The Diverse Cohort Selection Problem: Multi-Armed Bandits with Varied Pulls

Candice Schumann[†] and **Samsara N. Counts**[‡] and **Jeffrey S. Foster**[†] and **John P. Dickerson**[†]
[†]University of Maryland, College Park [‡]George Washington University

Abstract

How should a firm allocate its limited interviewing resources to select the optimal cohort of new employees from a large set of job applicants? How should that firm allocate cheap but noisy résumé screenings and expensive but in-depth in-person interviews? We view this problem through the lens of combinatorial pure exploration (CPE) in the multi-armed bandit setting, where a central learning agent performs costly exploration of a set of arms before selecting a final subset with some combinatorial structure. We generalize a recent CPE algorithm to the setting where arm pulls can have different cost, but return different levels of information, and prove theoretical upper bounds for a general class of arm-pulling strategies in this new setting. We then apply our general algorithm to a real-world problem with combinatorial structure: incorporating diversity into university admissions. We take real data from admissions at one of the largest US-based computer science graduate programs and show that a simulation of our algorithm produced more diverse student cohorts at low cost to individual student quality, and does so by spending comparable budget to the current admissions process at that university.

1 Introduction

How should a firm, school, or fellowship committee allocate its limited interviewing resources to select the optimal cohort of new employees, students, or awardees from a large set of applicants? Here, the central decision maker must first form a belief about the true quality of an applicant via costly information gathering, and then select a subset of applicants that maximizes some objective function. Furthermore, various types of information gathering can be performed—reviewing a résumé, scheduling a Skype interview, flying a candidate out for an all-day interview, and so on—to gather greater amounts of information, but also at greater cost.

Complicating matters further, the total utility of a cohort may not be the simple sum of its parts. For example, with a cohort of size two, a software engineering firm may value two Java developers less than a single Java developer and a web developer. That is, applicants with otherwise similar skillsets or backgrounds may yield diminishing marginal gain as more are added to a cohort. Optimizing with a nonlinear objective function—in this case, a submodular function—presents additional complexity, even assuming perfect knowledge of the applicants' underlying true utilities.

In this paper, we model the allocation of interviewing resources and subsequent selection of a cohort as a combinatorial pure exploration problem in the multi-armed bandit (MAB) setting. Here, each application is an arm, and a decision maker can *pull* the arm, at some cost, to receive a noisy signal about the underlying quality of that application. We further model different levels of interviews as *strong* and *weak* pulls—the former costing more to perform than the latter, but also resulting in a less noisy signal. We introduce the strong-weak arm-pulls (SWAP) algorithm, generalizing an algorithm by Chen et al. (2014), and provide theoretical upper bounds for a general class of our varied arm-pull strategies. To complement these asymptotic bounds, we provide simulation results comparing a variety of pulling strategies on a toy problem that mimics our theoretical assumptions.

We then validate our proposed method on a real-world scenario: accepting a diverse cohort of graduate students. We take recent data from one of the largest computer science graduate programs in the United States—applications including recommendation letters, statements of purpose, transcripts, and other information, as well as the department's reviews of applications and final admissions decisions—and run experiments comparing our algorithm's performance under a variety of assumptions to reviews and decisions made in reality. We find that our simulation of SWAP increased a diversity score (over gender and region of origin) with little loss in fit using roughly the same amount of resources as in practice. This gain demonstrates that SWAP can serve as a useful decision support tool to promote diversity in practice.

2 Related Work

The multi-armed bandit (MAB) problem is a classical setting for modeling sequential decision making; for a general overview of historical research, we direct the reader to Bubeck, Cesa-Bianchi, and others (2012).

Our work builds on contributions by Chen et al. (2014), who propose a general algorithm that relies on an oracle for insight into the true utility of a subset of arms, where subsets have some combinatorial structure. Chen et al. propose two algorithms: Combinatorial Lower-Upper Confidence Bound (CLUCB), for the fixed confidence setting, and Combinatorial Successive Accept Reject (CSAR), for the fixed budget and batch setting with variable batch sizes, detailed in Sec-

tion 4 and Appendix B. We adapt the CLUCB algorithm for the cohort selection setting with varying types of pulls to reflect different information gathering techniques.

All of the following MAB formulations select some high-utility subset of arms using a *single* type of arm pull, with varying ways of modeling decisions focusing on different problem features. In the Top-K problem, the goal is to select a subset of k probability distributions with highest means from a set of n total distributions, which Cao et al. (2015) study in the MAB setting. Their linear objective is less general than the setting in which we and Chen et al. (2014) operate. Locatelli, Gutzeit, and Carpentier (2016) address the thresholding bandit problem, where the aim is to identify the arms above and below threshold τ with precision ϵ. Jun et al. (2016) address the MAB problem with batch arm pulls, aiming to identify the top k-set from n arms while pulling arms in batches of size b. Singla et al. (2015) propose an algorithm that hires a team of workers for specific tasks from a crowdsourcing point of view, treating different types of workers as separate problems and arm-pulls as asking a worker to perform an action with uniform cost. Modeling workers and tasks in separate graphs, their algorithms take advantage of known side-observations between the graphs to inform optimal team selection. Thus, their algorithm can operate in both the *bandit setting* (no side observations) and the *information setting* (fully connected graphs). Our algorithm focuses solely on the bandit setting.

For the task of selecting the best subset while satisfying a submodular function, Singla, Tschiatschek, and Krause (2016) propose an algorithm that maximizes an unknown submodular function accessed through noisy evaluations, in comparison to SWAP maximizing a given submodular function. Yue and Guestrin (2011) introduce the linear submodular bandits problem to select diverse sets of content in an online learning setting for optimizing a class of feature-rich submodular utility models. Unlike our focus on self-reported, concrete features to prioritize diversity, they categorize arms based on the information they probabilistically contain, determined by a topic model like LDA, which is potentially noisy and biased. Their notion of *diverse*, similar to that of Radlinski, Kleinberg, and Joachims (2008), comes from the diversified retrieval setting, meaning they seek to avoid redundant coverage of topics in the set of arms they select. Radlinski, Kleinberg, and Joachims (2008) learn a diverse ranking from the behavior patterns of different users and then use a greedy algorithm to select the next document to rank. They treat each rank of documents as a separate MAB instance, rather than our approach using a single MAB to model the whole system.

We are motivated by the observation that, in many real-world settings, different levels of information gathering can be performed at different costs. Previous work has used stochastic, varying costs in the MAB setting; however, our costs are fixed for specific types of arm pulls. Ding et al. (2013) look at a MAB problem with variable rewards and cost with budget constraints. When an arm is pulled, a random reward is received and a random cost is taken from the budget—discovered only after pulling it. Over time they aim to maximize total rewards and minimize regret while complying with the pull budget. Similarly, Xia et al. (2016) propose a batch-arm-pull MAB solution to a problem with variable, random rewards and costs. Jain et al. (2014) frame the MAB problem with variable reward and cost in the setting of crowdsourcing. They select workers to perform binary tasks while achieving an assured accuracy for each task, where the costs of workers are unknown. This differs from our formulation in that we decide how many resources to use when pulling an arm.

Other authors have used standard supervised learning techniques to model higher education admissions decisions. However, previous work focuses on developing an accurate admissions classifier; none model how to *select* an optimal cohort in a novel way. Furthermore, they do not try to satisfy a certain objective, unlike our aim to select a more $diverse$ cohort. Lux et al. (2016) and Waters and Miikkulainen (2013) work alongside the admissions office, motivated by the desire to optimize the admissions process. Gupta, Sawhney, and Roth (2016) model the graduate admissions process from the student's perspective, to inform prospective applicants on which schools to apply to based on their likelihood of admission.

3 Problem Formulation

We now formally describe the stochastic multi-armed bandit setting in which we operate. For exposition's sake, we do so in the context of a decisionmaker reviewing a set of job applicants; yet, the formulation itself maintains full generality.

We represent a set of n applications as arms $a_i \in A$ for $i \in [n]$. Each arm has a true utility, $u(a_i) \in [0, 1]$, which is unknown, an empirical estimate $\hat{u}(a_i) \in [0, 1]$ of that underlying true utility, and an uncertainty bound $\text{rad}(a_i)$. Once arm a_i is pulled (e.g., application reviewed or applicant interviewed), $\hat{u}(a_i)$ and $\text{rad}(a_i)$ are updated.

The set of potential *cohorts*, or subsets of arms, is defined by a decision class $\mathcal{M} \subseteq 2^{[n]}$. Note that this need not be the power set of arms, but can include cardinality and other constraints. The total utility for a cohort is given by some monotone, submodular function $w : \mathcal{M} \to \mathbb{R}$ and takes as input the (unknown) true utilities $u(\cdot)$ of the constituent arms in the cohort. Thus, our overall goal is to accurately estimate the true utilities of arms and then select the optimal subset of arms $M^* = \arg\max_{M \in \mathcal{M}} w(M)$.

Throughout the paper, we assume a maximization oracle—that is, we do not focus on the combinatorial optimization problem of solving for M^* given information about the arms' utilities. Instead, we focus on the arm pulling policies that reveal estimates of the arms' utilities. Formally, the oracle can be written as

$$\text{Oracle}(\mathbf{v}) = \arg\max_{M \in \mathcal{M}} w(M), \tag{1}$$

where $\mathbf{v} \in \mathbb{R}^n$ is vector of weights—in this case, estimated or true utilities for each arm. In Section 5.1, we give an example submodular oracle based on a notion of diversity; other examples include linear and set maximum.

Following the notation of Chen et al. (2014), we can define a *gap* score for each arm. For each arm a that is in the optimal set M^*, the gap is the difference in optimality between M^* and the best set without arm a. For each

arm a that is not in the optimal set M^*, the gap is the suboptimality of the best set that includes arm a. Formally the gap is defined as

$$\Delta_a = \begin{cases} w(M^*) - \max_{M \in \mathcal{M}: a \in M} w(M), & \text{if } a \notin M^* \\ w(M^*) - \max_{M \in \mathcal{M}: a \notin M} w(M), & \text{if } a \in M^*. \end{cases} \tag{2}$$

This gap score serves as a useful signal for problem hardness, which we use in our theoretical analysis in Section 4. Formally, the hardness of the problem can be defined as the sum of inverse squared gaps

$$\mathbf{H} = \sum_{a \in A^T} \Delta_a^{-2}. \tag{3}$$

In reality, there is more than one way to gather information or receive rewards; therefore, we introduce two kinds of arm pulls. A *weak arm pull* has normalized cost 1 but results in a small amount of information. In our domain of graduate admissions, weak arm pulls are standard application reviews, which involve reading submitted materials and then making a recommendation. A *strong arm pull*, in contrast, has cost $j \geq 1$, but results in $s \geq 1$ times the information as a weak arm pull, modeled as s parallel pulls of an arm. The values s and j are parameters given to the SWAP algorithm. In our domain, strong arm pulls combine reading submitted materials with a Skype interview, followed by note-taking and a recommendation.

In our experience, the latter can reduce uncertainty considerably, which we quantify and discuss in Section 5. However, due to their high cost, such interviews are allocated relatively sparingly. We explore this problem in a formal setting in Section 4 and provide a formal algorithm for selecting which arms to pull, along with asymptotic upper bounds on total cost.

4 Strong Weak Arm Pulls

In this section, we propose a new multi-armed bandit algorithm, strong-weak arm-pulls (SWAP), that is parameterized by s and j and uses a combination of strong and weak arm pulls to gain information about arms. Our setting and the algorithm we present generalize the CLUCB algorithm proposed by Chen et al. (2014), which can be viewed as a special case with $s = j = 1$. We present the algorithm in generality here, and then instantiate it with a specific objective (a submodular diversity-promoting function) in the following section.

Algorithm 1 gives pseudocode for SWAP. It starts by weak pulling all arms once to initialize an empirical estimate of the true underlying utility of each arm. It then iteratively pulls arms, selects to weak or strong pull based on a general strategy, updates empirical estimates of arms, and terminates with the optimal (i.e., objective-maximizing) subset of arms with probability $1 - \delta$, for some user-supplied parameter δ.

During each iteration t, the algorithm starts by finding the set of arms M_t that, according to current empirical estimates of their means, maximizes the objective function; this is done via an oracle. It then computes a confidence radius ($\text{rad}_t(a)$) for each arm and estimates the worst-case utility of that arm with some corresponding bound. If an arm a is in the set M_t then the worst case is when the true utility of a is less than our estimate (the oracle may not have chosen arm a). Alternatively, if an arm is not in the set M_t then the worst case is when the true utility of a is greater than our estimate (the oracle may have chosen arm a). Using the worst-case estimates instead of the empirical means of the arms, we compute an alternate subset of arms $\tilde{M}_t$. If the utility of the initial set M_t and the worst-case set $\tilde{M}_t$ are equal then the algorithm terminates with output M_t, which is correct with probability $1 - \delta$ using similar reasoning to that found in Chen et al. (2014). If they differ, the algorithm looks at a set of candidate arms in the symmetric difference of M_t and $\tilde{M}_t$, and chooses the arm p_t with the largest radius—that is, the largest uncertainty bound.

The algorithm then chooses to either strong or weak pull the selected arm p_t, decided using a *strong pull policy* depending on value parameter s and cost parameter j. A strong pull policy is defined as $spp : \mathbb{R} \times \mathbb{R} \to [0, 1]$. For example, in the experiments in Section 5, we use the following pull policy:

$$spp(s, j) = \frac{s - j}{s - 1}. \tag{4}$$

Notice that this pull policy performs *only* strong pulls when $j = 1$ and $s > j$, and performs *only* weak pulls when $j = s$. For other values of s and j, it probabilistically chooses a type of arm pull.

We define $\bar{X}_{Cost} = E[Cost]$ as the expected cost and $\bar{X}_{Gain} = E[Gain]$ as the expected gain. Assume that each arm $a \in [n]$ has mean $u(a)$ with an σ-sub-Gaussian tail. Following Chen et al. (2014) set $\text{rad}_t(a) = \sigma\sqrt{2 \log\left(\frac{4nCost_t^3}{\delta}\right) / T_t(a)}$ for all $t > 0$ and $a \in [n]$.

Notice that if we use strong pull policy $spp(s, j) = 0$ then we only perform weak arm pulls, and Algorithm 1 reduces to Chen et al. (2014)'s CLUCB. We call this reduction the *weak only pull problem*. Chen et. al. proved that CLUCB returns the optimal set M_* and uses at most $\tilde{O}(\text{width}(\mathcal{M})^2\mathbf{H})$ samples.

Theorem 1 (Chen et al. 2014)**.** *Given any $\delta \in (0, 1)$, any decision class $\mathcal{M} \subseteq 2^{[n]}$, and any expected rewards $\mathbf{w} \in \mathbb{R}^n$, assume that the reward distribution φ_a for each arm $a \in [n]$ has mean $w(a)$ with an σ-sub-Gaussian tail. Let $M_* = \arg\max_{M \in \mathcal{M}} w(M)$ denote the optimal set. Set $rad_t(a) = \sigma\sqrt{2 \log\left(\frac{4nt^3}{\delta} / T_t(a)\right)}$ for all $t > 0$ and $a \in [n]$. Then, with probability at least $1 - \delta$, the SWAP algorithm with only weak pulls returns the optimal set $\texttt{Out} = M_*$ and*

$$T \leq O\left(\sigma^2 \, width(\mathcal{M})^2 \mathbf{H} \log(nR^2\mathbf{H}/\delta)\right) \tag{5}$$

where T denotes the number of samples used by the SWAP algorithm, $\mathbf{H}$ is defined in Eq.3.

Similarly, if we set $spp(s, j) = 1$ then we only perform strong arm pulls (Strong Only Pull Problem). We show that this version of SWAP returns the optimal set M_* and uses at most $\tilde{O}(\text{width}(\mathcal{M})^2\mathbf{H}/s)$ samples. The proof of the following theorem can be found in the appendix.

Algorithm 1 Strong Weak Arm Pulls (SWAP)

Require: Confidence $\delta \in (0,1)$; Maximization oracle: Oracle$(\cdot) : \mathbb{R}^n \to \mathcal{M}$

Weak pull each arm $a \in [n]$ once.
Initialize empirical means $\bar{\mathbf{u}}_n$
$\forall a \in [n]$ set $T_n(a) \leftarrow 1$, Total information gain for arm a
$Cost_n \leftarrow n$, Total resources spent
for $t = n, n+1, \ldots$ **do**
  $M_t \leftarrow$ Oracle$(\bar{\mathbf{u}}_t)$
  $\forall a \in [n]$ compute confidence radius $\text{rad}_t(a)$
  for $a = 1, \ldots, n$ **do**
    if $a \in M_t$ **then** $\tilde{u}_t(a) \leftarrow \bar{u}_t(a) - \text{rad}_t(a)$
    else $\tilde{u}_t(a) \leftarrow \bar{u}_t(a) + \text{rad}_t(a)$
  $\tilde{M}_t \leftarrow$ Oracle$(\tilde{\mathbf{u}}_t)$
  if $\tilde{w}(\tilde{M}_t) = \tilde{w}(M_t)$ **then**
    Out $\leftarrow M_t$
    return Out
  $p_t \leftarrow \arg\max_{a \in (\tilde{M}_t \setminus M_t) \cup (M_t \setminus \tilde{M}_t)} \text{rad}_t(a)$
  $\alpha \leftarrow spp(s, j)$
  with probability α **do**
    Strong pull p_t
    $T_{t+1}(p_t) \leftarrow T_t(p_t) + s$
    $Cost_{t+1} \leftarrow Cost_t + j$
  else
    Weak pull p_t
    $T_{t+1}(p_t) \leftarrow T_t(p_t) + 1$
    $Cost_{t+1} \leftarrow Cost_t + 1$
  Update empirical mean $\bar{\mathbf{u}}_{t+1}$ using observed reward
  $T_{t+1} \leftarrow T_t(a)\ \forall a \neq p_t$

Theorem 2. *Given any $\delta \in (0,1)$, any decision class $\mathcal{M} \subseteq 2^{[n]}$, and any expected rewards $\mathbf{w} \in \mathbb{R}^n$, assume that the reward distribution φ_a for each arm $a \in [n]$ has mean $w(a)$ with an σ-sub-Gaussian tail. Let $M_* = \arg\max_{M \in \mathcal{M}} w(M)$ denote the optimal set. Set $rad_t(a) = \sigma\sqrt{2\log\left(\frac{4nt^3j^3}{\delta}/T_t(a)\right)}$ for all $t > 0$ and $a \in [n]$. Then, with probability at least $1-\delta$, the SWAP algorithm with only strong pulls where $j \geq 1$ and $s > j$ returns the optimal set $\texttt{Out} = M_*$ and*

$$T \leq O\left(\frac{\sigma^2\, width(\mathcal{M})^2 \mathbf{H} \log(nj^3R^2\mathbf{H}/\delta)}{s}\right) \quad (6)$$

where T denotes the number of samples used by the SWAP algorithm, $\mathbf{H}$ is defined in Eq.3.

Although s and j are problem-specific, it is important to know when to use the strong only pull problem over the weak only pull problem. Corollary 2.1 provides weak bounds for s and j for the strong only pull problem.

Corollary 2.1. *SWAP with only strong pulls is equally or more efficient than SWAP with only weak pulls when $s > 0$ and $0 < j \leq C^{\frac{s}{3}-\frac{1}{3}}$ where $C = 4n\tilde{\mathbf{H}}/\delta$.*

Proof.

$$T_{strong} \leq T_{weak}$$

$$\frac{499\tilde{\mathbf{H}}\log(4nj^3\tilde{\mathbf{H}}/\delta)}{s} + 2n \leq 499\tilde{\mathbf{H}}\log(4nj^3\tilde{\mathbf{H}}/\delta) + 2n$$

$$\frac{\log(Cj^3)}{s} \leq \log(C) \quad (7)$$

Solving for Eq.7 we get $s > 0$ and $0 < j \leq C^{\frac{s}{3}-\frac{1}{3}}$. □

We now address the general case of SWAP, for any probabilistic strong pull policy parameterized by s and j. For this general case, we show in Theorem 3 that SWAP returns M_* in $\tilde{O}\left(\text{width}(\mathcal{M})^2\mathbf{H}/\bar{X}_{Gain}\right)$ samples. (The full proof can be found in Appendix C.)

Theorem 3. *Given any $\delta_1, \delta_2, \delta_3 \in (0,1)$, any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any expected rewards $\mathbf{w} \in \mathbb{R}^n$, assume that the reward distribution φ_a for each arm $a \in [n]$ has mean $w(a)$ with an σ-sub-Gaussian tail. Let $M_* = \arg\max_{M \in \mathcal{M}} w(M)$ denote the optimal set. Set $rad_t(a) = \sigma_1\sqrt{2\log\left(\frac{4nCost_t^3}{\delta}/T_t(a)\right)}$ for all $t > 0$ and $a \in [n]$, set $\epsilon_1 = \sigma_2\sqrt{2\log\left(\frac{1}{2}\delta_2/T\right)}$, and set $\epsilon_2 = \sigma_2\sqrt{2\log\left(\frac{1}{2}\delta_3/n\right)}$. Then, with probability at least $(1-\delta_1)(1-\delta_2)(1-\delta_3)$, the SWAP algorithm (Algorithm 1) returns the optimal set $\texttt{Out} = M_*$ and*

$$T \leq O\left(\frac{\sigma_1^2\, width(\mathcal{M})^2\mathbf{H}\log\left(nR^2\left(\bar{X}_{Cost} - \epsilon_1\right)^3\mathbf{H}/\delta\right)}{\bar{X}_{Gain} - \epsilon_2}\right), \quad (8)$$

where T denotes the number of samples used by Algorithm 1, $\mathbf{H}$ is defined in Eq. 3 and width$(\mathcal{M})$ is defined by Chen et. al (Chen et al. 2014).

Finding where the general version of SWAP is better than both the SWAP algorithm with only strong pulls and the SWAP algorithm with only weak pulls is nontrivial, given the asymptotic nature of all three bounds (Theorems 1, 2, and 3). Based on our experiments (§5), we conjecture that there is an optimal zone of s and j pairs where SWAP is the optimal algorithm, even for relatively low numbers of arm pulls, but this is problem specific.

5 Experiments

In this section, we experimentally validate the SWAP algorithm under a variety of arm pull strategies. We begin by concretely instantiating a submodular objective function (§5.1) that promotes *diversity* in the final cohort selected by the algorithm. Then, we explore (§5.2) in simulation the efficacy of our bounds in Theorem 3 and Corollary 2.1. Finally, we deploy SWAP on real data (§5.3) drawn from one of the largest computer science graduate programs in the United States. We show a significant increase in diversity with little loss in fit while using roughly the same amount of resources as used in practice.

5.1 Formally Promoting the Diversity of a Cohort

The SWAP algorithm presented in Section 4 is defined in the context of a generic submodular and monotone oracle; in practice, domain experts would specify an ap-

propriate function to maximize. Motivated by recent evidence that diversity in the workforce can increase productivity (Hunt, Layton, and Prince 2015; Desrochers 2001), in this section, we explore the effect of formally promoting diversity in the cohort selection problem.

Quantifying the diversity of a set of elements is of interest in a variety of fields, including recommender systems, information retrieval, computer vision, and others (Liu, Suel, and Memon 2014; Qin and Zhu 2013; Ashkan et al. 2015; Sha, Wu, and Niu 2016; Radlinski, Kleinberg, and Joachims 2008; Ahmed, Dickerson, and Fuge 2017). For our experiments, we choose a recent formalization from Lin and Bilmes (2011) and apply it to both simulated and real data. At a high level their diversity function f is defined as

$$f(M) = \sum_{i=1}^{K} \sqrt{\sum_{j \in P_i \cap M} u(a)}, \tag{9}$$

where $u(a) \geq 0$ and $M = P_1 \cup P_2 \cup \ldots \cup P_K$. Lin and Bilmes showed that f is submodular and monotone. Nemhauser, Wolsey, and Fisher (1978) proved that a close to optimal $\left(f(S^*) \geq \left(1 - \frac{1}{e}\right) \texttt{OPT}\right)$ greedy algorithm exists for submodular, monotone functions that are subject to a cardinality constraint. We use this algorithm in our implementation of the diverse oracle.

5.2 Gaussian Arm Experiment

We begin by validating the tightness of our theoretical results in a simulation setting that mimics the assumptions made in Section 4. We pull from a Gaussian distribution around each arm. When arm a is weak pulled, a reward is pulled from a Gaussian distribution with mean u_a, the arm's true utility, and standard deviation of σ. Similarly, when arm a is strong pulled, the algorithm is charged j cost, and a reward is pulled from a distribution with mean u_a and standard deviation of $\sigma/\sqrt{s}$. This strong pull distribution is equivalent to pulling the arm s times and averaging the reward, thus ensuring an information gain of s.

We ran all three algorithms—SWAP with the strong pull policy defined in Equation 4, SWAP with only strong pulls, and SWAP with only weak pulls—while varying s and j. For each s and j pair we ran the algorithms at least 4000 times on a randomly generated initialization of arm values. Random seeds were maintained across policies. We then compared the cost of running each of the algorithms to completion.[1]

The first comparison was of SWAP with only weak pulls and SWAP with only strong pulls; a heat map of performance is given as Figure 1. We found that Corollary 2.1 is a weak bound on the boundary value of j. Yet, SWAP with only strong pulls has a lower cost than SWAP with only weak pulls for values greater than $j < (4n\,\tilde{\mathbf{H}}\,/\delta)^{\frac{s}{3}-\frac{1}{3}}$, given this experiment.

The general version of SWAP should be used when it performs better—costs less—than both the strong only and weak only versions of SWAP. The zone where SWAP is effective varies with the problem. Figure 2 shows the optimal zone for the Gaussian Arm Experiment.

[1] All code to replicate this experiment will be posted after the double-blind review period concludes.

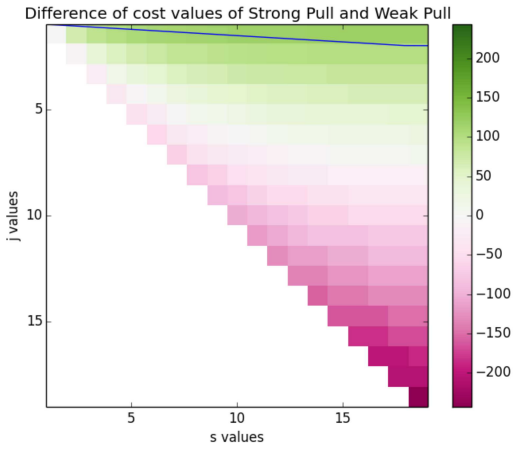


Figure 1: Heat map comparing only strong to only weak pulls in the Gaussian setting (§5.2). Green indicates better performance by strong pulls, and intensity indicates magnitude. The blue line is the bound on j given Corollary 2.1.

5.3 Graduate Admissions Experiment

Now, we describe a preliminary exploration of SWAP on real graduate admissions data from one of the largest CS graduate programs in the United States. The experiment was approved by the University's Institutional Review Board. Our dataset consists of three years of graduate admissions applications, graduate committee application review text and ratings, and final admissions decisions. Information was gathered from the first two academic years (treated as a training set), while the data from last academic year was used to evaluate the performance of SWAP (treated as a test set).

Dataset. During the admissions process, potential students from all over the world send in their applications. A single application consists of quantitative information such as GPA, GRE scores, TOEFL scores, nationality, sex, previous degrees and so on, as well as qualitative information in the form of recommendation letters and statements of purpose. In the most recent academic year we received approximately 1600 applications, with roughly 4500 applications over all three years. The most recent 1600 applications are roughly split into 1000 applicants seeking Masters Degrees and 600 applicants seeking Ph.D. Degrees. The acceptance rate for Masters students is about 3%, while the acceptance rate for Ph.D. students is about 20%. In the most recent academic year 24% of the applicants were female and 26% of the applicants admitted were female.

Once all of the applications are submitted, they are sent to a review committee of faculty and current graduate students at the university. Each application is reviewed at least once; for our dataset, each application received an average of 1.21 reviews. Generally, applicants at the top (who far exceed ex-

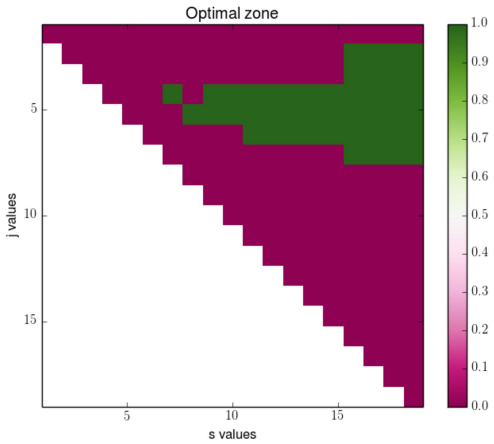


Figure 2: Heat map showing where the general version of SWAP outperformed (green) both SWAP with only strong pulls as well as SWAP with only weak pulls, and (maroon) where at least one of the latter outperformed.

pectations) and applicants at the bottom (who do not fulfill the program's strict requirements) only need one review.

The majority of reviews are quick looks at the application, equivalent to weak arm pulls. Some of the reviews are one-on-one interviews, equivalent to strong arm pulls. We estimate that our interviews are approximately six times longer than a quick review therefore we set our j value (the cost of a strong pull) to be 6. The gain of an interview is uncertain so we ran tests over a wide range of s values (the information gain of a strong pull). Once all reviews have been made the graduate director decides on the final applicants to admit.

Experimental Setup. We simulate an arm pull by returning a real score that a reviewer gave during the admissions process (first) or a score from a probabilistic classifier (second, if all committee members' reviews have been used already). An arm pull returns a score drawn from a distribution around the probabilistic result from the classifier. The distribution is structured similarly to the Gaussian arm experiment. This is done in order to simulate some human error or bias.

The classifier used is a random forest (Pedregosa et al. 2011) trained with 3-fold cross validation on the data from the two academic years prior to the most recent academic year. We used a number of features including general features found on the application (GPA, GRE scores, area of interest, sex, region, etc.) and features generated from the texts of the recommendation letters. Features pulled from the recommendation letters were letter length and word counts from the word groups discussed by Schmader, Whitehead, and Wysocki (2007) created from Chemistry and Biochemistry recommendation letters. We found that these word groups are general and translated well to Computer Science students.

Limitations. In this experiment, we assume that the true utility of an applicant can be modeled by our classifier,

	General	Diversity	M/F ratio
SWAP	8.5 (0.03)	12.0 (0.04)	1.3:1 (0.02)
Actual	8.6	11.8	2.9:1

(a) Gender Diversity Gain

	General	Diversity
SWAP	8.1 (0.03)	20.0 (0.03)
Actual	8.6	12.2

(b) Region of Origin Diversity Gain

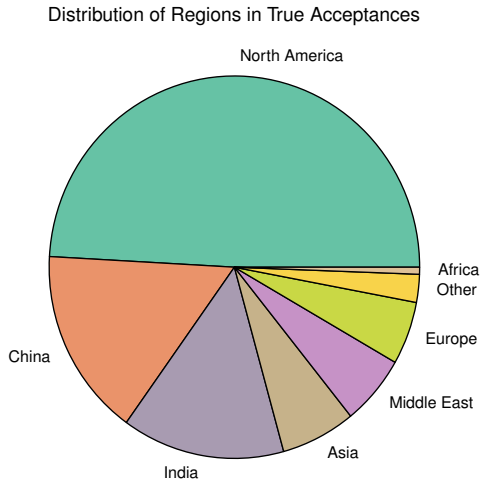
Table 1: SWAP's average gain in diversity over different classes. The first column shows general fit utility while the second shows diversity utility. In table 1a the third column shows male to female ratios. The standard deviation over the experiments of SWAP can be found in parentheses.

which is not entirely accurate. In reality, the true utility of an applicant is nontrivial to estimate as it depends on a wide range of factors. This experiment also simulated strong arm pulls—reviewers did not give any additional interviews of applicants during the experiment. Another limiting factor is that not every applicant we admit will matriculate into the program. Although the following results are promising, SWAP should be run in conjunction with an actual admissions process to assess its true performance.

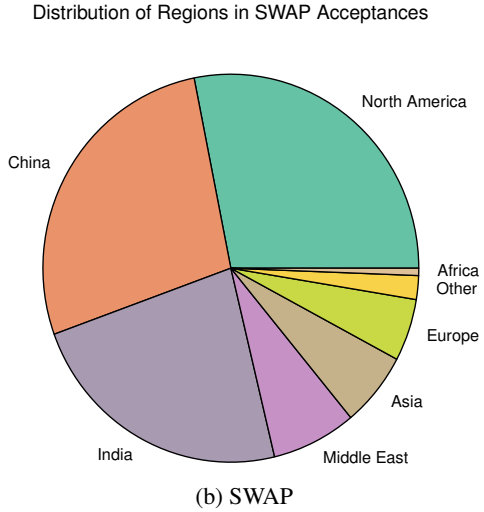
Results. We ran SWAP using the strong pull policy defined in Equation 4 over multiple values of s. We then compared the results of SWAP with the real decisions made during the admissions process. Averaging over all of the SWAP decisions, we compared the utility of the SWAP decision to that of the real decision, where the utility of each arm is defined by the classifier. There are two different types of objective functions that we take into account. The first treats all applicants as members of one global class, defined as $\sqrt{\sum_{a \in M} u(a)}$. This mimics a Top-K style objective, where applicants are valued based on individual merit alone. The second objective function is diversity-promoting, as defined in Equation 9, using reported gender (which is binary in our dataset) and region of origin for class memberships. We use those classes as our objective during separate runs of SWAP.

Table 1 and Figure 3 show experimental results on the test set (final year) of real admissions data. Since SWAP uses a diversity oracle (§5.1), we notice a slight drop in Top-K utility. We do, however, notice a gain in diversity.

SWAP, on average, used 1.17 pulls per arm, of which 0.05 were strong. During the last admissions decision process each applicant was reviewed on average 1.21 times. SWAP performed more strong pulls (interviews) of applicants than the graduate admissions committee did in practice, but did fewer weak pulls. SWAP spent roughly the same amount of total resources, when weighted by strong pull cost $j = 6$ and weak pull cost of 1, than the committee did. Given the gains in diversity, this supports SWAP's potential use in practice.



(a) Actual



(b) SWAP

Figure 3: Spread of acceptances over region of origin for SWAP (3b) and acceptances in practice (3a).

6 Conclusion

In this paper, we modeled the allocation of interviewing resources and subsequent selection of a cohort of applicants as a combinatorial pure exploration (CPE) problem in the multi-armed bandit setting. We generalized a recent CPE algorithm to the setting where arm pulls can have different costs—that is, where a decision maker can perform *strong* and *weak* pulls, with the former costing more than the latter, but also resulting in a less noisy signal. We presented the strong-weak arm-pulls (SWAP) algorithm and proved theoretical upper bounds for a general class of arm pulling strategies in that setting. We also provided simulation results to test the tightness of these bounds. We then applied SWAP to a real-world problem with combinatorial structure: incorporating diversity into university admissions. On real admissions data from one of the largest US-based computer science graduate programs, we showed that SWAP produces more diverse student cohorts at low cost to student quality while spending a budget comparable to that of the current admissions process at that university.

It would be of both practical and theoretical interest to tighten the upper bounds on convergence for SWAP, either for a reduced or general set of arm pulling strategies. SWAP should also be extended to include more than two types of pulls or information gathering strategies. We aim to incorporate a more realistic version of diversity and achieve a provably *fair* multi-armed bandit algorithm, as formulated by Joseph et al. (2016). Furthermore, we hope to determine whether or not there was bias in past admissions decisions and, if any, adapt the formulation of our diversity function to address it. Future work will include a live version of SWAP that will be used during the 2017-18 admissions period, with the goal of allocating resources more efficiently with an ever-increasing number of applicants.

References

[2017] Ahmed, F.; Dickerson, J. P.; and Fuge, M. 2017. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[2015] Ashkan, A.; Kveton, B.; Berkovsky, S.; and Wen, Z. 2015. Optimal greedy diversity for recommendation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1742–1748.

[2006] Bird, S. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, 69–72.

[2003] Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research (JMLR)* 3:993–1022.

[2012] Bubeck, S.; Cesa-Bianchi, N.; et al. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning* 5(1):1–122.

[2015] Cao, W.; Li, J.; Tao, Y.; and Li, Z. 2015. On top-k selection in multi-armed bandits and hidden bipartite graphs. In *Advances in Neural Information Processing Systems (NIPS)*, 1036–1044.

[2014] Chen, S.; Lin, T.; King, I.; Lyu, M. R.; and Chen, W. 2014. Combinatorial pure exploration of multi-armed bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 379–387.

[2001] Desrochers, P. 2001. Local diversity, human creativity, and technological innovation. *Growth and Change* 32(3):369–394.

[2013] Ding, W.; Qin, T.; Zhang, X.-D.; and Liu, T.-Y. 2013. Multi-armed bandit with budget constraint and variable costs. In *Conference on Artificial Intelligence (AAAI)*.

[2016] Gupta, N.; Sawhney, A.; and Roth, D. 2016. Will I get in? modeling the graduate admission process for American universities. In *International Conference on Data Mining (ICDM) Workshops*, 631–638.

[2015] Hunt, V.; Layton, D.; and Prince, S. 2015. Diversity matters. *McKinsey & Company*.

[2014] Jain, S.; Gujar, S.; Zoeter, O.; and Narahari, Y. 2014. A quality assuring multi-armed bandit crowdsourcing mechanism with incentive compatible learning. In *Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems (AAMAS)*.

[2016] Joseph, M.; Kearns, M.; Morgenstern, J. H.; and Roth, A. 2016. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems (NIPS)*, 325–333.

[2016] Jun, K.-S.; Jamieson, K.; Nowak, R.; and Zhu, X. 2016. Top arm identification in multi-armed bandits with batch arm pulls. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[2011] Lin, H., and Bilmes, J. 2011. A class of submodular functions for document summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT)*, 510–520.

[2014] Liu, X.; Suel, T.; and Memon, N. 2014. A robust model for paper reviewer assignment. In *Conference on Recommender Systems (RecSys)*.

[2016] Locatelli, A.; Gutzeit, M.; and Carpentier, A. 2016. An optimal algorithm for the thresholding bandit problem. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[2016] Lux, T.; Pittman, R.; Shende, M.; and Shende, A. 2016. Applications of supervised learning techniques on undergraduate admissions data. In *Proceedings of the International Conference on Computing Frontiers (CF)*.

[1978] Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming* 14(1):265–294.

[2011] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research (JMLR)* 12:2825–2830.

[2013] Qin, L., and Zhu, X. 2013. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[2008] Radlinski, F.; Kleinberg, R.; and Joachims, T. 2008. Learning diverse rankings with multi-armed bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, 784–791.

[2007] Schmader, T.; Whitehead, J.; and Wysocki, V. H. 2007. A linguistic comparison of letters of recommendation for male and female chemistry and biochemistry job applicants. *Sex Roles* 57(7-8):509514.

[2016] Sha, C.; Wu, X.; and Niu, J. 2016. A framework for recommending relevant and diverse items. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 3868–3874.

[2015] Singla, A.; Horvitz, E.; Kohli, P.; and Krause, A. 2015. Learning to hire teams. In *Third AAAI Conference on Human Computation and Crowdsourcing (HCOMP)*.

[2016] Singla, A.; Tschiatschek, S.; and Krause, A. 2016. Noisy Submodular Maximization via Adaptive Sampling with Applications to Crowdsourced Image Collection Summarization. In *Conference on Artificial Intelligence (AAAI)*.

[2013] Waters, A., and Miikkulainen, R. 2013. GRADE: Machine learning support for graduate admissions. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 1479–1486.

[2016] Xia, Y.; Qin, T.; Ma, W.; Yu, N.; and Liu, T.-Y. 2016. Budgeted multi-armed bandits with multiple plays. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

[2011] Yue, Y., and Guestrin, C. 2011. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems (NIPS)*, 2483–2491.

A Admissions Decisions Classifier

To effectively model the graduate admissions process, we needed a way to accurately represent whether a particular applicant will be admitted to the program. Using 3 years of previous admissions data, including letters of recommendation, we built a classifier modeling the graduate chair's decision for a particular applicant. The classifier's accuracy can be found in Table 2.

Type	% Correct	Precision	Recall
Ph.D.	77.8%	61.1%	39.7%
Masters	89.2%	13.1%	55.3%
Total	85.5%	33.5%	42.0%

Table 2: Current predictor results on the testing data

Some general features from the application are GPA, GRE scores, TOEFL scores, area of interest (Machine Learning, Theory, Vision, and so on), previous degrees, and universities attended. We included country of origin since the nature of applications may vary in different regions due to cultural norms. Another basic feature included was sex. We included this to check if the classifier picked up on any biased decision making (with sex and region).

Other features were generated from automatically processing the recommendation letters. Text from the letters was pulled from pdfs and OCR for scanned letters. We then cleaned the raw text with NLTK, removing stop words and stemming text (Bird 2006). One feature we chose was the length of recommendation letter, chosen after polling the admissions committee on what they thought would be important. Schmader, Whitehead, and Wysocki (2007) used Latent Dirichlet Allocation (LDA) to find word groups in recommendation letters for Chemistry and Biochemistry students (Blei, Ng, and Jordan 2003). Their five word groups included standout words (excellen*, superb, outstanding etc.), ability words (talent*, intell*, smart*, skill*, etc.), grindstone words (hardworking, conscientious, depend*, etc.), teaching words (teach, instruct, educat*, etc.), and research words (research*, data, study, etc.). We found that these word groups translated well to Computer Science students. Important words for acceptance were research words, standout words, and ability words. Letters that only included words from the teaching word group indicated a less useful recommendation letter. We used counts of the various word groups as a feature in the classifier.

B CLUCB Algorithm

The Combinatorial Lower-Upper Confidence Bound (CLUCB) algorithm by Chen et al. (2014) is shown in Algorithm 2. At the beginning of the algorithm, pull each arm once and initialize the empirical means with the rewards from that first arm pull. During iteration t of the algorithm, first find the set M_t using the Oracle. Then, compute the confidence radius for each arm. Find the worst case for each arm and compute a new set $\tilde{M}_t$ using the worst case estimates of the arms. If the utility of the initial set M_t and the worst case set $\tilde{M}_t$ are equal then output set M_t. Pull the most uncertain arm (the arm with the widest radius) from the symmetric difference of the two sets M_t and $\tilde{M}_t$. Update the empirical means.

Algorithm 2 Combinatorial Lower-Upper Confidence Bound (CLUCB)

Require: Confidence $\delta \in (0,1)$; Maximization oracle: Oracle$(\cdot) : \mathbb{R}^n \to \mathcal{M}$

Weak pull each arm $a \in [n]$ once.
Initialize empirical means $\bar{\mathbf{u}}_n$
$\forall a \in [n]$ set $T_n(a) \leftarrow 1$
for $t = n, n+1, \ldots$ **do**
  $M_t \leftarrow$ Oracle$(\bar{\mathbf{u}}_t)$
  $\forall a \in [n]$ compute confidence radius $\text{rad}_t(a)$
  for $a = 1, \ldots, n$ **do**
    if $a \in M_t$ **then** $\tilde{u}_t(a) \leftarrow \bar{u}_t(a) - \text{rad}_t(a)$
    else $\tilde{u}_t(a) \leftarrow \bar{u}_t(a) + \text{rad}_t(a)$
  $\tilde{M}_t \leftarrow$ Oracle$(\tilde{\mathbf{u}}_t)$
  if $\tilde{w}(\tilde{M}_t) = \tilde{w}(M_t)$ **then**
    Out $\leftarrow M_t$
    return Out
  $p_t \leftarrow \arg\max_{a \in (\tilde{M}_t \setminus M_t) \cup (M_t \setminus \tilde{M}_t)} \text{rad}_t(a)$
  Pull arm p_t
  Update empirical means $\bar{\mathbf{u}}_{t+1}$ using the observed reward
  $T_{t+1}(p_t) \leftarrow T_t(p_t) + 1$
  $T_{t+1} \leftarrow T_t(a)\ \forall a \neq p_t$

C Proofs

In this section, we formally prove the theorems discussed in our paper. Some lemmas we show directly feed from Chen et al. (2014)'s paper.

C.1 Strong Arm Pull Problem

The following maps to Lemma 8 in Chen et al. (2014).

Lemma 1. *Suppose that the reward distribution φ_a is a σ-sub-Gaussian distribution for all $a \in [n]$. And if, for all $t > 0$ and all $a \in [n]$, the confidence radius $rad_t(a)$ is given by*

$$rad_t(a) = \sigma \sqrt{\frac{2 \log\left(\frac{4nt^3j^3}{\delta}\right)}{T_t(a)}}$$

where $T_t(a)$ is the number of samples of arm a up to round t. Since $s > 1$ the number of samples in a single strong pull will be s each with cost j. Then, we have

$$\Pr\left[\bigcap_{t=1}^{\infty} \xi_t\right] \geq 1 - \delta.$$

Proof. Fix any $t > 0$ and $a \in [n]$. Note that φ_a is a σ-sub-Gaussian tail distribution with mean $w(a)$ and $\bar{w}_t(a)$ is the empirical mean of φ_a from $T_t(a)$ samples.

$$\Pr\left[|\bar{w}_t(a) - w_t(a)| \geq \sigma \sqrt{\frac{2 \log\left(\frac{4nt^3j^3}{\delta}\right)}{T_t(a)}}\right]$$

$$= \sum_{b=1}^{t-1} \Pr\left[|\bar{w}_t(a) - w_t(a)| \geq \sigma \sqrt{\frac{2\log\left(\frac{4nt^3j^3}{\delta}\right)}{bs}}, T_t(a) = bs \right] \tag{10a}$$

$$\leq \sum_{b=1}^{t-1} 2\exp\left(\frac{-bs\left(\sigma\sqrt{\frac{2\log\left(\frac{4nt^3j^3}{\delta}\right)}{bs}}\right)^2}{2\sigma^2} \right) \tag{10b}$$

$$= \sum_{b=1}^{t-1} \frac{\delta}{2nt^3j^3}$$

$$\leq \frac{\delta}{2nt^2j^3} \tag{10c}$$

where Eq.10a follows from the fact that $1 \leq T_t(a)/s \leq t-1$ and Eq.10b follows from Hoeffding's inequality. By a union bound over all $a \in [n]$, we see that $\Pr[\xi_t] \geq 1 - \frac{\delta}{2t^2j^3}$. Using a union bound again over all $t > 0$, we have

$$\begin{aligned}
\Pr\left[\bigcap_{t=1}^{\infty} \xi_t\right] &\geq 1 - \sum_{t=1}^{\infty} \Pr[\neg\xi_t] \\
&\geq 1 - \sum_{t=1}^{\infty} \frac{\delta}{2t^2j^3} \\
&= 1 - \frac{\pi^2}{12j^3}\delta \\
&\geq 1 - \delta
\end{aligned}$$

□

The rest of the lemmas in Chen et al. (2014)'s paper hold. We can now prove Theorem 2

Theorem 2. *Given any $\delta \in (0,1)$, any decision class $\mathcal{M} \subseteq 2^{[n]}$, and any expected rewards $\mathbf{w} \in \mathbb{R}^n$, assume that the reward distribution φ_a for each arm $a \in [n]$ has mean $w(a)$ with an σ-sub-Gaussian tail. Let $M_* = \arg\max_{M\in\mathcal{M}} w(M)$ denote the optimal set. Set $rad_t(a) = \sigma\sqrt{2\log\left(\frac{4nt^3j^3}{\delta}/T_t(a)\right)}$ for all $t > 0$ and $a \in [n]$. Then, with probability at least $1-\delta$, the CLUCB algorithm with only strong pulls where $j \geq 1$ and $s > j$ returns the optimal set $\texttt{Out} = M_*$ and*

$$T \leq O\left(\frac{\sigma^2\, width(\mathcal{M})^2 \mathbf{H}\log(nj^3R^2\mathbf{H}/\delta)}{s}\right) \tag{11}$$

where T denotes the number of samples used by the CLUCB algorithm, $\mathbf{H}$ is defined in Eq.3.

Proof. Lemma 1 indicates that the event $\xi \triangleq \bigcap_{t=1}^{\infty}\xi_t$ occurs with probability at least $1-\delta$. In the rest of the proof, we shall assume that this event holds.

By using Lemma 9 from Chen et al. (2014) and the assumption on ξ, we see that $\texttt{Out} = M_*$. Next, we focus on bounding the total number of T samples.

Fix any arm $a \in [n]$. Let $T(a)$ denote the total information gained from pulling arm $a \in [n]$. Let t_a be the last round which arm a is pulled, which means that $p_{t_a} = e$. It is easy to see that $T_{t_a}(a) = T(a) - s$. By Lemma 10 from chen et. al., we see that $\text{rad}_{t_a} \geq \frac{\Delta_a}{3\,\text{width}(\mathcal{M})}$. Using the definition of rad_{t_a}, we have

$$\frac{\Delta_a}{3\,\text{width}(\mathcal{M})} \leq \sigma\sqrt{\frac{2\log(4nt_a^3j^3/\delta)}{T(a)-s}} \leq \sigma\sqrt{\frac{2\log(4nT^3j^3/\delta)}{T(a)-s}}. \tag{12}$$

By solving Eq.12 for $T(a)$, we obtain

$$T(a) \leq \frac{18\,\text{width}(\mathcal{M})^2\sigma^2}{\Delta_a^2}\log(4nT^3j^3/\delta) + s \tag{13}$$

Define $\tilde{\mathbf{H}} = \max\{\text{width}(\mathcal{M})^2\sigma^2\mathbf{H}, 1\}$. Using similar logic to Chen et al. (2014) and the fact that the information gained per pull is s, we show that

$$T \leq \frac{499\,\tilde{\mathbf{H}}\log(4nj^3\,\tilde{\mathbf{H}}/\delta)}{s} + 2n \tag{14}$$

Theorem 2 follows immediately from Eq. 14.

If $n \geq \frac{1}{2}T$, then $T \leq 2n$ and Eq. 14 holds. For the second case we assume $n < \frac{1}{2}T$. Since $T > n$, we write

$$T = \frac{C\,\tilde{\mathbf{H}}\log\left(4nj^3\,\tilde{\mathbf{H}}/\delta\right)}{s} + n, \text{ for some } C > 0. \tag{15}$$

If $C < 499$, then Eq. 14 holds. Suppose, on the contrary, that $C > 499$. We know that $T = \frac{1}{s}\sum_{a\in[n]} T(a)$. Using this fact and summing Eq. 13 for all $a \in [n]$, we have

$$\begin{aligned}
T &\leq \frac{1}{s}\left(ns + \sum_{a\in[n]} \frac{18\,\text{width}(\mathcal{M})^2\sigma^2}{\Delta_a^2}\log(4nj^3T^3/\delta)\right) \\
&\leq n + \frac{18\,\tilde{\mathbf{H}}\log(4nj^3T^3/\delta)}{s} \\
&= n + \frac{18\,\tilde{\mathbf{H}}\log(4nj^3/\delta)}{s} + \frac{54\,\tilde{\mathbf{H}}\log(T)}{s} \\
&\leq n + \frac{18\,\tilde{\mathbf{H}}\log(4nj^3/\delta)}{s} \\
&\quad + \frac{54\,\tilde{\mathbf{H}}\log(2C\,\tilde{\mathbf{H}}\log(4nj^3\,\tilde{\mathbf{H}}/\delta))}{s} \qquad (16) \\
&= n + \frac{18\,\tilde{\mathbf{H}}\log(4nj^3/\delta)}{s} \\
&\quad + \frac{54\,\tilde{\mathbf{H}}\log(2C)}{s} + \frac{54\,\tilde{\mathbf{H}}\log(\tilde{\mathbf{H}})}{s} \\
&\quad + \frac{54\,\tilde{\mathbf{H}}\log\log(4nj^3\,\tilde{\mathbf{H}}/\delta)}{s} \\
&\leq n + \frac{18\,\tilde{\mathbf{H}}\log(4nj^3\,\tilde{\mathbf{H}}/\delta)}{s} \\
&\quad + \frac{54\,\tilde{\mathbf{H}}\log(2C)\log(4nj^3\,\tilde{\mathbf{H}}/\delta)}{s} \\
&\quad + \frac{54\,\tilde{\mathbf{H}}\log(4nj^3\,\tilde{\mathbf{H}}/\delta)}{s} + \frac{54\,\tilde{\mathbf{H}}\log(4nj^3\,\tilde{\mathbf{H}}/\delta)}{s} \qquad (17)
\end{aligned}$$

$$
\begin{aligned}
&= (126 + 54\log(2C))\frac{\tilde{\mathbf{H}}\log(4nj^3\,\tilde{\mathbf{H}}/\delta)}{s} \\
&< n + \frac{C\,\tilde{\mathbf{H}}\log(4nj^3\,\tilde{\mathbf{H}}/\delta)}{s} \qquad (18)\\
&= T, \qquad (19)
\end{aligned}
$$

where Eq. 16 follows from Eq. 15 and the assumption that $n < \frac{1}{2}T$; Eq. 17 follows from $\tilde{\mathbf{H}} \geq 1$, $j \geq 1$, and $\delta < 1$; Eq. 18 follows since $126 + 54\log(2C) < C$ for all $C > 499$; and Eq. 19 is due to Eq. 15. So Eq. 19 is a contradiction. Therefore $C \leq 499$ and we have proved Eq. 14. □

C.2 Strong Weak Arm Pull (SWAP)

The following corresponds to Lemma 8 in work by the Chen et al. (2014).

Lemma 2. *Suppose that the reward distribution φ_a is a σ_1-sub-Gaussian distribution for all $a \in [n]$. For all $t > 0$ and all $a \in [n]$, the confidence radius $rad_t(a)$ is given by*

$$rad_t(a) = \sigma_1\sqrt{\frac{2\log\left(\frac{4nCost_t^3}{\delta}\right)}{T_t(a)}}$$

where $T_t(a)$ is the number of samples of arm a up to round t. Since $s > 1$, the number of samples in a single strong pull are s each with cost j. Then, we have

$$\Pr\left[\bigcap_{t=1}^{\infty}\xi_t\right] \geq 1 - \delta.$$

Proof. Fix any $t > 0$ and $a \in [n]$. Note that φ_a is σ_1-sub-Gaussian tail distribution with mean $w(a)$ and $\bar{w}(a)$ is the empirical mean of φ_a from $T_t(a)$ samples. Then we have

$$
\begin{aligned}
&\Pr\left[|\bar{w}_t(a) - w_t(a)| \geq \sigma_1\sqrt{\frac{2\log\left(\frac{4nCost_t^3}{\delta}\right)}{T_t(a)}}\right] \qquad (20)\\
&= \sum_{b=1}^{t-1}\Pr\left[|\bar{w}_t(a) - w_t(a)| \geq \sigma_1\sqrt{\frac{2\log\left(\frac{4nCost_t^3}{\delta}\right)}{Gain_b}}\right] \qquad (21)\\
&\leq \sum_{b=1}^{t-1} 2\exp\left(\frac{-Gain_b\left(\sigma_1\sqrt{\frac{2\log\left(\frac{4nCost_t^3}{\delta}\right)}{Gain_b}}\right)^2}{2R^2}\right) \qquad (22)\\
&= \sum_{b=1}^{t-1}\frac{\delta}{2nAvCost^3t^3}\\
&\leq \frac{\delta}{2nt^2AvCost^3} \qquad (23)
\end{aligned}
$$

where $AvCost$ equal to the average cost until time t. Eq.21 follows from $1 \leq T_t(a)/Gain_t \leq t-1$ and Eq.22 follows from Hoeffding's inequality. By a union bound over all $a \in [n]$, we see that $\Pr[\xi_t] \geq 1 - \frac{\delta}{2t^2AvCost_t^3}$. Using a union bound again over all $t > 0$, we have

$$
\begin{aligned}
\Pr\left[\bigcap_{t=1}^{\infty}\xi_t\right] &\geq 1 - \sum_{t=1}^{\infty}\Pr[\neg\xi_t]\\
&\geq 1 - \sum_{t=1}^{\infty}\frac{\delta}{2t^2AvCost^3}\\
&= 1 - \frac{\pi^2}{12AvCost^3}\delta\\
&\geq 1 - \delta
\end{aligned}
$$

□

Given that the rest of the lemmas in the Chen et al. (2014) paper hold, we now prove the main theorem of our paper.

Theorem 3. *Given any $\delta_1, \delta_2, \delta_3 \in (0,1)$, any decision class $\mathcal{M} \subseteq 2^{[n]}$ and any expected rewards $\mathbf{w} \in \mathbb{R}^n$, assume that the reward distribution φ_a for each arm $a \in [n]$ has mean $w(a)$ with an σ_1-sub-Gaussian tail. Let $M_* = \arg\max_{M\in\mathcal{M}} w(M)$ denote the optimal set. Set $rad_t(a) = \sigma_1\sqrt{2\log\left(\frac{4nCost_t^3}{\delta}/T_t(a)\right)}$ for all $t > 0$ and $a \in [n]$, set $\epsilon_1 = \sigma_2\sqrt{2\log\left(\frac{1}{2}\delta_2/T\right)}$, and set $\epsilon_2 = \sigma_3\sqrt{2\log\left(\frac{1}{2}\delta_3/n\right)}$. Then, with probability at least $(1-\delta_1)(1-\delta_2)(1-\delta_3)$, the SWAP algorithm (Algorithm 1) returns the optimal set $\texttt{Out} = M_*$ and*

$$T \leq O\left(\frac{R^2\,width(\mathcal{M})^2\mathbf{H}\log\left(nR^2\left(\bar{X}_{Cost} - \epsilon_1\right)^3\mathbf{H}/\delta\right)}{\bar{X}_{Gain} - \epsilon_2}\right), \qquad (24)$$

where T denotes the number of samples used by Algorithm 1, $\mathbf{H}$ is defined in Eq. 3 and width$(\mathcal{M})$ is defined by Chen et al. (2014).

Proof. Lemma 2 indicates that the event $\xi \triangleq \bigcap_{t=1}^{\infty}\xi_t$ occurs with probability at least $1-\delta$. In the rest of the proof, we assume that this event holds.

Using Lemma 9 from Chen et al. (2014) and the assumption on ξ, we see that $\texttt{Out} = M_*$. Next, we bound the total number of T samples.

Fix any arm $a \in [n]$. Let $T(a)$ denote the total information gained from pulling arm $a \in [n]$. Let t_a be the last round which arm a is pulled, which means that $p_{t_a} = a$. Trivially, $T_{t_a}(a) = T(a) - s$. By Lemma 10 from Chen et al. (2014), we see that $\text{rad}_{t_a} \geq \frac{\Delta_a}{3\,\text{width}(\mathcal{M})}$. Using the definition of rad_{t_a}, we have

$$
\begin{aligned}
\frac{\Delta_a}{3\,\text{width}(\mathcal{M})} &\leq R\sqrt{\frac{2\log(4nCost_{t_a}^3/\delta)}{T(e) - Gain_{t_a}}}\\
&\leq R\sqrt{\frac{2\log(4nCost_T^3/\delta)}{T(a) - Gain_{t_a}}}. \qquad (25)
\end{aligned}
$$

Solving for $T(a)$ in Eq. 25 we get

$$T(a) \leq \frac{18\,\text{width}(\mathcal{M})^2 R^2}{\Delta_e^2} \log(4nCost_T^3/\delta) + Gain_{t_a} \tag{26}$$

Define $\bar{X}_{Cost} = \mathbb{E}[Cost]$ as the expected cost of pulling an arm. Since we strong pull an arm with probability $\alpha = \frac{s-j}{s-1}$, we know

$$\bar{X}_{Cost} = \mathbb{E}[Cost_T] = \alpha j + (1-\alpha). \tag{27}$$

Define X_{Cost_t} as the cost of pulling an arm at time t. Assuming that each random variable X_{Cost_t} is R_1-sub-Gaussian we can write the following using the Hoeffding inequality,

$$\Pr\left(\left|\frac{1}{T}\sum_{t=1}^{T} C_{Cost_t} - \bar{X}_{Cost}\right| \geq \epsilon_1\right) \leq 2\exp\left(-\frac{T\epsilon_1^2}{2R_1}\right) \tag{28}$$

If we set $\epsilon_1 = R_1\sqrt{2log(\frac{1}{2}\delta_2)/T}$ then with probability $(1-\delta_2)$

$$\frac{Cost_T}{T} \in \left(\bar{X}_{Cost} - \epsilon_1, \bar{X}_{Cost} + \epsilon\right). \tag{29}$$

Combining Eq. 26 and Eq. 29 we get

$$T(e) \leq \frac{18\,\text{width}(\mathcal{M})^2 R^2}{\Delta_e^2} \log(4n(\bar{X}_{Cost}-\epsilon_1)^3 T^3/\delta) + Gain_{t_e} \tag{30}$$

Define $\bar{X}_{Gain} = E[Gain]$ as the expected information gain from pulling an arm. Since we pull an arm with probability α, we know that

$$\bar{X}_{Gain} = E[Gain] = \alpha s + (1-\alpha) \tag{31}$$

Define X_{Gain_t} as the information gain of pulling an arm at time t. Assuming that each random variable X_{Gain_t} is R_2-sub-Gaussian we can write the following using the Hoeffding inequality.

$$\Pr\left(\left|\frac{1}{n}\sum_{e\in[n]} Gain_{t_e} - \bar{X}_{Gain}\right| \geq \epsilon_2\right) \leq 2\exp\left(\frac{-n\epsilon_2^2}{2R_2^2}\right) \tag{32}$$

If we set $\epsilon_2 = R_2\sqrt{2log(\frac{1}{2}\delta_3)/n}$ then with probability $(1-\delta_2)$

$$\frac{\sum_{e\in[n]} Gain_{t_e}}{n} \in \left(\bar{X}_{Gain} - \epsilon_2, \bar{X}_{Gain} + \epsilon_2\right). \tag{33}$$

Similarly to the proof for Theorem 2, define $\tilde{\mathbf{H}} = \max\{\text{width}(\mathcal{M})^2 R^2 \mathbf{H}, 1\}$. In the rest of the proof we will show that

$$T \leq \frac{499\,\tilde{\mathbf{H}} \log\left(4n\left(\bar{X}_{Cost}+\epsilon_1\right)^3 \tilde{\mathbf{H}}/\delta\right)}{\bar{X}Gain - \epsilon_2} + 2n \tag{34}$$

Notice that theorem follows immediately from Eq. 34.

If $n \geq \frac{1}{2}T$, then Eq. 34 holds. Let's then assume that $n < \frac{1}{2}T$. Since $T > n$, we can write

$$T = \frac{C\,\tilde{\mathbf{H}} \log(4n(X_{Cost}+\epsilon_1)^3 \tilde{\mathbf{H}}/\delta}{\bar{X}_{Gain} - \epsilon_2} + n \tag{35}$$

If $C \leq 499$ then Eq. 34 holds. Suppose then that $C > 499$. Notice that $T = \sum_{a\in[n]} T(a)/Gain_{t_a}$. By summing up Eq. 30 for all $a \in [n]$ we have

$$\begin{aligned}
T &\leq n + \sum_{a\in[n]} \frac{18\,\text{width}(\mathcal{M})^2 R^2 \log(4n(\bar{X}_{Cost}+\epsilon_1)T^3/\delta}{\Delta_a^2 Gain_{t_a}} \\
&\leq n + \frac{18\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3 T^3/\delta)}{\bar{X}_{Gain} - \epsilon_2} && (36) \\
&= n + \frac{18\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3/\delta)}{\bar{X}_{Gain} - \epsilon_2} + \frac{54\,\tilde{\mathbf{H}} \log(T)}{\bar{X}_{Gain} - \epsilon_2} \\
&\leq n + \frac{18\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3/\delta)}{\bar{X}_{Gain} - \epsilon_2} \\
&\quad + \frac{54\,\tilde{\mathbf{H}} \log(2c\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}-\epsilon_1)^3\,\tilde{\mathbf{H}}/\delta))}{\bar{X}_{Gain} - \epsilon_2} && (37) \\
&= n + \frac{18\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3/\delta)}{\bar{X}_{Gain} - \epsilon_2} + \frac{54\,\tilde{\mathbf{H}} \log(2C)}{\bar{X}_{Gain} - \epsilon_2} \\
&\quad + \frac{54\,\tilde{\mathbf{H}} \log(\tilde{\mathbf{H}})}{\bar{X}_{Gain} - \epsilon_2} \\
&\quad + \frac{54\,\tilde{\mathbf{H}} \log\log(4n(\bar{X}_{Cost}+\epsilon_1)^3\,\tilde{\mathbf{H}}/\delta)}{\bar{X}_{Gain} - \epsilon_2} \\
&\leq n + \frac{18\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3\,\tilde{\mathbf{H}}/\delta)}{\bar{X}_{Gain} - \epsilon_2} \\
&\quad + \frac{54\,\tilde{\mathbf{H}} \log(2C) \log(4n(\bar{X}_{Cost}+\epsilon_1)^3\,\tilde{\mathbf{H}}/\delta)}{\bar{X}_{Gain} - \epsilon_2} \\
&\quad + \frac{54\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3\,\tilde{\mathbf{H}}/\delta)}{\bar{X}_{Gain} - \epsilon_2} \\
&\quad + \frac{54\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3\,\tilde{\mathbf{H}}/\delta)}{\bar{X}_{Gain} - \epsilon_2} && (38) \\
&= n + (126 + 54\log(2C))\frac{\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3\,\tilde{\mathbf{H}}/\delta)}{\bar{X}_{Gain} - \epsilon_2} \\
&< n + \frac{C\,\tilde{\mathbf{H}} \log(4n(\bar{X}_{Cost}+\epsilon_1)^3\,\tilde{\mathbf{H}}/\delta)}{\bar{X}_{Gain} - \epsilon_2} && (39) \\
&= T, && (40)
\end{aligned}$$

where Eq. 36 follows from Eq. 33; Eq. 37 follows from Eq. 35 and the assumption $n < \frac{1}{2}T$; Eq. 38 follows from $\tilde{\mathbf{H}} \geq 1$, $\delta < 1$, and $\bar{X}_{Cost} + \epsilon \geq 1$; Eq. 39 follows since $126 + 54\log(2C) < C$ for all $C > 499$; and Eq. 40 is due to Eq. 35. So Eq. 40 is a contradiction. Therefore $C \leq 499$ and we have proved Eq. 34. □

D Experiments

D.1 Gaussian Experiments

While running SWAP, we first compare where the general, varied-cost version of SWAP is better than SWAP with strong pulls only (Figure 4) and where it is better than SWAP with only weak pulls (Figure 5). We then noticed that there should be an optimal zone where the general version of SWAP would perform better than both of the trivial cases.

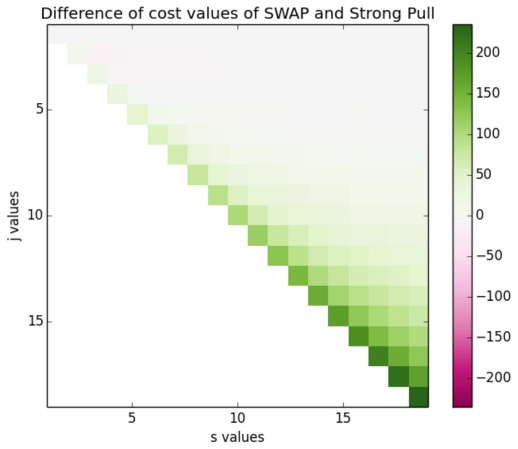


Figure 4: Heat map showing where SWAP is better than Strong Pull Only.

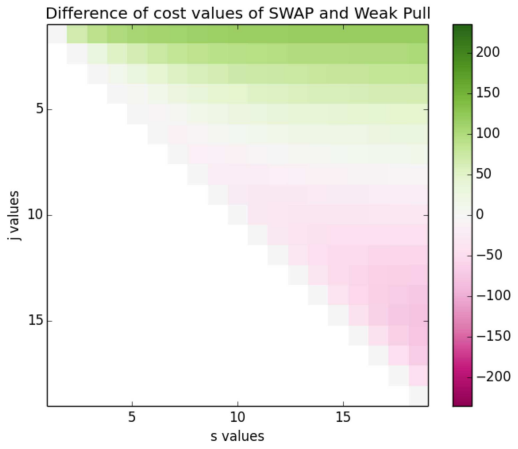


Figure 5: Heat map showing where SWAP is better than Weak Pull Only.

Both graphs examine the symmetric difference between the average cost values of SWAP and either Strong or Weak Pull only with different parameter values of s and j.

D.2 Graduate Admissions Experiment

We ran SWAP over both Masters and Ph.D. students over various values of s (Figure 6). The total cost of running these experiments aligns with the resources spent during the actual admissions decision process.

When running SWAP experiments to formally promote diversity, one experiment not listed in the main paper was testing our diverse SWAP algorithm over an applicant's main choice of research area (Table 3). In practice, the applicants accepted already had a high diversity utility in regards to research area. SWAP slightly increased this diversity utility.

Figure 6: Total cost of running SWAP over different s values

	General	Diversity
SWAP	8.3 (0.03)	32.5 (0.03)
Actual	8.6	27.4

Table 3: SWAP's average gain in reported area of study diversity over our actual acceptances. The first column shows general fit utility and the second diversity utility The standard deviation over the experiments of SWAP can be found in parentheses.