The Shortest Path Problem Under Partial Monitoring *

András György[1], Tamás Linder[2,1], and György Ottucsák[3]

[1] Informatics Laboratory, Computer and Automation Research Institute
of the Hungarian Academy of Sciences,
Lágymányosi u. 11, Budapest, Hungary, H-1111
gya@szit.bme.hu

[2] Department of Mathematics and Statistics,
Queen's University, Kingston, Ontario,
Canada K7L 3N6
linder@mast.queensu.ca

[3] Department of Computer Science and Information Theory,
Budapest University of Technology and Economics,
Magyar Tudósok Körútja 2., Budapest, Hungary, H-1117
oti@szit.bme.hu

Abstract. The on-line shortest path problem is considered under partial monitoring scenarios. At each round, a decision maker has to choose a path between two distinguished vertices of a weighted directed acyclic graph whose edge weights can change in an arbitrary (adversarial) way such that the loss of the chosen path (defined as the sum of the weights of its composing edges) be small. In the multi-armed bandit setting, after choosing a path, the decision maker learns only the weights of those edges that belong to the chosen path. For this scenario, an algorithm is given whose average cumulative loss in n rounds exceeds that of the best path, matched off-line to the entire sequence of the edge weights, by a quantity that is proportional to $1/\sqrt{n}$ and depends only polynomially on the number of edges of the graph. The algorithm can be implemented with linear complexity in the number of rounds n and in the number of edges. This result improves earlier bandit-algorithms which have performance bounds that either depend exponentially on the number of edges or converge to zero at a slower rate than $O(1/\sqrt{n})$. An extension to the so-called label efficient setting is also given, where the decision maker is informed about the weight of the chosen path only with probability $\epsilon < 1$. Applications to routing in packet switched networks along with simulation results are also presented.

1 Introduction

In a typical sequential decision problem, a decision maker has to perform a sequence of actions. After each action the decision maker suffers some loss, de-

* This research was supported in part by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, the Mobile Innovation Center of Hungary, by the Natural Sciences and Engineering Research Council (NSERC) of Canada, and by the Hungarian Inter-University Center for Telecommunications and Informatics (ETIK).

pending on the response (or state) of the environment, and its goal is to minimize its cumulative loss over a sufficiently long period of time. In the adversarial setting no probabilistic assumption is made on how the losses corresponding to different actions are generated. In particular, the losses may depend on the previous actions of the decision maker, whose goal is to perform well relative to a set of experts for any possible behavior of the environment. More precisely, the aim of the decision maker is to achieve asymptotically the same average loss (per round) as the best expert.

The basic theoretical results in this topic were pioneered by Blackwell [4] and Hannan [17], and brought to the attention of the machine learning community in the 1990's by Vovk [25], Littlestone and Warmuth [21], and Cesa-Bianchi *et al.* [6]. These results show that for any bounded loss function, if the decision maker has access to the past losses of all experts, then it is possible to construct on-line algorithms that perform, for any possible behavior of the environment, almost as well as the best of N experts. Namely, for these algorithms the per round cumulative loss of the decision maker is at most as large as that of the best expert plus a quantity proportional to $\sqrt{\ln N/n}$ for any bounded loss function, where n is the number of rounds in the decision game. The logarithmic dependence on the number of experts makes it possible to obtain meaningful bounds even if the pool of experts is very large. However, the basic prediction algorithms, such as weighted average forecasters, have a computational complexity that is proportional to the number of experts, and they are therefore practically infeasible when the number of experts is very large.

In certain situations the decision maker has only limited knowledge about the losses of all possible actions. For example, it is often natural to assume that the decision maker gets to know only the loss corresponding to the action it has made, and has no information about the loss it would have suffered had it made a different decision. This setup is referred to as the *multi-armed bandit problem*, and was solved by Auer *et al.* [1] and Cesa-Bianchi and Lugosi [7], who gave an algorithm whose average loss exceeds that of the best expert at most by an amount proportional to $\sqrt{N \ln N/n}$. Note that, compared to the *full information case* described above where the losses of all possible actions are revealed to the decision maker, there is an extra $\sqrt{N}$ term in the performance bound, which seriously limits the usefulness of the algorithm if the number of experts is large.

Another interesting example for the limited information case is the so-called *label efficient decision problem*, in which it is too costly to observe the state of the environment, and so the decision maker can query the losses of all possible actions for only a limited number of times. A recent result of Cesa-Bianchi, Lugosi, and Stoltz [8] shows that in this case, if the decision maker can query the losses m times during a period of length n, then it can achieve $O(\sqrt{\ln N/m})$ average excess loss relative to the best expert.

In many applications the set of experts has a certain structure that may be exploited to construct efficient on-line decision algorithms. The construction of such algorithms has been of great interest in computational learning theory. A partial list of works dealing with this problem includes Herbster and Warmuth [19], Vovk [26], Bousquet and Warmuth [5], Helmbold and Schapire [18], Takimoto and Warmuth [24], Kalai and Vempala [20], György, Linder, and Lugosi [12–14]. For a more complete survey, see Cesa-Bianchi and Lugosi [7, Chapter 5].

In this paper we discuss the on-line shortest path problem, a representative example of structured expert classes that has received attention in the literature for its many applications, including, among others, routing in communication networks, see, e.g., Takimoto and Warmuth [24], Awerbuch *et al.* [2], or György and Ottucsák [16], and adaptive quantizer design in zero-delay lossy source coding, see, György, Linder, and Lugosi [12, 13, 15]. In this problem, given is a weighted directed (acyclic) graph whose edge weights can change in an arbitrary manner, and the decision maker has to pick in each round a path between two given vertices, such that the weight of this path (the sum of the weights of its composing edges) be as small as possible.

Efficient solutions, with time and space complexity proportional to the number of edges rather than to the number of paths (the latter typically being exponential in the number of edges), have been given in the full information case, where in each round the weights of all the edges are revealed after a path has been chosen, see, e.g., Mohri [23], Takimoto and Warmuth [24], Kalai and Vempala [20], and György, Linder, and Lugosi [14].

In the bandit setting, where only the weights of the edges composing the chosen path are revealed to the decision maker, if one applies the general bandit algorithm of Auer *et al.* [1], then the resulting bound will be too large to be of practical use because of its square-root-type dependence on the number of paths N. On the other hand, utilizing the special graph structure in the problem, Awerbuch and Kleinberg [3] and McMahan and Blum [22] managed to get rid of the exponential dependence on the number of edges in the performance bound by extending black box predictors, and specifically the follow-the-perturbed-leader algorithm of Hannan [17] and the exponentially weighted average predictor [21], to the multi-armed bandit setting. However, their bounds do not have the right $O(1/\sqrt{n})$ dependence on the number of rounds.

In this paper we provide an extension of the bandit algorithm of Auer *et al.* [1] unifying the advantages of the above approaches, with performance bound that is only polynomial in the number of edges, and converges to zero at the right $O(1/\sqrt{n})$ rate as the number of rounds increases.

In the following, first we define formally the on-line shortest path problem in Section 2, then extend it to the multi-armed bandit setting in Section 3. Our new algorithm for the shortest path problem in the bandit setting is given in Section 4 together with its performance analysis. The algorithm is extended to solve the shortest path problem in a combined label efficient multi-armed bandit setting in Section 5. Simulation results are presented in Section 6.

2 The shortest path problem

Consider a network represented by a set of nodes connected by edges, and assume that we have to send a stream of packets from a source node to a destination node. At each time slot a packet is sent along a chosen route connecting source and destination. Depending on the traffic, each edge in the network may have a different delay, and the total delay the packet suffers on the chosen route is the sum of delays of the edges composing the route. The delays may change from one time slot to the next one in an arbitrary way, and our goal is to find a way of choosing the route in each time slot such that the sum of the total delays over time is not significantly more than that of the best fixed route in the network. This adversarial version of the routing problem is most useful when the delays on the edges can change very dynamically, even depending on our previous routing decisions. This is the situation in the case of ad-hoc networks, where the network topology can change rapidly, or in certain secure networks, where the algorithm has to be prepared to handle denial of service attacks, that is, situations where willingly malfunctioning nodes and links increase the delay, see, e.g., Awerbuch *et al.* [2].

This problem can be naturally cast as a sequential decision problem in which each possible route is represented by an action. However, the number of routes is typically exponentially large in the number of edges, and therefore computationally efficient algorithms are called for. Two solutions of very different flavors have been proposed. One of them is based on a follow-the-perturbed-leader forecaster, see Kalai and Vempala [20], while the other is based on an efficient computation of the exponentially weighted average forecaster, see, for example, Takimoto and Warmuth [24]. Both solutions have different advantages and may be generalized in different directions.

To formalize the problem, consider a (finite) directed acyclic graph with a set of edges $E = \{e_1, \ldots, e_{|E|}\}$ and a set of vertices V. Thus, each edge $e \in E$ is an ordered pair of vertices (v_1, v_2). Let u and v be two distinguished vertices in V. A *path* from u to v is a sequence of edges $e^{(1)}, \ldots, e^{(k)}$ such that $e^{(1)} = (u, v_1)$, $e^{(j)} = (v_{j-1}, v_j)$ for all $j = 2, \ldots, k-1$, and $e^{(k)} = (v_{k-1}, v)$, and let $\mathcal{R} = \{\boldsymbol{i}_1, \ldots, \boldsymbol{i}_N\}$ denote the set of all such paths. For simplicity, we assume that every edge in E is on some path from u to v and every vertex in V is an endpoint of an edge.

In each round $t = 1, \ldots, n$ of the decision game, the decision maker chooses a path $\boldsymbol{I}_t$ among all paths from u to v. Then a loss $\ell_{e,t} \in [0, 1]$ is assigned to each edge $e \in E$. We write $e \in \boldsymbol{i}$ if the edge $e \in E$ belongs to the path $\boldsymbol{i} \in \mathcal{R}$, and with a slight abuse of notation the loss of a path $\boldsymbol{i}$ at time slot t is also represented by $\ell_{\boldsymbol{i},t}$ (however, the meaning of the subscript of ℓ will always be clear from the context). Then $\ell_{\boldsymbol{i},t}$ is given as

$$\ell_{\boldsymbol{i},t} = \sum_{e \in \boldsymbol{i}} \ell_{e,t}$$

and therefore the cumulative loss of each path $\boldsymbol{i}$ takes the additive form

$$\sum_{s=1}^{t} \ell_{\boldsymbol{i},s} = \sum_{e \in \boldsymbol{i}} \sum_{s=1}^{t} \ell_{e,s}$$

where the inner sum on the right hand side is the loss accumulated by edge e during the first t rounds of the game.

It is well known that for a general loss sequence, the decision maker must be allowed to use randomization to be able to achieve the performance of the best expert, see, e.g., Cesa-Bianchi and Lugosi [7]. Therefore, the path $\boldsymbol{I}_t$ is chosen according to some distribution $\boldsymbol{p}_t$ over all paths from u to v. We study the normalized regret

$$\frac{1}{n}\left(\sum_{t=1}^{n}\ell_{\boldsymbol{I}_t,t}-\min_{\boldsymbol{i}\in\mathcal{R}}\sum_{t=1}^{n}\ell_{\boldsymbol{i},t}\right)$$

where the minimum is taken over all paths $\boldsymbol{i}$ from u to v.

For example, the exponentially weighted average forecaster [21], calculated over all possible paths, yields regret bound of the form

$$\frac{1}{n}\left(\sum_{t=1}^{n}\ell_{\boldsymbol{I}_t,t}-\min_{\boldsymbol{i}\in\mathcal{R}}\sum_{t=1}^{n}\ell_{\boldsymbol{i},t}\right)\le K\left(\sqrt{\frac{\ln N}{2n}}+\sqrt{\frac{\ln(1/\delta)}{2n}}\right)$$

with probability at least $1-\delta$, where N is the total number of paths from u to v in the graph and K is the length of the longest path.

3 The multi-armed bandit setting

In this section we discuss the "bandit" version of the shortest path problem. In this, in many applications more realistic problem, the decision maker has only access to the losses of those edges that are on the path it has chosen. That is, after choosing a path $\boldsymbol{I}_t$ at time t, the value of the loss $\ell_{e,t}$ is revealed to the forecaster if and only if $e\in\boldsymbol{I}_t$. For example, in the routing problem it means that information is available on the delay of the route the packet is sent on, and not on other routes in the network.

Formally, the on-line shortest path problem in the multi-armed bandit setting is given as follows: at each time slot $t=1,\ldots,n$, the decision maker picks a path $\boldsymbol{I}_t\in\mathcal{R}$ form u to v. Then the environment assigns loss $\ell_{e,t}\in[0,1]$ to each edge $e\in E$, and the decision maker suffers loss $\ell_{\boldsymbol{I}_t,t}=\sum_{e\in\boldsymbol{I}_t}\ell_{e,t}$, and the losses $\ell_{e,t}$ are revealed for all $e\in\boldsymbol{I}_t$. Note that $\ell_{e,t}$ may depend on $\boldsymbol{I}_1,\ldots,\boldsymbol{I}_{t-1}$, the earlier choices of the decision maker.

For the general multi-armed bandit problem, Auer *et al.* [1] gave an algorithm, based on exponential weighting with a biased estimate of the gains defined, in our case, as $g_{\boldsymbol{i},t}=K-\ell_{\boldsymbol{i},t}$, combined with uniform exploration. Applying an improved version of this algorithm due to Cesa-Bianchi and Lugosi [7] to the on-line shortest path problem in the bandit setting results in a performance that can be bounded with probability at least $1-\delta$ for any $0<\delta<1$ and fixed time horizon n as

$$\frac{1}{n}\left(\sum_{t=1}^{n}\ell_{\boldsymbol{I}_t,t}-\min_{\boldsymbol{i}\in\mathcal{R}}\sum_{t=1}^{n}\ell_{\boldsymbol{i},t}\right)\le\frac{11K}{2}\sqrt{\frac{N\ln(N/\delta)}{n}}+\frac{K\ln N}{2n}\ .$$

However, this bound is unacceptable in our scenario because, unlike in the full information case when a simple usage of the exponentially weighted average forecaster yielded a good performance bound, here the dependence on the number of all paths N is not merely logarithmic. In order to achieve a bound that does not grow exponentially with the number of edges of the graph, it is imperative to make use of the dependence structure of the losses of the different actions (i.e., paths). Awerbuch and Kleinberg [3] and McMahan and Blum [22] attempted to do this by extending low complexity predictors, such as the follow-the-perturbed-leader forecaster [17], [20] to the bandit setting. However, the obtained bounds do not have the right $O(1/\sqrt{n})$ decay in terms of the number of rounds.

4 A bandit algorithm for shortest paths

In the following we describe a carefully defined variant of the bandit algorithm of [1] that achieves the desired performance for the shortest path problem in the bandit setting. The new algorithm utilizes the fact that when the losses of the edges of the chosen path are revealed, then this also provides some information about the losses of each path sharing common edges with the chosen path.

For each edge $e \in E$, introduce *gains* $g_{e,t} = 1 - \ell_{e,t}$, and for each path $\boldsymbol{i} \in \mathcal{R}$, similarly to the losses, let the gain be the sum of the gains of the edges of the path, that is, let $g_{\boldsymbol{i},t} = \sum_{e \in \boldsymbol{i}} g_{e,t}$. The conversion from losses to gains is done in order to facilitate the subsequent performance analysis, see, e.g. [7]. To simplify the conversion, we assume that each path $\boldsymbol{i} \in \mathcal{R}$ is of the same length K for some $K > 0$. Note that although this assumption may seem to be restrictive at the first glance, from each acyclic directed graph (V, E) one can construct a new graph with adding at most $(K-2)(|V|-2)+1$ vertices and edges (with constant weight zero) to the graph without modifying the weights of the paths such that each path from u to v will be of length K, where K denotes the length of the longest path of the original graph. As typically $|E| = O(|V|^2)$, the size of the graph is usually not increased substantially.

A main feature of the algorithm below is that the gains are estimated for each edge and not for each path. This modification results in an improved upper bound on the performance with the number of edges in place of the number of paths. Moreover, using dynamic programming as in Takimoto and Warmuth [24], the algorithm can be computed efficiently. Another important ingredient of the algorithm is that one needs to make sure that every edge is sampled sufficiently often. To this end, we introduce a set $\mathcal{C}$ of *covering paths* with the property that for each edge $e \in E$ there is a path $\boldsymbol{i} \in \mathcal{C}$ such that $e \in \mathcal{C}$. Observe that one can always find such a covering set of cardinality $|\mathcal{C}| \le |E|$.

Note that the algorithm of [1] is a special case of the algorithm below: For any multi-armed bandit problem with N experts, one can define a graph with two vertices u and v, and N directed edges from u to v with weights corresponding to the losses of the experts. The solution of the shortest path problem in this case is equivalent to that of the original bandit problem, with choosing expert $\boldsymbol{i}$ if the corresponding edge is chosen. For this graph, our algorithm reduces to the original algorithm of [1].

A BANDIT ALGORITHM FOR SHORTEST PATHS

Parameters: real numbers $\beta > 0$, $0 < \eta, \gamma < 1$.
Initialization: Set $w_{e,0} = 1$ for each $e \in E$, $\boldsymbol{w}_{\boldsymbol{i},0} = 1$ for each $\boldsymbol{i} \in \mathcal{R}$, and $\overline{W}_0 = |\mathcal{R}|$. For each round $t = 1, 2, \ldots$

(a) Choose a path $\boldsymbol{I}_t$ according to the distribution $\boldsymbol{p}_t$ on $\mathcal{R}$, defined by

$$p_{\boldsymbol{i},t} = \begin{cases} (1-\gamma)\frac{\boldsymbol{w}_{\boldsymbol{i},t-1}}{\overline{W}_{t-1}} + \frac{\gamma}{|\mathcal{C}|} & \text{if } \boldsymbol{i} \in \mathcal{C} \\ (1-\gamma)\frac{\boldsymbol{w}_{\boldsymbol{i},t-1}}{\overline{W}_{t-1}} & \text{if } \boldsymbol{i} \notin \mathcal{C}. \end{cases}$$

(b) Compute the probability of choosing each edge e as

$$q_{e,t} = \sum_{\boldsymbol{i}:e\in\boldsymbol{i}} p_{\boldsymbol{i},t} = (1-\gamma)\frac{\sum_{\boldsymbol{i}:e\in\boldsymbol{i}} \boldsymbol{w}_{\boldsymbol{i},t-1}}{\overline{W}_{t-1}} + \gamma\frac{|\{\boldsymbol{i}\in\mathcal{C} : e \in \boldsymbol{i}\}|}{|\mathcal{C}|}.$$

(c) Calculate the estimated gains

$$g'_{e,t} = \begin{cases} \frac{g_{e,t}+\beta}{q_{e,t}} & \text{if } e \in \boldsymbol{I}_t \\ \frac{\beta}{q_{e,t}} & \text{otherwise.} \end{cases}$$

(d) Compute the updated weights

$$w_{e,t} = w_{e,t-1}e^{\eta g'_{e,t}}$$
$$\boldsymbol{w}_{\boldsymbol{i},t} = \prod_{e\in\boldsymbol{i}} w_{e,t} = \boldsymbol{w}_{\boldsymbol{i},t-1}e^{\eta g'_{\boldsymbol{i},t}}$$

where $g'_{\boldsymbol{i},t} = \sum_{e\in\boldsymbol{i}} g'_{e,t}$, and the sum of the total weights of the paths

$$\overline{W}_t = \sum_{\boldsymbol{i}\in\mathcal{R}} \boldsymbol{w}_{\boldsymbol{i},t}.$$

The analysis of the algorithm is based on that of the original algorithm of [1] with necessary modifications required to transform parts of the argument for edges from paths, and to utilize the connection between the gains of paths sharing common edges.

Theorem 1. *For any $\delta \in (0,1)$ and parameters $0 \le \gamma < 1/2$, $0 < \beta \le 1$, and $\eta > 0$ satisfying $2\eta K|\mathcal{C}| \le \gamma$, the performance of the algorithm defined above can be bounded with probability at least $1-\delta$ as*

$$\frac{1}{n}\left(\sum_{t=1}^{n} \ell_{\boldsymbol{I}_t,t} - \min_{\boldsymbol{i}\in\mathcal{R}}\sum_{t=1}^{n} \ell_{\boldsymbol{i},t}\right) \le K\gamma + 2\eta K^2|\mathcal{C}| + \frac{K}{n\beta}\ln\frac{|E|}{\delta} + \frac{\ln N}{n\eta} + |E|\beta.$$

In particular, choosing $\beta = \sqrt{\frac{K}{n|E|}\ln\frac{|E|}{\delta}}$, $\gamma = 2\eta K|\mathcal{C}|$, *and* $\eta = \sqrt{\frac{\ln N}{4nK^2|\mathcal{C}|}}$ *yields for all* $n \geq \max\left\{\frac{K}{|E|}\ln\frac{|E|}{\delta}, 4|\mathcal{C}|\ln N\right\}$,

$$\frac{1}{n}\left(\sum_{t=1}^{n}\ell_{\boldsymbol{I}_t,t} - \min_{\boldsymbol{i}\in\mathcal{R}}\sum_{t=1}^{n}\ell_{\boldsymbol{i},t}\right) \leq 2\sqrt{\frac{K}{n}}\left(\sqrt{4K|\mathcal{C}|\ln N} + \sqrt{|E|\ln\frac{|E|}{\delta}}\right).$$

Sketch of the proof. The proof of the theorem follows the main ideas of [1]. As usual, we start with bounding the quantity $\ln\frac{\overline{W}_n}{\overline{W}_0}$. The lower bound is obtained as

$$\ln\frac{\overline{W}_n}{\overline{W}_0} = \ln\sum_{\boldsymbol{i}\in\mathcal{R}} e^{\eta\sum_{t=1}^{n} g'_{\boldsymbol{i},t}} - \ln N \geq \eta\max_{\boldsymbol{i}\in\mathcal{R}}\sum_{t=1}^{n} g'_{\boldsymbol{i},t} - \ln N \tag{1}$$

where we used the fact that $\boldsymbol{w}_{\boldsymbol{i},n} = e^{\eta\sum_{t=1}^{n} g'_{\boldsymbol{i},t}}$.

On the other hand, from the conditions of the theorem it follows that $\eta g'_{\boldsymbol{i},t} \leq 1$ for all $\boldsymbol{i}$ and t, and so using the inequalities $\ln(x+1) \leq x$ for all $x > -1$ and $e^x < 1 + x + x^2$ for all $x \leq 1$, one can show for all $t \geq 1$ that

$$\ln\frac{\overline{W}_t}{\overline{W}_{t-1}} \leq \frac{\eta}{1-\gamma}\sum_{\boldsymbol{i}\in\mathcal{R}} p_{\boldsymbol{i},t}g'_{\boldsymbol{i},t} + \frac{\eta^2}{1-\gamma}\sum_{\boldsymbol{i}\in\mathcal{R}} p_{\boldsymbol{i},t}{g'}^2_{\boldsymbol{i},t}. \tag{2}$$

The sums on the right hand side can be bounded as

$$\sum_{\boldsymbol{i}\in\mathcal{R}} p_{\boldsymbol{i},t}g'_{\boldsymbol{i},t} = g_{\boldsymbol{I}_t,t} + |E|\beta \quad \text{and} \quad \sum_{\boldsymbol{i}\in\mathcal{R}} p_{\boldsymbol{i},t}{g'}^2_{\boldsymbol{i},t} \leq K(1+\beta)\sum_{e\in E} g'_{e,t}. \tag{3}$$

Summing (2) for $t = 1, \ldots, n$, and combining it with (1) and (3), it follows that

$$\sum_{t=1}^{n} g_{\boldsymbol{I}_t,t} \geq (1 - \gamma - \eta K(1+\beta)|\mathcal{C}|)\max_{\boldsymbol{i}\in\mathcal{R}}\sum_{t=1}^{n} g'_{\boldsymbol{i},t} - \frac{1-\gamma}{\eta}\ln N - n|E|\beta. \tag{4}$$

Now one can show based on [7, Lemma 6.7] that for any $\delta \in (0,1)$, $0 < \beta \leq 1$, and for all $e \in E$ we have

$$\mathbb{P}\left(\sum_{t=1}^{n} g_{e,t} > \sum_{t=1}^{n} g'_{e,t} + \frac{1}{\beta}\ln\frac{|E|}{\delta}\right) \leq \frac{\delta}{|E|}. \tag{5}$$

Then, applying the union bound, one can replace $\sum_{t=1}^{n} g'_{\boldsymbol{i},t}$ in (4) with $\sum_{t=1}^{n} g_{\boldsymbol{i},t}$ as

$$\sum_{t=1}^{n} g_{\boldsymbol{I}_t,t} \geq (1 - \gamma - \eta K(1+\beta)|\mathcal{C}|)\left(\max_{\boldsymbol{i}\in\mathcal{R}}\sum_{t=1}^{n} g_{\boldsymbol{i},t} - \frac{K}{\beta}\ln\frac{|E|}{\delta}\right) - \frac{\ln N}{\eta} - n|E|\beta$$

which holds with probability at least $1 - \delta$. Then, applying the conversions

$$\sum_{t=1}^{n}\ell_{\boldsymbol{I}_t,t} = Kn - \sum_{t=1}^{n} g_{\boldsymbol{I}_t,t} \qquad \text{and} \qquad \sum_{t=1}^{n}\ell_{\boldsymbol{i},t} = Kn - \sum_{t=1}^{n} g_{\boldsymbol{i},t},$$

after some algebra one obtains the first statement of the theorem. The second statement follows by substituting the optimized parameters given in the theorem. □

The algorithm can be implemented efficiently with time complexity $O(n|E|)$ and space complexity $O(|E|)$. The two complex steps of the algorithm are steps (a) and (b), both of which can be computed, similarly to Takimoto and Warmuth [24], using dynamic programming. To be able to perform these steps efficiently, first we have to order the vertices of the graph. Since we have an acyclic directed graph, its nodes can be labeled (in $O(|E|)$ time) from 1 to $|V|$ such that if $(v_1, v_2) \in E$ then $v_1 < v_2$, and $u = 1$ and $v = |V|$. For any pair of vertices $u_1 < v_1$ let $\mathcal{R}_{u_1,v_1}$ denote the set of paths from u_1 to v_1, and for any vertex $s \in V$, let

$$H_t(s) = \sum_{\boldsymbol{i} \in \mathcal{R}_{s,v}} \prod_{e \in \boldsymbol{i}} w_{e,t}$$

and

$$\widehat{H}_t(s) = \sum_{\boldsymbol{i} \in \mathcal{R}_{u,s}} \prod_{e \in \boldsymbol{i}} w_{e,t}.$$

Given the edge weights $\{w_{e,t}\}$, $H_t(s)$ can be computed recursively for $s = |V| - 1, \ldots, 1$, and $\widehat{H}_t(s)$ can be computed recursively for $s = 2, \ldots, |V|$ in $O(|E|)$ time (letting $H_t(v) = \widehat{H}_t(u) = 1$ by definition). In step (a), first one has to decide with probability γ whether $\boldsymbol{I}_t$ is generated according to the graph weights, or it is chosen uniformly from $\mathcal{C}$. If $\boldsymbol{I}_t$ is to be drawn according to the graph weights, it can be shown that its vertices can be chosen one by one such that if the first k vertices of $\boldsymbol{I}_t$ are $v_0 = u, v_1, \ldots, v_{k-1}$, then the next vertex of $\boldsymbol{I}_t$ can be chosen to be any $v_k > v_{k-1}$, satisfying $(v_{k-1}, v_k) \in E$, with probability $w_{(v_{k-1},v_k),t-1} H_{t-1}(v_k)/H_{t-1}(v_{k-1})$. The other computationally demanding step, namely step (b), can be performed easily by noting that for any edge (v_1, v_2),

$$\begin{aligned} q_{(v_1,v_2),t} = (1-\gamma) & \frac{\widehat{H}_{t-1}(v_1) w_{(v_1,v_2),t-1} H_{t-1}(v_2)}{H_{t-1}(u)} \\ & + \gamma \frac{|\{\boldsymbol{i} \in \mathcal{C} : (v_1, v_2) \in \boldsymbol{i}\}|}{|\mathcal{C}|}. \end{aligned}$$

5 Shortest path problem for a combination of the label efficient and the bandit settings

In this section we investigate a combination of the multi-armed bandit and the label efficient setting problems, where the gain of the chosen path is available only on request. Just as in the previous section, it is assumed that each path of the graph is of the same length K.

In the general label efficient decision problem, after taking the action, the decision maker has the option to query the losses of all possible actions (in the original problem formulation, the decision maker can query the response of the environment, referred to as "label", and can compute all losses from this information). To query the losses, the decision maker uses an i.i.d. sequence $S_1, S_2, \ldots, S_n$ of Bernoulli random variables with $\mathbb{P}(S_t = 1) = \epsilon$ and asks for the losses if $S_t = 1$. For this problem, Cesa-Bianchi *et al.* [8] proved an upper bound on the normalized regret of order $O(K\sqrt{\ln(4N/\delta)/(n\epsilon)})$ with probability at least $1 - \delta$.

We study a combined algorithm which, at each time slot t, queries the loss of the chosen path with probability ϵ (as in the label efficient case), and similarly to the multi-armed bandit case, computes biased estimates $g'_{\boldsymbol{i},t}$ of the true gains $g_{\boldsymbol{i},t}$. This combination is motivated by some realistic applications, where the information is costly in some sense, i.e., the request is allowed only for a limited number of times.

The model of label-efficient decisions is well suited to a particular packet switched network model, called the cognitive packet network, which was introduced by Gelenbe *et al.* [10, 11]. In these networks, capabilities for routing and flow control are concentrated in packets. In particular, one type of packets, called smart packets, do not transport any useful data, but are used to explore the network (e.g. the delay of the chosen path). The other type of packets are data packets, which do not collect information about their paths, but transport useful data. In this model the task of the decision maker is to send packets from the source to the destination over routes with minimum average transmission delay (or packet loss). In this scenario, smart packets are used to query the delay of the chosen path. However, as these packets do not transport information, there is a tradeoff between the number of queries and the utilization of the network. If data packets are α times larger than smart packets on the average (note that typically $\alpha \gg 1$), then $\epsilon/(\epsilon + \alpha(1-\epsilon))$ is the proportion of the bandwidth sacrificed for well informed routing decisions.

The algorithm differs from our bandit algorithm of the previous section only in step (c), which is modified in the spirit of [8]. The modified step is given below:

MODIFIED STEP FOR THE LABEL EFFICIENT BANDIT ALGORITHM FOR SHORTEST PATHS

(c') Draw a Bernoulli random variable S_t with $\mathbb{P}(S_t = 1) = \epsilon$, and compute the estimated gains

$$g'_{e,t} = \begin{cases} \frac{g_{e,t}+\beta}{q_{e,t}\epsilon} & \text{if } e \in \boldsymbol{I}_t \text{ and } S_t = 1 \\ \frac{\beta}{q_{e,t}\epsilon} & \text{if } e \notin \boldsymbol{I}_t \text{ and } S_t = 1 \\ 0 & \text{otherwise.} \end{cases}$$

The performance of the algorithm is analyzed in the next theorem, which can be viewed as a combination of Theorem 1 in the preceding section and Theorem 2 of [8].

Theorem 2. *For any* $\delta \in (0,1)$, $\epsilon \in (0,1]$ *and parameters* $\eta = \sqrt{\frac{\epsilon \ln N}{4nK^2|\mathcal{C}|}}$, $\gamma = \frac{2\eta K|\mathcal{C}|}{\epsilon} \le 1/2$ *and* $\beta = \sqrt{\frac{K}{n|E|\epsilon} \ln \frac{2|E|}{\delta}} \le 1$ *and for all*

$$n \ge \frac{1}{\epsilon} \max\left\{ \frac{K^2 \ln^2(2|E|/\delta)}{|E| \ln N}, \frac{|E| \ln(2|E|/\delta)}{K}, 4|\mathcal{C}| \ln N \right\}$$

the performance of the algorithm defined above can be bounded, with probability at least $1 - \delta$ *as*

$$\begin{aligned}
&\frac{1}{n}\left(\sum_{t=1}^{n} \ell_{\boldsymbol{I}_t,t} - \min_{\boldsymbol{i}\in\mathcal{R}} \sum_{t=1}^{n} \ell_{\boldsymbol{i},t}\right) \\
&\le \sqrt{\frac{K}{n\epsilon}}\left(4\sqrt{K|\mathcal{C}|\ln N} + 5\sqrt{|E| \ln \frac{2|E|}{\delta}} + \sqrt{8K \ln \frac{2}{\delta}}\right) + \frac{4K}{3n\epsilon} \ln \frac{2N}{\delta} \\
&\le \frac{25K}{2}\sqrt{\frac{|E| \ln \frac{2N}{\delta}}{n\epsilon}}.
\end{aligned}$$

Sketch of the proof. The proof of the theorem is a generalization that of Theorem 1, and follows the same lines with some extra technicalities to handle the effects of the modified step (c'). Therefore, in the following we emphasize only the differences. First note that (1) and (2) also hold in this case. Now, instead of (3), one obtains

$$\sum_{\boldsymbol{i}\in\mathcal{R}} p_{\boldsymbol{i},t} g'_{\boldsymbol{i},t} = \frac{S_t}{\epsilon}\left(g_{\boldsymbol{I}_t,t} + |E|\beta\right) \quad \text{and} \quad \sum_{\boldsymbol{i}\in\mathcal{R}} p_{\boldsymbol{i},t} {g'}^2_{\boldsymbol{i},t} \le \frac{1}{\epsilon} K(1+\beta) \sum_{e\in E} g'_{e,t}$$

which imply, together with (1) and (2),

$$\sum_{t=1}^{n} \frac{S_t}{\epsilon}\left(g_{\boldsymbol{I}_t,t} + |E|\beta\ \right) \ge \left(1-\gamma-\frac{\eta K(1+\beta)|\mathcal{C}|}{\epsilon}\right) \max_{\boldsymbol{i}\in\mathcal{R}} \sum_{t=1}^{n} g'_{\boldsymbol{i},t} - \frac{1-\gamma}{\eta} \ln N. \quad (6)$$

To relate the left hand side of the above inequality to the real gain $\sum_{t=1}^{n} g_{\boldsymbol{I}_t,t}$, notice that

$$X_t = \frac{S_t}{\epsilon}\left(g_{\boldsymbol{I}_t,t} + |E|\beta\right) - \left(g_{\boldsymbol{I}_t,t} + |E|\beta\right)$$

is a martingale difference sequence. Then, it can be shown by applying Bernstein's inequality (see, e.g., [9]) that

$$\mathbb{P}\left(\sum_{t=1}^{n} X_t > \sqrt{\frac{8K^2 n}{\epsilon} \ln \frac{2}{\delta}} + \frac{4K}{3\epsilon} \ln \frac{2}{\delta}\right) \le \frac{\delta}{2}. \quad (7)$$

Furthermore, similarly to (5) it can be proved that

$$\mathbb{P}\left(\sum_{t=1}^{n} g_{e,t} > \sum_{t=1}^{n} g'_{e,t} + \frac{4\beta n|E|}{K}\right) \le \frac{\delta}{2|E|}. \quad (8)$$

An application of the union bound for (7) and (8) combined with (6) yields, with probability at least $1 - \delta$,

$$\sum_{t=1}^{n} g_{\boldsymbol{I}_t,t} \geq \left(1 - \gamma - \frac{\eta K(1+\beta)|\mathcal{C}|}{\epsilon}\right)\left(\max_{\boldsymbol{i}\in\mathcal{R}} \sum_{t=1}^{n} g_{\boldsymbol{i},t} - 4\beta n|E|\right)$$
$$- \frac{1-\gamma}{\eta} \ln N - \beta n|E| - \sqrt{\frac{8K^2 n}{\epsilon} \ln\frac{2}{\delta}} - \frac{4K}{3\epsilon}\ln\frac{2}{\delta}.$$

Using $\sum_{t=1}^{n} g_{\boldsymbol{I}_t,t} = Kn - \sum_{t=1}^{n} \ell_{\boldsymbol{I}_t,t}$ and $\sum_{t=1}^{n} g_{\boldsymbol{i},t} = Kn - \sum_{t=1}^{n} \ell_{\boldsymbol{i},t}$, and substituting the values of η, β, and γ yield, after some algebra, the statement of the theorem. $\square$

6 Simulations

To further investigate our new algorithms, simulations were conducted. We tested our bandit algorithm for shortest paths in a simple communication network shown in Figure 1. The simulation consisted of sending 10000 packets, from source node $u = 1$ to destination node $v = 6$, and our goal was to pick a route for each packet with small delay. We assumed the infinitesimal user scenario, that is, our choice for a path does not affect the delay on the links of the network.

Each link has a fixed propagation delay which is 0.1 ms. To generate additional delays (so called traffic delays), three major flows were considered, with periodically changing dynamics with period length 1000 time slots. The flow is a path between two determined nodes (not necessary u and v), which is loaded by traffic for a limited time period. The first flow, shown by a thick line in Figure 1, has a constant load, resulting in a constant 20ms traffic delay on all of its edges. The second flow, denoted by a dashed line, starts sending packets at time slot 200 of each period, and the traffic delays on its edges increase to 20ms by time slot 400, and stay there until time slot 700, when the flow is stopped, and the corresponding traffic delays drop back to 0 (we do not consider the transmission delay on the links). The third flow, denoted by a dotted line, has similar characteristics as the second flow, but it starts at time slot 500 and it reaches 20ms at time slot 700 and it keeps this level until the end of the period. Finally, the two thin lines in the graph denote links which are not used by the major flows.

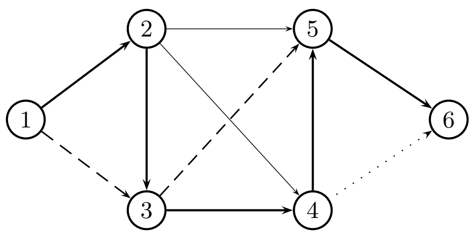

Fig. 1. Topology of the network.

The difficulty in this configuration is that the best fixed path switches 3 times during a period. From time 0 to time slot 200 there are three paths with the same performance: path $(1, 3, 5, 6)$, path $(1, 3, 4, 6)$, and path $(1, 2, 4, 6)$. From time slot 201 to time slot 700, path (1,2,4,6) has the smallest delay, and in the remainder of the period, path $(1, 3, 5, 6)$ is the best. In the long run these are the three best fixed paths, with $(1, 2, 4, 6)$ being the best, $(1, 3, 4, 6)$ the second best, and $(1, 3, 5, 6)$ the third.

In the simulations we ran the bandit algorithm for shortest paths with parameters optimized for $n = 10000$. We also ran an infinite horizon version of the algorithm, in which at each time instant t, the parameters η, β, and γ are set so that they are optimized for the finite horizon $n = t$. In this version $w_{e,t} = w_{e,t-1} \exp(\eta_t g'_{e,t})$, where η_t is decreasing in t and therefore this algorithm uses "reverse-discounted gains". Although we have not investigated the theoretical performance of this discounted style version, it can be observed that the modification substantially improves the performance of the algorithm in this example, and the modified version outperformed the second best route in the network. The reason for the good performance is that in the simulation the best fixed path in the long run does not change because of the periodicity of the flows and therefore a discounted algorithm can learn faster the best path than a non-discounted algorithm. We also compared our methods to that of Awerbuch and Kleinberg [3], and achieved better performance in all situations. The simulation results are summarized in Figure 2 that shows the normalized regret of the above algorithms (averaged over 30 runs), as well as the regrets of all fixed paths from node 1 to node 6 (the periodical small jumps on the curves correspond to the starting and ending times of the other flows). Note that in Figure 2, there is only 8 paths instead of 9, because of path (1,2,3,5,6) and path (1,3,4,5,6) have the same performance, and the curve for the best path $(1, 2, 4, 6)$ coincides with the x-axis.

7 Conclusions

Efficient algorithms have been provided for the on-line shortest path problem in the multi-armed bandit setting and in a combined label efficient multi-armed bandit setting. The regrets of the algorithms, compared to the performance of the best fixed path, converge to zero at an $O(1/\sqrt{n})$ rate as the time horizon n grows to infinity, and increases only polynomially in the number of edges (and nodes) of the graph. Earlier methods for the multi-armed bandit problem either do not have the right $O(1/\sqrt{n})$ convergence rate, or their regret increase exponentially in the number of edges for typical graphs. Simulation results showed the expected performance of the algorithms under realistic traffic scenarios.

Both problems are motivated by realistic problems, such as routing in communication networks, where the nodes do not have all the information about the state of the network. We have addressed the problem in the adversarial setting where the edge weights may vary in an arbitrary way, in particular, they may depend on previous routing decisions of the algorithm. Although this assumption may seem to be very strong in many network scenarios, it has applications in mobile ad-hoc networks, where the network topology changes dynamically in time, and also in certain secure networks that has to be able to handle denial of service attacks.

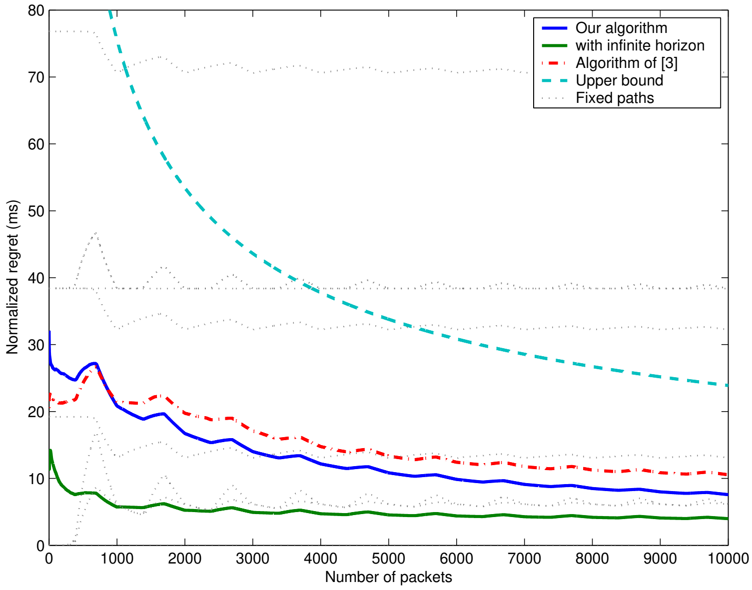

Fig. 2. Normalized regret of our bandit algorithm for shortest paths and that of the shortest path algorithm of [3].

Acknowledgments

The authors would like to thank Gábor Lugosi and László Györfi for useful discussions.

References

1. P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
2. B. Awerbuch, D. Holmer, H. Rubens, and R. Kleinberg. Provably competitive adaptive routing. In *Proceedings of IEEE INFOCOM 2005*, volume 1, pages 631–641, March 2005.
3. B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *Proceedings of the 36th Annual ACM Symposium on the Theory of Computing, STOC 2004*, pages 45–53, Chicago, IL, USA, Jun. 2004. ACM Press.
4. D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956.
5. O. Bousquet and M. K. Warmuth. Tracking a small set of experts by mixing past posteriors. *Journal of Machine Learning Research*, 3:363–396, Nov. 2002.
6. N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, D. Haussler, R. Schapire, and M. K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

7. N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, 2006.
8. N. Cesa-Bianchi, G. Lugosi, and G. Stoltz. Minimizing regret with label efficient prediction. *IEEE Trans. Inform. Theory*, IT-51:2152–2162, June 2005.
9. L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.
10. E. Gelenbe, M. Gellman, R. Lent, P. Liu, and P. Su. Autonomous smart routing for network QoS. In *Proceedings of First International Conference on Autonomic Computing*, pages 232–239, New York, May 2004. IEEE Computer Society.
11. E. Gelenbe, R. Lent, and Z. Xhu. Measurement and performance of a cognitive packet network. *Journal of Computer Networks*, 37:691–701, 2001.
12. A. György, T. Linder, and G. Lugosi. Efficient algorithms and minimax bounds for zero-delay lossy source coding. *IEEE Transactions on Signal Processing*, 52:2337–2347, Aug. 2004.
13. A. György, T. Linder, and G. Lugosi. A "follow the perturbed leader"-type algorithm for zero-delay quantization of individual sequences. In *Proc. Data Compression Conference*, pages 342–351, Snowbird, UT, USA, Mar. 2004.
14. A. György, T. Linder, and G. Lugosi. Tracking the best of many experts. In *Proceedings of the 18th Annual Conference on Learning Theory, COLT 2005*, pages 204–216, Bertinoro, Italy, Jun. 2005. Springer.
15. A. György, T. Linder, and G. Lugosi. Tracking the best quantizer. In *Proceedings of the IEEE International Symposium on Information Theory*, pages 1163–1167, Adelaide, Australia, June-July 2005.
16. A. György and Gy. Ottucsák. Adaptive routing using expert advice. *The Computer Journal*, 49(2):180–189, 2006.
17. J. Hannan. Approximation to bayes risk in repeated plays. In M. Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.
18. D. P. Helmbold and R. E. Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27:51–68, 1997.
19. M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998.
20. A. Kalai and S Vempala. Efficient algorithms for the online decision problem. In B. Schölkopf and M. Warmuth, editors, *Proceedings of the 16th Annual Conference on Learning Theory and the 7th Kernel Workshop, COLT-Kernel 2003*, pages 26–40, New York, USA, Aug. 2003. Springer.
21. N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.
22. H. B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Annual Conference on Learning Theory, COLT 2004*, pages 109–123, Banff, Canada, Jul. 2004. Springer.
23. M. Mohri. General algebraic frameworks and algorithms for shortest distance problems. Technical Report 981219-10TM, AT&T Labs Research, 1998.
24. E. Takimoto and M. K. Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4:773–818, 2003.
25. V. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 372–383, Rochester, NY, Aug. 1990. Morgan Kaufmann.
26. V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, 35(3):247–282, Jun. 1999.