On-Line Adaptation of Exploration in the One-Armed Bandit with Covariates Problem

Adam M. Sykulski
Institute for Mathematical Sciences, Imperial College London, UK
adam.sykulski@imperial.ac.uk

Niall M. Adams
Department of Mathematics, Imperial College London, UK
n.adams@imperial.ac.uk

Nicholas R. Jennings
School of Electronics and Computer Science, University of Southampton, UK
nrj@ecs.soton.ac.uk

***Abstract*—Many sequential decision making problems require an agent to balance exploration and exploitation to maximise long-term reward. Existing policies that address this tradeoff typically have parameters that are set *a priori* to control the amount of exploration. In finite-time problems, the optimal values of these parameters are highly dependent on the problem faced. In this paper, we propose adapting the amount of exploration performed on-line, as information is gathered by the agent. To this end we introduce a novel algorithm, ϵ-ADAPT, which has no free parameters. The algorithm *adapts as it plays* and sequentially chooses whether to explore or exploit, driven by the amount of uncertainty in the system. We provide simulation results for the one-armed bandit with covariates problem, which demonstrate the effectiveness of ϵ-ADAPT to correctly control the amount of exploration in finite-time problems and yield rewards that are close to optimally tuned off-line policies. Furthermore, we show that ϵ-ADAPT is robust to a high-dimensional covariate, as well as misspecified models. Finally, we describe how our methods could be extended to other sequential decision making problems, such as dynamic bandit problems with changing reward structures.**

***Keywords*-Exploration-exploitation tradeoff, sequential decision making, on-line learning, one-armed bandit problem**

I. Introduction

Sequential decision making problems often require an agent to act in an environment where the potential rewards for selecting different actions are unknown and, if selected, are subsequently observed with noise. In such problems, the agent must balance the need to exploit (choosing the predicted best action) and explore (trying alternative actions for potential future benefit), in order to maximise cumulative reward. The simplest formulation of this problem is the multi-armed bandit [1], based on the analogy of a slot machine or one-armed bandit, where the agent must sequentially learn to identify the action (or arm) that yields the highest expected reward. Bandit problems have been extensively studied in the fields of statistics [2], machine learning [3], economics [4] and evolutionary programming [5] and have applications in areas as diverse as on-line auctions [6], multi-target tracking [7], web advertising [8], pricing goods [4] and clinical trials [9].

Several policies for selecting arms in bandit problems have been constructed (see [3] for an overview), including ϵ-greedy, ϵ-first, SoftMax, UCB (Upper Confidence Bound) and Interval Estimation. All of these policies require a parameter to be set that effectively *controls the amount of exploration*. In finite-time problems, however, the optimal values of these parameters are highly dependent on the type of problem faced and the length of the game [10], [11].

In real-world applications, where decision making problems are always finite-length, setting these parameter values *a priori* to their optimal value is not feasible – this requires prior knowledge of the problem faced, which is precisely what the agent is trying to learn. Moreover, in finite-time problems, certain parameter values yield poor rewards, even though they can still converge to optimal behaviour asymptotically [11]. For these reasons, in this paper we propose a method for controlling exploration in finite-time problems. In more detail, the likelihood of exploring at each iteration is driven by the amount of *uncertainty* the agent currently has – captured by calculating statistics from past interactions with the environment.

We construct a novel algorithm, ϵ-ADAPT, for adapting exploration on-line for the one-armed bandit with covariates problem – which is an important special case of the multi-armed bandit problem (see Section II for more details). This problem is studied as it is the most basic formulation of the exploration-exploitation tradeoff, where there is only one action to explore – and is an important first step in the direction of constructing autonomous algorithms for exploration in general sequential decision making problems.

In general, planning out an exploration policy for the duration of a finite-length game is an intractable computation [3], scaling exponentially in the length of the game and the number of actions available. To reduce the computation to quadratic-time, we make use of the properties of the best performing off-line policies, such that ϵ-ADAPT need only decide whether to explore or exploit for the next time-step. Our algorithm therefore *adapts as it plays* to effectively tune the exploration parameter – without the need for any other free parameters.

This is the first such algorithm for adapting exploration on-line. Previous research in bandit problems in general [2], [12], and for the one-armed bandit with covariates problem in particular [9], [13], has focused on finding policies for selecting arms that converge to optimal behaviour asymptotically, and not for designing policies that autonomously maximise reward in finite time by adapting to the type of problem faced. Moreover, ϵ-ADAPT can be generalised to control exploration in a variety of bandit problems – including problems with multiple arms and

dynamic problems with changing reward structures, we discuss this aspect more in Section V.

The structure of the paper is as follows. In Section II, we provide a background on bandit problems in general, and the one-armed bandit with covariates problem. We also motivate the need for adapting exploration on-line with some simple examples. In Section III, we describe ϵ-ADAPT and provide pseudo-code. We perform a detailed numerical analysis of ϵ-ADAPT in Section IV, including high-dimensional problems and misspecified models. Section V concludes and discusses future work.

II. Background on the Bandit Problem

In the k-armed bandit problem an agent must, at each iteration t, pull an arm a_i from the set $A = (a_1, \ldots, a_k)$. The agent receives a noisy reward $r_{a_i}(t)$, which in the simplest case is Bernoulli $\{0, 1\}$ with some unknown probability [2] or an unknown fixed constant with added noise [10]. The objective of the agent is to find a policy for selecting arms a_i at each time-step t to maximise the cumulative reward $R(T)$ where:

$$R(T) = \sum_{t=1}^{T} r_{a_i}(t), \tag{1}$$

and T is the length of the game. The rewards are sometimes discounted over time in order to guarantee convergence of an action selection policy (for example, see [12]), but this is not a requirement in our work. Nevertheless, the unknown relationship between selecting an action and the potential reward received, is the reason why the agent must balance exploration with exploitation in order to maximise $R(T)$.

The simplest policy for selecting actions is to use a *greedy* strategy and select the action that is expected to yield the highest reward, given the information the agent has. This is a pure exploitation policy – and the lack of exploration has been shown to yield suboptimal results for a variety of bandit problems [3], but perform well in problems where rewards are non-noisy or the agent naturally selects each action sufficiently while exploiting [14]. A natural extension is the ϵ-greedy policy [3], which selects the greedy action with probability $1 - \epsilon$ and uniformly picks another action with probability ϵ. The ϵ parameter is often decreased over time [11] as the agent is more certain of the optimal action, although this feature often requires a second parameter value to be set. Another variant is ϵ-first [3] which uniformly explores for the first ϵ fraction of plays and is then greedy for the remainder of play. A SoftMax policy [3], selects an action with a weighted probability based on the likelihood of it being optimal. Finally, the Interval Estimation [3], UCB [11] and Poker [10] policies select the action which has the highest optimistic reward estimate, which is inflated the most for under-explored actions.

In a variety of empirical studies [10], [11], [14], well-tuned ϵ-greedy and ϵ-first policies were found to outperform all other policies in terms of maximising cumulative reward (or minimising regret). In particular, an optimally tuned ϵ-first policy is difficult to beat in static problems (where the relationships between actions and their rewards do not change over time), as the benefit of any exploration is largest if performed early in the game. The same studies, however, found that the performance of ϵ-greedy and ϵ-first policies degraded the most rapidly as the parameter moves away from its optimal value. Due to its simplicity and potential for large reward, we use the ϵ-first approach as the building block of our algorithm, and expect noticeable benefits from controlling exploration on-line.

In this paper we construct an algorithm for adapting exploration for the special case of the one-armed bandit with covariates problem, where the agent must choose between an arm with unknown expected reward and an arm with known expected reward. The agent receives additional side information, represented in the form of a covariate, prior to each action selection decision. The one-armed bandit problem was first studied in [15] for sequential clinical trials and generalised to include covariates in [9], where it was argued that side information is likely to be present in many applications. For example, covariate information such as age, sex, height and weight could influence the probability of success of a drug in sequential clinical trials [13]. More recent studies of bandits with covariates have included [16], [17], [18]. Indeed covariates have been used in a variety of applications for sequential decision making problems including matching advertisements to web-page content, adaptive generation of multimedia messages and video compression (see [19] and references therein). Two important properties of the aforementioned applications are first, that the covariate value affects the reward of the chosen action immediately and second, the selected action does not alter the subsequent realisation of covariates[1] – which motivates the reward structures used in [14], [18].

The agent must select between actions $\{a_1, a_2\}$, where the rewards for each action are given by:

$$r_{a_1}(t) = f(X(t), \alpha) + \eta_t, \tag{2}$$

$$r_{a_2}(t) = g(X(t), \beta) + \nu_t, \tag{3}$$

where $X(t)$ is the p-dimensional covariate observed at time t. η_t and ν_t are i.i.d. noise processes assumed to be centred Gaussians with variance σ_η^2 and σ_ν^2 respectively (both unknown to the agent). It is assumed the agent knows the functions f and g and the parameters of the known arm β but not the parameter values of the unknown arm α – this is precisely what the agent must learn. As in [18], the covariate is assumed to be drawn from a known distribution, with unknown parameters (for example a multivariate Gaussian with unknown mean vector μ_X and unknown covariance matrix Σ_X). The objective of the agent is to correctly partition the covariate space between areas where each arm is optimal – the agent hence has to learn this decision boundary accurately and quickly to achieve a high reward.

Exploration in the one-armed bandit with covariates problem involves selecting unknown arm a_1, when arm

[1] As a result, the agent is not facing the full reinforcement learning problem (see [3]).

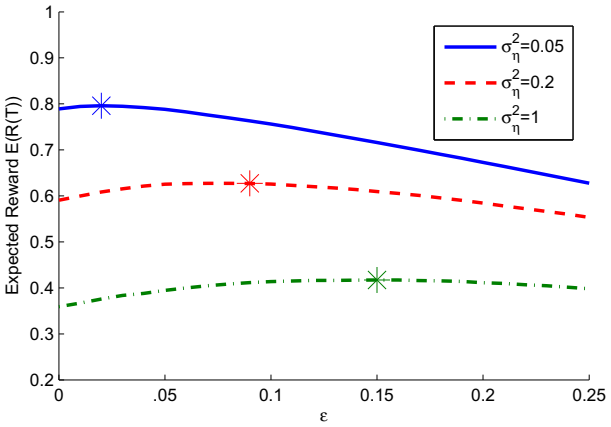

Figure 1. Expected reward of the ϵ-first policy (averaged over 10,000 repeats) for $0 \leq \epsilon \leq 0.25$ in the 10-dimensional setup used in Section IV. We include results for various values of the noise variance σ_η^2. The rewards are normalised such that the highest achievable reward is 1. The game length is 100 iterations.

a_2 is expected to yield a larger reward, given the observed covariate $X(t)$. To illustrate the need to control the amount of exploration performed in this problem, in Fig. 1 we show expected rewards using the ϵ-first policy with the 10-dimensional covariate problem studied in Section IV, for various values of the noise variance σ_η^2. The optimal value of ϵ (denoted by a star) is highly dependent on the level of noise variance, and furthermore the performance of ϵ-first can degrade quickly for badly tuned ϵ, particularly when the amount of noise is low – this motivates the construction of an algorithm to control exploration on-line, which we describe in the next section.

III. ϵ-ADAPT - An Algorithm for Controlling Exploration

In this section, we construct an on-line algorithm, ϵ-ADAPT, that learns to effectively control the amount of exploration in the one-armed bandit with covariates problem. At each iteration, the agent updates its predictions of the unknown parameters of the reward function and the covariate. For example, with a multivariate Gaussian covariate, the agent maintains estimates $\hat{\mu}_X$ and $\hat{\Sigma}_X$, along with $\hat{\alpha}$ and $\hat{\sigma}_\eta^2$, using the sample estimates from the past history of interactions. These statistics indicate the likelihood of each arm being optimal for future covariate values and the amount of uncertainty the agent has regarding this. It is this uncertainty that dictates the likelihood with which the agent will explore or exploit at each iteration.

In an on-line setting, the agent only needs to select an action for the next iteration, and does not have to submit a policy for the remainder of the game. Nevertheless, the agent cannot ignore the possible action choices that follow the current one, which presents the agent with an intractable calculation. To combat this issue, we use the ϵ-first policy as the building block of our algorithm (as motivated in Section II). In particular, we make use of Theorem 1 – a new Theorem regarding the optimality of ϵ-first policies in finite time. $R_{\epsilon F}(A, \epsilon A)$ denotes the expected cumulative reward of an ϵ-first policy for a game of length A that has an initial exploration phase of length ϵA (where action a_1 is always selected).

Theorem 1. *$R_{\epsilon F}(T, c) > R_{\epsilon F}(T, 0)$ and $c \geq 1 \Rightarrow R_{\epsilon F}(T, 1) > R_{\epsilon F}(T, 0)$ (for all $T, c \in \mathbb{Z}^+$).*

This Theorem states that if an ϵ-first policy that explores for 1 or more iteration outperforms a greedy policy then an ϵ-first policy that explores for exactly 1 iteration also outperforms a greedy policy. This property can be clearly seen in Fig. 1.

Proof (sketch): Suppose that $R_{\epsilon F}(T, 1) < R_{\epsilon F}(T, 0)$. It follows that if the game were to be 1 iteration shorter, then $R_{\epsilon F}(T-1, 1) < R_{\epsilon F}(T-1, 0)$ as shorter games require less exploration (to see this consider removing the last action at time T – this will subtract more from the reward of $R_{\epsilon F}(T, 1)$ than $R_{\epsilon F}(T, 0)$). It then follows that $R_{\epsilon F}(T, 2) < R_{\epsilon F}(T, 1)$ (as adding an extra exploration step at the beginning will add more reward to $R_{\epsilon F}(T-1, 0)$). We can then consider a game 2 iterations shorter and continue this inductive process ($c-1$ times) to show $R_{\epsilon F}(T, c) < R_{\epsilon F}(T, 0)$, which completes the proof. □

The significance of this result is that all the agent now needs to compute at time t is which policy is expected to yield a larger reward: $R_{\epsilon F}(T^*, 0)$ or $R_{\epsilon F}(T^*, 1)$, where $T^* = T - t + 1$ (the length of the game remaining). This follows because if any more exploration needs to be performed in a static problem, it should be performed immediately. Therefore, from Theorem 1, the agent needs to explore if and only if an ϵ-first policy with 1 initial exploration step outperforms a greedy policy (or ϵ-first with 0 initial exploration steps). If the agent could calculate this exactly, the optimal on-line policy follows Algorithm 1.

Algorithm 1 Optimal on-line policy

for $t = 1$ to T **do**
  Observe $X(t)$ {Covariate}
  Update unknown parameters of covariate distribution
  if $t \leq D$ or $R_{\epsilon f}(T^*, 1) > R_{\epsilon f}(T^*, 0)$ or $E[r(a_1|X(t), \hat{\alpha}) > E[r(a_2|X(t), \beta)$ **then**
    Select action a_1 and receive reward $r_{a_1}(t)$
    Update $\hat{\alpha}$ and $\hat{\sigma}_\eta^2$ {Only updated when $a_1$selected}
  else
    Select action a_2 and receive reward $r_{a_2}(t)$
  end if
end for
D is the required length of initialisation for sample estimates to exist.

This policy receives exactly the same reward as the optimally tuned off-line ϵ-first policy, but requires knowledge of the unknown parameters. The challenge therefore lies in approximating the unknown expected rewards, $R_{\epsilon f}(T^*, 1)$ and $R_{\epsilon f}(T^*, 0)$. We use sample estimates so that the agent, at time t, can perform a Monte Carlo (MC) simulation of the rest of the game, to see which policy yields the highest expected cumulative reward. This involves generating future covariate values and rewards ($Y(s)$ and $\bar{r}_{a_i}(s|Y(s), \hat{\alpha}, \hat{\sigma}_\eta^2)$ for $s = 1, \ldots, T^*$), and then simulating the game with each policy to see which performs better.

The shortcoming of this approach, however, is that if we simulate the rest of the game from time t using the sample estimates, then the greedy approach will always outperform ϵ-first as the MC estimate of $\hat{\alpha}$ (denoted $\bar{\alpha}$) will already have converged to $\hat{\alpha}$ and exploration is deemed

not to be required. To avoid this, we need to introduce uncertainty in the estimate of $\hat{\alpha}$, which reflects the uncertainty in the true game. So in addition to simulating the rest of the game, we also regenerate covariate values and rewards that were used to estimate $\hat{\alpha}$ prior to time t, such that the MC estimate $\bar{\alpha}$ is perturbed from the true sample estimate. This creates uncertainty in the simulated game that mimics the uncertainty in the true game, and provides a reason to explore. The on-line approximation of $R_{\epsilon f}(T^*, 0)$ (and $R_{\epsilon f}(T^*, 1)$) follows Algorithm 2.

Algorithm 2 On-line MC approximation of $R_{\epsilon f}(T^*, 0)$ (and $R_{\epsilon f}(T^*, 1)$)

Inputs: t (current time-step), T (length of game) C (no. of times a_1 selected prior to time t), sample estimates $(\hat{\alpha}, \hat{\sigma}^2_\eta, \ldots)$
for $s = 1$ to $T^* + C$ **do**
  Generate $Y(s)$ {New Covariate}
  if $s = C + 1$ **then**
    $Y(s) = X(t)$ {True covariate value kept at time t only}
  end if
  if $s \leq C^1$ or $E[\bar{r}(a_1|Y(s), \bar{\alpha}) > E[\bar{r}(a_2|Y(s), \beta)$ **then**
    Select action a_1 and receive reward $\bar{r}_{a_1}(s)$. Update $\bar{\alpha}$
  else
    Select action a_2 and receive reward $\bar{r}_{a_2}(s)$
  end if
end for
$R_{\epsilon f}(T^*, 0) = \sum_{s=C+1}^{T^*+C} \bar{r}_{a_i}(s)$ {MC approximation}
[1] Replace C with $C + 1$ to calculate approximation of $R_{\epsilon f}(T^*, 1)$.

Our algorithm for controlling exploration on-line, ϵ-ADAPT, follows Algorithm 1, with $R_{\epsilon f}(T^*, 0)$ and $R_{\epsilon f}(T^*, 1)$ approximated using Algorithm 2.

Notice that the covariate value at time t is *not* replaced by a new sample in Algorithm 2, but kept at the true observed value – a key component of ϵ-ADAPT. This allows ϵ-ADAPT to decide which regions of the covariate space are worth exploring and which are not. For example, if the expected reward of the unknown arm is only marginally smaller than the known arm (given the covariate value), then the benefits of exploration (through increased learning of parameters α and σ^2_η) can outweigh the costs (the myopic loss of selecting a sub-optimal action). Whereas with other covariate values the short-term costs might exceed the long-term benefits.

This includes a notion of cost-inclusive exploration to ϵ-ADAPT, where at earlier steps the algorithm is willing to forego a lot of reward for 1 exploration step whereas later exploration is only worthwhile if the cost to the reward is negligible. In this sense, ϵ-ADAPT attempts to detect *when* best to explore and not just how much. This is something that ϵ-greedy and ϵ-first policies are not able to do, as they are off-line policies, and this further motivates the use of an on-line exploration algorithm.

ϵ-ADAPT is based on the ϵ-first policy, but does not require any parameters to be set *a priori* that govern the amount of total exploration. In fact, ϵ-ADAPT has no free parameters whatsoever – apart from the parameters used within the estimation module, which are recursively estimated during play, and do not need to be set *a priori*. Occasionally ϵ-ADAPT might undershoot (due to poor sample estimates and/or error from the MC approximation) and start exploiting too early. For these reasons, we continue to decide whether to explore or exploit at *every* iteration until the end of the game. In this way, ϵ-ADAPT is naturally self-correcting and will explore at a later stage (if necessary) to compensate for any lack of earlier exploration.

ϵ-ADAPT is computationally efficient, scaling quadratically in T (as the MC approximation in Algorithm 2 is of maximum length T and is repeated T times). The relationship with the dimensionality of the covariate p depends on the inference procedure used to estimate α. For linear reward models, least squares can be used (order p^3) or recursive least squares (order p^2) if a further saving is required (at a marginal increase to the error of the estimates for low sample-length data). The MC computation given in Algorithm 2 can be repeated several times to smooth the estimates of the two competing policies, but this is not necessary for our algorithm to work. In fact, in the simulations performed in Section IV, the MC estimate was only repeated twice at each iteration, as more repeats had no particular extra benefit.

ϵ-ADAPT can handle non-linear reward functions, provided consistent estimation procedures are available. In addition, our algorithm can deal with non-Gaussian covariates and error terms by either explicitly coding them in (where tractable) or by using bootstrapping techniques.

IV. Numerical Results

In this section, we test ϵ-ADAPT for the one armed bandit with covariates problem. We use linear reward functions with a 5-dimensional covariate (Section IV-A) and 10-dimensional covariate (Section IV-B), to test the ability of ϵ-ADAPT to learn from a high-dimensional covariate. Linear reward functions are considered because, with the appropriate design of covariates, it is often the case that the reward function of an action can be well approximated by a linear function [3], [14], [19]. We assume the observation noise is Gaussian and centred at zero, although we also test the robustness of ϵ-ADAPT to a misspecified noise model (Section IV-C).

Throughout this section, we generate our covariate from a multivariate Gaussian distribution, $X(t) \sim \mathcal{N}(\mu_X, \Sigma_X)$ (see also [14], [18]), as this distribution can accurately model several real-world data sources [20], but again this is not a requirement for our algorithm to work.

A. 5-dimensional Covariate

The reward function of action a_1 and a_2 are given by:

$$r_{a_1}(t) = \sum_{i=1}^{p} \alpha_i X_i(t) + \eta_t, \quad r_{a_2}(t) = \sum_{i=1}^{p} \beta_i X_i(t) + \nu_t,$$

where β is known and α is unknown. p is the dimension of the covariate where $X_1(t) = 1$ (so that α_1 becomes the intercept of the reward plane) and $X_{2,\ldots,p}(t) \sim \mathcal{N}(\mu_X, \Sigma_X)$. We tested ϵ-ADAPT over 10,000 repeated simulations for a game of length 100 with the following parameter values:

$$\alpha = \begin{bmatrix} 0 \\ 0.2 \\ -0.2 \\ 0.1 \\ -0.1 \end{bmatrix} \mu_X = \begin{bmatrix} -0.1 \\ 0.2 \\ 0.3 \\ -0.3 \end{bmatrix} \Sigma_X = \begin{bmatrix} 1 & 0.1 & -0.2 & 0.5 \\ 0.1 & 0.2 & -0.2 & 0 \\ -0.2 & -0.2 & 0.4 & 0.1 \\ 0.5 & 0 & 0.1 & 0.8 \end{bmatrix}$$

The parameter values are selected such that each arm is optimal in approximately 50% of the covariate space. We simulated the problem for various values of the noise variance σ_η^2 and quantify this using the *covariance to noise ratio* [14], CNR$= \frac{\|\Sigma_X\|_1}{\sigma_\eta^2}$, where $\|\Sigma_X\|_1$ is the 1-norm of Σ_X. The larger the CNR, the more informative observations are about α, making the learning problem easier. The only other parameters we could change are the reward coefficients – this has a similar effect to changing the CNR though, in that separating the distance between arms dissipates the effect of noise (and vice-versa). We choose to report CNR values, however, as this allows our results to be commensurate across dimensions.

Table I displays results for ϵ-ADAPT and several ϵ-first policies (where ϵ is set with consideration of the data) for various CNR values. The rewards are normalised between 0 and 1, where 1 is the expected reward to an oracle that knows all parameters and 0 is the expected reward to a random policy. This allows results between experiments to be comparable and regret can be calculated by subtracting the rewards from 1. The standard errors of all average rewards reported in this section are less than 1×10^{-3}.

Lower CNR values correspond to lower rewards for each policy, as the learning problem is more difficult. For each CNR value, however, ϵ-ADAPT performs close to the best performing ϵ-first policy. Moreover, ϵ-ADAPT is the best overall when the rewards are averaged, even though these problems naturally favour exploration rates of 5-10% (which will not always be the case).

Table I
AVERAGE REWARDS WITH A 5-DIMENSIONAL COVARIATE

CNR	$\epsilon = 0$	0.02	0.05	0.1	0.15	0.2	ϵ-ADAPT
100	0.847	**0.859**	0.850	0.816	0.774	0.729	0.857
50	0.773	0.797	**0.803**	0.783	0.749	0.709	0.799
20	0.632	0.668	0.692	**0.695**	0.679	0.653	0.688
10	0.501	0.533	0.565	**0.587**	0.586	0.575	0.573
5	0.381	0.408	0.437	0.467	**0.476**	0.471	0.453
Avg.	0.627	0.653	0.669	0.670	0.653	0.627	**0.674**

Table II compares ϵ-ADAPT with the optimally tuned off-line ϵ-first policy from the same set of experiments. The reward is always within 95% of the off-line optimal, although the performance degrades as the noise increases – this is because the error in the sample estimates are larger, yielding on-line approximations (see Algorithm 2) that are not as accurate. Nevertheless, lower values of the CNR require more exploration from the agent, and ϵ-ADAPT has responded to this by performing more exploration steps on average (last column).

Table II
COMPARISON OF ON-LINE AND OPTIMAL OFF-LINE POLICIES

CNR	Off-line (ϵ-first)		On-line (ϵ-ADAPT)	
	Opt. ϵ	Reward	% Optimal	Avg. exp. steps
100	0.02	0.859	99.8%	3.99
50	0.04	0.804	99.4%	5.19
20	0.08	0.697	98.8%	6.79
10	0.11	0.589	97.4%	7.80
5	0.15	0.476	95.3%	8.61

To gain yet more insight, in Fig. 2 we show the average amount of exploration performed at each time-step t for various CNR values. Notice that the amount of exploration performed is naturally higher for low CNR values and furthermore, decreases as the game is played. This happens automatically as ϵ-ADAPT is more certain of its estimates and the horizon draws closer so exploration becomes more costly. Exploration continues until the end, as covariate values that are expected to yield only marginally suboptimal rewards to arm a_1 can be worth exploring even at a late stage of the game. Fig. 3 displays histograms of the number of exploration steps performed within a game – the spread is due to both the noise in the sample estimates (and subsequent MC approximation) and the circumstances of each game (a favourable start to the game means less exploration needs to be performed later and vice-versa).

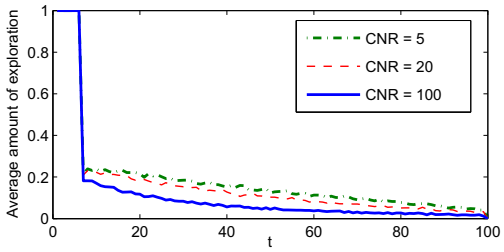

Figure 2. Average amount of exploration performed at time t using ϵ-ADAPT, for various CNR values. The first 6 iterations correspond to the initialisation phase, where ϵ-ADAPT explores to gain sample estimates.

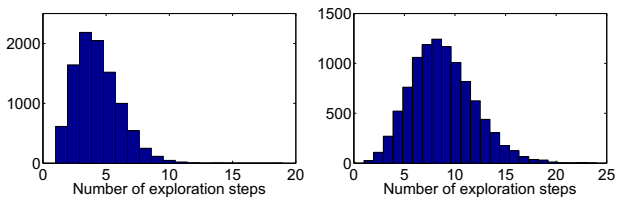

Figure 3. Histograms showing the total number of exploration steps for CNR values of (left) 100 (low noise) and (right) 5 (high noise).

B. 10-dimensional Covariate

We repeat the same experiments for a 10-dimensional covariate with parameters:

$$\alpha = \begin{bmatrix} 0.1 & 0.4 & -0.4 & 0 & 0.4 & -0.2 & 0.4 & -0.3 & 0.1 & -0.1 \end{bmatrix},$$
$$\mu_X = \begin{bmatrix} -0.5 & -0.2 & 0.1 & 0.4 & 0.2 & -0.4 & 0 & 0.3 & -0.3 \end{bmatrix},$$
$$\Sigma_X = \text{diag}(\begin{bmatrix} 0.5 & 0.3 & 0.7 & 0.1 & 0.9 & 0.8 & 0.1 & 1 & 0.1 \end{bmatrix}).$$

We choose a diagonal matrix for Σ_X to maximise the effect of the increased dimensionality on the learning problem, and α is set such that each covariate has a different impact on the reward function. Tables III and IV display the expected rewards for the same on-line and off-line policies, where the magnitude of CNR values has been deliberately reduced by increasing the noise. This is because higher dimension problems require less exploration (see [14] for a detailed explanation). ϵ-ADAPT has again yielded a reward that is within 95% of the optimal and furthermore, has performed best on average against the range of ϵ-first policies considered. The performance of ϵ-ADAPT has not been affected by the increased number of parameters it is required to learn as the algorithm is robust to any p-dimensional covariate.

C. Misspecified Noise Models

In this section, we explore the behaviour of ϵ-ADAPT when assumptions fail (as they would in the real world). In particular, we look at two common departures from

Table III
AVERAGE REWARDS WITH A 10-DIMENSIONAL COVARIATE

CNR	$\epsilon = 0$	0.02	0.05	0.1	0.15	0.2	ϵ-ADAPT
20	**0.833**	**0.833**	0.816	0.777	0.732	0.686	0.832
10	0.787	**0.794**	0.785	0.755	0.715	0.672	0.790
5	0.718	0.731	**0.734**	0.715	0.684	0.647	0.731
2	0.587	0.605	0.617	**0.623**	0.608	0.583	0.617
1	0.470	0.490	0.508	**0.522**	0.520	0.506	0.512
Avg.	0.679	0.691	0.692	0.678	0.652	0.619	**0.696**

Table IV
COMPARISON OF ON-LINE AND OPTIMAL OFF-LINE POLICIES

CNR	Off-line (ϵ-first)		On-line (ϵ-ADAPT)	
	Opt. ϵ	Reward	% Optimal	Avg. exp. steps
20	0.01	0.835	99.6%	3.36
10	0.02	0.794	99.5%	4.31
5	0.04	0.734	99.6%	5.46
2	0.07	0.625	98.8%	6.86
1	0.11	0.522	98.1%	7.74

a Gaussian noise model – asymmetric and heavy-tailed noise distributions. Specifically, we generate the noise process using $t(3)$ (heavy-tailed) and gamma$(2, \theta)$ (skewed) distributions (where the first parameter of the gamma distribution is the shape parameter and the second is the scale). This allows us to check whether ϵ-ADAPT is robust to misspecified noise models. We ran simulations for the 10-dimensional covariate with the alternative noise models, and scaled the noise so that we used the same range of CNR values. The results are presented in Table V. The performance of ϵ-ADAPT has not been affected despite false assumptions regarding the noise model – which is a particularly desirable type of robustness if these methods were to be applied to real-world problems.

Table V
PERFORMANCE OF ϵ-ADAPT WITH MISSPECIFIED NOISE MODELS

CNR	t(3) noise			gamma(2,θ) noise		
	ϵ-first		ϵ-ADAPT	ϵ-first		ϵ-ADAPT
	Opt. ϵ	Reward	% Opt.	Opt. ϵ	Reward	% Opt.
20	0.01	0.841	99.7%	0.01	0.836	99.5%
10	0.01	0.805	99.7%	0.02	0.799	99.3%
5	0.03	0.751	99.8%	0.03	0.743	99.4%
2	0.06	0.649	99.6%	0.07	0.634	99.0%
1	0.09	0.554	98.9%	0.10	0.532	98.7%

V. CONCLUSIONS AND FUTURE WORK

In this paper we have presented ϵ-ADAPT, the first algorithm that controls exploration on-line in bandit problems. ϵ-ADAPT decides at each iteration whether to explore or exploit and this is driven by the amount of uncertainty in the system. We have presented numerical results for the one-armed bandit with covariates problem, and demonstrated that ϵ-ADAPT is competitive with the optimal ϵ-first policy, which requires the exploration parameter to be set *a priori*. The amount of exploration performed by ϵ-ADAPT naturally decreases over time (in a static system), as the agent becomes more certain of the game it is playing and the horizon draws closer. We have shown that ϵ-ADAPT is robust to high-dimensional and misspecified models and explained how it can also work with non-linear and non-parametric reward functions.

Future work includes investigating bandit problems with multiple arms – where we extend ϵ-ADAPT such that the agent sequentially calculates the value of exploitation and also the value of exploration for each suboptimal action individually. The action with the highest value is then selected – in a similar vein to the *Gittins indices* [12], constructed for a different version of the bandit problem. Preliminary evidence suggests that this method is effective in adapting exploration on-line. Also of interest is extending ϵ-ADAPT to dynamic problems where parameters suddenly change or drift over time [18]. ϵ-ADAPT is already naturally suited to such a problem, as dynamics in the system will increase the uncertainty of the running estimates, which will subsequently increase exploration.

ACKNOWLEDGEMENTS

This research was undertaken as part of the ALADDIN (Autonomous Learning Agents for Decentralised Data and Information Networks) project and is jointly funded by a BAE Systems and EPSRC strategic partnership (EP/C548051/1).

REFERENCES

[1] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, pp. 527–35, 1952.
[2] D. Berry and B. Fristedt, *Bandit problems*. Chapman and Hall London, 1985.
[3] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.
[4] M. Rothschild, "A two-armed bandit theory of market pricing," *Journal of Economic Theory*, vol. 9, no. 2, pp. 185–202, 1974.
[5] J. Holland, *Adaptation in natural and artificial systems*. MIT press Cambridge, MA, 1992.
[6] A. Blum, V. Kumar, A. Rudra, and F. Wu, "Online learning in online auctions," *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 202–204, 2003.
[7] V. Krishnamurthy and R. Evans, "Hidden Markov model multiarm bandits: a methodology for beamscheduling in multitarget tracking," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 2893–2908, 2001.
[8] R. Kleinberg, A. Slivkins, and E. Upfal, "Multi-armed bandits in metric spaces," *Proceedings of the 40th annual ACM symposium on Theory of computing*, pp. 681–690, 2008.
[9] M. Woodroofe, "A one-armed bandit problem with a concomitant variable," *Journal of the American Statistical Association*, vol. 74, pp. 799–806, 1979.
[10] J. Vermorel and M. Mohri, "Multi-armed bandit algorithms and empirical evaluation," *Lecture notes in computer science*, vol. 3720, pp. 437–448, 2005.
[11] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235–256, 2002.
[12] J. Gittins, *Multi-armed bandit allocation indices*. Wiley, New York, 1989.
[13] J. Sarkar, "One-armed bandit problems with covariates," *The Annals of Statistics*, vol. 19, no. 4, pp. 1978–2002, 1991.
[14] N. Pavlidis, D. Tasoulis, and D. Hand, "Simulation studies of multi-armed bandits with covariates," *Proceedings of the 10th International Conference on Computer Modeling and Simulation*, pp. 493–498, 2008.
[15] H. Chernoff, "Sequential models for clinical trials," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 4, pp. 805–812, 1967.
[16] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," *Advances in Neural Information Processing Systems*, 2007.
[17] C. Wang, S. Kulkarni, and H. Poor, "Bandit problems with side observations," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 338–355, 2005.
[18] N. Pavlidis, D. Tasoulis, N. Adams, and D. Hand, "Dynamic multi-armed bandit with covariates," *Proceedings of the 18th European Conference on Artificial Intelligence*, pp. 777–779, 2008.
[19] N. Abe, A. Biermann, and P. Long, "Reinforcement learning with immediate rewards and linear hypotheses," *Algorithmica*, vol. 37, no. 4, pp. 263–293, 2003.
[20] D. Cox and N. Small, "Testing multivariate normality," *Biometrika*, vol. 65, no. 2, pp. 263–272, 1978.