ARL-TR-8754 • Aug 2019

DEVCOM ARMY RESEARCH LABORATORY

Multi-Armed Bandits with Delayed and Aggregated Rewards

by Jacob Tyo, Ojash Neopane, Jonathon Byrd,
Chirag Gupta, and Conor Igoe

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8754 ● AUG 2019

Multi-Armed Bandits with Delayed and Aggregated Rewards

by Jacob Tyo
Sensors and Electron Devices Directorate, CCDC Army Research Laboratory

Ojash Neopane, Jonathon Byrd, Chirag Gupta, and Conor Igoe
Carnegie Mellon University

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
August 2019	Technical Report	November 2018–February 2019

4. TITLE AND SUBTITLE
Multi-Armed Bandits with Delayed and Aggregated Rewards

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Jacob Tyo, Ojash Neopane, Jonathon Byrd, Chirag Gupta, and Conor Igoe

5d. PROJECT NUMBER
AH80

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Combat Capabilities Development Command Army Research Laboratory
ATTN: FCDD-RLS-SA
Adelphi Laboratory Center, MD 20783-1138

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-8754

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
primary author's email: <jacob.p.tyo.civ@mail.mil>.

14. ABSTRACT
We study the canonical multi-armed bandit problem under delayed feedback. Recently proposed algorithms have desirable regret bounds in the delayed-feedback setting but require strict prior knowledge of expected delays. In this work, we study the regret of such delay-resilient algorithms under milder assumptions on delay distributions. We experimentally investigate known theoretical performance bounds and attempt to improve on a recently proposed algorithm by making looser assumptions on prior delay knowledge. Further, we investigate the relationship between delay assumptions and marking an arm as suboptimal.

15. SUBJECT TERMS
multi-arm bandits, delayed feedback, aggregated feedback, best-arm-identification

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Jacob Tyo
Unclassified	Unclassified	Unclassified	UU	20	19b. TELEPHONE NUMBER (Include area code) 240-687-0134

Contents

List of Figures

1. Introduction and Problem Setting

Since its original formulation by Thompson,[1] the multi-armed bandit (MAB) problem has received much study in its various forms. However, the overwhelming majority of MAB research focuses on instances of the problem in which rewards are assumed to arrive *instantaneously* after choosing an action. Such an assumption is ill suited to many real-world problems. For example, in medical applications, it is typically unreasonable to assume that the impact of a chosen treatment, out of a set of possible treatments for a patient, will be immediately observable or observable after some known delay.[2] Moreover, in the case of testing multiple treatments sequentially, it is often difficult to determine the specific treatment (or treatments) that induced a change in patient health. Such problems are well modelled as a MAB problem with delayed, aggregated anonymous feedback.

In this work, we focus on the setting considered in Pike-Burke et al.[3] We have $K > 1$ arms in the set $\mathcal{A}$ available to the agent at each timestep. As in the canonical setting, we have a reward distribution ζ_j associated with each arm. Importantly, with each arm, we also have a delay distribution δ_j defined on $\mathbb{N}$ from which we assume delays are drawn independent and identically distributed (i.i.d.) with each arm choice. Let μ_j denote the mean of ζ_j and $\mu^* = \mu_{j^*} = \max_j \mu_j$. We define $\Delta_j = \mu^* - \mu_j$ to be the reward gap between the j^{th} and optimal arm. Let $R_{l,j}$ follow the distribution ζ_j and $\tau_{l,j}$ follow δ_j, where $l = 1, 2, \ldots$ denotes the timestep. We define $J_l \in \mathcal{A}$ to denote the action chosen by the agent at time l. The observation received at the end of the tth play is given by

$$X_t = \sum_{l=1}^{t} \sum_{j=1}^{K} R_{l,j} \times \mathbb{1}\{l + \tau_{l,j} = t, J_l = j\}. \tag{1}$$

We emphasize the aggregated nature of the received rewards in the previous definition: the agent does not learn the individual contributions that compose the received reward at each timestep but instead observes their aggregation (summation).

As in Pike-Burke et al.,[3] we consider the loss of all *generated* rewards and define the regret by

$$\mathfrak{R}_T = \sum_{t=1}^{T} (\mu^* - \mu_{J_t}) = T\mu^* - \sum_{t=1}^{T} \mu_{J_t}. \tag{2}$$

Under degenerate delay distributions, the setting described in the previous subsection reduces to the canonical MAB problem with instantaneous feedback. Such problems have been widely studied in statistics, with the celebrated upper confidence bound (UCB) algorithm taking principal focus in the majority of the literature since its original introduction by Lai.[4] Roughly, UCB maintains upper confidence bounds on all arms, and at any moment in time, the arm with the highest bound is chosen. A characteristic of UCB is frequent switching of sequential arm choices before an optimal arm becomes apparent. Consequently, UCB is ill suited to the delayed aggregated anonymous feedback setting, where rapid switching of arms is likely to introduce excessive difficulty in reward–action assignment. Recently, Pike-Burke et al.[3] proposed Optimism for Delayed, Anonymous, Aggregated Feedback (ODAAF), a batch/round-based algorithm capable of accommodating anonymous delays under certain strict delay distribution assumptions:

Assumption 1 (Bounded expectation)**.** $\mathbb{E}[\tau]$ *is bounded and known to the algorithm.*

Assumption 2 (Bounded support)**.** *There exists some constant* $d > 0$ *known to the algorithm such that the support of the delay distribution is bounded by* d*.*

Assumption 3 (Bounded variance)**.** $\mathbb{V}(\tau)$ *is bounded and known to the algorithm.*

In essence, the core focus of this work involves extending results in Pike-Burke et al.[3] to accommodate milder assumptions on delay. Sections 3 and 4 discuss this direction more concretely.

2. Related Works and Baseline Algorithms

The delayed feedback MAB setting has only recently received attention in the MAB literature, with the first major theoretical contribution from Joulani et al.[5] and later followed up by Mandel et al.[6] As discussed in the previous subsection, the first work to study the delayed, aggregated, anonymous feedback setting is by Pike-Burke et al.[3] In this section, we briefly review the contributions of relevant related works and discuss the baseline algorithms chosen for our experiments.

The baseline algorithms chosen for this experiment are UCB, delayed UCB, and ODAAF (with three variations). We propose two new algorithms that build upon ODAAF to improve efficiency and accumulate less regret.

2.1 Upper Confidence Bound

The UCB algorithm proposed by Auer[7] is well known and commonly used. Here, we use it as a "sanity" check, because due to the environment, this should act similarly to taking random actions at every step.

2.2 Queued Partial Monitoring with Delays

Queued partial monitoring with delays (QPMD)[5] is a black-box meta-algorithm that adapts algorithms for the stochastic i.i.d. partial monitoring setting[8] to handle delayed, *but not anonymous*, feedback. In the MAB setting, QPMD queues rewards observed from past actions, before updating a base MAB algorithm such as UCB with the delayed rewards.

2.3 Optimism for Delayed, Aggregated, Anonymous Feedback

The ODAAF algorithm was proposed by Pike-Burke et al.[3] and is the only existing algorithm developed specifically for the delayed, aggregated, anonymous feedback MAB setting. This procedure uses an iterated phased approach. Each iteration consists of three phases. The first phase samples each arm. The second phase uses the data gathered in phase one to compute estimates of the mean reward for each arm, eliminate suboptimal arms based on a tolerance factor, and then update the tolerance factor for the next iteration. Lastly, the third phase is a bridge period where the estimated best arm is pulled for some number of steps before repeating this process until the horizon is reached.

The most important aspect of this algorithm is how the constant that determines how many times to pull each arm is determined. There are three methods for determining these, each of which uses different assumptions. Equation 3 uses the assumption that the expected delay of the arm with the largest expected delay is known. Equation 4 assumes the same expectation is known as in Eq. 3, but also that there exists an upper bound on the delay. Lastly, Eq. 5 makes the same assumptions as Eq. 4 but also assumes the variance is known. These are referred to as "odaaf_ed", "odaaf_ebd", and "odaaf_bdev", respectively.

$$n_m = \left\lceil \frac{1}{\Delta_m^2} \left(\sqrt{2\log(T\Delta_m^2)} + \sqrt{2\log(T\Delta_m^2) + \frac{8}{3}\Delta_m \log(T\Delta_m^2) + 6\Delta_m m\mathbb{E}[\tau]} \right)^2 \right\rceil \tag{3}$$

$$n_m = \max\left\{ md_m, \left\lceil \frac{1}{\Delta_m^2} \left(\sqrt{2\log(T\Delta_m^2)} + \sqrt{2\log(T\Delta_m^2) + \frac{8}{3}\Delta_m \log(T\Delta_m^2) + 6\Delta_m m\mathbb{E}[\tau]} \right)^2 \right\rceil \right\} \tag{4}$$

$$n_m = \left\lceil \frac{1}{\Delta_m^2} \left(\sqrt{2\log(T\Delta_m^2)} + \sqrt{2\log(T\Delta_m^2) + \frac{8}{3}\Delta_m \log(T\Delta_m^2) + 4\Delta_m m\mathbb{E}[\tau] + 2\mathbb{V}(\tau)} \right)^2 \right\rceil \tag{5}$$

2.4 Phaseless Hedger

Here we propose our algorithms for the delayed, aggregated, anonymous feedback setting. Roughly, our algorithms are inspired by the original ODAAF algorithm modified to pull arms in a more conservative fashion. More specifically, we take advantage of previous observations to trade off exploring a new arm with exploiting a previously determined good arm while waiting for the delayed rewards to arrive.

Algorithm 1 Hedger (No Phases)

$\mathcal{A}$: Set of Arms, T: Horizon
$K \coloneqq |\mathcal{A}|$, $n \coloneqq \lfloor T/K \rfloor$, $\Delta_m^2 = 0.5$, $m \coloneqq \sqrt{2\log(T\Delta_m^2)}$
procedure NOPHASEHEDGER($\mathcal{A}$, T)
    Sample arm a_1 n times
    Set $\hat{\mu}_1 = \frac{1}{n}\sum_{t=1}^{n} X_t$
    Sample arm a_2 n times
    Set $\hat{\mu}_2 = \frac{1}{n}\sum_{t=n+1}^{2n} X_t$
    for $i = 3, \ldots, K$ **do**
        Sample a_i m times
        Sample $a_{\max}^i = \max\{a_1, \ldots, a_{i-1}\}$ $(n-m)$ times
        Set $\hat{\mu}_i = \frac{\sum_{t=(i-1)n}^{i\cdot n} X_t - n\hat{\mu}_{\max}^i}{(n-m)}$
    end for
    Sample arm $\max\{a_1, \ldots, a_K\}$ $(T - k\lfloor T/k \rfloor)$ times
end procedure

Algorithm 2 Hedger

$\mathcal{A}$: Set of Arms, T: Horizon
$K := |\mathcal{A}|$, n as in Equation 3, $\Delta_m^2 = 0.5$, $m := \sqrt{2\log(T\Delta_m^2)}$
procedure HEDGER($\mathcal{A}$, T)
    while t < T **do**
        Sample arm a_1 n times
        Set $\hat{\mu}_1 = \frac{1}{n}\sum_{t=1}^{n} X_t$
        Sample arm a_2 n times
        Set $\hat{\mu}_2 = \frac{1}{n}\sum_{t=n+1}^{2n} X_t$
        for $i = 3, \ldots, K$ **do**
            Sample a_i m times
            Sample $a_{\max}^i = \max\{a_1, \ldots, a_{i-1}\}$ $(n-m)$ times
            Set $\hat{\mu}_i = \frac{\sum_{t=(i-1)n}^{i \cdot n} X_t - n\hat{\mu}_{\max}^i}{(n-m)}$
        end for
        Remove arms i which satisfy $\hat{\mu}_i + \Delta_m^2 < \max_j \hat{\mu}_j$
        Set $\Delta_m^2 = \Delta_m^2/2$
    end while
end procedure

3. Theory

In this section, we discuss the theoretical results we were able to prove. We prove expected regret bounds on the phaseless hedger. The following is the main result:

Theorem 3.1. *The expected regret of Algorithm 1 after T timesteps is upper-bounded by*

$$\mathbb{E}[\mathfrak{R}_T] \leq \frac{2\mathbb{E}^{(1)}[\mathfrak{R}_T] + (K-2)\mathbb{E}^{(2)}[\mathfrak{R}_T]}{K}, \tag{6}$$

where $\mathbb{E}^{(1)}$ is as defined in Eq. 7 and $\mathbb{E}^{(2)}$ is as defined in Eq. 12.

To upper-bound the expected regret, we consider two cases: 1) when the optimal arm a_* is either the first or second arm and 2) when the optimal arm $a_* = a_j$ for some $j \geq 3$. Then, after proving the regret bounds in each of the two cases, we take the average over the cases to obtain an upper bound for the overall algorithm.

Lemma 3.2 (Regret Bound for Case 1)**.** *Suppose that $a_* = a_1$ or $a_* = a_2$ and that with probability at least $1 - \delta_i^{(1)}$, $a_* = max\{a_1, \ldots, a_i\}$ for each $i \in \{3, \ldots, K\}$, then we observe an expected regret of*

$$\mathbb{E}[\mathfrak{R}_T] \leq \tilde{\Delta}_m(n(1 + \delta^{(1)}(K-2)) + m(K-2)(1-\delta^{(1)})). \tag{7}$$

Proof. First, we have that either arm 1 or arm 2 is nonoptimal. In this case, we observe a regret contribution of $n\tilde{\Delta}_m$. Then, for each arm a_i for $i = 3, \ldots, K$ we observe a regret of $m\tilde{\Delta}_m + \delta_i^{(1)}\tilde{\Delta}_m(n-m)$. Letting $\delta^{(1)} = \max_{i \in \{3,\ldots K\}}$ and summing over these terms, our expected regret is upper-bounded by

$$\begin{aligned}
\mathbb{E}^{(1)}[\mathfrak{R}_T] &\leq n\tilde{\Delta}_m + (K-2)[m\tilde{\Delta} + \delta^{(1)}\tilde{\Delta}_m(n-m)] && (8)\\
&= \tilde{\Delta}_m(n + (K-2)[m + \delta^{(1)}(n-m)]) && (9)\\
&= \tilde{\Delta}_m(n + (K-2)[\delta^{(1)}n + m(1-\delta^{(1)})] && (10)\\
&= \tilde{\Delta}_m(n(1 + \delta^{(1)}(K-2)) + m(K-2)(1-\delta^{(1)})). && (11)
\end{aligned}$$

□

Lemma 3.3 (Regret Bound for Case 2)**.** *Suppose that $a_* = a_j$ for $j \geq 3$ and that with probability at least $1 - \delta_j^{(2)}$, $a_* = max\{a_1, \ldots a_{i-1}\}$ for each $i \in \{j+1, \ldots, K\}$, then the expected regret of Algorithm 1 after T timesteps is upper-bounded by*

$$\mathbb{E}[\mathfrak{R}_T] \leq \tilde{\Delta}_m(n(j(1-\delta^{(2)}) + \delta^{(2)}k) + m(1 + \delta^{(2)}(K-j))). \tag{12}$$

Proof. For $i \in \{1, \ldots, j\}$, we observe a regret of $n\tilde{\Delta}_m$. Multiplying by j, we observe a total regret of $j \cdot n\tilde{\Delta}_m$. Letting $\delta^{(2)} = \max_{i \in \{j+1,\ldots,K\}}$m, we observe an expected regret of $m\tilde{\Delta}_m + \delta^{(2)}\tilde{\Delta}_m(n-m)$ so that we observe a total expected regret of $m\tilde{\Delta}_m + \delta^{(2)}\tilde{\Delta}_m(n-m)(K-j)$. Summing these two terms, we have

$$\begin{aligned}
\mathbb{E}^{(2)}[\mathfrak{R}_T] &\leq j \cdot n\tilde{\Delta}_m + m\tilde{\Delta}_m + \delta^{(2)}\tilde{\Delta}_m(n-m)(K-j) && (13)\\
&= \tilde{\Delta}_m(jn + m + \delta^{(2)}Kn - \delta^{(2)}Km - \delta^{(2)}jn + \delta^{(2)}jm) && (14)\\
&= \tilde{\Delta}_m(n(j(1-\delta^{(2)}) + \delta^{(2)}k) + m(1 + \delta^{(2)}(K-j))). && (15)
\end{aligned}$$

□

We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. We can see that for each arm, with probability $1/K$, the following holds:

$$a_* = a_i \; \forall i \in [K].$$

Therefore, with probability $2/K$, Case 1 will occur and, with probability $(K-2)K$, Case 2 will occur. Taking a weighted average, we get that the expected regret is upper-bounded by

$$\mathbb{E}[\mathfrak{R}_T] \leq \frac{2\mathbb{E}^{(1)}[\mathfrak{R}_T] + (K-2)\mathbb{E}^{(2)}[\mathfrak{R}_T]}{K}. \tag{16}$$

□

3.1 Discussion

In Lemmas 3.2 and 3.3, we make the assumption that the event of interest occurs with probability $1 - \delta^{(1)}$ and $1 - \delta^{(2)}$, respectively. While we were not able to derive nontrivial lower bounds on these probabilities, we breifly discuss potential directions for deriving these lower bounds following the proofs from Pike-Burke et al.[3] We restate the result of interest from Pike-Burke et al.[3] and highlight the techniques they use in deriving the result that we believe to be pertinent to our use-case.

Lemma 3.4 (Lemma 1 from[3])**.** *For every fixed arm j and phase m, with probability* $1 - \frac{3}{T\Delta_m^2}$*, either* $j \notin \mathcal{A}_m$ *or* $\bar{X}_{m,j} - \mu_j \leq \tilde{\Delta}_m/2$.

The general idea in this is to decompose and upper-bound $\bar{X}_{m,j} - \mu_j$ as a sum of two martginale terms and then use Freedman's Inequality[9] to provide high-probability bounds on these events. This martingale decomposition implicitly deals with the delays in the problem and we believe that a similar decomposition can be found for our algorithm.

3.2 New Notions of Regret

One goal is to remove Assumptions 1, 2, and/or 3. Suppose we do not know the expected value of delay (Assumption 1), and the best possible arm has a very high delay. It is clearly impossible to achieve low regret with respect to this arm. We argue that in practice, instead of waiting for the optimal but highly delayed arm, it is best to ignore that arm. To this end, we consider new notions of the best possible arm:

- The best arm normalized with the expected value of delay:

$$\arg\max_i\{\mu_i/\mathbb{E}\left[\tau_i\right]]\}. \tag{17}$$

- The best arm with expected delay within a certain delay tolerance $\mathcal{D}$:

$$\arg\max_i \left\{\mu_i \mathbb{I}\left[\mathbb{E}\left[\tau_i\right] < \mathcal{D}\right]\right\}. \tag{18}$$

- The best arm with delay smaller than a delay tolerance $\mathcal{D}$ with high probability:

$$\arg\max_i \left\{\mu_i \mathbb{I}\left[P\left(\tau_i < \mathcal{D}\right) \geq \delta\right]\right\}. \tag{19}$$

One way to decide $\mathcal{D}$ is to set it to be a monotonically increasing function of a fixed horizon T, for example, 1) $\mathcal{D} = C\sqrt{T}$ or 2) $\mathcal{D} = C \log T$. Suppose we define $\mu^*(\mathcal{D})$ using one of these strategies, then the new notion of "tolerance-aware" regret with respect to a tolerance $\mathcal{D}$ is given by

$$\mathfrak{R}_T(\mathcal{D}) = T\mu^*(\mathcal{D}) - \sum_{t=1}^{T} \mu_{J_t}. \tag{20}$$

4. Experiments

4.1 Environment

Each of the algorithms mentioned in Section 2 along with the baseline algorithms of UCB,[10] QPMD[5] using UCB as the baseline (referred to as delayed UCB), and ODAAF[3] were implemented. Using the following environment, we developed several experiments to test the algorithms performance according to different environment dynamics. All experiments use a 10-arm bandit problem where the rewards of each arm are distributed as truncated normal distributions with means linearly spaced between 0.2 and 0.8, variance$= 0.1$, and truncated so that all rewards $\in [0, 1]$. The result of each experiment is averaged over 50 separate trials, where the order of the arms is randomly shuffled each time. Delay distributions are randomly assigned to arms each trial, unless otherwise noted. The performance of delayed UCB represents a sort of lower bound on performance, as the rewards are nonanonymous for this algorithm. Our implementations of delayed bandit environments and algorithms can be found at https://github.com/JacobTyo/gym-bandits/tr

ee/AnonymousDelayedBandits.

4.2 Experiment 1

The first experiment (referred to as Experiment 1) is aimed at understanding the performance of each algorithm in a simple case adhering to the delayed, aggregated, anonymous MAB ideology. Here, the delays are Poisson distributed with $\lambda \in \{1, 2, \dots 10\}$. The results of this can be seen in Fig. 1.

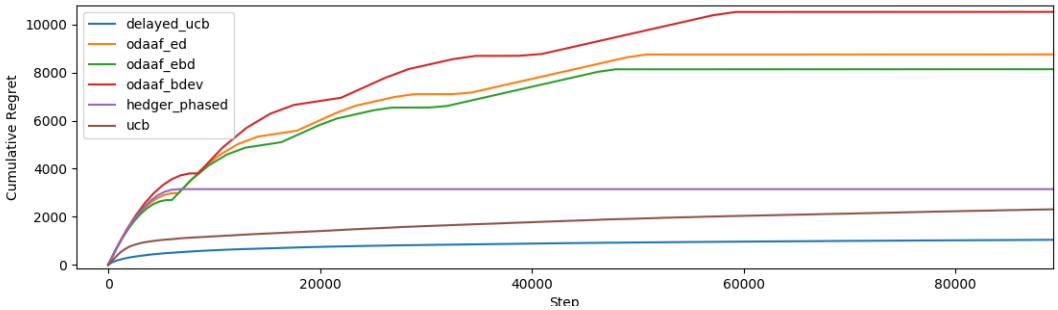


Fig. 1 This figure displays the cumulative regret vs. time on the environment described for Experiment 1

4.3 Experiment 2

Experiment 2 aims to investigate the performance of each algorithms when the delay is large. Thus, it is identical to Experiment 1, but the delays are instead Poisson distributed with $\lambda \in \{100, 200, \dots, 1000\}$. The results can be seen in Fig. 2.

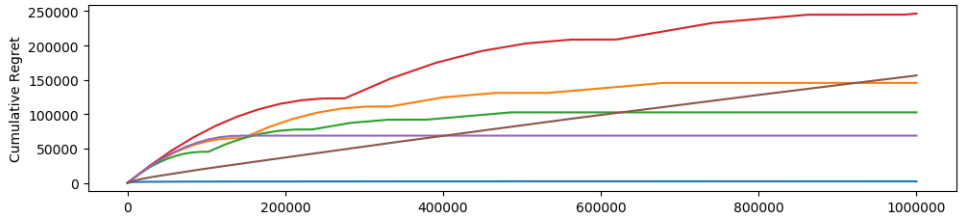
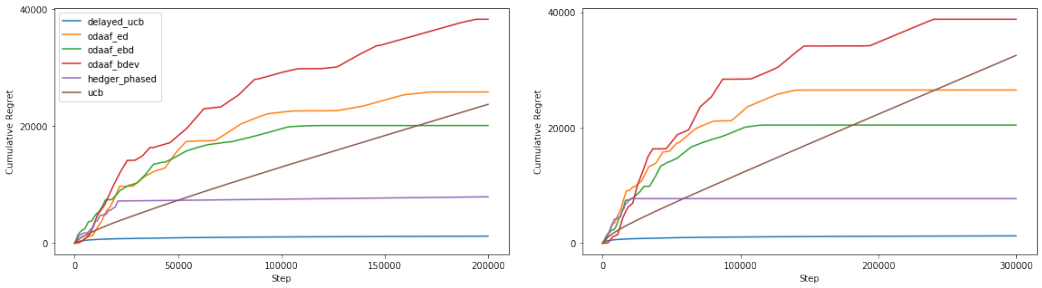


Fig. 2 This figure displays the cumulative regret vs. time on the environment described for Experiment 2. We use the same legend as in Fig. 2.

4.4 Experiment 3

Experiment 3 investigates how the presence of outliers affected each algorithm. The setup in this case was again the same as Experiment 1, except a random arm has

delays distributed as Poisson(100) instead of Poisson(10). The results are shown in Fig. 3.

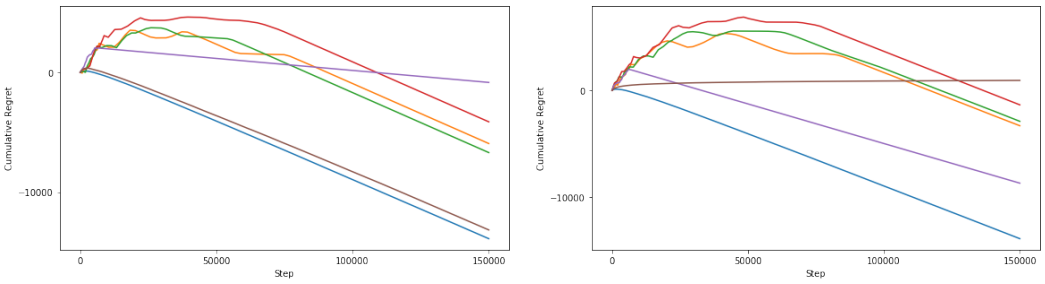


(a) Nonoptimal outlier arm **(b) Optimal outlier arm**

Fig. 3 Results from Experiment 3 with an outlier arm that las very long delays. a) shows results where the long-delayed arm is not the optimal arm, while b) shows results where the long-delayed arm also gives the highest expected reward. The Y-axis shows cumulative regret while the X-axis represents time steps.

4.5 Experiment 4

The previous algorithms have assumed that all of the information presented to each algorithm was correct (i.e., the expected delay was actually the expected delay). However, in some cases (such as in advertising), this information can only be approximated. As noted in Section 3, we can redefine our notion of regret to account for our uncertainty. Experiment 4 tests each algorithm when the expected delay parameter is approximated incorrectly. Specifically, we use the bandit problem from Experiment 3, but with the false assumption that there is no outlier (the assumptions are based off of the bandit problem from Experiment 1). We use the notion of regret as defined in Eq. 18. The results are shown in Fig. 4.



(a) Nonoptimal outlier arm **(b) Optimal outlier arm**

Fig. 4 Results from Experiment 4 with false assumptions on delay distribution and alternate definition of regret. The Y-axis shows cumulative regret while the X-axis represents timesteps.

4.6 Experiment 5

The Hedger algorithm has an intrinsic dependency on the ordering of the arms. Experiment 5 highlights that this dependency results in some failure cases. This experiment was set up in the same manor as Experiment 1, except the arms were ordered from worst to best. This forced the Hedger algorithm to pull suboptimal arms more and often resulted in the wrong estimate for the optimal arm. These results can be seen in Fig. 5.

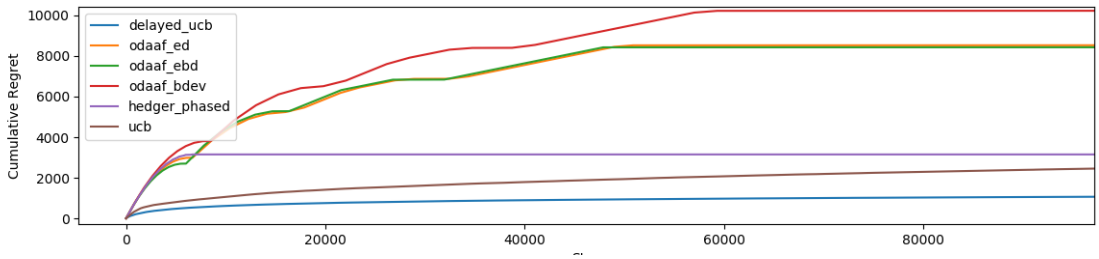


Fig. 5 This figure displays the cumulative regret vs. time on the environment described for Experiment 5.

5. Discussion and Conclusion

Hedger outperforms ODAAF in almost every experiment we ran by converging to the optimal arm more quickly (Figs. 1–5). It only accumulated more regret than ODAAF in Experiment 4 with a high-delay outlier arm, which was also suboptimal, under false assumtions (Fig. 4). As expected, delayed UCB outperformed all other

algorithms in every setting. Unexpectedly, the UCB algorithm performed well in settings where the expected delay was small (Figs. 1 and 5).

We expected Hedger to outperform ODAAF by accumulating less regret early on. However, it seems that instead Hedger is able to find the optimal arm more quickly. This is likely related to the phase length parameter n_m, which is chosen to achieve certain bounds on total regret. It would be interesting to instead investigate more empirically motivated choices of n_m for both Hedger and ODAAF.

One of the most interesting observations of this research was the performance of the UCB algorithm. This algorithm was implemented as a baseline and was expected to do no better than random guessing. However, this is not the case. This seems to be because UCB becomes confident enough in a single arm relatively quickly, and therefore, begins pulling it continuously. However, as this happens, it builds a good representation of the true mean reward of the arm. In doing this, it will often switch to another arm and repeat this process until it becomes confident that it has selected the correct arm. In other words, it seems that UCB implicitly takes on a phase-like approach in the delayed, aggregated, anonymous feedback setting. It would be interesting to compare this to the performance of the phase-based improved UCB.[11]

Before this report, the only existing algorithm to the delayed, aggregated, anonymous MAB setting proved to be effective, but often accumulated needless regret. We present a new method that builds on this original work by differentiating between the number of samples needed per arm to get an accurate estimate of said arm versus the number of pulls required to ensure that the delayed rewards are observed and correctly attributed.

6. References

1. Thompson WR. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika. 1933;25:285–294.

2. Sampford MR. The estimation of response-time distributions. I. Fundamental concepts and general methods. Biometrics. 1952;8(1):13–32.

3. Pike-Burke C, Agrawal S, Szepesvari C, Grunewalder S. Bandits with delayed, aggregated anonymous feedback. International Conference on Machine Learning; 2018. 4101–4110.

4. Lai TL. Adaptive treatment allocation and the multi-armed bandit problem. Annals of Statistics. 1987:1091–1114.

5. Joulani P, Gyorgy A, Szepesvári C. Online learning under delayed feedback. International Conference on Machine Learning. 2013:1453–1461.

6. Mandel T, Liu Y-E, Brunskill E, Popovic Z. The queue method: handling delay, heuristics, prior data, and evaluation in bandits. AAAI; 2015. 2849–2856.

7. Auer P. Using confidence bounds for exploitation-exploration trade-offs Journal of Machine Learning Research. 2002;3:397–422.

8. Bartók GP, Foster D, Pál D, Rakhlin A, Szepesvári C. Partial monitoring —classification, regret bounds, and algorithms. Mathematics of Operations Research. 2014.

9. Freedman DA. On tail probabilities for martingales. Annals of Probability. 1975:100–118.

10. Auer P. Using confidence bounds for exploitation-exploration trade-offs. J Mach Learn Res. 2003;3:397–422.

11. Auer P, Ortner R. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. Periodica Mathematica Hungarica. 2010;61(1–2):55–65.

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIR CCDC ARL IMAL HRA RECORDS MGMT FCDD RLD CL TECH LIB
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
5 (PDF)	DIR CCDC ARL FCDD RLS S A LADAS FCDD RLS SA N SROUR J HOUSER T WALKER J TYO