The Online Loop-free Stochastic Shortest-Path Problem

Gergely Neu[*†]
neu.gergely@gmail.com
[*]Department of Computer Science and Information Theory, Budapest University of Technology and Economics

András György[†]
gya@szit.bme.hu
[†]Machine Learning Research Group, Computer and Automation Research Institute of the Hungarian Academy of Sciences

Csaba Szepesvári[‡†]
szepesva@ualberta.ca
[‡]Department of Computing Science, University of Alberta

Abstract

We consider a stochastic extension of the shortest path problem with adversarial rewards. In this episodic Markov decision problem the agent traverses through an acyclic graph with random transitions: at each step of an episode the agent chooses an action, receives some reward, and arrives at a random next state, where the reward and the distribution of the next state depends on the actual state and the chosen action. We consider the bandit situation when only the reward of the just visited state-action pair is revealed to the agent. For this problem we develop algorithms that perform asymptotically as well as the best stationary policy in hindsight. Assuming that all states are reachable with probability $\alpha > 0$ under all policies, we give an algorithm and prove that its regret is $\mathcal{O}(L^2\sqrt{T|\mathcal{A}|/\alpha})$, where T is the number of episodes, $\mathcal{A}$ denotes the (finite) set of actions, and L is the length of the longest path in the graph. Variants of the algorithm are given that improve the dependence on the transition probabilities under specific conditions. The results are also extended to variations of the problem, including the case when we compete with time varying policies.

1 Introduction

Consider the problem of controlling an inventory so as to maximize the revenue. This is an optimal control problem, where the state of the controlled system is the stock, the action is the amount of stock ordered. The evolution of the stock is also influenced by the demand, which is assumed to be stochastic. Further, the revenue depends on the prices at which products are bought and sold. In this example the dynamics of one component of the controlled system (the evolution of stock) is well understood, while another part (the process generating the prices) can only be poorly modeled. In addition, the poorly known part is uncontrolled and its state is unobserved (i.e., this part evolves autonomously) and it influences the rewards only, while the state of the controlled part is available for measurement. This situation is shown in Figure 1.

If the uncontrolled part is hard to model, a better approach might be to solve the following robust control problem: Choose a sufficiently large class of controllers that has the property that no matter how the state of the uncontrolled part evolves, the class contains some controller that performs well. Then, design an algorithm that is able to perform almost as well as the best controller in the chosen class selected based on hindsight.

This problem formulation shares many similarities with the so-called expert framework, where the task is to find an algorithm that can predict (almost) as well as the best amongst a fixed set of experts in an arbitrary prediction environment (cf. Chapter 2 of Cesa-Bianchi and Lugosi, 2006 and the references therein). However, the control problem is made more complicated by the fact that one must take into account that the decisions of the controller influence future states and thus also future rewards. This, in fact, has two consequences: Firstly, in order to perform well, the controller must plan ahead in time. That is, the controller must address the usual temporal credit assignment problem. This is usually done by resorting to some form of (approximate) dynamic programming to maintain computational efficiency (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998). Secondly, the controller must also address the exploration-exploitation problem which arises because only the rewards associated with the state-action pairs visited are available for measurement. This is again made difficult by the fact that in order to be able to explore an action in a given state, the state must first be visited, which requires some planning.

In this paper we consider a special case of this general problem, which we call *the online, loop-free stochastic shortest-path (Online SSP, O-SSP) problem*. This problem is a generalization of two previously

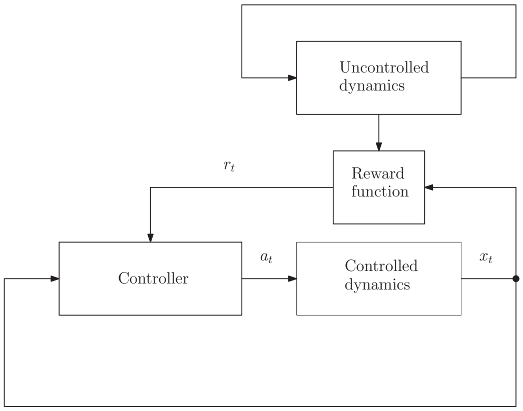


Figure 1: Illustration of the general problem whose special case is studied in the paper: The controlled system has two components. One component, whose state is controlled and observable and is perfectly known, while the other component is unknown and uncontrolled. The second component influences the rewards received and the rewards represent the only source of information about this component. When the uncontrolled part has a complex dynamics and/or a complex state, its identification is hopeless and one might be better off with implementing a robust optimal control strategy, such as the one described in this paper.

considered problems: it is an online extension of the (loop-free version of the) stochastic shortest-path problem (Bertsekas and Tsitsiklis, 1996) and a stochastic extension of the online shortest path problem (György et al., 2007). The problem is defined as follows: The controlled dynamics is stochastic. It is assumed that the number of states and actions is finite. There is a distinguished initial state and terminal state amongst the states and the state space has the structure of a layered graph: an action chosen at some state of some layer of the graph leads to another state in the next layer. When the terminal state is reached, a new episode starts: the state of the system is reset to the initial state. At the same time, a new reward function is chosen (since no state is visited twice, there is no reason to change the reward function before the end of an episode). Note that only the reward of the last state-action pair is made available to the algorithm, that is, we consider the so-called *bandit setting*. The class of controllers that our algorithm must compete with is selected to be the class of state-feedback policies, that is, policies that select actions according to the actual state, or the class of policies which switch between such state-feedback policies.

Clearly, the inventory management problem mentioned beforehand falls into this class of problem provided that we restrict our attention to its finite horizon variant when the stocks, orders and demand are measured in discrete units, the size of the inventory is limited to lie between a maximum size and zero (excess demands are lost) and where the demands are independent, identically distributed random variables. The O-SSP setup is also particularly suited to address the problem of robust adaptive routing in virtual networks over some (possibly wireless) base network with a fixed routing strategy. Other examples include machine maintenance, asset pricing, or production planning. In general, our framework captures operations research problems, where the control objective involves components which depend on some exogenously developing, hard to model prices.

The main results of this paper are as follows: Assuming that all states are reachable with probability $\alpha > 0$ under all policies, we give an algorithm and prove that its regret is $\mathcal{O}(L^2\sqrt{T|\mathcal{A}|}/\alpha)$, where L is the number of layers, T is the number of episodes and $\mathcal{A}$ denotes the (finite) set of actions (Theorem 4). Although the number of states in a given layer does not show up in the bound, the bound shows a scaling that is at least linear with the number states since $\max_{1\le l\le L}|\mathcal{X}_l| \le 1/\alpha$, where $|\mathcal{X}_l|$ is the number of states in the l-th layer. We also give a variant of this result that shows a possibly improved dependence on the transition probabilities (since α can be exponentially small in the size of the number of states). This result is given in Theorem 5. The results are also extended to compete with time-varying policies in Theorem 8. A nice property of the algorithms proposed is that they use bandit algorithms developed for the prediction (stateless) setting, the only requirement for the bandit algorithm being that it should return a probability distribution over the actions. Hence, our algorithm can make use of specially tailored, improved bandit algorithms, for example, algorithms with adaptive tuning that may achieve better performance (and bounds) when the best action has very large gains (Auer et al., 2002b) or algorithms with improved performance when *many* actions have relatively good performance (Exercise 2.6 of Cesa-Bianchi and Lugosi 2006). Specifically, when the **Exp3** algorithm of Auer et al. (2002a) is used, the dependence on α can be improved to $\mathcal{O}(1/\sqrt{(\alpha)}$ (Theorem 6). Finally, in this special case, under the less stringent assumption that for every state there is some policy that reaches the state with positive probability we give an algorithm whose expected regret per step vanishes over time (Theorem 7).

How do our results compare to those in earlier works in the online learning literature? As noted earlier, our work can be viewed as a stochastic extension of works that considered online shortest path problems in deterministic settings. Here, the closest to our ideas and algorithm is the paper by György et al. (2007). One major difference between the algorithms is that our algorithm is based on direct estimates of the *total* reward to go in every state-action pair, whereas the algorithm of György et al. (2007) estimates the reward to go via estimating the *immediate* rewards. Compared to the bound in György et al. (2007), our bounds are slightly larger (and thus weaker). In earlier work, Awerbuch and Kleinberg (2004) gave an $\mathcal{O}(T^{2/3})$ regret bound, while McMahan and Blum (2004) gave an $\mathcal{O}(T^{3/4})$ bound, building upon the exponentially weighted average forecaster and, respectively, the follow the perturbed leader algorithm, both under the assumption that the only information received is the total reward at the end of the episodes. More recently, Dani et al. (2008) proposed a generalization of **Exp3** due to Auer et al. (1995), which can be applied to this setting and which gives an expected regret of $\mathcal{O}(|\mathcal{X}|^{3/2}T^{1/2})$, where $|\mathcal{X}|$ is the size of the state space. More recently, Bartlett et al. (2008) showed that the algorithm can be extended so that the bound holds with high probability. We note in passing that Dani et al. (2008) suggest that their algorithm can be implemented efficiently for the MDP setting. However, this is not clear at all: Although, conceptually, the algorithm can be applied to our case, when policies are represented through the distributions that they induce over the state space, but this does not seem to lead to an algorithm that can be implemented.

Another thread of work that is closely related to ours considers algorithms for learning and acting in Markovian decision processes (MDPs) with arbitrary reward sequences. In fact, clearly, our framework is a special case of this more general framework. The first work that considered this setting is due to Even-Dar et al. (2005, 2009). In this work the restriction on the MDP is that it must be *unichain* (i.e., all stationary policies must generate a unique stationary distribution) and it is assumed that the worst mixing time, τ, over all policies is uniformly small (the mixing time appears in the bounds). This is similar to our assumption of the MDP being episodic, with all policies terminating after L steps (though strictly speaking, their assumption does not hold true in our setting). However, the major difference between our work and that of Even-Dar et al. (2005, 2009) is that they assume that the reward function is fully observable, whereas we consider the bandit setting. They propose an algorithm, MDP-E, which is very similar to ours in that it uses some (optimized) expert algorithm in every state which is fed with the action-values of the policy used in the last round (which, in our case, corresponds to the total reward to go). They prove a bound on the expected regret of this algorithm of the form $\mathcal{O}(\tau^2\sqrt{T\log|\mathcal{A}|})$. The improved dependence on the action set (as compared to our bound stated above) is the result of the assumption that the reward function is available at every step and not only the reward of the last state-action pair visited, otherwise the bound shows a dependence somewhat similar to ours in the main quantities. We actually prove a similar bound for our problem, just to fix some ideas, in Section 4.

More recently, Yu et al. (2009) proposed algorithms for the same (full information) problem and proved a bound on the expected regret of order $\mathcal{O}((\tau + |\mathcal{A}| + |\mathcal{X}|)\,\tau|\mathcal{A}|^2T^{3/4+\varepsilon}\log T)$ for arbitrary $\varepsilon \in (0, 1/3)$.[1] The algorithm proposed ("Lazy FPL") works with phases of length $m^{1/3-\varepsilon}$ and changes policies only at the end of the phases. At the end of a phase the optimal (differential) value function corresponding to the sum of past reward functions is first found. Within the phase, the action to be followed at some time step is then selected as the one that maximizes the one-step lookahead action value computed with this value function but with the immediate rewards perturbed randomly in an appropriate manner. The advantage of this algorithm to that of Even-Dar et al. (2009) is that it is computationally less expensive, which, however, comes at the price of an increased bound on the regret. Yu et al. (2009) introduced another algorithm ("Q-FPL") which is shown to enjoy a vanishing regret over time (i.e., the algorithm is Hannan consistent). The major advance, however, is that, for the first time, Yu et al. (2009) proposed an algorithm ("Exploratory FPL") to address the problem of learning in the bandit setting. This algorithm estimates the immediate rewards by appropriately weighting the rewards received and in a phase either uses a uniformly exploring policy or that of underlying their Lazy FPL algorithm. They prove that the regret of this algorithm vanishes almost surely.

Yu and Mannor (2009a,b) considered the problem of on-line learning in MDPs where the transition probabilities may also change arbitrarily after each transition. This problem is significantly more difficult than the case where only the reward function is changed arbitrarily. Accordingly, the algorithms proposed in these papers fail to achieve consistency. Yu and Mannor (2009b) also considered the case when rewards are only observed along the trajectory traversed by the agent. However, a closer investigation reveals that their result (Theorem IV.1) is incorrect: If the state space consists of only a single state, the problem becomes identical to the non-stochastic multi-armed bandit problem. Then it can be shown easily that the obtained bound on the expected regret is $\mathcal{O}(\sqrt{\log|\mathcal{A}|T})$, which contradicts the known $\Omega(\sqrt{|\mathcal{A}|T})$ lower bound on the regret for such problems.[2]

[1] The notion of mixing time in this paper is somewhat, but not essentially different than that of used by Even-Dar et al. (2005, 2009).

[2] To show this contradiction, one has to replace, in the bound of Theorem IV.1 of Yu and Mannor (2009b), the condition $T > N$ with an extra $\mathcal{O}(1/T)$ term, and then let ϵ and δ converge to zero at appropriate rates.

2 Problem definition

Formally, a Markovian Decision Process (MDP) M is defined by a state space $\mathcal{X}$, an action set $\mathcal{A}$, a transition function $P : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to [0,1]$, and a reward function $r : \mathcal{X} \times \mathcal{A} \to [0,1]$. In time step k, knowing the state $x_k \in \mathcal{X}$, a decision maker (or agent) acting in the MDP M, chooses an action $a_k \in \mathcal{A}(x)$ where $\mathcal{A}(x) \subset \mathcal{A}$ is the set of admissible actions at state x. As a result the process moves to state $x_{k+1} \in \mathcal{X}$ with probability $P(x_{k+1}|x_k, a_k)$ and the decision maker receives reward $r(x_k, a_k)$ (this implies that for any $x \in \mathcal{X}$ and $a \in \mathcal{A}(x)$, $P(\cdot|x,a)$ defines a probability distribution over $\mathcal{X}$). The goal of the agent is to maximize its average reward. In an episodic MDP there is a terminal state $x \in \mathcal{X}$: if this state is reached, the episode is ended and the whole process starts again with a designated starting state. For a more detailed introduction the reader is referred to, for example, Puterman (1994).

The loop-free stochastic shortest path (SSP) problem is a special case of episodic MDPs. Informally, given an acyclic directed graph an agent has to traverse repeatedly over paths between two given vertices of the graph. At each vertex the agent makes a decision, and based on the decision it follows a random edge of the graph to the next vertex and receives some reward. The goal of the agent is to maximize its average reward received over the paths. More formally, we consider MDPs where the state space $\mathcal{X}$ consists of layers, that is, $\mathcal{X} = \cup_{l=0}^{L} \mathcal{X}_l$, where $\mathcal{X}_l$ is called the lth layer of the state space and $\mathcal{X}_l \cap \mathcal{X}_k = \emptyset$ for all $l \neq k$. The first and last layers are singleton layers, that is, $\mathcal{X}_0 = \{x_0\}$ and $\mathcal{X}_L = \{x_L\}$. The significance of the layers is given by the fact that the state of the agent can only move between consecutive layers, that is, in each episode the agent starts at layer 0, and at time instant l it is at layer l until it reaches the terminal state x_L. This assumption is equivalent to assuming that each path in the graph is of equal length, and is reflected by the special structure of the transition function: for any $x_l \in \mathcal{X}_l$ and $a \in \mathcal{A}(x_l)$, $P(x_{l+1}|x_l, a_l) = 0$ if $x_{l+1} \notin \mathcal{X}_{l+1}$.[3] For any state $x \in \mathcal{X}$ we will use l_x to denote the index of the layer x belongs to, that is, $l_x = l$ if $x \in \mathcal{X}_l$.

In this paper we consider the online version of the loop-free SSP problem, in which case the reward function is allowed to change between episodes, that is, instead of a single reward function r, we are given a sequence of rewards $\{r_t\}$ describing the rewards at episode t that is assumed to be an individual sequence fixed in advance[4], that is, no statistical assumption is made about the reward values. Note that the constraint that r_t depends only on the current state and action is assumed only for simplicity: the results of the paper can easily be extended to the situation where r_t is allowed to depend on the next state as well (i.e., when the reward function is of the form $r_t(x_l, a_l, x_{l+1})$).

A stochastic stationary policy (or, in short: a policy) is a mapping $\pi : \mathcal{A} \times \mathcal{X} \to [0,1]$, where $\pi(a|x) \equiv \pi(a,x)$ is the probability of taking action a in state x. The *instantaneous value function* and *action-value function* with respect to π at episode t are defined, respectively, as

$$v_t^\pi(x_l) = \mathbb{E}\left[\sum_{k=l}^{L-1} r_t(\mathbf{x}_k, \mathbf{a}_k) \middle| \mathbf{x}_l = x_l\right]$$

$$q_t^\pi(x_l, a_l) = \mathbb{E}\left[\sum_{k=l_x}^{L-1} r_t(\mathbf{x}_k, \mathbf{a}_k) \middle| \mathbf{x}_l = x_l, \mathbf{a}_l = a_l\right],$$

where the sequence $(\mathbf{x}_0, \mathbf{a}_0), (\mathbf{x}_1, \mathbf{a}_1), \ldots, (\mathbf{x}_{L-1}, \mathbf{a}_{L-1})$ is generated by the policy π and the MDP, and the expectations are taken with respect to π and the transition function P. These values are equivalently defined by the Bellman equations:

$$\begin{aligned} q_t^\pi(x,a) &= r_t(x,a) + \sum_{x'} P(x'|x,a) v_t^\pi(x') \\ v_t^\pi(x) &= \sum_a \pi(a|x) q_t^\pi(x,a), \end{aligned} \tag{1}$$

with $v_t^\pi(x_L) = 0$. The *cumulative action-value* and *cumulative value functions* are defined, respectively, as

$$Q_t^\pi = \sum_{s=1}^{t} q_s^\pi \qquad \text{and} \qquad V_t^\pi = \sum_{s=1}^{t} v_s^\pi.$$

Each policy generates a probability distribution μ_π over each layer $\mathcal{X}_l$, $l = 0, 1, \ldots, L$, that is,

$$\mu_\pi(x_l) = \mathbb{P}[\mathbf{x}_l = x_l | \mathbf{x}_0 = x_0].$$

[3] Note that all loop-free state spaces can be transformed to one that satisfies our assumptions. A simple transformation algorithm is given in Appendix A of György et al. (2007).

[4] That is, we assume that we are dealing with a so called oblivious opponent.

The distribution μ_π can be computed recursively as

$$\mu_\pi(x_l) = \sum_{x_{l-1}, a_{l-1}} P(x_l | x_{l-1}, a_{l-1}) \pi(a_{l-1} | x_{l-1}) \mu_\pi(x_{l-1}), \tag{2}$$

for $l = 1, 2, \ldots, L$, with $\mu_\pi(x_0) = 1$. The *expected return* of a fixed policy π for a time horizon $T > 0$ is defined as $R_T^\pi = \sum_{t=1}^T v_t^\pi = V_T^\pi$. The return of the best policy in hindsight is given by

$$R_T^* = \sup_\pi \sum_{t=1}^T v_t^\pi(x_0) = \sup_\pi V_T^\pi(x_0).$$

It is known that there exists a stationary and deterministic policy π_T^* that achieves the above maximum (Puterman, 1994, Theorem 4.4.2), and so we can use $\max$ instead of $\sup$ in the above equation. By a slight abuse of the notation we will use $\pi_T^*(x)$ to denote the action for which $\pi_T^*(a|x) \neq 0$. The state distribution generated by the optimal policy will be denoted as $\mu_T^* \equiv \mu_{\pi_T^*}$.

Our goal is to construct a sequential decision algorithm (agent) that asymptotically achieves the above return averaged over the episodes. The decision algorithm may follow a different policy $\boldsymbol{\pi}_t$ at each episode $t = 1, 2, \ldots, T$. This policy may be random, as it may depend on the previous states the agent visited and the previous rewards it received. The random path traversed by the agent at episode t will be denoted by

$$\mathbf{u}_t = \left\{ \mathbf{x}_0^{(t)}, \mathbf{a}_0^{(t)}, \mathbf{x}_1^{(t)}, \mathbf{a}_1^{(t)}, \ldots, \mathbf{x}_{L-1}^{(t)}, \mathbf{a}_{L-1}^{(t)}, \mathbf{x}_L^{(t)} \right\},$$

and the path history up to episode t by

$$\mathbf{U}_t = \left\{ \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_t \right\},$$

for all $t = 1, 2, \ldots, T$ with $\mathbf{U}_0 = \emptyset$. Note that $\mathbf{U}_t$ covers all the randomness in the problem (including the random transitions and the possible randomness in the agent's decisions). Thus,

$$\boldsymbol{\pi}_t(a|x) = \mathbb{P}\left[\mathbf{a} = a | \mathbf{x} = x, \mathbf{U}_{t-1} \right].$$

The value function and the action-value function of policy $\boldsymbol{\pi}_t$ are given, respectively, by

$$\begin{aligned} \mathbf{v}_t(x_l) &= \mathbb{E}\left[\sum_{k=l_x}^{L-1} r_t(\mathbf{x}_k, \mathbf{a}_k) \middle| \mathbf{x}_l = x_l, \mathbf{U}_{t-1} \right] \\ \mathbf{q}_t(x_l, a_l) &= \mathbb{E}\left[\sum_{k=l_x}^{L-1} r_t(\mathbf{x}_k, \mathbf{a}_k) \middle| \mathbf{x}_l = x_l, \mathbf{a}_l = a_l, \mathbf{U}_{t-1} \right] \end{aligned}$$

where the sequence $(\mathbf{x}_0, \mathbf{a}_0), (\mathbf{x}_1, \mathbf{a}_1), \ldots, (\mathbf{x}_{L-1}, \mathbf{a}_{L-1})$ is generated by the policy $\boldsymbol{\pi}_t$ (that is fully determined by $\mathbf{U}_{t-1}$). We will also use $\mathbf{Q}_t = \sum_{s=1}^t \mathbf{q}_s$ and $\mathbf{V}_t = \sum_{s=1}^t \mathbf{v}_s$. The state distribution generated by $\boldsymbol{\pi}_t$ is denoted by $\boldsymbol{\mu}_t = \mu_{\boldsymbol{\pi}_t}$, where $\mu_{\boldsymbol{\pi}_t}(x) = \mathbb{P}[x \in \mathbf{u}_t | \mathbf{U}_{t-1}]$.

The expected return accumulated by the agent in the first T episodes is

$$\hat{R}_T = \sum_{t=1}^T \mathbb{E}\left[\mathbf{v}_t(x_0) \right] = \mathbb{E}\left[\mathbf{V}_T(x_0) \right],$$

and its relative loss with respect to the best fixed policy π_T^* in hindsight, called *regret*, is defined as

$$\hat{L}_T = R_T^* - \hat{R}_T = V_T^*(x_0) - \mathbb{E}\left[\mathbf{V}_T(x_0) \right].$$

The following lemma will be a key to our main results. Note that a similar argument is used by Even-Dar et al. (2009) to prove their main result about online learning in unichain MDPs in the full information case (cf. Lemma 4.1). The benefit of this lemma is that the problem of bounding the regret is essentially reduced to the problem of bounding the difference between action-values *of the policy followed by the agent*.

Lemma 1 *For any time horizon $T > 0$, let the state distribution generated by the optimal policy π_T^* be denoted by μ_T^*, and define*

$$V_T^+(x) = \mathbb{E}\left[\mathbf{Q}_T(x, \pi_T^*(x)) \right].$$

Then

$$V_T^*(x_0) - \mathbb{E}\left[\mathbf{V}_T(x_0) \right] = \sum_{l=0}^{L-1} \sum_{x_l \in \mathcal{X}_l} \mu_T^*(x_l) \big(V_T^+(x_l) - \mathbb{E}\left[\mathbf{V}_T(x_l) \right] \big).$$

Proof:

$$\begin{aligned} V_T^*(x_0) - \mathbb{E}\left[\mathbf{V}_T(x_0)\right] &= V_T^*(x_0) - V_T^+(x_0) + V_T^+(x_0) - \mathbb{E}\left[\mathbf{V}_T(x_0)\right] \\ &= Q_T^*(x_0, \pi_T^*(x_0)) - \mathbb{E}\left[\mathbf{Q}_T(x_0, \pi_T^*(x_0))\right] + V_T^+(x_0) - \mathbb{E}\left[\mathbf{V}_T(x_0)\right] \\ &= \sum_{x_1 \in \mathcal{X}_1} P(x_1|x_0, \pi_T^*(x_0))\big(V_T^*(x_1) - \mathbb{E}\left[\mathbf{V}_T(x_1)\right]\big) + V_T^+(x_0) - \mathbb{E}\left[\mathbf{V}_T(x_0)\right] \\ &= \cdots = \sum_{l=0}^{L-1} \sum_{x_l \in \mathcal{X}_l} \mu_T^*(x_l)\big(V_T^+(x_l) - \mathbb{E}\left[\mathbf{V}_T(x_l)\right]\big) \,. \end{aligned}$$

■

3 Sequential prediction with expert advice

A widely studied special case of our setting where the state space consists of a single state is called sequential prediction with expert advice (Cesa-Bianchi and Lugosi, 2006). In this context, actions are usually referred to as *experts*, and several algorithms have been developed that solve the many variants of the problem. Such algorithms E satisfy a regret bound of the form

$$\hat{L}_T \leq \rho_E(T, \mathcal{A}) \tag{3}$$

where $\rho_E(T, \mathcal{A})$ is a sublinear function of T, and so $\lim_{T\to\infty} \hat{L}_T/T \to 0$. Furthermore, we assume throughout the paper that $\rho_E(T, \mathcal{A})$ is a nondecreasing function of T and $|\mathcal{A}|$. As usually the regret scales linearly with the range of the rewards, it is assumed above that $r_t \in [0, 1]$. In the course of solving our O-SSP problem we are going to use such algorithms as basic building blocks. Note that depending on the actual form of the algorithm, E may be universal in the sense that (3) is satisfied for all T, while several algorithms require T-dependent parameter settings. On the other hand, these methods can be changed to be universal (sometimes at the price of slightly deteriorating the bounds) with either adaptively changing the parameters or simply by resorting to the doubling trick.

The type of the sequential decision problem is usually classified based on the amount of information available to the decision maker, the set of the reference experts and the way the rewards are generated. In the basic setup, known as the case of the *oblivious opponent*, the reward functions $r_1, r_2, \ldots$ are fixed in advance, while in the more general *non-oblivious* setup the rewards may depend on any quantity that is determined before round t. In the latter case, formally we have $\mathbf{r}_t = r_t(\mathbf{U}_{t-1})$.

Luckily, the following lemma, which can be obtained as a special case of a slight generalization of the first part of Lemma 4.1 of Cesa-Bianchi and Lugosi (2006), shows that algorithms that work in the oblivious case also work in the non-oblivious setting:

Lemma 2 *Consider a randomized algorithm E such that, for every $t = 1, 2, \ldots, T$, $\boldsymbol{\pi}_t$ is fully determined by the history $\mathbf{U}_{t-1}$ and the reward sequence $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{t-1}$. Assume that the regret of the algorithm satisfies* (3) *in the oblivious case. Then* (3) *also holds in the non-oblivious case.*

Note that the regret in the non-oblivious case is still defined as $\max_{a\in\mathcal{A}} \sum_{t=1}^{T} (\mathbf{r}_t(a) - \mathbf{r}_t(\mathbf{a}_t))$, where $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_T : \mathcal{A} \to \mathbb{R}$ are the reward functions that are obtained as a result of following E and $\mathbf{a}_t$ is the action taken by E at time step t. In particular, this definition does not take into account that the sequence of reward functions would be different if action a was followed from the beginning. Although this makes, in general, questionable the meaningfulness of this regret definition, in our case this regret definition will still be just good enough.

In the *full information* case the decision maker is informed about the rewards of all actions at the end of each episode; while in the *bandit setting* only the reward of the chosen action is revealed. An *optimized best expert algorithm* in the full information case is an algorithm that attains an expected regret of $\mathcal{O}(\sqrt{T \log |\mathcal{A}|})$, and similarly, an *optimized $|\mathcal{A}|$-armed bandit algorithm* is one that attains an expected regret of $\mathcal{O}(\sqrt{T|\mathcal{A}|})$. Optimized best expert algorithms include the *exponentially weighted average forecaster* (EWA) (a variant of Littlestone and Warmuth's (1994) weighted majority algorithm, and Vovk's (1990) aggregating strategies, also known as Hedge (Freund and Schapire, 1997)) and the *follow the perturbed leader* (FPL) algorithm (Kalai and Vempala, 2003). There exist a number of algorithms for the bandit case that attain regrets of $\mathcal{O}(\sqrt{T|\mathcal{A}| \log |\mathcal{A}|})$, such as **Exp3** by Auer et al. (2002a) and **Green** by Allenberg et al. (2006), while the algorithm presented by Audibert and Bubeck (2009) achieves the optimal rate $\mathcal{O}(\sqrt{T|\mathcal{A}|})$.

4 Full information O-SSP

In this section we give an algorithm and a very short proof that bounds the algorithm's regret in the full information case. The purpose is mainly to fix some ideas that will be useful later on.

In the full information case the reward function r_t is completely revealed after each episode t. We will use the value functions of the agent's policy at each episode t to construct the policy in the next round. Note that as we can exactly compute these value functions, the sequence of the agent's policies does not depend on previous decisions, that is, the policies and the value functions are fully determined by the algorithm. Algorithm 1 uses an arbitrary (optimized) best expert algorithm E in each state x to predict the actions to be taken at that state based on previous values of $q_t(x, \cdot)$. (Thus, the algorithm is essentially the same as the MDP-E algorithm of Even-Dar et al. 2009.)

Algorithm 1 Algorithm for the full information O-SSP.

```
1. Initialize an expert algorithm $E(x)$, an instance of algorithm $E$, for all states $x \in \mathcal{X}$.
2. For $t = 1, 2, \ldots, T$, repeat
    (a) For all $x \in \mathcal{X}$ and all $a \in \mathcal{A}$, let $\pi_t(a|x)$ be the probability that algorithm $E(x)$ chooses action $a$.
    (b) Traverse a path $\mathbf{u}_t$ following the policy $\pi_t$.
    (c) Observe the reward function $r_t$.
    (d) Compute $q_t$ using the Bellman equations (1) for $\pi_t$ and $r_t$.
    (e) For all states $x \in \mathcal{X}$, feed the algorithm $E(x)$ with $q_t(x, \cdot)$.
```

In order to understand how the algorithm works, consider some fixed state x. By definition, $\pi_{t+1}(\cdot|x)$ is the distribution computed by the expert algorithm $E(x)$ when used on a discrete prediction problem with the "reward sequence" $q_1(x, \cdot)$, $q_2(x, \cdot), \ldots$ and action set $\mathcal{A}(x)$. Since $q_t(x, \cdot)$ depends on π_t, which depends on the past rewards, the prediction problem is modeled as one with non-oblivious opponents. The cumulative *expected* reward of the algorithm up to episode T is $\mathbf{V}_T(x)$ and the reward of a constant action a is $\mathbf{Q}_T(x, a)$. Let E be a best expert algorithm with regret bound $\rho_E(T, \mathcal{A})$. By Lemma 2, for any action a at state x, we get

$$\mathbf{Q}_T(x, a) - \mathbf{V}_T(x) \le (L - l_x)\rho_E(T, \mathcal{A}),$$

where we used that $0 \le q_t(x, a) \le L - l_x$. Since in this case $\mathbf{Q}_T$ is non-random, $V_T^+(x) = \mathbf{Q}_T(x, \pi_T^*(x))$ and thus

$$V_T^+(x) - \mathbf{V}_T(x) \le (L - l_x)\rho_E(T, \mathcal{A}). \tag{4}$$

Based on this bound and Lemma 1, we immediately obtain a performance bound on this algorithm for our original problem:

Proposition 3 *Let E be an expert algorithm with regret bound $\rho_E(T, \mathcal{A})$. Then the regret of Algorithm 1 can be bounded as*

$$\hat{L}_T \le \frac{L(L+1)}{2}\rho_E(T, \mathcal{A})$$

Remark: Applying EWA with (time-horizon dependent) optimized parameters as the expert algorithm E, the above bound becomes[5]

$$\hat{L}_T \le \frac{L(L+1)}{2}\sqrt{\frac{T \log |\mathcal{A}|}{2}}\,.$$

Proof: By Lemma 1, we have

$$\hat{L}_T = \sum_{l=0}^{L-1} \sum_{x_l \in \mathcal{X}_l} \mu_T^*(x_l)\left(V_T^+(x_l) - \mathbb{E}[\mathbf{V}_T(x_l)]\right).$$

Using (4) to bound the terms on the right hand side yields the desired bound. ■

[5] See Theorem 2.2 in Cesa-Bianchi and Lugosi (2006).

5 Bandit O-SSP

In the bandit case, the rewards are only observed on the paths that the agent traverses at each episode t. In this section we give an algorithm and analyze its performance for this case.

First, we define conditionally unbiased estimates of $\mathbf{q}_t$ and $\mathbf{v}_t$ given $\mathbf{U}_{t-1}$ as follows:

$$\begin{aligned} \hat{\mathbf{q}}_t(x_l, a_l) &= \begin{cases} \frac{\sum_{k=l}^{L-1} r_t\left(\mathbf{x}_k^{(t)}, \mathbf{a}_k^{(t)}\right)}{\boldsymbol{\pi}_t(a_l|x_l)\boldsymbol{\mu}_t(x_l)} & \text{if } (x_l, a_l) = \left(\mathbf{x}_l^{(t)}, \mathbf{a}_l^{(t)}\right); \\ 0 & \text{otherwise.} \end{cases} & (5) \\ \hat{\mathbf{v}}_t(x_l) &= \sum_a \boldsymbol{\pi}_t(a|x_l)\hat{\mathbf{q}}_t(x_l, a)\ . & (6) \end{aligned}$$

Indeed, it is easy to check that $\mathbb{E}[\hat{\mathbf{q}}_t(x,a)|\mathbf{U}_{t-1}] = \mathbf{q}_t(x,a)$ and $\mathbb{E}[\hat{\mathbf{v}}_t(x)|\mathbf{U}_{t-1}] = \mathbf{v}_t(x)$. Note that the estimates $\hat{\mathbf{q}}_t$ and $\hat{\mathbf{v}}_t$ can only be computed after the end of episode t. We will also use the following key property of this estimate:

$$\mathbb{E}[\hat{\mathbf{q}}_t(x,a) - \hat{\mathbf{v}}_t(x)|\mathbb{I}_{x\in\mathbf{u}_t}, \mathbf{U}_{t-1}] = \mathbb{I}_{x\in\mathbf{u}_t}\frac{\mathbf{q}_t(x,a) - \mathbf{v}_t(x)}{\boldsymbol{\mu}_t(x)}. \tag{7}$$

Similarly to the full information case, Algorithm 2 given below employs an $|\mathcal{A}(x)|$-armed bandit algorithm B in each state x to choose actions using the observations from the previous paths that include x. The only assumption that we make about B is that it works with unbiased estimates of the rewards of the form (5), and its regret scales linearly with the range of the rewards. Note that algorithms like **Exp3** can be redefined to receive unbiased estimates of this form instead of the actual rewards. In the following, we use all bandit algorithms with these updates.

Algorithm 2 Algorithm for the bandit O-SSP.

```
1. Initialize an $|\mathcal{A}(x)|$-armed bandit algorithm $B(x)$, an instance of $B$, for all states $x \in \mathcal{X}$.
2. For $t = 1, 2, \ldots, T$, repeat
   (a) For all $x \in \mathcal{X}$ and all $a \in \mathcal{A}$, let $\boldsymbol{\pi}_t(a|x)$ be the probability that algorithm $B(x)$ chooses action $a$.
   (b) Compute $\boldsymbol{\mu}_t(x)$ for all $x \in \mathcal{X}$ using (2) recursively.
   (c) Traverse a path $\mathbf{u}_t$ following the policy $\boldsymbol{\pi}_t$.
   (d) Observe rewards $r_t(\mathbf{u}_t) = \left\{r_t\left(\mathbf{x}_0^{(t)}, \mathbf{a}_0^{(t)}\right), \ldots, r_t\left(\mathbf{x}_{L-1}^{(t)}, \mathbf{a}_{L-1}^{(t)}\right)\right\}$.
   (e) Construct estimates $\hat{\mathbf{q}}_t$ using equation (5).
   (f) For all states $x \in \mathcal{X}$, feed the algorithm $B(x)$ with $\hat{\mathbf{q}}_t(x, \cdot)$
```

Theorem 4 *Let B be an multi-armed bandit algorithm with regret bound $\rho_B(T, \mathcal{A})$. Assume that there exists some $\alpha > 0$ for which $\mu_\pi(x) \geq \alpha$ holds for all $x \in \mathcal{X}$ and all stationary policies π. Then the regret of Algorithm 2 can be bounded as*

$$\hat{L}_T \leq \frac{L(L+1)}{2\alpha}\rho_B(T, \mathcal{A}).$$

Remark: For example, using the algorithm of Audibert and Bubeck (2009) with appropriate parameters as the base bandit algorithm B yields

$$\hat{L}_T \leq \frac{15L(L+1)}{2\alpha}\sqrt{T|\mathcal{A}|}.$$

Also note that the conditions of the proposition are satisfied if, for example,

$$\min_{x\in\mathcal{X}_l, a\in\mathcal{A}, x'\in\mathcal{X}_{l+1}, l\in 1:L-1} P(x'|x,a) > 0.$$

In fact, our assumption of α being positive is closely related to the uniform mixing assumption used generally in the literature considering online learning in MDPs.

Proof: The set of episodes when state x is visited will be denoted by $\mathbf{T}_x = \{1 \leq t \leq T \,|\, x \in u_t\}$. By Lemma 1, we have

$$\hat{L}_T = \sum_{l=0}^{L-1}\sum_{x_l\in\mathcal{X}_l} \mu_T^*(x_l)\left[V_T^+(x_l) - \mathbb{E}\left[\mathbf{V}_T(x_l)\right]\right]. \tag{8}$$

On the other hand, we have, for any fixed x,

$$\begin{aligned} V_T^{+}(x) - \mathbb{E}\left[\mathbf{V}_T(x)\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\left[\hat{\mathbf{q}}_t(x, \pi_T^*(x)) - \hat{\mathbf{v}}_t(x) \,\middle|\, \mathbf{U}_{t-1}\right]\right] \\ &= \sum_{t=1}^{T} \mathbb{E}\left[\hat{\mathbf{q}}_t(x, \pi_T^*(x)) - \hat{\mathbf{v}}_t(x)\right]. \qquad (9) \end{aligned}$$

Therefore, by (7) we obtain

$$\begin{aligned} V_T^{+}(x) - \mathbb{E}\left[\mathbf{V}_T(x)\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \hat{\mathbf{q}}_t(x, \pi_T^*(x)) - \hat{\mathbf{v}}_t(x)\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}\left[\hat{\mathbf{q}}_t(x, \pi_T^*(x)) - \hat{\mathbf{v}}_t(x) \,\middle|\, \mathbb{I}_{x\in\mathbf{u}_t}, \mathbf{U}_{t-1}\right]\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{I}_{x\in\mathbf{u}_t} \frac{\mathbf{q}_t(x, \pi_T^*(x)) - \mathbf{v}_t(x)}{\boldsymbol{\mu}_t(x)}\right] \\ &= \mathbb{E}\left[\sum_{t\in\mathbf{T}_x} \frac{\mathbf{q}_t(x, \pi_T^*(x)) - \mathbf{v}_t(x)}{\boldsymbol{\mu}_t(x)}\right]. \qquad (10) \end{aligned}$$

As for every x we are using an independent $|\mathcal{A}(x)|$-armed bandit algorithm B with regret bound $\rho_B(T, \mathcal{A}(x))$ that is fed with values $\hat{\mathbf{q}}_t(x, \cdot)$ which are conditionally unbiased estimates of values that belong to $[0, (L - l_x)/\alpha]$, by Lemma 2 we have the following for any fixed a:

$$\mathbb{E}\left[\sum_{t\in\mathbf{T}_x} \frac{\mathbf{q}_t(x, a) - \mathbf{v}_t(x)}{\boldsymbol{\mu}_t(x)}\right] \le \frac{1}{\alpha}(L - l_x)\rho_B(T, \mathcal{A}(x)) \le \frac{1}{\alpha}(L - l_x)\rho_B(T, \mathcal{A}).$$

Combining this bound with (8)-(10) finishes the proof. ■

A problem with the above theorem is that the bound scales with $1/\alpha$, but in certain cases α can be exponentially small. On the other hand, if the minimal probability of visiting a state is exponentially small then the maximal probability of visiting the same state may often be also exponentially small (clearly this is the case in the grid-world example considered in the simulations in Section 6, see Figure 2). The following theorem can be very useful in these situations.

Theorem 5 *Let B be a multi-armed bandit algorithm with regret bound $\rho_B(T, \mathcal{A})$, and define*

$$\alpha(x) = \min_{\pi} \mu_\pi(x) \quad \textit{and} \quad \beta(x) = \max_{\pi} \mu_\pi(x).$$

Assume that $\kappa = \max_{x\in\mathcal{X}} \frac{\beta(x)}{\alpha(x)} < \infty$. Then the regret of Algorithm 2 can be bounded as

$$\hat{L}_T \le \kappa\, L|\mathcal{X}|\rho_B(T, \mathcal{A}).$$

Proof: Following the proof of Theorem 4 we obtain, for any l,

$$\begin{aligned} \sum_{x_l\in\mathcal{X}_l} \mu_T^*(x_l)\big(V_T^{+}(x_l) - \mathbb{E}\left[\mathbf{V}_T(x_l)\right]\big) &= \sum_{x_l\in\mathcal{X}_l} \mu_T^*(x_l)\mathbb{E}\left[\sum_{t\in\mathbf{T}_{x_l}} \frac{\mathbf{q}_t(x_l, \pi_T^*(x_l)) - \mathbf{v}_t(x_l)}{\boldsymbol{\mu}_t(x_l)}\right] \\ &\le \sum_{x_l\in\mathcal{X}_l} \beta(x_l)\frac{1}{\alpha(x_l)}(L - l)\rho_B(T, \mathcal{A}) \\ &\le |\mathcal{X}_l|\kappa L\rho_B(T, \mathcal{A}). \end{aligned}$$

Summing up for all l finishes the proof. ■

In particular, if we use **Exp3** (as described in Section 6.8 of Cesa-Bianchi and Lugosi 2006) as the bandit algorithm B, we can prove regret bounds that have slightly better dependence on α. The proof of the results, given in the following theorem, follows closely the derivation of the original regret bound of the **Exp3** algorithm (Auer et al., 2002a) and will be given in details in an extended version of this paper.

Theorem 6 *Assume that the conditions of Theorem 4 hold and the bandit algorithm B is the **Exp3** algorithm with parameters $0 < \gamma \le 1$ and $0 < \eta \le \frac{\alpha\gamma}{|\mathcal{A}|(L-l_x)}$. Then, if Algorithm 2 is used, for each state $x \in \mathcal{X}$ we have*

$$\mathbb{E}\left[\mathbf{Q}_T(x,a) - \mathbf{V}_T(x)\right] \le \left(\gamma + (e-2)\eta\frac{L-l_x}{\alpha}|\mathcal{A}|\right)(L-l_x)T + \frac{\ln|\mathcal{A}|}{\eta}.$$

An optimal choice of γ and η yields the following bound on the regret:

$$\hat{L}_T \le \frac{L(L+1)}{2}\sqrt{\frac{T|\mathcal{A}|\ln|\mathcal{A}|(e-2)}{\alpha}}.$$

Furthermore, let $\kappa' = \max_{x\in\mathcal{X}} \frac{\beta(x)}{\sqrt{\alpha(x)}} < \infty$ where $\alpha(x)$ and $\beta(x)$ are defined in Theorem 5. Then

$$\hat{L}_T \le \kappa'\, L|\mathcal{X}|\sqrt{T|\mathcal{A}|\ln|\mathcal{A}|(e-2)}.$$

In the above results we used the assumption that any stationary policy induces a distribution that visits each state with positive probability. However, this assumption may be too restrictive in many situations. If we only require that each state is reachable with positive probability for an adequately chosen policy, then using **Exp3** in our algorithm with different γ at each layer yields a consistent strategy with sublinear regret, although the convergence rate becomes very slow.

Theorem 7 *Let*

$$p_{\min} = \min_{x\in\mathcal{X}_l, a\in\mathcal{A}, x'\in\mathcal{X}_{l+1}, 1\le l\le L-1, P(x'|x,a)>0} P(x'|x,a)$$

*and assume that for each state x there is a policy π such that $\mu_\pi(x) > 0$. If Algorithm 2 is run with the **Exp3** algorithm with parameters $\gamma_l = T^{-2^{-l-1}}$ and $\eta_l = \frac{\gamma_l \prod_{i=1}^{l-1}(p_{\min}\gamma_i/|\mathcal{A}|)}{|\mathcal{A}|(L-l)}$ for each state $x_l \in \mathcal{X}_l$, then*

$$\hat{L}_T/T \le \frac{L(L+1)}{2}\left((e-1) + \frac{|\mathcal{A}|^{L+1}\ln|\mathcal{A}|}{p_{\min}^L}\right)T^{-2^{-L-1}}$$

Proof: For any l our assumptions imply $\boldsymbol{\mu}_t(x_l) \ge \prod_{i=0}^{l-1}(p_{\min}\gamma_i/|\mathcal{A}|)$. Therefore, similarly to the first statement of Theorem 6, for all x and a we have

$$\begin{aligned}
\mathbb{E}\left[\mathbf{Q}_T(x,a) - \mathbf{V}_T(x)\right] &\le \left(\gamma_{l_x} + \frac{(e-2)\eta_{l_x}(L-l_x)|\mathcal{A}|}{\prod_{i=0}^{l_x-1}(p_{\min}\gamma_i/|\mathcal{A}|)}\right)(L-l_x)T + \frac{\ln|\mathcal{A}|}{\eta_{l_x}} \\
&= \left((L-l_x)(e-1) + \frac{(L-l_x)|\mathcal{A}|^{l_x+1}\ln|\mathcal{A}|}{p_{\min}^{l_x}}\right)T^{1-2^{-l_x-1}}
\end{aligned}$$

by straightforward calculations. Summing up the above formula for $l_x = 0, \ldots, L-1$ proves the proposition by Lemma 1. ■

So far the regret of our algorithm was measured relative to the best fixed policy. On the other hand, in our motivating examples it may be the case that the best policy changes over time, and hence it is natural to compare our performance to the best time varying policy. Let $\pi_{1:T} = (\pi_1, \pi_2, \ldots, \pi_T)$ be a sequence of policies, and let $R_T(\pi_{1:T})$ denote the expected return, after T episodes, of the algorithm that applies policy π_t at episode t. Our goal is to minimize the expected loss $R_T(\pi_{1:T}) - \hat{R}_T$ relative to $\pi_{1:T}$.

Clearly, it is not possible to provide a uniform bound on this loss, as, in general, it is harder to achieve the performance of an algorithm that changes the employed policy more often (the extreme situation is when the policy changes in each time instant). In the following we will give an algorithm that bounds the tracking regret with the help of the complexity of $\pi_{1:T}$ that can be defined as

$$C(\pi_{1:T}) = 1 + |\{t : \pi_t \ne \pi_{t+1}, 1 \le t \le T-1\}|.$$

That is $C(\pi_{1:T})$ is the number of times the employed policy changes between consecutive episodes.

While this problem seems much harder than the ones considered before, the tracking algorithms for the prediction framework help us solve this problem. Several algorithms are known for the full information case with vanishing tracking regret under various conditions and with different rewards, see, for example, Willems (1996); Helmbold and Warmuth (1998); Shamir and Merhav (1999); Vovk (1999); György et al. (2008). These methods can be extended to the bandit case as well, see, for example, Auer et al. (2002a). Assume that we have an algorithm BT for the bandit sequential prediction problem (that is, when there is only one state) that satisfies, for every policy sequence $\pi_{1:T}$,

$$R_T(\pi_{1:T}) - \hat{R}_T \le \rho_{BT}(T, \mathcal{A}, C(\pi_{1:T})) \tag{11}$$

with some function $\rho_{BT}(T, \mathcal{A}, C(\pi_{1:T}))$ that is a nondecreasing function of T, $\mathcal{A}$, and $C(\pi_{1:T})$. Then using such an algorithm as the expert algorithm B in Algorithm 2 solves the tracking problem in the following sense.

Theorem 8 *Assume that BT is a multi-armed bandit algorithm that satisfies the regret bound* (11). *If κ, defined in Theorem 5, is finite and Algorithm 2 is used with the bandit algorithm BT, then the regret relative to any fixed sequence of policies $\pi_{1:T}$ can be bounded as*

$$R_T(\pi_{1:T}) - \hat{R}_T \leq \kappa L|\mathcal{X}|\rho_{BT}(T, \mathcal{A}, C(\pi_{1:T})).$$

Remark: In particular, if the **Exp3.S** algorithm of Auer et al. (2002a) is used, then if T is known in advance and is used optimally in setting the parameters of the algorithm, we obtain

$$R_T(\pi_{1:T}) - \hat{R}_T \leq \kappa L|\mathcal{X}| \left(C(\pi_{1:T})\sqrt{|\mathcal{A}|T\ln(|\mathcal{A}|T)} + 2e\sqrt{\frac{|\mathcal{A}|T}{\ln(|\mathcal{A}|T)}} \right).$$

Furthermore, if a bound C on the complexity of $\pi_{1:T}$ is known in advance (this is useful, if the complexity of the optimal $\pi_{1:T}$ is bounded), then using this value in setting the parameters of **Exp3.S**, we obtain

$$R_T(\pi_{1:T}) - \hat{R}_T \leq \kappa L|\mathcal{X}|\sqrt{e-1}\sqrt{|\mathcal{A}|T(C\ln(|\mathcal{A}|T) + e)}.$$

Proof: A simple generalization of Lemma 1 yields

$$\begin{aligned} R_T(\pi_{1:T}) - \hat{R}_T &= V_T^{\pi_{1:T}}(x_0) - \mathbb{E}\left[\mathbf{V}_T(x_0)\right] = \sum_{t=1}^{T} v_t^{\pi_t}(x_0) - \mathbb{E}\left[\mathbf{V}_T(x_0)\right] \\ &= \sum_{l=0}^{L-1} \sum_{x_l \in \mathcal{X}_l} \sum_{t=1}^{T} \mu_t(x_l)\mathbb{E}\left[\mathbf{q}_t(x_l, \pi_t(x_l)) - \mathbf{v}_t(x_l)\right] \end{aligned}$$

Now we have, for any x, similarly to (9),

$$\mathbb{E}\left[\mathbf{q}_t(x, \pi_t(x)) - \mathbf{v}_t(x)\right] = \mathbb{E}\left[\hat{\mathbf{q}}_t(x, \pi_t(x)) - \hat{\mathbf{v}}_t(x)\right].$$

Therefore, similarly to (10), we obtain

$$\begin{aligned} \sum_{t=1}^{T} \mu_t(x)\mathbb{E}\left[\mathbf{q}_t(x, \pi_t(x)) - \mathbf{v}_t(x)\right] &= \sum_{t=1}^{T} \mu_t(x)\mathbb{E}\left[\frac{\mathbf{q}_t(x, \pi_t(x)) - \mathbf{v}_t(x_l)}{\boldsymbol{\mu}_t(x)}\right] \\ &\leq \beta(x)\mathbb{E}\left[\sum_{t \in \mathbf{T}_x} \frac{\mathbf{q}_t(x, \pi_t(x)) - \mathbf{v}_t(x)}{\boldsymbol{\mu}_t(x)}\right] \end{aligned}$$

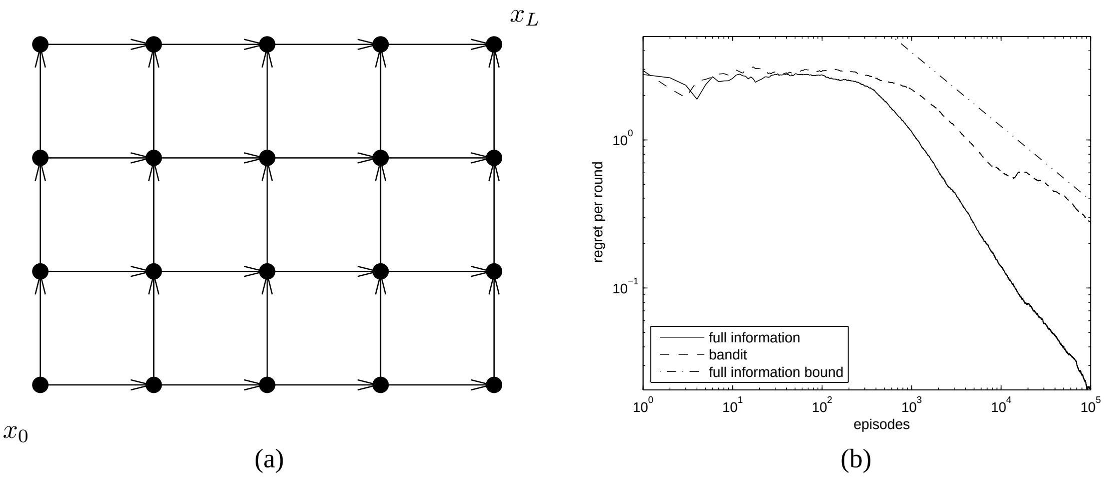
Finally, (11) and Lemma 2 yields, as at the end of the proof of Theorem 4,

$$\mathbb{E}\left[\sum_{t \in \mathbf{T}_x} \frac{\mathbf{q}_t(x, \pi_t(x)) - \mathbf{v}_t(x)}{\boldsymbol{\mu}_t(x)}\right] \leq \frac{L}{\alpha(x)}\rho_{BT}(T, \mathcal{A}, C(\pi_{1:T}))$$

since $\rho_{BT}(T, \mathcal{A}, C)$ is an increasing function of C by assumption. Combining the above results finishes the proof. ■

6 Simulations

We have run our experiments on a grid world of size 10×10, where in each episode the agent has to find the shortest path from the lower left corner to the upper right corner. The agent has two actions: Both make the agent move right or up, the “right” (“up”) action makes the agent move right (respectively, “up”) with probability 0.7, while it makes it move “up” (respectively, “right”) with probability 0.3. That is, we have $L = 20$, $|\mathcal{X}| = 100$, $\alpha = 0.3^{10}$, $\kappa = (0.7/0.3)^{10}$ (the values of α and κ correspond to the top-left and bottom-right corners). The experiment is run with $T = 100,000$, rewards are set randomly 20 times at episodes $t = 1$, 5000, 10000, $\ldots$ for all x, a, and change linearly in between. We have simulated the policies generated by EWA for the full information case, and the policies generated by **Exp3** for the bandit case. An example of the grid-world (of smaller size) and the results of a typical simulation are shown in Figure 2.



(a) (b)

Figure 2: (a) An example of a grid-world. (b) The average regret in an episode of the proposed algorithms as the function of the number of episodes in a simple MDP.

7 Conclusions and future work

In this paper we considered the problem of online learning in loop-free stochastic-shortest path problems in a bandit setting when only the reward of the current transitions is available for measurement. The per episode complexity of our algorithm is $\mathcal{O}(|\mathcal{A}|\,|\mathcal{X}|^2)$ and the algorithm is easy to implement. According to our knowledge, ours is the first algorithm that can be implemented efficiently and which is known to achieve an $\mathcal{O}(\sqrt{T|\mathcal{A}|})$ regret in the bandit setting, under the assumption that every policy reaches every state with positive probability. Unfortunately, the regret bound scales with the inverse of the minimal such probability, which is clearly undesirable in many situations. To alleviate this problem, variants of the original bound have been developed that may be preferred in certain specific cases. For the case when this latter condition does not hold, we proposed an algorithm whose expected regret vanishes over time. We view our results as a step towards algorithms that work efficiently and which can be implemented efficiently. However, much work remains to be done.

As for immediate future work, obvious directions include extending our results to the case of unichain MDPs setting, or, less ambitiously, to the case when the stochastic shortest-path problem may have loops. Although one can construct an unbiased estimate of the action values by plugging in an unbiased estimate of the rewards, these estimates are not of the form (5), thus our analysis does not apply. It is nontrivial whether a proper estimate of the action values can be found; even with a positive answer there are further obstacles to eliminate (e.g., the change rate of the distributions generated by the applied bandit algorithm has to be controlled in order to be able to apply the analysis of Even-Dar et al., 2009). Alternate directions to extend our results include the case of unknown transition probabilities, partial monitoring, high probability bounds, or when the state and action space are too large to keep a value for each of them, in which case one must resort to some form of function approximation, just to mention a few.

Acknowledgments

This work was supported in part by the PASCAL2 Network of Excellence under EC grant no. 216886, the Hungarian Scientific Research Fund and the Hungarian National Office for Research and Technology (OTKA-NKTH CNK 77782), NSERC, the Alberta Ingenuity Centre for Machine Learning and iCore.

References

Allenberg, C., Auer, P., Györfi, L., and Ottucsák, G. (2006). Hannan consistency in on-line learning in case of unbounded losses under partial monitoring. In *ALT*, pages 229–243.

Audibert, J.-Y. and Bubeck, S. (2009). Minimax policies for bandits games. In *COLT 2009*.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on the Foundations of Computer Science*, pages 322–331. IEEE press.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002a). The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77.

Auer, P., Cesa-Bianchi, N., and Gentile, C. (2002b). Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1).

Awerbuch, B. and Kleinberg, R. D. (2004). Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In Babai, L., editor, *STOC*, pages 45–53. ACM.

Bartlett, P. L., Dani, V., Hayes, T. P., Kakade, S., Rakhlin, A., and Tewari, A. (2008). High-probability regret bounds for bandit online linear optimization. In Servedio, R. A. and Zhang, T., editors, *21st Annual Conference on Learning Theory (COLT 2008)*, pages 335–342. Omnipress.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA.

Cesa-Bianchi, N. and Lugosi, G. (2009). Combinatorial bandits. In *Proceedings of the Twenty-Second Conference on Computational Learning Theory*.

Dani, V., Hayes, T. P., and Kakade, S. (2008). The price of bandit information for online optimization. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T., editors, *Advances in Neural Information Processing Systems 20*, pages 345–352. MIT Press.

Even-Dar, E., Kakade, S. M., and Mansour, Y. (2005). Experts in a markov decision process. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 401–408.

Even-Dar, E., Kakade, S. M., and Mansour, Y. (2009). Online Markov decision processes. *Mathematics of Operations Research*, 34(3):726–736.

Freund, Y. and Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139.

György, A., Linder, T., and Lugosi, G. (2008). Tracking the best quantizer. *IEEE Transactions on Information Theory*, 54:1604–1625.

György, A., Linder, T., Lugosi, G., and Ottucsák, G. (2007). The on-line shortest path problem under partial monitoring. *J. Mach. Learn. Res.*, 8:2369–2403.

Helmbold, D. and Warmuth, M. (1998). Tracking the best expert. *Machine Learning*, 32(2):151–178.

Kalai, A. and Vempala, S. (2003). Efficient algorithms for the online decision problem. In Schölkopf, B. and Warmuth, M., editors, *Proceedings of the 16th Annual Conference on Learning Theory and the 7th Kernel Workshop, COLT-Kernel 2003*, pages 26–40, New York, USA. Springer.

Littlestone, N. and Warmuth, M. (1994). The weighted majority algorithm. *Information and Computation*, 108:212–261.

McMahan, H. B. and Blum, A. (2004). Online geometric optimization in the bandit setting against an adaptive adversary. In Shawe-Taylor, J. and Singer, Y., editors, *COLT*, volume 3120 of *Lecture Notes in Computer Science*, pages 109–123. Springer.

Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience.

Shamir, G. I. and Merhav, N. (1999). Low-complexity sequential lossless coding for piecewise-stationary memoryless sources. *IEEE Trans. Inform. Theory*, IT-45:1498–1519.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An Introduction*. MIP Press.

Vovk, V. (1990). Aggregating strategies. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, pages 372–383.

Vovk, V. (1999). Derandomizing stochastic prediction strategies. *Machine Learning*, 35(3):247–282.

Willems, F. M. J. (1996). Coding for a binary independent piecewise-identically-distributed source. *IEEE Trans. Inform. Theory*, IT-42:2210–2217.

Yu, J. Y. and Mannor, S. (2009a). Arbitrarily modulated Markov decision processes. In *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*. IEEE Press.

Yu, J. Y. and Mannor, S. (2009b). Online learning in Markov decision processes with arbitrarily changing rewards and transitions. In *GameNets'09: Proceedings of the First ICST international conference on Game Theory for Networks*, pages 314–322, Piscataway, NJ, USA. IEEE Press.

Yu, J. Y., Mannor, S., and Shimkin, N. (2009). Markov decision processes with arbitrary reward processes. *Mathematics of Operations Research*, 34(3):737–757.