Relative Entropy Policy Search (REPS)

Amber Srivastava
Sreenath Sundar

Outline

Problem Statement

Obtain policies $\{\pi(a|s)\}$ such that

- The Expected reward $J(\pi)$ gets maximized, and
- The information loss is bounded.

Optimization problem :

$$\text{maximize}_{\pi,\mu^\pi} J(\pi) = \sum_{s,a} \mu^\pi(s)\pi(a|s)r(s,a)$$

$$\text{subject to } \epsilon \geq \sum_{s,a} \mu^\pi(s)\pi(a|s)\frac{\mu^\pi(s)\pi(a|s)}{q(s,a)}$$

$$\sum_{s'} \mu^\pi(s')\phi_{s'} = \sum_{s,a,s'} \mu^\pi(s)\pi(a|s)p(s'|s,a)\phi_{s'}$$

$$1 = \sum_{s,a} \mu^\pi(s)\pi(a|s)$$

Problem Solution

The corresponding Lagrangian

$$L = \Big(\sum_{s,a} p_{sa} r(s,a)\Big) + \eta\Big(\epsilon - \sum_{s,a} p_{sa} \log \frac{p_{sa}}{q(s,a)}\Big)$$
$$+ \sum_{s'} \theta^T \Big(\sum_{s,a} p_{sa} p(s'|s,a)\phi_{s'} - \sum_{a'} p_{s'a'}\phi_{s'}\Big) + \lambda\Big(1 - \sum_{s,a} p_{sa}\Big)$$

Problem Solution

The corresponding Lagrangian

$$L = \Big(\sum_{s,a} p_{sa} r(s,a)\Big) + \eta\Big(\epsilon - \sum_{s,a} p_{sa} \log \frac{p_{sa}}{q(s,a)}\Big)$$
$$+ \sum_{s'} \theta^T \Big(\sum_{s,a} p_{sa} p(s'|s,a)\phi_{s'} - \sum_{a'} p_{s'a'}\phi_{s'}\Big) + \lambda\Big(1 - \sum_{s,a} p_{sa}\Big)$$
$$= \sum_{s,a} p_{sa}\Big(r(s,a) - \eta \log \frac{p_{sa}}{q(s,a)} - \lambda - \theta^T \phi_s + \sum_{s'} p(s'|s,a)\theta^T \phi_{s'}\Big)$$
$$+ \eta\epsilon + \lambda$$

Problem Solution

The corresponding Lagrangian

$$
\begin{aligned}
L &= \Big(\sum_{s,a} p_{sa} r(s,a)\Big) + \eta\Big(\epsilon - \sum_{s,a} p_{sa} \log \frac{p_{sa}}{q(s,a)}\Big) \\
&\quad + \sum_{s'} \theta^T \Big(\sum_{s,a} p_{sa} p(s'|s,a)\phi_{s'} - \sum_{a'} p_{s'a'}\phi_{s'}\Big) + \lambda\Big(1 - \sum_{s,a} p_{sa}\Big) \\
&= \sum_{s,a} p_{sa}\Big(r(s,a) - \eta \log \frac{p_{sa}}{q(s,a)} - \lambda - \theta^T \phi_s + \sum_{s'} p(s'|s,a)\theta^T \phi_{s'}\Big) \\
&\quad + \eta\epsilon + \lambda
\end{aligned}
$$

- By setting $\frac{\partial L}{\partial p_{sa}} = 0$ obtain $\pi(a|s) = \frac{q(s,a)e^{\frac{1}{\eta}\delta(\theta,s,a)}}{\sum_{a'} q(s,a')e^{\frac{1}{\eta}\delta(\theta,s,a')}}$.
- Obtain θ_*, η_* by minimizing the dual
  $g(\theta,\eta) = \eta \log\Big(\sum_{s,a} q(s,a)\exp\big(\epsilon + \frac{1}{\eta}\delta(\theta,s,a)\big)\Big)$

Algorithm

Input: features $\phi(s)$, maximal information loss ϵ, and initial policy $\phi_0(a|s)$
while $k <$ *maximum policy updates* **do**
Sampling: Obtain N samples (s_i, a_i, s_i', r_i)
Optimization:

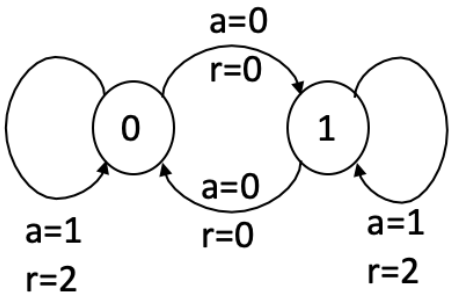
$$g(\theta, \eta) = \eta \log \left(\frac{1}{N} \sum_{i=1}^{N} e^{\epsilon + \frac{1}{\eta} \delta(\theta, s_i, a_i)} \right)$$

$$\theta_*, \eta_* = \arg\max_{\theta,\eta} \; g(\theta, \eta)$$

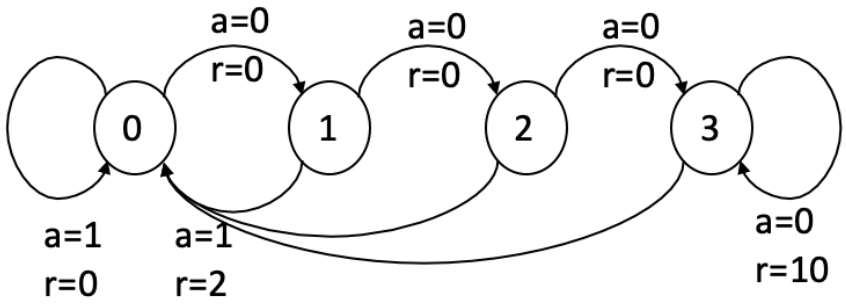
$$\pi_{k+1}(a|s) = \frac{\pi_k(a|s) e^{\frac{1}{\eta_*} \delta(\theta_*, s, a)}}{\sum_{a'} \pi_k(a'|s) e^{\frac{1}{\eta_*} \delta(\theta_*, s, a')}}$$

end

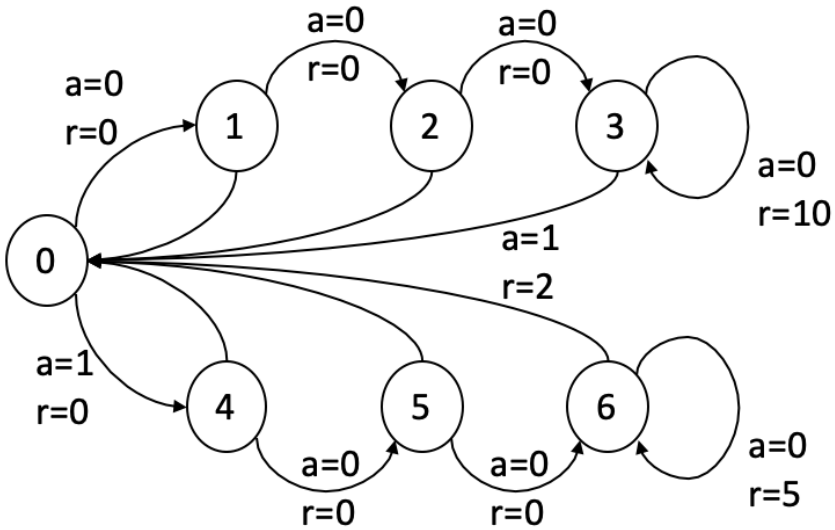
Environments



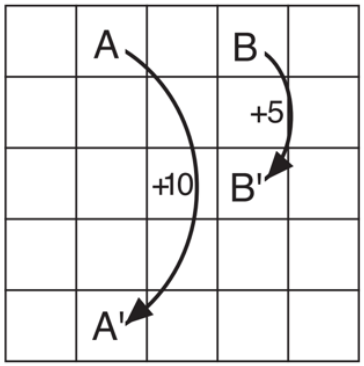
Two state



Single Chain



Double Chain



Gridworld

Simulations

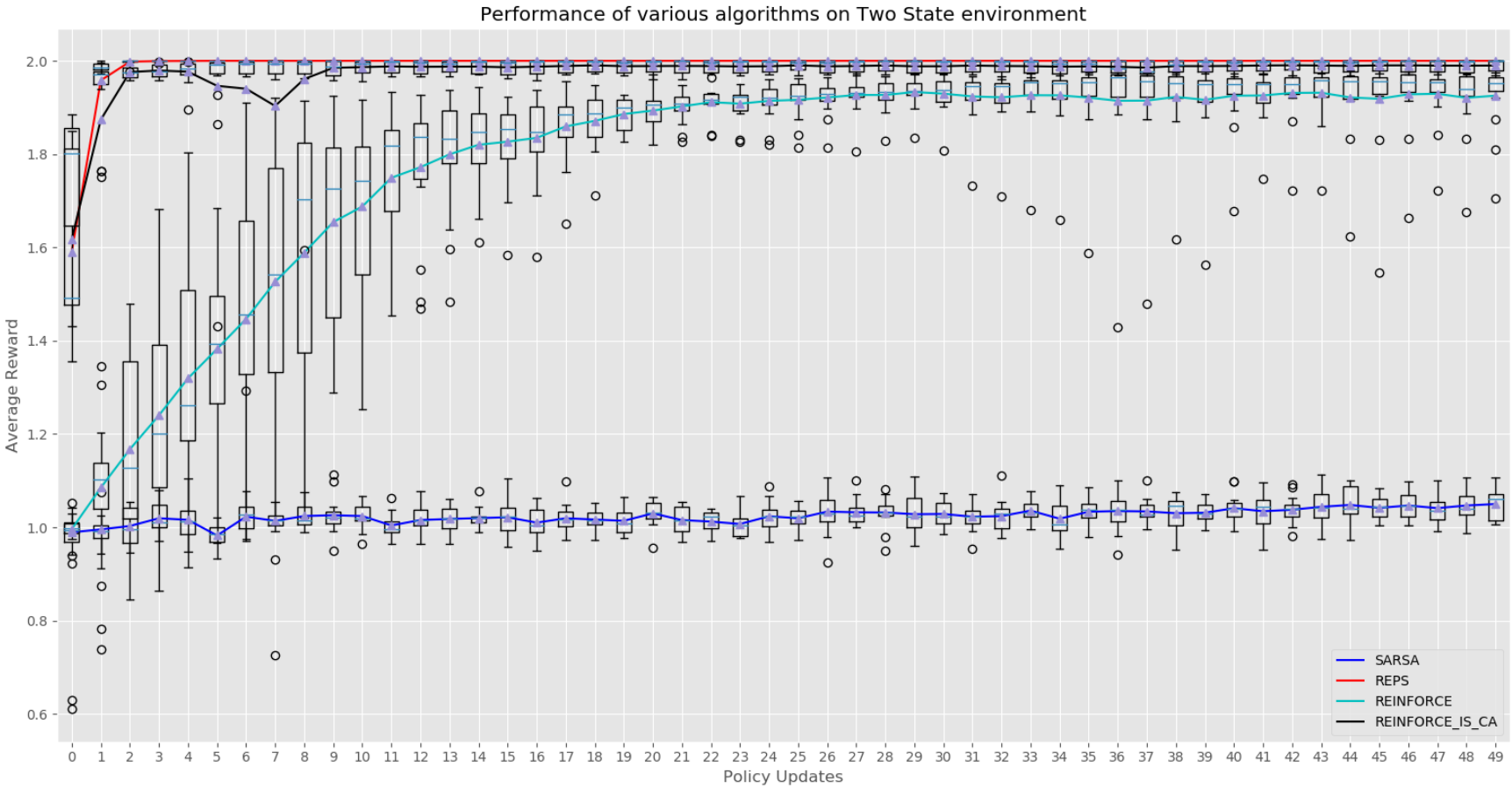


Figure 1: Two State Problem

Simulations

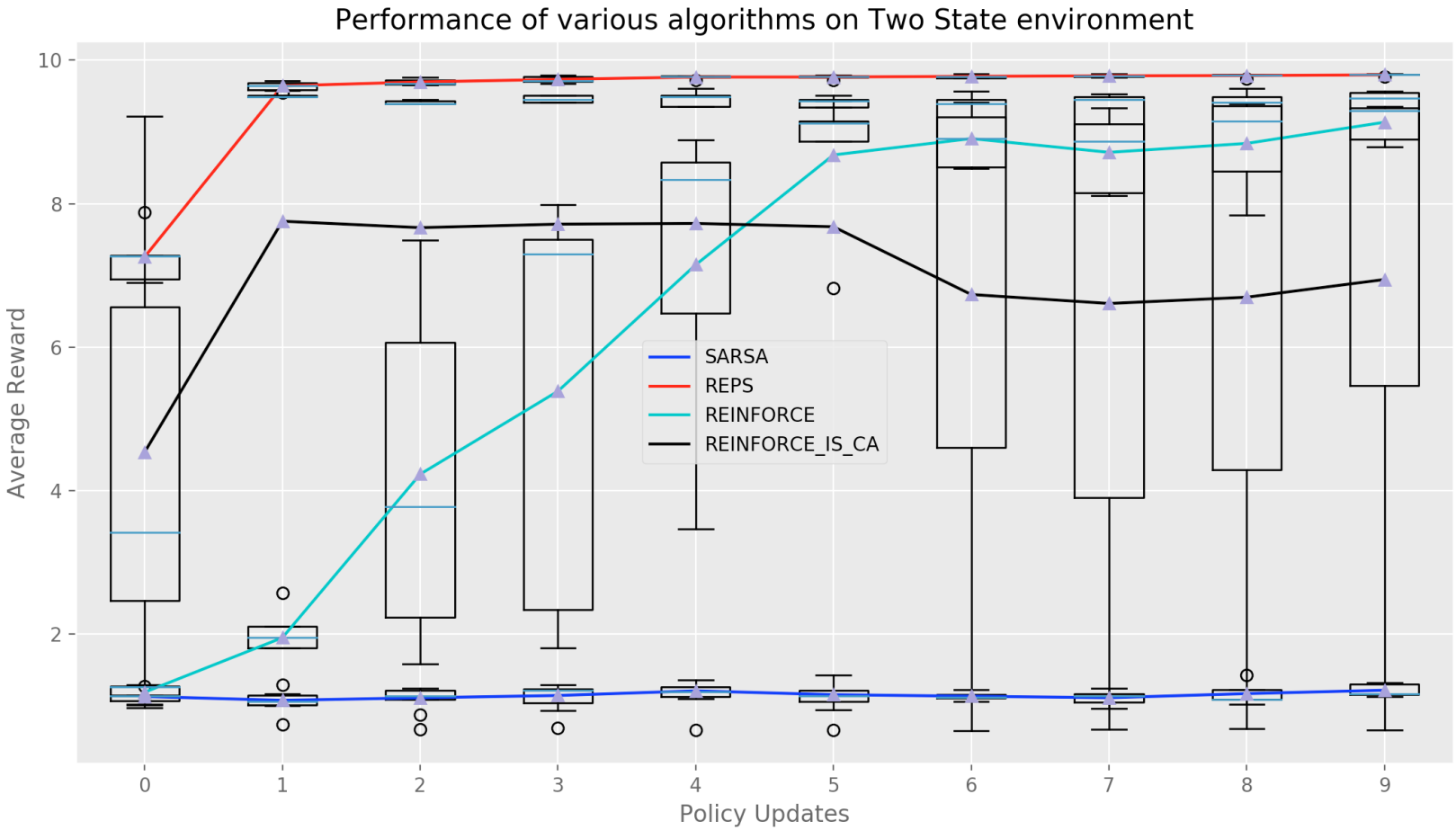


Figure 2: Single Chain Problem

Simulations

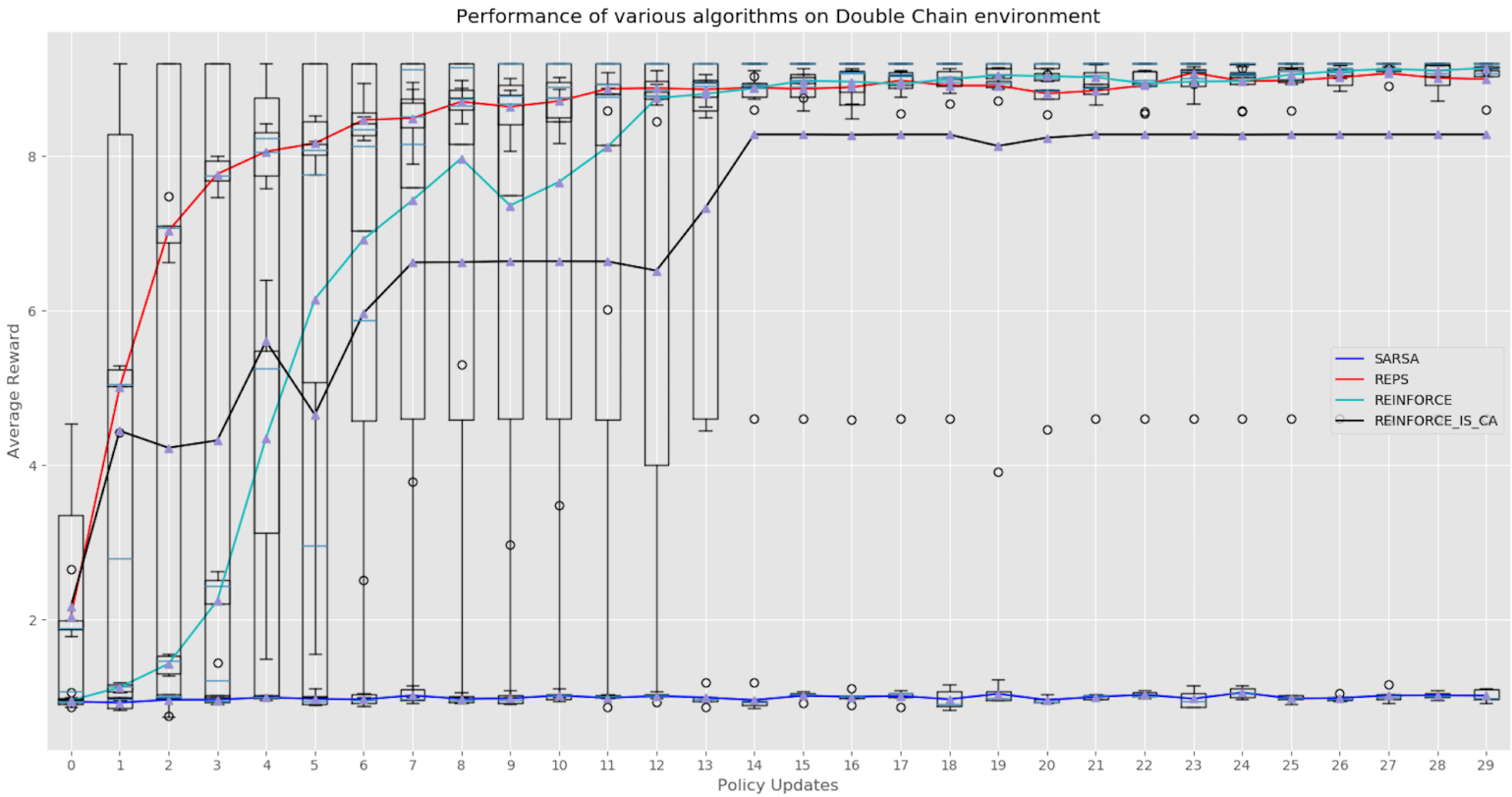


Figure 3: Double Chain Problem

Simulations

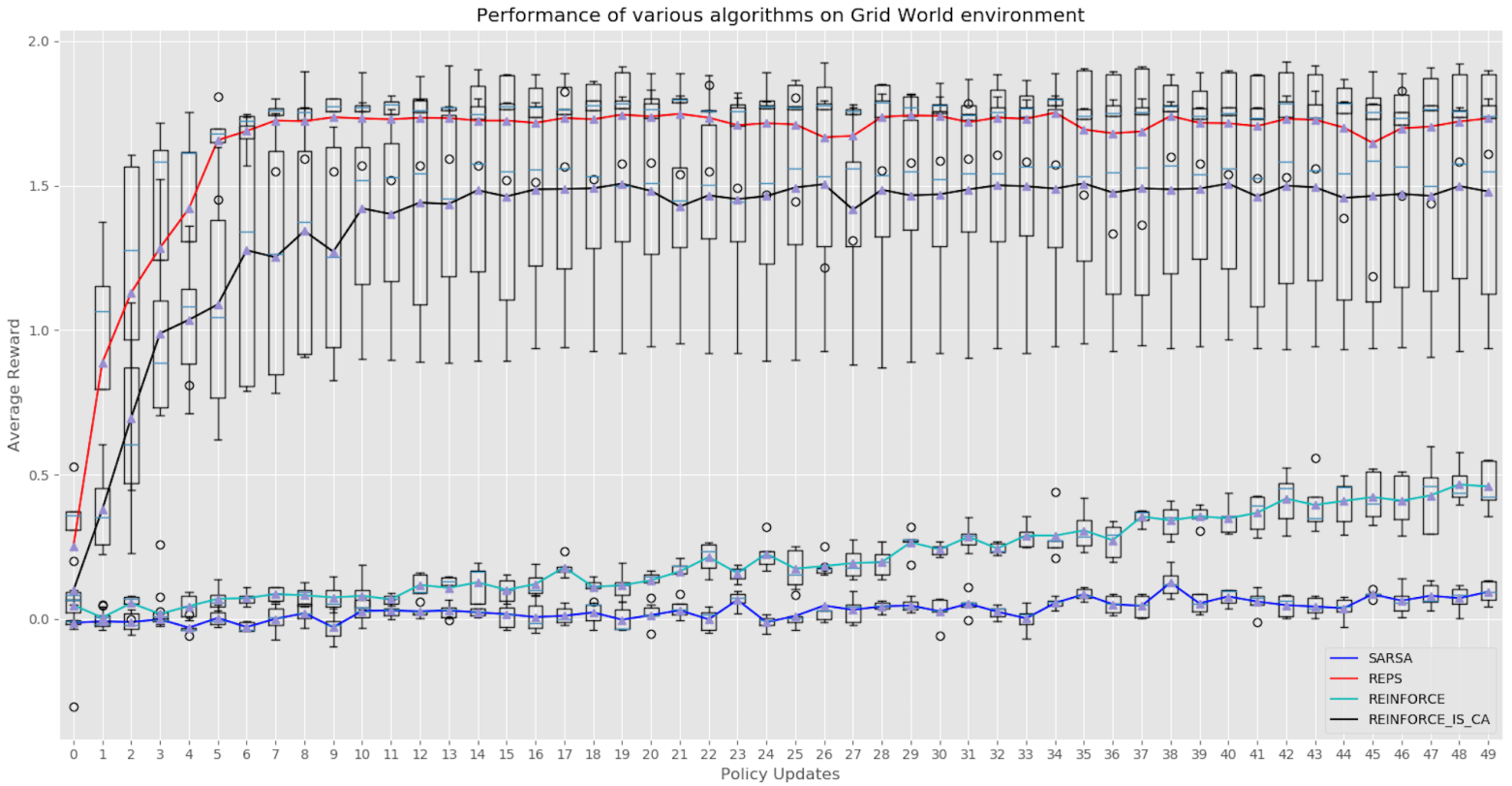


Figure 4: Gridworld Environment

Simulations

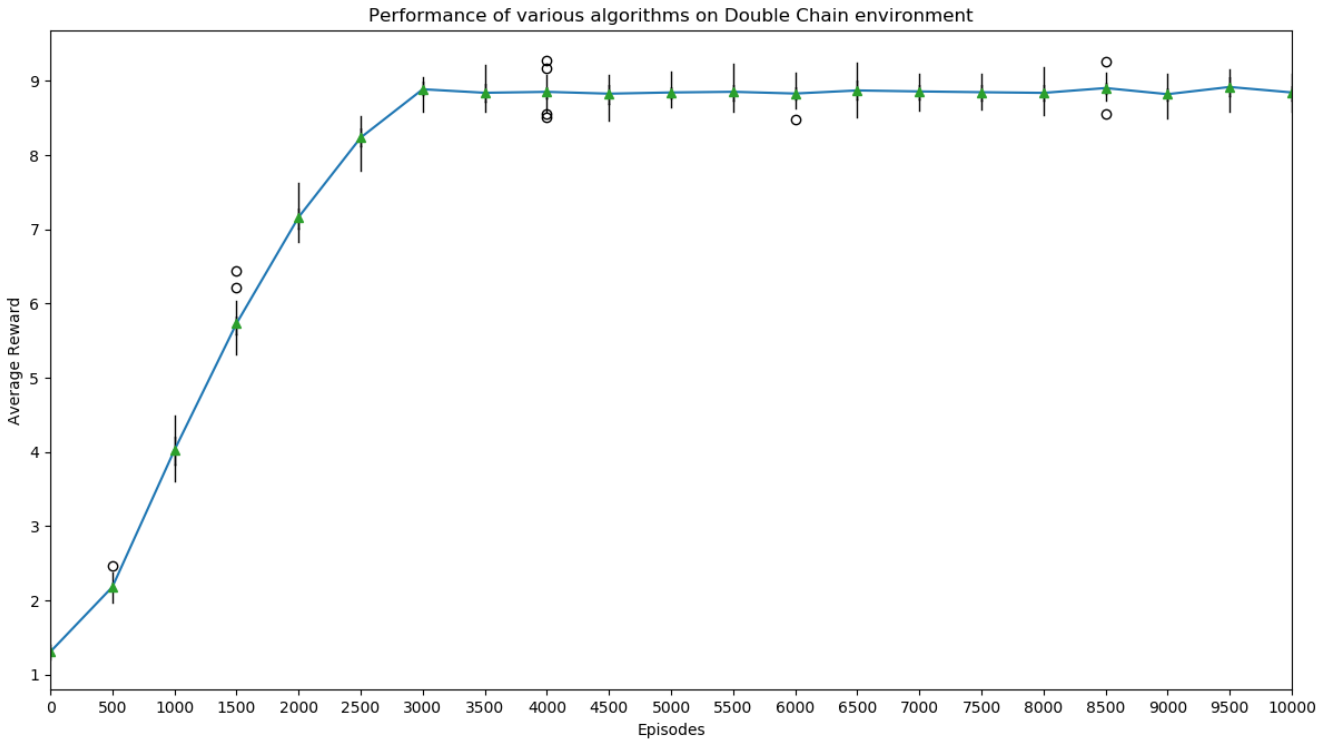


Figure 5: SARSA Convergence for double chain environment: Nearly 2000 times slower sub-optimal convergence!!

Extension

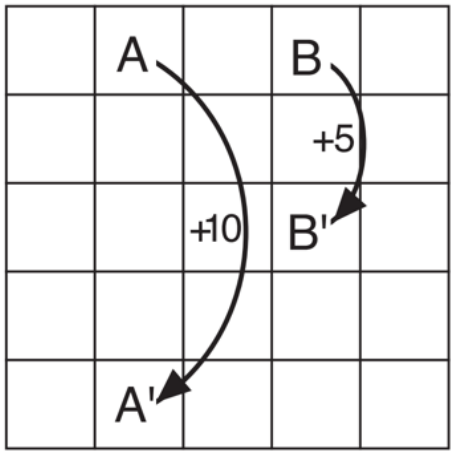
Usage of an action a is $u(a) = \sum_s p(s,a)$. An additional constraint on the usage of each action can be put as

$$\sum_s p(s,a) = c_a \ \forall \ a \in \mathcal{A}$$

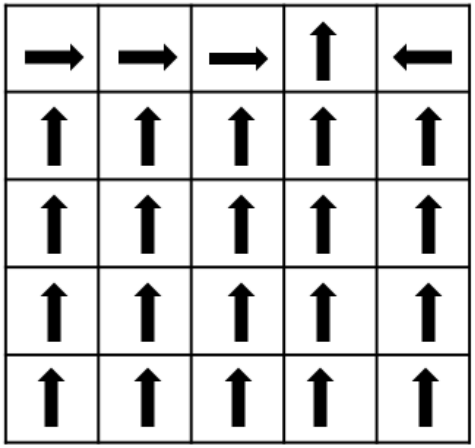
- Lagrangian gets updated $\bar{L} = L + \sum_{a \in \mathcal{A}} \zeta_a \Big(\sum_s p(s,a) - c_a \Big)$.
- The updated policy $p(s,a) = \frac{\alpha_{\mathbf{a}} q(s,a) e^{\frac{1}{\eta}\delta(\theta,s,a)}}{\sum_{s',a'} \alpha_{\mathbf{a}'} q(s',a') e^{\frac{1}{\eta}\delta(\theta,s',a')}}$ where $\alpha_a = e^{\zeta_a}$.
- Updates for $\alpha_a^{l+1} = \alpha_a^l \frac{c_a}{\sum_s p(s,a)}$

Extension

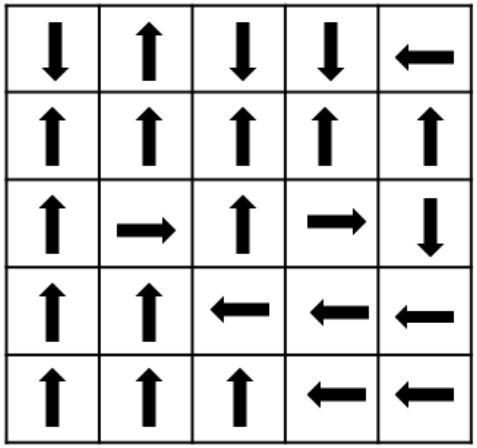
Constraining the actions in the final policy for the Gridworld environment.
Constraint Up:Down:Right:Left $= 10 : 3 : 5 : 7$



(a) Gridworld



(b) Unconstrained Policy
J = 1.79
21:0:3:1



(c) Constrained Policy
J = 0.07
10:3:5:7

Constraints are satisfied although the average reward reduces considerably.

Additional Observations

- REPS is significantly slower than other benchmark algorithms owing to the additional dual optimization problem.
- REPS requires features for every state. It required a lot of hit and trial to come up with reasonable features for states in all the environments.
- The $\epsilon-$bound on the policy is not always satisfied.

Thank You