Neural-Progressive Hedging: Enforcing Constraints in Reinforcement Learning with Stochastic Programming (Supplementary Materials)

Supriyo Ghosh[1] **Laura Wynter**[2] **Shiau Hong Lim**[2] **Duc Thien Nguyen**[3]

[1]Microsoft Research, Bangalore, India
[2]IBM Research AI, Singapore
[3]Singapore Management University, Singapore

ADDITIONAL NUMERICAL RESULTS

Figure 1 compares the warm-start (called NP-WS) version with the damped-guidance, or imitation-learning-type expert guidance (called NP). Both versions perform far better than the RL policy.

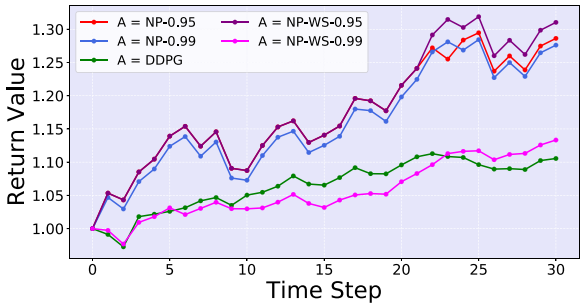


Figure 1: Performance comparison of two versions of the algorithm: warm start ($\kappa^1 = 1, \hat{\imath} = 1$) vs. imitation learning ($\kappa^i = (1+i)^{-2}, \hat{\imath} = 20$, in this example).

Figure 2 illustrates the episodic constraint violation cost for two benchmark constrained RL algorithms, CPO of Achiam et al. [2017], and PPO-Lagrangian of Ray et al. [2019]. Each episode duration is 30 time steps and in each time step t, we enforce a cost of 1 if the amount available in the liquid instrument is less than the cumulative account payable up through time t. Observe that CPO fails to learn the constraints during training. The PPO-Lagrangian method is able to bring down the episodic cost to 0 during training (the limit of the episodic cost is set to 0), but as shown in the main paper (see Figure 2(c)), the learned PPO-L policy is not able to satisfy the constraints during execution.

IMPLEMENTATION DETAILS

Discretization of the stochastic program scenario tree Consider a finite scenario tree formulation of a stochastic programming problem, such that the set of nodes in the scenario tree at time stage t are denoted N_t. A node denotes a point in time when a realisation of the random process becomes known and a decision is taken. Each node replicates the data of the optimization problem, conditioned on the probability of visiting that node from its parent node. A path from the root to each leaf node is referred to as a scenario; its probability of occurrence, p_s, is the product of the conditional probabilities of visiting each of the nodes on that scenario path. The discretized model-based stochastic program is thus:

$$\max \sum_{s=1\ldots S} F_s(x,\xi) := \sum_{s=1\ldots S} p_s \sum_{t=1\ldots T} f_t(x_s(t)). \quad (1)$$

The *non-anticipativity* constraints are critical for the implementability of the policy but they couple the scenario sub-problems by requiring that the action x_t at time t is the same across scenarios (i.e., sample paths) sharing the sample path up to and including time t. For each $\xi \in \Xi$, these coupling constraints are expressed as:

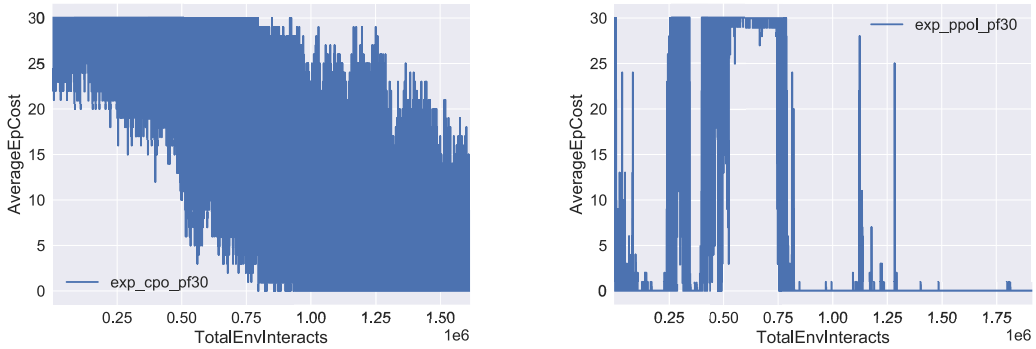
$$x(\xi) = (x_1, x_2(\xi_1), x_3(\xi_1,\xi_2), \ldots x_T(\xi_1 \ldots \xi_{T-1}). \quad (2)$$

Using the discretized formulation of (1), and following Rosa and Ruszczyński [1996] we can rewrite (2) in a manner that facilitates relaxation of those constraints: Define the last common stage of two scenarios s_1 and s_2 as

$$t^{\max}(s_1,s_2) := \max\{\hat{t} : s_1(t) = s_2(t), t = 1, \ldots \hat{t}\}, \quad (3)$$

and then re-order the scenarios $s = 1 \ldots S$, so that at every s, the scenario $s+1$ has the largest common stage with scenario i for all scenarios $s' > s$, that is $t^{\max}(s, s+1) := \max\{t^{\max}(u,v) : v > u\}$. Then, define the sibling of scenario s at time stage t as a permutation $\nu(s,t) := s+1$ if $t_{\max}(s, s+1) \geq t$ and $\nu(s,t) := \min\{t' : t^{\max}(s,t') \geq t\}$ otherwise. The inverse permutation shall be denoted $\nu^{-1}(s,t)$. Note that the sibling of a scenario depends upon the time stage, and that a scenario with no shared decisions at a time stage has by definition itself as sibling. Using the above, Rosa and Ruszczyński [1996] re-define the

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).



(a) (b)

Figure 2: Episodic (episode length = 30) constraint violation cost during training for (a) CPO; and (b) PPO-Lagrangian.

constraints enforcing measurability in terms of the sibling function as follows:

$$x_s(t) = x_{\nu(s,t)}(t) \;\; \forall (s,t), \;\; s \neq \nu(s,t). \tag{4}$$

Equation (4) is convenient in the primal-dual formulation in terms of discrete scenarios, presented next. We are interested in maintaining the separability of the subproblems which depend only on individual scenarios of the random variable to facilitate handling large problems via scenario-based decomposition. To do so, we relax the constraints using the following formulation

$$\mathcal{M} := \{x : M_1 x_1(\xi) + \ldots + M_S x_S(\xi) = 0\}, \tag{5}$$

where the matrices in (5) are defined so that each M_s is a matrix of -1, 0 and 1 such that at the root node $x_{11} = x_{12}, x_{12} = x_{23} = \cdots x_{1,s-1} = x_{1,s}$, at the stage $t = 2$, there are as many such sets of equalities as children nodes emanating from the root node, and so on up to stage $T-1$. At stage T, all nodes are leaves and no such linking constraints are required. The projection of a point x^i onto the subspace $\mathcal{M}$, $P_{\mathcal{M}}[x^i(\cdot)]$ can be computed by taking the conditional expectation of x^i, $E_{\xi \mid \xi_1,\ldots\xi_{i-1}}$. Lagrange relaxation of the measurability constraints (4) gives rise to the following Lagrange function, in terms of the discrete scenarios $s = 1 \ldots S$:

$$\mathcal{L}(x,\lambda) = \sum_{s=1\ldots S} p_s \sum_{t=1\ldots T} f_t(x_s(t)) + \sum_{s=1\ldots S} \sum_{t=1\ldots T-1} \lambda_s(t)(x_s(t) - x_{\nu(s,t)}(t)). \tag{6}$$

The scenario subproblems are re-defined as a function of the inverse permutation of the sibling function:

$$\min_{x_s \in G'_s} \mathcal{L}_s(x_s,\lambda_s) = p_s \sum_{t=1\ldots T} f_t(x_s(t)) + \sum_{t=1\ldots T-1} (\lambda_s(t) - \lambda_{\nu^{-1}(s,t)}(t)) x_s(t) \tag{7}$$

for each $s = 1 \ldots S$. The dual problem is given by

$$\max_{\lambda} D(\lambda) := \min_{x \in G'} \mathcal{L}(x,\lambda). \tag{8}$$

It is possible to further speed up convergence of our NP algorithm in practice using the approach of Zehtabian and Bastin [2016]. This approach monitors the primal and dual gap terms in convergence criteria separately to update the penalty parameters so as to reduce the convergence gap quickly.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 22–31. JMLR. org, 2017.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *Technical report, Open AI*, 2019.

Charles H Rosa and Andrzej Ruszczyński. On augmented lagrangian decomposition methods for multistage stochastic programs. *Annals of Operations Research*, 64(1):289–309, 1996.

Shohre Zehtabian and Fabian Bastin. *Penalty parameter update strategies in progressive hedging algorithm*. CIRRELT, 2016.