Trading off Rewards and Errors in Multi-Armed Bandits

Akram Erraqabi INRIA SequeL
Alessandro Lazaric INRIA SequeL
Michal Valko INRIA SequeL
Emma Brunskill CMU
Yun-En Liu EnLearn

Abstract

In multi-armed bandits, the most common objective is the maximization of the cumulative reward. Alternative settings include active exploration, where a learner tries to gain accurate estimates of the rewards of all arms. While these objectives are contrasting, in many scenarios it is desirable to trade off rewards and errors. For instance, in educational games the designer wants to gather generalizable knowledge about the behavior of the students and teaching strategies (small *estimation errors*) but, at the same time, the system needs to avoid giving a bad experience to the players, who may leave the system permanently (large *reward*). In this paper, we formalize this tradeoff and introduce the ForcingBalance algorithm whose performance is provably close to the best possible tradeoff strategy. Finally, we demonstrate on real-world educational data that ForcingBalance returns *useful* information about the arms without compromising the overall reward.

1 Introduction

We consider sequential, interactive systems when a learner aims at optimizing an objective function whose parameters are initially unknown and need to be estimated over time. We take the multi-armed bandit (MAB) framework where the learner has access to a finite set of distributions (*arms*), each one characterized by an expected value (*reward*). The learner does not know the distributions beforehand and it can only obtain a random sample by selecting an arm. The most common objective in MAB is to minimize the regret, i.e., the difference between the reward of the arm with the highest mean and the reward of the arms pulled by the learner. Since the arm means are unknown, this requires balancing *exploration* of the arms and *exploitation* of the mean estimates. An alternative setting is *pure exploration*, where the learner's performance is only evaluated upon the termination of the process, and its learning performance is allowed to be arbitrarily bad in terms of rewards accumulated over time. In *best-arm identification* [Even-Dar et al., 2006, Audibert et al., 2010], the learner selects arms to find the optimal arm either with very high probability or in a short number of steps. In *active exploration* [Antos et al., 2010, Carpentier et al., 2011], the objective is to estimate the value of all arms as accurately as possible. This setting, which is related to active learning and experimental optimal design, is particularly relevant whenever accurate predictions of the arms' value is needed to support decisions at a later time.

The previous objectives have been studied separately. However, they do not address the increasingly-prevalent situation where users participate in research studies (e.g., for education or health) that are designed to collect reliable data and compute accurate estimates of the performance of the available options. Here, the subjects/users themselves rarely care about the underlying research questions but wish to gain their own benefit, such as students seeking to learn new material, or patients seeking to find improvement for their condition. In order to serve these individuals and gather generalizable knowledge at the same time, we formalize this situation as a multi-objective bandit problem, where a designer seeks to trade off cumulative regret minimization (providing good direct reward for participants), with informing scientific knowledge about the strengths and limitations of the various conditions (active exploration to estimate all arm means). This tradeoff is especially needed in high-stakes domains, such as medicine or education, or when running experiments online, where poor experience may lead to users leaving the system permanently. A similar tradeoff happens in A/B testing. Here, the designer may want to retain the ability to set a desired level of accuracy in estimating the value of different alternatives (e.g., to justify decisions that are to be taken *posterior* to the experiment) while still maximizing the reward.

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54. Copyright 2017 by the author(s).

A natural initial question is whether these two different objectives, reward maximization and accurate arm estimation, or other alternative objectives, like best arm identification, are mutually compatible: Can one always recover the best of all objectives? Unfortunately, in general, the answer is negative. Bubeck et al. [2009] have already shown that any algorithm with sub-linear regret cannot be optimal for identifying the best arm. Though it may not be possible to be simultaneously optimal for both active exploration and reward maximization, we wish to carefully trade off between these two objectives. How to properly balance multiple objectives in MAB is a mostly unexplored question. Bui et al. [2011] introduce the *committing bandits*, where a given horizon is divided into an experimentation phase when the learner is free to explore all the arms but still pays a cost, and a commit phase when the learner must choose one single arm that will be pulled until the end of the horizon. Lattimore [2015] analyzes the problem where the learner wants to minimize the regret simultaneously w.r.t. two special arms. He shows that if the regret w.r.t. one arm is bounded by a small quantity B, then the regret w.r.t. the other arm scales at least as $1/B$, which reveals the difficulty of balancing two objectives at the same time. Drugan and Nowé [2013] formalize the multi-objective bandit problem where each arm is characterized by multiple values and the learner should maximize a multi-objective function constructed over the values of each arm. They derive variations of UCB to minimize the regret w.r.t. the full Pareto frontier obtained for different multi-objective functions. Finally, Sani et al. [2012] study strategies having a small regret versus the arm with the best mean-variance tradeoff. In this case, they show that it is not always possible to achieve a small regret w.r.t. the arm with the best mean-variance.

In this paper, we study the tradeoff between cumulative reward and accuracy of estimation of the arms' values (i.e., reward maximization and active exploration), which was first introduced by Liu et al. [2014]. Their work presented a heuristic algorithm for balancing this tradeoff and promising empirical results on an education simulation. In the present paper, we take a more rigorous approach and make several new contributions. **1)** We propose and justify a new objective function for the integration of rewards and estimation errors (Sect. 2), that provides a simple way for a designer to weigh directly between them. **2)** We introduce the ForcingBalance algorithm that optimizes the objective function when the arm distributions are unknown (Sect. 3). Despite its simplicity, we prove that ForcingBalance incurs a regret that asymptotically matches the minimax rate for cumulative regret minimization and the performance of active exploration algorithms (Sect. 4). This is very encouraging, as it shows that balancing a tradeoff between rewards and errors is not fundamentally more difficult than either of these separate objectives. Interestingly, we also show that a simple extension of UCB is not sufficient to achieve good performance. **3)** Our analysis requires only requires strong convexity and smoothness of the objective function and therefore our algorithm and the proof technique can be easily extended. **4)** We provide empirical simulations on both synthetic and educational data from Liu et al. [2014] that support our analysis (Sect. 5).

2 Balancing Rewards and Errors

We consider a MAB of K arms with distributions $\{\nu_i\}_{i=1}^K$, each characterized by mean μ_i and variance σ_i^2. For technical convenience, we consider distributions with bounded support in $[0,1]$. All the following results extend to the general case of sub-Gaussian distributions (used in the experiments). We denote the s-th i.i.d. sample drawn from ν_i by $X_{i,s}$ and we define $[K]=\{1,\ldots,K\}$. As discussed in the introduction, we study the combination of two objectives: reward maximization and estimation error minimization. Given a fixed sequence of n arms $\mathcal{I}_n=(I_1,I_2,..,I_n)$, where $I_t\in[K]$ is the arm pulled at time t, the average reward is defined as

$$\rho(\mathcal{I}_n)=\mathbb{E}\left[\frac{1}{n}\sum_{t=1}^{n}X_{I_t,T_{I_t,t}}\right]=\frac{1}{n}\sum_{i=1}^{K}T_{i,n}\mu_i, \quad (1)$$

where $T_{i,t}=\sum_{s=1}^{t-1}\mathbb{I}\{I_s=i\}$ is the number of times arm i is selected up to step $t-1$. The sequence maximizing ρ simply selects the arm with largest mean for all n steps. On the other hand, the estimation error is measured as

$$\varepsilon(\mathcal{I}_n)=\frac{1}{K}\sum_{i=1}^{K}\sqrt{n\mathbb{E}\Big[\big(\widehat{\mu}_{i,n}-\mu_i\big)^2\Big]}=\frac{1}{K}\sum_{i=1}^{K}\sqrt{\frac{n\sigma_i^2}{T_{i,n}}}, \quad (2)$$

where $\widehat{\mu}_{i,n}$ is the empirical average of the $T_{i,n}$ samples. Similar functions were used by Carpentier et al. [2011, 2015]. Notice that (2) is multiplying the root mean-square error by $\sqrt{n}$. This is to allow the user to specify a direct tradeoff between (1) and (2) regardless on how their average magnitude varies as a function of n.[1] Optimizing ε requires selecting all the arms with a frequency proportional to their standard deviations. More precisely, each arm should be pulled proportionally to $\sigma_i^{2/3}$. We define the tradeoff objective function balancing the two functions above as a convex combination,

[1] This choice also "equalizes" the standard regret bounds for the two separate objectives, so that the minimax regret in terms of ρ and the known upper-bounds on the regret w.r.t. ε are both $\widetilde{O}(1/\sqrt{n})$.

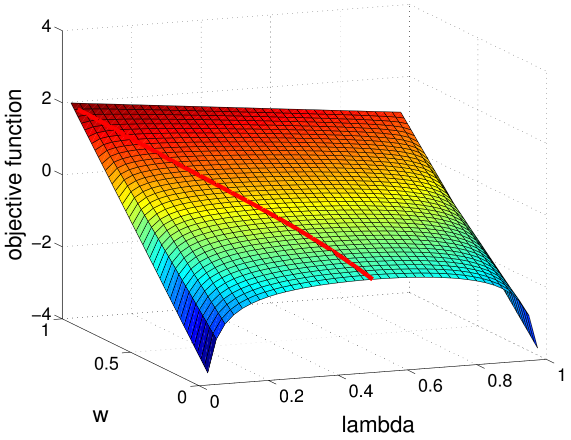

Figure 1: Function f_w and optimal solution $\boldsymbol{\lambda}^*$ for different values of w (*red* line) for a MAB with $K=2$, $\mu_1=1$, $\mu_2=2$, $\sigma_1^2=\sigma_2^2=1$. For small w, the problem reduces to optimizing the average estimation error. Since the arms have the same variance, $\boldsymbol{\lambda}^*$ is an even allocation over the two arms. As w increases, the ρ component in f_w becomes more relevant and the optimal allocation selects arm 2 more often, until $w=1$ when all the resources are allocated to arm 2.

$$\begin{aligned} f_w(\mathcal{I}_n;\{\nu_i\}_i) &= w\rho(\mathcal{I}_n) - (1-w)\varepsilon(\mathcal{I}_n) \\ &= w\sum_{i=1}^K \frac{T_{i,n}}{n}\mu_i - \frac{(1-w)}{K}\sum_{i=1}^K \frac{\sigma_i}{\sqrt{T_{i,n}/n}}, \qquad (3) \end{aligned}$$

where $w \in [0,1]$ is a weight parameter and the objective is to find the sequence of pulls $\mathcal{I}_n$ which maximizes f_w. For $w=1$, we recover the reward maximization problem, while for $w=0$, the problem reduces to minimizing the average estimation error. In the rest of the paper, we are interested in the case $w \in (0,1)$ since the extreme cases have already been studied. Using *root* mean square error for $\varepsilon(\mathcal{I}_n)$ gives f_w the *scale-invariant* property: Rescaling the distributions equally impacts ρ and ε. Furthermore, f_w can be equivalently obtained as a Lagrangian relaxation of a constrained optimization problem where we intend to maximize the reward subject to a desired level of estimation accuracy. In this case, the parameter w is directly related to the value of the Lagrange multiplier. Liu et al. [2014] proposed a similar tradeoff function where the estimation error is measured by Hoeffding confidence intervals, which disregard the variance of the arms and only depend on the number of pulls. In addition, in their objective, the optimal allocation radically changes with the horizon n, where a short horizon forces the learner to be more explorative, while longer horizons allow the learner to be more greedy in accumulating rewards. Overall, their tradeoff reduces to a mixture between a completely uniform allocation (that minimizes the confidence intervals) and a UCB strategy that maximizes the cumulative reward. While their algorithm demonstrated encouraging empirical performance, no formal analysis was provided. In contrast, f_w is *stable* over time and it allows us to compare the performance of a learning algorithm to a static optimal allocation. We later show that f_w also enjoys properties such as smoothness and strong concavity that are particularly convenient for the analysis. Besides the mathematical advantages, we notice that without normalizing ε by n, as w tends to 0, we would never be able to recover the optimal strategy for error minimization, since $\rho(\mathcal{I}_n)$ would always dominate f_w, thus making the impact of tuning w difficult to interpret.

Given a horizon n, finding the optimal $\mathcal{I}_n$ requires solving a difficult discrete optimization problem, thus we study its continuous relaxation,[2]

$$f_w(\boldsymbol{\lambda};\{\nu_i\}_i) = w\sum_{i=1}^K \lambda_i\mu_i - \frac{(1-w)}{K}\sum_{i=1}^K \frac{\sigma_i}{\sqrt{\lambda_i}}, \qquad (4)$$

where $\boldsymbol{\lambda} \in \mathcal{D}_K$ belongs to the K-dimensional simplex such that $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$. As a result, $\boldsymbol{\lambda}$ defines an allocation of arms and $f_w(\boldsymbol{\lambda};\{\nu_i\}_i)$ is its asymptotic performance if arms are repeatedly chosen according to $\boldsymbol{\lambda}$. We define the optimal allocation and its performance as $\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}\in\mathcal{D}_K} f_w(\boldsymbol{\lambda};\{\nu_i\}_i)$ and $f^* = f_w(\boldsymbol{\lambda}^*;\{\nu_i\}_i)$ respectively. Since f_w is concave and $\mathcal{D}_K$ is convex, $\boldsymbol{\lambda}^*$ always exists and it is unique whenever $w<1$ (and there is at least a non-zero variance) or when the largest mean is distinct from the second largest mean. Although a closed-form solution cannot be computed in general, intuitively $\boldsymbol{\lambda}^*$ favors arms with large means and large variance since allocating a large portion of the resources to them contribute to minimizing f_w by increasing the reward ρ and reducing the error ε. The parameter w defines the sensitivity of $\boldsymbol{\lambda}^*$ to the arm parameters, such that for large w, $\boldsymbol{\lambda}^*$ tends to concentrate on the arm with largest mean, while for small w, $\boldsymbol{\lambda}^*$ allocates arms proportionally to their standard deviations. Fig. 1, Sect. 5.1, and App. C provide additional examples illustrating the sensitivity of $\boldsymbol{\lambda}^*$ to the parameters in f_w. Let $\mathcal{I}_n^*$ be the optimal discrete solution to Eq. 3. Then, we show that the difference between the two solutions rapidly shrinks to 0 with n. In fact,[3] for any arm i, $|T^*_{i,n}/n - \lambda^*_i| \leq 1/n$ and according to Lem. 4 (stated later), this guarantees that the value of λ^* (f^*) differs from the optimum of $f_w(\mathcal{I}_n)$ by $1/n^2$.

[2] A more accurate definition of f_w over the simplex requires completing it with $f_w(\boldsymbol{\lambda}) = -\infty$ whenever there exists a component $\lambda_i = 0$ linked to a non-zero variance σ_i^2.

[3] Consider a real number $r \in [0,1]$ and R_n any rounding of rn (e.g., $R_n = \lfloor rn \rfloor$), then $|R_n - rn| \leq 1$. If we use $\widehat{r}_n = R_n/n$ as fractional approximation of r with resolution n, then we obtain that $|\widehat{r}_n - r| \leq 1/n$.

In the following, we consider the restricted simplex $\overline{\mathcal{D}}_K = \{\lambda_i \geq \lambda_{\min}, \sum_i \lambda_i = 1\}$ with $\lambda_{\min} > 0$ on which f_w is always bounded and it can be characterized by the following lemma.

Lemma 1. *Let* $\sigma_{\max} = \max_i \sigma_i$ *and* $\sigma_{\min} = \min_i \sigma_i > 0$ *be the largest and smallest standard deviations, the function* $f_w(\boldsymbol{\lambda}; \{\nu_i\})$ *is* α*-strongly concave everywhere in* $\mathcal{D}_K$ *with* $\alpha = \frac{3(1-w)\sigma_{\min}}{4K}$ *and it is* β*-smooth in* $\overline{\mathcal{D}}_K$ *with* $\beta = \frac{3(1-w)\sigma_{\max}}{4K\lambda_{\min}^{5/2}}$.

Finally, we define the performance of a learning algorithm. Let $\widetilde{\boldsymbol{\lambda}}_n$ be the empirical frequency of pulls, i.e., $\widetilde{\lambda}_{i,n} = T_{i,n}/n$. We define its regret w.r.t. the value of the optimal allocation as $R_n(\widetilde{\boldsymbol{\lambda}}_n) = f^* - f_w(\widetilde{\boldsymbol{\lambda}}_n; \{\nu_i\}_i)$. The previous equation defines the pseudo-regret of a strategy $\widetilde{\boldsymbol{\lambda}}_n$, since in Eq. 2 the second equality is true for fixed allocations. This is similar to the definition of Carpentier et al. [2015], where the difference between *true* and pseudo-regret is discussed in detail.

3 The ForcingBalance Algorithm

Why naïve UCB fails. One of the most successful approaches to bandits is the optimism-in-face-of-uncertainty, where we construct confidence bounds for the parameters and select the arm maximizing an upper-bound on the objective function. This approach was successfully applied in both regret minimization (see e.g., Auer et al. [2002]) and active exploration (see e.g., Carpentier et al. [2011]). As such, a first natural approach to our problem is to construct an upper-bound on f_w as (see Prop. 1 for the definition of the confidence bounds)

$$f_w^{UB}(\boldsymbol{\lambda}; \{\widehat{\nu}_{i,n}\}) = w\sum_{i=1}^{K} \lambda_i \bigg(\widehat{\mu}_{i,n} + \sqrt{\frac{\log(1/\delta_n)}{2T_{i,n}}}\bigg) \quad (5)$$
$$-(1-w)\sum_{i=1}^{K} \frac{1}{\sqrt{\lambda_i}} \bigg(\widehat{\sigma}_{i,n} - \sqrt{\frac{2\log(2/\delta_n)}{T_{i,n}}}\bigg).$$

At each step n, we compute the allocation $\widehat{\boldsymbol{\lambda}}_{i,n}^{UB}$ maximizing f_w^{UB} and select arms accordingly (e.g., by pulling an arm at random from $\widehat{\boldsymbol{\lambda}}_{i,n}^{UB}$). Although the confidence bounds guarantee that for any $\boldsymbol{\lambda}$, $f_w^{UB}(\boldsymbol{\lambda}; \{\widehat{\nu}_{i,n}\}) \geq f_w(\boldsymbol{\lambda}; \{\nu_i\})$ w.h.p., this approach is intrinsically flawed and it would perform poorly. While for large values of w, the algorithm reduces to UCB, for small values of w, the algorithm tends to allocate arms to balance the estimation errors on the basis of *lower-bounds* on the variances and thus arms with small lower-bounds are selected less. Since small lower-bounds may be associated with arms with large confidence intervals, and thus poorly estimated variances, this behavior would prevent the algorithm from correcting its estimates and improving its performance over time (see App. C for additional discussion and empirical simulations). Constructing lower-bounds on f_w suffers from the same issue. This suggests that a *straightforward* (naïve) application of a UCB-like strategy fails in this context. As a result, we take a different approach and propose a *forcing* algorithm inspired by the GAFS-MAX algorithm introduced by Antos et al. [2010] for active exploration.[4]

```
Input: forcing parameter $\eta$, weight $w$
for $t = 1, \dots, n$ do
    $U_t = \arg\min T_{i,t}$
    if $T_{U_t,t} < \eta\sqrt{t}$ then
        Select arm $I_t = U_t$ (forcing)
    else
        Compute optimal estimated allocation
            $\widehat{\boldsymbol{\lambda}}_t = \arg\max_{\boldsymbol{\lambda} \in \overline{\mathcal{D}}_K} f_w(\boldsymbol{\lambda}; \{\widehat{\nu}_{i,t}\}_i)$
        Select arm (tracking)
            $I_t = \arg\max_{i=1,\dots,K} \widehat{\lambda}_{i,t} - \widetilde{\lambda}_{i,t}$
    end if
    Pull arm $I_t$, observe $X_{I_t,t}$, update $\widehat{\nu}_{I_t}$.
end for
```

Figure 2: The ForcingBalance algorithm.

Forced sampling. The ForcingBalance algorithm is illustrated in Fig. 2. It receives as input an exploration parameter $\eta > 0$ and the restricted simplex $\overline{\mathcal{D}}_K$ defined by $\lambda_{\min}$. At each step t, the algorithm first checks the number of pulls of each arm and selects any arm with less than $\eta\sqrt{t}$ samples. If all arms have been sufficiently pulled, the allocation $\widehat{\boldsymbol{\lambda}}_t$ is computed using the empirical estimates of the arms' means and variances $\widehat{\mu}_{i,t} = \frac{1}{T_{i,t}} \sum_{s=1}^{T_{i,t}} X_{i,s}$ and $\widehat{\sigma}_{i,n}^2 = \frac{1}{2T_{i,t}(T_{i,t}-1)} \sum_{s,s'=1}^{T_{i,t}} \big(X_{i,s} - X_{i,s'}\big)^2$. Notice that the optimization is done over the restricted simplex $\overline{\mathcal{D}}_K$ and $\widehat{\boldsymbol{\lambda}}_t$ can be computed efficiently. Once the allocation $\widehat{\boldsymbol{\lambda}}_t$ is computed, an arm is selected. A straightforward option is either to directly implement the optimal estimated allocation by pulling an arm drawn at random from it or allocate the arms proportionally to $\widehat{\boldsymbol{\lambda}}_t$ over a short phase. Both solutions may not be effective since the final performance is evaluated according to the *actual* allocation realized over all n steps (i.e., $\widetilde{\lambda}_{i,n} = T_{i,n}/n$) and not $\widehat{\boldsymbol{\lambda}}_n$. Consequently, even when $\widehat{\boldsymbol{\lambda}}_n$ is an accurate approximation of $\boldsymbol{\lambda}^*$, the regret may not be small.[5] ForcingBalance explicitly

[4] Variations on the *forcing* or *forced sampling* approach have been used in many settings including standard bandits [Yakowitz and Lai, 1995, Szepesvári, 2008], linear bandits [Goldenshluger and Zeevi, 2013], contextual bandits [Langford and Zhang, 2007], and experimental optimal design [Wiens and Li, 2014].

[5] Consider the case of 3 arms, where after t steps, the

tracks the allocation $\widehat{\boldsymbol{\lambda}}_n$ by selecting the arm I_t that is under-pulled the most so far. This tracking step allows us to force $\widetilde{\boldsymbol{\lambda}}_n$ to stay close to $\widehat{\boldsymbol{\lambda}}_n$ (and its performance) at each step. The tracking step is slightly different from GAFS-MAX, which selects the arms with largest ratio between $\widehat{\lambda}_{i,n}$ and $\widetilde{\lambda}_{i,t}$. We show in the analysis that the proposed tracking rule is more efficient.

The parameter η defines the amount of exploration forced by the algorithm. A large η forces all arms to be pulled many times. While this guarantees accurate estimates $\widehat{\mu}_{i,t}$ and $\widehat{\sigma}^2_{i,n}$ and an optimal estimated allocation $\widehat{\boldsymbol{\lambda}}_t$ that rapidly converges to $\boldsymbol{\lambda}^*$, the algorithm would perform the tracking step very rarely and thus $\widetilde{\boldsymbol{\lambda}}_t$ would not track $\widehat{\boldsymbol{\lambda}}_t$ fast enough. In the next section, we show that any value of η in a wide range (e.g., $\eta = 1$) guarantees a small regret. The other parameter is $\lambda_{\min}$ which defines a restriction on the set of allocations that can be learned. From an algorithmic point of view, $\lambda_{\min} = 0$ is a viable choice since f_w is strongly concave and it always admits at least one solution in $\mathcal{D}_K$ (the full simplex). Nonetheless, we show next that $\lambda_{\min}$ needs to be strictly positive to guarantee uniform convergence of f_w for true and estimated parameters, which is a critical property to ensure regret bounds.

4 Theoretical Guarantees

In this section, we derive an upper-bound on the regret of ForcingBalance with explicit dependence on its parameters and the characteristics of f_w. We start with high-probability confidence intervals for the mean and the standard deviation (see Thm. 10 of Maurer and Pontil [2009]).

Proposition 1. *Fix $\delta \in (0,1)$. For any $n > 0$ and any arm $i \in [K]$, $\big|\widehat{\mu}_{i,n} - \mu_i\big| \le \sqrt{\frac{\log(1/\delta_n)}{2T_{i,n}}}$, $\big|\widehat{\sigma}_{i,n} - \sigma_i\big| \le \sqrt{\frac{2\log(2/\delta_n)}{T_{i,n}}}$, w.p. $1-\delta$, where $\delta_n = \delta/(4Kn(n+1))$.*

The accuracy of the estimates translates into the difference in estimated and true function f (we drop the dependence on w for readability).

Lemma 2. *Let $\widehat{\nu}_i$ be an empirical distribution characterized by mean $\widehat{\mu}_i$ and variance $\widehat{\sigma}_i$ such that $|\widehat{\mu}_i - \mu_i| \le \varepsilon_i^{\mu}$ and $|\widehat{\sigma}_i - \sigma_i| \le \varepsilon_i^{\sigma}$, then for any fixed $\boldsymbol{\lambda} \in \mathcal{D}_K$ we have $\big|f(\boldsymbol{\lambda}; \{\nu_i\}) - f(\boldsymbol{\lambda}; \{\widehat{\nu}_i\})\big| \le w \max_i \varepsilon_i^{\mu} + \frac{1-w}{\min_i \sqrt{\lambda_i}} \max_i \varepsilon_i^{\sigma}$.*

This lemma shows that the accuracy in estimating f is affected by the largest error in estimating the mean or the variance of any arm. This is due to the fact that $\boldsymbol{\lambda}$ may give a high weight to a poorly estimated arm, i.e., λ_i may be large for large ε_i. As a result, if ε_i^{μ} and ε_i^{σ} are defined as in Prop. 1, the lemma requires that all arms are pulled often enough to guarantee an accurate estimation of f. Furthermore, the upper-bound scales inversely with the minimum proportion $\min_i \lambda_i$. This shows the need of restricting the possible $\boldsymbol{\lambda}$s to allocations with a non-zero lower-bound to $\min_i \lambda_i$, which is guaranteed by the use of the restricted simplex $\overline{\mathcal{D}}_K$ in the algorithm. Finally, notice that here we consider a fixed allocation $\boldsymbol{\lambda}$, while later we need to deal with a (possibly) random choice of $\boldsymbol{\lambda}$, which requires a union bound over a cover of $\overline{\mathcal{D}}_K$ (see Cor. 1). Next two lemmas show how the difference in performance translates in the difference of allocations and vice versa.

Lemma 3. *If an allocation $\boldsymbol{\lambda} \in \mathcal{D}_K$ is such that $\big|f^* - f(\boldsymbol{\lambda}; \{\nu_i\})\big| \le \varepsilon^f$, then for any arm $i \in [K]$, $|\lambda_i - \lambda_i^*| \le \sqrt{\frac{2K}{\alpha}}\sqrt{\varepsilon^f}$, where α is the strong-concavity parameter of f_w (Lem. 1).*

Lemma 4. *The performance of an allocation $\boldsymbol{\lambda} \in \overline{\mathcal{D}}_K$ compared to the optimal allocation $\boldsymbol{\lambda}^*$ is such that $f(\boldsymbol{\lambda}^*; \{\nu_i\}) - f(\boldsymbol{\lambda}; \{\nu_i\}) \le \frac{3\beta}{2}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^*\|^2$.*

In both cases, the bounds depend on the shape of f through the parameters of strong concavity α and smoothness β, which in turn depends on the constrained simplex $\overline{\mathcal{D}}_K$ and the choice of $\lambda_{\min}$. Before stating the regret bound, we need to introduce an assumption on $\boldsymbol{\lambda}^*$.

Assumption 1. *Let $\lambda^*_{\min} = \min_i \lambda^*_i$ be the smallest proportion over the arms in the optimal allocation and let $\overline{\mathcal{D}}_K$ the restricted simplex used in the algorithm. We assume that the weight parameter w and the distributions $\{\nu_i\}_i$ are such that $\lambda^*_{\min} \ge \lambda_{\min}$, that is $\boldsymbol{\lambda}^* \in \overline{\mathcal{D}}_K$.*

Notice that whenever all arms have non-zero variance and $w < 1$, $\lambda^*_{\min} > 0$ and there always exists a non-zero $\lambda_{\min}$ (and thus a set $\overline{\mathcal{D}}_K$) for which the assumption can be verified. In general, the larger and more similar the variances and the smaller w, the bigger $\lambda^*_{\min}$ and less restrictive the assumption. The choice of $\lambda_{\min}$ also affects the final regret bound.

Theorem 1. *We consider a MAB with $K \ge 2$ arms with mean $\{\mu_i\}$ and variance $\{\sigma_i^2\}$. Under Asm. 1, ForcingBalance with a parameter $\eta \le 21$ and a simplex $\overline{\mathcal{D}}_K$ restricted to $\lambda_{\min}$ suffers a regret*

$$R_n(\widetilde{\boldsymbol{\lambda}}) \le \begin{cases} 1 & \text{if } n \le n_0 \\ 43K^{5/2}\dfrac{\beta}{\alpha}\sqrt{\dfrac{\log(2/\delta_n)}{\eta\lambda_{\min}}}\,n^{-1/4} & \text{if } n_0 < n \le n_2 \\ 153K^{5/2}\dfrac{\beta}{\alpha}\sqrt{\dfrac{\log(2/\delta_n)}{\lambda_{\min}\lambda^*_{\min}}}\,n^{-1/2} & \text{if } n > n_2, \end{cases}$$

empirical allocation $\widetilde{\boldsymbol{\lambda}}_t$ is $(0.5, 0.1, 0.4)$ and the estimated allocation $\widehat{\boldsymbol{\lambda}}_t$ is $(0.5, 0.4, 0.1)$. In the following steps, the most effective way to reduce the regret is not to use $\widehat{\boldsymbol{\lambda}}_t$, but to pull arm 2 more than 40%, in order to close the gap between $\widetilde{\boldsymbol{\lambda}}_t$ and $\widehat{\boldsymbol{\lambda}}_t$ as fast as possible.

with probability $1-\delta$ *(where* $\delta_n = \delta/(4Kn(n+1))$*) and*

$$n_0 = K(K\eta^2 + \eta\sqrt{K} + 1),$$
$$n_2 = \frac{C}{(\lambda^*_{\min})^8}\frac{K^{10}}{\alpha^4}\frac{\log^2(1/\delta_n)}{\lambda^2_{\min}},$$

where C *is a numerical constant.*

Remark 1 (dependence on n). The previous bound reveals the existence of three phases. For $n \leq n_0$, we are in a fully explorative phase where the pulls are always triggered by the forcing condition, the allocation $\widetilde{\boldsymbol{\lambda}}_n$ is uniform over arms; and it can be arbitrarily bad w.r.t. $\boldsymbol{\lambda}^*$. In the second phase, the algorithm interleaves forcing and tracking but the estimates $\{\widehat{\nu}_{i,n}\}$ are not accurate enough to guarantee that $\widehat{\boldsymbol{\lambda}}_n$ performs well. In particular, we can only guarantee that all arms are selected $\eta\sqrt{n}$, which implies the regret decreases very slowly as $\widetilde{O}(n^{-1/4})$. Fortunately, as the estimates become more accurate, $\widehat{\boldsymbol{\lambda}}_n$ approaches $\boldsymbol{\lambda}^*$, and after n_2 steps the algorithm successfully tracks $\boldsymbol{\lambda}^*$ and achieves the asymptotic regret of $\widetilde{O}(n^{-1/2})$. This regret matches the minimax rate for regret minimization and active exploration (e.g., GAFS-MAX). This shows that operating a trade-off between rewards and errors is not fundamentally more difficult than optimizing either of the objectives individually. While in this analysis, the second and third phases are sharply separated (and n_2 may be large), in practice the performance gradually improves as $\widehat{\boldsymbol{\lambda}}$ approaches $\boldsymbol{\lambda}^*$.

Remark 2 (dependence on parameters). $\lambda_{\min}$ has a major impact on the bound. The smaller its value, the higher the regret, both explicitly and through the smoothness β. At the same time, the larger $\lambda_{\min}$ the stricter Asm. 1, which limits the validity of Thm. 1. A possible compromise is to set $\lambda_{\min}$ to an appropriate decreasing function of n, thus making Asm. 1 always verified (for a large enough n), at the cost of worsening the rate of the regret. In the experiments, we run ForcingBalance with $\lambda_{\min} = 0$ without the regret being negatively affected. We conjecture that we can always set $\lambda_{\min} = 0$ (for which Asm. 1 is always verified), while the bound could be refined by replacing $\lambda_{\min}$ (the ForcingBalance parameter) with $\lambda^*_{\min}$ (the minimum optimal allocation). Nonetheless, we point out that this would require to significantly change the structure of the proof as Lem. 2 does not hold anymore when $\lambda_{\min} = 0$.

Remark 3 (dependence on the problem). The remaining terms in the bound depend on the number of arms K, w, $\sigma^2_{\min}$ (through α), and $\lambda^*_{\min}$. By definition of α, we notice that as w tends to 1 (pure reward maximization), the bound gets worse. This is expected since the proof relies on the strong convexity of f_w to relate the accuracy in estimating f_w and the accuracy of the allocations (see Lem. 3). Finally, the regret has an inverse dependence on $\lambda^*_{\min}$, which shows that if the optimal allocation requires an arm to be selected only a very limited fraction of the time, the problem is more challenging and the regret increases. This may happen in a range of different configurations such as the large value of w or when one arm has very high mean and variance, which leads to a $\boldsymbol{\lambda}^*$ highly concentrated on one single arm and $\lambda^*_{\min}$ very small. A very similar dependence is already present in previous results for active exploration (see e.g., Carpentier et al. [2011]).

Remark 4 (proof). A sketch and the complete proof are in App. B. While the proof shares a similar structure as GAFS-MAX's, in GAFS-MAX we have access to an explicit form of the optimal allocation $\boldsymbol{\lambda}^*$ and the proof directly measures the difference between allocations. Here, we have to rely on Lemmas 3 and 4 to relate allocations to objective functions and vice versa. In this sense, our analysis is a generalization of the proof in GAFS-MAX and it can be applied to any strongly convex and smooth objective function.

5 Experiments

We evaluate ForcingBalance on synthetic data and a problem directly derived from an educational application. Additional experiments are in the appendix.

5.1 Synthetic Data

We consider a MAB with $K = 5$ arms with mean and variance given in Fig. 4. While $\rho(\boldsymbol{\lambda})$ is optimized by always pulling arm 5, $\varepsilon(\boldsymbol{\lambda})$ is minimized by an allocation selecting more often arm 4 that has the larger variance (for $w = 0$, the optimal allocation λ^*_4 is over 0.41). For $w = 0.9$ (i.e., more weight to cumulative reward than estimation error) the optimal allocation $\boldsymbol{\lambda}^*$ is clearly biased towards arm 5 and only partially to arm 4, while all other arms are pulled only a limited fraction of time (well below 2%). We run ForcingBalance with $\eta = 1$ and $\lambda_{\min} = 0$ and we average over 200 runs.

Dependence on n. In Fig. 3-*(left)* we report the average and the 0.95-quantile of the rescaled regret $\widetilde{R}_n = \sqrt{n}R_n$. From Thm. 1 we expect the rescaled regret to increase as $\sqrt{n}$ in the first exploration phase, then to increase as $n^{1/4}$ in the second phase, and finally converge to a constant (i.e., when the actual regret enters into the asymptotic regime of $\widetilde{O}(n^{-1/2})$). From the plot we see that this is mostly verified by the empirical regret, although there is a transient phase during which the rescaled regret decreases over n, which suggests that the actual regret may decrease with a faster rate, at least in a first moment. This behavior may be captured in the theoretical analysis by replacing the use of Hoeffding bounds with Bernstein concentration in-

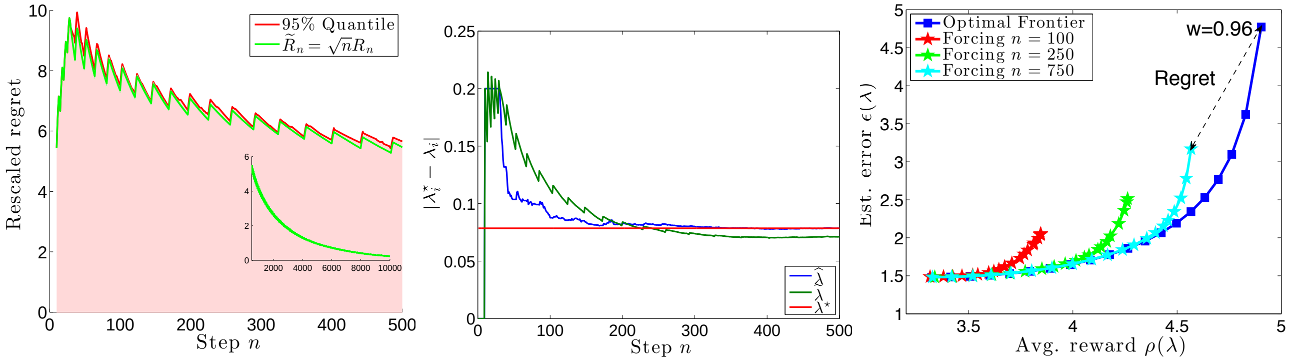

Figure 3: Rescaled regret *(left)*, allocations errors *(center)*, Pareto frontier *(right)* for the setting in Fig. 4.

	μ	σ^2	$\boldsymbol{\lambda}^*$
Arm1	1.0	0.05	0.0073
Arm2	1.5	0.1	0.01
Arm3	2.0	0.2	0.014
Arm4	4.0	**4.0**	0.0794
Arm5	**5.0**	0.5	0.8893

Figure 4: Arm mean, variance and optimal allocation for $w = 0.9$.

equalities, which may reveal faster rate (up to $\widetilde{O}(1/n)$) whenever n and the standard deviations are small.

Tracking. In Fig. 3-*(center)*, we study the behavior of the estimated allocation $\widehat{\boldsymbol{\lambda}}$ and the actual allocation $\widetilde{\boldsymbol{\lambda}}$ (we show $\widetilde{\lambda}_4$ and $\widetilde{\lambda}_4$) w.r.t. the optimal allocation ($\lambda^*_4 = 0.0794$). In the initial phase, $\widehat{\boldsymbol{\lambda}}$ is basically uniform ($1/K$) since the algorithm is always in forcing mode. After the exploration phase, $\widehat{\boldsymbol{\lambda}}$ is computed on the estimates that are already quite accurate, and it rapidly converges to $\boldsymbol{\lambda}^*$. At the same time, $\widetilde{\boldsymbol{\lambda}}$ keeps tracking the estimated optimal allocation and it also tends to converge to $\boldsymbol{\lambda}^*$ but with a slightly longer delay. We further study the tracking rule in the appendix.

Pareto frontier. In Fig. 3-*(right)* we study the performance of the optimal allocation $\boldsymbol{\lambda}^*$ for varying weights w. We report the Pareto frontier in terms of average reward $\rho(\boldsymbol{\lambda})$ and average estimation error $\varepsilon(\boldsymbol{\lambda})$. The optimal allocation smoothly changes from focusing on arm 4 to being almost completely concentrated on arm 5 ($\lambda^*_4 = 0.41$ and $\lambda^*_5 = 0.20$ for $w = 0.0$ and $\lambda^*_4 = 0.0484$ and $\lambda^*_4 = 0.9326$ for $w = 0.95$). As a result, we move from an allocation with very low estimation error but poor reward to a strategy with large reward but poor estimation. We report the Pareto frontier of ForcingBalance for different values of n. In this setting, ForcingBalance is more effective in approaching the performance of $\boldsymbol{\lambda}^*$ for small values of w. This is consistent with the fact that for $w = 0$, $\lambda^*_{\text{min}} = 0.097$, while it decreases to 0.004 for $w = 0.95$, which increases the regret as illustrated by Thm. 1.

5.2 Educational Data

Treefrog Treasure is an educational math game in which players navigate through a world of number lines. Players must find and jump through the appropriate fraction on each number line. To analyze the effectiveness of our algorithm when parameters are drawn from a real-world setting, we use data from an experiment in *Treefrog Treasure* to estimate the means and variances of a 64-arm experiment. Each arm corresponds to a different experimental condition: After a tutorial, 34,197 players each received a pair of number lines with different properties, followed by the same (randomized) distribution of number lines thereafter. We measured how many number lines students solved conditioned on the type of this initial pair; the hope is to learn which type of number line encourages player persistence on a wide variety of number lines afterwards. There were a total of $K = 64$ conditions, formed from choosing between 2 representations of the target fraction, 2 representations of the label fractions on the lines themselves, adding or withholding tick marks at regular intervals on the number line, adding or removing hinting animations if the problem was answered incorrectly, and 1-4 different rates of backoff hints that would progressively offer more and more detailed hints as the player made mistakes. The details of both the experiments and the experimental conditions are taken from Liu et al. [2014], though we emphasize that we measure a different outcome in this paper (player persistence as opposed to chance of correct answer).

We run ForcingBalance, standard UCB, GAFS-MAX (adapted to minimize the average estimation error) over $n = 25,000$ and 100 runs. Both ForcingBalance and GAFS-MAX use $\eta = 1$ and w is set to 0.6 to give priority to preference to the accuracy of the estimates and to 0.95 to favor the player's experience and entertainment. We study the performance according to the average reward $\rho(\boldsymbol{\lambda})$ (normalized by the largest mean), the estimation error $\varepsilon(\boldsymbol{\lambda})$ (normalized by the largest standard deviation), the rescaled regret $\sqrt{n}R_n$, the relative *discounted cumulative gain* (DCG) and the *RankErr* that measure how well arms are ranked on the basis of their

Figure 5: Treefrog Treasure, a math game about number lines.

Alg.	$\frac{\varepsilon(\boldsymbol{\lambda})}{\sigma^2_{\max}}$	$\frac{\rho(\boldsymbol{\lambda})}{\mu_{\max}}$	R_n	*RelDCG*	*RankErr*
$w = 0.95$					
$\boldsymbol{\lambda}^*$	6.549	0.9405	-	-	-
Force	6.708	0.9424	**1.878**	0.1871	5.935
UCB	11.03	**0.9712**	95.15	1.119	8.629
GAFS	**5.859**	0.9183	17.79	**0.1268**	**5.117**
Unif	5.861	0.9168	20.49	0.132	5.25
$w = 0.6$					
$\boldsymbol{\lambda}^*$	5.857	0.9189	-	-	-
Force	**5.859**	0.92	**0.4437**	**0.1227**	5.178
UCB	11.03	**0.9712**	1343	1.119	8.629
GAFS	**5.859**	0.9183	1.314	0.1268	**5.117**
Unif	5.861	0.9168	3.482	0.132	5.25

Figure 6: Results on the educational dataset.

mean.[6] Small values of *RelDCG* and *RankErr* mean that arms are estimated well enough to correctly rank them and can allow the experiment designer to later reliably remove the worst performing arms. The results are reported in Fig. 6. Since UCB, GAFS-MAX, and *Unif* do not depend on w, their performance is constant except for the regret, which is computed w.r.t. to different $\boldsymbol{\lambda}^*$s. As expected, UCB achieves the highest reward but it performs very poorly in estimating the arms' mean and in ranking them. GAFS-MAX does not collect much reward but is very accurate in the estimate of the means. On the other hand, ForcingBalance balances the two objectives and it achieves the smallest regret. We notice that ForcingBalance preserves a very good estimation accuracy without compromising too much the average reward (for $w = 0.95$). In this situation, effectively balancing between the two objectives allows us to rank the different game settings in the right order while providing players with a good experience. Had we used UCB, the outcome for players would have been better, but the designers would have less insight into how the different ways of providing number lines affect player behavior for when they need to design the next game (high *RankErr*). Alternatively, using GAFS-MAX would give the designer excellent insight into how different number lines affect players; however, if some conditions are too difficult, we could have caused many players to quit. ForcingBalance provides a useful feedback to the designer without compromising the players' experience (the *RankErr* is close to GAFS-MAX but the reward is higher). This is more evident when moving to $w = 0.95$, where ForcingBalance significantly improves the reward w.r.t. GAFS-MAX without losing much accuracy in ranking the arms.

6 Conclusions

We studied the tradeoff between rewards and estimation errors. We proposed a new formulation of the problem, introduced a variant of a forced-sampling algorithm, derived bounds on its regret, and we validated our results on synthetic and educational data.

There are a number of possible directions for future work. **1)** An active exploration strategy tends to pull all arms a linear fraction of time while minimizing regret requires selecting sub-optimal arms a sublinear number of times. It would be interesting to prove an explicit incompatibility result between maximizing $\rho(\boldsymbol{\lambda})$ and minimizing $\varepsilon(\boldsymbol{\lambda})$ similar to the result of Bubeck et al. [2009] for simple and cumulative regret. **2)** While a straightforward application of the UCB fails, alternative formulations, such as using upper-bounds on both means and variances, could overcome the limitations of (μ, σ)-Naive-UCB. Nonetheless, the resulting function $f_w(\cdot; \{\widetilde{\nu}_{i,n}\})$ is neither an upper nor a lower bound on the true function $f_w(\cdot; \{\nu_i\})$ and the regret analysis could be considerably more difficult than for ForcingBalance. Furthermore, it would be interesting to study how a Thompson sampling approach could be adapted. **3)** Finally, alternative tradeoffs can be formulated (e.g., simple vs. cumulative regret). Notice that the current model, algorithm, and analysis could be easily extended to any strongly-convex and smooth function defined over some parameters of the arms' distributions.

Acknowledgements The research presented in this paper was supported by French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council, Inria and Carnegie Mellon University associated-team project EduBand, and French National Research Agency projects ExTra-Learn (n.ANR-14-CE24-0010-01) and BoB (n.ANR-16-CE23-0003).

[6] Let π^* be the true ranking and $\widehat{\pi}$ the estimated ranking (i.e., $\widehat{\pi}(k)$ returns the identity of the arm ranked at position k), the DCG is computed as $\mathrm{DCG}_\pi = \sum_{k=1}^K \frac{\mu_{\pi(k)}}{\log(k+1)}$ and then we compute $(\mathrm{DCG}_{\pi^*} - \mathrm{DCG}_{\widehat{\pi}})/\mathrm{DCG}_{\pi^*}$, while $RankErr = 1/K \sum_{i=1}^K |\pi^*(i) - \widehat{\pi}(i)|$.

References

András Antos, Varun Grover, and Csaba Szepesvári. Active learning in heteroscedastic noise. *Theoretical Computer Science*, 411:2712–2728, June 2010.

Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *Conference on Learning Theory*, 2010.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 169–207. Springer, 2003.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory*, 2009.

Loc X. Bui, Ramesh Johari, and Shie Mannor. Committing bandits. In *Neural Information Processing Systems*. 2011.

Alexandra Carpentier, Alessandro Lazaric, Mohammad Ghavamzadeh, Rémi Munos, and Peter Auer. Upper-confidence-bound algorithms for active learning in multi-armed bandits. In *Algorithmic Learning Theory*, 2011.

Alexandra Carpentier, Remi Munos, and András Antos. Adaptive strategy for stratified monte carlo sampling. *Journal of Machine Learning Research*, 16:2231–2271, 2015.

Madalina M. Drugan and Ann Nowé. Designing multi-objective multi-armed bandits algorithms: A study. In *International Joint Conference on Neural Networks*, 2013.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7: 1079–1105, 2006.

Alexander Goldenshluger and Assaf Zeevi. A linear response bandit problem. *Stoch. Syst.*, 3(1):230–261, 2013. doi: 10.1214/11-SSY032.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Neural Information Processing Systems*, 2007.

Tor Lattimore. The pareto regret frontier for bandits. In *Neural Information Processing Systems*. 2015.

Yun-En Liu, Travis Mandel, Emma Brunskill, and Zoran Popovic. Trading off scientific knowledge and user learning with multi-armed bandits. In *International Conference on Educational Data Mining*, 2014.

A. Maurer and M. Pontil. Empirical bernstein bounds and sample-variance penalization. In *Conference on Learning Theory*, 2009.

Amir Sani, Alessandro Lazaric, and Rémi Munos. Risk averse multi-arm bandits. In *Neural Information Processing Systems*, 2012.

Csaba Szepesvári. Learning theory of optimal decision making. Lecture notes of the 2008 Machine Learning Summer School, 2008. URL `https://www.ualberta.ca/~szepesva/Talks/MLSS-IleDeRe-day1.pdf`.

Douglas P. Wiens and Pengfei Li. V-optimal designs for heteroscedastic regression. *Journal of Statistical Planning and Inference*, 145:125 – 138, 2014.

S. Yakowitz and T.-L. Lai. The nonparametric bandit approach to machine learning. In *IEEE Conference on Decision and Control*, volume 1, pages 568–572, 1995.