Online Planning for Large MDPs with MAXQ Decomposition

(Extended Abstract)

Aijun Bai
School of Computer Science
Univ. of Sci. & Tech. of China
Hefei, Anhui 230027 China
baj@mail.ustc.edu.cn

Feng Wu
School of Computer Science
Univ. of Sci. & Tech. of China
Hefei, Anhui 230027 China
wufeng@mail.ustc.edu.cn

Xiaoping Chen
School of Computer Science
Univ. of Sci. & Tech. of China
Hefei, Anhui 230027 China
xpchen@ustc.edu.cn

ABSTRACT

Markov decision processes (MDPs) provide an expressive framework for planning in stochastic domains. However, exactly solving a large MDP is often intractable due to the curse of dimensionality. Online algorithms help overcome the high computational complexity by avoiding computing a policy for each possible state. Hierarchical decomposition is another promising way to help scale MDP algorithms up to large domains by exploiting their underlying structure. In this paper, we present an effort on combining the benefits of a general hierarchical structure based on MAXQ value function decomposition with the power of heuristic and approximate techniques for developing an online planning framework, called MAXQ-OP. The proposed framework provides a principled approach for programming autonomous agents in a large stochastic domain. We have been conducting a long-term case-study with the RoboCup soccer simulation 2D domain, which is extremely larger than domains usually studied in literature, as the major benchmark to this research. The case-study showed that the agents developed with this framework and the related techniques reached outstanding performances, showing its high scalability to very large domains.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

Keywords

MDP, Online Planning, MAXQ, RoboCup 2D

1. MAIN RESULTS

Markov decision processes (MDPs) have been proved to be a useful model for planning under uncertainty. In general, online planning interleaves planning with execution and chooses the best action for the current step. Given the MAXQ [2] hierarchy of an MDP, the main procedure of MAXQ-OP evaluates each subtask by forward search to compute the recursive value functions $V^*(i,s)$ and $Q^*(i,s,a)$ online. This involves a complete search of all paths through the MAXQ hierarchy starting from the root task M_0 and ending with some primitive subtasks at the leaf nodes. After the search process, the best action $a \in A_0$ is selected for the root task M_0 based on the recursive Q function. Meanwhile, the true primitive action $a_p \in A$ that should be performed first can also be determined. This action a_p will be executed to the environment, leading to a transition of the system state. Then, the planning procedure starts over to select the best action for the next step.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

1.1 Task Evaluation over Hierarchy

The search starts with the root task M_i and the current state s. Then, the node of the current state s is expanded by trying each possible subtask of M_i. This involves a recursive evaluation of the subtasks and the subtask with the highest value is selected. The evaluation of a subtask requires the computation of the value function for its children and the completion function. The value function can be computed recursively. Therefore, the key challenge is to calculate the completion function.

Intuitively, the completion function represents the optimal value of fulfilling the task M_i after executing a subtask M_a first. Obviously, computing the optimal policy is equivalent to solving the entire problem. In principle, we can exhaustively expand the search tree and enumerate all possible state-action sequences starting with s, a and ending with s' to identify the optimal path. However, this may be inapplicable for large domains. In Section 1.2, we will present a more efficient way to approximate the completion function.

1.2 Completion Function Approximation

To compute the optimal completion function, $C^{\pi^*}(i,s,a)$, the agent must know the optimal policy π^*, which is unavailable in the online settings. Due to the time constraint, it is intractable to find the optimal policy online since the search process is equivalent to solve the entire problem. When applying MAXQ-OP to large problems, approximation should be made to compute the completion function for each subtask. We assume that each subtask M_i will terminate at its terminal states in G_i with a prior distribution of D_i. In principle, D_i can be any probability distribution associated with each subtask. It can also take into consideration of the task parameters. For simplicity, we take uniform distribution as an example, then $C^{\pi}(i,s,a)$ can be approximated

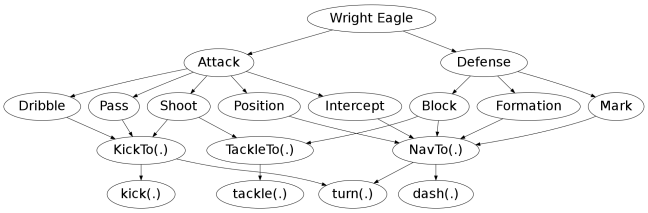

Figure 1: MAXQ task graph for Wright Eagle

as:

$$C^{\pi}(i,s,a) \approx \frac{1}{|\tilde{G_a}|} \sum_{s' \in \tilde{G_a}} V^{\pi}(i,s'), \quad (1)$$

where $\tilde{G_a} \subset G_a$ is a set of sampled states drawn from uniform distribution D_a. A recursive procedure is proposed to estimate the completion function. In practice, the prior distribution $P(s', N|s, a)$–a key distribution when computing the completion function, can be improved by considering the domain knowledge.

1.3 Heuristic Search in Action Space

For some domains with large action space, it may be very time-consuming to enumerate all possible actions (subtasks) exhaustively. Hence it is necessary to introduce some heuristic techniques (including prune strategies) to speed up the search. Intuitively, there is no need to evaluate those actions that are not likely to be better. Different heuristic techniques can be chosen for different subtasks, such as hill-climbing, gradient ascent, branch and bound, etc.

2. CASE STUDY: ROBOCUP 2D

It is our long-term effort to apply the MAXQ-OP framework to the RoboCup soccer simulation 2D domain–a very large testbed for the research of decision-theoretic planning [3]. In this section, we present a case-study of this domain and evaluate the performance of MAXQ-OP based on the general competition results with several high-quality teams in the RoboCup simulation 2D community. The goal is to test the scalability of MAXQ-OP and shows that it can solve large real-world problems that are previously intractable.

2.1 Solution with MAXQ-OP

The graphical representation of the MAXQ hierarchical structure of our team Wright Eagle[1] is shown in Figure 1, where a parenthesis after a subtask's name indicates this subtask will take parameters. It is worth noting that state abstractions are implicitly introduced by this hierarchy. To deal with the large action space, heuristic methods are critical when applying MAXQ-OP. Table 1 summarizes the general performance of our team with MAXQ-OP in the RoboCup completion of past 7 years.[2]

There are multiple factors contributing to the general performance of a RoboCup 2D team. It is our observation that our team benefits greatly from the abstraction we made for the actions and states. The key advantage of MAXQ-OP in our team is to provide a formal framework for conducting the search process over a task hierarchy. Therefore, the team can search for a strategy-level solution automatically online by given the pre-defined task hierarchy. To the best of our knowledge, most of the current RoboCup teams develop their team based on hand-coded rules and behaviors. Overall, the goal of this case-study is twofold: 1) it demonstrates the scalability and efficiency of MAXQ-OP for solving a large real-world application such as RoboCup soccer simulation 2D; 2) it presents a decision-theoretic solution for developing a RoboCup soccer team, which is more general and easy for programming high-level strategies.

[1]Team website: `http://www.wrighteagle.org/2d`

[2]Logfiles: `http://ssl.robocup-federation.org/ftp/2d/log/`

Table 1: History results of Wright Eagle

Competitions	Games	Goals	Win	Draw	Lost
RoboCup 2005	19	84 : 16	15	2	2
RoboCup 2006	14	57 : 6	12	2	0
RoboCup 2007	14	125 : 9	11	1	2
RoboCup 2008	16	74 : 18	13	1	2
RoboCup 2009	14	81 : 17	12	0	2
RoboCup 2010	13	123 : 7	11	0	2
RoboCup 2011	12	151 : 3	12	0	0

3. CONCLUSIONS

This paper presents MAXQ-OP–a novel online planning algorithm that benefits from both the advantage of hierarchical decomposition and the power of heuristics. A key contribution of this work is to approximate the prior distribution when computing the completion function. By given such prior distributions, MAXQ-OP can evaluate the root task online without actually computing the sub-policy for each subtask. Similar to our work, Barry et al. proposed an *offline* algorithm called DetH* to solve large MDPs hierarchically by assuming that the transitions between macro-states are totally deterministic [1]. In contrast, we assume a prior distribution over the terminal states of each subtask, which is more realistic. The case study shows that MAXQ-OP is able to solve a very large problem such as the RoboCup 2D that are previously intractable in the literature of the decision-theoretic planning. This demonstrates the soundness and stability of MAXQ-OP for solving large MDPs with the pre-defined task hierarchy. In the future, we plan to theoretically analyze MAXQ-OP with different task priors and try to generate these priors automatically.

4. ACKNOWLEDGMENTS

This work is supported by the National Hi-Tech Project of China under grant 2008AA01Z150 and the Natural Science Foundations of China under grant 60745002 and 61175057.

5. REFERENCES

[1] J. Barry, L. Kaelbling, and T. Lozano-Perez. Deth*: Approximate hierarchical solution of large markov decision processes. In *International Joint Conference on Artificial Intelligence*, pages 1928–1935, 2011.

[2] T. G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Machine Learning Research*, 13(1):63, May 1999.

[3] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12(2-3):233–250, 1998.