Constrained Best Arm Identification with Tests for Feasibility

Ting Cai, Kirthevasan Kandasamy

University of Wisconsin-Madison
tingcai@cs.wisc.edu, kandasamy@cs.wisc.edu

Abstract

Best arm identification (BAI) aims to identify the highest-performance arm among a set of K arms by collecting stochastic samples from each arm. In real-world problems, the best arm needs to satisfy additional feasibility constraints. While there is limited prior work on BAI with feasibility constraints, they typically assume the performance and constraints are observed simultaneously on each pull of an arm. However, this assumption does not reflect most practical use cases, e.g., in drug discovery, we wish to find the most potent drug whose toxicity and solubility are below certain safety thresholds. These safety experiments can be conducted separately from the potency measurement. Thus, this requires designing BAI algorithms that not only decide which arm to pull but also decide whether to test for the arm's performance or feasibility. In this work, we study feasible BAI which allows a decision-maker to choose a tuple (i, ℓ), where $i \in [K]$ denotes an arm and ℓ denotes whether she wishes to test for its performance ($\ell = 0$) or any of its N feasibility constraints ($\ell \in [N]$). We focus on the fixed confidence setting, which is to identify the *feasible* arm with the *highest performance*, with a probability of at least $1 - \delta$. We propose an efficient algorithm and upper-bound its sample complexity, showing our algorithm can naturally adapt to the problem's difficulty and eliminate arms by worse performance or infeasibility, whichever is easier. We complement this upper bound with a lower bound showing that our algorithm is *asymptotically ($\delta \to 0$) optimal*. Finally, we empirically show that our algorithm outperforms other state-of-the-art BAI algorithms in both synthetic and real-world datasets.

Extended version — https://arxiv.org/abs/2511.09808

Introduction

Best arm identification (BAI) contains K arms that allow a decision maker to repeatedly pull one of the arms and observes an i.i.d. sample drawn from the distribution associated with that arm (Bechhofer 1958; Paulson 1964). In the fixed confidence setting of BAI, given a target failure rate $\delta > 0$, the decision-maker aims to identify the arm with the *highest expected performance*, with probability at least $1 - \delta$, while keeping the number of pulls to a minimum. This problem has been widely applied in areas such as drug discovery, crowdsourcing, distributed systems, and A/B testing (Jun et al. 2016; Koenig and Law 1985; Schmidt, Branke, and Chick 2006; Van Aken et al. 2017; Zhou, Chen, and Li 2014).

In many real-world applications, we additionally wish to find the optimal arm that satisfies *feasibility constraints*, which cannot be modeled analytically and need to be evaluated via costly experimentation. Moreover, these feasibility experiments can usually be tested separately from the performance evaluation. As a motivating example, in drug discovery, we wish to find the most potent drug, but also need to guarantee that the drug is soluble in the bloodstream, and the risk of adverse side effects is small. Scientists have developed different experiments to measure the drug's potency (performance), solubility (feasibility), and toxicity (feasibility), which can be separately carried out independent of each other (Thall and Russell 1998). Similarly, in database tuning applications (Van Aken et al. 2017), we wish to minimize the end-to-end latency, while guaranteeing the risk of system-wide failures is small. The latency and robustness can also be evaluated via separate tests (Kanellis, Alagappan, and Venkataraman 2020).

Majority of the prior work on BAI does not study feasibility constraints. The only work on feasible BAI (Katz-Samuels and Scott 2018, 2019) assumes that the performance and feasibility tests are all conducted simultaneously, which does not hold in many real-world settings, such as the examples highlighted above Often, we can conduct separate experiments and only observe the result of the experiment conducted. Thus, applying methods in prior works to our setting by naively pulling all arms simultaneously can be unnecessarily expensive as it samples all performance and feasibility distributions when testing one arm and does not focus on the most important experiment to determine an arm's optimality and/or feasibility. Moreover, from a theoretical perspective, simultaneous pulls cannot capture the *actual complexity* of the number of individual tests.

In our paper, we introduce a novel BAI formalism. Given K arms, each arm is associated with $N + 1$ distributions. For $i \in [K]$ and $\ell \in \{0\} \cup [N]$, $\mu_{i,\ell} \in [0, 1]$ denotes the unknown mean of arm i's ℓ^{th} distribution. On each round, the decision-maker chooses a tuple (i, ℓ) where $i \in [K]$ denotes the arm and ℓ denotes whether she wishes to test for its performance ($\ell = 0$) or its ℓ^{th} feasibility constraint ($\ell \in [N]$).

The thresholds for each feasibility constraint are given and for simplicity[1], we assume the thresholds are all $\frac{1}{2}$. An arm is said to be *feasible* if $\mu_{i,\ell} < \frac{1}{2}$ for all $\ell \in [N]$. Given $\delta \in (0,1)$, the goal of the decision-maker is to identify the optimal arm, i.e. the feasible arm with the highest $\mu_{i,0}$, with probability at least $1-\delta$, while minimizing the total number of collected samples.

*The **key challenge** to solve this problem is to balance when to test for feasibility and when to test for performance for different types of suboptimal arms at the same time.* We illustrate it using two naive algorithms:

First, consider a naive two-stage algorithm that identifies all feasible arms first and then executes a BAI algorithm on these arms. This algorithm is inefficient on suboptimal arms whose feasibility means $\mu_{i,\ell}$ for $\ell \in [N]$ are close to $\frac{1}{2}$ but $\mu_{i,0}$ value is much smaller compared to the optimal arm. Because it will waste many samples[2] on testing feasibility while these arms could have been eliminated faster if we started by comparing performance.

Consider another algorithm that first tries to identify the best-performance arm and then tests its feasibility. It will eliminate the arm if it is infeasible and repeat with the remaining arms until it finds a feasible arm. This algorithm is inefficient on another type of suboptimal arms, whose $\mu_{i,0}$ values are slightly larger than the optimal arm but are clearly infeasible, i.e., $\mu_{i,\ell} \gg \frac{1}{2}$ for $\ell \in [N]$. Because it will waste many samples to differentiate the performance of these arms which could have been easily eliminated by feasibility. Note that these two algorithms could indeed efficiently eliminate the two types of suboptimal arms described for each other.

In summary, the **main contributions** of this paper are:

1. Problem formalism and lower bound: We define a novel feasible BAI problem and first quantify the complexity of the problem by developing a novel complexity term for each type of arm, which captures the best way to eliminate each type of suboptimal arm and the cost necessary to identify the optimal arm. Leveraging the insights from the complexity terms, we then provide a gap-dependent lower bound on the total expected sample complexity.

2. Algorithm design and upper bound: We propose an algorithm which tests performance and feasibility for each arm simultaneously but only tests its performance and/or at most one of its N feasibility constraints, which can eliminate all suboptimal arms in the easiest way. Moreover, the number of samples collected by the algorithm does not scale linearly with the number of feasibility constraints N. We then provide the upper bound on the expected sample complexity of the algorithm, which is shown to be asymptotically ($\delta \to 0$) optimal compared with the lower bound.

3. Experiments: We empirically compare our algorithm with other state-of-the-art algorithms on both synthetic and real-world datasets from drug discovery. In all the experiments, our algorithm outperforms all other algorithms.

[1] It can be easily extended to different thresholds for different constraints.

[2] Recall that, given two sub-Gaussian distributions with means μ and μ', and a threshold ξ, we require $O((\mu-\mu')^{-2})$ samples to decide if $\mu < \mu'$ or $\mu > \mu'$, and $O((\mu-\xi)^{-2})$ samples to decide if $\mu < \xi$ or $\mu > \xi$.

Related Work

The bandit framework is a popular paradigm to study exploration-exploitation tradeoffs that occur in sequential decision-making under uncertainty (Lai and Robbins 1985; Thompson 1933). Here, a decision-maker adaptively samples one of K arms from a bandit model so as to achieve a given objective. There is extensive prior work on developing algorithms for BAI, whose goal is to identify the arm with the highest mean in the fixed confidence or fixed budget setting (Bechhofer 1958; Bubeck, Munos, and Stoltz 2009; Even-Dar, Mannor, and Mansour 2002; Jamieson et al. 2014; Kalyanakrishnan et al. 2012; Karnin, Koren, and Somekh 2013; Paulson 1964). Several works have also developed hardness results for BAI (Kaufmann, Cappé, and Garivier 2016; Mannor and Tsitsiklis 2004). Our algorithm and theoretical results build on this rich line of work. In particular, our algorithm's sampling strategy is inspired by the LUCB algorithm of (Kalyanakrishnan et al. 2012). However, none of these work studies constrained BAI, where the feasibility constraints need to be tested separately.

Our model for testing feasibility is closely related to thresholding bandits, i.e., identify all arms whose mean is larger than a given threshold (Katz-Samuels and Scott 2018; Locatelli, Gutzeit, and Carpentier 2016; Mason et al. 2020). In our work, however, it is sufficient to identify just one constraint that is larger than the threshold to determine if it is infeasible. Another similar line of work is on finding if there is any arm below a threshold (Degenne and Koolen 2019; Kano et al. 2019; Katz-Samuels and Jamieson 2020; Kaufmann, Koolen, and Garivier 2018). While we use some ideas from these works when testing the feasibility of an arm, their algorithms and analysis cannot be directly applied to our setting, since we need to determine if the most efficient way to eliminate an arm is via its sub-optimality or infeasibility.

Perhaps the closest work to ours is (Katz-Samuels and Scott 2019), who study identifying the best m arms that satisfy the given feasibility constraints. They assume each arm i is associated with a D-dimensional distribution with mean $\boldsymbol{\mu}_i$ and aim to find the top m arms that maximize $\mathbf{r}^\top \boldsymbol{\mu}_i$ subject to the constraint $\boldsymbol{\mu}_i \in P$. Here $\mathbf{r}$ is a given direction for the reward and P is a subset of $\mathbb{R}^D$ denoting the feasible set. However, this is different from our setting, since we allow the objective and feasibility constraints to be tested separately. A naive application of their algorithm will require testing an arm's performance and all feasibility constraints each time when we wish to test an arm, which is sample inefficient in our setting. Chen et al. (2017) studies identifying the best set in a family of feasible subsets. Our work is different from theirs since they did not consider testing for feasibility separately and they assume the feasible subsets have various combinatorial structures.

Our work is also related to work in the Bayesian optimization literature that studies zeroth order constrained optimization and multi-objective optimization, where multiple performance objectives and constraints can be evaluated separately, similar to our setting (Gardner et al. 2014; Ungredda and Branke 2021; Eriksson and Poloczek 2021; Kirschner et al. 2022; Hernández-Lobato et al. 2016; Berkenkamp, Krause, and Schoellig 2021; Kirschner et al. 2019; Paria, Kandasamy, and Póczos 2020; Hernández-Lobato et al. 2016). These problem settings are distinctly different from ours as we consider the K–armed version of the problem.

Problem Setup

For $a \in \mathbb{N}_+$, we denote $[a] = \{1, \dots, a\}$. Given K arms. each arm $i \in [K]$ is associated with $N + 1$ distributions $\{\nu_{i,\ell}\}_{\ell=0}^{N}$, which are all 1 sub-Gaussian for simplicity[3]. Let $\mu_{i,\ell} = \mathbb{E}_{X\sim\nu_{i,\ell}}[X]$ be the mean of distribution $\nu_{i,\ell}$. We assume w.l.o.g that $\mu_{i,\ell} \in [0, 1]$ for all i, ℓ. For any arm i, $\nu_{i,0}$ is associated with its performance, and $\nu_{i,\ell}$, for $\ell \in [N]$ is associated with feasibility constraint ℓ. The set of all feasible arms $\mathcal{F}$ is defined as

$$\mathcal{F} = \left\{ i \in [K]\,;\ \mu_{i,\ell} < \frac{1}{2} \ \text{ for all } \ell \in [N] \right\}.$$

If $\mathcal{F} \neq \emptyset$, we define the optimal arm $i^\star$ as the feasible arm with the highest expected performance. Otherwise, we define $i^\star = K + 1$ (indicates there is no feasible arm).

$$i^\star = \begin{cases} \arg\max_{i\in\mathcal{F}} \mu_{i,0}, & \text{if } \mathcal{F} \neq \emptyset \\ K + 1, & \text{if } \mathcal{F} = \emptyset \end{cases} \tag{1}$$

W.l.o.g., we assume that $\mu_{1,0} > \mu_{2,0} > \cdots > \mu_{i^\star,0} > \cdots > \mu_{K,0}$. For each arm i, we assume that the $\{\mu_{i,\ell}\}_{\ell\in[N]}$ are also not equal to each other and $\mu_{i,\ell} \neq \frac{1}{2}$ for all i, ℓ. Note the mean values and the ordering are unknown.

An algorithm for this problem proceeds over a sequence of rounds, terminates, and then recommends an arm as the optimal feasible arm. On each round r, based on past observations, it chooses (i_r, ℓ_r) that determines which distribution, i.e., arm $i_r \in [K]$ and performance/feasibility $\ell_r \in \{0\} \cup [N]$, to sample from. When it stops, it outputs $\widehat{a} \in [K + 1]$. If $\widehat{a} \in [K]$, then the decision-maker recommends $\widehat{a}$ as the optimal feasible arm, and if $\widehat{a} = K + 1$, then it declares that there is no feasible arm.

Lower Bound

In this section, we will first define the complexity of the feasible BAI problem and then provide a lower bound for any δ-correct algorithm. To motivate the ensuing discussion, consider the bandit instance in Fig. 1, where we have $K = 5$ arms and $N = 1$ feasibility constraint. The optimal arm is $i^\star = 2$. Since $\mu_{1,0} > \mu_{2,0}$, the only way to determine $i^\star \neq 1$ is to verify arm 1 is infeasible, i.e. $\mu_{1,1} > \frac{1}{2}$. Since arm 4 is feasible $\mu_{4,1} < \frac{1}{2}$, the only way to determine that $i^\star \neq 4$ is to verify arm 4 has worse performance than $i^\star$, i.e., $\mu_{2,0} > \mu_{4,0}$. Note that arms 3 and 5 are both infeasible and have lower performance than $i^\star$, thus they can be eliminated via both ways. Of these, we may prefer to determine $i^\star \neq 3$ based on feasibility since $\mu_{3,0} \approx \mu_{2,0}$, but $\mu_{3,1} \gg \frac{1}{2}$. However, we may prefer to determine $i^\star \neq 5$ based on performance since $\mu_{5,1} \approx \frac{1}{2}$, but $\mu_{5,0} \ll \mu_{2,0}$.

[3]The variances can be easily extended to other numbers and we will get the same result via scaling

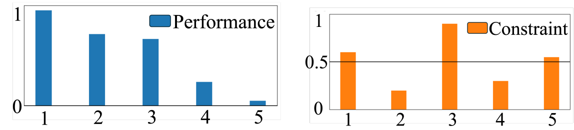

Figure 1: An example bandit instance when $K = 5$ and $N = 1$. The optimal feasible arm is $i^\star = 2$.

For each arm $i \in [K]$, we first define the complexity terms to determine its feasibility (θ_i) and performance (ϕ_i). We will start with feasibility. To verify that an arm i is feasible, we have to check $\mu_{i,\ell} < \frac{1}{2}$ for all $\ell \in [N]$. But to verify that an arm i is infeasible, it is sufficient to check only one constraint $\ell \in [N]$ such that $\mu_{i,\ell} > \frac{1}{2}$. If there are several such constraints, the easiest will be the constraint with the largest $\mu_{i,\ell}$. Hence, we define θ_i as:

$$\theta_i = \begin{cases} (\max_{\ell\in[N]} \mu_{i,\ell} - \frac{1}{2})^{-2}, & \text{if } i \notin \mathcal{F}, \\ \sum_{\ell=1}^{N} (\mu_{i,\ell} - \frac{1}{2})^{-2}, & \text{if } i \in \mathcal{F}. \end{cases} \tag{2}$$

We then define the complexity term ϕ_i to differentiate the performance between arm i and $i^\star$. Recall that arms are arranged in decreasing order of expected performance. For arms $i < i^\star$, i.e. those have better performance but are infeasible, the only way to determine $i \neq i^\star$ is to test for their feasibility; hence, we define $\phi_i = \infty$ for all $i < i^\star$. For $i > i^\star$, we let $\phi_i = (\mu_{i^\star,0} - \mu_{i,0})^{-2}$. Finally, for $i^\star$, we set $\phi_{i^\star} = 0$ if there are no other feasible arms since it is unnecessary to verify that its performance is better than other arms; otherwise, we set it to the inverse squared gap between $i^\star$ and the feasible arm with the second-highest performance. Putting this all together, we have:

$$\phi_i = \begin{cases} \infty, & \text{if } i < i^\star, \\ (\mu_{i^\star,0} - \mu_{i,0})^{-2}, & \text{if } i > i^\star, \\ 0, & \text{if } i = i^\star, |\mathcal{F}| = 1, \\ (\mu_{i^\star,0} - \max\limits_{j\in\mathcal{F}\setminus\{i^\star\}} \mu_{j,0})^{-2}, & \text{if } i = i^\star, |\mathcal{F}| > 1, \end{cases} \tag{3}$$

which stands true when there is no feasible arm, i.e. $i^\star = K+1$. Now, we define two sets $\mathcal{I}$ and $\mathcal{W}$ using the complexity terms in (2) and (3). Intuitively, $\mathcal{I}$ are the arms that are easier to eliminate due to infeasibility, and $\mathcal{W}$ are the arms that are easier to eliminate due to their worse performance than $i^\star$. We have,

$$\mathcal{I} = [i^\star - 1] \cup \Big(\{i^\star + 1, \dots, K\} \cap \{i \in \mathcal{F}^{\mathsf{c}} : \theta_i < \phi_i\}\Big),$$

$$\mathcal{W} = \{i^\star + 1, \dots, K\} \cap \Big(\mathcal{F} \cup \{i \in \mathcal{F}^{\mathsf{c}} :\, \theta_i \geq \phi_i\}\Big).$$

Observe that $[K] = \mathcal{I} \cup \mathcal{W} \cup \{i^\star\}$. This leads to the following definition of the complexity of an arm $\mathcal{H}_i$ and the complexity of the problem $\mathcal{H}$ which is the sum of the arm complexities.

$$\mathcal{H}_i = \begin{cases} \theta_i, & \text{if } i \in \mathcal{I}, \\ \phi_i, & \text{if } i \in \mathcal{W}, \\ \theta_i + \phi_i, & \text{if } i = i^\star \text{ and } i^\star \leq K, \end{cases}$$

$$\mathcal{H} = \sum_{i=1}^{K} \mathcal{H}_i. \tag{4}$$

It suggests that for all suboptimal arms, we only need to verify either they are infeasible or have worse performance than $i^\star$. For $i^\star$, we need to verify that it is both feasible and that its mean $\mu_{i^\star,0}$ is larger than the feasible arm with the second-highest performance. Next, we show the lower bound of the expected sample complexity for any δ-correct algorithms.

Theorem 1. *Let ν denote a bandit instance with Gaussian observations satisfying assumptions in §PROBLEM SETUP. Let $\delta \in (0, 1)$ and $\mathcal{H}$ defined in* (4). *Any algorithm $\mathcal{A}$ that is δ-correct has a stopping time τ on ν that satisfies*

$$\mathbb{E}_\nu[\tau] \geq 2\mathcal{H} \log \frac{1}{2.4\delta}.$$

The lower bound in Theorem 1 has an expected sample complexity aligning with the complexity terms in (4). The main challenge in proving the lower bound is to construct alternate bandit instances that match the complexity terms. All proofs are in the Appendix of the extended version.

Algorithm and Upper Bound

We now present our algorithm and its upper bound of the expected sample complexity. Our algorithm, outlined in Algorithm 1, proceeds over a series of epochs, indexed by t. At each epoch, it will choose up to two arms $a_t, b_t \in [K]$. For each arm, it will test the performance and/or one of its feasibility constraints.

To describe our algorithm in detail, we first define some quantities. For all $i \in [K], \ell \in \{0\} \cup [N]$, $N_{i,\ell}(t)$ denotes the number of times the tuple (i, ℓ) has been sampled and $M_i(t)$ denotes the total number of times arm i has been tested for feasibility up to epoch t:

$$N_{i,\ell}(t) = \sum_{s=1}^{t} \mathbb{1}[(i, \ell) \text{ is sampled on epoch } s],$$

$$M_i(t) = \sum_{\ell=1}^{N} N_{i,\ell}(t).$$

Next, we define the upper and lower confidence bounds for each mean value. Let $X_{i,\ell,s}$ be the sample collected from arm i's ℓ^{th} distribution when it is sampled for the s^{th} time. $\widehat{\mu}_{i,\ell}(t)$ denotes the empirical mean of the distribution up to epoch t. Let $D(n, \delta)$ denote the uncertainty term, the upper and lower confidence bounds $\overline{\mu}_{i,\ell}(t), \underline{\mu}_{i,\ell}(t)$ for $\mu_{i,\ell}$ are:

$$\widehat{\mu}_{i,\ell}(t) = \frac{\sum_{s=1}^{N_{i,\ell}(t)} X_{i,\ell,s}}{N_{i,\ell}(t)},\ D(n, \delta) = \sqrt{\frac{2}{n} \log \frac{4n^4}{\delta}}, \tag{5}$$

$$\overline{\mu}_{i,\ell}(t) = \widehat{\mu}_{i,\ell}(t) + D(N_{i,\ell}(t), \delta/(K(N+1))),$$

$$\underline{\mu}_{i,\ell}(t) = \widehat{\mu}_{i,\ell}(t) - D(N_{i,\ell}(t), \delta/(K(N+1))).$$

For all $i \in [K], \ell \in [N]$ at epoch t, we define $\widetilde{\mu}_{i,\ell}(t)$ as follows which will help us to determine which constraint to test when we wish to test the feasibility of arm i:

$$\widetilde{\mu}_{i,\ell}(t) = \widehat{\mu}_{i,\ell}(t) + \sqrt{\frac{2 \log M_i(t)}{N_{i,\ell}(t)}}. \tag{6}$$

At each epoch, Algorithm 1 samples arms that are most probable to be $i^\star$ and simultaneously eliminate suboptimal arms, which are either determined to be infeasible or have worse performance than an arm that has been determined to be feasible. It maintains a few sets of arms that are updated at the end of each epoch. First, S denotes the set of surviving arms, i.e. arms that are not eliminated. Second, we form a focus set P, which contains arms in S that have high performance based on the samples collected so far, see line 18. Next, we have F (and I) which are arms determined to be feasible (infeasible) with probability at least $1 - \delta$ based on the samples collected so far. Finally, for each arm $i \in [K]$, we maintains set $H_i \subset [N]$ which contains the feasibility constraints of i that have not yet been determined to be feasible. We initialize our algorithm by pulling each arm and each distribution once.

We first describe the main sampling rule in each epoch in lines 6–11, and then describe lines 2–4. On each epoch, we choose up to two arms a_t and b_t, and evaluate their performance. This sampling rule, which is adapted from the LUCB algorithm (Kalyanakrishnan et al. 2012), chooses a_t to be the arm with the highest sample mean in P, and b_t to be the arm with the highest upper confidence bound on performance in $P \backslash \{a_t\}$. Intuitively, this choice of a_t and b_t focuses on the arms that are most likely to be optimal, but are hardest to distinguish. After testing for performance, if a_t or b_t have not been determined to be feasible yet, i.e. if they are not in F, then they are tested for exactly one feasibility constraint as suggested by the subroutine SAMPLE-FEASIBILITY. The case where P contains only one arm i (lines 3–4), indicates that its performance has been deemed to be clearly higher than the rest of the arms in S. Hence, if it is feasible, it will clearly be the optimal arm. Thus, if it is already guaranteed to be feasible, we can stop and return this arm (line 3), but otherwise, we should keep testing if it is feasible (line 4) until it is clear as to whether it is feasible or not. Finally, if $S = \emptyset$, this is because all arms have been deemed infeasible, so we return $K + 1$ (line 2).

Inspired from Kano et al. (2019), we design the subroutine SAMPLE-FEASIBILITY, which recommends one constraint to test when we wish to test the feasibility of an arm. Each time we invoke it with an arm $i \in [K]$, it chooses the constraint in H_i that maximizes $\widetilde{\mu}_{i,\ell}(t)$, as in (6). Intuitively, via this choice, we are sampling the constraint that is most likely to be larger than the threshold $\frac{1}{2}$. After sampling, if the lower confidence bound for the sampled constraint is larger than $\frac{1}{2}$, we deem it to be infeasible so we add i to I; then i will be eliminated from S shortly. otherwise, if its upper confidence bound is smaller than $\frac{1}{2}$, we have verified that the sampled constraint is not in violation and remove it from H_i. In the event that H_i is empty, then none of the constraints are in violation, and hence we add i to F as we have verified that i is feasible.

At the end of each epoch, we update S and P as follows (lines 14–18). First, we eliminate all determined infeasible arms in I from S (line 14). Then, to eliminate worse-performance arms, S is updated to contain arms whose upper confidence bounds for performance are larger than the smallest lower confidence bounds in F. Intuitively, if arm i

Algorithm 1

Input: $[K]$ arms, $\delta \in (0, 1)$.
Parameter: $S \leftarrow [K], P \leftarrow [K], F \leftarrow \emptyset, I \leftarrow \emptyset, H_i \leftarrow [N]$ $\forall i \in [K]$.

for $t = 1, 2, \cdots$ **do**
  if $S = \emptyset$ **then**
    Return $K + 1$.
  else if $P = \{i\}$ is a singleton **and** $i \in F$ **then**
    Return i
  else if $P = \{i\}$ is a singleton **and** $i \notin F$ **then**
    SAMPLE-FEASIBILITY(i)
  else
    $a_t \leftarrow \arg\max_{i \in P} \widehat{\mu}_{i,0}(t)$
    $b_t \leftarrow \arg\max_{i \in P \setminus \{a_t\}} \overline{\mu}_{i,0}(t)$
    Sample performance of a_t and b_t.
    if $a_t \notin F$ **then**
      SAMPLE-FEASIBILITY(a_t)
    end if
    if $b_t \notin F$ **then**
      SAMPLE-FEASIBILITY(b_t)
    end if
  end if
  $S \leftarrow S \setminus I$
  if $F \neq \emptyset$ **then**
    $S \leftarrow \{i \in S : \overline{\mu}_{i,0}(t) > \max_{j \in F} \underline{\mu}_{j,0}(t)\}$
  end if
  $P \leftarrow \{i \in S : \overline{\mu}_{i,0}(t) > \max_{j \in S} \underline{\mu}_{j,0}(t)\}$
end for

Function: SAMPLE-FEASIBILITY(i)
$\ell_t \leftarrow \arg\max_{\ell \in H_i} \widetilde{\mu}_{i,\ell}(t)$
Sample constraint ℓ_t of arm i
if $\underline{\mu}_{i,\ell_t}(t) > 1/2$ **then**
  $I \leftarrow I \cup \{i\}$
else if $\overline{\mu}_{i,\ell_t}(t) < 1/2$ **then**
  $H_i \leftarrow H_i \setminus \{\ell_t\}$
  if $H_i = \emptyset$ **then**
    $F \leftarrow F \cup \{i\}$
  end if
end if

has been found to be feasible, and another arm j has a performance guaranteed to be lower than i, then we can eliminate j regardless its feasibility. Finally, we update P to be the arms in S whose upper confidence bounds for performance are larger than the smallest lower confidence bounds in S. Intuitively, P is the set of arms in S which have high performance, so our sampling in the next the round will focus on P. The benefit of forming the focus set P is if $i^\star \in P$, we do not need to spend extra samples on testing the feasibility of arms in $S \setminus P$; however, if $i^\star \notin P$, all arms in P would belong to $\mathcal{I}$ with high probability and since we need to test their feasibility anyway, it is harmless to focus on P first.

Upper bound

We will now state our main theoretical result, which shows that Algorithm 1 is δ-correct and upper bounds its expected sample complexity on the number of epochs, which is at most four times the number of rounds. To state this, we define the following terms, which are related to the gaps between different performance means and feasibility means to the threshold and will be useful in Theorem 2,

$$\Delta_{i,j} = \mu_{i,0} - \mu_{j,0}, \qquad \Gamma_{i,\ell} = (\mu_{i,\ell} - \frac{1}{2})^{-2},$$
$$\delta' = \delta/(K(N+1))^{\frac{1}{4}}.$$

Theorem 2. *Given $\delta \in (0, 1)$, for any bandit instance satisfying the assumptions in §PROBLEM SETUP, Algorithm 1 is δ-correct. If $i^\star \leq K$, the stopping time τ satisfies,*

$$\begin{aligned}\mathbb{E}[\tau] \leq & \sum_{i \in \mathcal{I}} 32\theta_i \log\left(\frac{91\theta_i}{\delta'} \log \frac{95\theta_i}{\delta'}\right) \\ & + 292\left(\sum_{i \notin \mathcal{I}} \phi_i\right) \cdot \log \frac{\sum_{i \notin \mathcal{I}} \phi_i}{\delta} \\ & + \sum_{\ell=1}^{N} 32\Gamma_{i^\star,\ell} \log\left(\frac{91\Gamma_{i^\star,\ell}}{\delta'} \log \frac{95\Gamma_{i^\star,\ell}}{\delta'}\right) \\ & + G_1 + G_2 + G_3. \end{aligned} \quad (7)$$

If $i^\star = K + 1$, i.e. if no feasible arm exists, τ satisfies

$$\mathbb{E}[\tau] \leq \sum_{i \in [K]} 32\theta_i \log\left(\frac{91\theta_i}{\delta'} \log \frac{95\theta_i}{\delta'}\right) + G_4 + G_5. \quad (8)$$

The quantities $G_1, \ldots, G_5$ are lower order terms which do not have a leading $\log(1/\delta)$ term or a leading δ term, which can be ignored when $\delta \to 0$, see their definition in Appendix.

The following corollary, which is straightforward to verify by comparing the terms in Theorem 2 with a leading $\log(1/\delta)$ term to $\mathcal{H}$ in (4), shows that our algorithm matches the lower bound as $\delta \to 0$.

Corollary 2.1. *Let $\delta \in (0, 1)$. Let $\mathcal{H}$ be defined as in Equation (4). Algorithm 1 satisfies*[4]*,*

$$\limsup_{\delta \to 0} \frac{\mathbb{E}[\tau]}{\log(1/\delta)} \in \widetilde{\mathcal{O}}(\mathcal{H})$$

The upper bound in Theorem 2 has a very similar structure to the lower bound in Theorem 1. It indicates that Algorithm 1 finds the easiest way to eliminate all non-optimal arms either by feasibility or worse performance. Corollary 2.1 shows that Algorithm 1 achieves the optimal sample complexity when $\delta \to 0$.

Gap between the lower and upper bound when $\delta \nrightarrow 0$: It is easy to verify that the sample complexity of Algorithm 1 matches the lower bound in Theorem 1 when $N = 1$ for any $\delta \in (0, 1)$. However, when $N > 1$, Algorithm 1 can still match the lower bound for any δ on all arms except for arms in $\mathcal{I}$. This mismatch is due to that Theorem 1 can only be achieved by an oracle who already knows the structure and

[4] $\widetilde{\mathcal{O}}$ denotes up to constants and logarithmic factors. We believe the logarithmic factors could be reduced by adopting a tighter confidence bound as in (Jamieson et al. 2014).

the means of all arms and samples just to *verify* that $i^\star$ is the optimal arm. Theorem 1 suggests that we should only sample the constraint that has the highest mean for arms in $\mathcal{I}$. However, since we assume the decision-maker does not have all the additional information, it is expected to invest in samples on other constraints (not the highest one) for *exploration*. This also explains why in Thoerem 2, we have the lower order terms $(G_1, \cdots, G_5)$, as they account for the *cost for exploration*. Previous works (Simchowitz, Jamieson, and Recht 2017; Karnin 2016) have discussed the difference between *verification* and *exploration* but their tools cannot be directly applied to our problem due to the definition of feasibility. Kaufmann, Koolen, and Garivier (2018) provides a tighter lower bound when the distributions of the arms come from an exponential family, however, it still has a large gap between their upper bound. It remains an open question to lower bound the sample complexity any algorithm need to spend on the cost for *exploration*.

Proof Sketch of Theorem 2:

Our key contribution in the upper bound is to take a multi-level decomposition on the stopping time τ based on different combinations of the two elimination criteria and different selection rules in Algorithm 1. The decomposition also helps us to bound the cost of *exploration* for the arms in $\mathcal{I}$. The full proofs of Theorem 2 and Corollary 2.1 are in the Appendix. Here we provide the proof sketch for Theorem 2. Based on the elimination criteria, we first decompose τ into two parts based on whether a_t and b_t are in $\mathcal{I}$, which is $\tau =$

$$\underbrace{\sum_{t=1}^{\infty} \mathbb{1}[t \leq \tau, a_t \text{ or } b_t \in \mathcal{I}]}_{A_1} + \underbrace{\sum_{t=1}^{\infty} \mathbb{1}[t \leq \tau, a_t \text{ and } b_t \notin \mathcal{I}]}_{A_2}.$$

For A_1, we decompose each arm $i \in \mathcal{I}$ into two cases based on whether i's feasibility has been determined or not. When it is not determined, we upper-bound the case by the sample complexity to determine its feasibility. When determined, it is still chosen indicating it was incorrectly identified as a feasible arm, which will only happen with probability less than δ since Algorithm 1 is δ-correct. We upper bound it by δ times the sample complexity to differentiate the performance of each arm in $\mathcal{I}$ with all other arms.

For A_2, we decompose it into two cases where $b_t \in P$ or $b_t \notin P$. $b_t \notin P$ indicates $P = \{a_t\}$ is a singleton. $P = \{i^\star\}$ can be upper bounded by the sample complexity to determine arm $i^\star$'s feasibility. $P = \{a_t\}$ for $a_t \in \mathcal{W}$ will only happen with probability less than δ, since this indicates an arm in $\mathcal{W}$ is wrongly identified as better than $i^\star$ or arm $i^\star$ has been wrongly eliminated. Second, $b_t \in P$ can be decomposed to whether arm a_t's and/or b_t's feasibility has been determined and can be bounded similarly. In the case that the decision-maker chooses two arms a_t and b_t and the arms in $\mathcal{W}$ and arm $i^\star$ are not wrongly identified as infeasible arms, it is similar to a standard BAI problem.

Experiments

We now empirically compare our algorithm against the following four methods. For all approaches, we use the same confidence bound as in Equation (5).

	ours	F-first	P-first	TF-LUCB-C	Naive
a	1.0	0.58	3.09	1.77	1.81
b	1.0	3.13	0.84	1.98	4.75
c	1.0	4.00	3.47	2.78	4.06

Table 1: Results for Experiment 1: number of samples required relative to Algrithm 1, averaged over 10 runs. The standard deviations are reported in the Appendix.

1. Feasibility-First (F-first): for each arm i, we first use the subroutine SAMPLE-FEASIBILITY, which samples one constraint at each round, to determine whether it is feasible. Then, we apply LUCB algorithm (Kalyanakrishnan et al. 2012) to find the best arm in the feasible set.
2. Performance-First (P-first): We first run LUCB algorithm (Kalyanakrishnan et al. 2012) on all arms' performance and then test the best arm's feasibility using the subroutine SAMPLE-FEASIBILITY. If the best arm is feasible, it stops, otherwise it eliminates this arm and repeats the same procedure on the remaining arms.
3. TF-LUCB-C (Katz-Samuels and Scott 2019): This procedure was developed for feasible BAI, but assume the performance and feasibility constraints can be sampled together. We implement the most natural adaptation of it to our setting: pull all performance and feasibility distributions together using $N + 1$ samples.
4. Naive BAI (Naive): It adapts from the Racing algorithm (Even-Dar, Mannor, and Mansour 2002), which is optimal for BAI. On each round, it eliminates the arms that are considered infeasible or have lower performance than arms that are already considered feasible. It repeatedly tests the performance and all feasibility constraints of all survival arms. It stops when only one feasible arm is left or no arm is considered feasible.

Experiment 1: We conduct a series of experiments to demonstrate our algorithm can naturally adapt to the difficulty of the problem. We consider a setting where $K = 5$ and $N = 3$ and all distributions to be Gaussian with variance 1. We set $\delta = 0.1$ and change the means of these distributions to construct three types of problem instances. The settings are as follows:

1. $i^\star = 5$ and all other arms have a higher reward but are infeasible. The performance means of all arms are linearly spaced in $[0, 1]$ and the means of the feasibility distributions are $[0.75, 0.25, 0.25]$ for infeasible arms and $[0.25]^3$ for $i^\star$. In this setting, we expect F-first to perform well and P-first to perform poorly because P-first will spend unnecessary samples on comparing the performances between each arm while the F-first algorithm directly eliminates the non-optimal arms by infeasibility.
2. $i^\star = 1$ and all other arms are feasible but have a lower reward. For all arms, the performance means are linearly spaced in $[1, 0]$ and the feasibility constraints are $[0.4]^3$.
3. We consider a problem between the two extremes and contains multiple types of suboptimal arms. We set

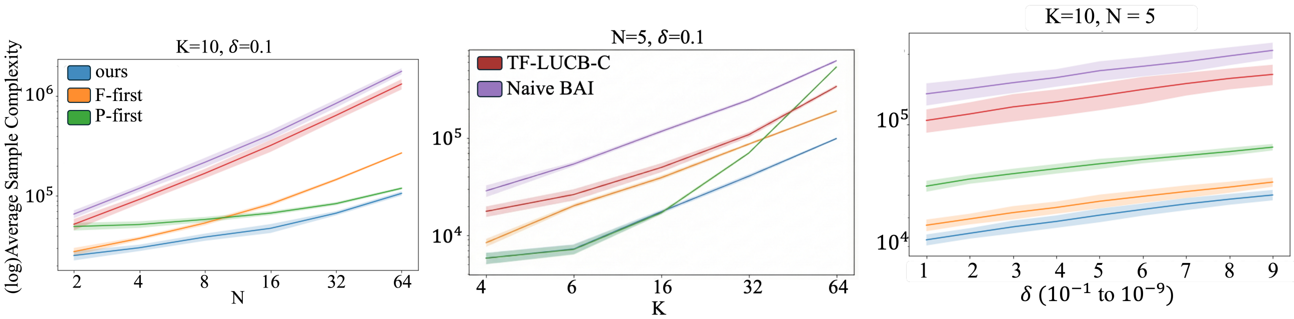

Figure 2: Results for Experiment 2. Results are averaged over 10 runs and the error bars are standard deviations.

$i^\star = 2$ and only arms 2 and 5 are feasible. The performance means are $[1, 0.9, 0.5, 0.25, 0]^T$. The feasibility constraints are: $[0.65, 0.4, 0.4]^T$ for arms $1, 3, 4$; $[0.3, 0.4, 0.4]^T$ for arm 2; $[0.45, 0.45, 0.45]^T$ for arm 5.

Synopsis for Experiment 1: Recall the key challenges, we expect F-first to perform the best on instance (a) but poorly on instance (b); and the opposite for P-first. The results in Table 1 match our expectations. Note that Algorithm 1 only performs slightly worse than the best algorithm on instances (a) and (b). Moreover, on instance (c) which contains multiple types of suboptimal arms, Algorithm 1 outperforms all other methods, showing that it can adapt to the difficulty of the problem.

Experiment 2: We compare the five methods when the number of arms, constraints, and δ change. We assume the distributions are Gaussian with variance 1. First, we set $K = 10$, $\delta = 0.1$, and increase N from 2 to 64. The performance means of all arms are linearly spaced in $[2, 0]$. Arms 4–7 are feasible arms and have feasibility constraints means to be $[0.25]^N$. The other arms are infeasible arms and have feasibility constraints to be $[0.75] \times [0.25]^{N-1}$. Second, we set $N = 5$, $\delta = 0.1$, and increase K from 4 to 64. The performance means of all arms are linearly spaced in $[10, 0]$. Arms indexes from $\lfloor \frac{K}{3} \rfloor$ to $\lfloor \frac{2K}{3} \rfloor$ are feasible arms whose feasibility constraints means are set to $[0.25]^N$. The other arms are infeasible arms and have feasibility constraint means to be $[0.75] \times [0.25]^{N-1}$. Third, we set $K = 10$, $N = 5$, and decrease δ from 10^{-1} to 10^{-9}. The mean values are set to the same when we change K. The results are given in Fig. 2. We see that our method achieves the best performance when compared to all other methods.

Drug Discovery: We investigate finding the most effective drug satisfying different feasibility constraints. The data comes from Tables 2 & 3 in Genovese et al. (2013). Each arm corresponds to a dosage level (in mg): [25, 75, 150, 300, Placebo] and has 3 attributes: probability of being effective $\mu_{i,0}$, probability of any adverse event $\mu_{i,1}$, and probability of causing an infection or infestation $\mu_{i,2}$. The threshold for constraint 1 is 0.5 and for constraint 2 is 0.25. We manually increase 0.25 to all rewards received from constraint 2 to make the thresholds equal. The means for different arms are $(.34, .519, .259)^T$, $(.469, .612, .184)^T$, $(.465, .465, .209)^T$, $(.537, .61, .293)^T$, and $(.36, .58, .16)^T$. We use Gaussian distribution with variance 1. Fig. 3 shows that our algorithm

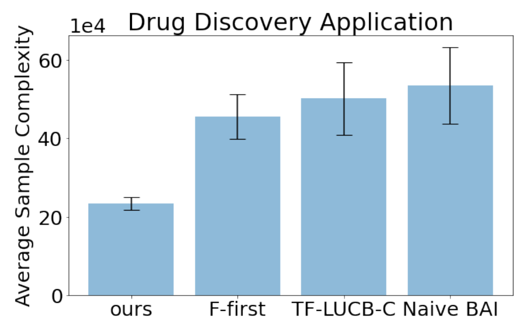

Figure 3: Drug discovery application: we exclude P-first due to poor performance. Results are averaged over 10 runs and the error bars are the standard deviations.

outperforms all other methods.

In all the experiments, all methods can find $i^\star$ correctly. Moreover, Algorithm 1, P-first, and F-first, which all use SAMPLE-FEASIBILITY, generally perform better than the other two methods which sample all constraints simultaneously, demonstrating that we should test each constraint individually. Since different confidence bounds will affect the performance of the algorithm (Jamieson et al. 2014), for a fair comparison, we use (5) on all methods. In the appendix, we compare Algorithm 1 and TF-LUCB-C with two different confidence bounds to see the influence of different confidence bounds. It shows that our algorithm outperforms TF-LUCB-C in both bounds and even when only TF-LUCB-C uses a tighter confidence bound.

Conclusion

In this work, we consider a new formalism for best arm identification (BAI) in the fixed confidence setting when the optimal arm needs to satisfy multiple given feasibility constraints, and the feasibility constraints can be tested separately. We quantify the complexity of this problem by considering the relative difficulty of deciding if a suboptimal arm is easier to eliminate via its performance or its feasibility. We propose a δ-correct algorithm and upper bound its expected sample complexity, showing that it is asymptotically optimal compared to the lower bound. Finally, we conduct empirical experiments showing our algorithm outperforms other natural baselines for this problem.

References

Bechhofer, R. E. 1958. A sequential multiple-decision procedure for selecting the best one of several normal populations with a common unknown variance, and its use with various experimental designs. *Biometrics*, 14(3): 408–429.

Berkenkamp, F.; Krause, A.; and Schoellig, A. P. 2021. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 1–35.

Bubeck, S.; Munos, R.; and Stoltz, G. 2009. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, 23–37. Springer.

Chen, L.; Gupta, A.; Li, J.; Qiao, M.; and Wang, R. 2017. Nearly optimal sampling algorithms for combinatorial pure exploration. In *Conference on Learning Theory*, 482–534. PMLR.

Degenne, R.; and Koolen, W. M. 2019. Pure exploration with multiple correct answers. *Advances in Neural Information Processing Systems*, 32.

Eriksson, D.; and Poloczek, M. 2021. Scalable constrained bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, 730–738. PMLR.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2002. PAC bounds for multi-armed bandit and Markov decision processes. In *International Conference on Computational Learning Theory*, 255–270. Springer.

Gardner, J. R.; Kusner, M. J.; Xu, Z. E.; Weinberger, K. Q.; and Cunningham, J. P. 2014. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, 937–945.

Genovese, M. C.; Durez, P.; Richards, H. B.; Supronik, J.; Dokoupilova, E.; Mazurov, V.; Aelion, J. A.; Lee, S.-H.; Codding, C. E.; Kellner, H.; et al. 2013. Efficacy and safety of secukinumab in patients with rheumatoid arthritis: a phase II, dose-finding, double-blind, randomised, placebo controlled study. *Annals of the rheumatic diseases*, 72(6): 863–869.

Hernández-Lobato, D.; Hernandez-Lobato, J.; Shah, A.; and Adams, R. 2016. Predictive entropy search for multi-objective bayesian optimization. In *International conference on machine learning*, 1492–1501. PMLR.

Hernández-Lobato, J. M.; Gelbart, M. A.; Adams, R. P.; Hoffman, M. W.; and Ghahramani, Z. 2016. A General Framework for Constrained Bayesian Optimization Using Information-Based Search. *J. Mach. Learn. Res.*, 17(1): 5549–5601.

Jamieson, K.; Malloy, M.; Nowak, R.; and Bubeck, S. 2014. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, 423–439. PMLR.

Jun, K.-S.; Jamieson, K.; Nowak, R.; and Zhu, X. 2016. Top arm identification in multi-armed bandits with batch arm pulls. In *Artificial Intelligence and Statistics*, 139–148. PMLR.

Kalyanakrishnan, S.; Tewari, A.; Auer, P.; and Stone, P. 2012. PAC subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, 655–662.

Kanellis, K.; Alagappan, R.; and Venkataraman, S. 2020. Too many knobs to tune? towards faster database tuning by pre-selecting important knobs. In *Workshop on Hot Topics in Storage and File Systems*.

Kano, H.; Honda, J.; Sakamaki, K.; Matsuura, K.; Nakamura, A.; and Sugiyama, M. 2019. Good arm identification via bandit feedback. *Machine Learning*, 108: 721–745.

Karnin, Z.; Koren, T.; and Somekh, O. 2013. Almost optimal exploration in multi-armed bandits. In *International Conference on Machine Learning*, 1238–1246. PMLR.

Karnin, Z. S. 2016. Verification based solution for structured mab problems. *Advances in Neural Information Processing Systems*, 29.

Katz-Samuels, J.; and Jamieson, K. 2020. The true sample complexity of identifying good arms. In *International Conference on Artificial Intelligence and Statistics*, 1781–1791. PMLR.

Katz-Samuels, J.; and Scott, C. 2018. Feasible arm identification. In *International Conference on Machine Learning*, 2535–2543. PMLR.

Katz-Samuels, J.; and Scott, C. 2019. Top feasible arm identification. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 1593–1601. PMLR.

Kaufmann, E.; Cappé, O.; and Garivier, A. 2016. On the complexity of best-arm identification in multi-armed bandit models. *The Journal of Machine Learning Research*, 17(1): 1–42.

Kaufmann, E.; Koolen, W. M.; and Garivier, A. 2018. Sequential test for the lowest mean: From Thompson to Murphy sampling. *Advances in Neural Information Processing Systems*, 31.

Kirschner, J.; Mutny, M.; Hiller, N.; Ischebeck, R.; and Krause, A. 2019. Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces. In *International Conference on Machine Learning*, 3429–3438. PMLR.

Kirschner, J.; Mutnỳ, M.; Krause, A.; de Portugal, J. C.; Hiller, N.; and Snuverink, J. 2022. Tuning Particle Accelerators with Safety Constraints using Bayesian Optimization. *arXiv preprint arXiv:2203.13968*.

Koenig, L. W.; and Law, A. M. 1985. A procedure for selecting a subset of size m containing the l best of k independent normal populations, with applications to simulation. *Communications in Statistics-Simulation and Computation*, 14(3): 719–734.

Lai, T. L.; and Robbins, H. 1985. Asymptotically Efficient Adaptive Allocation Rules. *Adv. in App. Math.*

Locatelli, A.; Gutzeit, M.; and Carpentier, A. 2016. An optimal algorithm for the thresholding bandit problem. In *International Conference on Machine Learning*, 1690–1698. PMLR.

Mannor, S.; and Tsitsiklis, J. N. 2004. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun): 623–648.

Mason, B.; Jain, L.; Tripathy, A.; and Nowak, R. 2020. Finding All \epsilon-Good Arms in Stochastic Bandits. In

Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 20707–20718. Curran Associates, Inc.

Paria, B.; Kandasamy, K.; and Póczos, B. 2020. A flexible framework for multi-objective bayesian optimization using random scalarizations. In *Uncertainty in Artificial Intelligence*, 766–776. PMLR.

Paulson, E. 1964. A sequential procedure for selecting the population with the largest mean from k normal populations. *The Annals of Mathematical Statistics*, 174–180.

Schmidt, C.; Branke, J.; and Chick, S. E. 2006. Integrating techniques from statistical ranking into evolutionary algorithms. In *Applications of Evolutionary Computing*, 752–763. Springer.

Simchowitz, M.; Jamieson, K.; and Recht, B. 2017. The simulator: Understanding adaptive sampling in the moderate-confidence regime. In *Conference on Learning Theory*, 1794–1834. PMLR.

Thall, P. F.; and Russell, K. E. 1998. A strategy for dose-finding and safety monitoring based on efficacy and adverse outcomes in phase I/II clinical trials. *Biometrics*, 251–264.

Thompson, W. R. 1933. On the Likelihood that one Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*.

Ungredda, J.; and Branke, J. 2021. Bayesian optimisation for constrained problems. *arXiv preprint arXiv:2105.13245*.

Van Aken, D.; Pavlo, A.; Gordon, G. J.; and Zhang, B. 2017. Automatic database management system tuning through large-scale machine learning. In *Proceedings of the 2017 ACM international conference on management of data*, 1009–1024.

Wang, Z.; Wagenmaker, A. J.; and Jamieson, K. 2022. Best Arm Identification with Safety Constraints. In *International Conference on Artificial Intelligence and Statistics*, 9114–9146. PMLR.

Zhou, Y.; Chen, X.; and Li, J. 2014. Optimal PAC multiple arm identification with applications to crowdsourcing. In *International Conference on Machine Learning*, 217–225. PMLR.