



# Planning and Learning in Environments with Delayed Feedback

Thomas J. Walsh, Ali Nouri, Lihong Li, Michael L. Littman

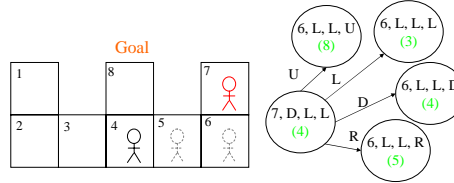
Rutgers Laboratory for Real Life Reinforcement Learning  
Computer Science Department, Rutgers University, Piscataway NJ

## Abstract

This work considers the problems of planning and learning in environments with constant observation and reward delays. We provide a hardness result for the general planning problem and positive results for several special cases with deterministic or otherwise constrained dynamics. We present an algorithm, Model Based Simulation, for planning in such environments and use model-based reinforcement learning to extend this approach to the learning setting in both finite and continuous environments. Empirical comparisons show this algorithm holds significant advantages over others for decision making in delayed environments.

## General Planning Theory

**Augmented Agent** - make 1 state for each  $\{s, a_1 \dots a_k\}$ . This approach is guaranteed to garner the optimal policy.



**Theorem 1:** The smallest regular MDP induced by a finite CDMDP can have a lower bound of  $|S'| = \Omega(|A|^k)$ .

**Theorem 2:** The general CDMDP planning problem is **NP-Hard**.

## Four Special Cases

**Case I:** Deterministic transitions, finite state.

**Case II:** Deterministic transitions, continuous state

**Case III:** "Mildly" stochastic trans., finite state:  $P(s, a, s') \geq 1 - \delta$

**Case IV:** Bounded-noise stochastic transitions, continuous state:

$$s_{t+1} = T(s_t, a_t) + w_t, \quad \|w_t\| \leq \Delta$$

## New Algorithms and Loss Bounds

### Model Based Simulation (MBS) - a CDMDP Planning Algorithm

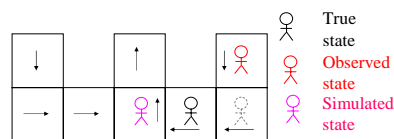
- Construct a deterministic version ( $M'$ ) of the *undelayed* MDP
  - use the most likely one-step outcome in the finite-state case
  - use the expected one-step outcome for continuous domains

• Find  $V^*$  and  $\pi^*$  for  $M'$

•  $s \leftarrow a_1 \dots a_k$  simulated from  $s$  in  $M'$

• Return  $\pi^*(s)$

**This may falter in stochastic domains:**

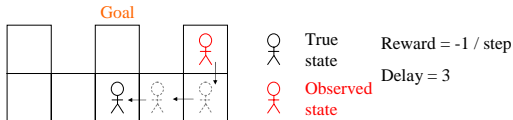


## The Constant Delay MDP (CDMPD)

A CDMDP is an extension of the standard Markov Decision Process (MDP) with state observations and rewards delayed by  $k$  timesteps.

**CDMDP Planning:** Given a CDMDP, starting information state  $I_k^0$ , and reward bound  $\theta$ , determine whether a policy exists that achieves expected discounted reward of  $\theta$  from  $I_k^0$

**CDMDP Learning:** Give an agent in a delayed environment knowing only  $S, A, \gamma$ , and  $k$ , find the optimal policy through experience.



## Simple but Ineffective Approaches

**Wait Agent** - wait  $k$  steps before acting.

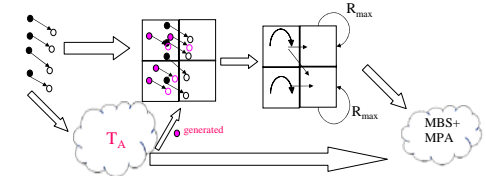
**Memoryless Agent** - use current observation. This approach can be enhanced with eligibility traces, but can still be suboptimal.

Case	$\ V^* - V^*\ $	Time
Deterministic Finite	0	$P( C )$
Deterministic Continuous	$\epsilon$	$P( C ) + T$
Mildly Stochastic (finite)	$\frac{\gamma \delta R_{\max}}{(1-\gamma)^2}$	$P( C )$
Bounded Noise (Continuous)	$\frac{2\gamma C_V \Delta}{1-\gamma} + \epsilon$	$P( C ) + T$

**Error bounds and computation time** for solving the CDMDP planning problem in the four special cases with MBS.

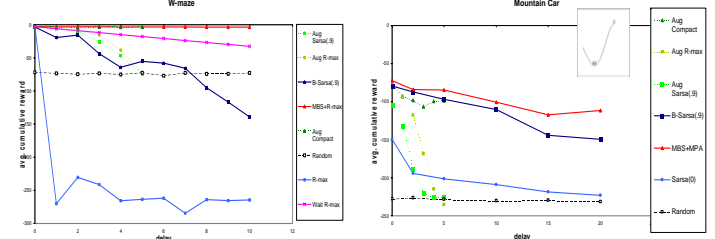
- $\epsilon$  - Error inherent in approximating  $V^*$  for a continuous MDP
- $C_V$  - Lipschitz constant for  $V^*$
- $|C|$  - Size of the input CDMDP
- $T$  - Time to produce the  $\epsilon$ -accurate continuous MDP value function.

**Model Parameter Approximation (MPA)** is used to extend MBS to the learning setting in continuous domains (in finite state environments,  $R$ -Max suffices).

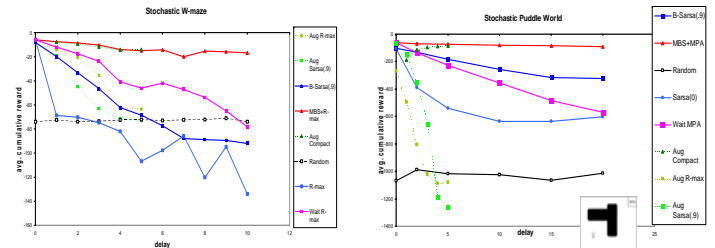


## Experiments

- Criteria: Cumulative reward over 300 steps for 200 episodes (10 runs)
- Competitors: MBS+(R-max/MPA), Wait+(R-max/MPA) R-max, MPA, Sarsa( $\lambda$ ), Batch-Sarsa( $\lambda$ ), Augmented (with a special compact version with low sample complexity).



Above: Experimental results in deterministic domains, W-maze (Case I, left) and Mountain Car (Case II, right)



Above: Experimental results in stochastic domains, W-maze (Case III, left) and Puddle World (Case IV, right)