A Multiagent Variant of Dyna-Q

Gerhard Weiß
Institut für Informatik, Technische Universität München
D-80290 München, Germany, weissg@in.tum.de

Abstract

This paper describes a multiagent variant of Dyna-Q called M-Dyna-Q. Dyna-Q is an integrated single-agent framework for planning, reacting, and learning. Like Dyna-Q, M-Dyna-Q employs two key ideas: learning results can serve as a valuable input for both planning and reacting, and results of planning and reacting can serve as a valuable input to learning. M-Dyna-Q extends Dyna-Q in that planning, reacting, and learning are jointly realized by multiple agents.

1 Introduction

Dyna-Q (e.g., [1] and [2, Chapter 9]) is a single-agent framework that integrates planning and reacting on the basis of learning. This integration is based on two key ideas:

- Learning results can serve as a valuable basis for both planning and reacting. Through learning the agents acquire information that makes it possible for them to plan and react more effectively and efficiently. More specifically, according to Dyna-Q the agents plan on the basis of an incrementally learnt world model and they react on the basis of incrementally learnt values that indicate the usefulness of their potential actions.
- Results of both planning and reacting can serve as a valuable basis for learning. The agents use the outcomes of their planning and reacting activities for improving their world model and the estimates of their actions' usefulness. More specifically, planning constitutes a basis for trial-and-error learning from hypothetical experience, while reacting at the same time constitutes a basis for trial-and-error learning from real experience.

This paper describes how the Dyna-Q framework can be extended to and applied in multiagent settings. This extension, called M-Dyna-Q, keeps to the key ideas underlying Dyna-Q, but goes beyond the single-agent setting by considering planning, reacting, and learning as processes that are jointly realized by multiple agents.

2 The M-Dyna-Q Framework

According to the M-Dyna-Q framework the overall multiagent activity results from the repeated execution of a basic working cycle consisting of two major joint activities, namely, action selection and learning. Each cycle runs either in real or hypothetical mode, where the agents synchronously switch between the two modes at a fixed and predefined rate. The real mode corresponds to (fast) "reactive behavior," whereas the hypothetical mode corresponds to (slower) "plan-based behavior." During action selection, the agents jointly decide what action should be carried out next (resulting in the next real or a new hypothetical state); this decision is made on the basis of the agents' distributed value function in the case of operating in the real mode, and on the basis of the agents' joint world model in the case of operating in the hypothetical mode. During learning the agents adjust both their world model and their value function if they act in the real mode, and just their world model if they act in the hypothetical mode. Below these two major activities are described in detail.

In the remaining the following simple notation is used and the following elementary assumptions are made. $\mathcal{A}g = \{A_1, \ldots, A_n\}$ $(n \in \mathbb{N})$ denotes the finite set of agents available in the MAS under consideration. The environment in which the agents act can be described as a discrete state space, and the individual real and hypothetical states are denoted by $\mathcal{S}, \mathcal{T}, \mathcal{U}, \ldots$ $\mathcal{A}c_i^{poss} = \{a_i^1, \ldots, a_i^{m_i}\}$ $(m_i \in \mathbb{N})$ denotes the set of all possible actions of the agent A_i, and is called its *action potential*. Finally, $\mathcal{A}c_i^{poss}[\mathcal{S}]$ denotes the set of all actions that A_i could carry out (identifies as "executable") in the environmental state $\mathcal{S}$.

Joint Action Selection. According to M-Dyna-Q each agent A_i maintains state-specific estimates of the usefulness of its actions for goal attainment. More specifically, an agent A_i maintains, for every state $\mathcal{S}$ and each of its actions a_i^j, a quantity $Q_i^j(\mathcal{S})$ that expresses its estimate of a_i^j's state-specific usefulness with respect to goal attainment. Based on these estimates, action selection works as follows. If the agents operate in the "real mode", then they analyze the current real state $\mathcal{S}$, and each agent A_i iden-

tifies and announces the set $\mathcal{A}c_i^{poss}[\mathcal{S}]$ of actions it could carry out immediately (assuming the availability of a standard blackboard communication structure and a time-out announcement mechanism). The identification of its potentential actions can be done by each agent independent of the other agents and on the basis of its own view of and knowledge about $\mathcal{S}$; in particular, action identification can be done concurrently by the agents. The action to be carried out is then selected among all announced actions dependent on the agents' action selection policy. A standard policy (which was also used in the experiments reported below) is that the probability of selecting an announced action a_i^j is proportional to the estimated usefulness of all actions announced in $\mathcal{S}$, i.e.,

$$\frac{e^{Q_i^j(\mathcal{S})}}{e^{\sum_{a_i^j} Q_i^j(\mathcal{S})}} \quad (1)$$

where the sum ranges over all currently announced actions (i.e., over $\bigcup_{i=1}^{n} \mathcal{A}c_i^{poss}[\mathcal{S}]$). If the agents operate in the "hypothetical mode," then they *(i)* randomly choose an environmental state $\mathcal{S}$ from those real states which they already encountered in the past and *(ii)* select an action a_i^j from those already carried out in this state according to (1). This means that in the hypothetical mode the agents simulate real activity and do as if $\mathcal{S}$ is the current real state and a_i^j had been selected for execution. Because the agents only choose hypothetical states that they already know from experience, they avoid to be forced to make speculative activity decisions under unknown hypothetical environmental circumstances. Note that the agents do single-step planning when operating in the hypothetical mode. This "planning in very small steps" has been adopted from the single-agent Dyna-Q framework with the intention to enable the agents to redirect their course of activity without unnecessarily wasted computation and communication whenever necessary.

Joint Learning. Learning is realized by the agents through adjusting the estimates of their actions' usefulness. Suppose that a_i^j has been selected in the real or hypothetical state $\mathcal{S}$ and $\mathcal{T}$ is the resulting successor state. All agents that could carry out actions in $\mathcal{T}$ inform the agent A_i about these actions' estimated usefulness. A_i determines the maximum

$$maxQ_k^l(\mathcal{T}) \quad =_{\text{def}} \quad \max\{Q_k^l(\mathcal{T}) : a_k^l \text{ is executable in } \mathcal{T}\} \quad (2)$$

of these estimates and adjusts its estimate $Q_i^j(\mathcal{S})$ according to

$$Q_i^j(\mathcal{S}) \quad = \quad Q_i^j(\mathcal{S}) + \alpha \cdot [R + \beta \cdot maxQ_k^l(\mathcal{T}) - Q_i^j(\mathcal{S})] \quad (3)$$

where R is the external reward (if any) and α and β are small constants called learning rates. ($maxQ_k^l(\mathcal{T})$ defines, so to say, the global value of the state $\mathcal{T}$.) This adjustment rule can be viewed as a straightforward multiagent realization of standard single-agent Q-learning [3] in which the individual Q-values and thus the value function is distributed over and maintained by several agents. Whereas the adjustment rule is applied in both the real and the hypothetical mode, the world model is updated by the agents only if they act in the real mode; this is reasonable because the most reliable way to improve a world model obviously is to observe the effects of real actions.

3 Concluding Remarks

We made initial experiments with several synthetic state-action spaces. The results of these experiments show that M-Dyna-Q leads to a robust performance improvement over a variety of parameter settings. Some of the results are described in [4] (see http://wwwbrauer.in.tum.de/cgi-bin/make-fki-list.perl).

M-Dyna-Q integrates joint planning, joint reacting, and joint learning within a single multiagent framework. It is this integration that makes the M-Dyna-Q framework very flexible and different from a number of related approaches, including approaches that rely on a combination of planning and learning as well as approaches that rely on a combination of reacting and learning. M-Dyna-Q in its current form is a rather straightforward multiagent realization of Dyna-Q that still shows several limitations. First, and most important, M-Dyna-Q requires the agents to strictly synchronize their action selection and learning activities. In particular, it requires that only one action is carried out in every working cycle, which means that the agents carry out their activities sequentially. Second, in the current form of M-Dyna-Q the planning depth is fixed to one. Although this makes sense in a variety of situations (especially in unknown environments), it is desirable that in general the agents handle the planning depth more flexibly. And third, M-Dyna-Q in its basic form described here implicitly assumes that the agents can maintain a joint world model without remarkable efforts. This is not the case, however, in domains in which the agents are not aware of all relevant effects of their actions or in which they sense different parts of their shared environment. These limitations require further research efforts.

References

[1] R. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *SIGART Bulletin*, 2:160–163, 1991.

[2] R. Sutton and A. Barto. *Reinforcement Learning. An Introduction*. MIT Press/A Bradford Book, Cambridge, MA, 1998.

[3] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge University, 1989.

[4] G. Weiß. A multiagent framework for planning, reacting, and learning. Technical Report FKI-233-99, Institut für Informatik, Technische Universität München, 1999.