535514 RL Team Project Proposal: Efficient Action-Constrained Reinforcement Learning via Acceptance-Rejection Method and Augmented MDPs

Wei Hung
National Yang Ming Chiao Tung University
{hwei1048576.cs08}@nycu.edu.tw

1 Project Overview

Track: 1. Algorithm

Abstract and project goal: Action-constrained reinforcement learning (ACRL) is a generic framework for learning control policies with zero action constraint violation, which is required by various safety-critical and resource-constrained applications. The existing ACRL methods can typically achieve favorable constraint satisfaction but at the cost of either a high computational burden incurred by the quadratic programs (QP) or increased architectural complexity due to the use of sophisticated generative models.

In this project, our goal is to propose a generic and computationally efficient framework that can adapt a standard unconstrained RL method to ACRL. We expect to explore the following two techniques: (i) To enforce the action constraints, we leverage the classic acceptance-rejection method, where we treat the unconstrained policy as the proposal distribution and derive a modified policy with feasible actions. (ii) To improve the acceptance rate of the proposal distribution, we propose to construct an augmented two-objective Markov decision process (MDP), which includes a penalty signal for the rejected actions and incentivizes the learned policy to stay close to the feasible action sets. We plan to evaluate our method in both robot control and resource allocation domains against state-of-the-art ACRL methods, in terms of training efficiency, constraint violation, and the action inference time.

TL;DR ("Too Long; Didn't Read"): An efficient ACRL algorithm that achieves zero action constraint violation without the need for costly quadratic programs or the high architectural complexity of sophisticated generative models.

1.1 Motivation

- **Why is the problem interesting?** ACRL is an important framework for various safety-critical and resource-constrained applications, such as robot control subject to the inherent kinematic constraints of the robots and dynamic resource allocation for networked systems subject to resource constraints. Despite this, it remains unknown how to develop a computationally efficient method that can achieve zero constraint violation.
- **Critical challenges.** Challenge 1: One major challenge is to enforce the action constraints without using a projection step. This significantly limits the set of possible solutions. Challenge 2: If we choose to utilize the acceptance-rejection method, the low acceptance rate is very likely to occur, especially in the early training stage.
- **Justify why the problem remains open or unsolved.** Existing ACRL methods have explored the following techniques, each of which relies either on costly quadratic programs or sophisticated generative models.

  - *Action projection*: As a conceptually simple and widely-used technique, action projection finds a feasible action closest to the original unconstrained action produced by the policy. The projection step can be used in action post-processing [Kasaura et al., 2023] or implemented by a differentiable projection layer as part of the policy network of a standard deep RL algorithm for end-to-end training [Pham et al., 2018, Dalal et al., 2018]. Despite the simplicity, to find close feasible actions, action projection needs to solve a quadratic program (QP), which is computationally costly and scales poorly to high-dimensional action spaces.
  - *Frank-Wolfe search*: Lin et al. [2021] propose to decouple policy updates from action constraints by a Frank-Wolfe search subroutine. Despite its effectiveness, Frank-Wolfe method requires solving multiple QPs per training iteration and therefore suffers from substantially higher training time.
  - *Generative models*: To replace the projection step, generative models, such as Normalizing Flows, have been employed as a learnable projection layer that is trained to satisfy the constraints [Brahmanage et al., 2023].
- **State-of-the-art methods.** SPre+ [Kasaura et al., 2023] and FlowPG [Brahmanage et al., 2023] are known to be recent competitive baselines and shall be considered state-of-the-art methods in ACRL.

2 Problem Formulation

In ACRL, we consider an action-constrained Markov Decision Process (MDP). Given a set $\mathcal{X}$, let $\Delta(\mathcal{X})$ denote the set of all probability distributions on $\mathcal{X}$. An action-constrained MDP is defined by a tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, r, \mathcal{C})$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ serves as the transition kernel, $\gamma \in (0, 1)$ is the discount factor, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the bounded reward function. Without loss of generality, we presume the reward $r(s, a)$ to lie in the $[0, 1]$ interval since we can rescale a bounded reward function to the range of $[0, 1]$ given the maximum and minimum possible reward values. For each $s \in \mathcal{S}$, there is a non-empty feasible action set $\mathcal{C}(s) \subseteq \mathcal{A}$ induced by the underlying collection of action constraints. As a result, no actions outside the feasible set $\mathcal{C}(s)$ can be applied to the environment, ensuring that only valid actions are considered within the system dynamics. Notably, we make no assumption on the structure of $\mathcal{C}(s)$ (and hence $\mathcal{C}(s)$ needs not be convex).

Our goal is to learn an optimal policy π^* in the sense that $Q(s, a; \pi^*) \geq Q(s, a; \pi)$, for all $s \in \mathcal{S}, a \in \mathcal{C}(s)$ and $\pi \in \Pi_{\mathcal{C}}$, where $Q(s, a; \pi) := \mathbb{E}\big[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a; \pi\big]$ and $\Pi_{\mathcal{C}} := \{\pi : \mathcal{S} \to \Delta(\mathcal{C})\}$, which denotes the set of all feasible policies. To learn a policy under large state and action spaces, we use the parameterized functions $\pi_\phi : \mathcal{S} \to \Delta(\mathcal{A})$ as a function approximator, which is typically a neural network in the deep RL literature.

3 Empirical Evaluation

3.1 Performance Metrics

We plan to evaluate the algorithms based on the following three metrics:

- *Training efficiency*: We record the evaluation returns at different training stages, in terms of both the wall clock time and the environment steps. To ensure fair measurements of wall clock time, we run each algorithm independently using the same computing device. Moreover, we report the cumulative number of QP operations as an indicator of the training computational overhead.
- *Valid action rate*: At the testing phase, we evaluate the valid action rate by sampling 100 actions from the policy network at each step of an episode. This metric reflects how effectively each method enforces the action constraints throughout the evaluation phase.

3.2 Baseline Methods

We will take into account the following recent benchmark ACRL algorithms:

- NFWPO [Lin et al., 2021]: NFWPO achieves favorable constraint satisfaction at the cost of high QP overhead as it enforces action constraints by Frank-Wolfe search.
- SPre+ [Kasaura et al., 2023]: SPre+ adapts the vanilla SAC to ACRL by using a QP-based projection step for action post-processing, learning the critic with pre-projected actions, and applying a penalty term to guide the policy updates.
- FlowPG [Brahmanage et al., 2023]: FlowPG enforces action constraints via a pre-trained Normalizing Flow model, and we use the official source code by the original paper.

Both SPre+ and FlowPG are strong baselines and are considered state-of-the-art methods in ACRL.

3.3 Benchmark Tasks or Datasets

We evaluate the algorithms in various benchmark domains widely used in the ACRL literature.

- *MuJoCo locomotion tasks* [Todorov et al., 2012]: These tasks involve training robots to achieve specified goals, such as running forward and walking at a speed within certain limits.
- *Resource allocation for networked systems*: These tasks involve properly allocating resource under capacity constraints, including NSFnet and Bike Sharing System (BSS) [Ghosh and Varakantham, 2017]. Regarding NSFnet, the learner needs to allocate packets of different flows to multiple communication links. The action constraints are induced by the per-link maximum total assigned packet arrival rate. We follow the configuration provided by [Lin et al., 2021] and use the open-source network simulator from PCC-RL [Jay et al., 2019]. Regarding BSS, the environment consists of m bikes and n stations, each with a capacity limit of c. The learner needs to reallocate bikes to different stations based on the current situation.

4 Methodology (Optional)

We would like to propose a new ACRL framework composed of the following three components: (1) Acceptance-rejection method: Use an oracle to verify whether the sampled action is in the feasible action set. (2) Augmented two-objective MDP: Assign penalties to invalid actions within an augmented MDP framework, thereby reducing the rate of action violations. (3) Multi-objective RL (MORL): Use an existing MORL method to discover well-performing policies under all penalty weights simultaneously.

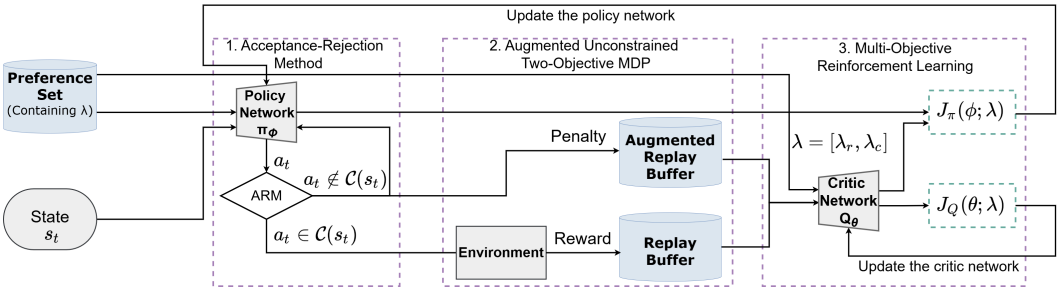

Figure 1: An illustration of the proposed framework.

References

Kazumi Kasaura, Shuwa Miura, Tadashi Kozuno, Ryo Yonetani, Kenta Hoshino, and Yohei Hosoe. Benchmarking actor-critic deep reinforcement learning algorithms for robotics control with action constraints. *Robotics and Automation Letters*, 2023.

Tu-Hoa Pham, Giovanni De Magistris, and Ryuki Tachibana. Optlayer-practical constrained optimization for deep reinforcement learning in the real world. In *International Conference on Robotics and Automation*, 2018.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Jyun-Li Lin, Wei Hung, Shang-Hsuan Yang, Ping-Chun Hsieh, and Xi Liu. Escaping from zero gradient: Revisiting action-constrained reinforcement learning via Frank-Wolfe policy optimization. In *Uncertainty in Artificial Intelligence*, 2021.

Janaka Brahmanage, Jiajing Ling, and Akshat Kumar. FlowPG: Action-constrained Policy Gradient with Normalizing Flows. *Advances in Neural Information Processing Systems*, 2023.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *International Conference On Intelligent Robots and Systems*, 2012.

Supriyo Ghosh and Pradeep Varakantham. Incentivizing the use of bike trailers for dynamic repositioning in bike sharing systems. In *International Conference on Automated Planning and Scheduling*, 2017.

Nathan Jay, Noga H Rotman, P Godfrey, Michael Schapira, and Aviv Tamar. A deep reinforcement learning perspective on internet congestion control. *International Conference on Machine Learning*, 2019.