Symphony of experts: orchestration with adversarial insights in reinforcement learning

Matthieu Jonckheere, Chiara Mignacco, Gilles Stoltz

Symphony of experts: orchestration with adversarial insights in reinforcement learning

Matthieu Jonckheere
LAAS CNRS

Chiara Mignacco
Université Paris-Saclay

Gilles Stoltz
Université Paris-Saclay, CNRS

Abstract

Structured reinforcement learning leverages policies with advantageous properties to reach better performance, particularly in scenarios where exploration poses challenges. We explore this field through the concept of orchestration, where a (small) set of expert policies guides decision-making; the modeling thereof constitutes our first contribution. We then establish value-functions regret bounds for orchestration in the tabular setting by transferring regret-bound results from adversarial settings. We generalize and extend the analysis of natural policy gradient in Agarwal et al. [2021, Section 5.3] to arbitrary adversarial aggregation strategies. We also extend it to the case of estimated advantage functions, providing insights into sample complexity both in expectation and high probability. A key point of our approach lies in its arguably more transparent proofs compared to existing methods. Finally, we present simulations for a stochastic matching toy model.

1 INTRODUCTION

Markov decision processes (MDPs) and their model-free counterparts in reinforcement learning (RL) face nowadays complex situations with large state or action spaces, where the learning processes can be prohibitively slow. Consequently, there is a growing emphasis in current research on pioneering the next generation of more 'democratic' algorithms that can be applied effectively across a wider range of practical context (Qian et al. [2019]). A promising avenue of research builds on leveraging the inherent knowledge and structure present in many control and learning problems, allowing to develop computationally more efficient algorithms. This approach finds applications in various fields, including computing systems scheduling, energy system control, and matching problems (Qian et al. [2019], Perera and Kamalaruban [2021]). In such applications, the underlying MDPs often exhibit unique structural characteristics. For instance, restless multi-armed bandit problems (Gittins et al. [2011], Whittle [1985]) serve as valuable models for scheduling and resource allocation, while switching curves provide efficient policy parameterizations in certain MDPs (Iravani et al. [1997]).

One setting of structured RL involves orchestrating a predefined set of expert policies chosen for their efficiency or robustness considering the specific problem at hand. Having exploration being guided by expert policies may avoid pitfalls consisting of suboptimal decisions stemming from local minima. Additionally, orchestration can foster the concept of transfer learning, while it may contribute to system robustness and reliability in situations marked by noise or uncertainty, provided that at least some expert policies are optimized for such challenging conditions.

A variety of aggregation strategies for combining the outputs of expert policies have been explored so far. Notable approaches include those proposed by Jiang [2007], Song et al. [2023], Cuayáhuitl et al. [2019], which define weighted probabilities associated with selecting a particular action at each state. Additionally, Wiering and Van Hasselt [2008], Song et al. [2023] investigate schemes that employ weights based on Boltzmann additive or multiplicative principles. Some alternative methodologies advocate for a voting system, with examples such as the majority-voting algorithm as discussed in Jiang [2007], Wiering and Van Hasselt [2008], Hans and Udluft [2010], plurality voting (winner-takes-all) as mentioned in Jiang [2007], or rank-voting based on a Boltzmann distribution as outlined in Wiering and Van Hasselt [2008], Song et al. [2023]. In the latter approach, the ranks are determined by the probabilities assigned to specific actions

Preliminary work. Under review.

by the expert policies. Moreover, other counting mechanisms, including Borda counting, find consideration in the work presented by Jiang [2007]. All of these proposals are supported by numerical illustrations, ranging from simple toy examples to more ambitious experiments. Nevertheless, despite the promising numerical outcomes, their theoretical foundations remain notably underdeveloped.

Contributions First, we model orchestration in Section 2 by considering expert policies as "super-arms", i.e., by drawing actions in two steps: first, by picking a policy, and then, by picking the action based on this policy.

Second, we propose and analyse in detail a set of elementary policy-building schemes maintaining weights over the expert policies, with weights evolving according to the advantage functions of policies built over time. When the weights rely on the exponential potential and an oracle is available, this scheme coincides with the application of natural policy gradient strategies, as exposed in Agarwal et al. [2021], but is otherwise different both from value-learning and policy-gradient alternatives. Actually, our approach consists of a third way between value learning and policy gradient, as we detail below (and in Appendix F).

Formally, we leverage the proof technique of Agarwal et al. [2021, Section 5.3] to work with arbitrary adversarial-learning strategies (see the monograph by Cesa-Bianchi and Lugosi [2006b]). We transfer the $B_{T,K}$ adversarial regret bounds achieved by these strategies for T rounds and K experts into value regret bounds of order $B_{T,K}/T + \sqrt{T}$ in the case no oracle to compute the advantage functions is available and the latter are estimated. As the optimal orders of magnitude of $B_{T,K}$ are $\sqrt{T \ln K}$, the achieved rates are consistent with the literature. Because we rely on bounded estimators for advantage functions (unlike in Agarwal et al. [2021, Section 6]), we are actually able to get value regret bounds not only in expectation but also with high probability. Another desirable aspect of our approach relies on its transparent and elementary proofs.

Finally, we illustrate our findings through a simulation scenario involving a matching system.

Literature review. There has been a recent surge of interesting theoretical findings around the concept of policy gradient, in particular using the concept of gradient domination (Fazel et al. [2018], Xiao [2022]). Agarwal et al. [2021] proposes a comprehensive review of policy-gradient methods. As indicated above, of special interest to us is the approach taken by Agarwal et al. [2021, Section 5.3], originating in Kakade [2001], which consists of the so-called natural policy gradient algorithm with the softmax parameterization (i.e., exponential weights). However, we explain in Appendix D why this approach does not accomodate well the estimation of advantage functions.

Concurrently to our work, another noteworthy study by Jin et al. [2022] transfers regret bound from the adversarial setting — but from the bandit adversarial setting, while our approach relies on the full-information setting. Actually, Jin et al. [2022] take a different angle, in the Q-learning vein, as they primarily concentrate on action-state value-function updates and thus learn an optimal policy indirectly. To be more precise, the algorithm by Jin et al. [2022] is based on a generalized policy improvement principle. They first update the value function (with some optimism additional term); they subsequently enhance the policy using an adversarial bandit mechanism. In the case of a single agent, they obtain bounds scaling as $1/\sqrt{T}$ and close to the optimal sample complexity for episodic settings without discount. Their emphasis lies in the multi-agent context, whereas our attention is directed towards orchestration in a single-agent scenario. Anyhow, we acknowledge the value of a comprehensive numerical comparative analysis between our methods and theirs, that is left for future research.

Additional references and more details on why our approach consists of a third way between policy gradient and value learning may be found in Appendix F.

Notation

We consider a Markov decision process (MDP) with finite state and action spaces $\mathcal{A}$ and $\mathcal{S}$, i.e., a transition kernel $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ and a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}\big([0,1]\big)$, where $\mathcal{P}(\mathcal{X})$ denotes the set of probability distributions over some set $\mathcal{X}$. We denote by $r : \mathcal{S} \times \mathcal{A} \to [0,1]$ the mean-payoff function associated with $\mathcal{R}$, i.e., $r(s,a)$ is the expectation of the distribution $\mathcal{R}(s,a)$, for each $s \in \mathcal{S}$ and $a \in \mathcal{A}$. We set some discount factor $\gamma \in (0,1)$ throughout this article.

A stationary policy π is a mapping $\mathcal{S} \to \mathcal{P}(\mathcal{A})$: we denote by $\pi(\cdot\,|s)$ the probability distribution over actions that it uses in state s. Starting from some initial state s_0, at each round $t \geqslant 0$, such a policy draws an arm $a_t \sim \pi(\cdot\,|s_t)$, after which it obtains a stochastic reward $r_t \sim \mathcal{R}(s_t, a_t)$ while the environment moves to a new state drawn as $s_{t+1} \sim \mathcal{T}(\cdot\,|s_t, a_t)$. The expected infinite-horizon discounted reward accumulated by π starting from s_0 equals, by the tower rule,

$$V_\pi(s_0) = \mathbb{E}^{(s_0,\pi)}\left[\sum_{t=0}^{+\infty} \gamma^t\, r_t\right] = \mathbb{E}^{(s_0,\pi)}\left[\sum_{t=0}^{+\infty} \gamma^t\, r(s_t,a_t)\right].$$

The notation $\mathbb{E}^{(s_0,\pi)}$ indicates that actions are drawn

according to policy π and that the initial state is s_0. The thus defined function V_π is called the value function of π. In this article, we will be interested in designing policies with V–values as large as possible.

Some additional pieces of notation will be used to define our learning strategies. For a given stationary policy π, an initial state s_0 and an action a such that $\pi(a|s_0) > 0$, we define the total reward conditional to the first action a_0 taken by π being equal to a:

$$Q_\pi(s_0, a) = \mathbb{E}^{(s_0,\pi)}\left[\sum_{t=0}^{+\infty} \gamma^t \, r(s_t, a_t) \;\middle|\; a_0 = a\right].$$

We set $Q_\pi(s_0, a)$ in an arbitrary manner otherwise, for instance, $Q_\pi(s_0, a) = 0$. We then define the so-called advantage function as $A_\pi(s_0, a) = Q_\pi(s_0, a) - V_\pi(s_0)$ By the law of total expectation,

$$\sum_{a\in\mathcal{A}} \pi(a|s_0)\, A_\pi(s_0, a) = 0\,. \tag{1}$$

Given a distribution $\nu = (\nu_a)_{a\in\mathcal{A}}$ over actions, we overload notation and define

$$A_\pi(s, \nu) = \sum_{a\in\mathcal{A}} \nu_a \, A_\pi(s, a)\,. \tag{2}$$

In particular, $A_\pi\big(s_0, \pi(\,\cdot\,|s_0)\big) = 0$.

2 SETTING AND OBJECTIVES

We fix a collection Π of $K \geqslant 2$ stationary policies $(\pi_1, \ldots, \pi_K)$, referred to as expert policies. They are indexed by $k \in [K]$, where $[K] = \{1, \ldots, K\}$. We combine these expert policies through state-dependent weights $q = \big(q(\,\cdot\,|s)\big)_{s\in\mathcal{S}} \in \mathcal{P}\big([K]\big)^{\mathcal{S}}$, i.e., each $q(\,\cdot\,|s)$ is a probability distribution over the expert policies. More precisely, we denote by $q\Pi$ the stationary policy

$$q\Pi : s \in \mathcal{S} \longmapsto q\Pi(\,\cdot\,|s) = \sum_{k\in[K]} q(k|s)\, \pi_k(\,\cdot\,|s)\,. \tag{3}$$

Picking an action a' according to $q\Pi(\,\cdot\,|s)$ amounts to performing a two-stage randomization: first, drawing a policy index $k' \sim q(\,\cdot\,|s)$, then drawing $a' \sim \pi_{k'}(\,\cdot\,|s)$. See the steps 1(a) and 1(b) in Box A. We consider the class $\mathcal{C}(\Pi)$ of all such policies:

$$\mathcal{C}(\Pi) = \Big\{q\Pi,\ q \in \mathcal{P}\big([K]\big)^{\mathcal{S}}\Big\},$$

and aim to learn a good policy in this class.

Comment 1. *The first contribution of this article is the modeling of policy orchestration described in this section. It basically consists of considering expert policies as "super-actions", i.e., actions in a lifted MDP.*

Example 1 (no expert policies)**.** *When $K = |\mathcal{A}|$ and the policies $\Delta = (\pi_a)_{a\in\mathcal{A}}$ are given by Dirac masses, i.e., $\pi_a(s) = \delta_a$ for all $s \in \mathcal{S}$, then*

$$\mathcal{C}(\Delta) = \big\{p\Delta,\ p \in \mathcal{P}(\mathcal{A})^{\mathcal{S}}\big\}$$

is the set of all stationary policies, stated in their tabular form via a direct parametrization (following the terminology of Agarwal et al., 2021, Section 3).

A standard result (based on the so-called Bellman's equations) states that there exist optimal stationary policies $\pi^\star : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, i.e., satisfying

$$\forall s_0 \in \mathcal{S}, \qquad V_{\pi^\star}(s_0) = \max_\pi V_\pi(s_0)\,,$$

where the maxima in the display above and below are over all stationary policies π. These optimal stationary policies also satisfy

$$\forall s_0 \in \mathcal{S}, \quad \forall a \in \mathcal{A}, \quad Q_{\pi^\star}(s_0, a) = \max_\pi Q_\pi(s_0, a)\,.$$

We may thus use the shorthand notation $V^\star = V_{\pi^\star}$ and $Q^\star = Q_{\pi^\star}$, and observe that a given stationary policy π is optimal if and only if

$$\forall s \in \mathcal{S}, \quad \mathrm{Supp}\big(\pi(\,\cdot\,|s)\big) \subseteq \operatorname*{argmax}_{\mathcal{A}} Q^\star(s, \,\cdot\,)\,, \tag{4}$$

where Supp denotes the support of a distribution. The standard results recalled right above in the no-expert case and the reduction stated in Comment 1 readily guarantee, in particular, that there exists an optimal policy $q^\star\Pi$ in $\mathcal{C}(\Pi)$, i.e., satisfying

$$\forall s_0 \in \mathcal{S}, \qquad V_{q^\star\Pi}(s_0) = \max_q V_{q\Pi}(s_0)\,,$$

where the maximum is over all $q \in \mathcal{P}\big([K]\big)^{\mathcal{S}}$. The weights $q^\star$ may actually be given by Dirac masses, in view of (4).

The literature is interested (see, among many others, Agarwal et al., 2021, Jin et al., 2022) in constructing sequences of stationary policies $\pi^{(t)}$, where $t \geqslant 1$, such that the sequence of regrets $V^\star(s) - V_{\pi^{(t)}}(s)$ vanishes asymptotically, for all $s \in \mathcal{S}$. Typical bounds are of order $1/\sqrt{t}$.

In this article, we decompose the above total regret into an approximation error, measuring how close in V–values the class $\mathcal{C}(\Pi)$ is from an optimal policy $\pi^\star$, and a regret with respect to $\mathcal{C}(\Pi)$, based on policies of the form $\pi^{(t)} = q_t\Pi$:

$$\begin{aligned} V^\star(s) - V_{q_t\Pi}(s) \\ = \underbrace{V^\star(s) - V_{q^\star\Pi}(s)}_{\text{approximation error}} + \underbrace{V_{q^\star\Pi}(s) - V_{q_t\Pi}(s)}_{\text{regret w.r.t. } \mathcal{C}(\Pi)}\,. \end{aligned} \tag{5}$$

We do so because learning based on K policies can be much more efficient than learning the optimal policy in tabular form when $\mathcal{A}$ is large. We summarize our setting and goal in Box A.

Box A: Aggregation of expert policies

MDP: state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$, reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}\big([0,1]\big)$ with mean-payoff function $r : \mathcal{S} \times \mathcal{A} \to [0,1]$

Expert policies: $\pi_1, \dots, \pi_K : \mathcal{S} \to \mathcal{P}(\mathcal{A})$

Initialization: state s_0, weights $q_0 \in \mathcal{P}\big([K]\big)^{\mathcal{S}}$

For rounds $t = 0, 1, 2, \dots$:

1. The learner
   (a) observes s_t,
   (b) chooses a policy index $k_t \sim q_t(\,\cdot\,|s_t)$,
   (c) picks an action $a_t \sim \pi_{k_t}(\,\cdot\,|s_t)$;
2. The learner gets a reward $r_t \sim \mathcal{R}(s_t, a_t)$, with conditional expectation $r(s_t, a_t)$;
3. The state of the environment is updated according to $s_{t+1} \sim \mathcal{T}(\,\cdot\,|s_t, a_t)$;
4. The learner picks state-dependent weights $q_{t+1} \in \mathcal{P}\big([K]\big)^{\mathcal{S}}$.

Goal: Maximize $s \in \mathcal{S} \mapsto V_{q_T\Pi}(s)$ or equivalently, minimize the regret function

$$s \in \mathcal{S} \longmapsto V_{q^\star\Pi}(s) - V_{q_T\Pi}(s)$$

On the approximation error

The approximation error is of course uniformly null in the situation of Example 1, as $\mathcal{C}(\Pi)$ is exactly the set of all stationary policies therein.

But it may well be that $\mathcal{C}(\Pi)$ is a strict subset of the set of all stationary policies while the approximation error is uniformly null. This is the case, by definition, as soon as $q^\star\Pi$ is an optimal policy, i.e., abides by the characterization (4). We summarize this discussion in the lemma below and draw consequences after its statement.

Lemma 1. *The approximation error $V^\star(s) - V_{q^\star\Pi}(s)$ is null for all $s \in \mathcal{S}$ if and only if*
$\forall s \in \mathcal{S},\ \exists\, w_s \in \mathcal{P}\big([K]\big)$ *s.t.*

$$\mathrm{Supp}\left(\sum_{k\in[K]} w_{s,k}\, \pi_k(\,\cdot\,|s)\right) \subseteq \operatorname*{argmax}_{\mathcal{A}} Q^\star(s,\,\cdot\,)\,.$$

The condition stated means that for each state s, there should exist a convex combination (possibly depending on s) of the distributions of actions recommended by the expert policies at this state s coinciding with the distribution recommended by an optimal stationary policy. This condition is satisfied, in particular, when for each state s, one of the expert policies (but not always the same), coincides with an optimal stationary policy. This observation motivates the use of experts policies for exploration when the state space is large.

3 ADVERSARIAL LEARNING

In this article, we show how strategies designed to control the regret in the so-called adversarial setting, i.e., satisfying guarantees as described in Definition 1 below, may be used to pick weights q_t so as to control the regret w.r.t. $\mathcal{C}(\Pi)$. More precisely, we show how the regret bounds $B_{T,K}$ of adversarial learning may be transferred into $B_{T,K}/T$ bounds on the regret w.r.t. $\mathcal{C}(\Pi)$. This is detailed in the next section and forms the main result of this article.

But for now, we must first briefly recall what the adversarial setting consists in; see the monograph by Cesa-Bianchi and Lugosi [2006a] for a more detailed exposition. At each round $t \geqslant 1$, based on the past, a learning strategy $\varphi = (\varphi_t)_{t\geqslant 1}$ picks a convex combination $w_t = (w_{t,1}, \dots, w_{t,K}) \in \mathcal{P}\big([K]\big)$ while an opponent player picks, possibly at random, a vector $g_t = (g_{t,1}, \dots, g_{t,K})$ of signed rewards. Both w_t and g_t are revealed at the end of the round. By "based on the past", we mean, for the learning strategy, that

$$w_t = \varphi_t\big((w_\tau, g_\tau)_{\tau\leqslant t-1}\big)\,.$$

Definition 1. *A sequential strategy controls the regret in the adversarial setting with rewards bounded by $M > 0$ if against all opponent players sequentially picking reward vectors in $[-M, M]$, for all $T \geqslant 1$,*

$$\max_{k\in[K]} \sum_{t=1}^{T} g_{t,k} - \sum_{t=1}^{T} \sum_{j\in[K]} w_{t,j}\, g_{t,j} \leqslant M\, B_{T,K}\,,$$

where $B_{T,K}/T \to 0$.

The optimal orders of magnitude of $B_{T,K}$ are $\sqrt{T \ln K}$. In Definition 1, the strategy may know M and rely on its value.

The potential-based strategies by Cesa-Bianchi and Lugosi [2003] are defined based on a non-decreasing function $\Phi : \mathbb{R} \to [0, +\infty)$. They resort to $w_{1,k} = 1/K$ and for $t \geqslant 2$,

$$w_{t,k} = \frac{v_{t,k}}{\sum_{j\in[K]} v_{t,j}}\,, \quad \text{where}$$
$$v_{t,k} = \Phi\left(\sum_{\tau=1}^{t-1} g_{\tau,k} - \sum_{\tau=1}^{t-1} \sum_{j\in[K]} w_{t,j} g_{\tau,j}\right). \quad (6)$$

Example 2. *Cesa-Bianchi and Lugosi [2003, Section 2] show that the strategy based on the polynomial potential $\Phi : x \mapsto \big(\max\{x, 0\}\big)^{2\ln K}$ provides the control $B_{T,K} = \sqrt{6T \ln K}$ for the regret in the adversarial setting.*

Example 3. *Another example includes the exponential potential $\Phi(x) = \exp(\eta x)$. Fixed values of η do*

not lead to sublinear adversarial regret bounds but to $B_{T,K} = (\ln K)/\eta + \eta T/2$*; see Cesa-Bianchi and Lugosi [2003, Section 2]. Appendix D discusses the specific case of the exponential potential with constant learning rate* η*, which is out of the scope of the main body of this article.*

Example 4. *However, Auer et al. studied the use of exponential potential with time-varying learning rates* $\eta_t = (1/M)\sqrt{(\ln K)/t}$*, i.e., using* $\Phi_t(x) = \exp(\eta_t x)$ *in* (6) *to define the weights at round* t*. This sequential strategy controls the regret with* $B_{T,K} = \sqrt{T \ln K}$ *in the adversarial setting*

A final example is of a different, not potential-based, nature.

Example 5. *The greedy projection algorithm of Zinkevich [2003] relies on a sequence* $(\eta_t)_{t \geqslant 1}$ *of positive step sizes and sets* $w_{t+1} = \mathrm{proj}\big(w_t + \eta_t\, g_t\big)$ *for* $t \geqslant 1$*, where* $w_1 = (1/K, \ldots, 1/K)$ *and where* proj *is the convex projection onto* $\mathcal{P}\big([K]\big)$ *in Euclidean norm. For the choices* $\eta_t = (1/M)\sqrt{2/(Kt)}$*, it achieves a regret bound of* $M\, B_{T,K} = 3M\sqrt{KT}$ *against sequences of gains in* $[-M, M]$*.*

Appendix A states a property, called monotonicity of weights, satisfied by the strategies stated above (except maybe the one of Example 4 relying on exponential potential with time-varying learning rates).

4 POLICY ORCHESTRATION WITH ORACLE HELP

In this section, we consider an ideal strategy, relying on oracle help, in the form of knowing the advantage functions. This section is meant as a warm-up conveying some main ideas, before the estimation of advantage functions is dealt with in Section 5.

4.1 The strategy and its regret bound

We consider an sequential strategy φ in the adversarial setting and construct the weights q_t as

$$\forall s \in \mathcal{S}, \quad q_t(\,\cdot\,|s) = \varphi_t\Big(\big(q_\tau(\,\cdot\,|s),\, \overline{A}_{q_\tau \Pi}(s,\,\cdot\,)\big)_{\tau \leqslant t-1}\Big)\,,$$

where we use the piece of notation introduced in (2) and define $\overline{A}_{q_\tau \Pi}(s,\,\cdot\,)$ as the vector

$$\big(\overline{A}_{q_\tau \Pi}(s,k)\big)_{k\in[K]} = \Big(A_{q_\tau \Pi}\big(s,\, \pi_k(\,\cdot\,|s)\big)\Big)_{k\in[K]}\,. \quad (7)$$

For the sake of concreteness, an example of this strategy based on potential-based strategies (6) is stated in Box B in Appendix C.

Remark 1. *The exponential potential of Example 2 leads to the so-called natural policy gradient ascent for the softmax parametrization (Agarwal et al., 2021, Section 5.3); see Appendix D.1.*

The theorem below is a "theoretical" result, as it relies on the oracle knowledge of the advantage functions (7) — an issue that we mitigate later in Section 5.

Theorem 1. *If the sequential strategy* φ *controls the regret in the adversarial setting (Definition 1) by* $B_{T,K}$ *then the stationary policies based on the weights* q_t *defined above control the regret w.r.t.* $\mathcal{C}(\Pi)$ *as:*
$\forall s_0 \in \mathcal{S},\ \forall T \geqslant 1,$

$$V_{q^\star \Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T} V_{q_t \Pi}(s_0) \leqslant \frac{B_{T,K}}{(1-\gamma)^2 T}\,.$$

If in addition, φ *satisfies monotonicity of weights (see Property 1 in Appendix A), then*

$$\forall s_0 \in \mathcal{S},\ \forall T \geqslant 1, \quad V_{q^\star \Pi}(s_0) - V_{q_T \Pi}(s_0) \leqslant \frac{B_{T,K}}{(1-\gamma)^2 T}\,.$$

We recall that examples of values of $B_{T,K}$, of optimal order $\sqrt{T \ln K}$, are provided in Examples 2–5.

The bound of Theorem 1 holds with a $1/T$ rate for the case of exponential potential with a constant learning rate η (see Appendix D.1), but this result is not robust to the lack of oracle help, i.e., to the mitigations discussed in Section 5 (see Appendix D.2 for details).

4.2 Proof of Theorem 1

The proof below heavily builds on techniques used in Agarwal et al. [2021, proof of Theorem 16]. To some extend, it can actually be seen as a simplified version of their proof, also extended to other adversarial learning strategies than with exponential potential.

Preparation. *First*, the adversarial guarantees entail that $\forall s \in \mathcal{S}$,

$$\max_{k\in[K]} \sum_{t=1}^{T} \overline{A}_{q_t \Pi}(s,k) - \sum_{t=1}^{T} \overbrace{\sum_{j\in[K]} q_t(j|s)\overline{A}_{q_t \Pi}(s,j)}^{=0}$$
$$= \max_{k\in[K]} \sum_{t=1}^{T} \overline{A}_{q_t \Pi}(s,k) \leqslant \frac{1}{1-\gamma}\, B_{T,K}\,, \quad (8)$$

where the equalities in the display below follow from (1) via the definitions (2)–(3)–(7):

$$\begin{aligned}
&\sum_{j\in[K]} q_t(j|s)\overline{A}_{q_t \Pi}(s,j) \\
&= \sum_{j\in[K]} q_t(j|s) \sum_{a\in[A]} \pi_j(a|s)\, A_{q_t \Pi}(s,a) \\
&= \sum_{a\in[A]} q_t \Pi(a|s)\, A_{q_t \Pi}(s,a) = 0\,. \qquad (9)
\end{aligned}$$

Second, for a given stationary policy π, we introduce, for each $s \in \mathcal{S}$,

$$\mu^{(s_0,\pi)}(s) = (1-\gamma)\sum_{t=0}^{+\infty}\gamma^t\,\mathbb{P}^{(s_0,\pi)}(s_t = s)\,,$$

i.e., $\mu^{(s_0,\pi)}$ is the discounted state visitation distribution starting from s_0 and taking actions drawn by π.

Lemma 2 (performance difference lemma; see Kakade and Langford, 2002a, Lemma 6.1)**.** *For any pair π, π' of stationary policies and all states s_0,*

$$\begin{aligned}V_\pi(s_0) - V_{\pi'}(s_0) &\\ = \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\mu^{(s_0,\pi)}(s)\sum_{a\in\mathcal{A}}\pi(a|s)\,A_{\pi'}(s,a)\,.&\end{aligned}$$

Now, the second building block is the following restatement of the lemma: for any pair $q, q' \in \mathcal{P}\big([K]\big)^{\mathcal{S}}$ of state-dependent weights and all states s_0:

$$\begin{aligned}&V_{q\Pi}(s_0) - V_{q'\Pi}(s_0)\\ &= \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\mu^{(s_0,q\Pi)}(s)\sum_{a\in\mathcal{A}}q\Pi(a|s)\,A_{q'\Pi}(s,a)\\ &= \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\mu^{(s_0,q\Pi)}(s)\sum_{k\in[K]}q(k|s)\,\overline{A}_{q'\Pi}(s,k)\,, \quad (10)\end{aligned}$$

where we used here again the definitions (2)–(3)–(7).

Proof of the first part of Theorem 1. We combine the two building blocks above in the following way. By (10),

$$\begin{aligned}V_{q^\star\Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T}V_{q_t\Pi}(s_0) &= \frac{1}{T}\sum_{t=1}^{T}\big(V_{q^\star\Pi}(s_0) - V_{q_t\Pi}(s_0)\big)\\ = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{1-\gamma}\sum_{s\in\mathcal{S}}\mu^{(s_0,q^\star\Pi)}(s)&\sum_{k\in[K]}q^\star(k|s)\,\overline{A}_{q_t\Pi}(s,k)\,.\end{aligned}$$

We conclude the proof by rearranging, then replacing the weighted sum over k by a maximum, and by finally substituting the adversarial regret bound (8):

$$\begin{aligned}&\frac{1}{(1-\gamma)T}\sum_{s\in\mathcal{S}}\mu^{(s_0,q^\star\Pi)}(s)\sum_{k\in[K]}q^\star(k|s)\sum_{t=1}^{T}\overline{A}_{q_t\Pi}(s,k)\\ &\leqslant \frac{1}{(1-\gamma)T}\sum_{s\in\mathcal{S}}\mu^{(s_0,q^\star\Pi)}(s)\max_{k\in[K]}\sum_{t=1}^{T}\overline{A}_{q_t\Pi}(s,k) \quad (11)\\ &\leqslant \frac{1}{(1-\gamma)^2T}\sum_{s\in\mathcal{S}}\mu^{(s_0,q^\star\Pi)}(s)B_{T,K} = \frac{B_{T,K}}{(1-\gamma)^2T}\,.\end{aligned}$$

Proof of the second part of Theorem 1. See Appendix A.

5 POLICY ORCHESTRATION BASED ON ESTIMATED VALUES

We extend the strategy of Section 4 to work without oracle help, by estimating the advantage functions. To do so, Agarwal et al. [2021] resorted to unbiased but unbounded estimators, which do not naturally lead to high-probability bounds (see detailed discussions in Appendix B.3). We rather resort to biased but bounded estimators, where boundedness is useful both to apply the bound of Definition 1 and to provide high-probability bounds.

The estimators rely on a target precision ε (the bias of the estimators), based on which we set a length of episodic settings H such that $\gamma^H/(1-\gamma) \leqslant \varepsilon$, and on the other hand, an average share κ of the number of states for which we will actually perform estimation. For each given a policy π, starting state s_0, and action a, we independently estimate $Q_\pi(s_0,a)$ by the discounted reward achieved in the first H round of an independent simulation times Z/κ, where Z is a Bernoulli distribution with parameter κ. Based on these estimates of Q–values, we build estimates of the advantage functions. We may extend this scheme from actions to "super-actions" given by expert policies. Details on the estimator construction, on the filtration $\mathcal{F}$ introduced now, and on the lemma below, are provided in Appendix B.1.

We denote by $\widetilde{\mathbb{E}}$ denotes the expectation with respect to auxiliary randomizations and by $\mathcal{F}_{t-1}$ the σ–algebra generated by randomizations used up to round $t-1$ included. As will become clear in (13), we denote by $\widetilde{q}_t$ the weights constructed based on estimates.

Lemma 3. *For all $t \geqslant 1$, the estimators $\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)$ are $\mathcal{F}_t$–measurable and satisfy, on the one hand,*

$$\begin{aligned}&\big|\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\big| \leqslant \frac{1}{\kappa(1-\gamma)} \quad \text{a.s.}\\ \textit{and} \qquad &\sum_{k\in[K]}\widetilde{q}_t(k|s)\,\widetilde{A}_{\widetilde{q}_t\Pi}(s,k) = 0 \quad \text{a.s.}, \quad (12)\end{aligned}$$

and on the other hand,

$$\Big|\widetilde{\mathbb{E}}\big[\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\,\big|\,\mathcal{F}_{t-1}\big] - \overline{A}_{\widetilde{q}_t\Pi}(s,k)\Big| \leqslant \varepsilon \quad \text{a.s.}$$

5.1 Definition and analysis of the strategy

We run a similar learning scheme as in Section 4, but with the $\widetilde{A}_{\widetilde{q}_\tau\Pi}(s,k)$ instead of the $\overline{A}_{q_\tau\Pi}(s,k)$, i.e., $\forall s \in \mathcal{S}$,

$$\widetilde{q}_t(\,\cdot\,|s) = \varphi_t\Big(\big(\widetilde{q}_\tau(\,\cdot\,|s),\,\widetilde{A}_{\widetilde{q}_\tau\Pi}(s,\,\cdot\,)\big)_{\tau\leqslant t-1}\Big)\,, \quad (13)$$

where $\widetilde{A}_{\widetilde{q}_\tau\Pi}(s,\,\cdot\,)$ denotes the vector with components $\widetilde{A}_{\widetilde{q}_\tau\Pi}(s,k)$ for $k \in [K]$. For the sake of concreteness,

an example of this strategy based on potential-based strategies (6) is stated in Box C in Appendix C.

Analysis in expectation. The first two building blocks of the analysis in Section 4.2 relied on deterministic guarantees and readily extend to the learning scheme with estimation. The counterpart of (8) reads

$$\max_{k\in[K]} \sum_{t=1}^{T} \widetilde{A}_{\widetilde{q}_t\Pi}(s,k) - \sum_{t=1}^{T} \overbrace{\sum_{j\in[K]} \widetilde{q}_t(j|s)\widetilde{A}_{\widetilde{q}_t\Pi}(s,j)}^{=0}$$
$$= \max_{k\in[K]} \sum_{t=1}^{T} \widetilde{A}_{\widetilde{q}_t\Pi}(s,k) \leqslant \frac{1}{\kappa(1-\gamma)}\, B_{T,K}\,. \quad (14)$$

The second building block (10) remains as it is: the performance difference lemma is to be stated in terms of the true (not estimated) advantage functions $\overline{A}$.

Theorem 2 (control in expectation)**.** *If the sequential strategy φ controls the regret in the adversarial setting (Definition 1) by $B_{T,K}$ then the stationary policies based on the weights $\widetilde{q}_t$ defined in* (13) *control the regret w.r.t. $\mathcal{C}(\Pi)$ as: $\forall s_0 \in \mathcal{S},\ \forall T \geqslant 1$,*

$$V_{q^\star\Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T} \widetilde{\mathbb{E}}\big[V_{\widetilde{q}_t\Pi}(s_0)\big] \leqslant \frac{\varepsilon}{1-\gamma} + \frac{B_{T,K}}{\kappa(1-\gamma)^2 T}\,.$$

The computational burden of the strategy (13) decreases with ε and $1/\kappa$, while the regret bound increases—as expected.

Proof. By (10), we have

$$V_{q^\star\Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T} V_{\widetilde{q}_t\Pi}(s_0) \quad (15)$$
$$= \frac{1}{(1-\gamma)T} \sum_{s\in\mathcal{S}} \mu^{(s_0,q^\star\Pi)}(s) \sum_{k\in[K]} q^\star(k|s) \sum_{t=1}^{T} \overline{A}_{\widetilde{q}_t\Pi}(s,k)\,.$$

By the tower rule, Lemma 3 implies in particular that

$$-\varepsilon \leqslant \widetilde{\mathbb{E}}\big[\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\big] - \widetilde{\mathbb{E}}\big[\,\overline{A}_{\widetilde{q}_t\Pi}(s,k)\big] \leqslant \varepsilon\,. \quad (16)$$

Therefore, since the quantities $\mu^{(s_0,q^\star\Pi)}(s)$ and $q^\star(k|s)$ are fully deterministic and sum up to 1, taking expectations with respect to $\widetilde{\mathbb{E}}$ in (15) guarantees that

$$V_{q^\star\Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T} \widetilde{\mathbb{E}}\big[V_{\widetilde{q}_t\Pi}(s_0)\big] \leqslant \frac{\varepsilon}{1-\gamma} + \quad (17)$$
$$\widetilde{\mathbb{E}}\left[\frac{1}{(1-\gamma)T} \sum_{s\in\mathcal{S}} \mu^{(s_0,q^\star\Pi)}(s) \sum_{k\in[K]} q^\star(k|s) \sum_{t=1}^{T} \widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\right]$$

By (14), we have, almost surely, for all s:

$$\sum_{k\in[K]} q^\star(k|s) \sum_{t=1}^{T} \widetilde{A}_{\widetilde{q}_t\Pi}(s,k) \leqslant \max_{k\in[K]} \sum_{t=1}^{T} \widetilde{A}_{\widetilde{q}_t\Pi}(s,k)$$
$$\leqslant \frac{1}{\kappa(1-\gamma)}\, B_{T,K}\,. \quad (18)$$

Substituting this inequality in (17) yields

$$V_{q^\star\Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T} \widetilde{\mathbb{E}}\big[V_{\widetilde{q}_t\Pi}(s_0)\big] \leqslant \frac{\varepsilon}{1-\gamma} + \frac{B_{T,K}}{\kappa(1-\gamma)^2 T}\,,$$

as claimed. □

Analysis in high probability. Compared to the expected bound of Theorem 2, the high-probability regret bound adds a factor $2\ln(1/\delta)/\big(\kappa(1-\gamma)^2\sqrt{T}\big)$, which is of the same order of magnitude as the main term $B_{T,K}/\big(\kappa(1-\gamma)^2 T\big)$.

Theorem 3 (high-probability control)**.** *If the sequential strategy φ controls the regret in the adversarial setting (Definition 1) by $B_{T,K}$ then the stationary policies based on the weights $\widetilde{q}_t$ defined in* (13) *control the regret w.r.t. $\mathcal{C}(\Pi)$ as follows: for all $\delta \in (0,1)$, for all $T \geqslant 1$, for all s_0, with probability at least $1-\delta$,*

$$V_{q^\star\Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T} V_{\widetilde{q}_t\Pi}(s_0)$$
$$\leqslant \frac{\varepsilon}{1-\gamma} + \frac{B_{T,K}}{\kappa(1-\gamma)^2 T} + \frac{2\ln(1/\delta)}{\kappa(1-\gamma)^2\sqrt{T}}\,.$$

The detailed proof may be found in Appendix B.2. It basically consists of replacing the T applications of inequality (16) for expectations by an application of the Hoeffding-Azuma lemma. Additional comments on Theorems 2 and 3 may also be found there (e.g., as far as learning schemes based on exponential potentials are concerned, or why we offer results only in terms of Cesàro averages, unlike in Theorem 1).

5.2 Comparison to Agarwal et al. [2021]

A first angle of discussion concerns the regret bounds themselves. On the one hand, the bounds in terms of Cesàro averages are stronger than the form of bounds that Agarwal et al. [2021, Section 6] offered, which corresponds to the left-hand side of this inequality:

$$\widetilde{\mathbb{E}}\Big[\min_{1\leqslant t\leqslant T}\big\{V_{q^\star\Pi}(s_0) - V_{\widetilde{q}_t\Pi}(s_0)\big\}\Big]$$
$$\leqslant V_{q^\star\Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T} \widetilde{\mathbb{E}}\big[V_{\widetilde{q}_t\Pi}(s_0)\big]\,.$$

However, on the other hand, we take a different angle and rely on the decomposition (5): the regret we control is not in terms of $V^\star$ but of $V_{q^\star\Pi}$. Our bounds and the ones by Agarwal et al. [2021, Section 6] are therefore difficult to compare. We advocate that our approach leads to more readable bounds, not relying on rather abstract estimation or transfer errors $\varepsilon_{\text{stat}}$ or $\varepsilon_{\text{bias}}$.

A second angle of discussion concerns the estimators used, see Appendix B.3.

6 SIMULATIONS

We consider a discrete-time version of the stochastic dynamic matching problem described in continuous time in, among others, Comte et al. [2017], Jonckheere et al. [2023]. This type of problem has applications in a wide range of fields, including exchange programs, online marketplaces, and supply-chain management.

Corresponding MDP. We are given I classes of items, indexed by $[I]$; all items in a class are equivalent. At each round $t \geqslant 1$, a new item arrives, of class i_t drawn at random according to a predefined probability distribution $\lambda = (\lambda_i)_{i\in[I]}$, independently from everything else. This item may be either matched with an existing item located in one of the I queues maintained, or placed in queue i_t for later matching, provided that this queue has not reached yet its maximal length L. Otherwise the item is trashed.

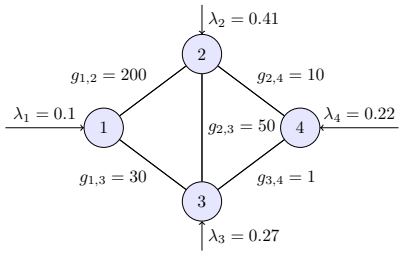
Admissible matchings are indicated by an undirected graph referred to as the compatibility graph: two items can be matched if their classes are connected in this graph. We denote by $\varrho_t = (\varrho_{t,i})_{i\in[I]}$ the number of unmatched items in each class queue at the beginning of round t, before the new item is taken care of. States $s_t = (\varrho_t, i_t)$ are composed of the lengths of the queues and of the index of the item to be taken care of. We start with initially fully empty queues and a random index; we denote by μ_0 the corresponding distribution over states.

The corresponding action space $\mathcal{A}$, state space $\mathcal{S}$, transition kernel $\mathcal{T}$, and deterministic reward function r, i.e., the corresponding MDP, is formally defined in Appendix E.1. It relies on some parameters that we will vary in the simulations (e.g., the compatibility graph, λ, and L described above, but also other parameters, like the penalty for the queue lengths). Each set of parameters will be referred to as a scenario.

Expert policies. The first expert policy π_1 is called "match the longest": if at least one match is possible, this policy always chooses the class with the most items in its queue (ties broken based on the payoffs). The other policies are of "edge-priority" type and select matches according to some (lack of) intrinsic priority order defined on the edges of the compatibility graph. If at least one match is possible, the expert policy π_2 chooses the match leading to the largest payoff (ties broken based on queue lengths). Similarly, the expert policy π_3 randomly selects at each round an edge among the possible matches. Sometimes, we will consider a final the expert policy π_4, picking some random priority order over $[I]$ at the time of its creation, and keeping it for all simulations (this is just a way to avoid figuring out an arbitrary order by ourselves when I is large). Otherwise, if no match is possible, all expert policies described above add the item to its queue, if the maximal length L of the latter is not achieved yet; and in last resort, they trash the item. These policies are also formally defined in Appendix E.1.

Results. For the sake of space, we provide only the outcome for one simple scenario. Other (more complex) scenarios are to be found in Appendix E and are presented in a similar way.

The scenario corresponds to $I = 4$ with the following compatibility graph, and $L = 5$.



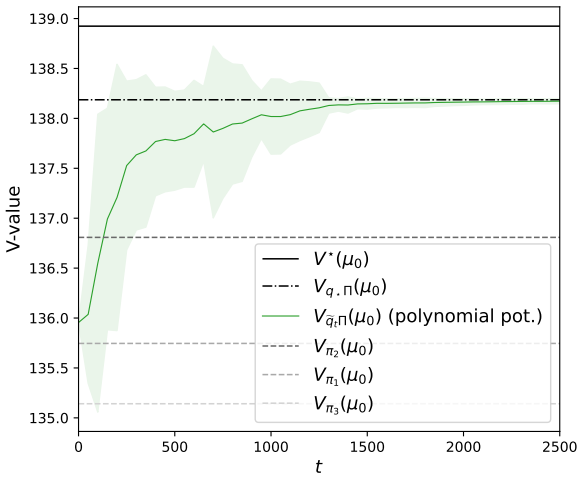
The values achieved based solely on the first three expert policies are stated below; they show that $q_\star\Pi$ is an interesting policy to target.

$V_{\pi_1}(\mu_0)$	$V_{\pi_2}(\mu_0)$	$V_{\pi_3}(\mu_0)$	$V_{q_\star\Pi}(\mu_0)$	$V^\star(\mu_0)$
135.74	136.80	135.14	138.18	138.92

As detailed in Appendix E, we run N times the learning schemes of Section 5, where the n-th run gives rise to weights $\widetilde{q}_{n,t}$ indexed by n, and report in solid lines

$$t \longmapsto \frac{1}{N}\sum_{n=1}^{N} V_{\widetilde{q}_{n,t}\Pi}(\mu_0)\,,$$

as well as the corresponding $\pm 2\times$ standard errors in shaded areas. We obtain the following picture with $N = 20$ and the learning scheme based on polynomial potential with $p = 3$; it achieves the target $V_{q_\star\Pi}(\mu_0)$ set, up to the bias ε. The performance of other learning schemes is discussed in the appendix.



References

A. Agarwal, S.M. Kakade, J.D. Lee, and G. Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Alekh Agarwal, Nan Jiang, and Sham M. Kakade. Reinforcement learning: Theory and algorithms. 2019.

P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75.

Daniel G. Brown. How I wasted too long finding a concentration inequality for sums of geometric variables, 2011. University of Waterloo repository — UWSpace, `http://hdl.handle.net/10012/17210`.

N. Cesa-Bianchi and G. Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51:239–261, 2003.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006a.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006b. doi: 10.1017/CBO9780511546921.

Céline Comte, Fabien Mathieu, and Thomas Bonald. Performance of balanced fairness in resource pools: A recursive approach. *arXiv preprint arXiv:1711.02880*, 2017.

Céline Comte, Fabien Mathieu, and Ana Bušić. Stochastic dynamic matching: A mixed graph-theory and linear-algebra approach. *arXiv preprint arXiv:2112.14457*, 2021.

Heriberto Cuayáhuitl, Donghyeon Lee, Seonghan Ryu, Yongjin Cho, Sungja Choi, Satish Indurthi, Seunghak Yu, Hyungtak Choi, Inchul Hwang, and Jihie Kim. Ensemble-based deep reinforcement learning for chatbots. *Neurocomputing*, 366:118–130, 2019.

Steven de Rooij, Tim van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15(37):1281–1316, 2014.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International conference on machine learning*, pages 1467–1476. PMLR, 2018.

John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

Alexander Hans and Steffen Udluft. Ensembles of neural networks for robust reinforcement learning. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 401–406. IEEE, 2010.

S.M.R. Iravani, M. Posner, and John. Buzacott. A two-stage tandem queue attended by a moving server with holding and switching costs. *Queueing Syst.*, 1997.

Ju Jiang. A framework for aggregation of multiple reinforcement learning algorithms. 2007.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning–a simple, efficient, decentralized algorithm for multiagent RL. In *ICLR*, 2022.

Matthieu Jonckheere, Pascal Moyal, Claudia Ramírez, and Nahuel Soprano-Loto. Generalized max-weight policies in stochastic matching. *Stochastic Systems*, 13(1):40–58, 2023.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML'02)*, pages 267–274, 2002a.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 267–274, 2002b.

Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

ATD Perera and Parameswaran Kamalaruban. Applications of reinforcement learning in energy systems. *Renewable and Sustainable Energy Reviews*, 137:110618, 2021.

Yichen Qian, Jun Wu, Rui Wang, Fusheng Zhu, and Wei Zhang. Survey on reinforcement learning applications in communication networks. *Journal of Communications and Information Networks*, 4(2): 30–39, 2019. doi: 10.23919/JCIN.2019.8917870.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Yanjie Song, PN Suganthan, Witold Pedrycz, Junwei Ou, Yongming He, and Yingwu Chen. Ensemble reinforcement learning: A survey. *arXiv preprint arXiv:2303.02618*, 2023.

P Whittle. Partial balance and insensitivity. *Journal of Applied Probability*, pages 168–176, 1985.

Marco A Wiering and Hado Van Hasselt. Ensemble algorithms in reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(4):930–936, 2008.

Lin Xiao. On the convergence rates of policy gradient methods. *The Journal of Machine Learning Research*, 23(1):12887–12922, 2022.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML'03)*, pages 928–936, 2003.

Supplementary material for

"Symphony of experts: orchestration with adversarial insights in reinforcement learning"

This appendix contains the following sections:

- Appendix A states and proves some monotonicity property alluded at in the second part of the statement of Theorem 1 in Section 4.
- Appendix B provides all details required for our main results stated in Section 5, e.g., among others, the precise definition of the estimators used and a proof of Theorem 3.
- Appendix C states more concrete examples of the policy learning schemes considered in Sections 4 and 5, based on potential-based adversarial strategies.
- Appendix D discusses the specific case of the exponential potential $\Phi : x \mapsto \exp(\eta x)$, for which a $1/T$ value regret bound may be proved with oracle help, but for which it seems difficult to extend the analysis beyond oracle help (we explain why this is so).
- Appendix E describes formally the simulation setting considered in Section 6 and provides all details on implementation, as well as a thorough exposition of the results achieved.
- Appendix F provides additional elements for the literature review, mostly on policy orchestration.

A AROUND THE MONOTONICITY PROPERTY OF AN ADVERSARIAL STRATEGY

In this appendix, we first define the monotonicity property of an adversarial strategy (in Appendix A.1) and then prove the second part of Theorem 1 in Appendix A.2), which relies on this property.

A.1 The monotonicity property: definition and examples

Property 1 (monotonicity of weights)**.** *For all sequences of rewards, for all $t \geqslant 1$, the convex weights output are such that*

$$\sum_{k\in[K]} w_{t+1,k}\left(g_{t,k} - \sum_{j\in[K]} w_{t,j}g_{t,j}\right) \geqslant 0\,.$$

We now prove that it is satisfied by many adversarial strategies stated in Section 3.

Lemma 4. *The potential-based strategies* (6) *enjoy monotonicity of weights in the sense of Property 1.*

Proof. Indeed, with the notation of Example 2, since Φ is non-decreasing, we have that

$$v_{t+1,k} \geqslant v_{t,k} \quad \text{if} \quad g_{t,k} - \sum_{j\in[K]} w_{t,j}g_{t,j} \geqslant 0$$

and $v_{t+1,k} \leqslant v_{t,k}$ if this difference is $\leqslant 0$. Therefore,

$$\sum_{k\in[K]} v_{t+1,k}\left(g_{t,k} - \sum_{j\in[K]} w_{t,j}g_{t,j}\right) \geqslant \sum_{k\in[K]} v_{t,k}\left(g_{t,k} - \sum_{j\in[K]} w_{t,j}g_{t,j}\right) = 0\,.$$

The result follows by normalization. □

The calculation above shows that the property is natural as it is satisfied as soon as weights for components k associated with a good (respectively, bad) reward in the previous round increase (respectively, decrease). This is why we termed it monotonicity of weights.

Lemma 5. *The greedy projection algorithm of Example 5 satisfies monotonicity of weights in the sense of Property 1.*

Proof. Indeed, by a property of Euclidean projection onto a convex set, the following inner product is non-positive:

$$0 \geqslant \big\langle w_t - w_{t+1},\ (w_t + \eta_t\, g_t) - w_{t+1}\big\rangle = \|w_t - w_{t+1}\|^2 + \eta_t\langle w_t - w_{t+1},\, g_t\rangle\,,$$

so that $\langle w_{t+1} - w_t,\, g_t\rangle \geqslant 0$, which is exactly monotonicity of weights. □

A.2 Proof of the second part of Theorem 1

The monotonicity property of the weights (see above) and the equality (9) entail that for all $t \geqslant 1$ and $s \in \mathcal{S}$,

$$\sum_{j\in[K]} q_{t+1}(j|s)\overline{A}_{q_t\Pi}(s,j) \geqslant \sum_{j\in[K]} q_t(j|s)\overline{A}_{q_t\Pi}(s,j) = 0\,. \tag{19}$$

As a consequence of (10) and (19), we have, for all $t \geqslant 1$ and s_0,

$$V_{q_{t+1}\Pi}(s_0) - V_{q_t\Pi}(s_0) \geqslant 0\,, \tag{20}$$

so that

$$V_{q^\star\Pi}(s_0) - V_{q_T\Pi}(s_0) \leqslant \frac{1}{T}\sum_{t=1}^{T}\big(V_{q^\star\Pi}(s_0) - V_{q_t\Pi}(s_0)\big)\,.$$

This proves the second part of Theorem 1 based on its first part.

B OMITTED PROOFS IN THE ANALYSIS OF SECTION 5

We provide details on several statements made in Section 5, and more precisely,

- we define and study in a rigorous way the estimators of Q–values and advantage functions defined therein (in Appendix B.1);
- we provide a proof of Theorem 3 and comments on Theorems 2 and 3 (in Appendix B.2);
- we compare the merits of our estimation scheme to the one considered by Agarwal et al. [2021, Section 6] (in Appendix B.3).

B.1 Definition and properties of estimators

We recall that the estimators rely on two parameters: on the one hand, a target precision ε (the bias of the estimators), based on which we set a length of episodic settings H such that $\gamma^H/(1-\gamma) \leqslant \varepsilon$, and on the other hand, an average share κ of the number of states for which we will actually perform estimation.

We start by explaining how we use κ. We consider a family $(Z_{t,s})_{t \geqslant 1,\, s \in \mathcal{S}}$ of i.i.d. variables, independent from everything else, distributed according to $\mathrm{Ber}(\kappa)$, the Bernoulli distribution with parameter κ. At each round $t \geqslant 1$, we perform the estimation of the advantage function only for those states s for which $Z_{t,s} = 1$.

To estimate the advantage function, it is necessary and sufficient to estimate the Q–values. Given a policy π, a starting state s_0, and an action a, we estimate $Q_\pi(s_0, a)$ as follows, based on an auxiliary randomization $U_{\pi,s_0,a}$, e.g., generated independently from everything else according to a uniform distribution over $[0,1]$:

1. We set $s'_0 = s_0$ and $a'_0 = a$, and we iteratively simulate $s'_\tau \sim T(\,\cdot\,|s'_{\tau-1}, a'_{\tau-1})$, then $a'_\tau \sim \pi(\,\cdot\,|s'_\tau)$, and finally $r'_\tau \sim \mathcal{R}(s'_\tau, a'_\tau)$, for $1 \leqslant \tau \leqslant H-1$. All these simulations are based on the value of $U_{\pi,s_0,a}$.
2. We define $\quad \widetilde{Q}_\pi(s_0,a) = \displaystyle\sum_{\tau=0}^{H-1} \gamma^\tau r'_\tau$.

We recall from Section 5 that the expectation with respect to auxiliary randomizations like $U_{\pi,s_0,a}$ here (or the $Z_{t,s}$ defined above) is denoted by $\widetilde{\mathbb{E}}$. We note that by the very simulation design and the tower rule,

$$\widetilde{\mathbb{E}}\big[\widetilde{Q}_\pi(s_0,a)\big] = \mathbb{E}^{(s_0,\pi)}\left[\sum_{\tau=0}^{H-1} \gamma^\tau\, r(s_\tau, a_\tau) \;\middle|\; a_0 = a\right]$$

and thus,

$$0 \leqslant Q_\pi(s_0,a) - \widetilde{\mathbb{E}}\big[\widetilde{Q}_\pi(s_0,a)\big] = \mathbb{E}^{(s_0,\pi)}\left[\sum_{\tau=H}^{+\infty} \gamma^\tau\, r(s_\tau, a_\tau) \;\middle|\; a_0 = a\right] \leqslant \frac{\gamma^H}{1-\gamma} \leqslant \varepsilon\,. \tag{21}$$

Put differently, $\widetilde{Q}_\pi(s_0,a)$ is a pessimistic estimator, with bias smaller than ε. We also note that by definition, $\widetilde{Q}_\pi(s_0,a)$ lies in the bounded range $\big[0,\, 1/(1-\gamma)\big]$.

We move to the estimation of advantage functions as follows. In rounds $t \geqslant 1$, for states $s \in \mathcal{S}$ such that $Z_{t,s} = 1$, we estimate $\overline{A}_{\widetilde{q}_t\Pi}(s,k)$ by

$$\widetilde{A}_{\widetilde{q}_t\Pi}(s,k) = \frac{Z_{t,s}}{\kappa} \sum_{a \in \mathcal{A}} \pi_k(a|s) \left(\widetilde{Q}_{\widetilde{q}_t\Pi}(s,a) - \sum_{a' \in \mathcal{A}} \widetilde{q}_t\Pi(a'|s)\, \widetilde{Q}_{\widetilde{q}_t\Pi}(s,a') \right), \tag{22}$$

and we set $\widetilde{A}_{\widetilde{q}_t\Pi}(s,k) = 0$ when $Z_{t,s} = 0$. Actually, we may use the first closed-form definition for $\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)$ whether $Z_{t,s} = 1$ or $Z_{t,s} = 0$.

We denote by

$$\mathcal{F}_{t-1} = \sigma\big((Z_{\tau,s})_{\tau \leqslant t-1,\, s \in \mathcal{S}},\, (U_{\widetilde{q}_\tau\Pi,s,a})_{\tau \leqslant t-1,\, s \in \mathcal{S},\, a \in \mathcal{A}}\big)$$

the σ–algebra generated by past randomizations. The second part of Lemma 3 (restated below for the convenience of the reader) follows easily from the independence between $\mathcal{F}_{t-1}$, one the one hand, and the variables $(Z_{t,s})_{s\in\mathcal{S}}$ and $(U_{\widetilde{q}_t\Pi,s,a})_{s\in\mathcal{S},a\in\mathcal{A}}$ on the other hand: the conditional expectation considered in this second part corresponds to integrating over the $(Z_{t,s})_{s\in\mathcal{S}}$ and the $(U_{\widetilde{q}_t\Pi,s,a})_{s\in\mathcal{S},a\in\mathcal{A}}$. The first part of Lemma 3 is by design—as in (9) for the equality (12). We recall once again that $\widetilde{\mathbb{E}}$ denotes the expectation with respect to the auxiliary randomizations used.

Lemma 3. *For all $t \geqslant 1$, the estimators $\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)$ are $\mathcal{F}_t$–measurable and satisfy, on the one hand,*

$$\big|\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\big| \leqslant \frac{1}{\kappa(1-\gamma)} \quad \text{a.s.}$$
$$\text{and} \qquad \sum_{k\in[K]} \widetilde{q}_t(k|s)\, \widetilde{A}_{\widetilde{q}_t\Pi}(s,k) = 0 \quad \text{a.s.}, \tag{12}$$

and on the other hand,

$$\Big|\widetilde{\mathbb{E}}\big[\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\,\big|\,\mathcal{F}_{t-1}\big] - \overline{A}_{\widetilde{q}_t\Pi}(s,k)\Big| \leqslant \varepsilon \quad \text{a.s.}$$

B.2 Proof of Theorem 3 and comments on Theorems 2 and 3

Theorem 3 may be proved by adapting the proof of Theorem 2, from which we extract two (in)equalities.

Proof. We use again the (in)equalities (15) and (18), which hold with probability 1, and based on Lemma 3, we therefore only need to explain why, with probability at least $1-\delta$,

$$\sum_{s\in\mathcal{S}} \mu^{(s_0,q^\star\Pi)}(s) \sum_{k\in[K]} q^\star(k|s) \sum_{t=1}^{T} \widetilde{\mathbb{E}}\big[\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\,\big|\,\mathcal{F}_{t-1}\big]$$
$$\leqslant \sum_{s\in\mathcal{S}} \mu^{(s_0,q^\star\Pi)}(s) \sum_{k\in[K]} q^\star(k|s) \sum_{t=1}^{T} \widetilde{A}_{\widetilde{q}_t\Pi}(s,k) + \frac{1}{\kappa(1-\gamma)}\sqrt{2T\ln\frac{1}{\delta}}\,.$$

The inequality above indeed follows from the Hoeffding-Azuma lemma, applied to the martingale difference sequence

$$X_t = \sum_{s\in\mathcal{S}} \mu^{(s_0,q^\star\Pi)}(s) \sum_{k\in[K]} q^\star(k|s) \Big(\widetilde{A}_{\widetilde{q}_t\Pi}(s,k) - \widetilde{\mathbb{E}}\big[\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\,\big|\,\mathcal{F}_{t-1}\big]\Big)\,,$$

whose increments are bounded by $2/\big(\kappa(1-\gamma)\big)$. □

Comment 2. *We were unable to provide an analysis for the exponential potential of Example 3 with a constant learning rate, see the difficulties discussed in Appendix D. However, the sequential strategy φ of Example 3, i.e., exponential potential with time-varying learning rates, satisfies the assumptions of Theorems 2 and 3.*

Comment 3. *We only control Cesàro averages because we were unable to show inequalities like*

$$\widetilde{\mathbb{E}}\big[V_{\widetilde{q}_{t+1}\Pi}(s_0)\big] - \widetilde{\mathbb{E}}\big[V_{\widetilde{q}_t\Pi}(s_0)\big] \geqslant -\varepsilon$$

To mimic the analysis of Section 4.2, we would apply (10) for the equality below and Lemma 3 for the inequality:

$$(1-\gamma)\big(V_{\widetilde{q}_{t+1}\Pi}(s_0) - V_{\widetilde{q}_t\Pi}(s_0)\big) = \sum_{s\in\mathcal{S}} \mu^{(s_0,\widetilde{q}_{t+1}\Pi)}(s) \sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \overline{A}_{\widetilde{q}_t\Pi}(s,k)$$
$$\geqslant -\varepsilon + \sum_{s\in\mathcal{S}} \mu^{(s_0,\widetilde{q}_{t+1}\Pi)}(s) \sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \widetilde{\mathbb{E}}\big[\widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\,\big|\,\mathcal{F}_{t-1}\big]\,.$$

Now, monotonicity of weights (Property 1), together with (12), guarantees

$$\sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \widetilde{A}_{\widetilde{q}_t\Pi}(s,k) \geqslant 0 \qquad \text{a.s.}\,,$$

but the quantity to control involves some conditional expectation.

B.3 On the estimators of Agarwal et al. [2021, Section 6]

This section may be summarized as opposing "bounded but biased estimators" (the estimation scheme of this work) vs. "unbiased but unbounded estimators" (the estimation scheme considered in Agarwal et al. [2021, Section 6]).

As noted by Agarwal et al. [2021, Section 6], the key for a practical implementation of aggregation of RL policies is the ability to estimate the value functions $Q_\pi : \mathcal{S} \times \mathcal{A} \to [0,1]$ for any policy π. We proposed an estimation scheme in (21) that leads to bounded and biased estimators, but we illustrated in Theorems 2 and 3 that the bias is not too much of an issue, as it can be taken of the same order of magnitude as the rest of the regret bound. The boundedness of the estimators allows for high-probability bounds via the classic Hoeffding-Azuma deviation bound.

Agarwal et al. [2021, Section 6] rather suggested unbounded but unbiased estimators of the following kind: given a policy π, a starting state s_0, and an action a,

1. draw a random variable according to a geometric distribution, $G \sim \mathrm{Geom}(1-\gamma)$, i.e., $\mathbb{P}(G = h) = (1-\gamma)\,\gamma^h$ for all $h \in \{0, 1, 2, \ldots\}$;
2. set $s'_0 = s_0$ and $a'_0 = a$, and if $G \geqslant 1$, iteratively simulate $s'_\tau \sim T(\cdot\,|s'_{\tau-1}, a'_{\tau-1})$, then $a'_\tau \sim \pi(\cdot\,|s'_\tau)$, and finally $r'_\tau \sim \mathcal{R}(s'_\tau, a'_\tau)$, for $1 \leqslant \tau \leqslant G$;
3. define $\displaystyle \widehat{Q}_\pi(s_0, a) = \sum_{\tau=0}^{G} r'_\tau$.

Of course, an additional layer of Bernoulli randomization via the variables $Z_{t,s}/\kappa$ could be added to reduce the computational burden, but this is unimportant for the discussion below.

The estimators thus defined are unbiased. Indeed, with the same arguments as in Appendix B.1 and still denoting by $\widetilde{\mathbb{E}}$ the expectation with respect to the auxiliary randomizations used, we note that by the very simulation design, by repeated applications of the tower rule, and by exchanging the sums overs h and τ,

$$\begin{aligned}
\widetilde{\mathbb{E}}\big[\widehat{Q}_\pi(s_0,a)\big] = \mathbb{E}^{(s_0,\pi)}\left[\sum_{\tau=0}^{G} r(s_\tau,a_\tau) \;\middle|\; a_0 = a\right] &= \mathbb{E}^{(s_0,\pi)}\left[\sum_{h=0}^{+\infty}(1-\gamma)\,\gamma^h \sum_{\tau=0}^{h} r(s_\tau,a_\tau) \;\middle|\; a_0 = a\right] \\
&= \mathbb{E}^{(s_0,\pi)}\left[\sum_{\tau=0}^{+\infty}(1-\gamma)\,\gamma^\tau r(s_\tau,a_\tau) \sum_{h=\tau}^{+\infty} \gamma^{h-\tau} \;\middle|\; a_0 = a\right] \\
&= \mathbb{E}^{(s_0,\pi)}\left[\sum_{\tau=0}^{+\infty} \gamma^\tau r(s_\tau,a_\tau)\right] = Q_\pi(s_0,a)\,.
\end{aligned}$$

The estimators $\widehat{Q}_\pi(s_0,a)$ are unfortunately unbounded, which raises two issues when extending the analysis conducted in Section 5.1 to strategies of the form (13) but relying instead on the estimators $\widehat{Q}_\pi(s_0,a)$.

First, the application (14) of the underlying adversarial guarantees needs to be adapted, by assuming sharper bounds than the uniform bounds considered in Definition 1 (which are called zero-order bounds, as they only depend on the range). These sharper bounds could be second-order bounds, involving quantities of the form $g_{t,j}^2$, with the notation of Definition 1. Transferring these adversarial regret bounds to regret bounds for aggregation of policies would certainly only require ensuring that the (conditional) second-order moments of the $\widehat{Q}_\pi(s_0,a)$ are bounded, which is the case.

Second, as far as high-probability bounds are concerned, as in Theorem (3), classic deviation inequalities for sums of martingale differences like the Hoeffding-Azuma inequality or even Bernstein's inequality are not applicable. However, as means of n i.i.d. geometric distributions enjoy $1/\sqrt{n}$ deviations bounds (see, e.g., Brown, 2011), and since the $\widehat{Q}_{\widetilde{q}_t\Pi}(s,a)$ are dominated (as t varies, for fixed s and s) by a sequence of i.i.d. random variables with distribution $\mathrm{Geom}(1-\gamma)$, suitable deviation results may be exhibited.

All in all, while we feel that regret guarantees (in expectation and in high-probability) could be proved for strategies of the form (13) but relying instead on the estimators $\widehat{Q}_\pi(s_0,a)$, we deem such an analysis more involved than the straightforward analysis we presented in Section 5.1, which relies on simple tools like the Hoeffding-Azuma inequality.

C EXAMPLES OF DESCRIPTIONS OF THE POLICY LEARNING SCHEMES CONSIDERED IN SECTIONS 4 AND 5

We describe in Boxes B and C, respectively, the policy learning schemes discussed in Sections 4 and 5, respectively, based on potential-based adversarial learning strategies, i.e., of the form (6). These descriptions are examples only, as other adversarial learning strategies may be considered, like the greedy projection algorithm of Zinkevich [2003] (see Example 5).

These descriptions relying on potential-based adversarial learning strategies take simple forms due to, respectively, equalities (9) and (12), which we restate here: for all $t \geqslant 0$,

$$\sum_{j\in[K]} q_t(j|s)\overline{A}_{q_t\Pi}(s,j) = 0 \qquad \text{and} \qquad \sum_{j\in[K]} \widetilde{q}_t(j|s)\,\widetilde{A}_{\widetilde{q}_t\Pi}(s,j) = 0 \quad \text{a.s.}$$

(The Box C statement is to be found on the next page.)

Box B: Potential-based strategy with oracle help

Setting: described in Box A

Parameter: a non-decreasing function $\Phi : \mathbb{R} \to [0, +\infty)$

Initialization: weights $q_0(\cdot\,|s) = (1/K, \ldots, 1/K)$ for all $s \in \mathcal{S}$

For rounds $t = 0, 1, 2, \ldots$:

1. The learner picks a policy index $k_t \sim q_t(\cdot\,|s_t)$, draws an action $a_t \sim \pi_{k_t}(\cdot\,|s_t)$, and gets a reward $r_t \sim \mathcal{R}(s_t, a_t)$;
2. The state of the environment is updated according to $s_{t+1} \sim T(\cdot\,|s_t, a_t)$;
3. An oracle reveals the value function $Q_{q_t\Pi} : \mathcal{S} \times \mathcal{A} \to [0, 1]$;
4. The learner deduces the advantage function $A_{q_t\Pi} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ for actions, and computes the advantage function $\overline{A}_{q_t\Pi} : \mathcal{S} \times [K] \to [0, 1]$ for the expert policies as
$$\overline{A}_{q_t\Pi}(s,k) = A_{q_t\Pi}\big(s,\, \pi_k(\cdot\,|s)\big) = \sum_{a\in\mathcal{A}} \pi_k(a|s)\, A_{q_t\Pi}(s,a)\,;$$
5. The learner picks the weights, for all $s \in \mathcal{S}$, $$q_{t+1}(k|s) = \frac{\Phi\left(\sum_{\tau=0}^{t} \overline{A}_{q_\tau\Pi}(s,k)\right)}{\sum_{j\in[K]} \Phi\left(\sum_{\tau=0}^{t} \overline{A}_{q_\tau\Pi}(s,j)\right)}\,.$$

Box C: Potential-based strategy estimating the advantage functions

Setting: described in Box A

Parameters: a non-decreasing function $\Phi : \mathbb{R} \to [0, +\infty)$, an estimation rate $\kappa \in (0, 1]$, the number H of simulation steps for estimation

Initialization: weights $\widetilde{q}_0(\cdot\,|s) = (1/K, \ldots, 1/K)$ for all $s \in \mathcal{S}$

For rounds $t = 0, 1, 2, \ldots$:

1. The learner picks a policy index $k_t \sim \widetilde{q}_t(\cdot\,|s_t)$, draws an action $a_t \sim \pi_{k_t}(\cdot\,|s_t)$, and gets a reward $r_t \sim \mathcal{R}(s_t, a_t)$;

2. The state of the environment is updated according to $s_{t+1} \sim T(\cdot\,|s_t, a_t)$;

3. The learner estimates the value function $\widetilde{Q}_{\widetilde{q}_t\Pi} : \mathcal{S} \times \mathcal{A} \to [0, 1]$ as follows:
   For each $s \in \mathcal{S}$, the learner draws $Z_{t,s} \sim \mathrm{Ber}(\kappa)$:
   - if $Z_{t,s} = 0$, then $\widetilde{Q}_{\widetilde{q}_t\Pi}(s, a) = 0$ for all $a \in \mathcal{A}$;
   - if $Z_{t,s} = 1$, then, **For** each $a \in \mathcal{A}$,
     (a) starting from $s'_0 = s$ and $a'_0 = a$, the learner simulates H steps of the MDP by drawing actions according to $\widetilde{q}_t\Pi$, and gets corresponding rewards $r'_0, \ldots, r'_{H-1}$;
     (b) the learner sets $\widetilde{Q}_{\widetilde{q}_t\Pi}(s, a) = \displaystyle\sum_{\tau=0}^{H-1} \gamma^\tau r'_\tau$;

4. The learner computes the estimated advantage function $\widetilde{A}_{\widetilde{q}_t\Pi} : \mathcal{S} \times [K] \to [0, 1]$ for the expert policies as: for all $s \in \mathcal{S}$ and $k \in [K]$,

$$\widetilde{A}_{\widetilde{q}_t\Pi}(s, k) = \frac{Z_{t,s}}{\kappa} \sum_{a \in \mathcal{A}} \pi_k(a|s) \left(\widetilde{Q}_{\widetilde{q}_t\Pi}(s, a) - \sum_{b \in \mathcal{A}} \widetilde{q}_t\Pi(b|s)\, \widetilde{Q}_{\widetilde{q}_t\Pi}(s, b) \right) ;$$

5. The learner picks the weights, for all $s \in \mathcal{S}$,
$$\widetilde{q}_{t+1}(k|s) = \frac{\Phi\left(\displaystyle\sum_{\tau=0}^{t} \widetilde{A}_{\widetilde{q}_\tau\Pi}(s, k) \right)}{\displaystyle\sum_{j \in [K]} \Phi\left(\sum_{\tau=0}^{t} \widetilde{A}_{\widetilde{q}_\tau\Pi}(s, j) \right)} .$$

D EXPONENTIAL POTENTIAL WITH CONSTANT LEARNING RATE η

In this appendix, we discuss the specific case of the exponential potential $\Phi : x \mapsto \exp(\eta x)$ with constant learning rate $\eta > 0$; see Example 2. The updates 6 take the simplified form

$$w_{t,k} = \frac{v_{t,k}}{\sum_{j\in[K]} v_{t,j}}\,, \qquad \text{where} \qquad v_{t,k} = \exp\left(\eta \sum_{\tau=1}^{t-1} g_{\tau,k}\right).$$

The associated policy learning scheme of Section 4, i.e., with oracle help, corresponds exactly to the so-called natural policy gradient ascent for the softmax parametrization (as underlined in Remark 1; see Agarwal et al., 2021, Section 5.3).

We prove in Section D.1 that in the oracle case, it is associated with a regret bound of order $1/T$ for the counterpart of Theorem 1, but point out in Section D.2 the difficulties encountered by replacing oracle calls with estimation. For such constant learning rates, we were unable to extend the results even only in expectation like the ones of Theorem 2. Of course, Theorems 2 and 3 hold however for the strategy using the exponential potential with time-varying rates η_t, as described in Example 3.

D.1 With oracle help

The result stated below is extracted from Agarwal et al. [2021, Section 5.3]. We provide its proof for the sake of completeness: this proof is exactly the one provided in Agarwal et al. [2021, Section 5.3], we merely change its exposition.

Theorem 4 (Agarwal et al. [2021, Section 5.3])**.** *The Box B strategy run with*

$$q_{t+1}(k|s) = \frac{\exp\left(\eta \sum_{\tau=0}^{t} \overline{A}_{q_\tau\Pi}(s,k)\right)}{\sum_{j\in[K]} \exp\left(\eta \sum_{\tau=0}^{t} \overline{A}_{q_\tau\Pi}(s,j)\right)}$$

controls the regret with respect to $\mathcal{C}(\Pi)$ as:

$$\forall s_0 \in \mathcal{S}, \quad \forall T \geqslant 1, \qquad V_{q^\star\Pi}(s_0) - V_{q_T\Pi}(s_0) \leqslant \frac{\ln K}{\eta(1-\gamma)T} + \frac{1}{(1-\gamma)^2 T}\,.$$

Proof. We follow the general analysis of Section 4.2 till (11) included, and perform sharper bounds instead of the last step.

Part 1 — We exhibit some adversarial regret bound, in the form of "pre-regret" bound, instead of a uniform $B_{T,K}$ bound (of the same flavor as the bounds by de Rooij et al., 2014, Section 2 in terms of so-called mixability gaps). First, by bounding "à la Pisier", for all sequences of payoffs $g_{t,j}$, possibly signed and unbounded:

$$\begin{aligned}\max_{k\in[K]} \sum_{t=1}^{T} g_{t,k} &= \frac{1}{\eta} \ln\left(\max_{j\in[K]} \exp\left(\eta \sum_{t=1}^{T} g_{t,j}\right)\right) \\ &\leqslant \frac{1}{\eta} \ln\left(\sum_{j\in[K]} \exp\left(\eta \sum_{t=1}^{T} g_{t,j}\right)\right) = \frac{\ln K}{\eta} + \frac{1}{\eta} \sum_{t=1}^{T} \ln\left(\sum_{j\in[K]} w_{t,j}\, \exp(\eta g_{t,j})\right),\end{aligned}$$

where the equality follows by telescoping. Second, Lemma 6 below exactly states that

$$\ln\left(\sum_{j\in[K]} w_{t,j}\, \exp(\eta g_{t,j})\right) \leqslant \eta \sum_{j\in[K]} w_{t+1,j}\, g_{t,j}\,.$$

All in all, we obtained the final "pre-regret" bound: for all sequences of payoffs $g_{t,j}$, possibly signed and unbounded:

$$\max_{k\in[K]} \sum_{t=1}^{T} g_{t,k} \leqslant \frac{\ln K}{\eta} + \sum_{t=1}^{T} \sum_{j\in[K]} w_{t+1,j}\, g_{t,j}\,. \tag{23}$$

Part 2 — We apply the adversarial bound exhibited. We instantiate (23) to our learning scheme (7):

$$\forall s \in \mathcal{S}, \quad \max_{k\in[K]} \sum_{t=1}^{T} \overline{A}_{q_t\Pi}(s,k) \leqslant \frac{\ln K}{\eta} + \sum_{t=1}^{T} \sum_{j\in[K]} q_{t+1}(j|s)\, \overline{A}_{q_t\Pi}(s,j)\,. \tag{24}$$

We note that for any learning policy π, any state $s_0 \in \mathcal{S}$, and any non-negative function $f : \mathcal{S} \to [0,+\infty)$,

$$(1-\gamma)\, f(s_0) \leqslant \sum_{s\in\mathcal{S}} \mu^{(s_0,\pi)}(s)\, f(s)\,.$$

Therefore, for each $t \geqslant 1$,

$$\sum_{j\in[K]} q_{t+1}(j|s)\, \overline{A}_{q_t\Pi}(s,j) \leqslant \frac{1}{1-\gamma} \sum_{s'\in\mathcal{S}} \mu^{(s,q_{t+1}\Pi)}(s') \sum_{j\in[K]} q_{t+1}(j|s')\, \overline{A}_{q_t\Pi}(s',j) = V_{q_{t+1}\Pi}(s) - V_{q_t\Pi}(s)\,; \tag{25}$$

the equality right above corresponds to the application (10) of the performance difference lemma. All in all, collecting all bounds above, we proved so far

$$\forall s \in \mathcal{S}, \qquad \max_{k\in[K]} \sum_{t=1}^{T} \overline{A}_{q_t\Pi}(s,k) \leqslant \frac{\ln K}{\eta} + \sum_{t=1}^{T} \big(V_{q_{t+1}\Pi}(s) - V_{q_t\Pi}(s)\big) \leqslant \frac{\ln K}{\eta} + \frac{1}{1-\gamma}\,, \tag{26}$$

where the last inequality follows by telescoping and from the fact that a V–value is smaller than $1/(1-\gamma)$ with rewards in $[0,1]$.

Part 3 — We conclude the proof by following the proof of Section 4.2 till (11) and by substituting (26) therein:

$$V_{q^\star\Pi}(s_0) - V_{q_T\Pi}(s_0) \leqslant \frac{1}{(1-\gamma)T} \sum_{s\in\mathcal{S}} \mu^{(s_0,q^\star\Pi)}(s) \max_{k\in[K]} \sum_{t=1}^{T} \overline{A}_{q_t\Pi}(s,k) \leqslant \frac{\ln K}{T\eta(1-\gamma)} + \frac{1}{T(1-\gamma)^2}\,.$$

This concludes the proof. □

Lemma 6. *For all convex weight vectors $(u_1,\dots,_K)$ and all real numbers $a_1,\dots,a_K$,*

$$\ln\left(\sum_{k\in[K]} u_k\, \exp(a_k)\right) \leqslant \sum_{k\in[K]} \frac{u_k\, \exp(a_k)}{\sum_{j\in[K]} u_k\, \exp(a_k)}\, a_k\,.$$

Proof. We apply Jensen's inequality with the convex function $x \mapsto x\ln x$ to get

$$\left(\sum_{k\in[K]} u_k\, \exp(a_k)\right) \ln\left(\sum_{k\in[K]} u_k\, \exp(a_k)\right) \geqslant \sum_{k\in[K]} u_k\, \exp(a_k) \ln\big(\exp(a_k)\big)\,,$$

which after rearranging is exactly the result to be proved. □

D.2 With estimation, in the absence of an oracle

We now wonder whether we may adapt the proof scheme above to handle estimated advantage functions, as we did in Section 5.1 for other strategies including the ones based on polynomial potentials.

We extract (or adapt) the following (in)equalities from Sections 5.1 and Appendix D.1. First, by T applications of (10), we have

$$V_{q^\star\Pi}(s_0) - \frac{1}{T}\sum_{t=1}^{T} V_{\widetilde{q}_T\Pi}(s_0) = \frac{1}{T}\sum_{t=1}^{T} \frac{1}{1-\gamma} \sum_{s\in\mathcal{S}} \mu^{(s_0,q^\star\Pi)}(s) \sum_{k\in[K]} q^\star(k|s)\, \overline{A}_{\widetilde{q}_t\Pi}(s,k)\,.$$

The pre-regret bound (23) yields the following adaptation of (24):

$$\forall s \in \mathcal{S}, \qquad \sum_{k\in[K]} q^\star(k|s) \sum_{t=1}^{T} \widetilde{A}_{\widetilde{q}_t\Pi}(s,k) \leqslant \frac{\ln K}{\eta} + \sum_{t=1}^{T} \sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\,.$$

The bound (25) may be adapted as

$$\sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \overline{A}_{\widetilde{q}_t\Pi}(s,k) \leqslant \frac{1}{1-\gamma} \sum_{s'\in\mathcal{S}} \mu^{(s,\widetilde{q}_{t+1}\Pi)}(s') \sum_{k\in[K]} \widetilde{q}_{t+1}(k|s')\, \overline{A}_{\widetilde{q}_t\Pi}(s',k) = V_{\widetilde{q}_{t+1}\Pi}(s) - V_{\widetilde{q}_t\Pi}(s)\,,$$

so that, after telescoping and after bounding the values by $1/(1-\gamma)$,

$$\sum_{t=1}^{T} \sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \overline{A}_{\widetilde{q}_t\Pi}(s,k) \leqslant \frac{1}{1-\gamma}\,.$$

To combine the inequalities above and conclude this proof scheme, we should therefore (and among others) relate, with high probability and up to the bias ε,

$$\sum_{k\in[K]} q^\star(k|s) \sum_{t=1}^{T} \overline{A}_{\widetilde{q}_t\Pi}(s,k) \qquad \text{and} \qquad \sum_{k\in[K]} q^\star(k|s) \sum_{t=1}^{T} \widetilde{A}_{\widetilde{q}_t\Pi}(s,k)\,,$$

which (by Lemma 3) may be readily achieved by an application of the Hoeffding-Azuma inequality, as in the proof of Theorem 3. However, it seems difficult to relate

$$\sum_{t=1}^{T} \sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \widetilde{A}_{\widetilde{q}_t\Pi}(s,k) \qquad \text{and} \qquad \sum_{t=1}^{T} \sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \overline{A}_{\widetilde{q}_t\Pi}(s,k) = \sum_{t=1}^{T} \sum_{k\in[K]} \widetilde{q}_{t+1}(k|s)\, \widetilde{\mathbb{E}}\big[\widetilde{A}_{\widetilde{q}_t\Pi}(s,k) \,\big|\, \mathcal{F}_{t-1}\big]\,.$$

Indeed, the quantities $\widetilde{q}_{t+1}(k|s)$ are $\mathcal{F}_t$–measurable but not necessarily $\mathcal{F}_{t-1}$–measurable, so that the classic deviation inequalities like the Hoeffding-Azuma inequality are not applicable. The issue here is similar to the one discussed in Comment 3.

E DETAILED SIMULATION STUDY

In this appendix, we provide full technical details on the setting alluded at in Section 6, as well as the detailed outcomes for all scenarios.

E.1 Formal description of the MPD and of the expert policies

We first formalize mathematically the description of the MDP and expert policies provided in Section 6. We recall that $\varrho_{i,t}$ denotes the number of items of class i in the queue at the beginning of round $t \geqslant 0$, and that each queue has a maximal length of L.

Action space. The actions consist of making a match, i.e., selecting an index in $[I]$, putting the item in its queue, which we denote by $\vDash$, or trashing it, which we denote by $\boxtimes$ if its queue is already full. That is,

$$\mathcal{A} = [I] \cup \{\vDash, \boxtimes\}$$

More precisely, the action taken a_t lies in $[I]$ if a match can be made between i_t and an item of class a_t: this requires compatibility between i_t and a_t (as indicated by the compatibility graph), and the availability of a least one item in the queue a_t, i.e., $\varrho_{t,a_t} \geqslant 1$. Otherwise, $a_t = \vDash$ if $\varrho_{i_t,t} \leqslant L-1$ and $a_t = \boxtimes$ if $\varrho_{i_t,t} = L$.

State space. The situation at the beginning of the round $t \geqslant 0$ is summarized by the pair $s_t = (\varrho_t, i_t)$, where $\varrho_t = (\varrho_{t,i})_{i\in[I]}$ is the vector of all queue sizes. The state space therefore equals

$$\mathcal{S} = [I] \times [L]^I .$$

Transition kernel. We next state the transition kernel $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ To do so, we consider a state $s = (\varrho, i)$ and a possible action a, i.e., such that i and a are compatible and $\varrho_a \geqslant 1$ whenever $a \in [I]$. The subsequent state $s' = (\varrho', i')$ is generated as follows: the index i' is drawn independently at random according to λ, while ϱ' is obtained in a deterministic way by

$$\varrho' = \begin{cases} \varrho & \text{if } a = \boxtimes, \\ \varrho + \mathbf{1}_i & \text{if } a = \vDash, \\ \varrho + \mathbf{1}_a & \text{if } a \in [I], \end{cases}$$

where $\mathbf{1}_j$ denotes indicator vector of $j \in [I]$, i.e., the vector with 0 everywhere except at component $j \in [I]$, where it equals 1. This fully determines $\mathcal{T}$.

Reward function. We now move to the (deterministic) reward function $r : \mathcal{S} \times \mathcal{A} \to [0, M]$, where M denotes its upper range. It turns out that (with no loss of generality) we do not normalize it, for the sake of more readable gaps between values. Positive rewards will be obtained in case of a match, but some matches will be more rewarding than others. Costs for maintaining the queues will be suffered in all cases. The actions of placing an item in a queue or trashing it lead to the same values of the reward function. More precisely, for a given state $s = (\varrho, i)$ and an action s, the reward function is given by

$$r(s,a) = c \sum_{j\in[I]} (L - \varrho_j) + \begin{cases} 0 & \text{if } a \in \{\boxtimes, \vDash\}, \\ g_{(i,a)} & \text{if } a \in [I], \end{cases}$$

where c is some penalty parameter, and where the parameters $g_{j,j'} = g_{j',j}$ quantify how desirable is a match between (compatible) items of respective classes j and j'. The values of $g_{j,j'}$ will be written on the compatibility graphs.

The range M is given by $M = cLI + \max\limits_{i\in[I]} \max\limits_{j\in\mathcal{N}(i)} g_{i,j}$.

Distribution on initial state. The initial state $s_0 = (\varrho_0, i_0)$ consists of an index i_0 drawn at random according to λ and of an empty queue: $\varrho_{0,j} = 0$ for all $j \in [I]$. We denote by μ_0 the corresponding distribution of s_0.

Expert policies. We denote by $\mathcal{N}(i) \subseteq [I]$ the item classes that are compatible with class i, and formalize the expert policies π_1, π_2, π_3, $\pi_4 : \mathcal{S} \to \mathcal{P}(\mathcal{A})$, which correspond, respectively, to match with the longest queue possible (π_1), match with the most-reward item (π_2), matching at random (π_3), and matching based on preferences determined by some random permutation σ of $[I]$ drawn beforehand (π_4). For a given state $s = (\varrho, i)$, we denote by

$$\mathcal{M}(s) = \big\{ j \in \mathcal{N}(i) : \varrho_j \geqslant 1 \big\}$$

the prospective matches. Neglecting the tie-breaking rules,

$$\text{if } \mathcal{M}(s) \neq \emptyset, \qquad \pi_1(\,\cdot\,|s) = \mathrm{Dirac}\left(\operatorname*{argmax}_{j \in \mathcal{M}(s)} \varrho_j \right), \qquad \pi_2(\,\cdot\,|s) = \mathrm{Dirac}\left(\operatorname*{argmax}_{j \in \mathcal{M}(s)} g_{i,j} \right),$$
$$\pi_3(\,\cdot\,|s) = \mathcal{U}_{\mathcal{M}(s)}\,, \qquad \pi_4(\,\cdot\,|s) = \mathrm{Dirac}\left(\operatorname*{argmax}_{j \in \mathcal{M}(s)} \sigma(j) \right),$$

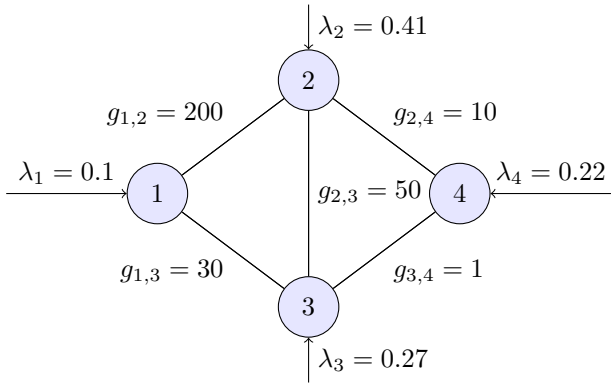
where $\mathrm{Dirac}(k)$ denotes the Dirac mass at j and where $\mathcal{U}_{\mathcal{M}(s)}$ denotes the uniform distribution over $\mathcal{M}(s)$; otherwise,

$$\forall k \in \{1,2,3,4\}\,, \qquad \pi_k(\,\cdot\,|s) = \mathrm{Dirac}(\vDash) \;\text{ if } \varrho_i \leqslant L-1 \qquad \text{and} \qquad \pi_k(\,\cdot\,|s) = \mathrm{Dirac}(\boxtimes) \;\text{ if } \varrho_i = L\,.$$

E.2 Scenario #1

We consider the system summarized below. We chose the arrival probabilities $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ such that the stability conditions of Comte et al. [2021] are satisfied.

Compatibility graph



Parameters

I	L	c	γ	κ	H	cLI	M	$M\varepsilon = M\gamma^H/(1-\gamma)$
4	5	0.5	0.8	0.1	55	10	210	≈ 0.005

The transitions and the rewards functions being known, we—unlike the learning schemes—may of course compute the exact value functions using Bellman's consistency equations for stationary policies (see, for instance, Agarwal et al. [2019] for the corresponding formulas). The table below reports the performance of the expert policies considered—only the first three in this scenario, i.e., $\Pi = (\pi_1, \pi_2, \pi_3)$ here—and of the optimal values, in the class $\mathcal{C}(\Pi)$ of policies of the form $q\Pi$, one the one hand, and among all stationary policies, on the other hand.

$V_{\pi_1}(\mu_0)$	$V_{\pi_2}(\mu_0)$	$V_{\pi_3}(\mu_0)$	$V_{q_\star \Pi}(\mu_0)$	$V^\star(\mu_0)$
135.74	136.80	135.14	138.18	138.92

The table above shows that the best combination of policies $V_{q_\star \Pi}(\mu_0)$ outperforms the best single expert policy, π_2 in the present scenario. The table below reports which expert policies are used in $q_\star$: remember from Section 2

that the $q^\star(\cdot|s)$ may be taken as Dirac masses on one of the expert policies. We may thus compute the average frequencies, over all states $s = (\varrho, i) \in \mathcal{S}$, of each expert policy appearing as the Dirac mass in $q^\star(\cdot|s)$. We obtain the following frequencies, which shows that all expert policies (even the ones that look bad performing) are useful.

Expert policy	π_1	π_2	π_3
Appearance rate in the $q^\star(\cdot\,|s)$	0.48	0.50	0.02

As explained in Section 6, we run the learning schemes of Section 5 a number $N = 20$ times each, where the n-th run gives rise to weights $\widetilde{q}_{n,t}$ indexed by n. In the graph below, we report in colored solid lines

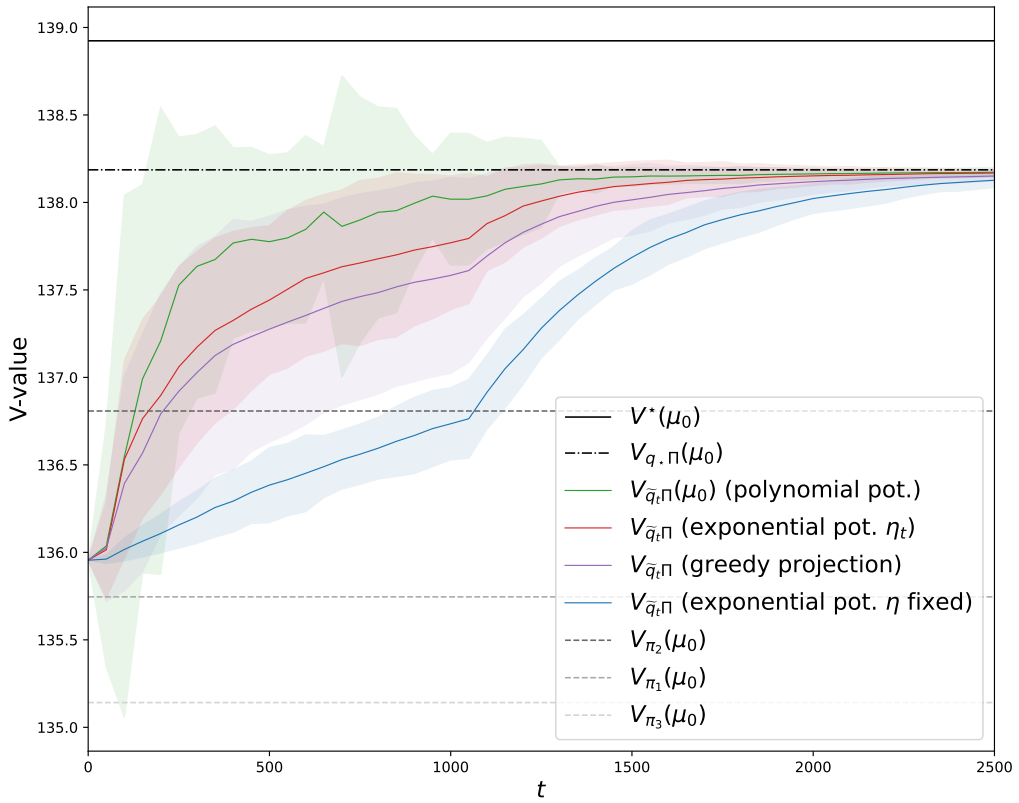
$$t \longmapsto \frac{1}{N}\sum_{n=1}^{N} V_{\widetilde{q}_{n,t}\Pi}(\mu_0)\,,$$

as well as the corresponding $\pm 2\times$ standard errors in shaded areas. We also report the values of the expert policies, as well as $V_{q_\star\Pi}(\mu_0)$ and $V^\star(\mu_0)$.

We obtain the following picture with the learning schemes based on:

- polynomial potential $\Phi : x \mapsto \big(\max\{x, 0\}\big)^p$ with $p = 3 \approx 2\ln K$, see Example 2;
- exponential potential with time-varying learning rates $\eta_t = 0.005/\sqrt{t} \approx (1/M)\sqrt{(\ln K)/t}$, see Example 4;
- exponential potential with fixed learning rate $\eta = 0.00014 \approx (1/M)\sqrt{2(\ln K)/T}$ when $T = 2500$, see Example 3;
- the greedy projection algorithm of Zinkevich [2003] with $\eta_t = 0.004/\sqrt{t} \approx (1/M)\sqrt{2/(Kt)}$, see Example 5.

All learning schemes thus considered achieve the target $V_{q_\star\Pi}(\mu_0)$ set, up to the bias ε.

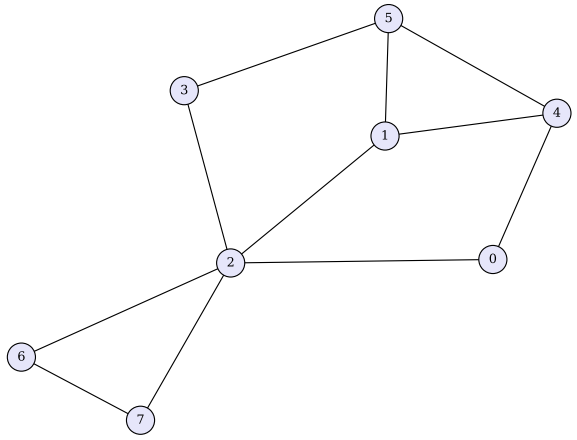


E.3 Scenario #2

We consider the system summarized below. The payoffs (not diplayed) were drawn uniformly over $[0, 20]$. The arrival probabilities λ_j were drawn uniformly over $[0, 1]$ and subsequently normalized to ensure that their sum equals 1. We report no parameter κ: this is because we alleviate the computational burden in a different manner. More precisely, we focus on states visited and perform the weight updates in a "lazy"manner, by updating at round t only the component $s = s_t$: we do so by replacing the estimators given in (22) by

$$\widetilde{A}_{\widetilde{q}_t\Pi}(s,k) = \begin{cases} 0 & \text{if } s \neq s_t, \\ \displaystyle\sum_{a\in\mathcal{A}} \pi_k(a|s)\Big(\widetilde{Q}_{\widetilde{q}_t\Pi}(s,a) - \sum_{a'\in\mathcal{A}} \widetilde{q}_t\Pi(a'|s)\,\widetilde{Q}_{\widetilde{q}_t\Pi}(s,a')\Big) & \text{if } s = s_t\,. \end{cases}$$

Compatibility graph



Parameters

I	L	c	γ	H	cLI	M	$M\varepsilon = M\gamma^H/(1-\gamma)$
8	2	5	0.8	45	80	100	≈ 0.02

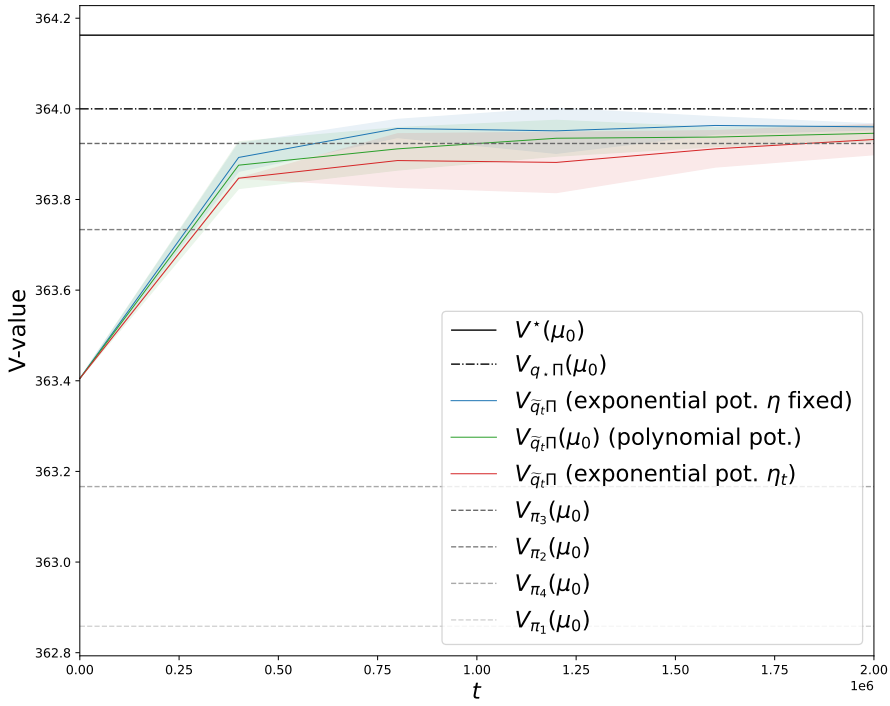
The table below reports the performance of the expert policies considered, $\Pi = (\pi_1, \pi_2, \pi_3, \pi_4)$, and of the optimal values in the class $\mathcal{C}(\Pi)$ of policies of the form $q\Pi$, one the one hand, and among all stationary policies, on the other hand.

$V_{\pi_1}(\mu_0)$	$V_{\pi_2}(\mu_0)$	$V_{\pi_3}(\mu_0)$	$V_{\pi_4}(\mu_0)$	$V_{q_\star\Pi}(\mu_0)$	$V^\star(\mu_0)$
362.85	363.73	363.16	363.92	364.00	364.16

The following table reports which expert policies are used in $q_\star$. Again, as in Scenario #1, all expert policies are useful.

Expert policy	π_1	π_2	π_3	π_4
Appearance rate in the $q^\star(\,\cdot\,|s)$	0.23	0.61	0.01	0.15

We use $N = 5$ repetitions and compute potential-based policy-learning schemes (only), with updated values: polynomial potential with $p = 5$, exponential potential with time-varying learning rates $\eta_t = 0.8/\sqrt{t}$ and exponential potential with fixed learning rate $\eta = 0.014$. As the figure shows on the next page, all algorithms considered achieve again the target $V_{q_\star\Pi}(\mu_0)$ set, up to the bias ε.


364.2
364.0
363.8
363.6
363.4
363.2
363.0
362.8
V-value
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
1e6
t
$V^\star(\mu_0)$
$V_{q_\cdot\Pi}(\mu_0)$
$V_{\widetilde{q}_t\Pi}$ (exponential pot. η fixed)
$V_{\widetilde{q}_t\Pi}(\mu_0)$ (polynomial pot.)
$V_{\widetilde{q}_t\Pi}$ (exponential pot. η_t)
$V_{\pi_3}(\mu_0)$
$V_{\pi_2}(\mu_0)$
$V_{\pi_4}(\mu_0)$
$V_{\pi_1}(\mu_0)$

F A THIRD WAY BETWEEN VALUE LEARNING AND POLICY GRADIENT

As usual in RL, many strategies with various pros and cons can be employed to optimize the overall objective. In the introduction, we motivated the need to follow experts to choose actions rather than attempting direct optimization. In the case of orchestration in a tabular context, a first and natural choice is to employ methods like value iteration or policy improvement, often referred to as value learning. This is the choice made, for instance, in Jin et al. [2022] for a multi-agent context. While a significant body of research provides guarantees for value-learning RL algorithms using tabular representations (see Jin et al. [2018]), many practitioners commonly resort to basic Q-learning algorithms, overlooking more refined schemes with optimistic bonuses. Unfortunately, this can result in exponentially poor convergence rates (see, again, Jin et al. [2018]). However, fine-tuning bonus terms become a daunting challenge for complex Markovian dynamics.

An alternative path is to follow a policy-gradient algorithm, which arguably offers greater flexibility. This is due in particular to their ability to directly learn a stochastic policy, mapping states to probability distributions over actions. Unlike value-based methods, policy-gradient algorithms directly parameterize the policy itself and naturally handle stochastic policies. This inherent stochasticity facilitates exploration, ensuring that the agent continues to discover and learn about the environment (see Schulman et al. [2017]). This exploration can be possibly tuned by a penalization term in the overall objective, allowing good exploration tradeoffs (see Agarwal et al. [2021]). In the absence of penalization, the optimal policy is deterministic. However, maintaining a sufficiently stochastic policy, strategically steering clear of quasi-deterministic states, prevents the problem of vanishingly small gradients. Unlike the well-established sample complexity results for value learning, the theory of convergence and sample complexity for policy gradients has only recently begun to take shape, marked by the seminal work discussed earlier—namely, the one by Agarwal et al. [2021].

In this article, we explore parameterizations of policies using state-dependent convex combinations of few expert policies, determined by adversarial learning algorithms. Interestingly, in one specific case, given by the exponential potential-based algorithm (see Remark 1), we retrieve the natural policy gradient scheme defined by Kakade [2001], Kakade and Langford [2002b]. For more general adversarial learning algorithms, it remains an open problem to link the resulting policy-learning schemes to (potentially modified) policy-gradient methods.