Tree-based Pruning for Multiagent POMDPs with Delayed Communication

(Extended Abstract)

Frans A. Oliehoek
MIT CSAIL / Maastricht University
Maastricht, The Netherlands
frans.oliehoek@maastrichtuniversity.nl

Matthijs T.J. Spaan
Delft University of Technology
Delft, The Netherlands
m.t.j.spaan@tudelft.nl

ABSTRACT

Multiagent POMDPs provide a powerful framework for optimal decision making under the assumption of instantaneous communication. We focus on a delayed communication setting (MPOMDP-DC), in which broadcast information is delayed by at most one time step. Such an assumption is in fact more appropriate for applications in which response time is critical. However, naive application of incremental pruning, the core of many state-of-the-art POMDP techniques, is intractable for MPOMDP-DCs. We overcome this problem by introducing a *tree-based pruning* technique. Experiments show that the method outperforms naive incremental pruning by orders of magnitude, allowing for the solution of larger problems.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Algorithms, Performance

Keywords

Multiagent planning under uncertainty, Multiagent POMDP, Delayed communication

1. INTRODUCTION

Planning under uncertainty in multiagent systems can be neatly formalized as a *decentralized partially observable Markov decision process (Dec-POMDP)*, but solving a Dec-POMDP is a complex (NEXP-complete) task. Communication can mitigate some of these complexities; by allowing agents to share their individual observations the problem reduces to a so-called *multiagent POMDP (MPOMDP)*, a special instance of the standard POMDP [3] which is 'merely' in PSPACE. However, this model requires the agents to perform full synchronization of their knowledge before selecting a next action, which is inappropriate in domains in which agents may need to act fast in response to their individual observations.

In this paper we focus on a class of problems where agents share their individual observations with a one step delay. That is, agents act using a *one step delayed sharing pattern*, resulting in an *MPOMDP with delayed communication (MPOMDP-DC)*. Solutions for such settings are also useful under longer delays [5]. Moreover, this class is particularly interesting, because it avoids the delay in action selection due to synchronization, while it is very similar to the standard POMDP. However, even though dynamic programming algorithms date back to the seventies [2], computational difficulties have limited the model's applicability.

The MPOMDP-DC value function is piecewise-linear and convex over the joint belief space [2], which is a property exploited by many regular POMDP solvers. However, *incremental pruning (IP)* [1], that performs a key operation, the so-called *cross-sum*, more efficiently, is not directly able to achieve the same improvements under delayed communication. A problem is the need to loop over a number of decision rules that is exponential both in the number of agents and in the number of observations.

In this paper, we target this additional complexity by proposing tree-based pruning with memoization, TBP-M, a method that operates over a tree structure in order to perform the cross-sum operation. Our experimental results indicate that it successfully avoids duplicate work by caching the result of computations at internal nodes and thus accelerates computation (at the cost of memory).

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.

2. MODEL

An MPOMDP consists of the following componenents: a finite set of n agents; a finite set of states $\mathcal{S}$; a set $\mathcal{A} = \{a^1, \ldots, a^{|\mathcal{A}|}\}$ of joint actions $a = \langle a_1,...,a_n\rangle$; a set $\mathcal{O} = \{o^1, \ldots, o^{|\mathcal{O}|}\}$ of joint observations $o = \langle o_1,...,o_n\rangle$; a transition and observation function that specify the probabilities $P^a(s'|s)$ and $O^a(o|s)$; a reward function that specifies the reward $R^a(s)$; and h is the (finite) horizon. An MPOMDP-DC is an MPOMDP where communication is received with a one-step delay. The joint policy $\pi = (\delta^0,\delta^1,\ldots,\delta^{h-1})$ in such settings is a sequence of joint decision rules that specify an individual decision rule $\delta^t = \langle \delta_1^t, \ldots, \delta_n^t\rangle$ for each agent. Each δ_i^t maps $\langle b^{t-1}, a^{t-1}, o_i^t\rangle$-tuples to individual actions a_i^t. The value of an MPOMDP-DC is a function of joint beliefs:

$$Q^t(b,a) = R^a_{\mathcal{B}}(b) + \max_{\beta} \sum_{o} P^a(o|b) Q^{t+1}(b', \beta(o)), \qquad (1)$$

where $\beta = \langle \beta_1, \ldots, \beta_n\rangle$ is a decentralized control law which the agents use to map individual observations to actions: $\beta(o) = \langle \beta_1(o_1), \ldots, \beta_n(o_n)\rangle$. That way we decompose δ^t into a collection of β, one for each $\langle b,a\rangle$-pair.

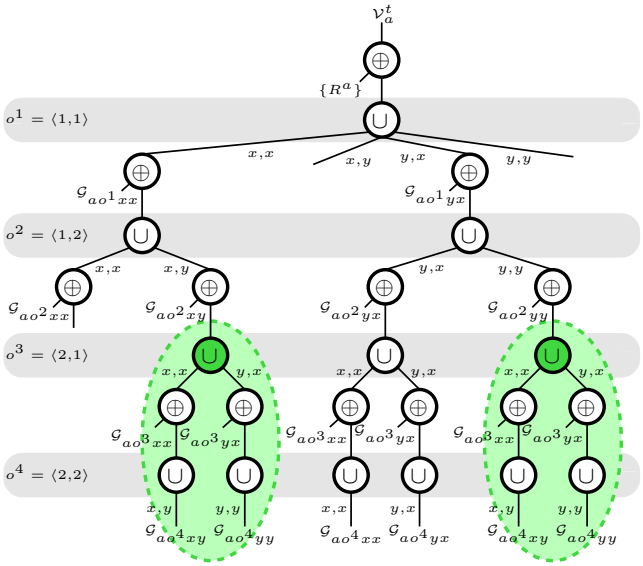

Figure 1: The computation tree of $\mathcal{V}_a^t$.

3. TREE-BASED PRUNING

As for an (M)POMDP, we can represent (1) using vectors [4]. However, in the MPOMDP-DC case not all combinations of next-stage vectors are possible; the actions they specify should be consistent with an admissible decentralized control law β. We can define 'back projected' vectors $g_{aoa'} \in \mathcal{G}_{aoa'}$ (see [4]). From these we construct the parsimonious representation

$$\mathcal{V}_a^t = \texttt{Prune} \bigcup_{\beta \in B} \left(\{R^a\} \oplus \mathcal{G}_{ao^1\beta(o^1)} \oplus \cdots \oplus \mathcal{G}_{ao^{|\mathcal{O}|}\beta(o^{|\mathcal{O}|})} \right) \quad (2)$$

where the cross-sum $A \oplus B = \{a + b \mid a \in A, b \in B\}$.

A naive way of performing incremental pruning (IP) [1] is to perform IP for each β. Their number, however, is exponential both in the number of agents and in the number of observations. Moreover, this method performs a lot of duplicate work. E.g., there are many β that specify $\beta(o^1) = a^k, \beta(o^2) = a^l$, but for each of them $\texttt{Prune}(\mathcal{G}_{ao^1a^k} \oplus \mathcal{G}_{ao^2a^l})$ is recomputed. In order to overcome these drawbacks, we propose a different approach: for each β, we directly construct construct the parsimonious representation via a computation tree.

In particular, it is possible to interpret β as a vector of joint actions, $\langle a_{(1)} \dots a_{(|\mathcal{O}|)} \rangle$, where $a_{(j)}$ denotes the joint action selected for the j-th joint observation. This allows us to decompose the union over β into dependent unions over joint actions, resulting in the computation tree illustrated in Fig. 1 for a fictitious 2-action (x and y) 2-observation (1 and 2) MPOMDP-DC. The root of the tree, $\mathcal{V}_a^t$, is the result of the computation. There are two 2 types of internal, or operator, nodes: cross-sum and union. All the leafs are sets of vectors. An operator node n takes as input the sets from its children, and propagates the result up to its parent. The j-th union node on a path from root to leaf performs the union $\cup_{a_{(j)}}$ and thus has children corresponding to different assignments of a joint action to o^j (indicated by the gray bands). It is important to realize that the options available for $a_{(j)}$ depend on the action choices $\left(a_{(1)}, \dots, a_{(j-1)}\right)$ made higher up in the tree; given those earlier choices, some $a_{(j)}$ may lead to conflicting individual actions for the same individual observation.

Problem(h)	TBP-M	NAIVE IP	TBP-NOM
Dec-Tiger(5)	0.13	0.23	0.09
Dec-Tiger(15)	0.98	2.54	1.19
OneDoor(3)	53.64	304.72	56.53
GridSmall(2)	3.93	64.03	3.80
MG2x2(2)	171.07	382093.00	516.03
MG2x2(4)	1115.06		2813.10
D-T Creaks(2)	63.14	109.27	121.99
D-T Creaks(5)	286.53	8277.32	2046.73
Box Push.(2)	132.13	1832.98	1961.38

Table 1: Timing results (in s).

Now, to compute $\mathcal{V}_a^t$ we propose *tree-based (incremental) pruning (TBP)*: it expands the computation tree and, when the results are being propagated to the top of the tree, it prunes dominated vectors at each internal node. However, Fig. 1 shows another important issue: there are identical sub-trees in this computation tree, as indicated by the dashed green ovals, which means that we would be doing unnecessary work. We address this problem by memoization, i.e., caching of intermediate results, and refer to the resulting method as TBP-M.

Table 1 shows timing results for six benchmark problems, for a set of planning horizons (depending on the problem). We can see that for all domains TBP-M outperforms NAIVE IP, often by an order of magnitude and up to 3 orders of magnitude. We also compared against TBP-NOM: a strawman version of TBP-M that does not perform any memoization and re-computes duplicate parts of the tree. It allows us to see the effect of tree-based pruning, without the extra speedups provided by memoization: memoization significantly speeds up computations.

4. CONCLUSIONS

We addressed the problem of the additional complexity that the MPOMDP-DC backup exhibits over the backup for the MPOMDP. We showed that the DC backup operator can be represented as a computation tree and presented TBP-M, a method to exploit this tree structure. An empirical evaluation on a number of benchmark problems that indicates that TBP-M can realize speedups of 3 orders of magnitude over the naive IP baseline.

Acknowledgments

Research supported in part by AFOSR MURI project #FA9550-09-1-0538 and NWO CATCH project #640.005.003. M.S. is funded by the FP7 Marie Curie Actions Individual Fellowship #275217 (FP7-PEOPLE-2010-IEF).

5. REFERENCES

[1] A. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *UAI*, pages 54–61. Morgan Kaufmann, 1997.

[2] K. Hsu and S. Marcus. Decentralized control of finite state Markov processes. *IEEE Transactions on Automatic Control*, 27(2):426–431, Apr. 1982.

[3] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[4] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis. Dec-POMDPs with delayed communication. In *MSDM (AAMAS Workshop)*, May 2007.

[5] M. T. J. Spaan, F. A. Oliehoek, and N. Vlassis. Multiagent planning under uncertainty with stochastic communication delays. In *ICAPS*, pages 338–345, 2008.