Fast-exploring reinforcement learning with applications to stochastic networks

Daniel Mastropietro, Urtzi Ayesta, Matthieu Jonckheere, Szymon Majewski

Fast-exploring reinforcement learning with applications to stochastic networks*

Daniel Mastropietro[1,2,*], Urtzi Ayesta[2,3,4], Matthieu Jonckheere[2], and Szymon Majewski[5]

[1]Université de Toulouse INP, 31071 Toulouse, France
[2]CNRS-IRIT, 31071 Toulouse, France
[3]Ikerbasque - Basque Foundation for Science, 48009 Bilbao, Spain
[4]UPV/EHU, University of the Basque Country, 20018 Donostia, Spain
[5]Ecole Polytechnique, Paris, France

September 9, 2025

Abstract

We introduce FVRL (Fleming-Viot reinforcement learning), a reinforcement learning algorithm for optimisation problems where a long-term objective is largely influenced by states that are very rarely observed under all policies. In this context, usual discovery techniques including importance sampling are inapplicable because no alternative policy exists that increases the observed frequency of the rare states. We instead propose a novel approach that uses Fleming-Viot particle systems, a family of stochastic processes evolving simultaneously under the same law, that exploits prior knowledge of the environment to boost exploration of the rare states. A renewal theory argument allows us to consistently estimate the stationary probability of the rare states from excursions that have considerably lower sample complexity than usual Monte-Carlo explorations. We demonstrate how to combine this estimator with policy gradient learning to construct the FVRL algorithm, which is suited to efficiently solve problems where the optimisation function is expressed as a long-run expectation, such as the long-run expected reward. We show that the FVRL algorithm converges to a local optimizer of the parameterised objective function, and illustrate the method on two optimisation problems that aim at minimizing the long-run expected cost under admission control policies of threshold type: a simple $M/M/1$ queue system and a two-job-class loss network. Our experimental results show that, under the same sample complexity, FVRL outperforms a vanilla Monte-Carlo reinforcement learning method by converging to the optimum thresholds considerably faster.

*Corresponding author: daniel.mastropietro@gmail.com
Research partially supported by the French National Research Agency (ANR) under the programs "Investments for the Future" with reference code CIMI ANR-11-LABX-0040 and ANR-22-CE25-0013-02 (ANR EPLER), and by project STIC-AMSUD LAGOON.

1 Introduction

Reinforcement learning methods (RL) [25] are very versatile in solving stochastic optimisation problems thanks to their little model assumptions. In the model-free paradigm, they only assume that the stochastic system can be modeled as a Markov decision process (MDP), but no knowledge of the transition probabilities or rates are required. From interactions between an agent and the system (environment), RL methods are able to learn near optimal policies by trial and error. The agent tries different possible actions on many different environment states and uses the signals received in the form of rewards or costs to incrementally adjust its decisions towards a better policy. Although this approach is appealing by the potential to tackle very complex optimisation problems whose modeling can be challenging, it makes RL algorithms a computational intensive task. Depending on the environment size and characteristics, the learning process can be prohibitively slow due to lack of sufficient coverage of all possible states and actions, some of which can be highly informative. This makes the design of efficient exploration schemes one of the key components of RL algorithms, particularly in environments where the reward landscape is sparse, which usually leads to situations where informative rewards are rarely observed.

Environments with rarely observed states (under all policies) are the object of study of the present work, in particular in the context of problems where the optimisation function is largely influenced by visiting those rare states. More specifically, we look into problems where the optimisation function is based on a long-run expectation criterion, such as the long-run expected reward. For example, a network system will have its long-run quality of service significantly degraded if the cost of interruption is large even when such interruptions are rare.

We propose to tackle this problem by leveraging prior or learned knowledge about the MDP in order to *force* exploration away from frequently visited states that are uninformative for the optimisation task at hand, while keeping the current policy in place. We integrate this prior knowledge into the parameterisation of a population evolution model known as Fleming-Viot in order to efficiently estimate the stationary probability distribution of the rare high-reward states, which is a fundamental piece of the long-run expected criterion being optimised.

To fix ideas, consider the MDP defined by a simple $M/M/1$ queue system and an admission control policy that is solely a function of the queue size, x. When the system load ρ is less than 1, we know from standard queueing theory that most of the excursion of the Markov chain defined by the queue size, takes place at lower values of x[1]. If the policy rejects an incoming job when the queue is at a predefined length K at which point a negative reward is accrued, observing a non-zero reward may be very rare. For instance, by PASTA (Poisson arrivals see time averages), the rejection probability is of the order of 10^{-7} when $K = 40$ and $\rho = 0.7$. Given a rejection cost function, reinforcement learning could be used to obtain the optimal K threshold that minimizes the long-run expected cost estimated from the job rejection events observed during a Monte-Carlo simulation of the MDP. This, however, may be a prohibitively time consuming task because Monte-Carlo would require a very long simulation when the rejection probability is very small. To tackle this problem, we instead propose a Fleming-Viot simulation (explained in the ***Contributions*** paragraph below) that forces the Markov chain excursions away from the high probability and non-informative lower occupation states, to obtain an estimation of the long-run expected cost objective with considerably lower sample complexity.

Related work The problem of sparse and rare rewards has been of central interest for a long time, as it is directly linked to the core topic of efficient exploration of RL environments. We cite a few approaches that have been proposed to tackle this problem: importance sampling, reward shaping, reward-free exploration, and curiosity-driven methods.

Importance sampling is an offline method that proposes exploring the environment using an alternative exploration policy specially designed to increase the occurrence probability of the rare events under the original policy. An appropriate adjustment factor is then applied to obtain a correct estimate of expected values that depend on the observation frequency of these events [27]. An important constraint of the alternative exploration policy is that it must have the same state coverage as the original policy, i.e. all states reachable by the original policy must also

[1]The stationary probability at x is $\rho^x(1-\rho)$, an exponentially decreasing function of x when $\rho < 1$.

be reachable by the alternative policy. However, this is not always possible. For instance, in the simple $M/M/1$ queue example mentioned above, there is no admission control policy that is able to increase the visit frequency of the rejection event while maintaining the state coverage of the original policy: if the rejection threshold K is large and the alternative proposed policy decreases its value in order to increase the frequency of the rejection event, all the states between the original and the new smaller threshold will never be visited.

Reward shaping was initially proposed by Mataric back in 1994 [18], as a method where an expert proposes or shapes the reward landscape of an environment in order to obtain policies that are able to visit certain states of interest. We believe this type of approach has two main drawbacks: domain knowledge is needed to shape the rewards, and only a few (due to handcraft shaping) and uncontrolled policies may be reached as a result of shaping.

Reward-free exploration methods propose exploration mechanisms that are based exclusively on the probability of state visit, regardless of the reward landscape –and thus of policies–, as their goal is to collect enough data that can later be used to optimise policies under *any* reward context [16]. These methods propose reducing the sample complexity by identifying a set of states worth visiting based on a minimum visit frequency criterion, assuming the states with very small visit probability have negligible impact on reward optimisation. Although, like ours, these methods aim at reducing sample complexity, they ignore the states at the tail of the visit frequency distribution, which nonetheless could bring relevant information for the optimisation process if their rewards happen to be sufficiently large. The method does not provide a sample complexity reduction mechanism for these states, which is the focus of our work.

Curiosity-driven methods encourage the exploration of less visited states by including a bonus term in the objective function [20, 6]. Our method could be considered a type of curiosity-driven method, with the important difference that, as we will see below, the exploration of the rare states is not just encouraged, but *forced*, by prohibiting excursions from visiting a subset of more frequently visited and uninformative states.

Reinforcement learning for stochastic networks: Several works have considered RL in the context of stochastic network models. Dai and Gluzman [15] propose a deep reinforcement learning algorithm for queueing network control. This study adapts the proximal policy optimisation (PPO) algorithm to Markov Decision Processes (MDPs) with infinite state spaces, unbounded costs, and long-run average cost objectives, which are common in queueing networks. Our primary goal here is actually to address a challenge explicitly mentioned by Dai and Gluzman as future work: developing more sample-efficient simulation methods by incorporating learned knowledge about the environment. Additionally, we test our method on multi-class stochastic networks, a benchmark also suggested by Dai and Gluzman for evaluating RL methods.

On the other hand, the work in [13] establishes convergence of Natural Policy Gradient algorithm, when initialized with the MaxWeight policy, in infinite-state queueing systems under the average-reward criterion. While deep reinforcement learning algorithms often fail to produce stable policies in unbounded state spaces, [21] introduces a Lyapunov-based cost-shaping technique and state transformations that improve stability and performance in queueing networks. We also refer to [4] which proposes a machine learning approach to the optimal control of multi-class fluid queueing networks. In contrast to the aforementioned papers, in our case, the optimal policy heavily depends on a state that is rarely visited, a problem that to the best of our knowledge has not yet been addressed in the queueing and stochastic network literature.

Contributions Our main contribution is the proposal of an exploration strategy based on a population evolution model known as Fleming-Viot particle system (FV), designed with the goal of increasing the discovery rate of rare states compared to the natural exploration of the environment by the MDP. The ultimate goal is the design of an efficient policy learning mechanism for optimisation problems with long-run objectives that may be affected by large rewards observed on those rare states.

Since learning a policy requires the ability to measure the function being optimised, we start by proposing FVEE (Fleming-Viot expectation estimator), an estimation scheme of the stationary probability distribution, based on renewal theory, that will allow us to estimate long-run expectations. In the FV context, individuals are called

particles which evolve according to the MDP dynamics, and are said to be *absorbed* when reaching the condition of visiting a predefined subset $\mathcal{A}$ of the environment's state space, called the *absorption set*. When a particle is absorbed, it is instantly regenerated with the characteristic given by the position (state) of a particle chosen uniformly at random among all the other particles that are still outside $\mathcal{A}$. Thus, all particles are *forced* to stay outside $\mathcal{A}$, a set that intuitively should contain frequently visited states, as we are interested in boosting the exploration of states visited rarely.

Armed with this estimator of the stationary probability of rare states, we then introduce FVRL (Fleming-Viot reinforcement learning), an RL control algorithm that integrates FVEE to estimate the gradient of a policy gradient approach, that may also be non-zero in a handful of states rarely observed. This allows the policy gradient algorithm to learn a policy with a smaller sample complexity than Monte-Carlo.

We provide a proof of consistency of FVEE and guarantees of convergence of FVRL to a local optimizer of the parameterised objective function, valid for finite state and action spaces. We showcase their application to optimisation problems on stochastic networks, namely a one-dimensional problem on the $M/M/1$ queueing system, and a two-dimensional problem on a two-job-class loss network. The results are compared with a vanilla Monte-Carlo benchmark, used both for expectation estimation and policy learning.

The rest of the paper is organized as follows. Section 2 describes the mathematical setting of the problem, Section 3 presents the general methodology, divided into the expectation estimation problem tackled by FVEE in Section 3.2, and the control problem tackled by FVRL in Section 3.3. Section 4 presents the results of applying FVEE and FVRL on an $M/M/1$ queue system (Sections 4.1.1 and 4.1.2) and on a loss network serving two classes of jobs (Sections 4.2.1 and 4.2.2). Section 5 concludes, summarizing the assumptions and limitations, and ideas for future work.

2 Problem description

We consider a continuous-time MDP $(\mathcal{S}, \mathcal{U}, q, \mathcal{R}, \Pi)$ with a finite state space $\mathcal{S}$, finite action space $\mathcal{U}$, jump rates $q \in \mathbb{R}^+$, bounded rewards $\mathcal{R}$, and family of policies Π, under the average reward criterion. Throughout the paper, as customary in the literature (e.g. [22, Chapter 11]), we assume the continuous-time Markov process X_t^π obtained by following policy $\pi \in \Pi$ is irreducible, making the MDP ergodic. We denote by p^π the stationary probability distribution of X_t^π, and by $\mathbb{E}^\pi(\eta)$ the long-run expectation (i.e. with respect to p^π, or equivalently $\mathbb{E}^\pi(\eta) \doteq \lim_{t\to\infty} \mathbb{E}^\pi(\eta(X_t^\pi))$) of a bounded function of interest, $\eta : \mathcal{S} \to \mathbb{R}$, such as the occupation measure or the expected one-step reward.

We will be interested in estimating $\mathbb{E}^\pi(\eta)$ under the assumption that the function η is zero everywhere except in a set of states $\mathcal{C} \subset \mathcal{S}$. For example, if the rewards are assumed to be sparse, the reward function $r : \mathcal{S} \times \mathcal{U} \to \mathbb{R}$ is allowed to be non-zero only in $\mathcal{C} \times \mathcal{U}$, from which we can define the expected one-step reward function η as $\eta(x) = \sum_{a\in\mathcal{U}} r(x,a)\pi(a|x)$. Thus, the expected value of η can be reduced to a calculation over the states in set $\mathcal{C}$ as

$$\mathbb{E}^\pi(\eta) = \sum_{x\in\mathcal{C}} p^\pi(x)\eta(x), \tag{1}$$

where we have used the ergodicity of the MDP under π to write $\mathbb{E}^\pi(\eta)$ as a p^π-expectation. The objective of estimating $\mathbb{E}^\pi(\eta)$ will be achieved as follows: we will first choose an absorption set $\mathcal{A} \subset \mathcal{S}$, such that $\mathcal{C} \cap \mathcal{A} = \emptyset$. Then, we will use Fleming-Viot particle systems to construct an estimator $\hat{p}^\pi_{FV}(x)$ of the stationary probability in $\mathcal{C}$, and finally compute $\hat{\mathbb{E}}^\pi(\eta) = \sum_{x\in\mathcal{C}} \hat{p}^\pi_{FV}(x)\eta(x)$ when $\eta(x)$ is known, or $\hat{\mathbb{E}}^\pi(\eta) = \sum_{x\in\mathcal{C}} \hat{p}^\pi_{FV}(x)\hat{\eta}(x)$, for some appropriate estimator $\hat{\eta}(x)$, when only the support of $\eta(x)$ is known.

In this work we will consider three different η functions for illustration of the methodology: *(i)* the occupation measure, $\eta(x) = \mathbf{1}_{X_t^\pi = x}$ (where $\mathbf{1}_B$ is the indicator function of set B), which will allow us to estimate the stationary probability $p^\pi(x)$ of states of interest; *(ii)* the expected one-step reward function, $\eta(x) = \overline{r}(x) \doteq \sum_{a\in\mathcal{U}} r(x,a)\pi(a|x)$,

which will allow us to estimate the long-run expected reward $g \doteq \lim_{t\to\infty} \mathbb{E}^\pi(\overline{r}(X_t^\pi))$ and *(iii)* its gradient, $\nabla_\theta g(\pi_\theta)$, presented in Section 3.3, which will allow us to update a policy π_θ parameterised by $\theta \in \mathbb{R}^d$ in control problems.

In the case of control problems, in Section 4 we will consider queueing problems where the optimal admission control policy is of threshold type. To solve the problem, we will choose a parameterised policy whose gradient is non-zero only near the threshold, making the gradient $\nabla_\theta g(\pi_\theta)$ sparse. We will then use Fleming-Viot particle systems to estimate the sparse gradient as described in Section 3.3.

Remark 1. *As briefly mentioned in the Introduction, the absorption set $\mathcal{A}$ can be defined following an initial exploration of the environment, by adding states that are visited frequently enough, e.g. above an appropriate relative frequency threshold. For ease of illustration, $\mathcal{A}$ will be assumed known throughout the paper.*

Remark 2. *We make the assumption of zero reward in the absorption set $\mathcal{A}$ to concentrate the attention on our proposed methods. This assumption however can be relaxed, considering that traditional expectation estimation methods (e.g. Monte-Carlo) can be used to estimate the contribution to $\mathbb{E}^\pi(\eta)$ from frequently visited states.*

3 Methodology

In this section we present a new method to estimate $\mathbb{E}^\pi(\eta)$ tailored to Markov decision processes with sparse and rare rewards. We then discuss how to use this algorithm to improve the estimation of gradients in the context of the policy gradient methodology to approximately solve optimal control problems with sparse and rare gradients. For simplicity of exposition and unless specified otherwise, we assume that we have direct access to the function η. We also assume throughout this section that policy π and the chosen set $\mathcal{A} \subset \mathcal{S}$, whose intersection with $\mathcal{C}$ is empty, are fixed.

3.1 Definitions

The following definitions will be instrumental in our discussion. We denote by $\vec{\partial}\mathcal{A}$ the entrance boundary of $\mathcal{A}$, i.e. the set of states $x \in \mathcal{A}$ for which there exists at least a state $y \in \mathcal{A}^c$ with positive jump rate to x, i.e. $q(y,x) > 0$. The entrance boundary of $\mathcal{A}^c$, $\vec{\partial}\mathcal{A}^c$, is defined analogously. We define the first time of entry into $\mathcal{A}$ as

$$\mathcal{T}_{\mathcal{A},0} \doteq \inf\{t > 0 : X_t^\pi \in \mathcal{A} \text{ and } X_{t^-}^\pi \notin \mathcal{A}\},$$

and we denote the entrance state distribution into $\mathcal{A}$ under stationarity as:

$$p^\pi_{\vec{\partial}\mathcal{A}}(x) \doteq \mathbb{P}(X^\pi_{\mathcal{T}_{\mathcal{A},0}} = x | X_0^\pi \sim p^\pi), \forall x \in \vec{\partial}\mathcal{A}.$$

Using Figure 1 as a visual aid, we further define the first time of entry into $\mathcal{A}^c$ following $\mathcal{T}_{\mathcal{A},0}$ as $T_{\mathcal{A}^c} \doteq \inf\{t > \mathcal{T}_{\mathcal{A},0} : X_t^\pi \in \mathcal{A}^c\}$, the first time of entry into $\mathcal{A}$ following $T_{\mathcal{A}^c}$ as $T_{\mathcal{A}} \doteq \inf\{t > T_{\mathcal{A}^c} : X_t^\pi \in \mathcal{A}\}$, and their difference as $T_{\mathcal{K}} \doteq T_{\mathcal{A}} - T_{\mathcal{A}^c}$, also referred to as the killing time. The stopping time $T_{\mathcal{A}}$ will be regarded in the sequel as a cycle return time to $\mathcal{A}$.

Finally, for any measurable subset B, we define $\mathbb{P}_{\vec{\partial}\mathcal{A}}(B) \doteq \mathbb{P}(B | X_0^\pi \sim p^\pi_{\vec{\partial}\mathcal{A}})$. For the complement set $\mathcal{A}^c$, we define the entrance state distribution into $\mathcal{A}^c$ under stationarity as:

$$p^\pi_{\vec{\partial}\mathcal{A}^c}(x) \doteq \mathbb{P}_{\vec{\partial}\mathcal{A}}(X^\pi_{T_{\mathcal{A}^c}} = x), \forall x \in \vec{\partial}\mathcal{A}^c.$$

The two state probability distributions defined above, $p^\pi_{\vec{\partial}\mathcal{A}}(x)$ and $p^\pi_{\vec{\partial}\mathcal{A}^c}(x)$, will be thoroughly used in the development of our proposed methodology to condition the start state of the Markov decision process X_t^π.

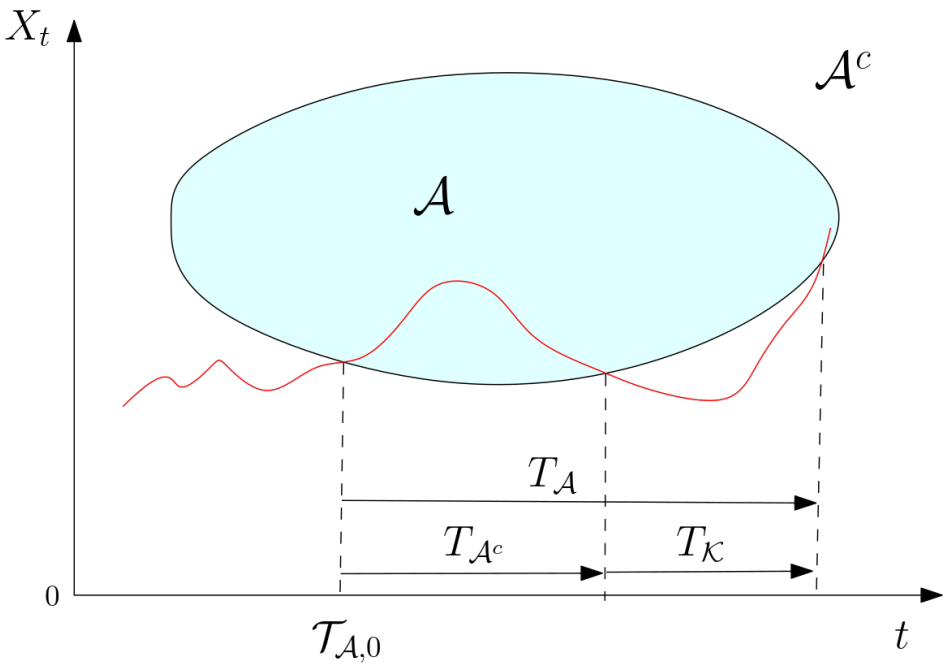


Figure 1: Sketch showing the stopping times defined in Section 3.1.

3.2 FVEE: Expectation estimation with Fleming-Viot particle systems

We now introduce the Fleming-Viot method as a way to tackle the problem of sparse and rare rewards and obtain a consistent estimator of $\mathbb{E}^\pi(\eta)$, which we call FVEE (Fleming-Viot expectation estimator). Our approach proposes a hard penalisation of trajectories that enter $\mathcal{A}$ in order to boost exploration of the subset $\mathcal{C}$ of the state space that is relevant for the estimation of the expectation, which is outside $\mathcal{A}$. The hard penalisation consists in immediately replacing trajectories that enter $\mathcal{A}$ by trajectories outside $\mathcal{A}$. To this end, we leverage the dynamics of the Fleming-Viot particle system [7] which has been used in the literature to simulate quasi-stationary distributions [7, 3].

More specifically, the Fleming-Viot N-particle system with driving process X_t^π and absorption set $\mathcal{A}$ is a continuous-time Markov process $(\xi_t^\nu)_{t\geq 0}$ defined on the state space $(\mathcal{A}^c)^N$ as follows: given a probability distribution ν on $\mathcal{A}^c$, an N-dimensional vector $\xi_0^\nu(k)_{k=1,\dots N}$ defines the initial state of the particles in the system and is obtained as i.i.d. samples from ν. Each particle $\xi_t^\nu(k)$ then evolves independently according to the dynamics of X_t^π, but whenever it hits a state in $\mathcal{A}$, it immediately jumps to the position of one of the other particles chosen uniformly at random. This mechanism allows us to only explore trajectories outside $\mathcal{A}$, which is where the informative rewards are located.

In order to exploit the Fleming-Viot particle system for the estimation of $\mathbb{E}^\pi(\eta)$, we leverage the renewal theory characterization of the stationary probability of a state in terms of return cycles [2, Chapter 6, Theorem 1.2], as follows: if we use the entrance to $\mathcal{A}$ as the event defining the beginning and end of a cycle, the stationary expectation of an arbitrary function η can be written in terms of the cycle time $T_{\mathcal{A}}$ as

$$\mathbb{E}^\pi(\eta) = \frac{\mathbb{E}_{\vec{\partial}\mathcal{A}}\big(\int_0^{T_{\mathcal{A}}} \eta(X_t^\pi)dt\big)}{\mathbb{E}_{\vec{\partial}\mathcal{A}} T_{\mathcal{A}}}. \tag{2}$$

Note that, if we used Monte-Carlo to estimate this expectation by simulating the Markov process X_t^π starting at $\vec{\partial}\mathcal{A}$ and observing cycle times $T_{\mathcal{A}}$, it might take a very long time before observing a non-zero contribution from η, as this function is assumed to be sparse with non-zero values rarely observed. The following proposition is key to defining the Fleming-Viot estimation method leading to FVEE.

Proposition 1. *Given a set $\mathcal{A} \subset \mathcal{S}$ and a function $\eta : \mathcal{S} \to \mathbb{R}$ that is non-zero on $\mathcal{C} \in \mathcal{A}^c$, the expectation $\mathbb{E}^\pi(\eta)$ is given by* (1)*, with*

$$p^\pi(x) = \frac{\int_0^\infty \mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_\mathcal{K} > t)\phi_t^{\vec{\partial}\mathcal{A}^c}(x)dt}{\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}} \qquad \forall x \in \mathcal{C}, \tag{3}$$

where $\phi_t^{\vec{\partial}\mathcal{A}^c}(x) \doteq \mathbb{P}_{\vec{\partial}\mathcal{A}^c}(X_t^\pi = x | T_\mathcal{K} > t)$ is the probability that the process X_t^π (started at a state in $\vec{\partial}\mathcal{A}^c$ chosen with probability $p^\pi_{\vec{\partial}\mathcal{A}^c}$) is in x provided it has not been absorbed into $\mathcal{A}$.

Proof. Since at the beginning of a cycle the process starts at a state in $\vec{\partial}\mathcal{A}$, in order to reach a state in the set of interest $\mathcal{C} \in \mathcal{A}^c$ (the set of states with non-zero values of η), the process needs to go through $\vec{\partial}\mathcal{A}^c$.

Thus, using the definition of the sojourn times $T_{\mathcal{A}^c}$ and $T_\mathcal{A}$ and recalling that $T_\mathcal{A} > T_{\mathcal{A}^c}$, expression (2) can be written as

$$\mathbb{E}^\pi(\eta) = \frac{\mathbb{E}_{\vec{\partial}\mathcal{A}}\big(\int_0^{T_{\mathcal{A}^c}} \eta(X_t^\pi)dt + \int_{T_{\mathcal{A}^c}}^\infty \eta(X_t^\pi)\mathbf{1}_{T_\mathcal{A} > t}dt\big)}{\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}}.$$

Since by assumption η is zero in $\mathcal{A}$, the first integral is zero. After changing the integration variable to $u = t - T_{\mathcal{A}^c}$ in the second integral and using the notation of the killing time $T_\mathcal{K} = T_\mathcal{A} - T_{\mathcal{A}^c}$, we get

$$\mathbb{E}^\pi(\eta) = \frac{\mathbb{E}_{\vec{\partial}\mathcal{A}}\big(\int_0^\infty \eta(X^\pi_{u+T_{\mathcal{A}^c}})\mathbf{1}_{T_\mathcal{K} > u}du\big)}{\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}}.$$

At $u = 0$, the Markov process is at $X^\pi_{T_{\mathcal{A}^c}}$ which, as stated in Section 3.1, is distributed according to $p^\pi_{\vec{\partial}\mathcal{A}^c}$ when $X_0^\pi \sim p^\pi_{\vec{\partial}\mathcal{A}}$, as is the case above. This allows us to redefine the time origin of the Markov process at $u = 0$ and replace $\mathbb{E}_{\vec{\partial}\mathcal{A}}$ with $\mathbb{E}_{\vec{\partial}\mathcal{A}^c}$ to obtain

$$\mathbb{E}^\pi(\eta) = \frac{\mathbb{E}_{\vec{\partial}\mathcal{A}^c}\big(\int_0^\infty \eta(X_u^\pi)\mathbf{1}_{T_\mathcal{K} > u}du\big)}{\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}}.$$

Then, expression (1) with $p^\pi(x)$ given by (3) follows from interchanging the order of the integral and the expectation (the state space is assumed finite), conditioning on the event $\mathbf{1}_{T_\mathcal{K} > t}$, and taking the summation over x (coming from the expectation in the numerator) out of the integral. ■

When $\eta(x)$ is a known function, the FVEE of η is constructed as

$$\text{FVEE}(\eta) \doteq \hat{\mathbb{E}}^\pi_{FV}(\eta) \doteq \sum_{x \in \mathcal{C}} \hat{p}^\pi_{FV}(x)\eta(x), \tag{4}$$

where $\hat{p}^\pi_{FV}(x)$ is the FV estimator of the stationary probability for all states x in $\mathcal{C}$. This estimator is constructed by estimating the denominator and the numerator in (3) using the following two-step procedure: *(i)* the denominator $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}$ is estimated using a standard Monte-Carlo estimator from observations of the return cycle time $T_\mathcal{A}$ coming from the simulation of X_t^π, and *(ii)* for each $x \in \mathcal{C}$, the numerator is estimated by summing the product of the finite-support piecewise-constant estimators of the two functions in the integral, $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_\mathcal{K} > t)$ and $\phi_t^{\vec{\partial}\mathcal{A}^c}(x)$, obtained from the simulation of the FV N-particle system driven by X_t^π with absorption set $\mathcal{A}$. The estimation details are given in Appendix A.

If only the support of $\eta(x)$ is known but not its values, an estimator $\hat{\eta}(x)$ must be obtained separately from $\hat{p}^\pi_{FV}(x)$ and then plugged into the expression for the FVEE(η) in (4). This will be the case when estimating gradients for the FVRL algorithm described in Section 3.3.1.

Consistency of FVEE(η) It has been proved that for finite state spaces [8, 9], uniform in time propagation of chaos holds for Fleming-Viot particle systems. We let $m(\cdot,\xi) : \mathcal{A}^c \to [0,1]$ denote the empirical distribution of the N particles with positions described by vector ξ, defined as the empirical mean $m(x,\xi) \doteq \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\xi(i)=x}, \forall x \in \mathcal{A}^c$. When ν is a probability measure on $\mathcal{S}$, under the assumption of proper initialization [8, Theorem 1.4] shows the following bound on the speed of convergence w.r.t. the number of particles N of the empirical mean $m(.,\xi_t^\nu)$ towards ϕ_t^ν:

$$\sup_{x\in\mathcal{S}} \sup_{t\geq 0} \mathbb{E}\Big| m(x,\xi_t^\nu) - \phi_t^\nu(x)\Big| \leq \frac{C_{\text{FV}}}{\sqrt{N}}, \tag{5}$$

where C_{FV} is a positive constant depending on the characteristics of the driving process.

Using this result, we can show the following bound for the expected absolute error for the FV estimator of the stationary probability, $\hat{p}^\pi_{FV}(x)$. The theorem is valid in an idealized case where the simulation used to estimate the denominator $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_{\mathcal{A}}$ is started according to $p^\pi_{\vec{\partial}\mathcal{A}}$, and the Fleming-Viot simulation used to estimate $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_{\mathcal{K}} > t)$ and $\phi_t^{\vec{\partial}\mathcal{A}^c}(x)$, is started from i.i.d. samples of $p^\pi_{\vec{\partial}\mathcal{A}^c}$.

Theorem 2 (Bound for the expected absolute error of $\hat{p}^\pi_{FV}(x)$)**.** *Assume that we start the simulation of X_t^π for the estimator of $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_{\mathcal{A}}$ in* (3) *according to the distribution $p^\pi_{\vec{\partial}\mathcal{A}}$, that M return cycles to $\mathcal{A}$ under stationarity are observed during that simulation, and that we compute the estimator of $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_{\mathcal{K}} > t)$ and $\phi_t^{\vec{\partial}\mathcal{A}^c}(x)$ in* (3) *using the FV particle system started at the positions of N i.i.d samples from $p^\pi_{\vec{\partial}\mathcal{A}^c}$. Then, there exists a constant $C > 0$ such that*

$$\mathbb{E}\,|\hat{p}^\pi_{FV}(x) - p^\pi(x)| \leq C\Big(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{N}}\Big), \qquad \forall x \in \mathcal{A}^c.$$

The proof is given in Appendix B.

Theorem 2 shows that the FV estimator of the stationary probability, $\hat{p}^\pi_{FV}(x)$, is consistent. And when function $\eta(x)$ is known, it implies that the FVEE(η) defined in (4) is consistent as well, since it is an η-weighted sum of the $\hat{p}^\pi_{FV}(x)$ values on the finite set $\mathcal{C}$.

A few additional comments on the theorem are in order. On the one hand, it is important to observe that Theorem 2 ensures the estimation error is of the same order as Monte-Carlo, which gives minimal convergence properties to FVEE. The constant C depends in an intricate manner on the state and action space sizes and on the precise underlying Markovian dynamics (see [8] for more details). It remains an open problem to characterize for which type of dynamics/rewards, this constant and hence the global error will be smaller than the corresponding ones for vanilla Monte-Carlo.

On the other hand, the mean square estimation error should not be our only focus to evaluate the difference between FVEE and Monte-Carlo, especially when the ultimate goal is the convergence rate of a reinforcement learning algorithm. Indeed, in the control problem, a noisy but still informative signal might be very useful compared to no signal at all. Our main idea when replacing MC by FV is to trade the observation of a very rare event in the original problem by the observation of a more common event for the FV particle system.

Although a fully rigorous analysis of the large deviation probability to observe a non-zero reward is out of the scope of this paper (see [11] for a recent analysis in the context of random walks and Lévy processes), we can give some intuition using the results of [8, 12]. The dynamics of a tagged particle of the FV process converges when $N \to \infty$ to a one-dimensional Markov process having the quasi-stationary distribution ν_{QS} of the original absorbed process as stationary measure (see [12]), where

$$\nu_{QS}(B) = \lim_{t\to\infty} \phi_t^{\vec{\partial}\mathcal{A}^c}(B),$$

for any set of states B. If the state space is finite, this one dimensional process in turn converges in distribution exponentially fast to its stationary distribution, ν_{QS}. Hence, the probability of finding a non-zero signal (by visiting

states in set $\mathcal{C}$) for the FV process in a finite time interval is of the order of $\nu_{QS}(\mathcal{C})$, which can be considerably larger than the original $p^{\pi}(\mathcal{C})$.

As an illustration, we bring back the $M/M/1$ queue example with load $\rho < 1$ and admission control policy that rejects an incoming job when the queue size is K. Thus $\mathcal{C} = \{K\}$. If the absorption set $\mathcal{A}$ is chosen as the singleton containing the high probability state of zero jobs in the queue, the quasi-stationary distribution of the absorbed process has $\nu_{QS}(K) \sim \sqrt{K}\rho^{K/2}$ [10], which is considerably larger than the stationary probability of the original process at K, given by $p^{\pi}(K) \sim \rho^K$. For example, $\nu_{QS}(K)$ is about 8 thousand times larger than $p^{\pi}(K)$ when $K = 40$ and $\rho = 0.7$, and 6 million times larger when $K = 40$ and $\rho = 0.5$.

3.3 FVRL: Policy gradient learning with Fleming-Viot particle systems

In this subsection we show how the Fleming-Viot expectation estimator (FVEE) introduced in Section 3.2 can be combined with the policy gradient algorithm to approximately solve optimal control problems in environments with sparse and rare rewards under the average reward criterion. This leads to FVRL, the Fleming-Viot reinforcement learning method.

We provide here a general description of the FVRL method to speed up learning in those types of scenarios. The methodology will be illustrated on two network optimisation problems described in Section 4: an $M/M/1$ queue, and a loss network system serving two classes of jobs.

Let us consider in detail the case in which an agent interacts with an environment (available in practice through a simulator or emulator), with the aim of maximizing the long-run expected reward. Following the customary RL framework, we consider a discrete-time representation X_n^{π} of the continuous-time process X_t^{π} such that both processes have the same stationary probability distribution. This can be achieved by different methods, such as uniformization [22, Chapter 11.5] or, in the case of queueing systems with Poisson arrivals, by defining X_n^{π} as the chain embedded at arrival times [5] (using PASTA). In our queueing examples we will rely on the latter.

As customary in the literature, we let X_n, A_n and R_{n+1} respectively denote the state, action and reward at the n-th discrete time step, $n \geq 0$. We denote by π_θ the policy function parameterised by $\theta \in \mathbb{R}^d$, and we propose to use a gradient-based algorithm to learn the parameter θ that maximizes $g(\pi_\theta)$, the long-run expected reward [25, Chapter 13],

$$\begin{aligned} g(\pi_\theta) &= \lim_{H\to\infty} \frac{1}{H} \sum_{n=1}^{H} \mathbb{E}^{\pi_\theta}[R_n | X_0, A_{0:n-1} \sim \pi_\theta] \\ &= \sum_{x\in\mathcal{S}} p^{\pi_\theta}(x) \sum_{a\in\mathcal{U}} r(x,a)\pi_\theta(a|x), \end{aligned} \tag{6}$$

where the last equality comes from the ergodicity assumption of the MDP under all parameterised π_θ policies considered, making the long-run expected reward independent of the start state X_0.

It follows from the policy gradient theorem for continuing learning problems [25, 26] that the gradient of $g(\pi_\theta)$ can be written as:

$$\begin{aligned} \nabla_\theta g(\pi_\theta) &= \sum_{x\in\mathcal{S}} p^{\pi_\theta}(x) \sum_{a\in\mathcal{U}} Q^{\pi_\theta}(x,a) \nabla_\theta \pi_\theta(a|x) \\ &= \sum_{x\in\mathcal{S}} p^{\pi_\theta}(x) \nabla_\theta g_x(\pi_\theta), \end{aligned} \tag{7}$$

where we have defined the contribution to the gradient by state x as

$$\nabla_\theta g_x(\pi_\theta) \doteq \sum_{a\in\mathcal{U}} Q^{\pi_\theta}(x,a) \nabla_\theta \pi_\theta(a|x), \tag{8}$$

and

$$Q^{\pi_\theta}(x,a) \doteq \mathbb{E}^{\pi_\theta}\left[\sum_{n=1}^{\infty}(R_n - g(\pi_\theta)) \mid X_0 = x, A_0 = a\right] \tag{9}$$

is the action value function for continuing learning problems under the average reward criterion.

We note that the gradient in (7) has the form of the expectation in (1) if we let $\eta(x) = \nabla_\theta g_x(\pi_\theta)$ and define the set $\mathcal{C} = \{x \in \mathcal{S} : \nabla_\theta g_x(\pi_\theta) \neq 0\}$. Therefore, we can leverage FVEE to obtain an estimate of the gradient in (7) by combining the FV estimator of the stationary probability presented in Section 3.2, for states $x \in \mathcal{C}$, with an estimate of $\nabla_\theta g_x(\pi_\theta)$. This leads to the FVRL algorithm described in detail below. We note that the advantage of our approach lies on the fact that in a wide class of queueing problems, a smart choice of the parameterised policy leads to a sparse $\nabla_\theta g_x(\pi_\theta)$ as a function of x. This is the case for instance in problems in which the optimal policy is of threshold type where, as we will see in Section 4.1.2, an appropriate parameterisation yields $\nabla_\theta g_x(\pi_\theta) = 0$ in all states x except in a neighbourhood of the thresholds[2].

3.3.1 FVRL algorithm

The FVRL algorithm is a gradient ascent algorithm that uses estimates of the gradient defined in (7) to learn a policy that is locally optimal in the parameter space. Assuming a policy parameterisation $\nabla_\theta \pi(a|x)$ with sparse gradients in x and known support $\mathcal{C}$, the gradient in (7) will also be sparse in x. Thus, we propose to estimate the gradient $\nabla_\theta g(\pi_\theta)$ using (4), i.e. by estimating the stationary probability $p^{\pi_\theta}(x)$ for $x \in \mathcal{C}$ using an FV particle system, as described in Section 3.2. However, contrary to the estimation problem presented in that section, the η function in this case, $\eta(x) = \nabla_\theta g_x(\pi_\theta)$ defined in (8), is only known up to its support $\mathcal{C}$, as $Q^{\pi_\theta}(x,a)$ is unknown. This requires that $\nabla_\theta g_x(\pi_\theta)$ be estimated as well (as explained below) to obtain the final estimator of the gradient as:

$$\hat{\nabla}_\theta g(\pi_\theta) = \sum_{x \in \mathcal{C}} \hat{p}^{\pi_\theta}_{FV}(x) \hat{\nabla}_\theta g_x(\pi_\theta).$$

Estimation of $\nabla_\theta g_x(\pi_\theta)$ We propose an estimation procedure that leverages the sum-to-zero property of the policy gradient in combination with a coupling argument, as described in the following lemma.

Lemma 3. *Given a state $x \in \mathcal{S}$, let $\{X_n^{(x,a)}\}_{a \in \mathcal{U}, n \in \mathbb{N}}$ be a family of $|\mathcal{U}|$ identical Markov processes driven by policy $\pi_\theta(\cdot|x)$, all starting at x, and each taking one of the different possible actions $a \in \mathcal{U}$ at time $n = 0$. Define the coupling time τ when all such chains meet, i.e. $\tau = \inf\{n > 0 : X_n^{(x,a)} = X_n^{(x,a')}, \forall a, a' \in \mathcal{U}\}$. Then, the function $\nabla_\theta g_x(\pi_\theta)$ of x, defined in* (8)*, can be written as:*

$$\nabla_\theta g_x(\pi_\theta) = \sum_{a \in \mathcal{U}} \mathbb{E}^{\pi_\theta}\left[\sum_{n=1}^{\tau} R_n | X_0 = x, A_0 = a\right] \nabla \pi_\theta(a|x), \qquad \forall x \in \mathcal{S}.$$

Proof. After the stopping time τ, the contribution to $Q^{\pi_\theta}(x,a)$ in (9) does not depend anymore on the initial action a taken by each chain $X_n^{(x,a)}$ because they all have the same expected trajectory under the fixed policy $\pi_\theta(\cdot|x)$. Hence,

[2]Even if the gradient is not sparse, it may be the case that important contributions to its value comes, via $\nabla_\theta g_x(\pi_\theta)$, from states x that are rarely observed. Failing to observe these states frequently enough may hinder the convergence speed of the optimisation process to their actual optimum.

we can write $\nabla_\theta g_x(\pi_\theta)$ in (8) as

$$\begin{aligned}
\nabla_\theta g_x(\pi_\theta) &= \sum_{a\in\mathcal{U}} \mathbb{E}^{\pi_\theta}\left[\sum_{n=1}^{\infty}(R_n - g(\pi_\theta))\,\big|X_0 = x, A_0 = a\right]\nabla\pi_\theta(a|x) \\
&= \sum_{a\in\mathcal{U}} \mathbb{E}^{\pi_\theta}\left[\sum_{n=1}^{\tau}(R_n - g(\pi_\theta))\,\big|X_0 = x, A_0 = a\right]\nabla\pi_\theta(a|x) \\
&\quad + \mathbb{E}^{\pi_\theta}\left[\sum_{n=\tau+1}^{\infty}(R_n - g(\pi_\theta))\,\big|X_0 = x\right]\sum_{a\in\mathcal{U}}\nabla\pi_\theta(a|x) \\
&= \sum_{a\in\mathcal{U}} \mathbb{E}^{\pi_\theta}\left[\sum_{n=1}^{\tau} R_n\big|X_0 = x, A_0 = a\right]\nabla\pi_\theta(a|x) - \mathbb{E}^{\pi_\theta}[\tau]g(\pi_\theta)\sum_{a\in\mathcal{U}}\nabla\pi_\theta(a|x) \\
&= \sum_{a\in\mathcal{U}} \mathbb{E}^{\pi_\theta}\left[\sum_{n=1}^{\tau} R_n\big|X_0 = x, A_0 = a\right]\nabla\pi_\theta(a|x),
\end{aligned}$$

where we have repeatedly used the sum-to-zero property of the policy gradient, namely $\sum_{a\in\mathcal{U}}\nabla\pi_\theta(a|x) = \nabla\sum_{a\in\mathcal{U}}\pi_\theta(a|x) = \nabla 1 = 0, \forall x \in \mathcal{S}$. ■

Using Lemma 3, we propose an estimator of $\nabla_\theta g_x(\pi_\theta)$ based on simulating the $|\mathcal{U}|$ copies of the process under policy π_θ as described by the lemma. When any two copies meet at the same state, we impose they continue evolving together forever. The simulation stops when either all copies meet or when a maximum number of steps taken by each copy is reached. Note that, due to Doeblin's theorem on coupling, [5, Section 4.3.1], the coupling time τ of all processes is almost surely finite and actually has an exponential moment when the state space is finite.

We remark that this estimation procedure can only improve over the traditional estimation procedure of independently simulating separate process copies for a predefined number of steps S. In fact, should the coupling time of all processes be smaller than S, the simulation will terminate earlier than the traditional method which only terminates after each copy has performed S steps.

The algorithm The FVRL policy gradient algorithm is summarized in Algorithm 1.

Algorithm 1 FVRL algorithm for finding a local optimizer $\theta^* \in \mathbb{R}^d$ of $g(\pi_\theta)$, the long-run expected reward defined in (6), using a parameterised policy π_θ for the MDP introduced in Section 2.

Data:

A. Simulator or emulator of the MDP.

B. Absorption set $\mathcal{A}$ required by the FVEE algorithm presented in Section 3.2.

C. N_0, M_0: respectively, the initial number of FV particles and the initial number of return cycles to $\mathcal{A}$, used by the FVEE algorithm to estimate the stationary probability distribution, p^{π_θ}. We let N_k and M_k denote the value of the same concepts at policy iteration k.

D. V, S_0: respectively, the number of replications and the initial maximum number of time steps allowed to estimate the function $\nabla_\theta g_x(\pi_\theta)$ defined in (8), with $\theta \in \mathbb{R}^d$. We let S_k denote the value of the same concept at policy iteration k.

E. $\mathcal{K}$: Compact set defining the support for θ, e.g. a hyper-rectangle.

F. θ_0: initial guess of parameter $\theta \in \mathcal{K}$.

G. $\epsilon > 0$: exponent defining the rates of increase of N_k, M_k and S_k in terms of policy iteration k.

H. $\alpha_0 > 0$: initial learning rate or step size, used in the θ update performed by the algorithm.

Result: An estimate of a local optimizer $\theta^* \in \mathbb{R}^d$ for the long-run expected reward $g(\pi_\theta)$.

Initialize policy iteration, $k \leftarrow 0$

Let $\mathcal{S}'$ be the set of states where the policy gradient is not zero for at least one action, i.e. $\mathcal{S}' = \{x \in \mathcal{S} : \exists \text{ action } a \in \mathcal{U} \text{ s.t. } \nabla_{\theta_k} \pi_{\theta_k}(a|x) \neq 0\}$.

repeat

a) Simulate or emulate V times $|\mathcal{U}|$ parallel copies of the $\{X_n\}_{n\in\mathbb{N}}$ process, each starting at $x \in \mathcal{S}'$ and at a different $a \in \mathcal{U}$, and collect the rewards $\{R_{v,n}^{(x,a)}\}_{n\geq 1}$ observed by each a-th copy for replication v. Strongly couple any two chains after they meet and run the simulation until all $|\mathcal{U}|$ simulated chains meet or until S_k steps are taken on each parallel chain.

b) Estimate the contribution to the gradient by each $x \in \mathcal{S}'$:
$\hat{\nabla}_\theta g_x(\pi_{\theta_k}) \leftarrow \frac{1}{V} \sum_{v=1}^{V} \sum_{a\in\mathcal{U}} \sum_{n=1}^{\min(\tau_v, S_k)} R_{v,n}^{(x,a)} \nabla_{\pi_{\theta_k}} \pi_{\theta_k}(a|x)$,
where τ_v is the coupling time of all $|\mathcal{U}|$ chain copies described above for replication v.

c) Compute $\hat{p}_{FV}^{\pi_{\theta_k}}(x)$, the FV estimator of the stationary probability $p^{\pi_{\theta_k}}$ using N_k particles and M_k return cycles to $\mathcal{A}$, as described in Section 3.2.

d) $\hat{\nabla}_{\theta_k} g(\pi_{\theta_k}) \leftarrow \sum_{x\in\mathcal{S}'} \hat{p}_{FV}^{\pi_{\theta_k}}(x) \hat{\nabla}_\theta g_x(\pi_{\theta_k})$

e) $\theta_{k+1} \leftarrow \mathcal{P}_\mathcal{K} [\theta_k + \alpha_k \hat{\nabla}_{\theta_k} g(\pi_{\theta_k})]$, where $\mathcal{P}_\mathcal{K}$ is a projection operator on compact set $\mathcal{K}$.

f) $k \leftarrow k + 1$

g) Increase N_k, M_k and S_k as $N_k = \lceil N_0 k^\epsilon \rceil$, $M_k = \lceil M_0 k^\epsilon \rceil$, $S_k = \lceil S_0 \log k^\epsilon \rceil$, where $\lceil \cdot \rceil$ is the ceiling operator.

h) Reduce the learning step by $\alpha_k = \alpha_0 / k$

until done;

3.3.2 Convergence guarantees of policy gradient learning with FVRL

FVRL is a stochastic gradient ascent algorithm with a biased gradient estimate. In this context, Theorem 2.1 of [14, Section 5.2] can be used to provide a convergence guarantee to a local optimizer of $g(\pi_\theta)$. The convergence property and the conditions under which it is valid are stated in the following proposition.

Proposition 4 (Convergence of FVRL to a local optimizer of $g(\pi_\theta)$)**.** *Assume $\pi_\theta, \theta \in \mathcal{K} \subset \mathbb{R}^d$ compact, is continuously differentiable. Assume moreover that the hyperparameters defined in the FVRL policy gradient Algorithm 1, scale as follows:*

$$M_k \sim k^\epsilon,\ N_k \sim k^\epsilon,\ S_k \sim \log k^\epsilon,\ \text{ and } \alpha_k \sim 1/k.$$

Then the FVRL policy gradient algorithm converges to a local optimizer in $\mathcal{K}$ of the long-run expected reward $g(\pi_\theta)$ defined in (6)*.*

The proof is given in Appendix C and makes use of the following lemma bounding the finite sample bias $\beta = \mathbb{E}^{\pi_\theta}\left[\hat{\nabla}_\theta g(\pi_\theta)\right] - \nabla_\theta g(\pi_\theta)$ of the FVRL estimator of the gradient.

Lemma 5 (Bias of the estimated gradient)**.** *The bias β of the FVRL estimator of the gradient of the long-run expected reward is bounded by*

$$|\beta| \leq A\left(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{N}}\right) + Be^{-cS} + Ce^{-cS}\left(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{N}}\right),$$

for some constants $A, B, C, c > 0$, where M and N are defined in Theorem 2, and S is the maximum number of time steps allowed in the simulation used to estimate the function $\nabla_\theta g_x(\pi_\theta)$ defined in (8)*.*

The proof is given in Appendix C.

We note that under a general parameterised policy setting, global convergence is only ensured under convexity assumptions. See for example [19] for the finite sample convergence analysis of stochastic gradient descent under a convex objective, and [1] for a detailed convergence analysis of policy gradient methods under the discounted reward criterion.

4 Application to stochastic networks with blocking

In this section we apply the methodologies outlined in Sections 3.2 and 3.3 to two different systems:

1. an $M/M/1$ queue, which is a unidimensional state system that will help us illustrate the basic concepts;
2. a loss network system with R servers receiving two classes of jobs. which we will use to illustrate the application to a multidimensional state system.

In both cases, we assume there exists a small subset $\mathcal{C}$ of the state space where costs may be generated due to the rejection of an incoming job (blocking), and that there is no reward for job acceptance. This makes the reward landscape sparse. For each system we will consider two goals:

a) the accurate estimation of the long-run expected cost, $\mathbb{E}^\pi(\eta)$, defined in (1), where $\eta(x)$ is the cost of rejecting an incoming job when the system is at state x;

b) the learning of the blocking sizes that minimize $\mathbb{E}^\pi(\eta)$.

To simplify the exposition and to be able to do theoretical computations against which results are compared, we fix the parameters of the underlying Markov process to known values. Nevertheless, the method is designed to be applied in practice to systems with possibly unknown parameters, the only condition being that the system can be simulated or emulated, i.e. that its next state and the reward received are known once an action is applied at any given system state.

The algorithms for the FVEE estimation of the expected cost and of its minimisation using FVRL, applied to both the $M/M/1$ queue system and the loss network, are presented in Algorithms 2 and 3 in Appendix E, respectively, and implemented in the following repository: https://github.com/mastropi/RL, which includes a few details about the estimation and learning process. We note that to keep implementation simple the values of hyperparameters M, N, and S are not scaled by the policy iteration step as required by Proposition 4.

In each application problem, the method's performance is compared with vanilla Monte-Carlo (MC) which is used as benchmark. To ensure a fair comparison, the MC estimation of expectations is based on the same number of events as the FVEE estimation, and the MC learning is based on the same number of events per policy iteration as the average number of events observed in FVRL across the whole policy learning process. The details of the estimation and learning processes by MC are further described in each subsection below.

4.1 M/M/1 system

We consider the MDP representing the dynamics of the well-known $M/M/1$ queue system under threshold-type admission control policies. Following the terminology defined in Section 2, for some $L \in \mathbb{N}$, the MDP can be described as: $(\mathcal{S} = \{0, 1, \ldots, L\}, \mathcal{U} = \{0, 1\}, q = \{\lambda, \mu\}, \mathcal{R} = r(x, a) = \mathbf{1}_{\{x=K, a=0\}}, \overline{\Pi}_L)$ where the Markov process X_t^π is the length of the queue having job arrival rate λ and service rate μ, actions 0 and 1 respectively indicate "reject" and "accept" of an incoming job, and $\overline{\Pi}_L$ is the class of admission control policies that reject an incoming job at just one state, when $X_t^\pi = K$ for some $0 \leq K \leq L$.

All presented results use $\lambda = 0.7$ and $\mu = 1.0$.

4.1.1 Estimation of the expected cost using FVEE

We apply the methodology outlined in Section 3.2 to efficiently estimate the long-run expected cost (thereafter "expected cost") when blocking is a rare event. Since the rejection cost is 1, estimating the expected cost is tantamount to estimating the blocking probability, which can be quite small depending on the $M/M/1$ system parameters and admission control policy (see numerical examples in the Introduction).

In order to estimate the blocking probability using Fleming-Viot, we define the function η introduced in Section 2 as $\eta^{\pi_\theta}(x) = \mathbf{1}_{\{x=K\}}$, which is sparse and non-zero in the single-state set $\mathcal{C} = \{K\}$. As a consequence and using the PASTA property of Poisson arrivals, the expectation $\mathbb{E}^\pi(\eta)$ in (1) becomes the blocking probability, which can be written as

$$p^\pi(K) = \frac{\int_0^\infty \phi_t^J(K) \mathbb{P}_J(T_\mathcal{K} > t) dt}{\mathbb{E}_{J-1} T_\mathcal{A}}, \tag{10}$$

where we have used that any set $\mathcal{A} = \{0, 1, \ldots, J-1\}$ with $J \leq K$ is a valid absorption set, making $\vec{\partial} A$ and $\vec{\partial} A^c$ two single-state entrance boundary sets equal to $\{J-1\}$ and $\{J\}$, respectively, which makes it possible to simplify (3) into (10).

According to Theorem 2, the FV estimator of $p^\pi(K)$ converges to its true value as both the number N of particles of the FV system, and the number M of return cycles to $\mathcal{A}$ increase. While N is a hyperparameter that is easy to define, as it is a fixed number, M is not as straightforward, as it is a random number (it depends on the number of return cycles to $\mathcal{A}$ observed during the Markov chain excursion). We will therefore parameterise the Markov chain simulation for the estimation of the denominator, $\mathbb{E}_{J-1} T_\mathcal{A}$, by the number of arrival events T that must be observed

before ending the simulation, which we will refer to as "the number of arrival events for estimating $\mathbb{E}(T_{\mathcal{A}})$" thereafter (we omit the subscript of the expectation for conciseness).

The three quantities in (10) contributing to the blocking probability are estimated following the steps described in Algorithm 2 in Appendix E, which implements the estimation methodology described in Appendix A.

Remark 3. *There is a trade-off in the choice of J, the state that defines the size of the absorption set $\mathcal{A}$: for smaller J, the return times to $\mathcal{A}$ will be smaller, requiring a smaller number T of arrival events for estimating $\mathbb{E}(T_{\mathcal{A}})$ in* (10)*; at the same time, however, visiting the rare blocking state K becomes rarer, requiring a larger number of particles N for an accurate estimation of the numerator in* (10)*. The opposite is true for larger values of J. A detailed analysis on this trade-off, in terms of the impact on estimation accuracy of different N and T choices given the value of J, is presented in Appendix D.*

We now study the convergence of FVEE as either N or T increases, and compare it with the benchmark MC estimator, obtained from a direct application of expression (2), i.e. as the fraction of the time spent at state K and the total time of return cycles to the initial state $x = J - 1$ observed during the simulation. To guarantee a fair comparison between the two methods, we start the simulation at $x = J - 1$, so that both methods start at the same "distance" from the blocking state K, and let the simulation run until the same number of events observed by FVEE is reached.

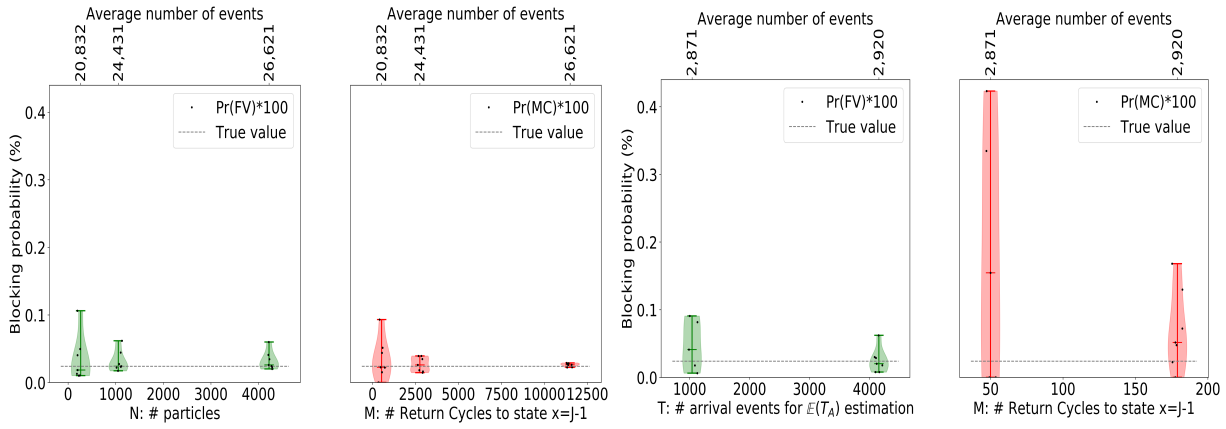
Figure 2 shows violin plots of FVEE and of the MC estimator, as N increases on the left column plots, and as T increases on the right column plots. Their values were chosen as described in Appendix D. We considered the cases $K = 20$ (top row) and $K = 40$ (bottom row), which are regarded to represent moderate and large capacities based on their blocking probabilities at the considered value for $\rho = 0.7$ of order 10^{-4} and 10^{-7}, respectively. The size J of the absorption set $\mathcal{A}$ is held fixed at $J = 12$, which corresponds to choosing the states with stationary probability larger than 0.5%, for both $K = 20$ and $K = 40$.

We observe the following in terms of convergence of FVEE and of the MC estimators to the true blocking probability, which in the plots is represented by a horizontal gray line:

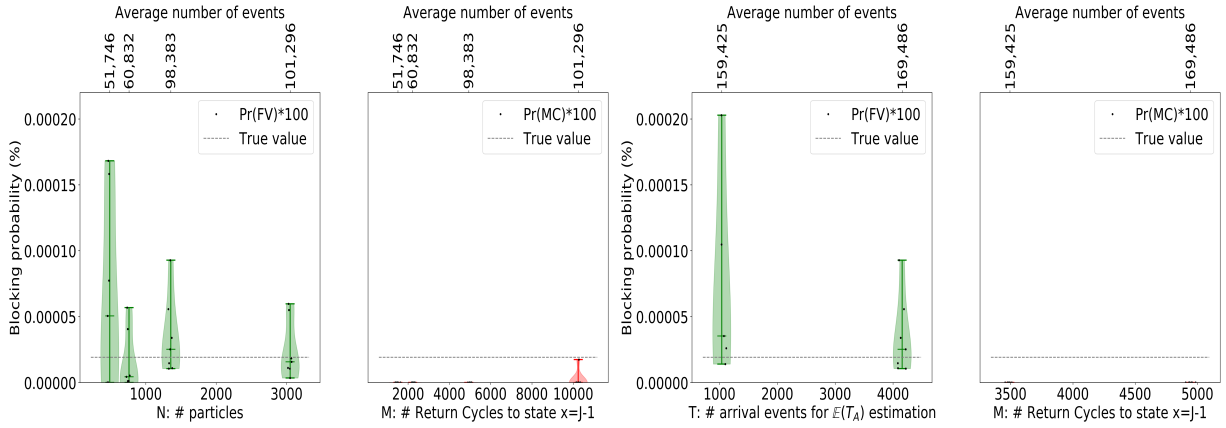
1. Both FVEE and MC converge to the true blocking probability when $K = 20$ (Figures 2(a,b)). MC presents a smaller variability than FVEE in the convergence analysis with N (Figure 2(a)) but a larger variability than FV in the convergence analysis with T (Figure 2(b)). This is due to the fact that Figure 2(a) is obtained from simulations whose events largely outnumber the events in Figure 2(b) by as much as 10 times ($\sim 20,000$ vs. $\sim 3,000$), which allows MC to observe the blocking event at $x = K$ frequently enough for an accurate estimation. On the other hand, Figure 2(b) tells us that a much smaller number of events ($\sim 3,000$) is sufficient for FVEE to estimate the blocking probability accurately enough, but is not sufficient for the MC estimator. This demonstrates the higher efficiency of FV than MC in discovering the rare event at K.

2. When $K = 40$ (Figures 2(c,d)) the MC estimator basically fails as it almost never observes the blocking state K.

3. Increasing N has a larger impact on increasing FVEE's computational complexity than increasing T, but with no particular gain on the estimation accuracy or variability. This conclusion is obtained by observing that the number of events at the rightmost violin plot in Figure 2(a) ($\sim 20,000$ events), where $N \sim 4,000$ and $T \sim 4,000$, is about 10 times larger than the number of events at the rightmost violin plot in Figure 2(b) ($\sim 3,000$ events), where $N = 30$ and $T \sim 4,000$, while the distribution of the estimated probabilities is almost the same. That is, a 10-fold increase of the number of particles N increased ten times the total number of observed events but did not increase the estimation accuracy nor decrease its variability. On the other hand, a mere 4-fold increase in T does not impact the order of magnitude of the total number of observed events (see panels (b) and (d)) but decreases the variability of the blocking probability estimate (to about half in the case of panel (d)). This confirms the conclusion of the analysis presented in Appendix D that a larger value of T is more important for a quality estimation of the blocking probability than a larger value of N in the $M/M/1/K$ queue system.



(a) $K = 20, Pr(K) = 2.4 \times 10^{-5} : T = 4157$ (b) $K = 20, Pr(K) = 2.4 \times 10^{-5} : N = 30$



(c) $K = 40, Pr(K) = 1.9 \times 10^{-7} : T = 4157$ (d) $K = 40, Pr(K) = 1.9 \times 10^{-7} : N = 1356$

Figure 2: Violin plots showing the Fleming-Viot (left, green) (ours) and Monte-Carlo (right, red) estimations of the blocking probability for an $M/M/1$ queue system with capacity K and $\lambda = \rho = 0.7$, for two different capacities $K = 20, 40$, and different hyperparameter settings N and T of the Fleming-Viot expectation estimator (FVEE). In total, 8 replications were run for each hyperparameter setting, but not all of them yielded an estimate of the blocking probability by FVEE due to a requirement of observing at least 5 cycles in the estimation of $\mathbb{E}(T_{\mathcal{A}})$. The left plots (a) and (c) analyze the convergence characteristics of the estimator as the number of particles N increases with T fixed at the indicated value. The right plots (b) and (d) analyze the convergence characteristics of the estimator as the number T of arrival events for estimating $\mathbb{E}(T_{\mathcal{A}})$ increases with N fixed at the indicated value. The absorption set size is $J = 12$ in all cases, whose value was chosen as described in the text. The interior colored horizontal line in each violin plot represents the median. In the corresponding MC experiments, since neither N nor T are defined as hyperparameters of the simulation, they cannot be used as horizontal axis. In their place, the average of the number M of observed return cycles to the initial state $J - 1$, across replications, is shown at the bottom axis. In both the FVEE and the MC plots, the top horizontal axis reports the average number of events observed across replications, which coincide in FVEE and MC by experiment design (see Section 4).

4.1.2 Learning an optimal blocking size K with FVRL

We illustrate the FVRL algorithm for learning an optimal blocking size K by defining a cost function of rejecting an incoming job that is exponentially increasing with the queue size at the time of the job arrival. Note that the cost needs to be exponentially increasing with the queue size in order to obtain an optimisation problem with a non-trivial optimum (i.e. where the optimum K is finite). This is due to the fact that the stationary probability of the $M/M/1$ queue is exponentially decreasing with the queue size when $\rho < 1$.

More precisely, given the queue size x, the cost function is defined as $r(x, a) = B(1 + b^{x - x_{\text{ref}}})\mathbf{1}_{\{a=0\}}$, where B, b

and x_{ref} are positive constants. As described in Section 4.1, the MDP is driven by a threshold-type policy $\overline{\pi} \in \overline{\Pi}_L$, where $\overline{\pi}$ rejects an incoming job when $x = K$ for some $0 \leq K \leq L$. Under policy $\overline{\pi}$, the long-run expected cost to minimize becomes:

$$g(\overline{\pi}) = p^{\overline{\pi}}(K)r(K,0) = \frac{(1-\rho)\rho^K}{1-\rho^{K+1}} B(1 + b^{K-x_{\text{ref}}}). \tag{11}$$

Given the queue load $\rho < 1$, we choose $b > 1/\rho^{1+\rho}$ so that this objective function is a convex function of K with a non-trivial (i.e. finite) minimum[3]. The value of x_{ref} is the reference queue size that is instrumental in defining the optimum threshold, K^* (see Figure 3).

We now use the FVRL Algorithm 1 presented in Section 3.3.1 to learn the optimum threshold K^*. Following [17], we propose a parameterised acceptance policy $\pi(a = 1|x)$ that is a linear step function of the state x, which is deterministic for x outside the interval $(\theta, \theta + 1)$ and decreases linearly from 1 to 0 within the interval. That is, the acceptance policy parameterised by the positive real-valued θ, is defined as:

$$\pi_\theta(a = 1|x) = \begin{cases} 1 & \text{if } x \leq \theta, \\ 1 - (x - \theta) & \text{if } \theta < x < \theta + 1, \\ 0 & \text{if } x \geq \theta + 1. \end{cases} \tag{12}$$

Note that the policy is fully deterministic for integer-valued θ, in which case the blocking size is $K = \theta + 1$. Otherwise, K is defined as $\lceil \theta \rceil + 1$.

Using (7), the gradient of $g(\pi_\theta)$ becomes

$$\frac{\partial g(\pi_\theta)}{\partial \theta} = p^{\pi_\theta}(K-1)\left[Q^{\pi_\theta}(K-1,1) - Q^{\pi_\theta}(K-1,0)\right], \tag{13}$$

where $K - 1$ is the smallest integer that is larger than or equal to θ. We observe that this parameterisation leads, as expected, to gradients being 0 for $x \neq K - 1$, that is, the gradient function is sparse in x. We note also that the gradient is discontinuous at θ and $\theta + 1$, making the assumptions of Proposition 4 not fully satisfied. However, these two points have measure zero and therefore, with probability 1, no discontinuity is observed.

Under the piecewise-linear parameterised policy proposed in (12), the long-run expected cost becomes:

$$\begin{aligned} g(\pi_\theta) &= p^{\pi_\theta}(K-1)r(K-1,0)(K-1-\theta) + p^{\pi_\theta}(K)r(K,0) \\ &= p^{\pi_\theta}(\lceil\theta\rceil)r(\lceil\theta\rceil,0)(\lceil\theta\rceil-\theta) + p^{\pi_\theta}(\lceil\theta\rceil+1)r(\lceil\theta\rceil+1,0), \end{aligned} \tag{14}$$

where we have used that $K = \lceil \theta \rceil + 1$, with $\lceil \cdot \rceil$ the ceiling operator.

Details of the learning algorithm are given in Algorithm 3 in Appendix E.

To illustrate the FVRL algorithm, we consider an MDP with the following characteristics: the system load is $\rho = 0.7$, the blocking cost function $r(x, a)$ is defined with the parameters $b = 3$ ($> 1/\rho^{1+\rho} = 1.834$ as indicated above), $B = 5$, and $x_{\text{ref}} = 18$, giving $K^* = 17$. To help understand the optimisation problem, Figure 3 shows a plot of the long-run expected cost, $\mathbb{E}^\pi(C) = \sum_{x\in\mathcal{S}} p^\pi(x) \sum_{a\in\mathcal{U}} r(x,a)\pi(a|x)$, both for the threshold-type deterministic policy $\pi = \overline{\pi}$ as a function of K, and for the parameterised stochastic piecewise-linear policy $\pi = \pi_\theta$ defined in (12).

The setup of the learning experiments is as follows: we choose the value $(x_{\text{ref}} + 10)$ for the initial blocking size guess, so that, already at the onset, blocking occurs rarely. Since the value of K is no longer fixed (as was the case

[3] The convexity of $g(\overline{\pi})$ when $b > 1/\rho^{1+\rho}$ is derived as follows: we regard $g(\overline{\pi})$ as a function $f(x)$ of a real-valued variable x taking the place of the integer-valued K. We then write $f(x)$ as $Ah(x) + Bp^{\overline{\pi}}(x)$, where $A = Bb^{-x_{\text{ref}}}(1-\rho)$ and $h(x) = (\rho b)^x/(1-\rho^{x+1})$. It is easy to see that $p^{\overline{\pi}}(x)$ is convex for all $x > 0$, so it suffices to show that $h(x)$ is convex when $b > 1/\rho^{1+\rho}$. We get: $h'(x) = \frac{h(x)\log(\rho b)}{1-\rho^{x+1}}$ and $h''(x) = \frac{h(x)\log^2(\rho b)}{(1-\rho^{x+1})^2}[1 + \rho^{x+1}\frac{\log b}{\log(\rho b)}]$. The root of the first derivative is $x^* = x_{\text{ref}} + \frac{\log(-\log\rho/\log(\rho b))}{log(b)}$ as long as $b > 1/\rho$ (this guarantees that $\log(\rho b) > 0$, making the argument of the outer log positive when $\rho < 1$). Under this condition, the convexity of $h(x)$ is assured as long as $1 + \rho^{x+1}\log\rho/\log(\rho b) > 0$, or equivalently $b > 1/\rho^{1+\rho^{x+1}}, \forall x \geq 0$ (recall $\log(\rho b) > 0$). Since $1/\rho^{1+\rho^{x+1}} \leq 1/\rho^{1+\rho}, \forall x \geq 0$ when $\rho < 1$, it suffices that $b > 1/\rho^{1+\rho}$, as claimed.

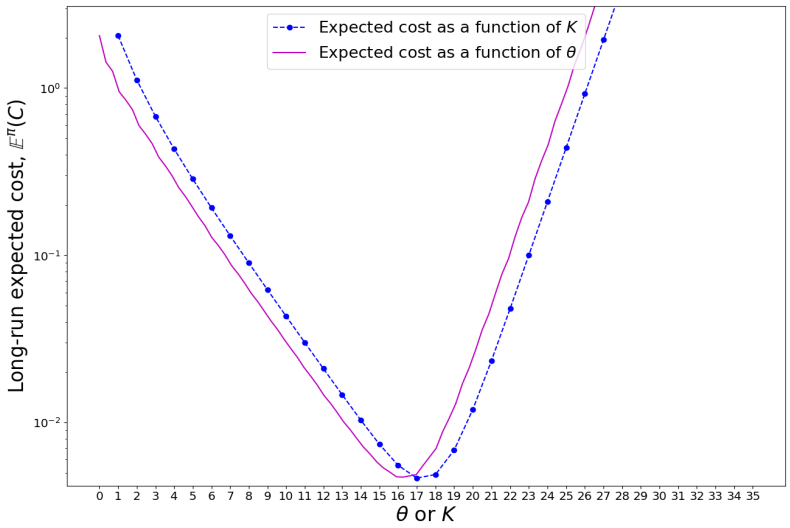


Figure 3: Curves showing the long-run expected cost (log scale) for the $M/M/1$ queue system when the blocking cost is defined as a function of the blocking size x, as: $5(1+3^{x-x_{\text{ref}}})$, with $x_{\text{ref}}=18$. The blue dotted curve is given by (11), evaluated at integer K values and represents the case of the threshold-type deterministic policy $\overline{\pi}$ with rejection at $x=K$, as a function of integer-valued K. The violet solid curve is given by (14) on an equally-spaced grid of real θ values and represents the case of the parameterised stochastic piecewise-linear policy π_θ defined in (12). The minimizer of the blue curve is $K^*=17$, a value that is very close to x_{ref}. The almost constant horizontal separation between the two curves is due to the relationship between K and θ as $K=\lceil\theta\rceil+1$. The queue load is $\rho=0.7$.

when estimating the blocking probability) but is now learned by the algorithm, it is not possible to choose a fixed value J for the size of the absorption set $\mathcal{A}$. Instead, we consider a fixed J/K fraction that adapts J to each value of K at the start of each learning step. In order to experiment with different sizes of the absorption set $\mathcal{A}$, we consider two different scenarios: $J=\lceil 0.3K\rceil$ and $J=\lceil 0.5K\rceil$, where $\lceil\cdot\rceil$ denotes the ceiling operator. Following the conclusions of the choice of J, N and T and their impact in the FV estimator accuracy outlined in Appendix D, given J and using the approximate relative error formulas provided therein, we adjust N and T at each learning step to obtain approximate expected relative errors of $\epsilon_\phi\sim 100\%$ and $\epsilon_{ET}\sim 20\%$, respectively for the estimations of $\phi_t^J(K-1)$ and $\mathbb{E}_{J-1}T_{\mathcal{A}}$; i.e. we set the approximate expected error for $\hat{\phi}_t^J(K-1)$ much larger than the approximate expected error for $\hat{\mathbb{E}}_{J-1}T_{\mathcal{A}}$. For comparison purposes and to further check the conclusions of the error analysis in Appendix D, we also considered a scenario where the pre-defined relative errors are inverted, namely $\epsilon_\phi\sim 20\%$ and $\epsilon_{ET}\sim 100\%$.

For each setup we ran the FVRL policy learner on 100 learning steps. At each learning step, using (13) and as long as a positive FVEE estimate of the stationary probability $p^{\pi_\theta}(K-1)$ has been obtained, we estimate the sparse $\nabla_\theta g_x(\pi_\theta)$ function defined in Section 3.3. To estimate the difference of the two Q functions in (13), we use the coupling procedure described in Lemma 3, allowing up to 250 arrival events in each of 100 replications. The value of $\hat{\nabla}_\theta g_x(\pi_\theta)$ is finally computed as the average Q-difference over these replications. Parameter θ is then updated by gradient descent using a constant learning rate of $\alpha=10$, bounding any negative value of θ to 0.1[4]. The estimated optimum threshold is set to $\hat{K}^*=\text{round}(\hat{\theta}^*)+1$, where $\hat{\theta}^*$ is the value of the θ parameter obtained at the end of the learning process[5].

The benchmark MC learning algorithm uses the same policy gradient approach as FVRL with the only difference

[4]The value of θ is not allowed to go negative (although it could theoretically be as small as -1) because it would lead to a degenerate value of the J parameter, namely $J=1$ which makes state $K-1$ be equal to $J-1$. Since $J-1$ is at the boundary of the absorption set $\mathcal{A}$, the stationary probability $p^{\pi_\theta}(K-1)$ would not be estimated by Fleming-Viot, thus precluding the estimation of the policy gradient. Finally, θ is lower bounded by 0.1 and not 0 in order to keep a real-valued θ.

[5]Note that the estimated optimum threshold is not set to $\lceil\hat{\theta}^*\rceil+1$, as in the definition of the parameterised policy, so that $\hat{K}^*$ is more naturally chosen to be e.g. 5 when $\hat{\theta}^*=4.01$, rather than 6.

that it estimates the probability $p^{\pi_\theta}(K-1)$ in expression (13) using Monte-Carlo instead of Fleming-Viot, i.e. based on a single trajectory of X_t^π as the ratio between the continuous time that the system spends at $K-1$ and the total return cycle time to $J-1$ (by renewal theory), where J is defined as a function of K as in FVRL. Each replication of the MC learner is started at $J-1$ and is stopped when the average number of observed events over all learning steps in the respective FVRL replication is observed. These two conditions allow a fair comparison between the two methods.

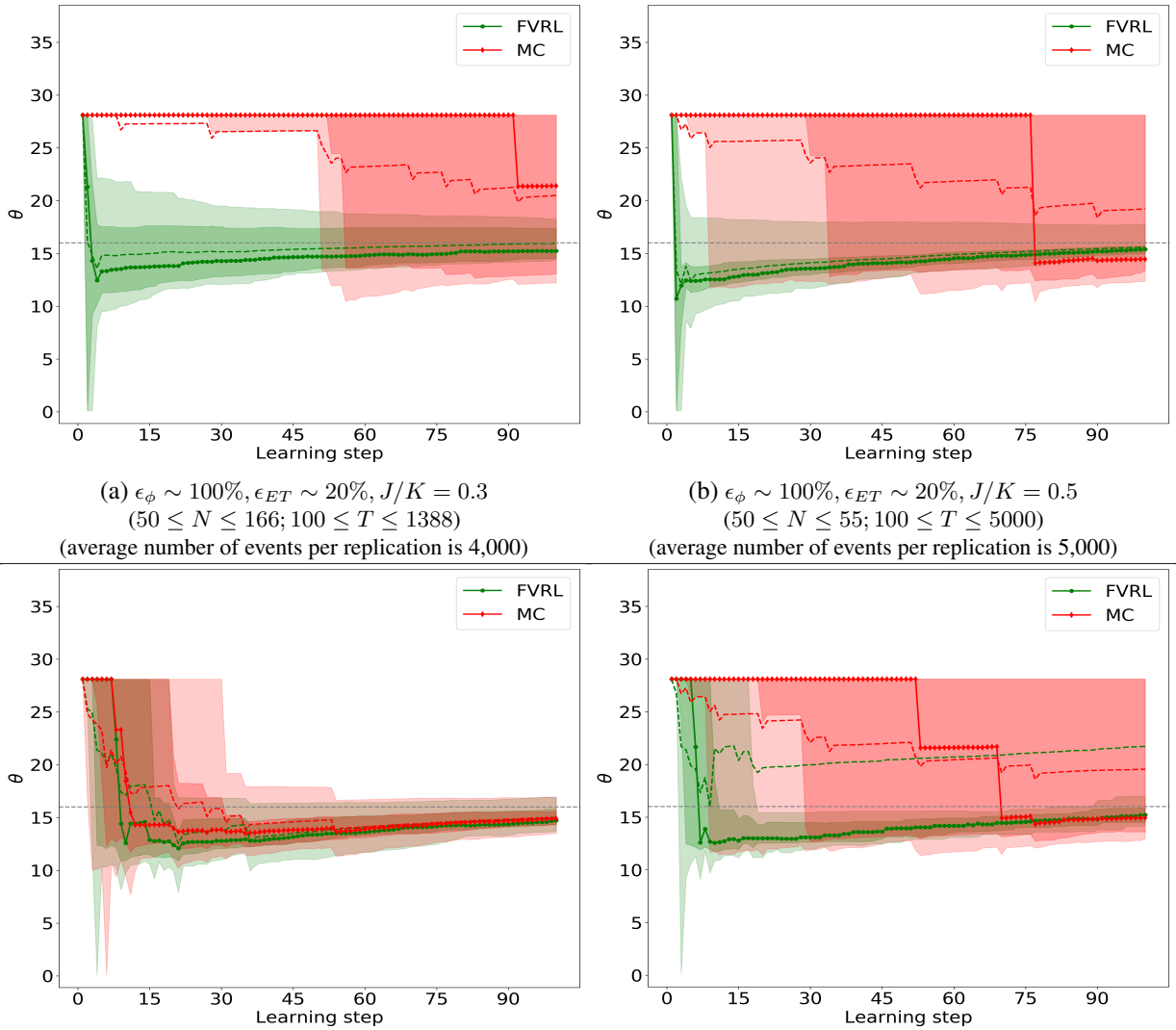
The results of the above procedure run on 20 replications are shown in Figure 4. We see that FVRL clearly outperforms the benchmark MC learning in three out of the four scenarios considered, depicted in Figures 4(a), 4(b) and 4(d). In the scenario depicted in Figure 4(c), learning by the two methods is similar because the average number of events is about 10 times larger than the one observed in the other three scenarios ($\sim 40,000$ vs. $\sim 5,000$), which allows MC to also observe the rare blocking state and thus learn almost as fast as FVRL. This is the same situation previously observed about the probability estimation results presented in Figure 2.

Of the two setups, $(\epsilon_\phi \sim 100\%, \epsilon_{ET} \sim 20\%)$ used for Figures 4(a) and 4(b), and $(\epsilon_\phi \sim 20\%, \epsilon_{ET} \sim 100\%)$ used for Figures 4(c) and 4(d), the former is the one that provides fastest learning (observe the steeper decrease of the curves in panels (a) and (b) compared to those in panels (c) and (d)), the smallest sample complexity, and the highest stability. This is consistent with the conclusions of the analysis presented in Appendix D of the choice of J, N and T in the FV estimation procedure, namely that, for an accurate estimation of the stationary probability, more importance should be given to achieve a smaller ϵ_{ET} error in the estimation of $\mathbb{E}_{J-1}T_{\mathcal{A}}$ than to achieve a smaller ϵ_ϕ error in the estimation of $\phi_t^J(K)$.

Regarding the two values considered for the absorption set size, $J = 0.3K$ and $J = 0.5K$, results are very similar although using $J = 0.3K$ yields learning curves that tend to be closer to the optimum θ^* parameter.

Finally, in terms of stability we note that, for the least convenient setup of $(\epsilon_\phi \sim 20\%, \epsilon_{ET} \sim 100\%)$ of Figure 4(d), in one replication the θ parameter suddenly increases from 28 to 140 (not shown but made apparent by the mean learning curve being significantly above the median learning curve). This overshoot impedes further learning because the blocking probability becomes extremely small at $K = 140$. This clearly illustrates the risks of allowing a too large error for the estimation of $\mathbb{E}_{J-1}(T_{\mathcal{A}})$, which may yield a blocking probability largely overestimated (due to an underestimation of $\mathbb{E}_{J-1}T_{\mathcal{A}}$) generating such out-of-control excursions of the θ parameter.

Summing up, the best FVRL setup in terms of learning speed, sample complexity, and stability is to use an intermediate value of the J/K factor, such as a value in the range $[0.3K, 0.5K]$. Once J is defined, a large enough value of T guaranteeing a small error in the estimation of $\mathbb{E}_{J-1}T_{\mathcal{A}}$ (say less than 50%) should be preferred over a large number of particles N controlling the estimation error of $\phi_t^J(K-1)$, which can be as large as 100%. Note that, although larger J values suggest larger T values, increasing T doesn't affect as much the sample complexity as increasing N does, as seen in the estimation problem in Section 4.1.1.



(a) $\epsilon_\phi \sim 100\%, \epsilon_{ET} \sim 20\%, J/K = 0.3$
$(50 \leq N \leq 166; 100 \leq T \leq 1388)$
(average number of events per replication is 4,000)

(b) $\epsilon_\phi \sim 100\%, \epsilon_{ET} \sim 20\%, J/K = 0.5$
$(50 \leq N \leq 55; 100 \leq T \leq 5000)$
(average number of events per replication is 5,000)

(c) $\epsilon_\phi \sim 20\%, \epsilon_{ET} \sim 100\%, J/K = 0.3$
$(50 \leq N \leq 500; 100 \leq T \leq 100)$
(average number of events per replication is 40,000)

(d) $\epsilon_\phi \sim 20\%, \epsilon_{ET} \sim 100\%, J/K = 0.5$
$(50 \leq N \leq 500; 100 \leq T \leq 5000)$
(average number of events per replication is 5,000)

Figure 4: Learning curves for 20 replications of FVRL (green, ours) vs. Monte-Carlo learning (red) of the optimum parameter $\theta^* = 16$ (horizontal dashed gray line) of the parameterised acceptance policy π_θ of incoming jobs to the $M/M/1$ queue system with $\lambda = \rho = 0.7$. For each learning method (color), the dashed line indicates the mean learning curve, the marked solid line the median learning curve, the dark and light shades indicate, respectively, the location of 50% and 80% of learning curves closer to the median. Each plot caption indicates the following learning settings for the FVRL approach which impact the estimations of the quantities in (10) at each learning step: the approximate expected relative error wished for the estimation of $\phi_t^J(K)$ (which defines N, the number of particles *at each learning step*), the approximate expected relative error wished for the estimation of $\mathbb{E}_{J-1}T_{\mathcal{A}}$ (which defines T, the number of arrival events for estimating $\mathbb{E}(T_{\mathcal{A}})$, *at each learning step*), and the fraction J/K defining the size of the absorption set. The range of N and T values used, as well as the average number of events over learning steps and across replications are also reported. In all cases the learning parameter is kept constant at $\alpha = 10$, and the initial θ guess is 28.1, which defines a "large" blocking size of $K = 30$ whose small stationary probability ($\sim 10^{-6}$) makes blocking a rare event already at the onset of learning.

4.2 Loss network system

Consider now a loss network that processes jobs of I different classes which, as in the $M/M/1$ system, arrive following independent Poisson processes and are served with independently exponentially distributed times by one of R available servers. We are interested in analyzing the number of jobs of each class being served by the system at a given time t, $\mathbf{X}_t^\pi = (X_{t,1}^\pi, ..., X_{t,I}^\pi)$, whose dynamics is governed by the system characteristics and the policy π being applied. As before, we also consider a policy π that belongs to the family of deterministic threshold-type admission control policies that reject an incoming job of a given class i when the system is already serving a predefined number of jobs K_i of that class, in which case a class-dependent rejection cost is accrued.

Let us denote by λ_i, μ_i, C_i, K_i the job arrival rate, the service rate, the rejection cost, and the blocking size of each job class $i = 1, \ldots, I$. Thus, the system can be represented by an MDP ($\mathcal{S} = \{\mathbf{X} \in \mathbb{N}^I : 0 \leq X_i \leq R, i = 1, \ldots, I \text{ s.t. } \mathbf{X}^T\mathbf{1} \leq R, \}, \mathcal{U} = \{0, 1\}, q = \{\lambda_1, \ldots, \lambda_I; \mu_1, \ldots, \mu_I\}, \mathcal{R} = r(\mathbf{x}, a) = \sum_{i=1}^I C_i \mathbf{1}_{\{\mathbf{x} \in \mathcal{S}_i, \mathcal{L}_i, a=0\}}, \overline{\Pi}_R)$, where $\mathcal{S}_i$ is the set of states where blocking occurs following an i-class job arrival denoted by $\mathcal{L}_i$, and $\overline{\Pi}_R$ is the class of threshold-type admission control policies for a system with R servers.

To facilitate illustration, we consider in our experiments the smallest multi-class loss network, namely one that serves just two classes of job.

4.2.1 Estimation of the expected cost using FVEE

Contrary to the $M/M/1$ case, estimating the expected cost in the loss network system is not equivalent to estimating the blocking probability, because blocking occurs at more than one state whose cost in general depends on the class of the rejected job.

The expression of the long-run expected cost $\mathbb{E}^\pi(C)$ (thereafter "expected cost") under the threshold policy $\pi \in \overline{\Pi}_R$ and costs $\mathcal{R}$ is obtained from the total probability theorem applied on all possible job arrival classes and all possible accept/reject actions on those arrivals, as:

$$\mathbb{E}^\pi(C) = \sum_{x \in \mathcal{S}} p^\pi(\mathbf{x}) c(\mathbf{x}), \tag{15}$$

where $p^\pi(\mathbf{x}) \doteq Pr(\mathbf{X}_n^\pi = \mathbf{x})$ is the stationary probability that the SMDP $\mathbf{X}_n^\pi$ (defined in Section 4.1.2) is at state $\mathbf{x}$, and $c(\mathbf{x}) = \sum_{i=1}^I C_i \lambda_i / \Lambda \left[\mathbf{1}_{\{\sum_{j=1}^I x_j = R\}} + \mathbf{1}_{\{\sum_{j=1}^I x_j < R\}} \mathbf{1}_{\{x_i = K_i\}} \right]$ is the expected cost of rejection at $\mathbf{x}$ over all possible arriving job classes, where $\Lambda \doteq \sum_{i=1}^I \lambda_i$ is the total job arrival rate, and the indicator terms reflect, respectively, that rejection can happen either when the network is operating at full capacity or when it is serving the maximum allowed number of jobs of the arriving class. To better isolate the impact of the estimation quality of the stationary probability $p^\pi(\mathbf{x})$ on the expected cost estimation, we use the true values of the job arrival rates λ_i to compute the expected cost $c(\mathbf{x})$, although in practice these values are normally unknown and would have to be estimated from the simulation. Finally, the blocking probability is given by the same expression (15) when setting $C_i = 1$ in $c(\mathbf{x})$, for all $i = 1, \ldots, I$.

A loss network with the following characteristics was considered for the estimation of the expected cost: capacity $R = 6$ servers, $\lambda = [1, 5]$, $\mu = [3.33, 50.0]$ (hence $\rho = [0.3, 0.1]$), $C = [2 \times 10^3, 2 \times 10^5]$, $K = [4, 6]$. The choice of the rejection costs allows us to illustrate the benefits of the FV approach over MC as it makes a few of the cost-generating states with smaller probability as important as those with larger probability in terms of their contribution to the system's expected cost. These contributions are listed in Table 1 where states are listed in decreasing order of their respective true stationary probabilities, computed following the product form of the stationary distribution of the stochastic knapsack described in [24, chapter 4]. We observe that the top three states in terms of probability contribute to about 72% of the expected cost, while the bottom four, with probability smaller than 10^{-6}, contribute to as much as 28%, therefore obtaining accurate estimates of the probability of a few of those smaller probability states is important for an accurate estimation of the expected cost.

State $\mathbf{x}$	**Prob.** $p^\pi(\mathbf{x})$	**Expected cost** $c(\mathbf{x})$	$p^\pi(\mathbf{x})c(\mathbf{x})$	**% Exp. cost** $\mathbb{E}^\pi(C)$
$[4,0]$	2.2×10^{-4}	0.333×10^3	0.1889	20.0%
$[4,1]$	2.3×10^{-5}	0.333×10^3	0.0075	2.0%
$[4,2]$	1.1×10^{-6}	167×10^3	0.0754	49.7%
$[3,3]$	5.0×10^{-7}	167×10^3	0.0839	22.0%
$[2,4]$	1.3×10^{-7}	167×10^3	0.0210	5.5%
$[1,5]$	1.7×10^{-8}	167×10^3	0.0028	0.74%
$[0,6]$	9.3×10^{-10}	167×10^3	0.0002	0.04%
Total			7.214	100.0%

Table 1: Contribution to the expected cost $\mathbb{E}^\pi(C)$ by each possible blocking state in the loss network system according to expression (15), with $R = 6$, $K = [4, 6]$, $\lambda = [1, 5]$, $\rho = [0.3, 0.1]$, $C = [2 \times 10^3, 2 \times 10^5]$, sorted by decreasing probability $p^\pi(\mathbf{x})$. The bottom four states with occurrence probability smaller than 10^{-6} contribute to $\sim 28\%$ of the expected cost.

The true blocking probability and true expected cost are computed from (15), against which the estimations by the analyzed methods are compared. The quality of their estimations is visualized using the violin plots presented in Figure 5 obtained from 10 replications of the estimation process. For each panel, the green violin plots on the left column of the figure analyze the convergence of FVEE, the FV expectation estimator of the quantity indicated on the vertical axis, in terms of increasing T [6] while keeping the number of particles N fixed at 400. The absorption set $\mathcal{A}$ is chosen as all the states whose server occupation is strictly smaller than 2 and 3, respectively for job classes 1 and 2, i.e. $\mathcal{A} = \{[0,0], [0,1], [0,2], [1,0], [1,1], [1,2]\}$. On the other hand, the red violin plots in each panel on the right column of Figure 5 present the results of the benchmark estimator, which is based on an MC simulation of the loss network. In order to make a fair comparison, the start state for MC is chosen uniformly at random in $\vec{\partial}\mathcal{A} = \{[0,2], [1,2], [1,0], [1,1]\}$ –so that both FVEE and MC methods start at a similar "distance" from the set of blocking states– and the MC simulation is let run until the number of events observed in the FVEE simulation is reached, whose average over replications is indicated at the top horizontal axes of the plots.

From Figure 5 we observe that both FVEE and MC successfully estimate the blocking probability and the expected cost in terms of obtaining a distribution of estimates that include their respective true values. However, FVEE shows a consistently smaller error than MC –as indicated by the median values in each experiment pair– as well as a consistently smaller variability.

4.2.2 Learning blocking sizes with FVRL

The FVRL algorithm for learning the blocking sizes $\mathbf{K}^* \in \mathbb{N}^I$ of a loss network serving I jobs classes learns the values of θ_i of I parameterised acceptance policies, each of the form (12). For each θ_i observed during learning, the deterministic blocking size K_i of the i-th policy is defined as $K_i = \lceil \theta_i \rceil + 1$, but, as in the $M/M/1$ case, the estimated blocking size obtained at the end of the learning process is defined as $\hat{K}_i^* = \text{round}(\hat{\theta}_i^*) + 1$.

Upon arrival of a job of class i, the respective π_{θ_i} acceptance policy is applied, making the system's acceptance policy equal to $\pi_\theta(a = 1|\mathbf{x}) = \sum_{i=1}^I \pi_{\theta_i}(a = 1|x_i)\mathbf{1}_{\mathcal{L}_i}$, where $\mathcal{L}_i$ denotes the arrival event of a class-i job. Then, the policy derivative w.r.t. θ_i is non-zero only at states $\mathbf{x}$ for which $x_i = K_i - 1$ as long as they satisfy the R-server constraint for a possible job acceptance, $\mathbf{x}^T\mathbf{1} < R$. At those states the derivative is equal to $+1$ for action $a = 1$ (accept) and -1 for action $a = 0$ (reject).

[6] We chose the number T of arrival events for estimating $\mathbb{E}(T_\mathcal{A})$ for the convergence analysis –instead of choosing the number of particles N– based on the conclusion obtained in Appendix D about the higher impact of T over N in the estimation accuracy of the stationary probability in the $M/M/1/K$ queue system.

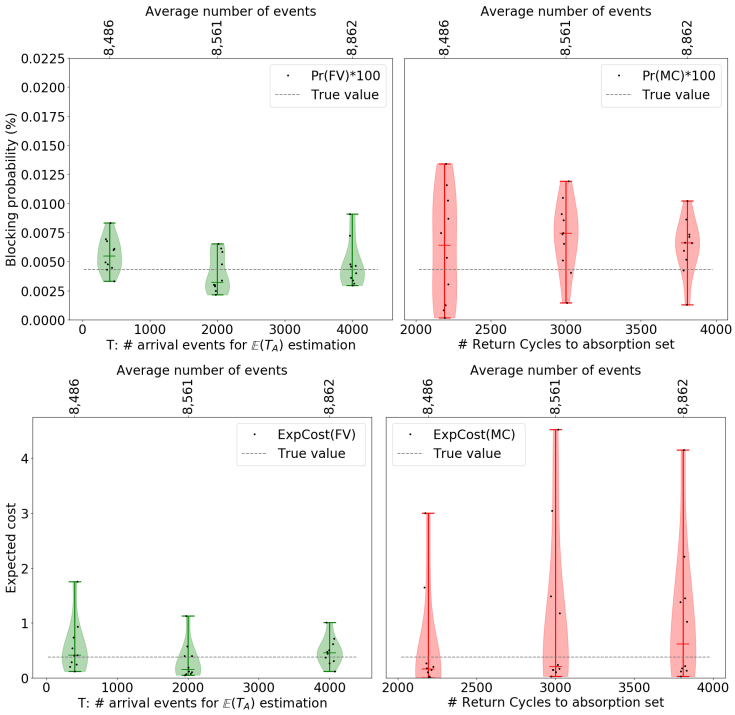


Figure 5: Violin plots showing the convergence properties of the Fleming-Viot (left, green) (ours) and Monte-Carlo (right, red) estimators of the blocking probability (top) and of the expected cost (bottom) in a two-job-class loss network with $R = 6$ servers, $\lambda = [1, 5]$, $\rho = [0.3, 0.1]$, blocking at $K = [4, 6]$ with costs $C = [2 \times 10^3, 2 \times 10^5]$, as the number T of arrival events for estimating $\mathbb{E}(T_{\mathcal{A}})$ increases. The absorption set sizes by job class are $J = [2, 3]$ and the number of particles is $N = 400$. The horizontal tick inside each violin plot is the median estimated value over 10 replications. In the corresponding MC experiments, the average number of return cycles to the absorption set is used on the horizontal axis, which is a measure of sample size. Finally, the top horizontal axes show the average number of events observed in the experiments run in each set, which by design coincide between each paired FV-MC execution, as mentioned in the text.

Hence, the partial derivative of $g(\pi_\theta)$ w.r.t. θ_i in (13) becomes:

$$\frac{\partial g(\pi_\theta)}{\partial \theta_i} = \sum_{\mathbf{x} \in \mathcal{S}_i} p^{\pi_\theta}(\mathbf{x}) \left[Q^{\pi_\theta}(\mathbf{x}, 1) - Q^{\pi_\theta}(\mathbf{x}, 0) \right], \text{ for } i = 1, \ldots, I, \tag{16}$$

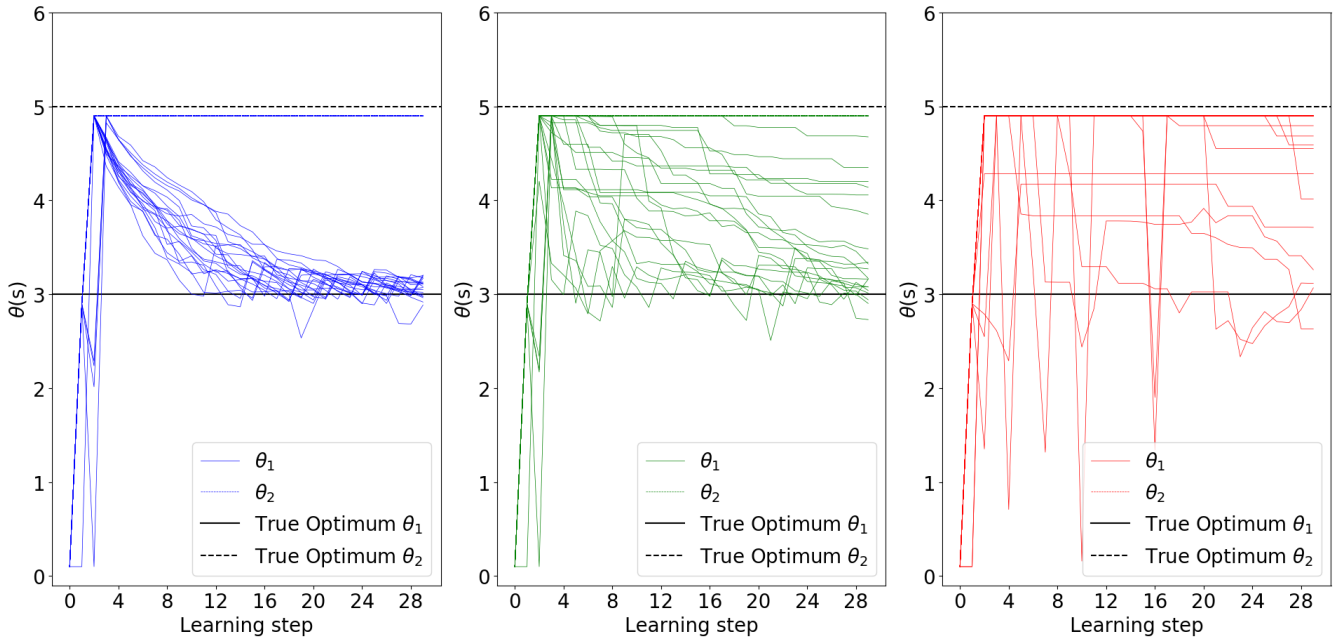
where $\mathcal{S}_i$ is the set of states $\mathbf{x}$ where $x_i = K_i - 1$ and $\mathbf{x}^T \mathbf{1} < R$.

To illustrate the FVRL algorithm, we consider the same loss network used for the estimation problem (see previous Section 4.2.1). For this choice of parameters, the optimum blocking sizes of the threshold policy are equal to $K^* = [4, 6]$ (i.e. $\theta^* = [3, 5]$), found by evaluating the true expected cost on all possible threshold combinations. Note that λ_i and μ_i do not need to be known to apply the algorithm.

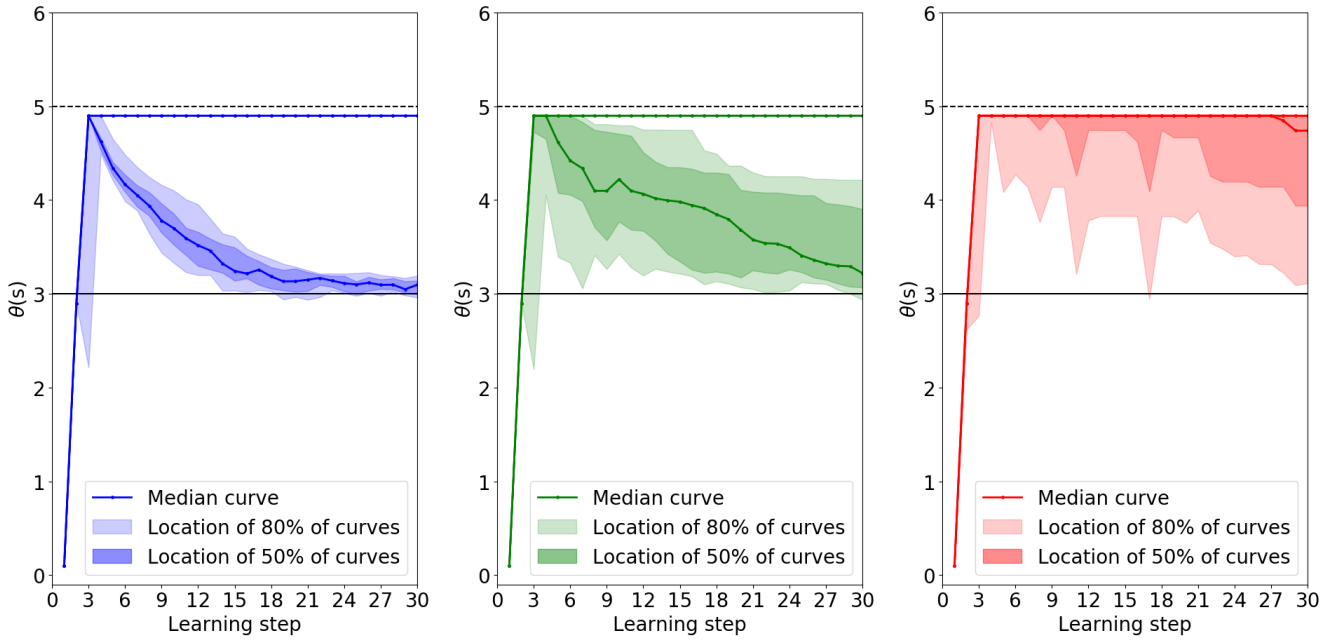
The setup of the learning experiments is as follows: we set the initial blocking size guess to $K = [0.1, 0.1]$, and the following simulation parameters for the estimation of the stationary probabilities $p^{\pi_\theta}(\mathbf{x})$ for each $\mathbf{x} \in \mathcal{S}_i$: $J/K = 0.5$ for both classes, the number of FV particles $N = 100$ (four times smaller than the value of N used in the estimation problem), and the number of arrival events for estimating $\mathbb{E}(T_{\mathcal{A}})$, $T = 500$ (very close to the smallest T value considered in the estimation problem). The start state of each FV particle is chosen following the estimated entrance distribution of the states in $\vec{\partial}\mathcal{A}^c$. The estimation of the gradient in (16) is completed by estimating the Q-difference at each $\mathbf{x}$ following the same coupling setup of Lemma 3 used for the $M/M/1$ case. The learning rate α is set initially to 1.0 and then decreased inversely proportional to the learning step. Finally, at each learning step, the estimated θ_i values are bounded to the interval $[0.1, K_i - 1 - 0.1]$.

The results of the above procedure, run on 20 replications, are shown in Figure 6 and described in the caption. Besides showing the FVRL and MC learning curves, the figure includes a reference learning path depicted in blue on the left, where the learning process uses the *true* stationary probabilities (presented in Table 1) in the gradient expression (16), so that only the Q-differences need to be estimated for the computation of the gradient. The aim is to have a reference for comparison that corresponds to the most ideal available learning scenario.

We observe that FVRL clearly outperforms MC learning as, in median, parameter θ_1 is correctly learned by FVRL but not by MC, where it practically remains stuck near 5. FVRL also presents smaller variability in the learning paths than MC. As seen in the center panel in Figure 6(a)), however, 30 learning steps were not enough for *all* 20 replications to reach the optimal θ_1 value, although most of them started learning.



(a) Learning paths of 20 replications



(b) Distribution of learning paths of 20 replications

Figure 6: Two comparisons of the FVRL method with Monte-Carlo learning on 20 replications. The top row shows each observed learning trajectory and the bottom row shows their distribution. On each row, the left panel represents the scenario closest to the true learning process as it corresponds to learning using the true stationary probabilities, where only the difference in the Q values is estimated. The center panel corresponds to FVRL and the right panel to Monte-Carlo learning. The plots show the results over 20 experiments run over 30 learning steps on a loss network with $R = 6$ servers, $\lambda = [1, 5]$, $\rho = [0.3, 0.1]$, with blocking costs $C = [2 \times 10^3, 2 \times 10^5]$. At each learning step, the FVRL setup sets the absorption set size by job class at $J_i/K_i = [0.5, 0.5]$, the number of particles at $N = 100$ and the number of arrival steps for estimating $\mathbb{E}(T_{\mathcal{A}})$ at $T = 500$. In the MC learning case, each experiment uses as many number of events per learning step as the average number of events observed in FVRL by learning step. In all cases the initial guess is $\theta = [0.1, 0.1]$ ($K = [2, 2]$), the learning parameter starts at $\alpha = 1.0$ and is then decreased inversely proportional to the learning step. The top row shows each of the 20 realized paths. The bottom row shows the distribution of the visited θ values by learning step, where the darker band includes 50% of the paths around their median and the lighter band includes 80% of the paths. The interesting learning paths to compare are those associated to θ_1, as θ_2 is learned equally well as 4.9 by all methods (which is shown as dotted lines in (a)).

5 Conclusion

We presented the Fleming-Viot particle system (FV) as an alternative to Monte-Carlo for a more efficient exploration of reinforcement learning environments by leveraging prior knowledge about a set $\mathcal{A}$ of frequently visited states with no rewards. The method helps to accelerate the discovery of rarely observed states that may yield large rewards affecting optimisation objectives, thus speeding up policy learning. It is proposed particularly for situations that preclude the use of offline exploration techniques, such as importance sampling, and to avoid the need for expert knowledge to guide learning.

The use of FV is two-fold: to estimate a long-run optimisation objective, and to learn a policy for control problems. Its use was illustrated on the design of threshold-type admission control policies to minimize the long-run expectation cost in two stochastic network environments: an $M/M/1$ queue system and a loss network accepting two classes of job.

For the estimation problem, the FV estimation approach proved to be much more efficient than Monte-Carlo in the $M/M/1$ system for large capacities K, where Monte-Carlo completely fails using the same event budget. In the loss network system, FV was able to estimate the expected cost more accurately and precisely than Monte-Carlo.

For the control problem, the presented FVRL method –which uses FV to estimate sparse gradients of a policy gradient algorithm– is able to find the optimal thresholds of admission control policies considerably faster than Monte-Carlo, in both the $M/M/1$ and the loss network cases. We note that in the control problem the estimation accuracy of the gradient is not as crucial as in the estimation problem, because the algorithm is able to learn as long as it receives a signal from the rare states. We also note that the FVRL algorithm appears to converge to a global optimum in these problems, even though the theoretical convergence properties of Proposition 4 only guarantee reaching a local optimum, and even if not all convergence requirements are satisfied (e.g. increasing number of cycles M and number of particles N).

The presented experimental results showed the validity of the FV methodology in rather small problems, namely on finite state and action spaces whose cardinality does not go over a couple of hundreds. Proving its feasibility on larger environments is left for future work.

We finally note that the presented FV methodology is applicable insofar as a "black-box" simulator or emulator of the system under study is available, which must be able to provide the next state and reward observed when the agent takes any possible action at any possible system's state. This requirement also means that FV is currently not applicable to online or buffer replay contexts, i.e. when trajectories are collected from an ongoing or previously observed agent-environment interaction.

In future work we will generalize the algorithms so that they include a step where the set $\mathcal{A}$ of frequently visited states is defined from an initial exploration of the environment, as well as allow for the possibility of non-zero rewards received in $\mathcal{A}$.

We also intend to extend the algorithms to more classical RL environments, such as labyrinths, games, mountain car, etc., which are episodic in nature, and to optimisation problems with discount.

References

[1] Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. Journal of Machine Learning Research, 22(98):1–76, 2021.

[2] S. Asmussen. Applied Probability and Queues. Applications of mathematics : stochastic modelling and applied probability. Springer, 2003.

[3] A. Asselah, P.A. Ferrari, and P. Groisman. Quasistationary distributions and Fleming-Viot processes in finite spaces. J. Appl. Probab., 48(2):322–332, 2011.

[4] Dimitris Bertsimas and Cheol Woo Kim. Optimal control of multiclass fluid queueing networks: A machine learning approach. CoRR, abs/2307.12405, 2023.

[5] Pierre Brémaud. Markov chains: Gibbs fields, Monte Carlo simulation, and queues, volume 31. Springer Science & Business Media, 2013.

[6] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. arXiv preprint arXiv:1808.04355, 2018.

[7] K. Burdzy, R. Holyst, and P. March. A Fleming-Viot particle representation of the Dirichlet Laplacian. Comm. Math. Phys., 214(3):679–703, 2000.

[8] Bertrand Cloez and Josué Corujo. Uniform in time propagation of chaos for a moran model. arXiv preprint arXiv:2107.10794, 2021.

[9] Bertrand Cloez and Marie-Noémie Thai. Quantitative results for the fleming–viot particle system and quasi-stationary distributions in discrete space. Stochastic Processes and their Applications, 126(3):680–702, 2016.

[10] J. N. Darroch and E. Seneta. On quasi-stationary distributions in absorbing continuous-time finite Markov chains. J. Appl. Probability, 4:192–196, 1967.

[11] Ernesto Garcia, Paola Bermolen, Matthieu Jonckheere, and Seva Shneer. Probabilistic insights for efficient exploration strategies in reinforcement learning, 2025.

[12] P. Groisman and M. Jonckheere. Simulation of quasi-stationary distributions on countable spaces. Markov Process. Related Fields, 19(3):521–542, 2013.

[13] Isaac Grosof, Siva Theja Maguluri, and R Srikant. Convergence for natural policy gradient on infinite-state queueing mdps. arXiv preprint arXiv:2402.05274, 2024.

[14] G. Yin H. J. Kushner. Stochastic Approximation and Recursive Algorithms and Applications. Springer-Verlag, 2003.

[15] Mark Gluzman J. G. Dai. Queueing network controls via deep reinforcement learning. Stochastic Systems, 12(1):30–67, 2022.

[16] Chi Jin, Akshay Krishnamurthy, Max Simchowitz, and Tiancheng Yu. Reward-free exploration for reinforcement learning. In Hal Daumé III and Aarti Singh, editors, Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 4870–4879. PMLR, 13–18 Jul 2020.

[17] Antonio Massaro, Francesco De Pellegrini, and Lorenzo Maggi. Optimal trunk-reservation by policy learning. In IEEE INFOCOM 2019, apr 2019.

[18] Maja J Mataric. Reward functions for accelerated learning. In Machine learning proceedings 1994, pages 181–189. Elsevier, 1994.

[19] Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 24. Curran Associates, Inc., 2011.

[20] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 2778–2787. PMLR, 06–11 Aug 2017.

[21] Brahma S. Pavse, Matthew Zurek, Yudong Chen, Qiaomin Xie, and Josiah P. Hanna. Learning to stabilize online reinforcement learning in unbounded state spaces. In Proceedings of the 41st International Conference on Machine Learning, ICML'24. JMLR.org, 2024.

[22] M. L. Puterman. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, 1994.

[23] Herbert E. Robbins. A stochastic approximation method. Annals of Mathematical Statistics, 22:400–407, 1951.

[24] Keith W. Ross. Multiservice Loss Models for Broadband Telecommunication Networks. Springer, 1995.

[25] Richard Sutton and Andrew Barto. Reinforcement Learning, an introduction. MIT Press, 2018.

[26] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In Advances in neural information processing systems, pages 1057–1063, 2000.

[27] Andrea Tirinzoni, Andrea Sessa, Matteo Pirotta, and Marcello Restelli. Importance weighted transfer of samples in reinforcement learning. In International Conference on Machine Learning, pages 4936–4945. PMLR, 2018.

[28] Martin J Wainwright. High-dimensional statistics: A non-asymptotic viewpoint, volume 48. Cambridge University Press, 2019.

A Estimation of the stationary probability distribution $p^\pi(x)$

using Fleming-Viot particle systems Based on expression (3) of Proposition 1, the FV estimator of the stationary probability $p^\pi(x)$ for states x outside the absorption set $\mathcal{A}$ is constructed as follows: we first define functions $f(x,t)$ and $g(t)$ as:

$$\begin{aligned} f(x,t) &\doteq \phi_t^{\vec{\partial}\mathcal{A}^c}(x), \\ g(t) &\doteq \frac{\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_\mathcal{K} > t)}{\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}}, \end{aligned}$$

where $\phi_t^{\vec{\partial}\mathcal{A}^c}(x)$ is defined in Proposition 1 and the other concepts are defined in the Definitions Section 3.1. These functions, together with their estimators $\hat{f}$ and $\hat{g}$, will be instrumental in Appendix B in proving the consistency of the FV estimator of the stationary probability provided by Theorem 2.

Using the estimators $\hat{f}(x,t)$ and $\hat{g}(t)$ described below, the FV estimator of the stationary probability is computed as:

$$\hat{p}^\pi_{FV}(x) = \int_0^\infty \hat{f}(x,t)\hat{g}(t)dt, \qquad \forall x \in \mathcal{A}^c. \tag{17}$$

Firstly, we explain the construction of the estimator $\hat{g}(t)$, which is a ratio of the quantities $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_\mathcal{K} > t)$ and $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}$, each of which is estimated from two separate simulations, as follows: (i) a simulation of the Markov process starting at an arbitrary state $x \in \vec{\partial}\mathcal{A}$, and (ii) a simulation of a predefined number of N independent copies of the Markov process starting at a randomly selected state $x \in \vec{\partial}\mathcal{A}^c$ following the entrance state distribution into $\mathcal{A}^c$ under stationarity, estimated from the first simulation. As explained below, simulations (i) and (ii) contribute to the estimation of $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}$ while simulation (ii) contributes to the estimation of $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_\mathcal{K} > t)$.

The details are as follows: we set $\tau_{\mathcal{A},0} \doteq 0$ and define a sequence of stopping times $\tau_{\mathcal{A}^c,i}$, $\tau_{\mathcal{A},i}$, associated to the events of entry, in the i-th cycle, into $\mathcal{A}^c$ and into $\mathcal{A}$, respectively, as follows:

$$\begin{aligned} \tau_{\mathcal{A}^c,i} &= \inf_{t > \tau_{\mathcal{A},i-1}} \{X_t^\pi \in \mathcal{A}^c\}, \\ \tau_{\mathcal{A},i} &= \inf_{t > \tau_{\mathcal{A}^c,i}} \{X_t^\pi \in \mathcal{A}\}, \end{aligned}$$

for $i \geq 1$. We also define $T_{E,i} = \tau_{\mathcal{A}^c,i} - \tau_{\mathcal{A},i-1}$, the entry time into $\mathcal{A}^c$ within cycle i, and $T_{\mathcal{K},i} = \tau_{\mathcal{A},i} - \tau_{\mathcal{A}^c,i}$, the killing time within cycle i. We note that $T_{\mathcal{A},i} = T_{E,i} + T_{\mathcal{K},i}$, which will be used below to construct the estimator of the denominator of g, $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_\mathcal{A}$.

The first simulation consists of running the process X_t^π until a predefined number $M_0 + M$ entry times $\{T_{E,i}\}_{i=1}^{M_0+M}$ are observed, where we consider the first M_0 observations to be burn-in in order to assume stationarity of the process thereafter. From this simulation we compute the empirical entrance state distribution into $\mathcal{A}^c$, which can be considered as an estimator of $p^\pi_{\vec{\partial}\mathcal{A}^c}$, as $\hat{p}^\pi_{\vec{\partial}\mathcal{A}^c} = \frac{1}{M}\sum_{i=M_0+1}^{M_0+M} \mathbf{1}_{X^\pi_{\tau_{\mathcal{A}^c,i}}}$.

The second simulation of the N independent copies of X_t^π is started at states $x_i \in \vec{\partial}\mathcal{A}^c$ randomly chosen according to the estimated stationary distribution $\hat{p}^\pi_{\vec{\partial}\mathcal{A}^c}$ (this is trivial when $\vec{\partial}\mathcal{A}^c$ has a single state), yielding N killing times $\{T_{\mathcal{K},i}\}_{i=1}^N$.

From the last M entry times and the N killing times, we define a Monte-Carlo estimator of g as the ratio

$$\hat{g}(t) = \frac{\frac{1}{N}\sum_{i=1}^N \mathbf{1}_{T_{\mathcal{K},i} > t}}{\frac{1}{M}\sum_{i=M_0+1}^{M_0+M} T_{E,i} + \frac{1}{N}\sum_{i=1}^N T_{\mathcal{K},i}}. \tag{18}$$

Next, we explain the construction of the estimator $\hat{f}(x,t)$, which is the conditional occupation probability $\phi_t^{\vec{\partial}\mathcal{A}^c}(x)$, defined in Proposition 1, at each $x \in \mathcal{A}^c$ for all times t, using the Fleming-Viot N-particle system driven by X_t^π. The

Fleming-Viot system, denoted by $(\xi_t^\nu)_{t\geq 0}$, is simulated as described in Section 3.2, using $\mathcal{A}$ as the absorption set and starting each particle i at x_i a randomly chosen state according to the distribution $\nu(y) = p^\pi_{\vec{\partial}\mathcal{A}^c}(y)\mathbf{1}_{y\in\vec{\partial}\mathcal{A}^c}$. That is, all particles start at the boundary of $\mathcal{A}^c$, as required by the quantity to estimate, $\phi_t^{\vec{\partial}\mathcal{A}^c}(x)$. We let $m(\cdot,\xi) : \mathcal{A}^c \to [0,1]$ denote the empirical distribution of the N particles with positions described by vector ξ, defined as the empirical mean $m(x,\xi) \doteq \frac{1}{N}\sum_{i=1}^N \mathbf{1}_{\xi(i)=x}, \forall x \in \mathcal{A}^c$. Since $m(x,\xi_t^\nu)$ is an estimator of $\phi_t^{\vec{\partial}\mathcal{A}^c}(x)$ (because ν is restricted to the boundary of $\mathcal{A}^c$ where it is equal to the entrance state distribution into $\mathcal{A}^c$ under stationarity), f can be estimated by

$$\hat{f}(x,t) = m(x,\xi_t^\nu). \tag{19}$$

We note that, by construction, $\hat{g}(t) = 0$ for $t > T_{\mathcal{K},\max} = \max\{T_{\mathcal{K},i} : 1 \leq i \leq N\}$. Therefore, since we wish to compute $\hat{p}^\pi(x) = \int_0^\infty \hat{f}(x,t)\hat{g}(t)dt$, we only need to simulate the Fleming-Viot process until time $T_{\mathcal{K},\max}$ is reached. Also, since both $\hat{f}(x,t)$ and $\hat{g}(t)$ are almost surely piecewise constant functions and $\hat{g}(t) = 0$ for $t > T_{\mathcal{K},\max}$, the integral $\int_0^\infty \hat{f}(x,t)\hat{g}(t)dt$ is a finite sum that can be easily computed.

B Proof of Theorem 2: error of the Fleming-Viot estimator of the stationary probability, $\hat{p}^\pi_{FV}$

Using the notation f, g introduced in Appendix A and estimators $\hat{f}, \hat{g}$ given in (18) and (19) respectively, we have $p^\pi(x) = \int_0^\infty f(x,t)g(t)dt$ and $\hat{p}^\pi_{FV}(x) = \int_0^\infty \hat{f}(x,t)\hat{g}(t)dt$, for states x outside the absorption set $\mathcal{A}$. We are thus interested in bounding:

$$\mathbb{E}\left|\int_0^\infty \hat{f}\hat{g}dt - \int_0^\infty fgdt\right|$$

We start by decomposing the problem of upper bounding the above quantity into two subproblems in the following way:

$$\mathbb{E}\left|\int_0^\infty \hat{f}\hat{g}dt - \int_0^\infty fgdt\right| \leq \mathbb{E}\left|\int_0^\infty (f-\hat{f})gdt\right| + \mathbb{E}\left|\int_0^\infty \hat{f}(\hat{g}-g)dt\right| \tag{20}$$

We start by bounding the first term on the right hand side. For this purpose we will need the uniform propagation of chaos bound presented in (5) which bounds the estimation error of $m(\cdot,\xi_t^\nu)$ defined in Appendix A, that is:

$$\sup_{\|\phi\|_\infty\leq 1}\sup_{t\geq 0}\mathbb{E}\Big|[m(\cdot,\xi_t^\nu)(\phi)] - \phi_t^\nu(\phi)\Big| \leq \frac{C_{\text{FV}}}{\sqrt{N}},$$

for some $C_{FV} > 0$. It follows that:

$$\sup_{t\geq 0}\mathbb{E}\Big|\hat{f}(x,t) - f(x,t)\Big| \leq \frac{C_{\text{FV}}}{\sqrt{N}}, \qquad \forall x \in \mathcal{A}^c. \tag{21}$$

As was mentioned in subsection 3.2, this bound follows directly from [8, Theorem 1.4]. The assumptions of [8, Theorem 1.4] have a very general form, but it is easy to check that they are trivially satisfied in our simple case. The assumption (I) on initialization is satisfied by our assumption that the FV particle system is started at the position of N i.i.d samples from $p^\pi_{\vec{\partial}\mathcal{A}^c}$. Assumption (C1) has several parts: the uniform bound on selection rates (which are, in our case, the rates of jumps out of $\mathcal{A}^c$), follows from the fact that the state space is finite); the rest of assumption (C1) is trivially satisfied when we take $V_\mu^d(x)$ to be the rate of jump out of $\mathcal{A}^c$ from the state $x \in \mathcal{A}^c$ for any μ, and set function $V_\mu^b(y), V_\mu^s(x,y)$ equal to zero. Finally, assumption (C2) follows from the fact that we are working with an irreducible Markov chain on a finite state space. Therefore, using the triangle inequality and the inequality (21), we

obtain:

$$\begin{aligned}\mathbb{E}\left|\int_0^\infty (f-\hat{f})g dt\right| &\leq \mathbb{E}\int_0^\infty \left|f-\hat{f}\right| g dt \\ &\leq \int_0^\infty \mathbb{E}\left|f-\hat{f}\right| g dt \\ &\leq \frac{C_{\text{FV}}}{\sqrt{N}}\int_0^\infty g dt = \frac{\mathbb{E}_{\vec{\partial}\mathcal{A}^c}T_{\mathcal{K}}}{\mathbb{E}_{\vec{\partial}\mathcal{A}}T_{\mathcal{A}}}\frac{C_{\text{FV}}}{\sqrt{N}},\end{aligned}$$

where in the last line we also used the 'wedding cake decomposition', $\int_0^\infty \mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_{\mathcal{K}} > t)dt = \mathbb{E}_{\vec{\partial}\mathcal{A}^c}T_{\mathcal{K}}$.
Since $\hat{f}(x,t)$ is a probability, we get:

$$\left|\int_0^\infty \hat{f}(\hat{g}-g)dt\right| \leq \int_0^\infty |\hat{g}-g|\, dt.$$

We thus wish to bound $\mathbb{E}\int_0^\infty |\hat{g}-g|\, dt$. For convenience, we define a random variable $\overline{T}_{\mathcal{A}}$ with the distribution of $T_{\mathcal{A}}$ when X_t^π is started with distribution $p^\pi_{\vec{\partial}\mathcal{A}}$, and a random variable $\overline{T}_{\mathcal{K}}$ with the distribution of $T_{\mathcal{K}}$ when X_t^π is started with distribution $p^\pi_{\vec{\partial}\mathcal{A}^c}$. Since we start the simulation of X_t^π for the purpose of estimating g with distribution $p^\pi_{\vec{\partial}\mathcal{A}}$, we do not need any burn-in. We therefore take $M_0 = 0$. We also note that since we start the simulation at the distribution $p^\pi_{\vec{\partial}\mathcal{A}}$, it follows from renewal theory [2] that the inter-arrival times $\mathcal{T}_{\mathcal{A},i}$ used to construct the estimator $\hat{g}$ are i.i.d. with distribution $\overline{T}_{\mathcal{A}}$.

We also introduce additional shorthand notation for the numerators and denominators of $g(t)$ and $\hat{g}(t)$. We denote $N_t = \mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_{\mathcal{K}} > t)$ and $D_{\mathcal{A}} = \mathbb{E}_{\vec{\partial}\mathcal{A}}T_{\mathcal{A}}$, $D_{\mathcal{K}} = \mathbb{E}_{\vec{\partial}\mathcal{A}^c}T_{\mathcal{K}}$. We also denote by $\hat{N}_t, \hat{D}_{\mathcal{A}}$ the estimators of $N_t, D_{\mathcal{A}}$, that is $\hat{N}_t = \frac{1}{N}\sum_{i=1}^N \mathbf{1}_{T_{\mathcal{K},i}>t}$ and $\hat{D}_{\mathcal{A}} = \hat{D}_{\mathbb{E}} + \hat{D}_{\mathcal{K}}$, with

$$\hat{D}_{\mathbb{E}} = \frac{1}{M}\sum_{i=1}^M T_{E,i},$$

$$\hat{D}_{\mathcal{K}} = \frac{1}{N}\sum_{i=1}^N T_{\mathcal{K},i},$$

We thus have $g(t) = \frac{N_t}{D}$ and $\hat{g}_t = \frac{\hat{N}_t}{\hat{D}_{\mathcal{A}}}$.
We are interested in bounding:

$$\mathbb{E}\int_0^\infty \left|\frac{\hat{N}_t}{\hat{D}_{\mathcal{A}}} - \frac{N_t}{D_{\mathcal{A}}}\right|$$

Using the triangle inequality $\left|\frac{\hat{N}_t}{\hat{D}_{\mathcal{A}}} - \frac{N_t}{D_{\mathcal{A}}}\right| \leq \left|\frac{\hat{N}_t}{\hat{D}_{\mathcal{A}}} - \frac{\hat{N}_t}{D_{\mathcal{A}}}\right| + \left|\frac{\hat{N}_t}{D_{\mathcal{A}}} - \frac{N_t}{D_{\mathcal{A}}}\right|$ we get:

$$\mathbb{E}\int_0^\infty \left|\frac{\hat{N}_t}{\hat{D}_{\mathcal{A}}} - \frac{N_t}{D_{\mathcal{A}}}\right| dt \leq \mathbb{E}\int_0^\infty \hat{N}_t \left|\frac{1}{\hat{D}_{\mathcal{A}}} - \frac{1}{D_{\mathcal{A}}}\right| dt + \mathbb{E}\int_0^\infty \frac{1}{D_{\mathcal{A}}}\left|\hat{N}_t - N_t\right| dt,$$

Using the formula $\int_0^\infty \hat{N}_t dt = \frac{1}{N}\sum_{i=1}^N T_{\mathcal{K},i} = \hat{D}_{\mathcal{K}}$, the first term on the right hand side of the above bound is equal to $\mathbb{E}\left|\frac{\hat{D}_{\mathcal{K}}}{\hat{D}_{\mathcal{A}}} - \frac{\hat{D}_{\mathcal{K}}}{D_{\mathcal{A}}}\right|$. To bound this quantity, we introduce an event $B = \{\hat{D}_{\mathcal{A}} < \frac{1}{2}D_{\mathcal{A}}\}$. We use the decomposition

$$\mathbb{E}\left|\frac{\hat{D}_{\mathcal{K}}}{\hat{D}_{\mathcal{A}}} - \frac{\hat{D}_{\mathcal{K}}}{D_{\mathcal{A}}}\right| = \mathbb{E}\mathbf{1}_B\left|\frac{\hat{D}_{\mathcal{K}}}{\hat{D}_{\mathcal{A}}} - \frac{\hat{D}_{\mathcal{K}}}{D_{\mathcal{A}}}\right| + \mathbb{E}\mathbf{1}_{B^c}\left|\frac{\hat{D}_{\mathcal{K}}}{\hat{D}_{\mathcal{A}}} - \frac{\hat{D}_{\mathcal{K}}}{D_{\mathcal{A}}}\right| \tag{22}$$

and bound each of the terms separately.

Since we always have $0 \leq \frac{\hat{D}_{\mathcal{K}}}{\hat{D}_{\mathcal{A}}} \leq 1$ and on the set B we have $0 \leq \frac{\hat{D}_{\mathcal{K}}}{D\mathcal{A}} \leq \frac{1}{2}$, we have:

$$\mathbb{E}\mathbf{1}_B \left| \frac{\hat{D}_{\mathcal{K}}}{\hat{D}_{\mathcal{A}}} - \frac{\hat{D}_{\mathcal{K}}}{D_{\mathcal{A}}} \right| \leq \mathbb{P}(B).$$

Since the Markov Process X_t^π is irreducible and the state space $\mathcal{S}$ is finite, it is geometrically ergodic [5]. It follows then that there exist constants $C, \lambda > 0$, such that $\mathbb{P}(\overline{T}_{\mathcal{A}} > t) \leq C \exp(-\lambda t)$. Therefore, by [28, Theorem 2.13], the random variable $\overline{T}_{\mathcal{A}}$ is subexponential. Furthermore $\mathbb{P}(B) \leq \mathbb{P}_{\vec{\partial}\mathcal{A}^c}\left(\left| \hat{D}_{\mathcal{A}} - D_{\mathcal{A}} \right| \geq \frac{1}{2} D_{\mathcal{A}} \right)$. From the concentration bound for the standard estimator of the mean of subexponential variables [28][Equation 2.18], it follows that there exists $c > 0$:

$$\mathbb{P}(B) \leq e^{-c\sqrt{1/N + 1/M}}.$$

To bound the second term in (22), we observe that the function $h(x) = 1/x$ is Lipschitz continuous on $[a, \infty)$ for any $a > 0$, with the Lipschitz constant $L_a = \sup_{x \in [a,\infty)} |h'(x)| = \frac{1}{a^2}$. Using this fact with $a = D_{\mathcal{A}}/2$, we have:

$$\mathbb{E}\mathbf{1}_{B^c} \hat{D}_{\mathcal{K}} \left| \frac{1}{\hat{D}_{\mathcal{A}}} - \frac{1}{D_{\mathcal{A}}} \right| \leq \frac{4}{D_{\mathcal{A}}^2} \mathbb{E}\hat{D}_{\mathcal{K}} \left| \hat{D}_{\mathcal{A}} - D_{\mathcal{A}} \right|.$$

Using the Cauchy-Schwartz inequality, we get:

$$\begin{aligned}
\mathbb{E}\hat{D}_{\mathcal{K}} \left| \hat{D}_{\mathcal{A}} - D \right| &\leq \left(\mathbb{E}\hat{D}_{\mathcal{K}}^2 \right)^{1/2} \left(\mathbb{E} \left| \hat{D}_{\mathcal{A}} - D_{\mathcal{A}} \right|^2 \right)^{1/2} \\
&\leq \left(\left(\mathbb{E}\hat{D}_{\mathcal{K}} \right)^2 + \text{Var}(\hat{D}_{\mathcal{K}}) \right)^{1/2} \left(\text{Var}(\hat{D}_{\mathcal{A}}) \right)^{1/2} \\
&\leq \Big[\frac{\left(\text{Var}_{\vec{\partial}\mathcal{A}} T_E \right)^{1/2}}{\sqrt{M}} + \frac{\left(\text{Var}_{\vec{\partial}\mathcal{A}} T_{\mathcal{K}} \right)^{1/2}}{\sqrt{N}} \Big] \left((D_{\mathcal{K}})^2 + \frac{1}{M} \text{Var}_{\vec{\partial}\mathcal{A}^c}(D_{\mathcal{K}}) \right)^{1/2}
\end{aligned}$$

We therefore obtain

$$\mathbb{E} \int_0^\infty \hat{N}_t \left| \frac{1}{\hat{D}_{\mathcal{A}}} - \frac{1}{D_{\mathcal{A}}} \right| dt \leq \mathbb{E}_{\vec{\partial}\mathcal{A}^c} T_{\mathcal{K}} \Big[\frac{\left(\text{Var}_{\vec{\partial}\mathcal{A}} T_E \right)^{1/2}}{\sqrt{M}} + \frac{\left(\text{Var}_{\vec{\partial}\mathcal{A}} T_{\mathcal{K}} \right)^{1/2}}{\sqrt{N}} \Big] + \mathcal{O}(\frac{1}{M}).$$

We are left with bounding

$$\frac{1}{D_{\mathcal{A}}} \mathbb{E} \int_0^\infty \left| \hat{N}_t - N_t \right| dt.$$

Since $\hat{N}_t$ is an average of M Bernoulli random variables with mean N_t, we have:

$$\begin{aligned}
\frac{1}{D_{\mathcal{A}}} \mathbb{E} \int_0^\infty \left| \hat{N}_t - N_t \right| dt &= \frac{1}{D_{\mathcal{A}}} \int_0^\infty \mathbb{E} \left| \hat{N}_t - N_t \right| dt \\
&\leq \frac{1}{D_{\mathcal{A}}} \int_0^\infty \sqrt{\mathbb{E} \left| \hat{N}_t - N_t \right|^2} dt \\
&= \frac{1}{\sqrt{N} D_{\mathcal{A}}} \mathbb{E} \int_0^\infty \sqrt{N_t(1 - N_t)},
\end{aligned}$$

where in the first inequality we use $\mathbb{E}Y \leq \sqrt{\mathbb{E}Y^2}$ which follows from Cauchy-Schwartz inequality. We note, that $N_t = \mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_\mathcal{K} > t) = 1 - \mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_\mathcal{K} \leq t) = 1 - F_\mathcal{K}(t)$. Thus we have

$$\frac{1}{D_\mathcal{A}}\mathbb{E}\int_0^\infty \left|\hat{N}_t - N_t\right| dt \leq \frac{1}{\sqrt{N}\mathbb{E}_{\vec{\partial}\mathcal{A}}T_\mathcal{A}}\int_0^\infty \sqrt{F_\mathcal{K}(t)(1-F_\mathcal{K}(t))}dt.$$

Combining all of the above inequalities in an obvious manner, we obtain the bound from the thesis.

It follows from exponential ergodicity of X_t^π that $\overline{T}_\mathcal{A}, \overline{T}_\mathcal{K}$ have exponential tails, that is, there exist constants $C_\mathcal{A}, \lambda_\mathcal{A}, C_\mathcal{K}, \lambda_\mathcal{K}$ such that $\mathbb{P}_{\vec{\partial}\mathcal{A}}(T_\mathcal{A} > t) \leq C_\mathcal{A}\exp(-\lambda_\mathcal{A} t)$ and $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_\mathcal{K} > t) \leq C_\mathcal{K}\exp(-\lambda_K t)$. Therefore all the moments and the integral above are finite.

C Proof of the convergence of the FVRL algorithm to a local optimizer

C.1 Proof of Proposition 4

Our optimisation control problem falls within the scope of Theorem 2.1 in [14, Section 5.2]. This theorem provides a convergence guarantee to a limit set of a projected ordinary differential equation (ODE), which includes the FVRL algorithm as a special case. To see this, we note that the θ update rule given by FVRL can be thought of as the discretisation of the projected ODE on the time variable t, $\dot{\theta}(t) = \nabla_\theta g(\pi_\theta) + z(t)$, where $g(\pi_\theta)$ is defined in (6) and $z(t)$ is the projection term representing the minimum force needed to keep $\theta(t)$ in a pre-defined compact set $\mathcal{K} \subset \mathbb{R}^d$ (for more details see [14, Section 4.3, "Projected ODE"]). This projected ODE is a special case of the more general case considered in Theorem 2.1, where the driving process defining the dynamics of $\theta(t)$ is a gradient funciont of $\theta(t)$ itself. Under this context, the limit set to which the trajectories of the projected ODE converge is actually called a stationary set. In turn, the stationary points in this set are easily seen to be local optimizers of the $g(\pi_\theta)$ function. In fact, they are either stationary points of $g(\pi_\theta)$ when inside the compact set $\mathcal{K}$ where $z(t)$ is zero, or located at the boundary of the compact set, where $z(t) = -\nabla_\theta g(\pi_\theta)$, making $\dot{\theta} = 0$.

We now prove that all the assumptions (1.1), (A2.1)-(A2.5) for Theorem 2.1 in [14] are satisfied by the FVRL algorithm, with which a biased estimate of the unknown gradient $\nabla_\theta g(\pi_\theta)$ is obtained. For ease of notation we use thereafter ∇ to mean ∇_θ.

Assumptions (1.1) and (A2.4) require that the learning rate satisfies the Robbins-Monro conditions for the convergence of stochastic approximation algorithms [23], $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 < \infty$. This is directly satisfied by our choice of the learning rate scheduling as $\alpha_k = \alpha_0/k$ for some $\alpha_0 > 0$, as per item h) of the FVRL Algorithm 1.

Assumption (A2.1) requires finite second moments of the estimated gradient for all iterates k. This follows from:

$$\begin{aligned}\sup_k \mathbb{E}^{\pi_{\theta_k}}\left\|\hat{\nabla} g(\pi_{\theta_k})\right\|^2 &= \sup_k \mathbb{E}^{\pi_{\theta_k}}\Big[\sum_{x\in\mathcal{S}}\sum_{y\in\mathcal{S}} \hat{p}_{FV}^{\pi_{\theta_k}}(x)\hat{p}_{FV}^{\pi_{\theta_k}}(y)\times \\ &\qquad \sum_{a\in\mathcal{U}}\sum_{b\in\mathcal{U}}\langle \hat{Q}^{\pi_{\theta_k}}(x,a)\nabla\pi_{\theta_k}(a|x), \hat{Q}^{\pi_{\theta_k}}(y,b)\nabla\pi_{\theta_k}(b|y)\rangle\Big] \\ &\leq |\mathcal{S}|^2|\mathcal{U}|^2 \sup_k \mathbb{E}^{\pi_{\theta_k}}\left\{\max_{z\in\mathcal{S},c\in\mathcal{U}}\left[\hat{Q}^{\pi_{\theta_k}}(z,c)\left\|\nabla\pi_{\theta_k}(c|z)\right\|\right]^2\right\}.\end{aligned}$$

Because rewards are by assumption bounded, $\hat{Q}^{\pi_{\theta_k}}(z,c)$ is bounded for all states z and actions c, and since policy π_θ is assumed to be a continuously differentiable function of θ in a compact set $\mathcal{K} \in \mathbb{R}^d$, $\|\nabla\pi_{\theta_k}(c|z)\|$ is also bounded for all states z and actions c, and this holds for all k. Hence the supremum over k of the second moment of the estimated long-run expected reward gradient is finite.

Assumption (A2.2) requires that the expectation of the θ-update driving process conditional to the past be decomposed as a measurable function of θ plus a random process β_k. In our case, this is automatically satisfied because

the θ-update driving process, $\hat{\nabla} g(\pi_{\theta_k})$, is a function of θ_k and thus independent of its past values, $\hat{\nabla} g(\pi_{\theta_l}), l < k$. Hence, $\mathbb{E}[\hat{\nabla} g(\pi_{\theta_k}) | \theta_0, \hat{\nabla} g(\pi_{\theta_l}), l < k] = \mathbb{E}[\hat{\nabla} g(\pi_{\theta_k})]$. The assumption then follows from the fact that $\mathbb{E}[\hat{\nabla} g(\pi_{\theta_k})] = \nabla g(\pi_{\theta_k}) + \beta_k$, where β_k is a deterministic bias term.

Assumption (A2.3) requires that $\nabla g(\pi_\theta)$ is continuous. This immediately follows from the expression for $\nabla g(\pi_\theta)$ in (7), using the proposition's assumption that π_θ is a continuous differentiable function of θ in a compact set $\mathcal{K}$, that the state and action spaces are finite, and that the action value function $Q^{\pi_\theta}(x, a)$ is bounded because rewards are bounded.

Assumption (A2.5) requires that the bias term $\{\beta_k\}_{k \geq 0}$ of the gradient goes to zero fast enough, namely such that $\sum_k \alpha_k |\beta_k| < \infty$. If hyperparameters M_k, N_k scale at least as fast as k^ϵ and hyperparameter S_k scales at least as fast as $\log k^\epsilon$ for some $\epsilon > 0$, namely, $M_k = \lceil M_0 k^\epsilon \rceil$, $N_k = \lceil N_0 k^\epsilon \rceil$, $S_k = \lceil S_0 \log k^\epsilon \rceil$, then using Lemma 5, there exist constants $A, B, C, c > 0$ such that the bias β_k is bounded by

$$\begin{aligned} |\beta_k| &\leq A \left(\frac{1}{\sqrt{M_0 k^\epsilon}} + \frac{1}{\sqrt{N_0 k^\epsilon}} \right) + B e^{-c S_0 \log k^\epsilon} + C e^{-c S_0 \log k^\epsilon} \left(\frac{1}{\sqrt{M_0 k^\epsilon}} + \frac{1}{\sqrt{N_0 k^\epsilon}} \right) \\ &= \left(\frac{A'}{k^{\epsilon/2}} + \frac{B}{k^{\epsilon c S_0}} + \frac{C'}{k^{\epsilon(1/2 + c S_0)}} \right), \end{aligned}$$

where $A' = A \left(\frac{1}{\sqrt{M_0}} + \frac{1}{\sqrt{N_0}} \right)$ and $C' = C \left(\frac{1}{\sqrt{M_0}} + \frac{1}{\sqrt{N_0}} \right)$.

This yields the convergence of the $\sum_k \alpha_k |\beta_k|$ series, since by assumption $\alpha_k = \alpha_0 / k$, and

$$\sum_k \alpha_k |\beta_k| \leq \alpha_0 \left(A' \sum_k \frac{1}{k^{1+\epsilon/2}} + B \sum_k \frac{1}{k^{1+\epsilon c S_0}} + C' \sum_k \frac{1}{k^{1+\epsilon(1/2+cS_0)}} \right) < \infty,$$

as all the exponents of k are strictly larger than 1 for all $\epsilon, c > 0$.

Remark 4. *If a more general learning rate scheduling of the FVRL algorithm were used, such as $\alpha_k = \alpha_0 / k^\gamma$ for some $\frac{1}{2} < \gamma \leq 1$ (still satisfying assumptions (1.1) and (A2.4)), Assumption (A2.5) would no longer be satisfied for all $\epsilon > 0$. The reason is that the exponents of k in the two last series of the last expression in the proof would be equal to $\gamma + \epsilon c S_0$ and $\gamma + \epsilon(1/2 + c S_0)$, respectively. This would require knowledge of the constant c in order to determine the minimum value of ϵ guaranteeing that those exponents are greater than 1, as required for the series in (A2.5) to converge.*

C.2 Proof of Lemma 5

Let $\eta^{\pi_\theta}(x) \doteq \nabla_\theta g_x(\pi_\theta)$ and its estimator be denoted by $\hat{\eta}^{\pi_\theta}(x)$. Then, the bias β of the gradient estimated by item d) of the FVRL Algorithm 1 is given by:

$$\begin{aligned} \beta &= \mathbb{E}^{\pi_\theta} \left[\hat{\nabla}_\theta g(\pi_\theta) \right] - \nabla_\theta g(\pi_\theta) = \sum_{x \in \mathcal{S}'} \mathbb{E}^{\pi_\theta} \left[\hat{p}_{FV}^{\pi_\theta}(x) \right] \mathbb{E}^{\pi_\theta} \left[\hat{\eta}^{\pi_\theta}(x) \right] - \sum_{x \in \mathcal{S}} p_{FV}^{\pi_\theta}(x) \eta^{\pi_\theta}(x) \\ &= \sum_{x \in \mathcal{S}} (p^{\pi_\theta}(x) + b_p(x))(\eta^{\pi_\theta}(x) + b_\eta(x)) - \sum_{x \in \mathcal{S}} p_{FV}^{\pi_\theta}(x) \eta^{\pi_\theta}(x) \\ &= \sum_{x \in \mathcal{S}} b_p(x) \eta^{\pi_\theta}(x) + \sum_{x \in \mathcal{S}} b_\eta(x) p^{\pi_\theta}(x) + \sum_{x \in \mathcal{S}} b_p(x) b_\eta(x), \end{aligned} \tag{23}$$

where we have used the fact that the estimators $\hat{p}_{FV}^{\pi_\theta}(x)$ and $\hat{\eta}^{\pi_\theta}(x)$ are mutually independent (since they are estimated from independent simulations), and we have extended the sum over $\mathcal{S}'$ to the sum over $\mathcal{S}$, given that the gradient of the policy (intervening in $\hat{\eta}^{\pi_\theta}$) is zero outside $\mathcal{S}'$ (by definition of $\mathcal{S}'$ as the set of states where the policy gradient is not zero for at least one action –see initialization step of Algorithm 1).

We now bound each of the three bias terms in (23).

Bound for the bias $b_p(x)$ of $\hat{p}^{\pi_\theta}_{FV}(x)$: Given the number of particles N and the number M of return cycles to $\mathcal{A}$, by Theorem 2 the bias $b_p(x)$ of the stationary probability estimator $\hat{p}^{\pi_\theta}_{FV}(x)$ is upper bounded by:

$$|b_p(x)| = \left|\mathbb{E}[\hat{p}^{\pi_\theta}_{FV}(x)] - p^{\pi_\theta}(x)\right| \leq \mathbb{E}\left|\hat{p}^{\pi_\theta}_{FV}(x) - p^{\pi_\theta}(x)\right| \leq C_p\left(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{N}}\right),$$

for some $C_p > 0$.

Bound for the bias $b_\eta(x)$ of $\hat{\eta}^{\pi_\theta}(x)$: Using Lemma 3, the state contribution to the gradient, $\eta^{\pi_\theta}(x)$, can be estimated following the procedure described by items a) and b) for the FVRL Algorithm 1 on at most S time steps, as

$$\hat{\eta}^{\pi_\theta}(x) = \frac{1}{V}\sum_{v=1}^{V}\sum_{a\in\mathcal{U}}\sum_{n=1}^{\min(\tau_v,S)} R^{(x,a)}_{v,n}\nabla\pi_\theta(a|x),$$

where V is the number of replications run and τ_v is the value of the coupling time defined in Lemma 3 observed at replication v.

Using that all rewards $\{R^{(x,a)}_{v,n}\}_n$ for each replication v are identically distributed, the expectation of $\hat{\eta}^{\pi_\theta}(x)$ can be written as:

$$\mathbb{E}^{\pi_\theta}[\hat{\eta}^{\pi_\theta}(x)] = \sum_{a\in\mathcal{U}}\mathbb{E}^{\pi_\theta}\Big[\sum_{n=1}^{\tau} R_n\mathbf{1}_{\{\tau\leq S\}} + \sum_{n=1}^{S} R_n\mathbf{1}_{\{\tau>S\}}|X_0 = x, A_0 = a\Big]\nabla\pi_\theta(a|x), \tag{24}$$

where $\{R_n\}_n$ is the reward process and τ is the coupling time defined in Lemma 3.

Hence, using (24), the expression of $\eta^{\pi_\theta}(x)$ given by Lemma 3, and the assumption of bounded rewards, there exists a positive constant C_r such that $|R_n| \leq C_r$ and the bias of $\hat{\eta}^{\pi_\theta}(x)$ is bounded by:

$$\begin{aligned}
|b_\eta(x))| &= |\mathbb{E}^{\pi_\theta}[\hat{\eta}^{\pi_\theta}(x)] - \eta^{\pi_\theta}(x)| \\
&= \left|\sum_{a\in\mathcal{U}}\mathbb{E}^{\pi_\theta}\left[\sum_{n=\tau+1}^{S} R_n\mathbf{1}_{\{\tau>S\}}\big|X_0 = x, A_0 = a\right]\nabla\pi_\theta(a|x)\right| \\
&\leq \sum_{a\in\mathcal{U}}\mathbb{E}^{\pi_\theta}\left[\sum_{n=1}^{S}|R_n|\,\mathbf{1}_{\{\tau>S\}}\big|X_0 = x, A_0 = a\right]\|\nabla\pi_\theta(a|x)\| \\
&\leq C_r S\sum_{a\in\mathcal{U}}\mathbb{P}(\tau > S\big|X_0 = x)\,\|\nabla\pi_\theta(a|x)\|,
\end{aligned}$$

Since the state space is finite and the MDP is assumed to be ergodic for all policies in the parameterised policy class, $\mathbb{P}(\tau > S|X_0 = x) \leq e^{-c_0 S}$ for some positive constant c_0 (Doeblin's coupling theorem for ergodic Markov chains, [5, Section 4.3.1]). Finally, because θ takes values in a compact set and the policy is assumed to be a continuous differentiable parameterisation by θ, the norm of the policy gradient is bounded by some $C_\pi > 0$, and this yields the following bound for the bias:

$$|b_\eta(x))| \leq C_r C_\pi|\mathcal{U}|Se^{-c_0 S} \leq C_\eta e^{-cS} \text{ for some } C_\eta > 0, c > 0,$$

i.e. the bias of $\hat{\eta}^{\pi_\theta}(x)$ tends to zero exponentially fast with an increasing S.

Finally, using that the state and action spaces are finite, that π_θ is a continuous differentiable function of θ in a compact set $\mathcal{K}$, and that the action value function $Q^{\pi_\theta}(x,a)$ is bounded because rewards are bounded by assumption,

$\eta^{\pi_\theta}(x)$ is upper bounded by some $B_\eta > 0$. Thus, the bias term β can be bounded as:

$$\begin{aligned}|\beta| &\leq \sum_{x\in\mathcal{S}} |b_p(x)\eta^{\pi_\theta}(x)| + \sum_{x\in\mathcal{S}} |b_\eta(x)|p^{\pi_\theta}(x) + \sum_{x\in\mathcal{S}} |b_p(x)||b_\eta(x)| \\ &\leq |\mathcal{S}| \left(C_p B_\eta \left(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{N}} \right) + C_\eta e^{-cS} + C_p C_\eta e^{-cS} \left(\frac{1}{\sqrt{M}} + \frac{1}{\sqrt{N}} \right) \right).\end{aligned}$$

The lemma then follows by defining the positive constants $A = |\mathcal{S}|C_p B_\eta, B = |\mathcal{S}|C_\eta, C = |\mathcal{S}|C_p C_\eta$.

D Heuristics for the choice of the simulation parameters for FVEE

In this section we leverage theoretical results on the well-studied $M/M/1/K$ queue system to provide insights about the appropriate choice of hyperparameters J, N and T of the Fleming-Viot estimation procedure of the blocking probability described in Section 4.1.1, so that they can be used as an initial guideline for the choice of these parameters in a more general setting such as multidimensional problems.

According to Theorem 2, the error of the Fleming-Viot estimator is controlled by two independent quantities: N, the number of Fleming-Viot particles, and M, the number of return cycles to $\mathcal{A}$ observed in the simulation used to estimate the denominator, $\mathbb{E}(T_\mathcal{A})$ (where the expectation subscript has been dropped for conciseness). While N is a hyperparameter of the Fleming-Viot simulation, M is a random variable that depends on both J and T. As stated in Section 4.1.1, J is problem dependent (e.g. it should be chosen as a function of the state visit frequency), which in practice makes M a function of hyperparameter T. Clearly, for a fixed value of J, a larger number of cycles M is expected for a larger number of arrivals T allowed in the $\mathbb{E}(T_\mathcal{A})$ simulation, since more time is allowed for the simulation.

Therefore, in what follows our analysis focuses on how the error of the Fleming-Viot estimator varies as N and T vary, while keeping J fixed to a predefined quantity. The value of J is chosen so that a few meaningful (N, T) value pairs can be considered for the estimation of the blocking probability which involve simulations that take a reasonable amount of time to run. The criterion that J defines the number of states in the $M/M/1/K$ system having a stationary probability larger than 0.5% proved reasonable. For a blocking size $K = 20$ and system load $\rho = 0.7$, this criterion yields $J = 12$.

We focus the estimator error analysis on the two quantities (out of the three involved in expression (10) of Section 4.1.1) whose estimation is highly affected by the trade-off of the choice of J described in remark 3: $\phi_t^J(K)$ and $\mathbb{E}_{J-1}T_\mathcal{A}$; the remaining quantity, $\mathbb{P}_J(T_\mathcal{K} > t)$, does not strongly depend on J (see Appendix B)[7]. The trade-off described in the remark states that, for constant values of N and T, the error in the estimation of $\phi_t^J(K)$ tends to decrease as J increases, while the error in the estimation of $\mathbb{E}_{J-1}T_\mathcal{A}$ tends to increase. As mentioned above, we now invert our reasoning: we fix J, and set the values of N and T required to approximately satisfy predefined expected relative errors in these estimators. We then run simulations using each (N, T) value pair dictated by the relative error pairs considered, and study statistics on the relative error of the Fleming-Viot estimator of the blocking probability to understand how it varies as a function of N and T.

The values of N and T for each expected relative error pair are determined based on the following heuristics that approximately express the two expected relative errors as a function of J, N, T, and the system's characteristics:

1. **Relative error of $\hat{\phi}_t^J(K)$**: For fixed J, the expected relative error of $\hat{\phi}_t^J(K)$ is approximately inversely proportional to $\sqrt{N}$.

   Derivation: Considering that in an $M/M/1/K$ queue system with $\rho < 1$, the conditioned blocking probability $\phi_t^{J=1}(K)$ converges as $t \to \infty$ towards $\sqrt{K}\rho^{K/2}$ [10], i.e. $\sim O(\rho^{K/2})$, similarly, the conditioned blocking

[7] Although the error of the estimator of $\mathbb{P}_J(T_\mathcal{K} > t)$ depends on N, it is sensible to assume that its estimation error decreases as N increases similarly or faster than the error of the estimator of $\phi_t^J(K)$, so we can limit our analysis to the error of the estimator of the latter.

probability for any absorption set size J, $\phi_t^J(K)$, increases to $\sim O(\rho^{(K-J+1)/2})$ as $t \to \infty$. As a very rough approximation and to get order of magnitude relations, we could think of $\phi_t^J(K)$ as the blocking probability q_J of a single-server queue system with capacity equal to $\lceil \frac{K-J+1}{2} \rceil$ ($\lceil \cdot \rceil$ is the ceiling operator), namely $q_J = \frac{(1-\rho)\rho^{K_J}}{1-\rho^{K_J+1}}$, where $K_J = \lceil (K-J+1)/2 \rceil$. Under this assumption, and ignoring the interdependence among FV particles, the relative error of estimating $\phi_t^J(K)$ with the empirical mean on N samples is derived from the variance of a Binomial(q_J) random variable as $\epsilon_\phi \approx \sqrt{(1-q_J)/Nq_J}$. Thus, given q_J, the value of N to approximately satisfy a desired expected relative error ϵ_ϕ for $\hat{\phi}_t^J(K)$ is obtained as $N \approx \lceil \frac{1-q_J}{q_J \epsilon_\phi^2} \rceil$, which in turn can be approximated as $N \approx \lceil \frac{1}{q_J \epsilon_\phi^2} \rceil$, if $q_J \ll 1$, i.e. when J is sufficiently small compared to K.

We finally solve for ϵ_ϕ to get its inverse relationship with $\sqrt{N}$ stated above.

2. **Relative error of** $\hat{\mathbb{E}}_{J-1} T_{\mathcal{A}}$: For fixed J, the expected relative error of $\hat{\mathbb{E}}_{J-1} T_{\mathcal{A}}$ is approximately inversely proportional to $\sqrt{T}$.

   Derivation: In an $M/M/1/K$ queue system with $\rho < 1$, the expected return time to the state $J-1$, when starting at $J-1$, is equal to $\mathbb{E}_{J-1}(T_{J-1}) = \frac{1}{\lambda p_J (1+\rho^{-1})}$ [5], where p_J is the stationary probability of the state $x = J-1$, i.e. $p_J = \frac{(1-\rho)\rho^{J-1}}{1-\rho^{K+1}}$. Given the absorption set $\mathcal{A} = \{0, 1, ..., J-1\}$, it can easily be shown that the expected return time to $\mathcal{A}$, when starting at $J-1$, is equal to $\mathbb{E}_{J-1} T_{\mathcal{A}} = \mathbb{E}_{J-1}(T_{J-1})(1+\rho^{-1})$. If we want to observe M return cycles to $\mathcal{A}$, we should simulate the queue system for as long as $t_S = M\mathbb{E}_{J-1} T_{\mathcal{A}} = M\mathbb{E}_{J-1}(T_{J-1})(1+\rho^{-1})$, i.e. $t_S = \frac{M}{\lambda p_J}$. Since λt_S is the expected number of arrival events T observed in the time span t_S, we should simulate the system for as long as $T \approx M/p_J$ arrival events. On the other hand, given M return cycles to $\mathcal{A}$, the standard error of the moment estimator of $\mathbb{E}_{J-1} T_{\mathcal{A}}$ is given by $\sigma(T_{\mathcal{A}})/\sqrt{M}$. It is reasonable to assume (confirmed by experiments) that $\sigma(T_{\mathcal{A}}) \approx \mathbb{E}_{J-1} T_{\mathcal{A}}$, hence the relative error of the estimator of $\mathbb{E}_{J-1} T_{\mathcal{A}}$, ϵ_{ET}, is of the order of $1/\sqrt{M}$, which makes $M \approx 1/\epsilon_{ET}^2$. Thus, given p_J, the value of T to approximately satisfy a desired expected relative error ϵ_{ET} for $\hat{\mathbb{E}}_{J-1} T_{\mathcal{A}}$ is obtained as $T \approx \lceil \frac{1}{p_J \epsilon_{ET}^2} \rceil$.

   We finally solve for ϵ_{ET} to get its inverse relationship with $\sqrt{T}$ stated above.

From the above, the following common aspects are observed about the minimum values of N and T required to satisfy predefined expected relative errors in the estimators of $\phi_t^J(K)$ and $\mathbb{E}_{J-1} T_{\mathcal{A}}$:

1. N affects the relative error of $\hat{\phi}_t^J(K)$ and T affects the relative error of $\hat{\mathbb{E}}_{J-1} T_{\mathcal{A}}$.
2. The relationship of N and T with their respective relative errors is of the same form, i.e. proportional to the inverse of a stationary probability and to the inverse of the squared expected relative error.

On the other hand, the following difference is observed: for $\rho < 1$ and sufficiently large K[8], the value of N as a function of the stationary probability q_J is dominated by an increasing exponential function of $(K-J)/2$, i.e. $\sim O(\rho^{-(K-J)/2})$ whereas the value of T as a function of the stationary probability p_J is dominated by an increasing exponential function of J, i.e. $\sim O(\rho^{-J})$ which, remarkably, does not depend on K. Thus, as mentioned in remark 3, the closer J is to 0, the smaller the required T and the larger the required N for fixed expected relative errors, while the opposite is true when J gets closer to K.

More importantly, experiments showed that the computational complexity –in terms of number of observed events– required to satisfy a given ϵ_ϕ value (which impacts the number of observed events of the FV simulation) is much larger than the computational complexity required to satisfy the same value in ϵ_{ET} (which impacts the number of observed events in the single simulation of the X_t^π process). Concrete values of the respective computational complexities can be derived from the results shown in Figure 2 of Section 4.1.1. Thus, from the computational

[8] K is considered sufficiently large in this context when ρ^{K-J} can be neglected w.r.t. 1.

perspective alone, it is more convenient to favour a smaller error in $\hat{\mathbb{E}}_{J-1}T_{\mathcal{A}}$ than a smaller error in $\hat{\phi}_t^J(K)$. Below we will see that this is also the case in terms of the error of the FV estimator of the blocking probability.

We completed the study of the appropriate choice of N and T by analyzing the impact of the errors of estimating $\phi_t^J(K)$ and $\mathbb{E}_{J-1}T_{\mathcal{A}}$ on the FV estimator of the blocking probability. To this end, we ran experiments on different combinations of expected relative errors ϵ_ϕ and ϵ_{ET} and measured the accuracy in the estimation of the blocking probability of an $M/M/1/K$ system with $\rho = \lambda = 0.7, K = 20$, using a constant absorption set size of $J = 12$, which corresponds to choosing the states with stationary probability larger than 0.5%. The results of these experiments are shown in the heatmap of Figure 7 in terms of the estimation accuracy of the blocking probability, $\hat{p}_{FV}(K)$.

From this heatmap, we conclude that it is more important to control the relative error in $\hat{\mathbb{E}}_{J-1}T_{\mathcal{A}}$ than the relative error in $\hat{\phi}_t^J(K)$, as the contour lines are almost parallel to the axis where the relative error in $\hat{\phi}_t^J(K)$ is plotted, and their values indicate that the estimated blocking probability is larger than 1.5 times the true blocking probability when the relative error in $\hat{\phi}_t^J(K)$ is larger than $30\% - 40\%$. Note that, for each experiment run, the FV estimator of the blocking probability was computed only when a minimum of 5 return cycles to $J-1$ were observed (these cycles are used to estimate the denominator $\mathbb{E}_{J-1}T_{\mathcal{A}}$) after the 10 initial transitions of the system. This 10 transitions were used as a burn-in period to allow the system to get closer to the stationary regime. Therefore, if the number T of arrival events for estimating $\mathbb{E}(T_{\mathcal{A}})$ is not large enough, the sample size on which the plotted median FV estimator is computed may be smaller than the 7 replications used for each N-T combination.

In Figure 2 we used these heuristics to choose the different values of N and T on which the convergence properties of the FV estimator were analyzed: given $J = 12$, for the left plots (a) and (c), T was fixed at the value associated to an approximate expected relative error in $\hat{\mathbb{E}}_{J-1}T_{\mathcal{A}}$ equal to $\epsilon_{ET} = 20\%$, whereas the values of N were chosen for approximate expected relative errors in $\hat{\phi}_t^J(K)$ equal to $\epsilon_\phi = 20\%, 10\%, 5\%$ for $K = 20$, and equal to $\epsilon_\phi = 80\%, 60\%, 40\%$[9]; for the right plots (b) and (d), N was fixed at the value associated to an approximate expected relative error in $\hat{\phi}_t^J(K)$ equal to $\epsilon_\phi = 60\%$, whereas the values of T were chosen for approximate expected relative errors in $\hat{\mathbb{E}}_{J-1}T_{\mathcal{A}}$ equal to $\epsilon_{ET} = 40\%, 20\%$[10].

[9]The larger relative errors chosen for $K = 40$ compared to $K = 20$ have to do with obtaining the same orders of magnitude for N in each K scenario.

[10]The expected relative error ϵ_{ET} depends on K only through $1 - \rho^{K+1}$, and this term can be safely approximated by 1 for large enough values of K such as 20 and 40 when $\rho = 0.7$. Thus, doing this approximation, the value of T satisfying a given ϵ_{ET} only depends on J and is thus the same for both K scenarios.

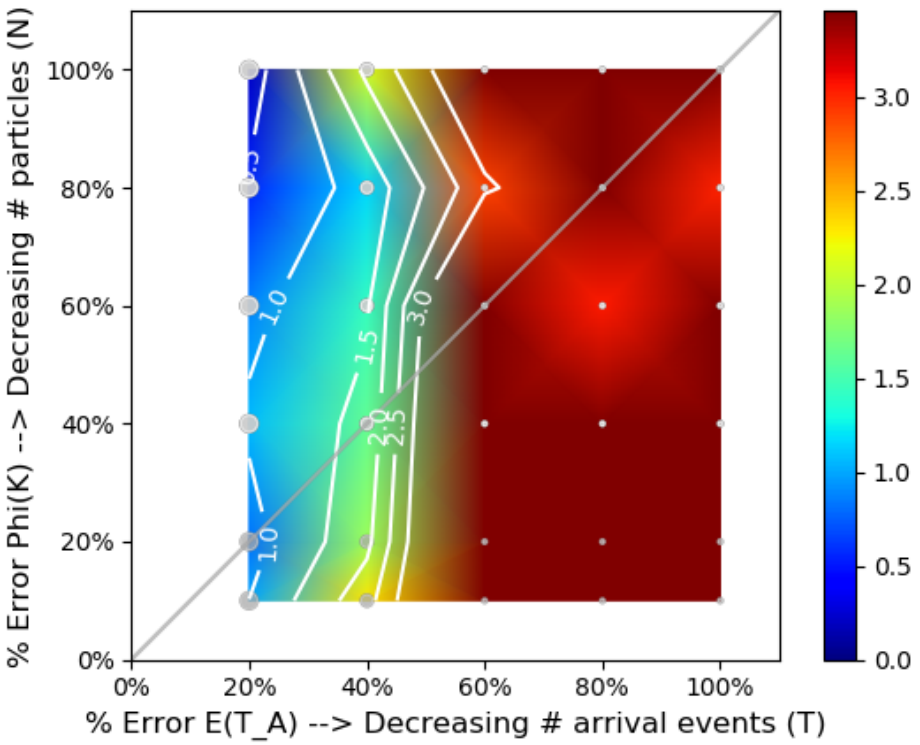


Figure 7: Heatmap showing the impact of different combinations of the expected relative errors of $\hat{\phi}_t^J(K)$ and $\hat{\mathbb{E}}_{J-1}T_{\mathcal{A}}$ (indicated respectively on the vertical and horizontal axis) on the accuracy of the FV estimator of the blocking probability, $\hat{p}_{FV}(K)$, in an $M/M/1/K$ queue system over up to 7 valid experiments carried out for each combination. The system characteristics are $K = 20, \lambda = 0.7, \mu = 1$. The accuracy is shown as the ratio between the median $\hat{p}_{FV}(K)$ value and the true blocking probability $p(K)$. Thus, a ratio of 1 (light blue) implies 100% median accuracy, a ratio smaller than 1 (dark blue) implies underestimation, and a ratio larger than 1 (from cyan to red) implies overestimation. Selected contour levels are overlaid. The gray diagonal represents the line of equal expected relative errors in the two analyzed dimensions, and the gray points indicate the $5 \times 6 = 30$ error combinations on which experiments were run, whose size is proportional to the number of experiments (7 at the largest points, down to 2 at the smallest points). Smaller points are associated to a larger expected relative error in the estimation of $\mathbb{E}_{J-1}T_{\mathcal{A}}$ which tend to preclude its estimation due to an insufficient number of observed return cycles to the absorption set $\mathcal{A}$, as described in the text. The color scale is chosen as the transformation $log_2(1 + z)$ where z is the ratio between the estimated and the true blocking probability; note that this transformation maps 0 to 0 and 1 to 1.

E FVEE and FVRL algorithms for threshold-type policies

This section presents the two algorithms used throughout this work to apply the Fleming-Viot methodology to network systems under threshold-type admission control policies, in order to *(i)* estimate the rejection probability and the expected rejection cost using the Fleming-Viot expectation estimator (FVEE Algorithm 2), and *(ii)* learn locally optimal blocking sizes of threshold-type admission control policies using the Fleming-Viot reinforcement learning method (FVRL Algorithm 3). Note that the latter is essentially the FVRL Algorithm 1 presented in the paper's body, adapted to piecewise-linear admission control policies defined by (12).

Algorithm 2 FVEE algorithm for the estimation of the expected rejection cost in an $M/M/I/R$ loss network serving I different job classes with R servers, of which the $M/M/1$ queue system is a particular case (with $I = 1, R = 1$).

Data:

A. System characteristics: a loss network with R servers serving jobs of I different classes whose state is represented by the number of jobs of each class in the system, $\mathbf{x} = (x_1, ..., x_I)$.

B. System dynamics: the job arrival rate λ_i and the service rate μ_i for each job class $i = 1, \ldots, I$ are assumed known for the estimation problem.

C. Job acceptance policy: an incoming job of class i is either accepted ($a = 1$) or rejected ($a = 0$). It is accepted whenever the system is not operating at full capacity R and when the number of jobs of the arriving class being served by the system is less than a constant K_i, $i = 1, \ldots, I$. Otherwise, it is rejected, in which case a cost C_i is accrued. Thus, the job acceptance policy, when the system is at state $\mathbf{x}$ and a job of class i arrives ($\mathcal{L}_i$), is $\pi(a = 1|\mathbf{x}, \mathcal{L}_i) = \mathbf{1}_{\{\sum_{j=1}^{I} x_j < R\}} \mathbf{1}_{\{x_i < K_i\}}$, and the set of blocking states is $\mathcal{C} = \{\mathbf{x} : \sum_{j=1}^{I} x_j = R \text{ or } \sum_{j=1}^{I} x_j < R, x_i = K_i \text{ for some } i = 1, \ldots I\}$.

D. J_i, $i = 1, \ldots, I$, $0 \leq J_i \leq K_i$: size of the absorption set $\mathcal{A}$ in dimension i, that is whenever the state of an FV particle visits a state $\mathbf{x}$ having $x_i = J_i - 1$ for some i, the particle is considered absorbed.

E. N: number of FV particles used to estimate $\phi_t^{\vec{\partial}\mathcal{A}^c}(K)$ and $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_{\mathcal{K}} > t)$ in (3).

F. T: number of arrival events (incoming jobs) for estimating $\mathbb{E}(T_{\mathcal{A}})$, which has a direct impact on the number M of return cycles to $\mathcal{A}$ used to estimate $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_{\mathcal{A}}$ in (3).

G. B: number of burn-in time steps to assure stationarity, typically 10-20.

H. M_0: minimum number of return cycles to $\mathcal{A}$ to have a reliable estimate of $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_{\mathcal{A}}$, typically 5-10.

Result: An estimate of the expected rejection cost under stationarity.

Steps: The algorithm is divided into the following three steps:

1. Estimation of $\mathbb{E}_{\vec{\partial}\mathcal{A}} T_{\mathcal{A}}$:
   (a) Simulate the continuous-time Markov process $\mathbf{X}_t^\pi$ starting at $\mathbf{X}_0^\pi$ chosen uniformly at random from the set $\vec{\partial}\mathcal{A} = \{\mathbf{x} : x_i = J_i - 1 \text{ for some } i = 1, \ldots, I\}$.
   (b) Record the observed values of the cycle time $T_{\mathcal{A}}$ defined in Section 3.1, measured as return times to the set $\mathcal{A}$ after the first visit to a state in $\vec{\partial}\mathcal{A}$ following the burn-in period given by parameter B.
   (c) Record the states at which the process enters $\mathcal{A}^c$.
   (d) Stop the simulation when the number of arrival events is equal to T, and record the number M of observed complete return cycles to $\mathcal{A}$ after the burn-in period.
   (e) If $M > M_0$, estimate $\mathbb{E}^{\vec{\partial}\mathcal{A}} T_{\mathcal{A}}$ as $\frac{1}{M}\sum_{k=1}^{M} T_{\mathcal{A},k}$, o.w. conclude that the expected rejection cost cannot be reliably estimated, and end the process here.
   (f) Estimate the entrance distribution to $\mathcal{A}^c$, $\hat{p}^\pi_{\vec{\partial}\mathcal{A}^c}(\mathbf{x})$ as $\frac{1}{\overline{M}}\sum_{k=1}^{\overline{M}} \mathbf{1}_{\mathbf{x}(k)=\mathbf{x}}, \forall \mathbf{x} \in \vec{\partial}\mathcal{A}^c$, where $\overline{M}$ is the number of observed entrances to $\mathcal{A}^c$ at states $\mathbf{x}(k)$.

2. Estimation of $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_{\mathcal{K}} > t)$ and $\phi_t^{\vec{\partial}\mathcal{A}^c}(\mathbf{x}_{\mathcal{C}})$, where $\mathbf{x}_{\mathcal{C}}$ is any state in the set of blocking states $\mathcal{C}$:
   (a) Simultaneously simulate N trajectories (FV particles) that independently follow the law of the Markov process $\mathbf{X}_t^\pi$, starting at $\mathbf{X}_0^\pi$ chosen in $\vec{\partial}\mathcal{A}^c$ following $\hat{p}^\pi_{\vec{\partial}\mathcal{A}^c}$. Record the observed values of the first killing times $T_{\mathcal{K}}$ at which each of the N particles enters $\mathcal{A}$.
   (b) Every time one of the particles is killed at $\mathcal{A}$, reinitialize it to the state of one of the other $N-1$ particles, selected uniformly at random.
   (c) Stop when all N particles have been killed at least once, as this is the maximum time t that will contribute to the integral in expression (3).
   (d) For each t in the set $\{T_{\mathcal{K},k}\}_{k=1\ldots N}$, estimate $\mathbb{P}_{\vec{\partial}\mathcal{A}^c}(T_{\mathcal{K}} > t)$ as $\frac{1}{N}\sum_{k=1}^{N} \mathbf{1}_{T_{\mathcal{K},k}>t}$.
   (e) For each time t at which one of the particles changes state, estimate $\phi_t^{\vec{\partial}\mathcal{A}^c}(\mathbf{x}_{\mathcal{C}})$ as the proportion of particles that are at state $\mathbf{X}_t^\pi = \mathbf{x}_{\mathcal{C}}$.

3. Estimation of the expected rejection cost:
   Let $\eta(\mathbf{x})$ be the expected cost accrued at rejection over all possible arriving job classes when the system is in state $\mathbf{x} \in \mathcal{C}$, namely: $\eta(\mathbf{x}) = \sum_{i=1}^{I} C_i \lambda_i / \Lambda \left[\mathbf{1}_{\{\sum_{j=1}^{I} x_j = R\}} + \mathbf{1}_{\{\sum_{j=1}^{I} x_j < R\}} \mathbf{1}_{\{x_i = K_i\}}\right]$, where $\Lambda \doteq \sum_{i=1}^{I} \lambda_i$ is the total job arrival rate. Estimate the expected rejection cost, $\mathbb{E}(\eta)$, using (1), with the stationary probability estimated using (4). The integral giving $\hat{p}^\pi_{FV}(\mathbf{x})$ is easily computed as a finite sum of the piecewise constant function of time, resulting from the product $\hat{\mathbb{P}}_{\vec{\partial}\mathcal{A}^c}(T_{\mathcal{K}} > t)\ \hat{\phi}_t^{\vec{\partial}\mathcal{A}^c}(\mathbf{x})$, which is 0 for $t > \max(\{T_{\mathcal{K},k}\}_{k=1\ldots N})$.

Algorithm 3 FVRL algorithm for piecewise-linear admission control policies to find blocking sizes K_i that locally minimize

the expected cost defined in (15).
The system is an $M/M/I/R$ loss network serving I different job classes with R servers, of which the $M/M/1$ queue system is a particular case (with $I = 1, R = 1$).

Data:

A. A simulator or emulator of the loss network or queue system to control (in our case, a simulator was obtained by defining the arrival rate λ_i and the service rate μ_i of each job class $i = 1, \ldots, I$).

B. Cost of rejecting an arriving job class, C_i, $i = 1, \ldots, I$.

C. $\pi_{\theta_i}(\text{"accept"}|x_i)$: the job acceptance policy for an arriving job of class i parameterised by the positive real-valued θ_i, as defined in expression (12).

D. $\theta = (\theta_i)_{i=1,\ldots,I}$: positive non-integral initial values of the parameter to optimise, from where K_i can be obtained as $K_i = \text{ceiling}(\theta_i + 1)$.

E. F_i: the size of the absorption set $\mathcal{A}$ in dimension i as a fraction of K_i.

F. L: number of learning steps, i.e. the number of times an update of θ will be computed by the gradient-based algorithm.

G. N, T: respectively, the number of FV particles and the number of arrival events for estimating $\mathbb{E}(T_{\mathcal{A}})$ used to estimate the blocking probability with the FVEE Algorithm 2.

H. V, S: respectively, the number of replications and the maximum number of time steps allowed to estimate the function $\nabla_\theta g_x(\pi_\theta)$ defined in (8).

I. $\alpha_0 > 0$: initial learning rate used in the θ update (e.g. $\alpha_0 = 1$).

Result: An estimate of blocking sizes $\hat{K}_i^*$, $i = 1, \ldots, I$, that locally minimize
the expected rejection cost defined in (15).

Steps:

1. Compute the deterministic blocking sizes $K_i = \text{ceiling}(\theta_i + 1)$, $i = 1, \ldots, I$. Set J_i, the size of the absorption set $\mathcal{A}$ in dimension i as $J_i = \lceil F_i K_i \rceil$.

2. Estimate the stationary probabilities $p^{\pi_\theta}(\mathbf{x})$ for each $\mathbf{x}$ such that $x_i = K_i - 1$ and $\sum_{j=1}^{I} x_j < R$ using the FVEE Algorithm 2.

3. For each $\mathbf{x}$ considered in the previous step simulate V times two independent $\{X_n\}_{n\in\mathbb{N}}$ processes in parallel under the current policy π_θ, one with initial action $a = 0$ (reject first incoming job) and the other with initial action $a = 1$ (accept first incoming job). Collect the rewards observed by each copy, respectively $\{R_{v,n}^{(x,1)}\}_{n\geq 1}$ and $\{R_{v,n}^{(x,0)}\}_{n\geq 1}$ and stop the simulation when the two processes meet for the first time, or when each takes S steps. Use Lemma 3 to compute the moment estimate of the $\nabla_\theta g_x(\pi_\theta)$ function defined in (8), which in this piecewise-linear policy case, as per (16), reduces to: $\hat{\nabla}_\theta g_x(\pi_\theta) = \frac{1}{V}\sum_{v=1}^{V}\sum_{n=1}^{\min(\tau_v,S)}(R_{v,n}^{(x,1)} - R_{v,n}^{(x,0)})$, where τ_v is the first time the two processes meet at replication v.

4. Use the estimates of steps 2 and 3 to estimate the gradient of the long-run expected cost, $\hat{\nabla} g(\pi_\theta) = \{\frac{\partial g(\pi_\theta)}{\partial \theta_i}\}_{i=1,\ldots,I}$, using (16).

5. Update θ using gradient descent: $\theta \leftarrow \theta - \frac{\alpha_0}{k}\hat{\nabla} g(\pi_\theta)$, where k is the current learning step. Bound θ_i $(i = 1, \ldots, I)$ to the valid range, if applicable.

6. Repeat steps (1)-(5) until the number of learning steps L is reached.

7. Use the final value θ^L to compute the final blocking sizes estimated by the algorithm, $\hat{K}_i^*$, as $\text{round}(\theta_i^L) + 1$, $i = 1, \ldots, I$.