Supplementary Information for *Robust Lagrangian and Adversarial Policy Gradient for Robust Constrained Markov Decision Processes*

David M. Bossens

Appendix A: Proof of Robust Constrained Policy Gradient theorem

Using the notation $\mathbf{r}(s,a) = r(s,a) - \lambda c(s,a)$ to formulate the problem as an MDP, we have

$$\begin{aligned}
\nabla_\theta \mathbf{V}_\pi(s) &= \nabla_\theta \left(\sum_{a \in \mathcal{A}} \pi(a|s) \mathbf{Q}_\pi(s,a) \right) \qquad \text{(definition)} \\
&= \sum_{a \in \mathcal{A}} \mathbf{Q}_\pi(s,a) \nabla_\theta \pi(a|s) + \pi(a|s) \nabla_\theta \mathbf{Q}_\pi(s,a) \qquad \text{(product rule)} \\
&= \sum_{a \in \mathcal{A}} \mathbf{Q}_\pi(s,a) \nabla_\theta \pi(a|s) + \pi(a|s) \nabla_\theta \sum_{s',\mathbf{r}} \mathbb{P}(s', \mathbf{r}|a,s) \left(\mathbf{r}(s,a) + \mathbf{V}_\pi(s') \right) \qquad \text{(bootstrap from next Q)} \\
&= \sum_{a \in \mathcal{A}} \mathbf{Q}_\pi(s,a) \nabla_\theta \pi(a|s) + \pi(a|s) \nabla_\theta \sum_{s'} P(s'|a,s) \nabla_\theta \mathbf{V}_\pi(s') \\
&\qquad \text{(noting that } (P(s'|a,s) = \sum_{r} \mathbb{P}(s', \mathbf{r}|a,s)) \\
&= \sum_{a \in \mathcal{A}} \mathbf{Q}_\pi(s,a) \nabla_\theta \pi(a|s) + \pi(a|s) \sum_{s'} P(s'|s,a) \\
&\qquad \left(\sum_{a' \in \mathcal{A}} \nabla_\theta \pi(a'|s') \mathbf{Q}_\pi(s',a') + \pi(a'|s') \sum_{s''} P(s''|s',a') \nabla_\theta \mathbf{V}_\pi(s'') \right) \qquad \text{(unpacking analogously)} \\
&= \sum_{s_{\text{next}} \in \mathcal{S}} \sum_{k=0}^{\infty} \mathbb{P}(s \to s_{\text{next}}|k,\pi) \sum_{a} \mathbf{Q}_\pi(s_{\text{next}},a) \nabla_\theta \pi(a|s_{\text{next}}) \,. \qquad \text{(repeated unpacking)}
\end{aligned}$$

To demonstrate the objective is satisfied from $t = 0$ to $t = \infty$, the proof continues from the initial state s_0. There it is useful to consider the average number of visitations of s in an episode, $n(s) := \sum_{k=0}^{\infty} \mathbb{P}(s_0 \to s|k,\pi)$, and its relation to the on-policy distribution $\mu(s|\pi)$, the fraction of time spent in each state when taking actions from π:

$$\begin{aligned}
\nabla_\theta \mathbf{V}_\pi(s_0) &= \sum_{s \in \mathcal{S}} n(s) \sum_{a} \mathbf{Q}_\pi(s,a) \nabla_\theta \pi(a|s) \\
&\propto \sum_{s \in \mathcal{S}} \mu(s|\pi) \sum_{a} \mathbf{Q}_\pi(s,a) \nabla_\theta \pi(a|s) \\
&= \mathbb{E}_{\pi,P} \left[\sum_{a} \mathbf{Q}_\pi(s_t,a) \nabla_\theta \pi(a|s_t) \right] \\
&= \mathbb{E}_{\pi,P} \left[\sum_{a} \mathbf{Q}_\pi(s_t,a) \pi(a|s_t) \frac{\nabla_\theta \pi(a|s_t)}{\pi(a|s_t)} \right] \\
&= \mathbb{E}_{\pi,P} \left[\frac{\mathbf{Q}_\pi(s_t,a_t) \nabla_\theta \pi(a_t|s_t)}{\pi(a_t|s_t)} \right] \\
&= \mathbb{E}_{\pi,P} \left[\mathbf{Q}_\pi(s_t,a_t) \nabla_\theta \log\left(\pi(a_t|s_t) \right) \right] \,.
\end{aligned}$$

□

Appendix B: Proof of Robust Constrained Adversarial Policy Gradient Theorem

First note that the gradient of $\mathbf{V}_\pi(s_t)$ of a state s_t at time t is given by

$$\begin{aligned}
\nabla_{\theta_{\text{adv}}}\mathbf{V}_\pi(s_t) &= \nabla_{\theta_{\text{adv}}}\left(\sum_a \pi(a|s_t)\mathbf{Q}_\pi(s_t,a)\right) \qquad \text{(definition)}\\
&= \sum_a \pi(a|s_t)\nabla_{\theta_{\text{adv}}}\mathbf{Q}_\pi(s_t,a) \qquad (\pi \text{ independent of } \pi_{\text{adv}}\ )\\
&= \sum_a \pi(a|s_t)\nabla_{\theta_{\text{adv}}}\left(\mathbb{P}(s',\mathbf{r}|s_t,a)\left(\sum_{\mathbf{r},s'}\mathbf{r}(s_t,a)+\mathbf{V}_\pi(s')\right)\right) \qquad \text{(expand the Q-value)}\\
&= \sum_a \pi(a|s_t)\nabla_{\theta_{\text{adv}}}\left(\sum_{s'}\mathbb{P}(s'|s_t,a)\mathbf{V}_\pi(s')\right) \qquad (\text{reward distribution independent of } \pi_{\text{adv}})\\
&= \sum_a \pi(a|s_t)\nabla_{\theta_{\text{adv}}}\left(\sum_{s'}\pi_{\text{adv}}(s'|s_t,a)\mathbf{V}_\pi(s')\right) \qquad (\text{use } \pi_{\text{adv}} \text{ to generate the next state})\\
&= \sum_a \pi(a|s_t)\left(\sum_{s'}\mathbf{V}_\pi(s')\nabla_{\theta_{\text{adv}}}\pi_{\text{adv}}(s'|s_t,a)+\pi_{\text{adv}}(s'|s_t,a)\nabla_{\theta_{\text{adv}}}\mathbf{V}_\pi(s')\right) \qquad \text{(product rule)}\\
&= \sum_a \pi(a|s_t)\left(\sum_{s'}\mathbf{V}_\pi(s')\pi_{\text{adv}}(s'|s_t,a)\frac{\nabla_{\theta_{\text{adv}}}\pi_{\text{adv}}(s'|s_t,a)}{\pi_{\text{adv}}(s'|s_t,a)}+\nabla_{\theta_{\text{adv}}}\mathbf{V}_\pi(s')\right)\\
&\quad \text{(divide and multiply by } \pi_{\text{adv}})\\
&= \mathbb{E}_{\pi,\pi_{\text{adv}}}\left[\mathbf{V}_\pi(s_{t+1})\frac{\nabla_{\theta_{\text{adv}}}\pi_{\text{adv}}(s_{t+1}|s_t,a_t)}{\pi_{\text{adv}}(s_{t+1}|s_t,a_t)}+\nabla_{\theta_{\text{adv}}}\mathbf{V}_\pi(s_{t+1})\right] \qquad (\text{expectation over } \pi \text{ and } \pi_{\text{adv}}\ )\\
&= \mathbb{E}_{\pi,\pi_{\text{adv}}}\left[\mathbf{V}_\pi(s_{t+1})\nabla_{\theta_{\text{adv}}}\log\left(\pi_{\text{adv}}(s_{t+1}|s_t,a_t)\right)+\nabla_{\theta_{\text{adv}}}\mathbf{V}_\pi(s_{t+1})\right]\ . \qquad \text{(derivative of logarithm)}
\end{aligned}$$

Therefore, expanding this sum across all times $t = 0, \dots, T-1$, were T is the horizon of the decision process, the expression for $t = 0$ is given by

$$\nabla_{\theta_{\text{adv}}}\mathbf{V}_\pi(s_0) = \sum_{k=0}^{T-1}\mathbb{E}_{\pi,\pi_{\text{adv}}}\left[\mathbf{V}_\pi(s_{k+1})\nabla_{\theta_{\text{adv}}}\log\left(\pi_{\text{adv}}(s_{t+1}|s_t,a_t)\right)\right]\ .$$

□

Table I
PARAMETER SETTINGS OF THE EXPERIMENTS

Parameter	**Setting**
Discount	0.99
Entropy regularisation for π	5.0
Architecture for π and π_{adv}	100 hidden RELU units, softmax output
Learning rates for θ,λ,θ_{adv}, and λ_{adv}	0.001, 0.0001, 0.001, and 0.0001, multiplier $\frac{1}{1+n//500}$ for episode n
Initialisation of λ and λ_{adv}	both 50 for Inventory Management, both 1 for Safe Navigation 1 & 2
Critic	learning rate 0.001, 100 hidden RELU units, linear output, Adam optimisation of MSE, batch is episode
Uncertainty set	Hoeffding-based L1, 1 pseudocount, 90% confidence interval

APPENDIX C: EXPERIMENT DETAILS

Inventory Management

For each item, the purchasing cost is 2.49, the selling price is 3.99, and the holding cost is 0.03. The reward $r(s, a)$ is the expected revenue minus the ordering costs and the holding costs. The demand distribution is Gaussian with mean μ and standard deviation σ. Each episode consists of $T = 100$ steps and the discount is set to $\gamma = 0.99$. The constraint-cost is $c(s, a) = \max(0, a - L(s)$, where the purchasing limit is set to $L(s) = \mu + \sigma$ for $s \leq 2$ and $L(s) = \mu$ for $s > 2$. The constraint-cost budget is set to $d = 6.0 \approx \sum_{t=0}^{T-1} \gamma^t 0.1$ which allows the action to exceed the purchasing limit on average roughly one item every 10 time steps. The constraint-cost function is not adjusted for tests; that is, the original μ and σ are used in its computation rather than the perturbed parameters.

Safe Navigation

The objective is to move from start, $s_0 = (0, 0)$ to goal, $(4, 4)$, as quickly as possible while avoiding areas that incur constraint-costs. The agent observes its (x, y)-coordinate and outputs an action going one step left, one step right, one step up, or one step down. The episode is terminated if either the agent arrives at the goal square or if more than T time steps have passed. Instead of using the full state space as next states in the uncertainty sets, the probability vectors $\mathcal{P}_{s,a}$ consider for the next state only the 5 states in the Von Neumann neighbourhood $\mathcal{N}(s)$ with Manhattan distance of at most 1 from s; this requires setting $\alpha(s, a) = \sqrt{\frac{2}{n(s,a)} \ln\left(\frac{2^{S'} SA}{\delta}\right)}$, replacing S by $S' = 5$ in the set of outcomes.

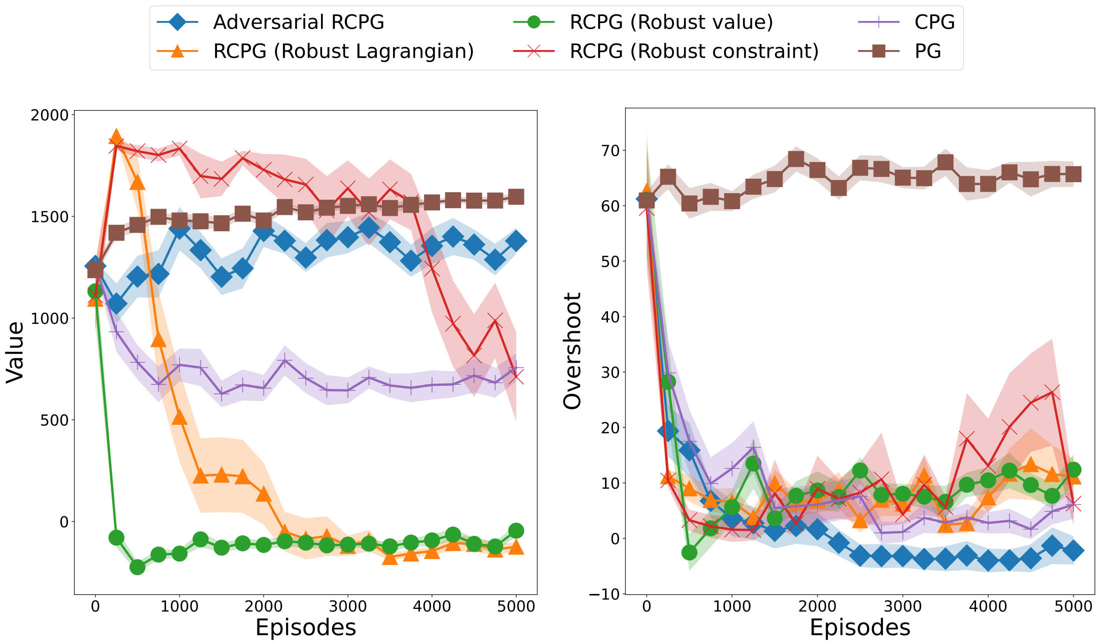
APPENDIX D: TRAINING HYPERPARAMETERS

A. Hyperparameters

Hyperparameters are set according to Table I. The discount is common at 0.99 and the architecture was chosen such that it is large enough for both domains. The entropy regularisation is higher than usual training procedures because of the Lagrangian yielding larger numbers in the objective. Learning rates were tuned in $\{0.10, 0.01, 0.001\}$ for policy parameters (θ and θ_{adv}) and in $\{0.01, 0.001, 0.0001\}$ for Lagrangian multipliers (λ and λ_{adv}); the setting shown in the table is the best setting for Inventory Management and Safe Navigation domains and this loosely corresponds to the two time scale stochastic approximation criteria [1]. The critic was fixed to 0.001 for both domains as this is a reliable setting for the Adam optimiser. For Inventory Management, it is possible to satisfy the constraint from the initial stages of learning so the initial Lagrangian multiplier λ is set to 50. For Safe Navigation domains, the initial λ is set to 1 since it is not immediately possible to satisfy the constraints without learning viable paths to goal. To encourage stochasticity in case of limited samples, each state-action pair (s, a) is initialised with a pseudo-count $n(s, a) \leftarrow 1$, representing the uniform distribution as a weak prior belief. The error probability is $\delta = 0.10$ for a 90% confidence interval.

Appendix E: Training and model estimation

In Inventory Management, the model estimation phase is based on 100 episodes with $\mu = S/4$ and $\sigma = S/6$, yielding uncertainty sets with budget α ranging in $[0.3, 0.9]$ across the state-action space. The widely varying values and overshoots during training (see Figure 1) reflect in part a different training environment. Fig. 1 shows the performance in the policy training phase.



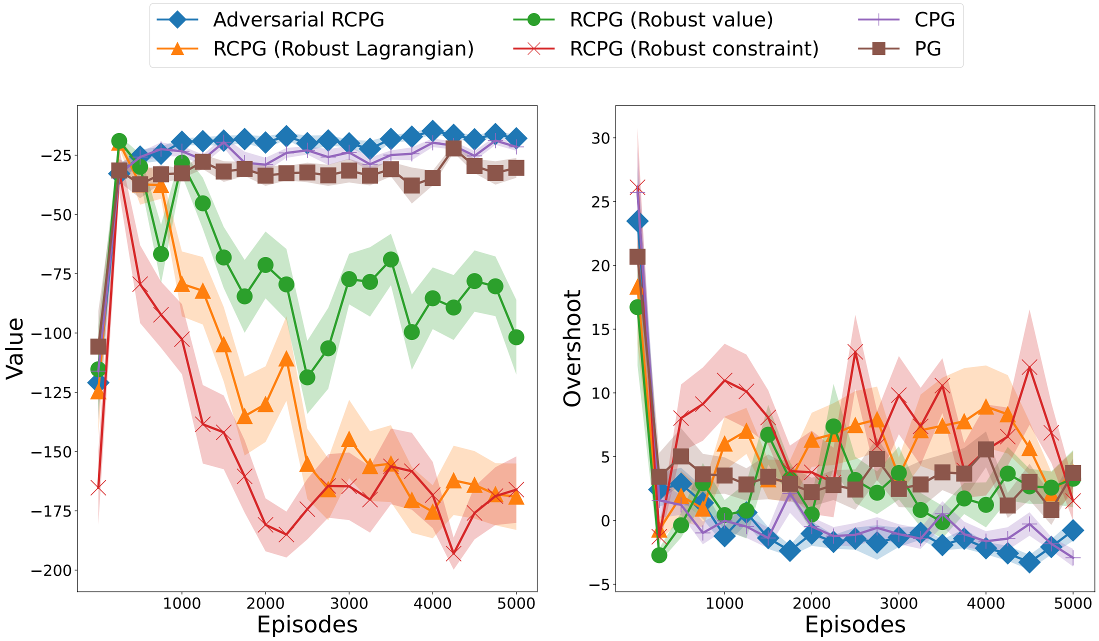
(a) Training value

(b) Training overshoot

Figure 1. Training performance metrics of the algorithms over 5,000 episodes on Inventory Management. Note that the training performance corresponds to the performance on the simulated transition dynamics, which is defined differently for the different algorithms.

In Safe Navigation 1, the model estimation phase is based on 100 episodes with $P_{\text{success}} = 0.80$, which results in the uncertainty budget α ranging in $[0.25, 0.7]$ across the state-action space. Fig. 2 shows the performance in the policy training phase.

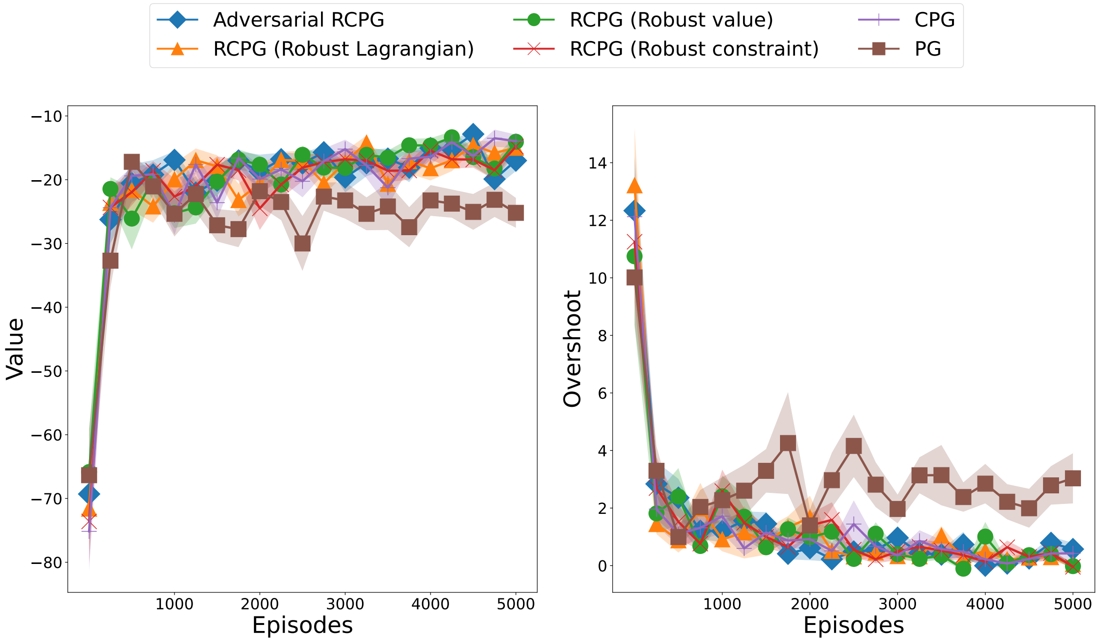
In Safe Navigation 2, the model estimation phase is based on 10,000 episodes with $P_{\text{success}} = 1.0$. The resulting uncertainty set has a smaller uncertainty budget, with α ranging in $[0.03, 0.085]$ across the state-action space. Fig. 3 shows the performance in the policy training phase.



(a) Training value

(b) Training overshoot

Figure 2. Training performance metrics of the algorithms over 5,000 episodes on Safe Navigation 1. Note that the training performance corresponds to the performance on the simulated transition dynamics, which is defined differently for the different algorithms.



(a) Training value

(b) Training overshoot

Figure 3. Training performance metrics of the algorithms over 5,000 episodes on Safe Navigation 2. Note that the training performance corresponds to the performance on the simulated transition dynamics, which is defined differently for the different algorithms.

References

[1] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Springer, second ed., 2022.