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102, 1970.

[4] A. Stolcke. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201, 1995.