Bandit problems

Vishakh Gopu, Frederik Jensen

Abstract

In this summary, we review three papers related to the multi-armed bandit problem, discuss the main conceptual ideas and anlyze the algorithms and theoretical results.

1 Introduction

The multi-armed bandit scenario is synonymous to a class of computer science problems in which an online learner seeks to minimise her regret over a number of actions. In particular to the bandit setting, at each round t only the loss of the given action is disclosed to the learner. The learner thus finds herself in a partial information setting where learning becomes more difficult as reflected in the increase of the optimal algorithmic bound from $\sqrt{\ln(K)T}$ in the full information setting to $\sqrt{KT}$ with K representing the number actions available to the learner at each round; T, the total rounds.

In this paper, we review the findings of the papers (Alon *et al.* , 2014; Hazan & Kale, 2009; Bertsimas & Niño-Mora, 2000). Common to all of the papers, the authors expand on the analytical body of work related to the multi-armed bandit problem. In particular, (Alon *et al.* , 2014) considers the adversarial bandit problem in which the loss of several actions are revealed based on a feedback graph at each round. (Hazan & Kale, 2009) develops a series of linear programming relaxations for approximately solving restless bandit problem. Lastly, (Bertsimas & Niño-Mora, 2000) bounds the regret in the non-stochastic bandit setting by the total variation in the loss functions. In the following, we first expound on the problem areas of the papers. Subsequently, we introduce the papers' novel conceptual ideas before backing them up with their respective analyses. Lastly, we sketch some connections between the papers before rounding off with a paragraph on open problems.

2 Problem Definitions

We adopt the notation from (Alon *et al.* , 2014; Hazan & Kale, 2009): formally, consider the family of action sets $\mathcal{K} \subset \mathbb{R}^n$. At round $t = 1, 2, 3...$, the learner choses an action set $\mathbf{x} \in \mathcal{K}$, and suffers a loss $\mathbf{f}_t \cdot \mathbf{x}_t$ with $\mathbf{f}_t \in [0,1]^n$ denoting the cost vector at time t. In the general multi-armed bandit setting, the problem consists of minimising the regret R_T, that is, the loss with respect to the best action vector in hindsight[1], for $T \to \infty$ defined as

$$R_T = \sum_{t=1}^{T} \mathbf{f}_t \cdot \mathbf{x}_t - \min_{\mathbf{x} \in \mathcal{K}} \sum_{t=1}^{T} \mathbf{f_t} \cdot \mathbf{x}, \qquad (1)$$

where $\mathbf{x}_t = \mathbf{e}_{t,i}$ the standard basis vector with 1 at position $i = \{1, ..., n\}$ and 0 everywhere else. Thus, at each round t, the learner choses arm $\mathbf{e}_i$ and is only revealed the loss $\mathbf{f}_t \cdot \mathbf{e}_i$ pertaining to that specific action i, that is $\mathbf{f}_{t,i}$.

In (Alon *et al.* , 2014), the authors extend the problem definition by introducing a *feedback system*, $\{S_{t,i}\}_{i \in \mathcal{K}}$. The feedback system represents possible correlations between a set of actions. For instance, in the case of deciding which ads to display to a website user to maximise hits, if the user already clicked on a shoe ad, we can assume that the same user would likely click on another shoe ad. Formally, at each round, the feedback system defines the set of losses that will be revealed if action $\mathbf{x}_{i,t}$ was taken with $\mathbf{x}_{t,i}$ denoting the i^{th} element of $\mathbf{x}_t$. Since $\{S_{t,i}\}_{i \in \mathcal{K}}$ can be modelled as a graph, the authors ask how the feedback system influences the regret bounds: Do denser or thinner graphs lead to better regret bounds? Which graph properties influence the regret bound?

The paper by (Hazan & Kale, 2009) considers the bandit linear optimization problem in which $\mathcal{K}$ is a compact set, and the cost function $\mathbf{f}_t$ is linear. Note that $\mathbf{x}_t \neq \mathbf{e}_i$

[1] Depending on the specific subfield, the problem is reformulated as maximising the reward function instead.

necessarily, but $\mathbf{x}_t$ now represents a point in the simplex $\mathcal{K}$ on which we are doing coordinate descent for which the gradient is disclosed for a given direction only after the corresponding action is taken. The authors are motivated by the need to minimise commuting time from home to work. Since traffic patterns are unknown and vary every day, it would be more realistic to use total variation as a bound for an time-optimising bandit algorithm rather than considering the adverserial regret. Let $Q_T = \sum_{t=1}^{T} \|f_t - \mu\|^2$ be the total quadratic variation of the cost functions $\mathbf{f}_t$. The authors then ask: Can Q_T bound the regret?

Lastly, (Alon *et al.* , 2014) studies the restless bandit problem. This problem differs in that each action i in $\mathbf{x}_t$ is considered either *active* or *passive*, denoted as $\mathbf{x}^1_{t,i}$ and $\mathbf{x}^0_{t,i}$ respectively. Moreover, a finite state space $\mathcal{S}_i$ is attributed to each action that can take on any of the $s_i \in \mathcal{S}_i$ states, denoted as $\mathbf{x}_{t,i,s_i}$. Rewards are associated with each action based on its state, denoted as $R^1_{s_i}$ when active, and $R^0_{s_i}$ when passive. Moreover a Markovian transition matrix prescribes the probability $p^1_{s_i s'_i}$ with which an active action transitions from state s_i into s'_i; $p^0_{s_i s'_i}$ in the passive case. In addition, let $0 < \beta < 1$ be a reward discount factor. The problem is then regarded as finding a Markovian policy $u \in \mathcal{U}$, where $\mathcal{U}$ is the set of all admissible policies, such that

$$\max_{u \in \mathcal{U}} E\left[\sum_{t=0}^{\infty} (R^{\alpha_1(t)}_{s_1(t)} + ...R^{\alpha_n(t)}_{s_n(t)})\beta^t\right], \qquad (2)$$

with $s_i(t)$ and $a_i(t)$ denoting the state and mode (active or passive) for action i.

3 Conceptual and Mathematical Ideas

With the problem areas well-defined, this section compiles the conceptual ideas from the papers. Overall, (Alon *et al.* , 2014; Hazan & Kale, 2009) presents novel ideas and proof techniques, while (Bertsimas & Niño-Mora, 2000) via a simple proof invent a range of LP relaxations for the restless bandit. The latter paper is thus more relevant for applications. Since our focus is mainly theoretical, however, we tone down the findings (Bertsimas & Niño-Mora, 2000) in favor of the two other papers.

3.1 Feedback System

As mentioned in the introduction, the authors in (Alon *et al.* , 2014) set about exploring how semantic connections in the action set can influence the bandit model. Motivated by the example of web advertising in particular, they argue that similarities between actions can strengthen the regret bound of the bandit algorithm, because such similarities can reveal information about the losses of other actions.

The authors cleverly model the semantic relationships in a *feedback graph* $G_t = (\mathcal{K}, D_t)$ where $\mathcal{K}$ is the set of actions and D_t the set of arcs. Let any arc $(i, j) \in D_t$ for $i \neq j$ if and only if playing action $i \in \mathcal{K}$ reveals the loss of action $j \in \mathcal{K}$ at time t. We write $i \xrightarrow{t} j$. The full information setting then corresponds to a complete graph; the bandit setting, an empty graph. Furthermore, any sub-graph in between represents the partial information setting which interpolates between the expert and the bandit setting. In order to prove guarantees in this scenario, the authors examine some graph theoretic notions including *independence numbers*, *dominating sets*, and *maximum acyclic graphs*. Recall that the independence number, $\alpha(G)$, is the cardinality of the maximal subset $T \subset \mathcal{K}$ such that no two $i, j \in T$ are connected by an edge; that is $(i, j) \notin D_t$. An independent set T is maximal if no proper superset of T forms an independent set. Moreover, $R \subset G$ is a dominating set if for any $j \notin R$ there exists an $i \in R$ that exhibits a directed edge to j. Lastly, the maximum acyclic subgraph in G, denoted $\text{mas}(G)$, is the largest subgraph with no directed cycles.

The authors then distinguish between the *informed* and *uninformed* setting. In the informed setting the feedback graph is known to the learner in advance of deciding which action to take; in the uninformed setting, only after an action has been taken. Moreover, in each scenario, it is useful to distinguish between the *undirected* and the *directed* case as the regret bounds vary with these.

Based on these formal structures, the authors analyze and give regret bounds on modified versions of the Exp3 algorithm. A key concept in their analysis is the central quantity[2]:

$$V_t = \sum_{i \in |\mathcal{K}|} \frac{p_{i,t}}{q_{i,t}} = \sum_{i \in |\mathcal{K}|} \frac{p_{i,t}}{\sum_{j:j \xrightarrow{t} j} p_{j,t}}, \qquad (3)$$

where $p_{i,t}$ is the probability of selecting action i at time t, and $q_{i,t}$ is the probability of observing the loss of action. For the uninformed setting, the authors modify the Exp3 algorithm to Exp3-SET to incorporate , which shows up as a new surrogate loss defined as $\hat{l}_{i,t} = \frac{l_{i,t}}{q_{i,t}} 1_{i \in S_{I_t,t}}$. Here, I_t is the action taken at time t, and 1 is the indicator function. Then, they bound the regret by V_t in the following foundational lemma.

[2] In the paper they denote it as Q_t but due to conflict with (Hazan & Kale, 2009) we rewrite it as V_t.

Lemma 3.1. *The regret of Exp3-SET satisfies*

$$R_t \leq \frac{\ln|\mathcal{K}|}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}[V_t]. \tag{4}$$

The proof is a straightforward application of the Exp3 proof technique where care is taken to incorporate the new loss definition properly. For the uninformed setting this leads to the regret bound in the directed and undirected setting.

Theorem 3.1. *In the asymmetric case, setting $\eta = \sqrt{(2\ln|\mathcal{K}|)\sum_{t=1}^{T} m_t}$ with m_t bounding* $\text{mas}(G_t)$ *for $t = 1, ..., T$, the regret of Exp3-SET satisfies*

$$R_T \leq \frac{\ln K}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}[\text{mas}(G_t)] \tag{5}$$

An straightforward consequence is the following corollary.

Corollary 3.1. *In the symmetric case, setting $\eta = \sqrt{(2\ln|\mathcal{K}|)\sum_{t=1}^{T}\alpha_t}$ with α_t bounding $\alpha(G_t)$ for $t = 1, ..., T$, the regret of Exp3-SET satisfies*

$$R_T \leq \frac{\ln K}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\mathbb{E}[\alpha(G_t)]. \tag{6}$$

The following lemma links V_t with $\text{mas}(G_t)$

Lemma 3.2. *Let $G = (V, D)$ be a directed graph with vertex set $V = \{1, ..., |\mathcal{K}|\}$ and arc set D. Then for any distribution p over V it follows that*

$$\sum_{i=1}^{|\mathcal{K}|}\frac{p_i}{p_i + \sum_{j:j\to i} p_j} \leq \text{mas}(G). \tag{7}$$

The lemma's proof is included here as it shows how to connect the graph theoretic property $\text{mas}(G_t)$ with the central sum V_t.

Proof. Let N_i^- be the set of vertices such that $j \in N_i^-$ iff $j \to i$. N_i^- is the in-neighborhood of i. The lemma is proved by adding elements to an initially empty set V'. Let

$$\Phi_0 = \sum_{i=1}^{|\mathcal{K}|}\frac{p_i}{p_i + \sum_{j:j\to i} p_j}$$

and let i_1 be the vertex that minimizes $p_i + \sum_{j\in N_i^-}$ over $i \in V$. Now delete i_1, $N_{i_1}^-$ and all edges incident to these vertices from G. Let $N_{i,1}^-$ be the in-neighborhoods after the first step. Note that the contribution of all the deleted vertices to Φ_0 is

$$\begin{aligned}
&\sum_{r\in N_{i_1}^-\cup\{i_1\}}\frac{p_r}{p_r + \sum_{j\in N_r^-} p_j} \\
&\leq \sum_{r\in N_{i_1}^-\cup\{i_1\}}\frac{p_r}{p_{i_1} + \sum_{j\in N_{i_1}^-} p_j} = 1,
\end{aligned}$$

where the inequality comes from the minimality of i_1. Let $V' \leftarrow V' \cup \{i_1\}$ and $V_1 = V\backslash(N_{i_1}^- \cup \{i_1\})$. Then, the first step yields

$$\begin{aligned}
\Phi_1 &= \sum_{i\in V_1}\frac{p_i}{p_i + \sum_{j\in N_{i,1}^-} p_j} \\
&\geq \sum_{i\in V_1}\frac{p_i}{p_i + \sum_{j\in N_i^-} p_j} \\
&\geq \Phi_0 - 1.
\end{aligned}$$

This process is repeated over Φ_1, and then Φ_2... until no vertices are left in the graph. This gives

$$\Phi_0 \leq s = |V'| = \text{mas}(G),$$

with $V' = \{i_1, i_2, ..., i_s\}$. Moreover, each step $r = 1, ..., s$ removes all incoming arcs to i_r, V' cannot contain cycles. □

Comparing this lemma with lemma 3.1 immediately yields the regret bounds in 3.1.

For the undirected case, corollary is tight. However, in the directed case, theorem 3.1, the bound is loose which can be seen by the following construction.

Corollary 3.2. *Let $G = (V, D)$ be a total order on $V = \{1, ..., |\mathcal{K}|\}$ such that for all $i \in V$, arc $(j, i) \in D$ for all $j = i + 1, ..., K$. Let $p = (p_1, ..., p_{|\mathcal{K}|})$ be a distribution on V such that $p_i = 2^{-1}$, for $i < |\mathcal{K}|$ and $p_k = 2^{-|\mathcal{K}|+1}$. Then, by the geometric series,*

$$\begin{aligned}
Q &= \sum_{i=1}^{|\mathcal{K}|}\frac{p_i}{p_i + \sum_{j:j\to i} p_j} \\
&= \sum_{i=1}^{|\mathcal{K}|}\frac{p_i}{\sum_{j=1}^{|\mathcal{K}|} p_j} = \frac{|\mathcal{K}| + 1}{2}.
\end{aligned}$$

While these proofs show the gist of the ideas in the paper, the authors move on to provide a general bound that holds in the directed case. Due to the corollary above, this is only possible in the informed setting. They modify the Exp3 algorithm and name it Exp3-DOM. It uses the Greedy Set Cover algorithm to approximate the minimal dominating set, which it uses to bound the regret with $\alpha(G_t)$. The bound is tight up to logarithmic dependencies on K, and follows a similar proof outline as above.

Theorem 3.2. *If Exp3-DOM uses the Greedy Set Cover algorithm to compute dominating sets, then the regret of Exp-DOM using the doubling trick satisfies*

$$R_T = \mathcal{O}\Bigg(\ln(|\mathcal{K}|)\sqrt{\ln(|\mathcal{K}|T)\sum_{t=1}^{T}\alpha(G_t)} \\ + \ln(|\mathcal{K}|)\ln(|\mathcal{K}|T)\Bigg).$$

3.2 Total Variation

The paper by (Hazan & Kale, 2009) introduces the notion of bounding the regret of a learner by the total variation of the loss vector $\mathbf{f} = \{\mathbf{f}_1, ..., \mathbf{f}_T\}$ in the bandit optimisation setting. Intuitively, it should be easier to learn in a setting where the change in rewards very little compared to a lot. Since typical bounds for adversarial bandits don't take this into account and, instead, implicitly assume the worst of the adversary those bounds, argue the authors, are not realistic. As mentioned in the introduction, for instance, if you are planning a route to travel into work, the traffic may be unpredictable in general, yet there certain things can be exploited such as correlations between congestion and time of day.

From this point of view (Hazan & Kale, 2009) uses the *quadratic variation* of the loss vectors defined as $Q_T = \sum_{t=1}^{T} ||f_t - \mu||^2$, where f_t is the cost vector at time t and μ is the mean of all the cost vectors $\mu = \sum_{t=1}^{T} f_t$.

This quantity provides a useful measure of the unpredictability of the losses assigned by our adversary and, as will be shown, appears in the bound of the regret. While the idea of a bound by variation seems conceptually straight forward, it is unclear how to integrate it into the problem setting. The gist of the paper revolves around this integration and the authors' solution is two-fold:

1. model the mean loss of each action $\mathbf{x}_t \in \mathcal{K}$ in an unbiased way to keep a tracking estimator of the loss vector; and
2. sample a new point taking advantage of the knowledge of the old means of each action, $\tilde{\mu}$.

With respect to the first part, the estimator is kept and updated using a technique known as *reservoir sampling*. For each coordinate $i \in [1, ..., n]$ of the simplex $\mathcal{K}$ we keep a vector of arbitrary size k and denote it as the reservoir $S_{i,k}$. The estimator of the mean $\tilde{\mu_t}(i)$ then becomes $\tilde{\mu}(i) = \frac{i}{k}\sum_{j=1}^{k} S_{i,j}$. The authors' show that each resevoir gives an unbiased estimator for all $i \in [1, ..., n]$.

With the resevoir method in place, the authors present an algorithm based on *follow-the-regularized-leader* (FTRL) with a particular regularisation $\mathcal{R}(x)$ enforced by *self-concordant barriers*. This means that any sampled point should minimize: $\arg\min_{\mathbf{x}_t \in \mathcal{K}} \eta \sum_{t=1}^{t-1} \tilde{f}_\tau^{\;T} + \mathcal{R}(x)$ where η is a learning rate.

In order to understand self-concordant barriers, some definitions are needed.

Definition 3.1. *A convex function $\mathcal{R}(\mathbf{x})$ defined on the interior of the convex compact set $\mathcal{K}$ and having three continuous derivatives is said to be a $\mathcal{V}$-concordant barrier if the following conditions hold:*

1. *$\mathcal{R}(\mathbf{x}_i) \to \infty$ along every sequence of points x_i in the interior of $\mathcal{K}$ converges to a boundary point of $\mathcal{K}$.*
2. *$\mathcal{R}$ satisfies*

$$|\nabla^3\mathcal{R}(\mathbf{x})[\mathbf{h},\mathbf{h},\mathbf{h}]| \leq 2(\mathbf{h}^\top[\nabla^2\mathcal{R}(\mathbf{x})]\mathbf{h})^{\frac{3}{2}},$$
$$|\nabla\mathcal{R}(\mathbf{x})^\top\mathbf{h}| \leq \mathcal{V}^{\frac{1}{2}}[\mathbf{h}^\top\nabla^2\mathcal{R}(\mathbf{x})\mathbf{h}]^{\frac{1}{2}}$$

The self-concordant barrier definition is key in ensuring that when sampling new points, these points remain within the feasible set. In order for this to work completely, however, some more technical machinery is needed.

Definition 3.2. *The Dikin ellipsoid, is an elipsoid of radius r centered at $\mathbf{x}$ defined as*

$$W_r(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^n : ||\mathbf{y} - \mathbf{x}||_{\mathbf{x}} \leq r\}.$$

Furthermore, the following Minkowsky function definition is needed.

Definition 3.3. *For any two distinct points $\mathbf{x}$ and $\mathbf{y}$ in the interior of $\mathcal{K}$, the Minkowsky function $\pi_{\mathbf{x}}(\mathbf{y})$ on $\mathcal{K}$ is*

$$\pi_{\mathbf{x}}(\mathbf{y}) = \inf\{t \geq 0 : \mathbf{x} + t^{-1}(\mathbf{y} - \mathbf{x}) \in \mathcal{K}\}.$$

From these definitions it follows that $W_1(\mathbf{x}) \subseteq \mathcal{K}$ for any $\mathbf{x} \in \mathcal{K}$, which allows for sampling in an unbiased way from the feasible set.

The main algorithm consists of two steps: The *SimplexSample* and the *EllipsoidSample* step. SimplexSample takes care of the reservoir sampling and maintains the estimator of the loss vector $\mathbf{f}_t$. EllipsoidSample randomly chooses an actual point $\mathbf{y_t}$ from the endpoints of the principle axes of the Dikin ellipsoid $W_1(\mathbf{x})$ centered at $\mathbf{x}_t$. From the definitions above this sample point is in the feasible set of points. Furthermore its been shown in (Abernethy *et al.* , 2008) that the sampling is unbiased and has low variation. The $\tilde{\mu_t}$ that is calculated in SimplexSample is then incorporated in EllipsoidSample for smarter exploration which allows regret bounds based on the total variation. Refer to section 4 for a discussion of the algorithm.

The main theoretical result of (Hazan & Kale, 2009) combines the above concepts into a bound on the regret in terms of the total variation of the cost vectors for oblivious adversarial bandits. Particularly we are in the regime where the points $\mathbf{x}_t \in \mathcal{K}$ where $\mathcal{K} \in \mathbb{R}^n$ is a compact convex set. Let Q be an estimated upperbound on Q_T, then for

$$\eta = \min\{\sqrt{\frac{logT}{\eta Q}}, \frac{1}{25n}\}$$

we have the following bound on regret:

$$E[Regret_t] = O(n\sqrt{\mathcal{V}QlogT} + nlog^2(T) + n\mathcal{V}log(T)) \tag{8}$$

The assumed upperbound on Q_T is to simplify the exposition and is not required to carry out the proof as is shown in the paper. The main series of arguments made to demonstrate the bound can be listed as a series of lemmas. In order to show how the authors manages to pull in the total variation they are included and discussed below under the numbering provided by (Hazan & Kale, 2009) for reference.

Lemma 7. *For any $u \in \mathcal{K}$*

$$E[\sum_{t=1}^{T} f_t^T(y_t - u)] \leq E[\sum_{t=1}^{T} \tilde{f_t^T}(x_t - u)] + 2nlog^2(T). \tag{9}$$

This Lemma serves to relate the expected regret of the algorithm with the cost vectors of another algorithm that plays just $\mathbf{x}_t$ and not the points $\mathbf{y}_t$ as derived from $\mathbf{x}_t$. This relationship links the key quantities in $\sum_{t-1}^{T} \tilde{\mathbf{f}_t^T}(\mathbf{x}_t - \mathbf{u})$. Furthermore, notice that the expectations of $\tilde{t_t}$ and $\tilde{y_t}$ are $\mathbf{f}_t$ and $\mathbf{x}_t$ respectively. Thus their expected costs can be related per round as done in the lemma. The expected number of rounds grows as $O(nklog(T))$ which explains the last term in the expression: $2n\log^2(T)$. This is used to our advantage when applying typical proof techniques used in FTRL type algorithms.

Lemma 8. *For any sequence of cost vectors $\{\tilde{f}_1, ..., \tilde{f}_t\} \in \mathbb{R}^n$ the FTRL algorithm with a $\mathcal{V}$-self concordant barrier $\mathcal{R}$ admits the following guarantee: for any $\mathbf{u} \in \mathcal{K}$ we have:*

$$\sum_{t=1}^{T} \tilde{F_t^T}(\mathbf{x}_t - \mathbf{u}) \leq sum_{t=1}^{T}(\mathbf{x}_t - \mathbf{x}_{t+1}) + \frac{2}{\eta}\mathcal{V}logT \tag{10}$$

This statement appeals to a commonly used technique of bounding a FTRL type algorithm by how closeness of the the succesive values $\mathbf{x}_t, \mathbf{x}_{t+1}$. Next, the the different sample steps, namely EllipsoidSample and SimplexSample, are related to the bound.

Lemma 9. *Let t be an EllipsoidSample step. Then,*

$$\begin{aligned}&\tilde{\mathbf{f}_t^T}(\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\quad \leq 64\eta n^2||\mathbf{f}_t - \mu_t||^2 + 64\eta n^2||\mu_t - \tilde{\mu_t}||^2 + 2\mu_t^T(\mathbf{x}_t - \mathbf{x}_{t+1}).\end{aligned} \tag{11}$$

To understand Lemma 9, turn instead to the equivalent case for SimplexSample where the analysis is simpler and then fill in the remaining parts. Since in SimplexSample $\tilde{\mathbf{f}}_t = 0$ the following must be true: $\tilde{\mathbf{f_t}}^T(\mathbf{x}_t - \mathbf{x}_{t+1}) = 0 = 2\mu_t^T(\mathbf{x}_t - \mathbf{x}_{t+1})$. Thus for any SimplexSample step we get

$$\tilde{\mathbf{f}_t^T}(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 64\eta n||\mathbf{f}_t - \mu_t||^2 2\mu_t^T(\mathbf{x}_t - \mathbf{x}_{t+1})$$

Call the set of all EllipsoidSample steps T_E and sum over the time periods. Doing so yields terms that can be bounded by the inequalities for SimplexSample, which gets us the final form of Lemma 9.

Next, some facts about how the sampling procedures affect the guarantee are needed.

Lemma 10.

$$\sum_{t=1}^{T} ||\mathbf{f}_t - \mu_t||^2 \leq Q_T \tag{12}$$

This simply follows from using the properties of the variance of the estimators of the mean reward loss $\tilde{\mu}$ when using resevoir sampling which is shown by (Vitter, 1985). Finally, the total variation is introduced into the bounds by recognizing that the following upperbound exists in the equalities shown so far.

Lemma 11.

$$E[\sum_{t \in T_E} ||\mu_t - \tilde{\mu_t}||^2] \leq \frac{\log T}{k} Q_T \tag{13}$$

This step upperbounds the succesive difference of means seen in Lemma 10 by the total variation. Entering some messier terrain, the hyperparameters are set to optimal values such that the bound becomes interpretable and tighter.

Lemma 12.

$$\sum_{t=1}^{T} \mu_t^T(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 2\log(Q_T + 1) + 4 \tag{14}$$

The first such optimal value inserted is for the resevoir of size k. Plug in the bounds from lemmas 10, 11, and 12 into the bound from lemma 4 and use the value $k = log(T)$ to obtain the simpler expression:

$\sum_{t=1}^{T} \mathbf{f}_t^T(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 128\eta^2 Q + 4\log(Q_T + 1)) + 8$

Now, consider the value of the exploration-exploitation tradeoff parameter η, and assume that the value of η is larger than some value $\frac{\log(Q_t+1)}{8n^2 Q}$. Using this assumption (which we are in control of since η is a hyperparameter that we can choose) gives the following upperbound on total variation $Q_T + 1 \leq 8\eta n^2 Q$.

Using lemmas 8 and 7, for any $u \in \mathcal{K}$ it holds that:

$$E[\sum_{i=1}^{T} \mathbf{f}_t^T(\mathbf{y}_t - \mathbf{y})]$$
$$\leq 128n^2 Q + \frac{2\mathcal{V}}{\eta} logT + 2n\log^2(T) + 4\log(Q_T + 1) + 8$$

Then choosing a suitable value for η that respects the assumptions made earlier about its value recovers the original regret bound:

$$E[\sum_{t=1}^{T} f_t^T(y_t - u)] \qquad (15)$$
$$\leq O(n\sqrt{\mathcal{V}QlogT} + n\mathcal{V}log(T) + nlog^2(T)).$$

3.3 Linear Programming Relaxations

The main contribution of (Bertsimas & Niño-Mora, 2000) is to provide a hierarchy of LP-relaxations to approximately solve the restless bandits problem which is described in the introduction. The technique employed by (Bertsimas & Niño-Mora, 2000) directly draws inspiration from previous work and ideas in combinatorial optimization and stochastic optimization.

A first order relaxation for the restless bandits problem was introduced by (Whittle, 1988). This version could be solved effeciently in polynomial time but there were many restrictions and assumptions made by the approach which makes it less than ideal. Another motivating result was the discovery by (Papadimitriou & Tsitsiklis, 1999) that the general restless bandits is PSPACE-hard which makes it intractable in general. This obviously puts emphasis on finding strong approximate solutions rather than solving the problem exactly.

In the following we show how (Bertsimas & Niño-Mora, 2000) phrase the restless bandit problem as a LP problem.

Let the state-action space be $\ell = \{(i, a) : i \in \mathcal{L}, a \in \mathcal{A}_i\}$ with states i and actions a. Then the following performance measures can be introduced

$$x_j(u) = E_u[\sum_{t=0}^{\infty} I_j^a(t)\beta^t]$$

where $I_j^a = 1$ if an action is taken at the time step t in the state j, and 0 otherwise. The optimization process that follows from the performance measure is then

$$Z^* = max_{u\in\mathcal{U}} \sum_{(i,a)\in\mathcal{L}} R_j^a x_j^a(u).$$

The authors show the explicit performance region spanned by the vectors $x(u)$ under all the policies that are in consideration. This leads to a mathematical program given as

$$Z^* = max_{x\in\mathcal{X}} \sum_{(i,a)\in\mathcal{L}} R_i^a x_i^a$$

where $x = (x_j^a)$.

As is typical of most linear programming formulations, the feasible set lives on a polytope of some type, in this case in $\mathcal{P}$

$$\mathcal{P} = \{x \in R_+^{|\mathcal{L}|}\} : \sum_{a\in\mathcal{A}} x_j^a = \alpha_j + \beta \sum_{(i,a)\in\mathcal{L}} p_{i,j}^a x_i^a, j \in \mathcal{L}.$$

Most crucial to the paper, the authors present an argument that leads to an LP formulation of the problem. They present a known theorem that connects the polytope and the performance region.

Theorem 3.3. *The following statements hold with X being the performance region:*

1. *$X = \mathcal{P}$*
2. *The vertices of the polytope: $\mathcal{P}$ are achievable by stationary deterministic policies*

While the direction $X \subseteq \mathcal{P}$ was proved in 1960 by d'Epenoux, the authors contribute a proof of construction for the other direction.

With this in mind, the notion of performance region becomes important to the formulation of the problem as a LP. First take the following two performance measures

$$x_{i_n}^1 = E_u[\sum_{t=0}^{\infty} I_{i_n}(t)\beta^t], \text{ and}$$

$$x_{i_n}^0(u) = E_u[\sum_{t=0}^{\infty} I_{i_n}(t)\beta^t],$$

where u is an admissible scheduling policy. Let $I_{i_n}^1(t)$ be 1 if project n is in state i_n and active at time t, and 0 otherwise. Similarly let $I^0_{i_n}(t)$ be 1 if the project n is in state i_n and passive at time t and 0 otherwise.

Thus for the current set of active and passive tasks one can give a performance measure, and then combine them to give a performance measure for the entire region:

$$X = \{x = (x^{a_n}_{i_n}(u))_{i_n \in \mathcal{L}_n, a_n \in \{0,1\}, n \in \mathcal{N}} | u \in \mathcal{U}\}$$

From the theorem shown earlier we know that there is a correspondence between a discounted MDC and a polytope which means we can take the performance measures and create a linear program associated to the polytope:

$$Z^* = max_{x \in \mathcal{X}} \sum_{n \in \mathcal{N}} \sum_{i_n \in \mathcal{L}_n} \sum_{a_n \in \{0,1\}} R^{a_n}_{i_n} x^{a_n}_{i_n}$$

In this way, the authors achieve a general formulation for a linear program associated to the restless bandit problems by phrasing the discounted MDC formulation as a polytope with its associated linear program.

4 Algorithms

4.1 Exp3-SET

The Exp3-SET is exactly as the Exp3 algorithm, except for the formation of the surrogate loss. Given in algorithm 1, instead of dividing the surrogate loss $\ell_{i,t}$ with the probability of the action i, it is divided by $q_{i,t}$ as defined in the analysis part; that is, the probability that the loss of action i is revealed. It is thus a direct and very natural extension of the Exp3 algorithm.

4.2 Exp3-DOM

Since the Exp3-DOM holds with sharp bounds in the more general case of assymmetric graphs, it also leverages more techniques. Given in algorithm 2, as seen by the superscript $^{(b)}$ the algorithm runs K $Exp3$ algorithms and then computes an approximation of the dominating set R_t using the Greedy Set Cover algorithm. The size of R_t is used as an index over Exp3 algorithms and decides which probability distribution to pick an action from. Interestingly, note in line 8 that whenever the exploration parameter γ is small little exploration is done.

Conversely, when γ is closed to one, all the weight is put on exploration, and the weights from the previous timestep counts for nothing. Lastly, the loss is expressed as a fraction of $\gamma^{(b_t)}/R_t$ which penalises for exploration.

Algorithm 1 Exp3-SET

```
1: Parameter: $\eta \in [0,1]$
2: Initialize: $w_{i,1} = 1$ for all $i \in V = \{1,\dots,K\}$
3: for $t = 1$ to T do
4:     Feedback system $\{S_{i,t}\}_{i \in V}$ and losses $\ell_t$ are generated but not disclosed ;
5:     Set $p_{i,t} = \frac{w_{i,t}}{W_t}$ for each $i \in V$, where $W_t = \sum_{j \in V} w_{j,t}$ ;
6:     Play action $I_t$ drawn according to distribution $p_t = (p_{1,t},\dots,p_{K,t})$ ;
7:     Observe:
           1. pairs $(i, \ell_{i,t})$ for all $i \in S_{I_t,t}$;
           2. Feedback system $\{S_{i,t}\}_{i \in V}$ is disclosed;
8:     For any $i \in V$ set $w_{i,t+1} = w_{i,t} \exp\big(-\eta\,\widehat{\ell}_{i,t}\big)$, where
           $\widehat{\ell}_{i,t} = \frac{\ell_{i,t}}{q_{i,t}} \mathbb{I}\{i \in S_{I_t,t}\}$   and   $q_{i,t} = \sum_{j\,:\,j \xrightarrow{t} i} p_{j,t}$ .
9: end for
```

Algorithm 2 Exp3-DOM

```
1: Input: Exploration parameters $\gamma^{(b)} \in (0,1]$ for $b \in \big\{0,1,\dots,\lfloor \log_2 K \rfloor\big\}$
2: Initialization: $w^{(b)}_{i,1} = 1$ for all $i \in V = \{1,\dots,K\}$ and $b \in \big\{0,1,\dots,\lfloor \log_2 K \rfloor\big\}$
3: for $t = 1$ to T do
4:     Feedback system $\{S_{i,t}\}_{i \in V}$ is generated and disclosed, (losses $\ell_t$ are generated and not disclosed);
5:     Compute a dominating set $R_t \subseteq V$ for $G_t$ associated with $\{S_{i,t}\}_{i \in V}$ ;
6:     Let $b_t$ be such that $|R_t| \in \big[2^{b_t}, 2^{b_t+1} - 1\big]$;
7:     Set $W^{(b_t)}_t = \sum_{i \in V} w^{(b_t)}_{i,t}$;
8:     Set $p^{(b_t)}_{i,t} = \big(1 - \gamma^{(b_t)}\big) \frac{w^{(b_t)}_{i,t}}{W^{(b_t)}_t} + \frac{\gamma^{(b_t)}}{|R_t|} \mathbb{I}\{i \in R_t\}$;
9:     Play action $I_t$ drawn according to distribution $p^{(b_t)}_t = \big(p^{(b_t)}_{1,t},\dots,p^{(b_t)}_{K,t}\big)$ ;
10:    Observe pairs $(i, \ell_{i,t})$ for all $i \in S_{I_t,t}$;
11:    For any $i \in V$ set $w^{(b_t)}_{i,t+1} = w^{(b_t)}_{i,t} \exp\big(-\gamma^{(b_t)}\,\widehat{\ell}^{(b_t)}_{i,t}/2^{b_t}\big)$, where
           $\widehat{\ell}^{(b_t)}_{i,t} = \frac{\ell_{i,t}}{q^{(b_t)}_{i,t}} \mathbb{I}\{i \in S_{I_t,t}\}$   and   $q^{(b_t)}_{i,t} = \sum_{j\,:\,j \xrightarrow{t} i} p^{(b_t)}_{j,t}$ .
12: end for
```

4.3 Bandit Online Linear Optimization

In this section we take a look at the main results from (Hazan & Kale, 2009). Specifically the algorithms used in the paper are presented and elaborated on. The main algorithm, given in algorithm 3, is composed of two prominent steps that are alternated with some probability. This algorithm uses reservoir sampling and the properties of Dikin-ellipsoids to efficiently perform online bandit linear optimization that, as shown, can be bounded in terms of total variation.

The algorithm has three main parts: calling SimplexSample with some probability, calling EllipsoidSample with the remaining probability mass, and updating the value x_t at every round. At a high level the SimplexSample is an exploration step, while the EllipsoidSample is an exploration and exploitation step.

Lines 1-4 initialises various quantities. Seen listed are the parameters of the algorithm: η is the exploration/exploitation trade off rate, the algorithm is $\mathcal{V}$-self-concordant via $\mathcal{R}$, and a size parameter k denotes the size of the reservoir kept for each of coordinate. The variable x is also initialised to be the index of the minimum point in $\mathcal{R}(x)$; the estimate for the mean loss of each action, zero.

Then, at time t some action is taken in a for-loop until time T. At each round the algorithm has the choice of exploring alone with a SimpleSample or exploring and exploiting with an EllipsoidSample step. The proportion of time spent in either of these tasks is determined by the parameter η and the stage of the optimization as denoted by t. The reservoir of size k is also taken into account due to how it affects the accuracy of our estimates of the mean losses being estimated. More generally, let the proportion of time spent exploring grows smaller as time goes on (time spent in SimplexSample) and the proportion of time spent exploring and exploiting increases in later stages (time spent in EllipsoidSample).

Clearly, the estimate of the mean $\tilde{\mu}$ of each coordinate computed by the reservoir sampling procedure only gets updated during the SimplexSample call, but only gets used in the EllipsoidSample call. Intuitively this shpws that exploitation is akin to using the information of losses that have been collected for each action while exploration involves gathering this information. In line with that $\tilde{f}_t$ is not updated when in the SimplexSample step, but only in the EllipsoidSample step.

Algorithm 3 Bandit online linear optimization

```
 1: Input: $\eta > 0, \mathcal{V}$-self-concordant $\mathcal{R}$, resevoir size parameter $k$
 2: Initialization: for all $i \in [n], j \in [k]$ set $S_{i,j} = 0$ Set
    $x_i = argmin_{x \in \mathcal{K}}[\mathcal{R}(x)]$ and $\tilde{\mu_o} = 0$
    Let $\pi : \{1, 2, ..., nk\} \to \{1, 2, .., nk\}$ be a random
    permutation.
 3: for $t = 1$ to T do
 4:   Set $r = 1$ with probability min $\{\frac{nk}{t}, 0\}$, and 0
      with probability $1 -$ min $\{\frac{nk}{t}, 1\}$
 5:   if r = 1 then
 6:
 7:     if $t \leq nk$ then
 8:       Set $i_t = (\pi(t) \bmod n) + 1$
 9:     else
10:       Set $i_t$ uniformly at random from $\{1, 2, ..., n\}$
11:     end if
        Set $\tilde{\mu_t} \leftarrow SIMPLEXSAMPLE(i_t)$
        Set $\tilde{f}_t = 0$
12:   else
13:     Set $\tilde{\mu_t} = \mu_{t-1}$
        Set $\tilde{f}_t \leftarrow ELLIPSOIDSAMPLE(x_t, \tilde{\mu_t})$
14:   end if
      $x_{t+1} = argmin_{x \in \mathcal{K}}[\eta \sum_{\tau=1}^{t} \tilde{f_\tau^T} x + \mathcal{R}(x)]$
15: end for
```

Algorithm 4 SimplexSample(i_t)

```
 1: Predict $y_t = \gamma e_{i,t}$that is, the $i_t$-th standard basis
    vector scaled by $\gamma$
 2: Observe the cost $f_t^T y_t = f_t(i_t)$
 3: if some bucket for $i_t$ is empty then
 4:   Set $j$ to the index of empty bucket
 5: else
 6:   Set j uniformly at random from $\{1, ..., k\}$
 7: end if
 8: Update the sample $S_{i,j} = \frac{1}{\gamma} f_t(i_t)$
 9: if $t \leq nk$ then
10:   Return $\tilde{\mu_t} = 0$
11: else
12:   Return $\tilde{\mu_t}$ defined as : $\forall i \in \{1, 2, ..., n\}$ Set
      $\tilde{\mu_t}(i) := \frac{1}{k} \sum_{j=1}^{k} S_{i,j}$
13: end if
```

Algorithm 5 EllipsoidSample(x_t, $\tilde{u}_t$)

```
 1: Let $\{v_1, .., v_n\}$ and $\{\lambda_1, ..., \lambda_n\}$ be the set of orthog-
    onal eigenvectors and eigenvalues of $\nabla^2 \mathcal{R}(x_t)$
 2: Choose $i_t$ uniformly at random from $\{1, ..., n\}$ and
    $\epsilon_t \pm 1$ with probability $\frac{1}{2}$
 3: Predict $y_t = x_t + \epsilon_t \lambda_{i_t}^{-\frac{1}{2}} v_{i_t}$
 4: Observe the cost $f_t^T y_t$
 5: Return $\tilde{f}_t$ defined as :$\tilde{f}_t = \tilde{\mu}_t + \tilde{g}_t$
    Where $\tilde{g}_t := n(f_t^T y_t - \tilde{\mu_t} y_t) \epsilon_t \lambda_{i_t} v_{i_t}$
```

4.3.1 SimplexSample

Turning our attention to the SimplexSample procedure, its job is to implement reservoir sampling on all the points in the feasible set. SimplexSample samples a random coordinate $i \in [n]$ uniformly, the actual sampled point y_t is the corresponding vertex γe_{i_t}. This vertex is in the scaled n-dimensional simplex and, by assumption, it has to be contained inside $\mathcal{K}$. The loss is immediately received as $f_t(i_t)$ after which the resevoir sample is done: if one of the slots in the reservoir for that coordinate is empty then the loss is put into that slot, otherwise a random element of the resevoir is discarded and the received loss is inserted. The discarding is done so at random, uniformly. In this way reservoir sampling is implemented correctly and guarantees an unbiased estimate of the mean loss for that coordinate in the limit. In the return step the entire vector of means to be used in the main algorithm is handed back. Every element of that vector is the estimate of the mean of the coordinate denoted by its index, at the current round t.

4.3.2 EllipsoidSample

EllipsoidSample exploits the algorithm's knowledge of the adversary (loss surface) by using the estimate of the mean. This is a modification of a similar procedure outlined in (Abernethy *et al.* , 2008). (Abernethy *et al.* , 2008) manages to prove that this style of sampling procedure is unbiased and has low variation with respect to the regularization which in this case corresponds to the self-concordant functions. Like in SimplexSample, a point y_t is chosen, however, this time it is chosen from the endpoints of the principal axes of the Dikin ellipsoid $W_1(x_t)$ centered at x_t. Recall that x_t was constructed to minimize the FTRL loss $\eta \sum_{\tau=1}^{T} \tilde{f}_t^T x + \mathcal{R}$. Since the loss minimized was in terms of the estimate of the loss vector, this is where the algorithm exploits its prior knowledge of the losses associated with the coordinates. Furthermore, the estimate of $\tilde{f}_t$ is updated at the end of EllipsoidSample to incorporate the new knowledge of the mean $\tilde{\mu_t}$ and the actual loss suffered f_t. It is done by adding f_t to $\eta(f_t^T y_t - \tilde{\mu}_t^T y_t)\epsilon\lambda_{it}^{\frac{1}{2}} v_{it}$.

The procedure is quite akin to the exploration-exploitation steps taken in FTRL or Exp3, but there is a lot of mathematical machinery needed to ensure that the sampled point y_t will be in the feasible set and that the ellipsoid sampling procedure has nice properties that ensure correctness and efficiency.

5 Connections and Overlap

There are some interesting connections between the papers at a high level that are worth considering. One is the notion of using mathematical structure to create a solution to a problem, and the second is the idea of generalizing a set of related problems such that every one of the specific problems are special cases in the generalization.

In all the papers a key element of the solution was a novel or sophisticated application of a mathematical tool. In (Hazan & Kale, 2009) the authors recognize that they can augment previous approaches that used self-concordant functions and FTRL with resevoir sampling to extend those solutions to the bandit setting.

In (Alon *et al.* , 2014) the authors use graphs to represent the mutual information sets that might exist in a general online learning problem to interpolate between the bandit and full-information setting cleanly.

In (Bertsimas & Niño-Mora, 2000) the key result of the paper was a novel approach to solving the restless bandit problem that was due to the realization that there was a stronger relationship between the polytope of the LP and the restless bandit feasible set than previously realized.

Another overarching theme was generalizing a problem such that previously disparate problems are special cases of the generalization. This was expressed in both (Hazan & Kale, 2009) and (Alon *et al.* , 2014). In (Hazan & Kale, 2009) they do this by phrasing regret in terms of total variation, thus having no variation takes you to the full information setting, and having lots of variation to the adversarial bandit.

In (Alon *et al.* , 2014) they express the information sets in an online learning problem via graphs, such that edges between vertices means that rewards for one are recieved when they are for the other. In this way, setting the graph to be fully connected takes you to the full information setting and removing all edges to the bandit.

6 Open Problems

With regards to open problems, (Hazan & Kale, 2009) asks if the regret bounds can be improved and if one can bound the regret based on the variation of the best expert in hindsight. In (Alon *et al.* , 2014) the authors asks if one can make the revealed feedback graph dependent on the specific loss obtained at that round. Lastly, (Abernethy *et al.* , 2008) conjectures that the Exp3 could be bounded by the total variation of losses of the experts. While these papers provides a strong platform for diving into this problem, unfortunately we did not manage such a solution.

References

Abernethy, Jacob, Hazan, Elad, & Rakhlin, Alexander. 2008. *Competing in the dark: An efficient algorithm for bandit linear optimization*. Pages 263–273.

Alon, Noga, Cesa-Bianchi, Nicolò, Gentile, Claudio, Mannor, Shie, Mansour, Yishay, & Shamir, Ohad. 2014. Nonstochastic Multi-Armed Bandits with Graph-Structured Feedback. *CoRR*, **abs/1409.8428**.

Bertsimas, Dimitris, & Niño-Mora, José. 2000. Restless Bandits, Linear Programming Relaxations, and a Primal-Dual Index Heuristic. *Operations Research*, **48**(1), 80–90.

Hazan, Elad, & Kale, Satyen. 2009. Better Algorithms for Benign Bandits. *Pages 38–47 of: Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '09. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics.

Papadimitriou, Christos H., & Tsitsiklis, John N. 1999. The Complexity of Optimal Queuing Network Control. *Math. Oper. Res.*, **24**(2), 293–305.

Vitter, Jeffrey S. 1985. Random Sampling with a Reservoir. *ACM Trans. Math. Softw.*, **11**(1), 37–57.

Whittle, P. 1988. Restless bandits: activity allocation in a changing world. *A Celebration of Applied Probability. J. Appl. Probab.*, **25A 287-298**, 287–298.