

# Planning and Learning in Environments with Delayed Feedback

Thomas J. Walsh, Ali Nouri, and Lihong Li  
Rutgers University, Department of Computer Science  
{thomaswa,nouri,lihong}@cs.rutgers.edu

We consider planning and learning in a class of sequential decision-making problems where state and reward observations are delayed by a constant number of timesteps.<sup>1</sup> A real world example in this setting is piloting the Mars Rover from Earth, where direct control of the agent is limited by the vast communication latency. Even on terra firma, agents that receive observations over networks, or do expensive processing of observations (such as image processing), will experience delay between *observing* the environment, and *acting* based on this information. We provide formal definitions of planning and learning problems in this setting, and then develop and test theoretically guaranteed algorithms for solving these problems.

We extend the standard Markov Decision Process (MDP) formalism [3] (defined by the tuple  $\langle S, A, P, R, \gamma \rangle$ ) to a model called the CDMDP (Constant Delayed MDP), which includes a parameter ( $k$ ) indicating how many timesteps will elapse before the agent perceives the “current” state and reward observations. A CDMDP *policy*,  $\pi$ , is a mapping from histories to actions and has an associated *value function*, representing its expected cumulative discounted reward. Prior work [2] has shown that CDMDP policies need only consider a limited window of history, specifically, the latest observation  $s$  and the last  $k$  actions  $a_1, \dots, a_k$ . Formally, we call  $I_k = \{s, a_1 \dots a_k\}$  an *information state*. The *CDMDP planning problem* is defined as: given a CDMDP, initial information state  $I_k^0$ , and a reward threshold  $\theta$ , determine whether a policy exists that achieves an expected discounted reward of at least  $\theta$ . In the *CDMDP learning problem*, an agent deployed in a delayed-feedback environment knowing only  $S, A, \gamma$ , and  $k$  is tasked with finding an optimal policy for the environment online.

Several techniques are commonly used in the presence of observation delays. One is the **wait** strategy, which “waits”  $k$  steps to perceive the current state before taking real actions. Unfortunately, this strategy leads to suboptimal reward for an agent that can make better use of its time. Another simple, but suboptimal, approach is the **memoryless** strategy, where an agent blindly ignores the observation delay; specifically, it solves the undelayed version of the CDMDP to get  $\pi^*(s)$  and then follows the policy  $\pi(I_k) = \pi^*(s)$ . In contrast, the **augmented** approach [2], constructs and solves an MDP where each state is one of the information states ( $I_k$ ). This technique is guaranteed to return an optimal policy. Unfortunately, we prove that the exponential blowup in the state space induced by this strategy is unavoidable in modeling general CDMDPs as regular MDPs, and that the CDMDP planning problem itself is NP-Hard.

In light of these results, we focus on the following special cases, motivated by real world assumptions, where we show that the exponential blowup can be avoided with bounded loss on the accuracy of the optimal value function: (I) Deterministic finite MDP, (II) Deterministic continuous MDP, (III) Mildly stochastic finite MDP, where  $\forall s \exists s' P(s, a, s') \geq 1 - \delta$  for some fixed, small  $\delta > 0$ , and (IV) Bounded-noise continuous MDP, whose transitions are governed by  $s_{t+1} = T(s_t, a_t) + w_t$  where  $T$  is a deterministic transition function and  $w_t$  is bounded noise with  $\|w_t\|_\infty \leq \Delta$  for some fixed, small  $\Delta > 0$ . To solve the CDMDP planning problem in these

---

<sup>1</sup>This work first appeared in the Eighteenth European Conference on Machine Learning [5].

four settings, we introduce a new algorithm, Model Based Simulation (MBS).

MBS constructs a deterministic version of the delayed environment, using the most likely (finite MDPs) or expected (continuous MDPs) one-step outcome of a state/action pair as the next state. MBS then simulates the previous  $k$  steps (from the last known state) in this deterministic model to estimate the current state and acts greedily with respect to the deterministic model. We can bound the error in the value function of the deterministic model as  $\frac{\gamma \delta R_{\max}}{(1-\gamma)^2}$  (case III,  $R_{\max}$  is the maximum reward) and  $\frac{2\gamma C_V \Delta}{1-\gamma} + \epsilon$  (Case IV, with  $C_V$  as a Lipschitz constant of the optimal CDMDP value function and  $\epsilon$  as the error introduced in solving for the optimal value function of a continuous MDP). Thus, MBS can answer the CDMDP planning problem in Cases III and IV with these accuracies, in polynomial time, in addition to the time required to solve (approximately) the deterministic MDP. We note that these results imply that MBS solves the CDMDP planning problem exactly in case I and with only error  $\epsilon$  in case II.

We empirically tested MBS in four benchmark domains, one for each special case, against the strategies discussed earlier. These testbeds were learning domains so we extended the planning strategies in the following ways. In finite-state domains, MBS was used with R-max [1] and in continuous domains we introduced a model-based reinforcement learning algorithm, Model Parameter Approximation (MPA), which trains a function approximator to model state transitions. The trained function approximator was used for the simulation step of MBS. The memoryless strategy was implemented in a number of ways, including R-max, MPA, Sarsa( $\lambda$ ) [4], and a Batch version of Sarsa( $\lambda$ ). The augmented approach was deployed using both R-max and Sarsa( $\lambda$ ) in the expanded state space. A “compact” version that learned a one-step model and then built the expanded space was also tested. The wait agent was coupled with R-max or MPA as appropriate. In all the experiments, MBS outperformed all the other algorithms studied. While the “compact” augmented approach performed well in terms of sample complexity, all the augmented algorithms became intractable as delay increased. The memoryless learners were often unable to represent the optimal policies, though eligibility traces helped to some extent. A complete analysis is provided elsewhere [5]. Both the theoretical and empirical analyses indicate that MBS is an efficient, near-optimal strategy for planning and learning in several practically motivated classes of environments with delayed feedback.

## References

- [1] Ronen I. Brafman and Moshe Tennenholtz. R-max—a general polynomial time algorithm for near-optimal reinforcement learning. *JMLR*, 3:213–231, October 2002.
- [2] Konstantinos V. Katsikopoulos and Sascha E. Engelbrecht. Markov decision processes with delays and asynchronous cost collection. *IEEE Transactions on Automatic Control*, 48:568–574, 2003.
- [3] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, 1994.
- [4] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, March 1998.
- [5] Thomas J. Walsh, Ali Nouri, Lihong Li, and Michael L. Littman. Planning and learning in environments with delayed feedback. In *ECML-07*, pages 442–453, 2007.