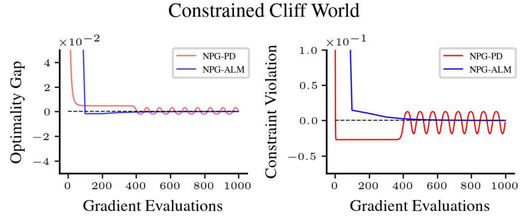
Augmented Lagrangian Method for Last-Iterate Convergence for Constrained MDPs

Michael Lu, Qiushi (Max) Lin, Mo Chen, Sharan Vaswani

Motivation

- Reinforcement learning (RL) has found success in many applications such as robotics or training LLMs.
- However, real-world systems must also operate under operational and safety constraints.
- For example, a robot needs to reach a goal while minimizing the amount of energy used.
- While primal-dual (PD) methods have strong theoretical results, they typically for the *mixture policy.* On deployment a policy selected at random from all previous policies.
- Practitioners, generally use the final policy (last-iterate) but it can be sensitive to hyper-parameters and may oscillate.



Main Goal

- Develop a general, practical and theoretically principled augmented Lagrangian (AL) framework for last-iterate convergence in constrained MDPs (CMDPs).

Background

- The objective is to find a stochastic policy that maximizes the cumulative reward while satisfying the constraints

$$\max_{\pi} V_r^{\pi}(\rho) \quad \text{s.t.} \quad V_{c_i}^{\pi}(\rho) \geq b_i \quad \forall i \in [m]$$

- The AL method is a classical approach to constrained optimization problems that adds a quadratic penalty to the Lagrangian: $\max_{\pi} \min_{\lambda} \mathcal{L}^{\beta}(\pi, \lambda)$

$$\mathcal{L}^{\beta}(\pi, \lambda) := V_r^{\pi}(\rho) + \frac{\beta}{2} \sum_{i=1}^{m} \left(- \min \left\{ V_{c_i}^{\pi}(\rho) - b_i - \frac{\lambda_i}{\beta}, 0 \right\}^2 + \frac{\lambda_i^2}{\beta^2} \right)$$

- At each iteration *t* the inexact AL method seeks to find an approximate solution to AL subproblem:

$$\mathcal{L}^{\beta}(\pi_{t+1}, \lambda_t) \geq \arg\max_{\pi} \mathcal{L}^{\beta}(\pi, \lambda_t) - \epsilon_t$$

- The dual variable is the updated as:

$$\lambda_{t+1}[i] = \lambda_t[i] - \frac{\beta}{2} \left(V_{c_i}^{\pi_{t+1}}(\rho) - [b_i + \xi_i(\pi_{t+1})] \right)$$

with slack variable $\xi_i(\pi) := \max \left\{ V_{c_i}^{\pi}(\rho) - b_i - \frac{\lambda_t[i]}{\beta}, 0 \right\}$

Main Results

Algorithm 1: Generic AL Policy Optimization Method for Problem 1

1: **Input**: π_1 (primal variable), $\lambda_1 = 0$ (dual variable), $\beta > 0$ (penalty parameter), $\epsilon_t = \mathcal{O}\left(\frac{1}{t^2}\right)$ (target sub-optimality for AL subproblem), T (number of iterations)
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Form $\mathcal{L}^{\beta}(\pi, \lambda_t) := V_r^{\pi}(\rho) + \frac{\beta}{2} \sum_{i=1}^{m} \left(- \min \left\{ V_{c_i}^{\pi}(\rho) - b_i - \frac{\lambda_i}{\beta}, 0 \right\}^2 + \frac{\lambda_i^2}{\beta^2} \right)$
4: $\pi_{t+1} = \texttt{Oracle-AL}(\mathcal{L}^{\beta}, \lambda_t, \epsilon_t)$
5: $\lambda_{t+1}[i] = \lambda_t[i] - \frac{\beta}{2} \left(V_{c_i}^{\pi_{t+1}}(\rho) - [b_i + \xi_i(\pi_{t+1})] \right)$, $\xi_i(\pi) := \max \left\{ V_{c_i}^{\pi}(\rho) - b_i - \frac{\lambda_t[i]}{\beta}, 0 \right\}$
6: **end for**
7: **Return**: π_{T+1}

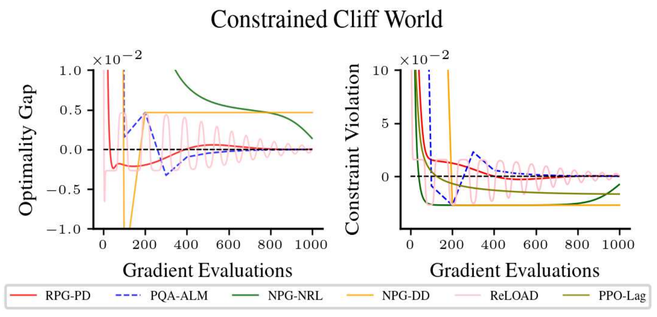
Theorem: For a sufficiently accurate oracle solver, and target $\epsilon \geq 0$ Algorithm 1 returns an ϵ-accurate policy after $T = O(1/\epsilon^2)$ iterations.

- To instantiate the framework, use the following policy gradient:

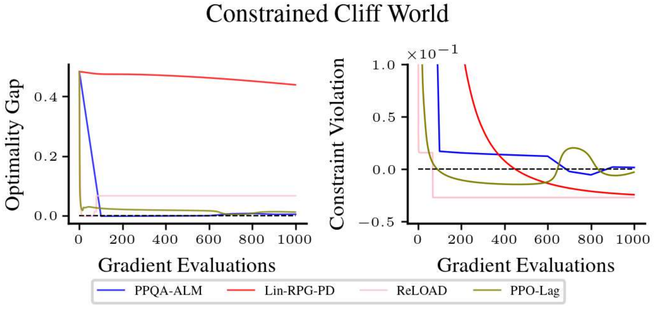
$$\nabla \mathcal{L}^{\beta}(\pi, \lambda) = \sum_s \frac{d^{\pi}(s)}{1-\gamma} \sum_a Q_{R(\pi)}^{\pi}(s, a)$$

$$R(\pi) := r - \beta \sum_{i=1}^{m} c_i \min \left\{ V_{c_i}^{\pi}(\rho) - b_i - \frac{\lambda_i}{\beta}, 0 \right\}$$

Corollary: For a specific policy gradient method, Algorithm 1 returns an ϵ-accurate policy after $T = O(1/\epsilon^6)$ iterations.

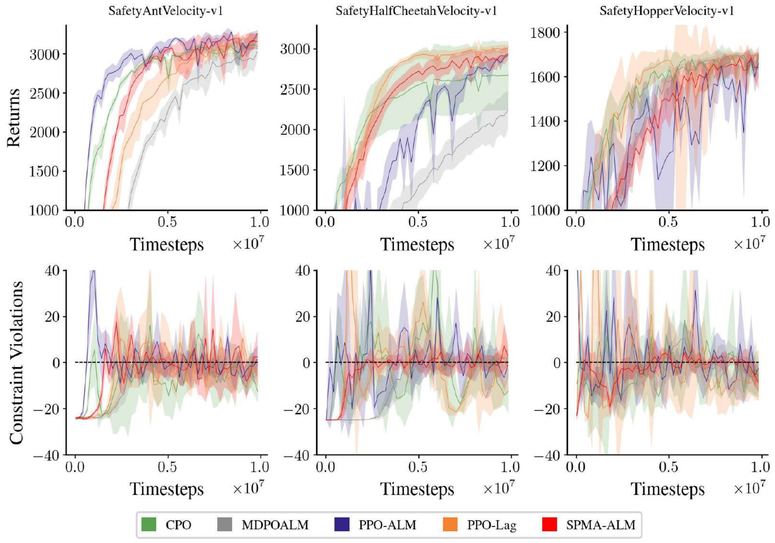


- Under suitable assumptions we can attain a similar result when using linear function approximation.



Large-Scale Experiments

- We evaluate our framework on three MuJCo tasks (each with a single constraint) from Safety-Gymnasium
- To instantiate the oracle solver, we use PPO and the more recent SPMA.
- For each experiment, we use 5 seeds and report the 95% confidence intervals



Algorithm	SafetyAntVelocity-v1 Return	SafetyAntVelocity-v1 CV	SafetyHalfCheetahVelocity-v1 Return	SafetyHalfCheetahVelocity-v1 CV	SafetyHopperVelocity-v1 Return	SafetyHopperVelocity-v1 CV
SPMA-ALM	**3202.29 ± 79.14**	-2.68 ± 5.69	2933.20 ± 69.90	-0.74 ± 3.34	**1687.37** ± 7.64	-6.06 ± 2.27
PPO-ALM	3042.76 ± 161.36	2.40 ± 5.45	2933.91 ± 21.05	-7.06 ± 1.87	1644.06 ± 45.45	-9.92 ± 2.85
PPO-Lag	3223.58 ± 52.21	-4.96 ± 2.50	**3010.19 ± 37.12**	-7.74 ± 3.30	1675.45 ± 42.51	-6.56 ± 9.12
CPO	3011.19 ± 123.74	-11.26 ± 5.13	2685.90 ± 419.20	6.18 ± 31.42	1682.68 ± 40.78	3.94 ± 23.10

- The AL-based methods achieve comparable performance to baseline approaches. However, unlike *PPOLag* or *CPO* our approach is backed by a clear theoretical foundation, yielding both a method that is both practical and principled.

Conclusion

- We develop a policy optimization framework with the AL method for CMDPs and provide last-iterate guarantees.
- We attain similar theoretical gurantees compared to prior work while still ensuring our algorithmic is deployable on large-scale environments.
- Future work includes incorporating actor-critic or off-policy updates directly into the theoretical framework