Mechanisms with Learning for Stochastic Multi-armed Bandit Problems

Shweta Jain[1], Satyanath Bhat[1], Ganesh Ghalme[1], Divya Padmanabhan[1], and Y. Narahari[1]

Department of Computer Science and Automation,
Indian Institute of Science,
Bengaluru,
India 560012

Abstract. The multi-armed bandit (MAB) problem is a widely studied problem in machine learning literature in the context of online learning. In this article, our focus is on a specific class of problems namely stochastic MAB problems where the rewards are stochastic. In particular, we emphasize stochastic MAB problems with strategic agents. Dealing with strategic agents warrants the use of mechanism design principles in conjunction with online learning, and leads to non-trivial technical challenges. In this paper, we first provide three motivating problems arising from Internet advertising, crowdsourcing, and smart grids. Next, we provide an overview of stochastic MAB problems and key associated learning algorithms including Upper Confidence Bound (UCB) based algorithms. We provide proofs of important results related to regret analysis of the above learning algorithms. Following this, we present mechanism design for stochastic MAB problems. With the classic example of sponsored search auctions as a backdrop, we bring out key insights in important issues such as regret lower bounds, exploration separated mechanisms, designing truthful mechanisms, UCB based mechanisms, and extension to multiple pull MAB problems. Finally we provide a bird's eye view of recent results in the area and present a few issues that require immediate future attention.

1 Introduction

The stochastic multi-armed bandit (MAB) problem is a classical problem, originally described by Robbins (1952)[23]. In the MAB problem, a gambler is required to pull one of the K arms of a gambling machine for T time periods. Each arm, when pulled, yields a random reward. The rewards associated with the arms are independent and are drawn from distributions which are unknown to the gambler. The objective of the gambler is to pull the arms in a way that the expectation of the sum of the rewards over T time periods is maximized. Since the reward distributions associated with the arms are unknown, there is a need to learn the parameters of the distributions while observing the rewards. Only the actual rewards of the pulled arm can be observed by the gambler. In each time period, the gambler has to make a choice between (1) pulling an arm that has given the best reward so far and (2) exploring the other arms that may potentially give better rewards in the future. Thus, this problem captures a situation where the gambler is faced with a trade-off between exploration (pulling less explored arms in search of an arm with better reward) and exploitation (pulling the arm known to be best till the current time instant, in terms of yielding the maximum reward). There are numerous practical applications of this problem which includes the first motivating problem introduced by Robbins (1952) [23], namely, clinical trials where the arms correspond to candidate treatments for a patient and the objective is to minimize health losses. More recently, solution techniques to the MAB problem have been applied to many modern applications such as online advertisements, crowdsourcing, and smart grids (we describe these is more detail in the next section).

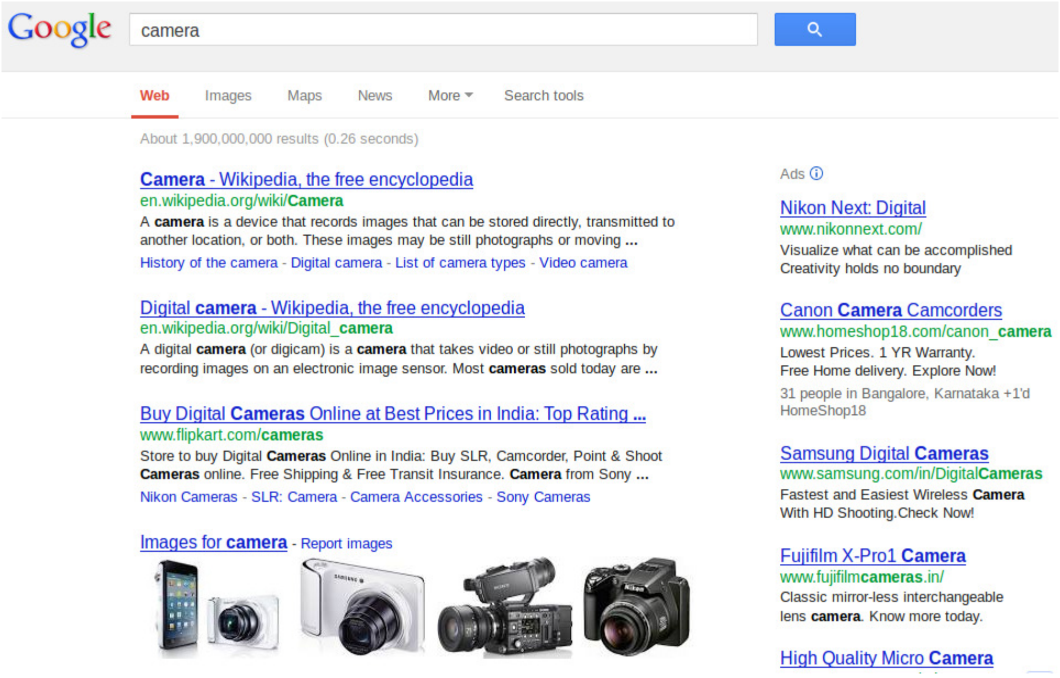
Besides stochastic multi-armed bandit problems, there exist many other variations of MAB problems, such as Markovian bandits and adversarial bandits (see, for example, the survey article by Bubeck and Cesa-Bianchi [8]). In this paper we will restrict our discussion to stochastic MAB problems.

In many settings like online advertising and crowdsourcing, the role of the arms is played by *strategic* advertisers and workers respectively. These strategic agents may hold some private information which is of interest to the learner. Since the agents are interested only in maximizing their own utilities, they could misreport the information they have, thereby hampering the learning process. Therefore, available MAB solutions are not directly applicable in the presence of strategic agents. To deal with strategic play by the agents, there is a need to model the strategic behavior of the agents and use ideas from mechanism design to ensure that the agents report their private information truthfully. This leads to the notion of *multi-armed bandit mechanisms*. We now present three real world applications where MAB mechanisms find use.

1.1 Motivating Examples

1.1.1 Sponsored Search Auctions Sponsored advertisements play a major role in the revenue earned by the search engine companies. Whenever an internet user searches for a keyword in a search engine like Bing or Google or Yahoo!, various links related to the keyword are displayed along with certain sponsored advertisements. Figure 1 shows an example of such sponsored search links displayed by Google. There is space for showing only a limited number of slots (typically two to five) and the search engine has to allocate the slots among thousands of competing advertisers. Each advertiser has a certain valuation for the click and an interesting challenge arises in the so called pay-per-click auctions where advertisers pay only when a user clicks on the advertisement.

In order to select the advertisements to be displayed, and the price to be levied on the corresponding advertisers, the search engine runs an auction among the interested advertisers. Advertisers are asked to bid their valuations for each click they receive. If the advertisement displayed gets clicked by a user, then the advertiser pays a certain amount of money to the search engine, otherwise the advertiser does not pay any amount. Typically, the frequency of clicks (also known as click through rate) that an advertisement would receive is unknown to the search engine as well as to the advertiser. The auction is typically repeated for a certain period of rounds (say T).



About 1,900,000,000 results (0.26 seconds)

Camera - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Camera**
A **camera** is a device that records images that can be stored directly, transmitted to another location, or both. These images may be still photographs or moving ...
History of the camera - Digital camera - List of camera types - Video camera

Digital **camera** - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Digital_**camera**
A digital **camera** (or digicam) is a **camera** that takes video or still photographs by recording images on an electronic image sensor. Most **cameras** sold today are ...

Buy Digital **Cameras** Online at Best Prices in India: Top Rating ...
www.flipkart.com/**cameras**
Store to buy Digital **Cameras** Online in India: Buy SLR, Camcorder, Point & Shoot **Cameras** online. Free Shipping & Free Transit Insurance. **Camera** from Sony ...
Nikon Cameras - SLR: Camera - Camera Accessories - Sony Cameras

Images for **camera** - Report images

Nikon Next: Digital
www.nikonnext.com/
Visualize what can be accomplished
Creativity holds no boundary

Canon **Camera** Camcorders
www.homeshop18.com/canon_**camera**
Lowest Prices. 1 YR Warranty.
Free Home delivery. Explore Now!
31 people in Bangalore, Karnataka +1'd HomeShop18

Samsung Digital **Cameras**
www.samsung.com/in/Digital**Cameras**
Fastest and Easiest Wireless **Camera**
With HD Shooting.Check Now!

Fujifilm X-Pro1 **Camera**
www.fujifilm**cameras**.in/
Classic mirror-less interchangeable
lens **camera**. Know more today.

High Quality Micro **Camera**

Fig. 1: Sponsored search by Google search engine

Suppose the click through rate of the advertisements is known in advance. Then, to elicit the true valuations per click from the advertisers, a standard mechanism like the Clarke mechanism [9] could be used. However, in practical scenarios, the click through rates of the advertisers are not known to the search engine. On the other hand, if we assume that the valuations per click of the advertisers are known, then the problem of allocating advertisers to the slots reduces to the MAB problem where each advertisement is mapped to an arm.

Thus, the central issue for such pay-per-click auctions is to estimate the click through rates and, at the same time, design a mechanism which incentivizes the advertisers to bid their true valuations per click. Hence, the problem of designing mechanisms combines aspects of both online learning and strategic behavior. This problem could be modeled as a problem of designing MAB mechanisms [24, 5, 12, 10].

1.1.2 Crowdsourcing Crowdsourcing is emerging as a powerful means available to employers or service requesters to get intelligent or human computation tasks executed in a timely, scalable, and cost-effective manner. As an immediate example, crowdsourcing is used widely in the AI community to collect labels for large scale learning problems. There is now available a plethora of platforms or crowdsourcing marketplaces depending upon the expertise required for her tasks. On the other hand, workers around the globe, with varying qualifications and backgrounds, benefit from crowdsourcing as it provides a means to supplement their incomes.

The diverse and heterogeneous demographics of crowd workers is both a boon and bane. In particular, since there could be a large variance in the worker qualities, the requester has the additional task of weeding out spammers. This task can be posed as a MAB problem where the arms correspond to the workers and the requester can either explore them to learn their qualities or choose to exploit the best set of workers identified so far [17, 2, 8]. An additional challenge is that the workers differ in their expectation of remuneration for identical tasks. A simple posted price scheme, such as employed by the popular platform Amazon Mechanical Turk (AMT), could prove counterproductive if prices are set too low (since high quality workers may simply drop out). A suitably chosen procurement auction with a reserve price can elegantly address this issue [6]. The problem becomes far more challenging when the costs of the workers (which are private to the workers) need to be elicited truthfully and the worker qualities have to be learnt simultaneously. One can pose this problem as that of designing a MAB mechanism [7, 6, 14].

1.1.3 Smart Grids Power companies across the world typically face two major challenges which are peak demand and power imbalance (supply minus demand). Peak demand corresponds to a time window in which the demand for power is significantly higher than the average supply level. In order to tackle this, users are charged higher rates during the peak demand period. This scheme encourages a user to shift her electricity load from peak time to non-peak time, thus reducing the overall load on the electricity grid.

Even though increasing prices in response to the peak demand encourages users to shift their consumption to non-peak time, dynamic prices in electricity grids leads to user inconvenience. As an alternative to dynamic pricing, power companies could make monetary offers to users to make them reduce their consumption. The monetary offer appropriate enough for a user to reduce his consumption can depend on his preferences which may be private to him. Moreover, even if the incentives offered are sufficient to a user, she may still not reduce the consumption due to the stochasticity involved in her needs for electricity. This problem can again be cast as a multi-armed bandit mechanism to elicit user preferences truthfully and to learn the stochasticity involved [15].

1.2 Outline of the Paper

The rest of the paper follows the trajectory outlined below.

- **Section 2: The Stochastic MAB Problem**: We set up the notation for the stochastic MAB problem and provide a proof of a general logarithmic lower bound on regret.
- **Section 3: Stochastic MAB Algorithms**: We discuss learning algorithms in the stochastic MAB context under two main categories. The first category of algorithms use the *frequentist* approach. Here we discuss: exploration separated algorithms; upper confidence bound (UCB) family of algorithms - UCB1, UCB-Normal, (α, ψ)-UCB, and KL-UCB algorithms. The second category of algorithms use the *Bayesian* approach, where we discuss Thompson sampling and Bayes-UCB algorithm.
- **Section 4: Mechanism Design Overview**: We provide an overview of classical mechanism design including key definitions and concepts.
- **Section 5: Stochastic MAB Mechanisms**: First, we describe a mechanism design environment for MAB mechanisms. With the sponsored search auction problem as a backdrop, we define important notions for MAB mechanisms. Following this, we present a lower bound regret analysis for MAB mechanisms and contrast the bounds with the bounds in the absence of strategic agents. Next, we discuss exploration separated mechanisms. Then we describe a popular procedure for designing a truthful mechanism starting from a monotone allocation
rule. We subsequently discuss UCB-based mechanisms and conclude with a discussion of the complexities that arise when multiple arms are pulled instead of a single arm.
- **Section 6: Recent Work in the Literature**: There is a steady stream of papers, mostly in the context of sponsored search auctions, in the recent literature, which we review here. This includes some of our own work as well.
- **Section 7: Summary and Directions of Future Work**: This section presents what we believe are some important open problems in this area.

Besides stochastic multi-armed bandit problems, there exist other variations like Markovian bandits and adversarial bandits. There has been some work on mechanism design under these settings [22, 21, 5]. However, in this paper we will restrict our discussion to mechanism design for stochastic MAB problems.

2 The Stochastic MAB Problem

2.1 Model and Notations

In the classical stochastic MAB setting, there are K independent arms with reward distributions of known form characterized by unknown but fixed parameters $\rho_1, \rho_2, \dots, \rho_K \in \Upsilon$. When an arm j is pulled at any time t, it generates a reward $X_j(t)$ which is drawn independently from distribution ν_{ρ_j} whose expectation is denoted by $\mu_j = \mu(\rho_j)$. The algorithm $\mathcal{A}$ sequentially pulls the arms till T time periods or rounds. The arm pulled by the algorithm at time t is denoted by I_t. We also refer to algorithm $\mathcal{A}$ interchangeably as the allocation strategy. The goal of the algorithm is to maximize the expected reward over T rounds i.e.

$$\mathbb{E}_{\mathcal{A}}[\textstyle\sum_{t=1}^{T}\sum_{j=1}^{K} X_j(t)\mathbb{1}(I_t = j)],$$

where, $\mathbb{1}(.)$ is an indicator function which is 1 if arm j is pulled at time t and is 0 otherwise. The performance of any MAB algorithm is measured by the regret it suffers. Regret is defined as the difference between the reward obtained from the proposed algorithm $\mathcal{A}$ and the reward given by a hypothetical omniscient algorithm which knows all the reward distributions. Let $\mu^* = \max\{\mu_j : 1 \leqslant j \leqslant K\}$ and $N_T(j)$ denote the optimal reward and number of pulls of arm j till time T respectively. Let $\Delta_j = \mu^* - \mu_j$, denote the sub-optimality of arm j then the expected regret of algorithm $\mathcal{A}$ is given by,

$$\begin{aligned} R_T(\mathcal{A}) = \mathbb{E}_{\mathcal{A}}\left[\sum_{t=1}^{T} \mu^* - \mu_{I_t}\right] &= \sum_{j=1}^{K} (\mu^* - \mu_j)\,\mathbb{E}_{\mathcal{A}}\left[N_T(j)\right] \\ &= \sum_{j=1}^{K} (\Delta_j)\,\mathbb{E}_{\mathcal{A}}\left[N_T(j)\right]. \end{aligned} \tag{1}$$

The expectation is taken with respect to the randomization involved in the pulling strategy given by the algorithm $\mathcal{A}$. For any arm j, since $\Delta_j \geqslant 0$, the regret of any algorithm $\mathcal{A}$ increases as number of rounds T increases. A trivial allocation policy (allocating a random arm at every time instance) can incur linear regret with $R_T(\mathcal{A}) = \Omega(T)$. In this paper, we will look at algorithms that incur sub-linear regret i.e. $R_T(\mathcal{A}) = o(T)$. We formally define sub-linear regret as follows:

Definition 1 (Sub-linear Regret). *An algorithm $\mathcal{A}$ is said to have sub-linear regret if for any $c > 0$, $\exists T_0$ such that:*

$$T > T_0 \implies 0 \leqslant R_T(\mathcal{A}) < cT.$$

We next provide a lower bound on the regret which says that any algorithm with sub-linear regret has to suffer $\Omega(\ln T)$ regret. Table 1 lists the notations that are used in this section.

Notation	Description
T	Time horizon
K	Number of arms
$\mathcal{K}$	Set of arms $\mathcal{K} = \{1, 2, \dots, K\}$
μ_i	Expected reward of arm i
$\mu^* = \max_i \mu_i$	Expected reward of the best arm
$\Delta_i = \mu^* - \mu_i$	Sub-optimality of arm i
I_t	Index of the arm pulled at time t
$X_{I_t}(t)$	Reward at time t
$N_i(t) = \sum_{s=1}^{t} \mathbb{1}\{I_s = i\}$	Total number of pulls of arm i till time t
$S_i(t) = \sum_{s=1}^{t} \mathbb{1}\{I_s = i\}X_i(s)$	Cumulative reward of an arm i till time t

Table 1: Notation Table

2.2 Lower Bound on Regret

We now present a detailed derivation of lower bound on regret which any MAB algorithm must suffer. The original proof appears in Lai and Robbins [17], we provide this proof with more details filled in. The lower bound on regret in this section is meaningful only when the reward distributions of the arms satisfy certain properties. These properties are general and capture most of the well known distributions of discrete or continuous random variable.

Let the reward distribution of each arm be such that it is specified by its density function $f(x;\rho_j)$ with respect to some measure ω, where $f(.;.)$ is known and ρ_j is an unknown parameter belonging to some set Υ. The original paper relies on a measure theoretic specification as it offers unified treatment to all classes of distributions. For readers unfamiliar with measure theory, the density $f(.;.)$ can be interpreted as being the probability density function (or probability mass function) and the integrations (or summations for discrete RVs) being taken in usual Riemann sense.

1. The parameter space Υ is such that $\forall \rho_j \in \Upsilon$, $\forall j \in \mathcal{K}$, $\int_{-\infty}^{\infty} |x| f(x;\rho_j) d\omega(x) < \infty$.
2. $\forall \lambda \in \Upsilon$ and $\forall \delta > 0$, $\exists \lambda' \in \Upsilon$ such that $\mu(\lambda) < \mu(\lambda') < \mu(\lambda) + \delta$. $\mu(\lambda)$ denotes the mean reward of an arm with parameter λ.
3. For any $\rho, \lambda \in \Upsilon$, let $I(\rho_j, \lambda)$ denote the Kullback-Leibler number, defined as
$$I(\rho_j,\lambda) = \int_{-\infty}^{\infty} \left[\ln\left(\frac{f(x;\rho_j)}{f(x;\lambda)} \right) \right] f(x;\rho_j) d\omega(x).$$
We have $\forall \epsilon > 0$ and $\forall \rho_j \in \Upsilon, \lambda$ such that $\mu(\lambda) > \mu(\rho_j)$, $\exists \delta = \delta(\epsilon, \rho_j, \lambda) > 0$ for which $|I(\rho_j,\lambda) - I(\rho_j,\lambda')| < \epsilon$ whenever $\mu(\lambda) \leqslant \mu(\lambda') \leqslant \mu(\lambda) + \delta$.

For any reward distributions which meet the above conditions, the lower bound on regret is given by the following theorem.

Theorem 1. *Let the reward distribution of arms satisfy the aforementioned assumptions on parameter space Υ. Let $\rho = (\rho_1, \rho_2, \dots \rho_K)$ be a parameter vector, and $\mathcal{A}$ be any allocation strategy that satisfies: $\forall \rho_j \in \Upsilon$ $\forall j \in \mathcal{K}$, as $T \to \infty$ and $R_T(\mathcal{A}) = o(T^a)$ for every $a > 0$. Denote the best arm with parameter ρ^* and expectation $\mu^* = \mu(\rho^*)$. Assume there exists an arm j such that $\rho_j \neq \rho^*$ and $\mu(\rho_j) \neq \mu^*$, then we have*

$$\liminf_{T\to\infty} \frac{R_T(\mathcal{A})}{\ln(T)} \geqslant \sum_{j:\mu(\rho_j)<\mu^*} \frac{\mu^* - \mu(\rho_j)}{I(\rho_j, \rho^*)}. \tag{2}$$

Proof: WLOG, fix ρ such that arm 2 is a best arm with parameter ρ_2 i.e. $\mu(\rho_2) > \mu(\rho_1)$ and $\mu(\rho_2) \geqslant \mu(\rho_i)$ for $3 \leqslant i \leqslant k$. Fix any $0 < \delta < 1$, by the assumption 2 and 3 on the distribution of rewards we can choose $\lambda \in \Upsilon$ such that

$$\mu(\lambda) > \mu(\rho_2) \text{ and } |I(\rho_1,\lambda) - I(\rho_1,\rho_2)| < \delta I(\rho_1,\rho_2). \tag{3}$$

Define a new parameter vector $\gamma = (\lambda, \rho_2, \dots, \rho_k)$ such that under γ, arm 1 is the unique best arm. The original arm distribution given by ρ and the newly constructed arm distribution given by γ forms the basis of arguments in the proof. We denote the probability under respective distributions as $\mathbb{P}_\rho$ and $\mathbb{P}_\gamma$. A similar notation is used for expectations as well. Let $N_i(T)$ denote the number of times an arm i is pulled in T pulls overall by the allocation rule $\mathcal{A}$. During $T - N_1(T)$ trials, a sub-optimal arm gets pulled and these trials contribute to the regret $R_T(\mathcal{A})$. Fix $0 < a < \delta$. Since $R_T(\mathcal{A}) = o(T^a)$, for rewards distributed as per γ, we have the following.

$$\mathbb{E}_\gamma(T - N_1(T)) = \sum_{h\neq 1} \mathbb{E}_\gamma(N_h(T)) = o(T^a).$$

Consider the event $\{N_1(T) < (1-\delta)(\ln(T))/I(\rho_1,\lambda)\}$, we have,

$$\mathbb{P}_\gamma \left\{ N_1(T) < (1-\delta)\frac{\ln(T)}{I(\rho_1,\lambda)} \right\} = \mathbb{P}_\gamma \left\{ T - N_1(T) \geqslant T - (1-\delta)\frac{\ln(T))}{I(\rho_1,\lambda)} \right\}.$$

Through an application of Markov's inequality we have,

$$(T - O(\ln(T)))\mathbb{P}_\gamma \left\{ N_1(T) < (1-\delta)\frac{\ln(T)}{I(\rho_1,\lambda)} \right\} \leqslant \mathbb{E}_\gamma(T - N_1(T)) = o(T^a).$$

The allocation rule $\mathcal{A}$ has access only to the reward realizations of arms, and not the reward distributions. Let $Y_1, Y_2, \dots$ denote successive realizations from arm 1 (sub-optimal arm). This lets us argue about a lower bound on regret.

We denote $L_m = \sum_{i=1}^{m} \ln(f(Y_i;\rho_1)/f(Y_i,;\lambda))$ and consider an event,

$$C_T = \left\{N_1(T) < (1-\delta)\tfrac{\ln(T)}{I(\rho_1,\lambda)} \text{ and } L_{N_1(T)} \leqslant (1-a)\ln(T)\right\}.$$

Through the inequality previously mentioned,

$$\mathbb{P}_\gamma\{C_T\} = o(T^{a-1}).$$

Observe the following identity

$$\begin{aligned}
&\mathbb{P}_\gamma\{N_1(T) = T_1, \ldots, N_k(T) = T_k \text{ and } L_{T_1} \leqslant (1-a)\ln(T)\} \\
&= \int_{\{N_1(T)=T_1,\ldots,N_k(T)=T_k \text{ and } L_{T_1} \leqslant (1-a)\ln(T)\}} \prod_{i=1}^{T_1} \frac{f(Y_i;\lambda)}{f(Y_i;\rho)} d\mathbb{P}_\rho \\
&= \int_{\{N_1(T)=T_1,\ldots,N_k(T)=T_k \text{ and } L_{T_1} \leqslant (1-a)\ln(T)\}} e^{-L_{T_1}} d\mathbb{P}_\rho \\
&\geqslant \exp(-(1-a)\ln(T)) \times \\
&\qquad \mathbb{P}_\rho\{N_1(T) = T_1, \ldots, N_k(T) = T_k \text{ and } L_{T_1} \leqslant (1-a)\ln(T)\} \\
&= T^{-(1-a)}\mathbb{P}_\rho\{N_1(T) = T_1, \ldots, N_k(T) = T_k \text{ and } L_{T_1} \leqslant (1-a)\ln(T)\}.
\end{aligned}$$

The first equality is due to the implicit assumption that the allocation strategy can only depend on the reward realization of an arm i it can observe via pulling that arm and possibly some inherent randomness in the allocation. The reward realization of the arm however is arising from some fixed underlying distribution which is assumed to have a density of known form but unknown parameter.

C_T is a disjoint of events of the form discussed in the identity above with $T_1 + T_2 + \ldots + T_k = T$ and $T_1 < (1-\delta)\ln(T)/I(\rho_1,\lambda)$, it now follows that as $T \to \infty$, we have

$$\mathbb{P}_\rho\{C_T\} \leqslant T^{1-a}\mathbb{P}_\gamma\{C_T\} \to 0. \tag{4}$$

By strong law of large numbers, $L_m/m \to I(\rho_1,\lambda)$, and $\max_{i \leqslant m} L_i/m \to I(\rho_1,\lambda)$ almost surely under $\mathbb{P}_\rho$. Therefore, $\mathbb{P}_\rho\{\max_{i\leqslant m} L_i/m > I(\rho_1,\lambda)\} \to 0$. We note the following, denote $m = (1-\delta)\ln(T)/I(\rho_1,\lambda)$, we have,

$$\begin{aligned}
&\left\{L_i > (1-a)\ln(T) \text{ for some } i < \frac{(1-\delta)(\ln(T))}{I(\rho_1,\lambda)}\right\} = \\
&\left\{\frac{L_i I(\rho_1,\lambda)}{(1-\delta)\ln(T)} > \frac{(1-a)I(\rho_1,\lambda)}{(1-\delta)} \text{ for some } i < \frac{(1-\delta)(\ln(T))}{I(\rho_1,\lambda)}\right\} = \\
&\left\{\frac{L_i}{m} > \frac{(1-a)I(\rho_1,\lambda)}{(1-\delta)} \text{ for some } i < m \text{ where } m = \frac{(1-\delta)(\ln(T))}{I(\rho_1,\lambda)}\right\} \subseteq \\
&\left\{\max_{i\leqslant m}\frac{L_i}{m} > \frac{(1-a)I(\rho_1,\lambda)}{(1-\delta)} \text{ where } m = \frac{(1-\delta)(\ln(T))}{I(\rho_1,\lambda)}\right\} \subseteq \\
&\left\{\max_{i\leqslant m}\frac{L_i}{m} > I(\rho_1,\lambda) \text{ where } m = \frac{(1-\delta)(\ln(T))}{I(\rho_1,\lambda)}\right\}. \qquad \text{(as } \delta < a\text{)}
\end{aligned}$$

As the probability under ρ of the last event goes to zero, we have,

$$\mathbb{P}_\rho\{L_i > (1-a)\ln(T) \text{ for some } i < (1-\delta)(\ln(T))/I(\rho_1,\lambda)\} \to 0. \tag{5}$$

From Equation (4) and Equation (5) we have that

$$\lim_{T\to\infty} \mathbb{P}_\rho\left\{N_1(T) < \frac{(1-\delta)(\ln(T))}{I(\rho_1,\lambda)}\right\} = 0.$$

In other words,

$$\lim_{T\to\infty} \mathbb{P}_\rho\left\{N_1(T) < \frac{(1-\delta)(\ln(T))}{(1+\delta)I(\rho_1,\rho_2)}\right\} = 0.$$

Thus, we have,

$$\liminf_{T\to\infty} \mathbb{E}_\rho \frac{N_1(T)}{\ln(T)} \geqslant \frac{1}{I(\rho_1,\rho_2)}. \tag{6}$$

The above inequality is true for any j such that $\mu(\rho_j) < \mu^*$ then we have regret given by

$$R_T(\mathcal{A}) = \sum_{j:\mu(\rho_j)<\mu^*} (\mu^* - \mu(\rho_j))\mathbb{E}_\rho N_j(T).$$

By Equation (6) we are done. ■

When the reward distribution of the arms are Bernoulli, a simpler proof for the lower bound on regret is given in the survey article by Bubeck and Cesa-Bianchi [8].

3 Stochastic MAB Algorithms

We now present some of the algorithms for providing an allocation strategy to achieve sub-linear regret. We will first look at a very simple strategy also known as exploration separated algorithms where all the arms are explored for some number of rounds and then the best arm based on the rewards obtained in the initial rounds is chosen for the rest of the rounds. The number of exploration rounds are fixed in advance and do not depend on the rewards obtained so far. Thus, these strategies do not give good regret guarantees. We will then present the UCB family of algorithms. These algorithms maintain an index for each arm and the arm with higher index is pulled at every time. The other type of algorithms are based on the Bayesian approach where a distribution is maintained over each arm. A sample from this distribution is drawn and the arm with highest sample is pulled. One algorithm based on Bayesian approach is known as Thompson Sampling algorithm and was proposed in the year 1933 by Thompson [26]. Until recent time, there were no theoretical guarantees on the regret of Thompson sampling algorithm. Some recent work has proved the theoretical guarantees on regret. Agrawal and Goyal [1] proved the regret which is of the same order as is obtained by UCB index based algorithms. A tighter bound is provided by Kauffman et. al. [16] where the authors showed that the regret achieved by Thompson sampling matches the lower bound given by Equation (2).

3.1 Frequentist Approaches

3.1.1 Exploration Separated Algorithms

Solutions to MAB problem involve designing strategies to trade-off between exploration (pulling the arm that has not been explored enough number of times) and exploitation (pulling the arm that has provided best reward so far). Exploration separated algorithms give a simple strategy for this trade-off wherein all the arms are pulled in round robin fashion for ϵT number of rounds and then, the arm with best empirical reward so far is pulled for the rest of the $(1-\epsilon)T$ number of rounds. Here, the parameter ϵ is set to minimize the regret. The algorithm is presented in Algorithm 1.

Algorithm 1: Exploration Separated Algorithm

```
Input: Time horizon $T$, exploration parameter $\epsilon$, number of arms $K$
Output: Allocation policy $\mathcal{A} = \{I_1, I_2, \dots, I_T\}$
Initialize: $t = 1, S_i = 0, N_i = 0$
• while $t < \lfloor \frac{\epsilon T}{K} \rfloor K$ do
      - for $i = 1 : K$ do
            * $I_t = i$
            * $t = t + 1$
            * $N_i = N_i + 1$
            * If click is observed, $S_i = S_i + 1$

• Let $\hat{\mu}_i = \frac{S_i}{N_i}$.
• for $t = \lfloor \frac{\epsilon T}{K} \rfloor K + 1, \cdots, T$ do
      - $I_t = i$ if $i = \arg\max_i \hat{\mu}_i$
```

Let N_i denote the number of exploration rounds each agent faces. Thus, $N_i = \lfloor \frac{\epsilon T}{K} \rfloor$ $\forall i \in \mathcal{K}$. Let $c_i = \sqrt{\frac{2 \ln T}{N_i}}$ and $j = \arg\max_i \hat{\mu}_i$. Also denote $i^* = \arg\max_i \mu_i$ and $\mu^* = \max_i \mu_i$ as the index and the mean reward of the optimal arm respectively. For notational convenience, we assume $\lfloor \frac{\epsilon T}{K} \rfloor = \frac{\epsilon T}{K}$. Regret in this setting is given as:

$$
\begin{aligned}
R_T(\mathcal{A}) = \mathbb{E}_{\mathcal{A}}\left[\sum_{t=1}^{T} \mu^* - \mu_{I_t}\right] &= \sum_{i=1}^{K} (\mu^* - \mu_i)\,\frac{\epsilon T}{K} + (1-\epsilon)T\,(\mu^* - \mu_j) \\
&\leqslant \epsilon T\mu^* + T(\mu^* - \mu_j).
\end{aligned}
$$

Here $j = \arg\max_i \hat{\mu}_i$. From Hoeffding's inequality, it can be seen that for any $i \in \mathcal{K}$:

$$\mathbb{P}\{\mu_i > \hat{\mu}_i + c_i\} \leqslant T^{-4}. \tag{7}$$

$$\mathbb{P}\{\mu_i < \hat{\mu}_i - c_i\} \leqslant T^{-4}. \tag{8}$$

Thus, we have:

$$
\begin{aligned}
\mu^* - \mu_j &\leqslant \mu^* - \hat{\mu}_j + c_j && \text{(with probability at least } 1 - T^{-4}\text{)} \\
&\leqslant \mu^* - (\hat{\mu}_j + c_j) + 2c_j && \text{(adding and subtracting } c_j\text{)} \\
&\leqslant \mu^* - (\hat{\mu}_{i^*} + c_{i^*}) + 2c_j && (\because j = \arg\max_i \hat{\mu}_i \text{ and } c_j = c_{i^*}) \\
&\leqslant \hat{\mu}_{i^*} + c_{i^*} - (\hat{\mu}_{i^*} + c_{i^*}) + 2c_j. && \text{(with probability at least } 1 - T^{-4}\text{)}
\end{aligned}
$$

Thus, expected regret can be bounded by:

$$
\begin{aligned}
\mathbb{E}[R_T(\mathcal{A})] &\leqslant \epsilon T\mu^* + 2Tc_j(1 - T^{-4}) + T^{-4}T \\
&\leqslant \epsilon T\mu^* + 2T\sqrt{\frac{2K\ln T}{\epsilon T}} + T^{-3} \\
&\leqslant \epsilon T\mu^* + 2T^{1/2}\sqrt{\frac{2K\ln T}{\epsilon}} + T^{-3}.
\end{aligned}
$$

If $\epsilon = T^{-1/3}$, then we get $\mathbb{E}[R_T(\mathcal{A})] = O(T^{2/3})$.

3.1.2 Upper Confidence Bound (UCB) Based Algorithms

One of the widely used family of algorithms for multi-armed bandit is the UCB family. The first of these algorithms, UCB1, was introduced by [2] for the scenario where the rewards of the arms were drawn from a distribution with bounded support. These algorithms, however, are fairly general and can be applied to other scenarios as well.

At every time instant t, UCB algorithms in general pull the arm with the maximum value of the index $\text{ucb\_ind}(i,t) = \hat{\mu}_{i,t-1} + c_{i,t-1}$, where $\hat{\mu}_{i,t-1} = S_i(t-1)/N_i(t-1)$ is the empirically estimated mean of the arm i after $t-1$ pulls and $c_{i,t-1}$ is the associated confidence bound. Intuitively, at a given time instant t, a lower value of $N_i(t-1)$ implies that the arm i has not been explored much and one would like to explore the arm more, thereby setting $c_{i,t-1}$ to be large. The trade-off between exploration and exploitation is captured by the term $c_{i,t-1}$. The structure of a general UCB based algorithm is described in Algorithm 2.

The choice of the index $c_{i,t-1}$ depends on the underlying reward distributions of the arms. We now list a few such algorithms that have been commonly used in practice.

1. **UCB1 Algorithm**

   When the support of the distributions is $[0,1]$, and value of $c_{i,t}$[1] $= \sqrt{\frac{2\ln(t)}{N_i(t)}}$, this version of the UCB algorithm is referred to as UCB1 [2]. The regret of UCB1 algorithm is proved to be asymptotically optimal and is given by the following lemma,

   Lemma 1. *For all $K > 1$, if UCB1 policy is run on K arms with arbitrary reward distributions with means $\mu_1, \mu_2, \ldots, \mu_K$ and with support in $[0,1]$ then its expected regret after T number of pulls is at most,*

   $$\mathbb{E}[\mathcal{R}_T(\mathcal{A})] \leqslant 8 \sum_{i:\Delta_i > 0} \left(\frac{\ln(T)}{\Delta_i}\right) + \left(1 + \frac{\pi^2}{3}\right) \sum_{i=1}^{K} \Delta_i.$$

[1] The value of $c_{i,t}$ is fixed carefully so as to achieve desired regret bounds. In Item 3, we will see a principled way of selecting $c_{i,t}$

```
Algorithm 2: UCB based Algorithm
Input: Number of arms $K$, Time horizon $T$
Output: Allocation policy $\mathcal{A} = \{I_1, I_2, \ldots, I_n\}$
  • Play every arm once and set $N_i(K) = 1, I_i = i$, for $i = 1 \cdots K$
  • Observe $X_{I_i}(i)\ \forall i \in \mathcal{K}$ and set $S_i(K) = X_{I_i}(i)$
  • $\hat{\mu}_{i,K} = \frac{S_i(K)}{N_i(K)}$
  • for $t = K+1, \cdots, T$ do
        - Allocate $I_t = \arg\max_{i=1,\cdots,K} \hat{\mu}_{i,N_i(t-1)} + c_{i,t-1}$
        - Observe $X_{I_t}(t)$
        - Update
            * $N_{I_t}(t) = N_{I_t}(t-1) + 1$
            * $S_{I_t}(t) = S_{I_t}(t-1) + X_{I_t}(t)$
            * $\forall j \neq I_t$
                · $N_j(t) = N_j(t-1)$
                · $S_j(t) = S_j(t-1)$
            * $\hat{\mu}_{i,N_i(t)} = \frac{S_i(t)}{N_i(t)},\ \forall i \in \mathcal{K}$
```

Proof: Let $N_{i^*}(t)$ denote the number of pulls of the optimal arm in t trials. Let $\hat{\mu}_{i,t}$ denote the empirical mean of the rewards from arm i till time t. We upper bound $N_i(T)$, the number of times a sub-optimal arm i is pulled, in any sequence of T plays. i.e.

$$N_i(T) = 1 + \sum_{t=K+1}^{T} \mathbb{1}\{I_t = i\}.$$

Let l be an arbitrary positive integer and $\bar{c}_{t,s} = \sqrt{\frac{2\ln(t)}{s}}$ be the exploration term with s number of pulls among t rounds. Moreover, when $I_t = i$, then $\hat{\mu}_{i,N_i(t-1)} + \bar{c}_{t-1,N_i(t-1)} \geqslant \hat{\mu}_{i^*,N_{i^*}(t-1)} + \bar{c}_{t-1,N_{i^*}(t-1)}$. Thus, we have,

$$\begin{aligned}
N_i(T) &\leqslant l + \sum_{t=K+1}^{T} \mathbb{1}\{I_t = i, N_i(t-1) \geqslant l\} \\
&\leqslant l + \sum_{t=K+1}^{T} \mathbb{1}\left\{\hat{\mu}_{i^*,N_{i^*}(t-1)} + \bar{c}_{t-1,N_{i^*}(t-1)} \leqslant \hat{\mu}_{i,N_i(t-1)} + \bar{c}_{t-1,N_i(t-1)}, N_i(t-1) \geqslant l\right\} \\
&\leqslant l + \sum_{t=K+1}^{T} \mathbb{1}\left\{\min_{0<s<t} \hat{\mu}_{i^*,s} + \bar{c}_{t-1,s} \leqslant \max_{l\leqslant s_i<t} \hat{\mu}_{i,s_i} + \bar{c}_{t-1,s_i}\right\} \\
&\leqslant l + \sum_{t=1}^{\infty} \sum_{s=1}^{t-1} \sum_{s_i=l}^{t-1} \mathbb{1}\left\{\hat{\mu}_{i^*,s} + \bar{c}_{t,s} \leqslant \hat{\mu}_{i,s_i} + \bar{c}_{t,s_i}\right\}.
\end{aligned}$$

Now observe that $\hat{\mu}_{i^*,s} + \bar{c}_{t,s} \leqslant \hat{\mu}_{i,s_i} + \bar{c}_{t,s_i}$ implies that at least one of the following must hold,

$$\hat{\mu}_{i^*,s} \leqslant \mu^* - \bar{c}_{t,s}. \tag{9}$$

$$\hat{\mu}_{i,s_i} \geqslant \mu_i + \bar{c}_{t,s_i}. \tag{10}$$

$$\mu^* < \mu_i + 2\bar{c}_{t,s_i}. \tag{11}$$

We bound the probability of events (Equation (9)) and (Equation (10)), using Chernoff bound,

$$\mathbb{P}\{\hat{\mu}_{i^*,s} \leqslant \mu^* - \bar{c}_{t,s}\} \leqslant e^{-4\ln(t)} = t^{-4}.$$
$$\mathbb{P}\{\hat{\mu}_{i,s_i} \geqslant \mu_i + \bar{c}_{t,s_i}\} \leqslant e^{-4\ln(t)} = t^{-4}.$$

For $s_i = [(8\ln(T))/\Delta_i^2]$, the event (Equation (11)) is false. In fact

$$\mu^* - \mu_i - 2c_{t,s_i} = \mu^* - \mu_i - 2\sqrt{2\ln(t)/s_i} \geqslant \mu^* - \mu_i - \Delta_i = 0.$$

For $l \geqslant (8\ln(T)/\Delta_i^2)$ we get,

$$\begin{aligned}
\mathbb{E}[N_i(T)] &\leqslant \frac{8\ln(T)}{\Delta_i^2} + \sum_{t=1}^{\infty}\sum_{s=1}^{t-1}\sum_{s_i=8\ln(T)/\Delta_i^2}^{t-1} \left(\mathbb{P}\{\hat{\mu}_{i^*,s} \leqslant \mu^* - c_{t,s}\} + \mathbb{P}\{\hat{\mu}_{i,s_i} \geqslant \mu_i + c_{t,s_i}\}\right) \\
&\leqslant \frac{8\ln(T)}{\Delta_i^2} + \sum_{t=1}^{\infty}\sum_{s=1}^{t}\sum_{s_i=1}^{t} 2t^{-4} \\
&\leqslant \frac{8\ln(T)}{\Delta_i^2} + 1 + \frac{\pi^2}{3}.
\end{aligned}$$

■

2. **UCB-Normal Algorithm**

When the rewards of the arms are drawn from normal distributions $\mathcal{N}(\mu, \sigma^2)$, the value of $c_{i,t} = \sqrt{16\frac{q_i(t) - N_i(t)\hat{\mu}_{i,t}^2}{N_i(t) - 1}\frac{\ln(t-1)}{N_i(t)}}$, where $q_i(t) = \sum_{s=1}^{t} \mathbb{1}\{I_t = i\}X_i^2(s)$ is the sum of squares of the rewards from arm i up to time t. The original UCB-Normal algorithm [2] requires each arm to be pulled at least $\lceil 8\ln T \rceil$ number of times. However, it was later proved that this condition is not required [8].

Lemma 2. *For all $K > 1$, if UCB-Normal policy is run with K arms having normal distributions with means $\mu_1, \cdots, \mu_K$ and variances $\sigma_1^2, \cdots, \sigma_K^2$, then its expected regret after any T number of plays is at most,*

$$\mathbb{E}[R_T(\mathcal{A})] \leqslant 256\ln(T) \sum_{i:\mu_i<\mu_{i^*}} \left(\frac{\sigma_i^2}{\Delta_i}\right) + \left(1 + \frac{\pi^2}{2} + 8\ln(T)\right)\sum_{i=1}^{K} \Delta_i.$$

Proof: Proof of Lemma 2 follows along similar lines as the proof of Lemma 1 and can be found in [2]. ■

3. **(α, ψ)-UCB Algorithm**

(α, ψ)-UCB strategy [8] is a fairly general algorithm to come up with the upper confidence intervals for UCB algorithms. The rewards $X_i(t)$ of arm i at time t may be sampled from any arbitrary distribution.

We first analyze the general scenario where we have samples from any arbitrary random variable Y and use the empirical mean of the samples as an estimate of the true mean of Y. For any random variable Y, assume a convex function $\psi_Y(.)$ exists, such that for all $\lambda > 0$,

$$\ln\mathbb{E}[\exp(\lambda(Y - \mathbb{E}[Y]))] \leqslant \psi_Y(\lambda) \text{ and } \ln\mathbb{E}[\exp(\lambda(\mathbb{E}[Y] - Y))] \leqslant \psi_Y(\lambda). \tag{12}$$

We also know that,

$$\mathbb{P}\{Y - \mathbb{E}[Y] > \epsilon\} = \mathbb{P}\{\exp(\lambda(Y - \mathbb{E}[Y])) > \exp(\lambda\epsilon))\} \tag{13}$$

$$\leqslant \frac{\mathbb{E}[\exp(\lambda(Y - \mathbb{E}[Y]))]}{\exp(\lambda\epsilon)} \leqslant \exp(\psi_Y(\lambda) - \lambda\epsilon). \tag{14}$$

The above inequality arises as a result of Markov inequality, and Equation (12). In order to obtain tight confidence intervals, we require the value $\exp(\psi_Y(\lambda) - \lambda\epsilon)$ to be as low as possible for a given ϵ. Define the function $\psi_Y^*(\epsilon) = \sup_{\lambda\geqslant 0} \lambda\epsilon - \psi_Y(\lambda)$. $\psi_Y^*(\epsilon)$ is known as the Legendre-Fenchel transform of $\psi_Y(.)$ and thereby, $\exp(-\psi_Y^*(\epsilon))$ is a tight confidence interval.

We now analyze the scenario where our random variable of interest is the mean of t iid random variables Y_i where $i = 1, \cdots, t$ with true mean γ, provided an appropriately defined convex function $\psi_{Y_i}(.)$ exists. By setting $Y = \sum_{i=1}^{t} Y_i/t$ and $\mathbb{E}[Y] = \gamma$ in Equation (13), we get,

$$\mathbb{P}\left\{\sum_{i=1}^{t}\frac{Y_i}{t} - \gamma > \epsilon\right\} = \mathbb{P}\left\{\sum_{i=1}^{t} Y_i - t\gamma > t\epsilon\right\} \leqslant \frac{\mathbb{E}[\exp(\lambda(\sum_{i=1}^{t} Y_i - t\gamma))]}{\exp(\lambda t\epsilon)} \tag{15}$$

$$= \frac{\mathbb{E}[\prod_{i=1}^{t}\exp(\lambda(Y_i - \gamma))]}{\exp(\lambda t\epsilon)} = \prod_{i=1}^{t}\frac{\mathbb{E}[\exp(\lambda(Y_i - \gamma))]}{\exp(\lambda t\epsilon)} \tag{16}$$

$$\leqslant \prod_{i=1}^{t}\exp(\psi_{Y_i}(\lambda)) - \exp(\lambda t\epsilon) = \exp(t(\psi_{Y_1}(\lambda) - \lambda\epsilon)) \tag{17}$$

$$\leqslant \exp(-t\psi_{Y_1}^*(\epsilon)). \tag{18}$$

Parameter	$\psi(\lambda)$	α	$\psi^*(\epsilon)$
UCB1	$\lambda^2/8$	4	$2\epsilon^2$
UCB-Normal	$\lambda^2\sigma^2/2$	8	$\epsilon^2/(2\sigma^2)$

Table 2: (α, ψ) UCB Parameter setting

Also, by symmetry,

$$\mathbb{P}\left\{\gamma - \sum_{i=1}^{t} \frac{Y_i}{t} > \epsilon\right\} \leqslant \exp(-t\psi^*_{Y_1}(\epsilon)). \tag{19}$$

Let δ be the desired probability that the true mean lies within ϵ confidence interval around the empirical mean. To achieve this, we can set $\exp(-t\psi^*_{Y_1}(\epsilon)) = \delta$, thereby $\epsilon = (\psi^*_{Y_1})^{-1}\left(\frac{1}{t}\ln\frac{1}{\delta}\right)$.
In the MAB scenario, we use the empirical mean of the rewards of the arms as the estimator of the true mean reward. The rewards from each arm is an iid random variable and therefore, the variable of interest $Y_i = S_i(t)/N_i(t)$, the empirical mean of the rewards from arm i upto time t. Therefore $\gamma = \mu_i$. (α, ψ)-UCB algorithm first involves computing the function $\psi_{X_i(1)}(.)$. for the reward distributions $X_i(1)$. The only requirement for application of ψ-UCB is the existence of a convex function $\psi_{X_i(1)}(.)$. Further, for a required confidence interval δ, $\epsilon = (\psi^*_{X_i(1)})^{-1}\left(\frac{1}{N_i(t)}\ln\frac{1}{\delta}\right)$ as only $N_i(t)$ samples are observed from arm i. By Equation (19), with probability at least $1-\delta$,

$$\frac{S_i(t)}{N_i(t)} + (\psi^*_{X_i(1)})^{-1}\left(\frac{1}{N_i(t)}\ln\frac{1}{\delta}\right) > \mu_i. \tag{20}$$

This naturally leads to an index based scheme, whereby setting $\delta = t^{-\alpha}$ yields us,

$$I_t \in \operatorname*{arg\,max}_{i=1,\cdots,K} \frac{S_i(t-1)}{N_i(t-1)} + (\psi^*_{X_i(1)})^{-1}\left(\frac{\alpha\ln(t)}{N_i(t-1)}\right). \tag{21}$$

Thus, setting $c_{i,t} = (\psi^*_{X_i(1)})^{-1}\left(\frac{\alpha\ln(t)}{N_i(t-1)}\right)$, it can be seen that UCB1 and UCB-Normal [2] are special instances of (α, ψ)-UCB for appropriately selected α values and $\psi(.)$ functions as shown in Table 2.

Lemma 3. *If there exists a convex function $\psi(.)$ satisfying Equation (12), the expected regret of (α, ψ) UCB algorithm $\mathcal{A}$ with $\alpha > 2$ satisfies,*

$$\mathbb{E}[R_T(\mathcal{A})] \leqslant \textstyle\sum_{i:\Delta_i>0}\left(\frac{\alpha\Delta_i}{\psi^*(\Delta_i/2)} + \frac{\alpha}{\alpha-2}\right).$$

Proof: Proof of Lemma 3 follows along similar lines as the proof of Lemma 1. ■

3.1.3 KL-UCB

KL-UCB [11] is an algorithm based on the upper confidence bound of the empirically estimated parameters of the reward distribution. It was originally proposed for the scenario when the reward distribution is Bernoulli. The idea in KL-UCB is to first estimate the mean μ of the reward distributions of the arms from the samples. Then, for each arm i, the set of possible values whose divergence from the empirical mean is bounded above by a factor $(\ln(t) + c\ln\ln(t))/N_i(t)$ is computed and therein the maximum of this set of parameters, which we denote by q_i, is considered. Finally the arm with a maximum value of q_i is selected in the round $t+1$. The complete algorithm is provided in Algorithm 3.

KL-UCB matches the lower bound for the regret given by Lai and Robbins [17] asymptotically, when the reward distribution is Bernoulli as per the following with $c = 3$,

$$\limsup_{n\to\infty} \frac{\mathbb{E}(R_T(\mathcal{A}))}{\ln(T)} \leqslant \sum_{i:\mu_i \leqslant \mu_{i^*}} \frac{\mu_{i^*} - \mu_i}{D(\mu_i||\mu_{i^*})}. \tag{22}$$

KL-UCB can be adapted to fairly general reward distributions by a careful selection of the divergence function $D(.||.)$.

1. In the case of Bernoulli rewards, a divergence function which satisfies the Large Deviations Principle may be used.
2. When the rewards are distributed from an exponential family, we have, $D(x||y(\rho)) = \sup_{\lambda}\{\lambda x - \ln(\mathbb{E}_{\rho}[\exp(\lambda X)])\}$. In the case of exponentially distributed rewards $D(x||y) = \frac{x}{y} - 1 - \ln(\frac{x}{y})$.
3. For Poisson distributed rewards, the divergence function is chosen to be $D(x||y) = y - x + x\ln(\frac{x}{y})$.

Algorithm 3: KL-UCB : Bernoulli distributed rewards

```
Input: Time Horizon= $T$, No of arms = $K$, Reward Distribution= $X_i \in \{0,1\} \forall i \in \mathcal{K}$,c
Output: Allocation policy $\mathcal{A} = \{I_1, I_2, \ldots, I_T\}$
for t=1:K do
    • Play arm:
      $I_t = t$
    • Observe reward:
      $X_{I_t}(t) \in \{0,1\}$
    • Update:
        - $S_{I_t}(K) = X_{I_t}(t)$
        - $N_{I_t}(K) = 1$
for t=K+1: T do
    • Play arm, $I_t = \arg\max_i q_i = \max\{q \in [0,1] : (N_i(t-1))KL(\frac{S_i(t-1)}{N_i(t-1)}, q) \leqslant \ln(t) + c\ln\ln(t)\}$
    • Observe reward:
      $X_{I_t}(t) \in \{0,1\}$
    • Update
      $S_{I_t}(t) = S_{I_t}(t-1) + X_{I_t}(t)$
      $N_{I_t}(t) = N_{I_t}(t-1) + 1$
```

3.2 Bayesian Approach

MAB algorithms deal with pulling the arms with unknown reward distribution parameterized by ρ. The task here is to decide the strategy $(I_t)_{t\geqslant 1}$ to pull a single arm sequentially for T times. If the expected cumulated regret is used as a measure of performance, as in the frequentist view, it can be shown that Bayesian techniques such as Thompson sampling and Bayes-UCB are also asymptotically optimal. In the frequentist viewpoint, only the past trials and the resulting rewards are used to estimate the unknown parameters. Whereas in the Bayesian view, we begin with a prior distribution over each of the unknown parameters. After every trial, the prior is updated based on the observed reward. Note that the randomness in the model comes from the prior over the parameters as well as the reward distributions.

In this section, we explain two Bayesian algorithms, namely Thompson sampling and Bayes-UCB. We know that the UCB family is asymptotically optimal when the expected cumulated reward is used as a performance measure. We evaluate the Bayesian techniques on the same measure and see that both Thompson sampling and Bayes-UCB are also asymptotically optimal. Bayesian techniques are generally easy to implement and are more robust to delayed feedback, that is, when the rewards are not obtained instantly. This is due to the fact that the extra randomization introduced by the prior distributions alleviates the regret introduced due to sub-optimal arm pull in a batch processing setting by a deterministic rule (such as the UCB family).

3.2.1 Thompson Sampling

Of late, Thompson sampling [26] has gained significant attention due its simplicity, empirical efficacy and robustness to delayed feedback. We describe Thompson sampling for the scenario where the rewards follow a Bernoulli distribution. We denote by $S_i(t)$ and $F_i(t)$ the number of successes and failures produced by arm i in t trials of the algorithm respectively. As in all Bayesian methods, we begin with the prior distribution over the parameters and after every trial t, we use $S_i(t)$ and $F_i(t)$ to obtain

Algorithm 4: Thompson Sampling for Bernoulli Bandits

```
Input: Time Horizon $T$, number of arms $K$
Output: Allocation policy $\mathcal{A} = \{I_1, I_2, \ldots, I_T\}$
Initialize: $S_i(0) = F_i(0) = 0 \; \forall i \in \mathcal{K}$
for t= 1 : T do
    • Sample:
      $\lambda_i(t) \sim Beta(1 + S_i(t-1), 1 + F_i(t-1)) \; \forall i \in \mathcal{K}$
    • Play arm:
      $I_t = \arg\max_i \lambda_i(t)$
    • Observe Reward:
      $X_{I_t}(t) \in \{0, 1\}$
    • Update:
      $S_{I_t}(t) = S_{I_t}(t-1) + X_{I_t}(t)$
      $F_{I_t}(t) = F_{I_t}(t-1) + 1 - X_{I_t}(t)$
      $S_i(t) = S_i(t-1), \; F_i(t) = F_i(t-1) \; \forall i \neq I_t$
```

the posterior distributions over the parameters. When the reward distribution follows a Bernoulli distribution, the posterior distribution over the parameter follows a Beta distribution. A sample is drawn from the updated posterior distribution over all the arms. The arm corresponding to the largest value of the sample is selected as the arm to be pulled next. The algorithm is provided in Algorithm 4.

Recently there has been a surge in research around making Thompson Sampling algorithm work for any distribution in general [1]. Theoretical guarantees given by Agrawal et. al. [1] and Kaufmann et. al. [16], have proved that Thompson sampling does better than UCB in terms of the regret and reaches close to the information theoretic bound in [17].

Lemma 4. *[16] For any $\epsilon > 0$, there exists a problem dependent constant $C(\epsilon, \mu_1, \mu_2, \ldots, \mu_K)$ such that the regret of Thompson sampling algorithm $\mathcal{A}$ is given as:*

$$R_T(\mathcal{A}) \leqslant (1+\epsilon) \sum_{i \in \mathcal{K}, \mu_i \neq \mu^*} \frac{\Delta_i(\ln T + \ln\ln T)}{D(\mu_i || \mu^*)} + C(\epsilon, \mu_1, \mu_2, \ldots, \mu_K).$$

Here, $D(\mu_i||\mu^*)$ denotes the Kullback-Leibler divergence between the two Bernoulli distributions ρ_i and ρ^* with parameters μ_i and μ^* respectively. It can be derived by Pinsker's inequality that $2D(\mu_i||\mu^*) > \Delta_i^2$. Thus, the regret of Thompson sampling algorithm is more closer to lower bound provided in Section 2.2 as compared to UCB based algorithms.

3.2.2 Bayes-UCB

Bayes-UCB Algorithm [16] is also an algorithm devised to take into account any prior information of the reward distributions under consideration. Let the prior over the parameters of the reward distributions of the arms be denoted as $\pi_1^0, \cdots, \pi_K^0$. Upon observing the rewards of the arms up to time t, the distributions over the parameters get revised to the posterior distributions $\pi_1^t, \cdots, \pi_K^t$. Let the associated posterior reward distributions be denoted as $\lambda_1^t, \cdots, \lambda_K^t$.

At every time step t, through the Bayesian framework, the posterior distributions $\pi_{I_t}^t, \lambda_{I_t}^t$ are updated. For every distribution $\lambda_i^t, i = 1, \cdots, K$, let $Q(\alpha, \lambda_i^t)$ denote the quantile function for an appropriate choice of α. In general $Q(p, D)$ returns the x such that $\mathbb{P}_D\{Y < x\} = p$, where $\mathbb{P}_D\{Y < x\}$ denotes the probability that a random variable Y with distribution D takes a value less than x. At every time step t, $q_i(t) = Q(1-\alpha, \lambda_i^t)$ can be computed for all the arms i. Intuitively an arm i will yield a reward lower than $q_i(t)$ with a probability of $1-\alpha$. Bayes-UCB selects the arm with the maximum value of $q_i(t)$. The value of α is fixed as $1/(t(\ln(T))^c)$, where c is a constant, in order to achieve suitable regret bounds. The Bayes-UCB algorithm is provided in Algorithm 5.

Lemma 5. *For any $c \geqslant 5$ in Bayes-UCB and $\epsilon > 0$, the expected regret of the Bayes-UCB algorithm $\mathcal{A}$ is upper bounded by,*

$$\mathbb{E}[R_T(\mathcal{A})] \leqslant \sum_{i \neq i^*} \left(\frac{1+\epsilon}{D(\mu_i||\mu^*)} \ln(T)\Delta_i \right) + o_{\epsilon,c}(\ln(T)) \sum_{i=1}^K \Delta_i.$$

Proof: The proof of Lemma 5 is provided in [16]. ■

Algorithm 5: Bayes-UCB

```
Input: Time Horizon $T$, Number of arms $K$,c
Output: Allocation policy $\mathcal{A} = \{I_1, I_2, \ldots, I_T\}$
Initialize: $\pi_i^0$ Uniform $\forall i \in \mathcal{K}$
for t= 1 : T do
    • Compute $q_j(t) = Q\left(1 - \frac{1}{t(\ln(T))^c}, \lambda_j^{t-1}\right) \;\; \forall i \in \mathcal{K}$
    • Select arm $I_t \in \arg\max_{j \in [K]} q_j^t$
    • Observe reward $X_{I_t,t}$
    • Update the prior $\pi_i^t$
```

4 Mechanism Design: A Quick Review

This section provides a quick review on mechanism design. The material of this section is taken from [20]. The following provides a general setting for formulating, analyzing, and solving mechanism design problems.

- There are K agents, $1, 2, \ldots, K$, with $\mathcal{K} = \{1, 2, \ldots, K\}$. The agents are rational and intelligent, and interact strategically among themselves towards making a collective decision.
- $\mathcal{X}$ is a set of *alternatives* or *outcomes*. The agents are required to make a collective choice from the set $\mathcal{X}$.
- Prior to making the collective choice, each agent privately observes his preferences over the alternatives in $\mathcal{X}$. This is modeled by supposing that agent i privately observes a parameter or signal θ_i that determine his preferences. The value of θ_i is known to agent i and may not be known to the other agents. θ_i is called a *private value* or *type* of agent i.
- We denote by Θ_i the set of private values of agent i, $i = 1, 2, \ldots, K$. The set of all type profiles is given by $\Theta = \Theta_1 \times \ldots \times \Theta_K$. A typical type profile is represented as $\theta = (\theta_1, \ldots, \theta_K)$.
- It is assumed that there is a common prior distribution $\mathbb{P} \in \Delta(\Theta)$. To maintain consistency of beliefs, individual belief functions $p_i : \Theta_i \to \Delta(\Theta_{-i})$ (where Θ_{-i} is the set of type profiles of all agents other than i) can all be derived from the common prior.
- Individual agents have preferences over outcomes that are represented by an utility function $u_i : \mathcal{X} \times \Theta_i \to \mathbb{R}$. Given $x \in \mathcal{X}$ and $\theta_i \in \Theta_i$, the value $u_i(x, \theta_i)$ denotes the payoff that agent i, having type $\theta_i \in \Theta_i$, receives from an outcome $x \in \mathcal{X}$. In a more general case, u_i depends not only on the outcome and the type of player i, but could depend on the types of the other players as well, and so $u_i : \mathcal{X} \times \Theta \to \mathbb{R}$. We restrict our attention to the former case in this paper.
- The set of outcomes $\mathcal{X}$, the set of players $\mathcal{K}$, the type sets Θ_i $(i = 1, \ldots, K)$, the common prior distribution $\mathbb{P} \in \Delta(\Theta)$, and the payoff functions u_i $(i = 1, \ldots, K)$ are assumed to be *common knowledge* among all the players. The specific type θ_i observed by agent i is private information of agent i.

Since the preferences of the agents depend on the realization of their types $\theta = (\theta_1, \ldots, \theta_K)$, it is logical and natural to make the collective decision depend on θ. This leads to the definition of a social choice function.

Definition 2 (Social Choice Function). *Suppose $\mathcal{K} = \{1, 2, \ldots, K\}$ is a set of agents with type sets $\Theta_1, \Theta_2, \ldots, \Theta_K$ respectively. Given a set of outcomes $\mathcal{X}$, a social choice function is a mapping $f : \Theta_1 \times \ldots \times \Theta_K \to \mathcal{X}$ that assigns to each possible type profile $(\theta_1, \theta_2, \ldots, \theta_n)$, an outcome from the set $\mathcal{X}$. The outcome corresponding to a type profile is called a social choice or collective choice for that type profile.*

One can view mechanism design as the process of solving an incompletely specified optimization problem where the specification is first elicited and then the underlying optimization problem or decision problem is solved. To elicit the type information from the agents in a truthful way, there are broadly two kinds of approaches, which are aptly called *direct mechanisms* and *indirect mechanisms*. We define these below.

Definition 3 (Direct Mechanism). *Suppose $f : \Theta_1 \times \ldots \times \Theta_K \to \mathcal{X}$ is a social choice function. A direct mechanism (also called a direct revelation mechanism) corresponding to f consists of the tuple $(\Theta_1, \Theta_2, \ldots, \Theta_K, f(.))$.*

The idea of a direct mechanism is to *directly* seek the type information from the agents by asking them to reveal their true types.

Definition 4 (Indirect Mechanism). *An indirect mechanism (also called an indirect revelation mechanism) consists of a tuple* $(S_1, S_2, \dots, S_K, f(.))$ *where* S_i *is a set of possible actions for agent* i *(*$i = 1, 2, \dots, K$*) and* $f : S_1 \times S_2 \times \dots \times S_K \rightarrow \mathcal{X}$ *is a function that maps each action profile to an outcome.*

The idea of an indirect mechanism is to provide a choice of actions to each agent and specify an outcome for each action profile. One example of an indirect mechanism is the auction mechanism, where each agent or player is asked his bid, and the social choice function provides an outcome consisting of an allocation and a payment function that depends on the bids. Thus, strategy of each player i, S_i is denoted by the bid he provides and the function f provides an allocation and a payment mechanism that depends on these bids. We now present an example of auction mechanism namely, Clarke mechanism to illustrate the idea behind auction mechanism. Clarke mechanisms assume a restricted environment called the quasilinear environment.

Definition 5 (Quasilinear Environment). *In the quasilinear environment, an outcome* $x \in \mathcal{X}$ *is a vector of the form* $x = (a, p_1, p_2, \dots, p_K)$*, where* $a \in \mathcal{A}$ *is an element of finite set* $\mathcal{A}$ *also called as set of allocations and the term* $p_i \in \mathbb{R}$ *represents the monetary transfer to agent* i*. Moreover, the utility function is quasilinear i.e.,*

$$u_i(x, \theta_i) = u_i((a, p_1, p_2, \dots, p_K), \theta_i) = v_i(a, \theta_i) + m_i + p_i.$$

Here, m_i *is agent* i's *initial endowment of the money and the function* $v_i(.,.)$ *is known as agent* i's *valuation function.*

We now define some desirable properties that we want a social choice function to satisfy:

Definition 6 (Dominant Strategy Incentive Compatibility (DSIC)). *A social choice function* $f : \Theta_1 \times \dots \times \Theta_K \rightarrow \mathcal{X}$ *is said to be dominant strategy incentive compatible (or truthfully implementable in dominant strategies) if the indirect revelation mechanism* $\mathcal{D} = ((S_i(\Theta_i))_{i \in \mathcal{K}}, f(\cdot))$ *has a* weakly dominant strategy equilibrium $s^*(\cdot) = (s_1^*(\cdot), \dots, s_n^*(\cdot))$ *in which* $s_i^*(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, \forall i \in \mathcal{K}$ *i.e.*

$$u_i(f(\theta_i, s_{-i}(\theta_{-i})), \theta_i) \geqslant u_i(f(s_i(\theta_i), s_{-i}(\theta_{-i})), \theta_i),$$

$\forall s_i(\theta_i) \in \Theta_i, \forall s_{-i}(\theta_{-i}) \in \Theta_{-i}$. *Here,* $s_{-i}(\theta_{-i})$ *denotes the strategy of all the agents other then agent* i *with type profile* θ_{-i}.

Definition 7 (Allocative Efficiency). *We say that a social choice function* $f(.) = (a(.), p_1(.), p_2(.), \dots, p_K(.))$ *is allocatively efficient if for each* $\theta \in \Theta$*,* $a(\theta)$ *satisfies the following condition:*

$$a(\theta) \in \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{K} v_i(a, \theta_i).$$

The above definition implies that for every $\theta \in \Theta$, the allocation $a(\theta)$ maximizes the sum of values of the players. The sum of valuation of all the players is also known as social welfare and an allocatively efficient social choice function maximizes the social welfare. We denote an allocatively efficient social choice function by $a^*(\theta)$.

Example 1 (Clarke Mechanism). Let us denote the strategic bid profile from agents as $b = (b_1, b_2, \dots, b_K)$. Let the SCF $f(.) = (a^*(.), p_1(.), p_2(.), \dots, p_K(.))$ be allocatively efficient. Then Clarke payment is given as:

$$p_i(b_i, b_{-i}) = \sum_{j \neq i} v_j(a^*(b), b_j) - \sum_{j \neq i} v_j(a^*_{-i}(b), b_j), \ \forall b_{-i}, \forall i.$$

Here $a^*_{-i}(b)$ is derived from the allocatively efficient social choice function in the absence of agent i i.e. $a_{-i}(b_{-i}) \in \arg\max_{a_{-i} \in \mathcal{A}_{-i}} \sum_{j \neq i} v_j(a_{-i}, b_i)$. It can be shown that this mechanism has desirable properties allocative efficiency and dominant strategy incentive compatibility. The payment structure is externality based as the second term in the payment computation of agent i computes the social welfare in the absence of agent i.

4.1 Characterizing Truthful Mechanisms

We now present Myerson's [19] theorem for characterizing DSIC social choice functions in quasilinear environment. We first define an important property called monotonicity of an allocation that yields truthfulness as follows:

Definition 8 (Monotone Allocation Rule). *Consider any two bids for agent* i, b_i *and* b_i^- *such that* $b_i \geqslant b_i^-$. *An allocation rule* $a \in \mathcal{A}$ *is called monotone if for every agent* i, *we have* $a_i(b_i, b_{-i}) \geqslant a_i(b_i^-, b_{-i})\ \forall b_{-i} \in \Theta_{-i}\ \forall \theta_i \in \Theta_i$.

Theorem 2. *[19] A mechanism* $\mathcal{M} = (a, p)$ *is truthful if and only if:*

- *Allocation rule* a *is monotone*
- *The payment rule for any player* i *satisfies*

$$p_i(b_i, b_{-i}) = p_i^0(b_{-i}) + b_i a_i(b_i, b_{-i}) - \int_{-\infty}^{b_i} a_i(u, b_{-i})du,$$

where $p_i^0(b_{-i})$ *does not depend on* b_i.

So far we have seen few examples of MAB algorithms and mechanism design in separate terms. In the next section, we explain how one can design MAB mechanisms for the settings where each agent is also associated with certain stochastic reward apart from holding a private information. In order to maximize the social welfare, mechanism designer also need to learn the stochastic reward while eliciting the private information from the agents. Thus, in the next section we describe ways to meld techniques from MAB algorithm and mechanism design.

5 Stochastic MAB Mechanisms

5.1 MAB Mechanism Design Environment

Stochastic MAB mechanisms capture the interplay between the online learning and strategic bidding. Similar to the setting we described for mechanism design, the setting for stochastic MAB mechanisms can be described as follows:

- There are K agents, $1, 2, \ldots, K$, with $\mathcal{K} = \{1, 2, \ldots, K\}$. The agents are rational and intelligent.
- Each agent i privately observes his valuation θ_i that determines his preferences. The value of θ_i is known to agent i and is not known to the other agents.
- The set of private values of agent i is denoted by Θ_i. The set of all type profiles is given by $\Theta = \Theta_1 \times \ldots \times \Theta_K$. A typical type profile is represented as $\theta = (\theta_1, \ldots, \theta_K)$.
- Each agent i is also parametrized by the parameter $\rho_i \in \Upsilon$. Each agent i has a certain stochastic reward associated with him that comes from distribution ν_{ρ_i} and has expectation $\mu_i \in \mathbb{R}$.
- The parameters, ρ_i and μ_i are unknown to the the agents and to the mechanism designer, and hence need to be learnt over time.
- In order to learn the reward expectations μ_i, mechanism design problem is repeated over time. These time instances are denoted by $t \in \{1, 2, \ldots, T\}$.
- At any time t, the mechanism consists of a tuple $(a^t \in \mathcal{A}, p^t \in \mathcal{P})$ and the mechanism design problem involves finding the allocation rule $a^t \in \mathcal{A}$ and the payment rule $p^t \in \mathcal{P}$ by eliciting the private valuations θ_i from the agents and by observing the rewards obtained so far. Let us denote set of all possible mechanisms by $\mathcal{X}$.
- At any time t, allocation rule $a^t = \{a_1^t, a_2^t, \ldots, a_K^t\}$ denotes whether an agent i is allocated at time t or not i.e. $a_i^t \in \{0, 1\}\ \forall i \in \mathcal{K}, \forall t \in \{1, 2, \ldots, T\}$. If agent i is allocated at time t then $a_i^t = 1$ and is 0 otherwise. Note that an agent corresponds to an arm of multiarmed bandit and allocation rule a provides an arm pulling strategy where $a_i^t = 1$ implies pulling an arm i at time t.
- Given an allocation rule a^t, $X_i(t) \sim \nu_{\rho_i} \in \mathbb{R}$ denotes the stochastic reward obtained from agent i at time t. Note that rewards are observed only for those agents who are allocated i.e. whose $a_i^t = 1$. We assume that $X_i(t) = 0$ if agent i is not allocated at time t i.e. $a_i^t = 0$.
- Utility function at instance t is given by $u_i^t : \mathcal{X} \times \Theta_i \times \mathbb{R} \to \mathbb{R}$. Given $x^t \in \mathcal{X}$ and $\theta_i^t \in \Theta_i$, $X_i(t) \sim \nu_{\rho_i} \in \mathbb{R}$, the value $u_i^t(x^t, \theta_i^t, X_i(t))$ denotes the payoff that agent i, having type $\theta_i \in \Theta_i$, reward $X_i(t)$, receives from an outcome $x^t \in \mathcal{X}$.

- The agents might be strategic and may not report their true valuations θ_i^t to the mechanism designer so as to increase their utilities.
- Total expected utility of player i is denoted by $u_i = \mathbb{E}_{X_i(t)\sim\nu_{\rho_i}}[\sum_{t=1}^{T} u_i^t(x^t, \theta_i^t, X_i(t))]$
- Note that, in some settings private valuations θ_i^t can change over time. However, in this paper we will focus on the settings where private valuations do not change over time i.e. $\theta_i^t = \theta_i$, $\forall i \in \mathcal{K}\ \forall t \in \{1, 2, \ldots, T\}$ and agents are asked to report their private values only once.

5.2 An Example: Single Slot Sponsored Search Auction

To understand the MAB mechanism setting more clearly, we provide an example of single slot sponsored search auction (SSA) which is also known as pay-per-click auction. Single slot SSA is the simplest example of MAB mechanism and the setting is described as follows:

- There are K advertisers, $1, 2, \ldots, K$, with $\mathcal{K} = \{1, 2, \ldots, K\}$ competing for a single slot available to the search engine.
- If the advertisement i is displayed on the slot, it gets a click with probability μ_i. The click probability μ_i is also known as click through rate and is unknown to the advertisers and to the search engine. The click through rate therefore corresponds to the stochastic reward of the advertiser with mean μ_i and ν_{ρ_i} has Bernoulli distribution with parameter μ_i.
- Each advertiser i values θ_i for the click which is only known to him and is unknown to the other advertisers and search engine.
- Each advertiser i is asked to bid his valuation and the bid given by the advertiser i is denoted by b_i.
- We denote by Θ_i the set of private values of advertisers i, $i = 1, 2, \ldots, K$.
- The parameters μ_i are unknown to the the advertisers and the search engine and hence need to be learnt over the time.
- In order to learn the click probabilities μ_i, mechanism design problem is repeated over time. At each time $t \in \{1, 2, \ldots, T\}$ the click is observed for the allocated advertisement and this observation is used to make future allocation decisions.
- The goal of the search engine is to design a truthful and individual rational mechanism $\mathcal{M} = (a^t(b), p^t(b))_{t\in\{1,2,\ldots,n\}}$ based on the bid profile b given by the advertisers. Here, $a^t(b) = (a_1^t, a_2^t, \ldots, a_K^t)$ denotes the allocation rule $a_i^t = 1$ if the slot is allocated to advertiser i at time t and is 0 otherwise. The payment given to the advertisers at time t is denoted by $p^t = (p_1^t, p_2^t, \ldots, p_K^t)$
- Allocation a^t also depends on observed rewards till time t. If an agent is allocated at time t i.e. if $a_i^t = 1$ then $X_i(t) = 1$ represents getting a click at time t and $X_i(t) = 0$ represents not getting a click at time t.
- Valuation of an agent i at time t given the allocation t is denoted by $\theta_i X_i(t) a_i^t$.
- Utility function of an advertiser i at instance t is given by $u_i^t = \theta_i a_i^t(b) - p_i^t(b)$ if advertiser i receives a click and is 0 otherwise. Thus, expected utility of the advertiser i at time t is given by $u_i^t = \mu_i(\theta_i a_i^t(b) - p_i^t(b))$.
- The advertisers might be strategic and may not report their true valuations $\theta_i's$ to the search engine so as to increase their utilities.

5.3 MAB Mechanisms: Key Notions

When learning is not involved, allocation and payment depend only on the bids provided by the agents. However, when there is learning involved, allocation and payment functions at any round t also depend on how the rewards or success are observed in the previous allocations. We now define the notion of reward realization in the context of sponsored search auction.

Definition 9 (Reward Realization). *A reward realization denotes an instance of the reward. Reward realization, in the case of SSA, is a matrix $s \in \{0, 1\}^{K\times T}$ where each parameter s_i^t is an independent Bernoulli random variable with parameter μ_i. Thus, for any time t,*

$$s_i^t = \begin{cases} 1 \textit{ with probability } \mu_i, \\ 0 \textit{ with probability } 1 - \mu_i. \end{cases}$$

Note that depending on the algorithm, only a part of reward realization can be observed. If the algorithm allocates the slot to advertiser i at time t then the reward s_i^t is observed. We have defined reward realization in the context of SSA where a reward is binary parameter which is 1 if click is obtained and 0 if the click is not obtained. The reward realization can be similarly defined in other contexts, for example in the crowdsourcing setting, reward can correspond to allocated worker submitting the task on time or giving the correct answer of the task. In general, if $X_i(t)$ denotes the sample obtained from distribution ρ_i with expectation μ_i at time t then the reward realization matrix s is given as $s_i^t = X_i(t)$. We now define some of the MAB mechanisms properties that are desirable.

Definition 10 (Ex-Post Monotone Allocation Rule). *We say that an allocation rule "a" is ex-post monotone if allocation rule "a" is monotone for every reward realization i.e.,* $\forall i \in \mathcal{K},\ \forall b_i \geqslant b_i^-$

$$\begin{aligned} &a_i(b_i, b_{-i}; s) \geqslant a_i(b_i^-, b_{-i}; s), \\ &\forall b_{-i} \in \Theta_{-i}, s \in \{0,1\}^{K \times T} . \end{aligned}$$

Here, $a_i(b_i, b_{-i}; s) = \sum_{t=1}^{T} a_i^t(b_i, b_{-i}; s)$.

Note that the allocation at any time t can only depend on the reward realization observed till time $t-1$. But to avoid notation clutter we denote this dependency by complete reward realization s.

Definition 11 (Stochastic Monotone Allocation Rule). *We say that an allocation rule "a" is stochastic monotone if it is monotone in expectation with respect to reward realizations,*

$$\begin{aligned} &\mathbb{E}_s[a_i(b_i, b_{-i}; s)] \geqslant \mathbb{E}_s[a_i(b_i^-, b_{-i}; s)], \\ &\forall b_i \geqslant b_i^-,\ \forall b_{-i} \in \Theta_{-i} . \end{aligned}$$

Definition 12 (Ex-Post Incentive Compatible Mechanism). *We say that a mechanism is ex-post incentive compatible if all the bidders are truthful for every reward realization irrespective of the bids of other workers,*

$$\begin{aligned} &u_i(a_i(\theta_i, b_{-i}; s), p_i(\theta_i, b_{-i}; s), \theta_i; s) \geqslant u_i(a_i(b_i, b_{-i}; s), p_i(b_i, b_{-i}; s), \theta_i; s), \\ &\forall \theta_i \in \Theta_i, b_i \in \Theta_i, b_{-i} \in \Theta_{-i}, s \in \{0,1\}^{K \times T} . \end{aligned}$$

For single slot SSA this implies:

$$\begin{aligned} &\theta_i a_i(\theta_i, b_{-i}; s) + p_i(\theta_i, b_{-i}; s) \geqslant \theta_i a_i(b_i, b_{-i}; s) + p_i(b_i, b_{-i}; s), \\ &\forall b_i \in \Theta_i, b_{-i} \in \Theta_{-i}, s \in \{0,1,\}^{K \times T} . \end{aligned}$$

Definition 13 (Stochastic Incentive Compatible Mechanism). *We say that a mechanism is stochastic incentive compatible if all the bidders are truthful in expectation with respect to reward realization irrespective of the bids of other workers,*

$$\begin{aligned} &\mathbb{E}_s[u_i(a_i(\theta_i, b_{-i}; s), p_i(\theta_i, b_{-i}; s), \theta_i; s)] \geqslant \mathbb{E}_s[u_i(a_i(b_i, b_{-i}; s), p_i(b_i, b_{-i}; s), \theta_i; s)], \\ &\forall \theta_i \in \Theta_i, b_i \in \Theta_i, b_{-i} \in \Theta_{-i}, s \in \{0,1\}^{K \times T} . \end{aligned}$$

Definition 14 (Ex-Post Individual Rational Mechanism). *A mechanism* $M = (a, p)$ *is said to be ex-post individually rational if participating in the mechanism always gives any advertiser non-negative utility. That is,* $\forall i \in \mathcal{K}$,

$$u_i(a_i(\theta_i, b_{-i}; s), p_i(\theta_i, b_{-i}; s), \theta_i; s) \geqslant 0,\ \forall \theta_i \in \Theta_i, b_{-i} \in \Theta_{-i}, s \in \{0,1\}^{K \times T}.$$

For single slot SSA this implies $\forall i \in \mathcal{K}, t \in \{1, 2, \ldots, T\}$,

$$\theta_i a_i(\theta_i, b_{-i}; s) + p_i(\theta_i, b_{-i}; s) \geqslant 0,\ \forall \theta_i \in \Theta_i, b_{-i} \in \Theta_{-i}, s \in \{0,1\}^{K \times T}.$$

Definition 15 (Social Welfare Regret of MAB Mechanisms). *Social welfare regret of any MAB mechanism* $\mathcal{M} = (a, p)$ *is defined as the difference between social welfare obtained by the MAB mechanism when stochastic parameters of the reward realization are not known and social welfare obtained by the mechanism when they are known. Formally,*

$$R_T(\mathcal{M}) = \sum_{t=1}^{T} \mathbb{E}_{X_i(t) \sim \nu(\rho_i)} \Big[\sum_{i=1}^{K} \nu_i(a^*, \theta_i, X_i(t)) - \sum_{i=1}^{K} \nu_i(a_i^t, \theta_i, X_i(t)) \Big].$$

Here, a^* *denotes the allocative efficient allocation when the reward parameters* ρ_i *and* μ_i *for all agents* i *are known. In the case of SSA, the social welfare regret of a mechanism* $\mathcal{M} = (a, p)$ *is given as:*

$$R_T(\mathcal{M}) = \sum_{t=1}^{T} \Big[\mu_{i^*} \theta_{i^*} - \sum_{i=1}^{K} a_i^t \mu_i \theta_i \Big].$$

Here, $i^* = \arg\max_i \mu_i \theta_i$.

One can similarly define the revenue regret of MAB mechanism where the regret is defined in terms of payments obtained from MAB mechanism. In this paper, we are generally interested in social welfare regret. And in the rest of the paper, when we talk about regret it would mean social welfare regret. We start with some characterization results available for the stochastic MAB mechanisms and then we present some mechanisms available in the literature.

5.4 A Lower Bound on Regret of MAB Mechanisms

This section presents a lower bound result on the regret that was proposed in [5] where the authors characterize any deterministic, dominant strategy incentive compatible mechanism. This characterization holds only in the setting where the agents are aware of the complete reward realization and can strategise based on future outcome. Therefore the goal of the mechanism designer is to design a deterministic, incentive compatible mechanism for any reward realization which can compute the allocation and payments based on the rewards observed so far. The authors further make assumptions of normalized mechanism and non-degenerate, scale-free allocation rules. Before presenting the characterization result, we first provide some definitions:

Definition 16 (Normalized Mechanism). *A mechanism is said to be normalized if for any agent $i \in \mathcal{K}$, for bids of other agents $b_{-i} \in \Theta_{-i}$ and for any reward realization $s \in \{0,1\}^{K \times T}$, we have:*

$$p_i(b_i, b_{-i}; s) \to 0 \textit{ as } b_i \to 0$$

A normalized mechanism fetches zero payment for an agent with zero bid.

Definition 17 (Non-degenerate Allocation Rule). *Consider a fixed bid profile b, reward realization s, rounds t, t' such that $t \leqslant t'$. Let agent i be given the allocation at time t with bid profile b and reward realization s i.e. $a_i^t(b; s) = 1$. Consider another reward realization s' such that $s_i'^t = 0$ if $s_i^t = 1$ and is 1 otherwise. An allocation rule a is called non-degenerate with respect to (b, s, t, t') if there exists an interval I of positive length containing b_i such that*

$$a_i^j(x, b_{-i}; \varphi) = a_i^j(x, b_{-i}; \varphi),\ \forall \varphi \in \{s, s'\},\ \forall j \in \{t, t'\}, \forall x \in I.$$

An allocation rule is called non-degenerate if it is non-degenerate with respect to each tuple (b, s, t, t').

Definition 18 (Scalefree Allocation Rule). *An allocation rule a is scale-free if it is invariant under multiplication of all the bids by the same positive number i.e.*

$$a_i^t(b; s) = a_i^t(cb; s),\ \forall c > 0, \forall s, \forall b$$

Definition 19 (Influential Round). *Round t is called (b, s) influential if for some round $t' > t$, if we have $a_i^{t'}(b; s) \neq a_i^{t'}(b; s')$ where $s_i'^t = 1$ if $s_i^t = 0$ and $s_i'^t = 0$ if $s_i^t = 1$ and for all $k \neq i$, $j \neq t$, $s_k^j = s_k'^j$. Round t is called influential with respect to reward realization s if there exists a bid profile b such that it is (b, s) influential.*

Definition 20 (Exploration Separated Allocation Rule). *An allocation rule a is called exploration separated if for every reward realization s and round t that is influential for s, we have $a^t(b; s) = a^t(b'; s)$ for any two bid profiles b and b'.*

Exploration separated allocation rules ensure that the allocation in any influential round does not depend on the bid of the agent. We now provide the main characterization result:

Theorem 3. *Let a be a non-degenerate, scale-free deterministic allocation rule. If $M = (a, p)$ is a normalized truthful mechanism for some payment rule p, then it is exploration separated.*

Thus, from the above theorem, for any normalized truthful mechanism with non-degenerate and scale-free allocation rule, an influential round does not depend on the bids of the agents. Also, note that the influential rounds are crucial for learning as the rewards obtained in these rounds decide future allocations. Thus, we get the following lower bound on the regret:

Theorem 4. *If an allocation rule a is exploration separated and deterministic, then the regret of an algorithm $\mathcal{A}$ that induces allocation rule a is given as $R_T(\mathcal{A}) = \Omega(T^{2/3})$*

The proof of the above Theorems can be found in [3].

5.5 Exploration Separated Mechanisms

Given the characterization results provided in the previous section, we know that any deterministic truthful MAB mechanism has to be exploration separated. In this section, we present the exploration separated mechanism where the agents are allocated in round robin fashion for ϵT number of rounds. These rounds are also known as exploration rounds as they do not depend on bids and no payment is made by agents in these rounds. Let $\hat{\mu}_i$ denote the empirical estimate of the click probabilities that is obtained in exploration rounds. From Hoeffding's bound with probability greater than $1 - T^{-4}$, we have,

$$\mu_i \leqslant \hat{\mu}_i + \sqrt{2\lfloor \frac{K}{\epsilon T} \rfloor \log T} = \hat{\mu}_i^+$$

The mechanism then allocates $(1-\epsilon)T$ rounds to the agent having highest value of $b_i \hat{\mu}_i^+$ where b_i represents the bid of the agent i. The payments are derived from the VCG payment rule. The mechanism is given in Algorithm 6.

```
Algorithm 6: Exploration Separated Mechanism
  Input: Bids from advertisers $(b_i) \in \Theta_i \ \forall i \in \mathcal{K}$, number of rounds $T$, parameter $\epsilon$
  Output: Mechanism $M = (a^t, p^t)_{t \in \{1,2,\ldots T\}}$
  Initialize: $t = 1, S_i = 0, N_i = 0, a_i^t = p_i^t = 0, \ \forall i \in \mathcal{K}, \forall t \in \{1, 2, \ldots, T\}$
   • while $t < \lfloor \frac{\epsilon T}{K} \rfloor K$ do
        - for $i = 1 : K$ do
             * $a_i^t = 1, p_i^t = 0$
             * $t = t + 1$
             * $N_i = N_i + 1$
             * If click is observed, $S_i = S_i + 1$
   • Let $\hat{\mu}_i = \frac{S_i}{N_i}$.
   • for $t = \lfloor \frac{\epsilon T}{K} \rfloor K + 1, \cdots, T$ do
        - $a_i^t = 1$ if $i = \arg\max_i b_i \hat{\mu}_i^+ = \arg\max_i b_i(\hat{\mu}_i + \sqrt{\frac{2 \log T}{N_i}})$
        - Payment from the advertiser $i$ is given by $p_i^t = \max_{j \neq i} \frac{\hat{\mu}_j^+ b_j}{\hat{\mu}_i^+}$
```

Since the algorithm is similar to exploration separated algorithm described in Section 3.1.1, the regret achieved by the Algorithm 6 is $O(T^{2/3})$. Also, the payments are scaled VCG payments (scaled by parameter $\hat{\mu_i}^+$). This scaling is necessary to achieve individual rationality

Lemma 6. *The mechanism in Algorithm 6 is dominant strategy incentive compatible and individually rational.*

Proof: The utility of any agent i at round $t \leqslant \lfloor \frac{\epsilon T}{K} \rfloor K$ if he receives a click is given as:

$$u_i^t = \theta_i a_i^t, \ \forall b_i \in \Theta_i. \quad \text{(as the payments in these rounds are 0 and allocation does not depend on bids.)}$$

If $i = \arg\max_i b_i \hat{\mu}_i^+$, then utility of an agent i at round $t > \lfloor \frac{\epsilon T}{K} \rfloor$ if he receives a click is given as:

$$\begin{aligned} u_i^t &= b_i - \max_{j \neq i} \frac{\hat{\mu}_j^+ b_j}{\hat{\mu}_i^+} \qquad \text{(if click is observed)} \\ &= \frac{\hat{\mu}_i^+ b_i - \max_{j \neq i} \hat{\mu}_j^+ b_j}{\hat{\mu}_i^+}. \end{aligned}$$

By the choice of i, we have $u_i^t \geqslant 0 \ \forall t \in \{1, 2, \ldots, T\}$. Thus, mechanism is individually rational. From Example 1, VCG payment is given as:

$$\begin{aligned} p_i^t(b_i, b_{-i}) &= \sum_{j \neq i} v_j(a^*(b), b_j) - \sum_{j \neq i} v_j(a^*_{-i}(b), b_j \ \forall b_{-i}), \forall i \in \mathcal{K} \\ &= 0 - \max_{j \neq i} \hat{\mu}_j^+ b_j. \end{aligned}$$

Thus, payment given by Algorithm 6 are scaled VCG payments where scaling factor does not depend on the bid of the player. Hence, the mechanism is dominant strategy incentive compatible.

■

5.6 Generic Transformation for Truthful Mechanisms

Myerson's characterization result (Theorem 2) provides the conditions to be satisfied by the allocation rule and the payment rule so as to result in a truthful mechanism. The only condition to be satisfied by the allocation rule is monotonicity. However, looking at the payment scheme, it seems hard to find the payment rule given the monotone allocation rule. In the MAB mechanism problem, computing the integration involved in payment rule is much harder. Allocation with different bid profiles cannot be computed as allocation depends on the successes observed so far and successes for other bid profiles can not be observed. This section provides a generic transformation that produces a truthful and individually rational mechanism given a monotone allocation rule by invoking the allocation rule only once [4]. Using this transformation, the problem of designing truthful mechanism reduces to that of finding a monotone allocation rule. The generic procedure creates a randomized mechanism that is truthful in expectation and which attains the same outcome as the original allocation rule with high probability.

Thus, for parameter $\mu \in [0,1]$, the transformation takes any monotone allocation rule a and gives a truthful mechanism $\tilde{M} = (\tilde{a}, \tilde{p})$, such that following properties hold:

- It converts the bid vector b to a randomized bid vector $\tilde{b}$ and invokes $a(\tilde{b})$ only once.
- For any bid vector b and any fixed parameter μ, $\tilde{a}(b) = a(\tilde{b})$ and $a(b)$ are identical with probability at least $1 - K\mu$.
- $\tilde{M}$ is ex-post individually rational and never pays any agent i more than $a_i(\tilde{b})\left(b_i\left(\frac{1}{\mu} - 1\right) - \frac{\underline{b}}{\mu}\right)$. Here, $\underline{b}$ is the lower bound on the bid i.e. $\underline{b} \leqslant b_i,\ \forall b_i \in \theta_i,\ \forall i \in \mathcal{K}$.

Note that to get the truthfulness it is sufficient to design the following payment rule for any reward realization s if the allocation rule is monotone:

$$p_i(b_i, b_{-i}; s) = b_i a_i(b_i, b_{-i}; s) - \int_{-\infty}^{b_i} a_i(u, b_{-i}; s)du\ . \tag{23}$$

The challenge here is to compute the integral as the allocation depends on how the successes are observed. To compute this integral, a sampling procedure is used. The following lemma is then used to compute the integral using the sampling procedure.

Lemma 7. *Let* $\mathcal{F} : I \to [0,1]$ *be any strictly increasing function that is differentiable and satisfies* $\inf_{z\in I}\mathcal{F}(z) = 0$ *and* $\sup_{z\in I}\mathcal{F}(z) = 1$. *If* Y *is a random variable with cumulative distribution function* $\mathcal{F}$, *then*

$$\int_I g(z)dz = \mathbb{E}\left[\frac{g(Y)}{\mathcal{F}'(Y)}\right]\ . \tag{24}$$

The sampling procedure takes the bids and produces two random vectors α and β. The vector α is used for determining the allocation rule and the payments are derived using the vector β. For deriving truthful mechanisms, we need sampling procedure to satisfy certain properties which we describe below:

Definition 21 (Self-resampling Procedure). *A self-resampling procedure with support* $I = [\underline{b}, \overline{b}]$ *and resampling probability* $\mu \in (0,1)$ *is a randomized algorithm that outputs random vectors* $\alpha \in I$ *and* $\beta \in I$ *given the input bid vector* $b \in I$ *and satisfies the following properties,* $\forall i \in \mathcal{K}$:

1. $\alpha_i(b_i)$ *and* $\beta_i(b_i)$ *are non-decreasing functions of* b_i.
2. (A) *With probability* $(1-\mu)$, $\alpha_i(b_i) = \beta_i(b_i) = b_i$.
   (B) *With probability* μ, $\underline{b} \leqslant \alpha_i(b_i) \leqslant \beta_i(b_i) < b_i$.
3. $\mathbb{P}[\alpha_i(b_i) < a_i | \beta_i(b_i) = b_i'] = \mathbb{P}[\alpha_i(b_i') < a_i]\ \ \forall a_i \leqslant b_i' < b_i$.
4. *The function* $\mathcal{F}(a_i, b_i) = \mathbb{P}[\beta_i(b_i) < a_i | \beta_i(b_i) < b_i]$ *is called the distribution function of the self resampling procedure. For each* b_i, *the function* $F(., b_i)$ *is differentiable and strictly increasing on the interval* $I \cap (-\infty, b_i)$.

We now present one example of such self-resampling procedure in Algorithm 7.

Lemma 8. *The procedure in Algorithm 7 is a self-resampling procedure with distribution* $F(a_i, b_i) = \frac{a_i - \underline{b}}{b_i - \underline{b}}$.

Algorithm 7: Self-resampling Procedure

```
Input: bid $b_i \in [\underline{b}, \overline{b}]$, parameter $\mu \in (0, 1)$
Output: $(\alpha_i, \beta_i)$ such that $\underline{b} \leqslant \alpha_i \leqslant \beta_i \leqslant b_i$
  • with probability $(1 - \mu)$
      - $\alpha_i \leftarrow b_i,\ \beta_i \leftarrow b_i$
  • with probability $\mu$
      - Pick $b_i' \in [\underline{b}, b_i]$ uniformly at random.
      - $\alpha_i \leftarrow \mathrm{recursive}(b_i'),\ \beta_i \leftarrow b_i'$

function Recursive($b_i$)

  • with probability $(1 - \mu)$
      - return $b_i$
  • with probability $\mu$
      - Pick $b_i' \in [\underline{b}, b_i]$ uniformly at random.
      - return Recursive($b_i'$)
```

Proof: Properties 1 and 2 of self-resampling procedure given in Definition 21 are immediate from the algorithm. If $\beta_i(b_i) = b_i' < b_i$, it means that the algorithm has picked α_i from the subroutine recursive with parameter b_i' and thus property 3 follows. Property 4 follows from the fact that distribution of $\beta_i(b_i)$ is uniform in the interval $[\underline{b}, b_i]$ conditional on the event $\beta_i(b_i) < b_i$. ■

The algorithm that outputs the transformed allocation and the payment is described in Algorithm 8.

Algorithm 8: Transformation Mechanism

```
Input: $\forall i$, bids $b_i \in [\underline{b}, \overline{b}]$, parameter $\mu \in (0, 1)$, allocation rule $a$
Output: Allocation rule $\tilde{a}$ and the payment rule $\tilde{p}$
  • Obtain modified bids as $(\alpha, \beta) = ((\alpha_1(b_1), \beta_1(b_1)), (\alpha_2(b_2), \beta_2(b_2)), \ldots, (\alpha_n(b_n), \beta_n(b_n))$
  • Allocate according to $\tilde{a}(b) = a(\alpha(b))$
  • Ask payment from each advertiser $i$, $\tilde{p}_i(b) = b_i \tilde{a}_i(b) - R_i$, where,

      $R_i = \begin{cases} \frac{1}{\mu} \frac{a_i(\alpha(b))}{F_i'(\beta_i(b_i), b_i)}, & \text{if} \beta_i(b_i) < b_i \\ 0, & \text{otherwise.} \end{cases}$
```

Theorem 5. *Let a be any monotone allocation rule. Then the transformed mechanism $\tilde{M} = (\tilde{a}, \tilde{p})$ given in Algorithm 8 satisfy the following properties:*

1. *$\tilde{M} = (\tilde{a}, \tilde{p})$ is truthful, ex-post individually rational.*
2. *For K advertisers and any bid vector b allocations $\tilde{a}(b)$ and $a(b)$ are identical with probability at least $1 - K\mu$.*
3. *If all the private types are positive then the mechanism $\tilde{m}$ never pays any advertiser i more than $(b_i - \underline{b}) a_i(\alpha)(\frac{1}{\mu}) - b_i a_i(\alpha)$.*

Proof: Transformed allocation rule and payment rule produced by Algorithm 8 are denoted by $\tilde{a}$ and $\tilde{p}$ respectively. For all reward realizations s, we will prove two properties: (1) Allocation rule $\tilde{a}$ is monotone in terms of bids, and (2) the expected payment rule $\tilde{p}$ satisfies Equation (23).

The monotonicity of the allocation rule $\tilde{a}$ follows from the monotonicity of a and the monotonicity property 1 of Algorithm 7.

We now prove that $\mathbb{E}_{\alpha,\beta}[R_i] = \int_{-\infty}^{b_i} \tilde{a}_i(z, b_{-i}; s) dz$, where expectation is taken over the randomization of the Algorithm 8 (due to parameter μ).

$$\begin{aligned} &\mathbb{E}_{\alpha,\beta}[R_i] \\ &= \mathbb{E}_{\beta_i} \mathbb{E}_{\alpha|\beta_i}[R_i] && (R_i \text{ does not depend on } \beta_j,\ j \neq i) \end{aligned}$$

$$\begin{aligned}
&= \mathbb{P}\{\beta_i < b_i\}\mathbb{E}_{\beta_i|\beta_i<b_i}\mathbb{E}_{\alpha|\beta_i}[R_i] && (R_i = 0 \text{ if } \beta_i = b_i)\\
&= \mu\mathbb{E}_{\beta_i|\beta_i<b_i}\mathbb{E}_{\alpha|\beta_i}\left[\frac{a_i(\alpha(b);s)}{\mu F_i'(\beta_i(b_i), b_i)}\right] && \text{(Property 2 of Algorithm 7)}\\
&= \mathbb{E}_{\beta_i|\beta_i<b_i}\frac{1}{F_i'(\beta_i, b_i)}\mathbb{E}_{\alpha}[a_i(\alpha_i(\beta_i), \alpha_{-i}(b_{-i}); s)] && \text{(Property 3 of Algorithm 7)}\\
&= \mathbb{E}_{\beta_i|\beta_i<b_i}\frac{\tilde{a}_i(\beta; s)}{\mathcal{F}_i'(\beta_i, b_i)}\\
&= \int_{\underline{b}}^{b_i} \tilde{a}_i(z, b_{-i}; s)dz && \text{(Lemma 7)}\\
&= \int_{-\infty}^{b_i} \tilde{a}_i(z, b_{-i}; s)dz && \text{Allocation is zero for all bids } x \leqslant \underline{b}
\end{aligned}$$

Mechanism $\tilde{M}$ is individually rational because agent i is never charged more than $b_i\tilde{a}_i(b; s)$.

The probability that $\alpha_i(b_i) = b_i$ for all i is at least $1 - K\mu$. Thus the allocation rule $\tilde{a}(b) = a(b)$ with probability at least $1 - K\mu$. Note that term R_i is either 0 or equals to $\frac{1}{\mu}\frac{\mathcal{A}_i(\alpha(b))}{F_i'(\beta_i(b_i), b_i)}$. By Lemma 8, $F_i'(\beta_i(b_i), b_i) = \frac{1}{b_i - \underline{b}}$. Thus, R_i is either 0 or equals to $\frac{1}{\mu}(b_i - \underline{b})$, thus proving the last part of the theorem. ■

Note that the generic procedure provided above works when the agents have non-negative private types and the payments are asked from the agents. Such mechanisms are also known as forward auction. This generic transformation can also be extended to the case of reverse auction (or procurement auction) when the payments are given to the agents having non-positive private types. This extension is provided in more detail in [6].

5.7 UCB based Mechanisms

Given the generic procedure in the previous section, we now present a UCB based MAB mechanism that achieve the optimal regret as is given by MAB algorithms. The idea of the mechanism is to design a monotone allocation rule and then use the transformation rule presented in previous section to derive a truthful and individual rational mechanism. In this section, we will first prove that the allocation rule derived by the UCB algorithm is stochastic monotone. We will explain the UCB based mechanism in the context of sponsored search auction. However, the algorithm can be extended to different settings by calculating the UCB index appropriately.

Allocation rule based on UCB algorithm for sponsored search auction is given in Algorithm 9.

Algorithm 9: Allocation rule given by UCB Algorithm for SSA

```
Input: Bid vector $b$ from the advertisers.
Output: Allocation rule $a = (a_i^t)_{i \in \mathcal{K}, t \in \{1,2,\dots,T\}}$
• $a_i^t = 0, \forall i \in \mathcal{K}, \forall t \in \{1, 2, \dots, T\}$
• for $t = 1, \cdots, K$ do
    - Allocate the slot to advertiser $t$ and set $N_t(K) = 1$, $a_t^t = 1$.
    - Observe the click, $S_t(K) = 1$, if click is observed, $0$ otherwise.
• for $t = K + 1, \cdots, T$ do
    - $\forall i \in \mathcal{K}$, $e_{i,t} = \frac{S_i(t-1)}{N_i(t-1)}$, $c_{i,t} = \sqrt{\frac{2 \log t}{N_i(t-1)}}$
    - Select the advertiser $I_t = \arg\max_i b_i(e_{i,t} + c_{i,t})$
    - $N_{I_t}(t) = N_{I_t}(t-1) + 1$
    - $S_{I_t}(t) = S_{I_t}(t-1) + X_{I_t}(t)$
    - $a_{I_t}^t = 1$
    - for all other advertisers $j \neq I_t$ do
        * $N_j(t) = N_j(t-1)$
        * $S_j(t) = S_j(t-1)$
```

Lemma 9. *Allocation rule given by Algorithm 9 is stochastic monotone.*

Proof: To prove this, we first define a stack realization which is $K \times T$ table where $(i,t) = 1$ if advertiser i gets the click when he is allocated for the t^{th} time and is 0 if he does not receive the click when displayed for the t^{th} time. Note that stack realization is different from reward realization in a sense that stack realization does not capture the exact round when a particular advertiser gets a click and corresponding to one stack realization there can be multiple reward realizations possible. If the allocation rule a is completely determined by the stack realization and is monotone with respect to every stack realization, then it is also monotone with respect to expectation over reward realization. Now to prove the Lemma, we fix a stack realization σ, advertiser i, and bids of other advertisers b_{-i}. Now consider two bids by advertiser i, b_i and b_i^+ such that $b_i \leqslant b_i^+$. We will prove that the allocation rule given by Algorithm 9 allocates the slot to advertiser i more number of times with bid b_i^+ compared to the bid b_i. For notation convenience, let us denote allocation to advertiser i till time t with bid profile (b_i, b_{-i}) for a stack realization σ by $a_i^t(b_i)$ and with bid profile (b_i^+, b_{-i}) by $a_i^t(b_i^+)$. We will prove by induction that $a_i^t(b_i^+) \geqslant a_i^t(b_i)$.

When $t = 1$, any advertiser is allocated the slot irrespective of their bids, thus $a_i^1(b_i^+) \geqslant a_i^1(b_i)$. Thus, by induction hypothesis assume that at time t, $a_i^t(b_i^+) \geqslant a_i^t(b_i)$. Thus, we have to show that at time $t+1$, $a_i^{t+1}(b_i^+) \geqslant a_i^{t+1}(b_i)$. Without loss of generality, we can assume that $a_i^t(b_i) = a_i^t(b_i^+)$, otherwise condition is trivially satisfied. In this case, we will show that for any two time period t, s if $t - a_i^t(b_i) = s - a_i^s(b_i^+)$ then allocation to all the other advertisers are same i.e. $a_j^t(b_i) = a_j^s(b_i^+)\ \forall j \neq i$. The above statement implies that if the number of times advertiser i is not allocated the slot remains same for two time periods t and s with different bids b_i and b_i^+, then these different bids do not affect the allocations to other agents since their bids are not changed.

To prove this we use induction. As the base step consider $t - a_i^t(b_i) = s - a_i^s(b_i^+) = 0$. This implies that all the allocations till time t with bid b_i and till time s with bid b_i^+ is given to advertiser i and thus allocation to other agents is 0. Thus, the condition is trivially satisfied. Now assume that $t - a_i^t(b_i) = s - a_i^s(b_i^+) = a$ then by induction hypothesis $a_j^t(b_i) = a_j^s(b_i^+)\ \forall j \neq i$. Now, we will prove that if $t - a_i^t(b_i) = s - a_i^s(b_i^+) = a + 1$ then $a_j^t(b_i) = a_j^s(b_i^+)\ \forall j \neq i$. Let t' and s' be the latest rounds such that $t' - a_i^{t'}(b_i) = s' - a_i^{s'}(b_i^+) = a$, then we have $a_j^{t'}(b_i) = a_j^{s'}(b_i^+)\ \forall j \neq i$. Now, note that t' and s' are chosen such that $a_i^{t'+1}(b_i) = a_i^{s'+1}(b_i^+) = 0$. Thus, these rounds are given to some other agents. But Algorithm 9 allocates the advertiser on the basis of its UCB index which only depend on his bid and does not depend on other bidders. Also since allocations in the previous rounds are same and stack realization is fixed, the index will remain same. Thus, same agent will be selected. Moreover, from rounds $t' + 2$ to t and $s' + 2$ to s agent i will be allocated with bids b_i and b_i' respectively.

Thus, we have if $a_i^t(b_i) = a_i^t(b_i^+)$ then $a_j^t(b_i) = a_j^t(b_i^+)\ \forall j \neq i$. Now, since UCB index is non decreasing in bid b_i, UCB index of advertiser i will be more with bid b_i^+. Also since all other parameters are fixed, advertiser i will be selected with bid b_i^+ if it gets selected with bid b_i. Thus, we have $a_i^t(b_i^+) \geqslant a_i^t(b_i)$. ■

Lemma 10. *Allocation rule given by Algorithm 9 achieves regret of* $O(\ln(T))$.

The allocation rule given by the Algorithm 9 is similar to that of UCB algorithm (Algorithm 2) excluding the bid term of the agent. If the goal of the mechanism is to maximize the social welfare i.e. to select the advertiser with highest expected valuation which is given by $\theta_i \mu_i$, then it can be shown that the regret achieved by the Algorithm 9 will be of the order $O(\ln T)$ (same as UCB) given the mechanism is truthful.

Note that, stochastic truthfulness is a weaker notion as compared to ex-post truthfulness. Babaioff et. al. [4] present an ex-post incentive compatible and ex-post individual rational mechanism that achieve regret similar to UCB algorithm by designing an ex-post monotone allocation rule.

5.8 Multiple Pull Variants of MAB Mechanisms

So far, we have discussed the scenario where there is a single arm that must be pulled out of K competing arms. In the example of sponsored search auction, it means that the K advertisers compete for a single slot. Now we look at recent work where the K advertisers compete for $1 < l < K$ ordered slots. A slot $i < j$ is preferred by every advertiser. The advertisers submit bids and the mechanism designer must select l winning advertisers based on their bids and the corresponding click probabilities of the ads.

Gatti et. al. [12] design a mechanism for such a framework by proposing an exploration-separated mechanism where the payments to the advertisers are given as per the VCG mechanism. They considered a cascade model where the click probabilities of advertisements shown at a particular slot gets discounted based on the slots and advertisers that precedes it. They specifically address the problem where click precedence property is assumed where the assumption is that if an advertisement i is clicked when displayed at slot j then it also gets clicked when displayed at slot m with $m < j$.

The allocation rule is such that agents with the maximum values of $\mu_i b_i$ are selected, where μ_i is the probability that the ad i is clicked, once it has been observed by a user and b_i is the bid of the corresponding advertiser. The click probabilities $\mu_i's$ are learned in the exploration phase where 0 payment is asked from the advertisers. The learned values $\hat{\mu}_i's$ are then used to determine the allocation. There are three variants considered in the work: 1) position dependent, 2) ad dependent, and 3) contextual bandits. In position dependent variant click probabilities depend only on the position of the slot and not on the advertisers. In this case, advertisers with l highest values of $\hat{\mu}_i b_i's$ are allocated to l slots. Regret bounds are proved by bounding the number of exploration rounds required and is proved to be $O(T^{\frac{2}{3}} l^{\frac{2}{3}})$.

In the ad dependent setting, the probability of an advertisement i being clicked, when displayed at slot j, is a function of the position j and also varies across ads. The regret in this setting is proved to be $O(n^{\frac{2}{3}} l^{\frac{4}{3}} K^{\frac{1}{3}})$. The final variant considered in this work is of contextual bandits. The assumption is that μ_i is a function of some contextual information represented as a vector $\mathbf{x}_i$, that is, $\mu_i = \phi(\mathbf{x}_i)^\top \mathbf{w}$, where $\phi(.)$ is a collection of d basis functions. The vector $\mathbf{w}$ is unknown and is estimated using regression in the exploration phase. The regret becomes $O(n^{\frac{2}{3}} l^{\frac{2}{3}} n(d+1)^{\frac{1}{3}})$.

Later Mandal et. al. [18] prove impossibility results which state that if click precedence property is not satisfied than it is impossible to get sublinear regret in the case of position and ad dependent externalities. Moreover, they also provide an ex-post monotone allocation rule for position dependent externalities by assuming click precedence property. The allocation rule along with the payment given by general transformation mechanism described in Section 5.6 produces an ex-post truthful mechanism and achieves a regret of $O(kT^{1/2})$.

Though a mechanism is provided by [12], a characterization of the MAB mechanisms for multiple slot machines was not provided. Sharma et al [24] provided a characterization of mechanisms under several situations depending on the nature of the click through probabilities (CTRs). They defined the notion of pointwise monotonicity (strong and weak), type-1 and type-II separatedness for allocation rules and developed the characterization using these notions. An allocation rule is said to be weak pointwise monotone when for every agent i with bid b_i is alloted a slot j, bidding a higher value b_i^+ fetches him a slot $j' > j$ when all other parameters (the bids of the rest of the agents and clicks of the slots) do not change. In general an allocation rule is clickwise monotone if the number of clicks for an advertisement is a non-decreasing function of the bid. Influential agent-slot pairs are those such that a change in their number of clicks changes the allocation of some agent in future rounds. Type-I separatedness refers to the property that when an agent increases her bid, the allocation in the originally influential slots does not change, provided all other parameters are fixed. Type-II separatedness refers to the stronger property that, when an agent increases her bid, the allocation in the originally influential slots does not change, provided all other parameters are fixed and in addition, the originally influential slots continue to remain influential. We now state the key characterization result by [24].

Lemma 11. *Characterization of multiple pull MAB mechanisms*

1. *When the CTRs are unconstrained, any truthful mechanism must satisfy strong pointwise monotonicity. The regret is* $O(T)$ *for such mechanisms.*
2. *When the clicks on the ads follow a click precedence property, the necessary condition for a truthful MAB mechanism is weak pointwise monotonicity.*
3. *If a pre-estimate of the CTRs are available which maybe updated during the* T *rounds, weak pointwise monotonicity and type-I separatedness are necessary while weak pointwise monotonicity and type-II separatedness are sufficient conditions for the MAB mechanism to be truthful.*
4. *If the CTRs can be decomposed into agent-specific and slot-specific terms, the MAB mechanisms are truthful in expectation if the allocation is weak pointwise monotone and type-II separated.*

6 Recent Results in MAB Mechanisms

In the previous section, we have mainly focused on MAB mechanism in the context of sponsored search auctions. In this section, we provide pointers to more recent literature on MAB mechanisms for other applications.

Jain et al. [13] look at a multiple pull variant of MAB mechanism for crowdsourcing. The objective is to choose a minimum cost subset of workers whose aggregated label is guaranteed to achieve an assured accuracy for each task. Since the target accuracy depends on the quality of the workers which is unknown, the optimization problem involves stochastic constraints. Moreover, workers are strategic in reporting their costs. A MAB mechanism called Constrained Confidence Bound for a strategic setting (CCB-S) was proposed to elicit the costs from the workers while simultaneously learning the qualities and also satisfying the constraint. The authors further show that, the allocation rule in CCB-S is ex-post monotone and thereby one can use transformation rule presented in 5.6 to derive a mechanism that is ex-post incentive compatible and ex-post individually rational.

A single pull sleeping bandit variant for crowdsourcing is considered in [7]. Here each task assigned to the worker has a strict deadline until which the worker is not available for other tasks. They further assume that the requester has a certain budget and he cannot assign the tasks once the budget is exhausted. Thus, the number of pulls is constrained by the budget of the requester. They consider a homogeneous task deadline model where every task has the same deadline and thus every allocated worker is not available for a fixed number of time steps. The authors pose this as a knapsack problem and provide an exploration-separated mechanism for it. They further characterize any MAB mechanism in this setting and prove that any deterministic truthful MAB mechanism in this setting has to be exploration-separated.

A contextual bandit variant in the context of smart grids is considered in [15]. The authors seek to solve the problem of supply demand imbalance by proposing incentives for the consumers to cut down their electrical consumption. To design the right incentives for the consumers, their preferences are elicited. The probability of accepting such an incentive is stochastic and is learnt via MAB mechanism. The shortage of electricity that the distributor company faces at every time is an input to the mechanism. This input is also taken into consideration for selection of the appropriate subset of consumers to be given monetary offers.

The papers discussed so far in this survey consider the problem of designing MAB mechanisms that maximize social welfare. Social welfare denotes the sum of valuations of all the agents. However, the design of optimal mechanisms (that maximize the revenue) have not been addressed. Recent work by Bhat et. al. [6] tackles this problem in a multi-dimensional setting in the context of procurement auction. Each agent besides having their cost as private information are limited by the number of resources they can provide. This limited number of resources is also the private information of the agents. Each resource supplied by the agent has a certain quality which is stochastic in nature. The authors address this problem by proposing a MAB mechanism in this setting. They first discuss a structure of an optimal mechanism in this setting which is always truthful and individual rational. They further design a MAB mechanism consisting of a monotone allocation rule based on UCB algorithm and then extend the transformation mechanism provided in Section 5.6 to the procurement setting.

In this paper, we have talked about auction mechanisms where the agents are asked to report their valuations and their qualities are learnt. However, there are recent papers on MAB mechanisms which involve designing pricing strategies with online workers in crowdsourcing. Babaioff et al. [3] use a MAB mechanism to determine an optimal pricing mechanism for crowdsourcing assuming homogeneous qualities under a specified budget (known as bandits with knapsack). Singla and Krause [25] assume costs to be private information and propose a posted price mechanism to elicit the true costs from the users using MAB mechanisms while maintaining a budget constraint. [3, 25] consider homogeneous quality workers arriving online and seek to learn the distribution of the valuation of the workers.

7 Summary and Directions for Future Work

This paper discusses fundamental results and some recent advances in the area of stochastic multi-armed bandit mechanisms. As discussed in the paper there are many applications where stochastic multi-armed bandit mechanisms are useful. Furthermore, there is wide scope for their application to many other interesting emerging problems involving online learning. While the multi-armed bandit problem is well studied, literature in multi-armed bandit mechanisms is still nascent. There are many open problems in this area. Here, we list a few of them:

- **Characterization results for the relaxed problem:** The characterization results in the current literature make the strong assumption of agents being aware of the reward realizations. This leads to the requirement that the mechanism should be truthful with respect to all reward realizations. In a practical scenario however, agents are also unaware of the reward realization. It is therefore sufficient to design mechanisms that are truthful in expectation of the reward realizations. This characterization is still an open question.

- **Randomized mechanisms with low variance:** The transformation provided in Section 5.6 exhibits a trade-off between the variance and loss in revenue (controlled by parameter μ). If the parameter μ is kept very low so as to achieve low variance in the mechanism, then the rebate R_i given to agent i may be high and thus the mechanism's revenue may be very low. A study of other randomized mechanisms which achieve lower variance is not addressed in literature.
- **Mechanisms based on Thompson sampling:** It is shown in [16] that the regret achieved by both Thompson sampling and KL-UCB is slightly better than that of UCB based algorithms. However, the current state of the art does not include any mechanism that uses the allocation policy given by Thompson sampling or KL-UCB. Towards design of such mechanisms, it is worthwhile to prove or disprove the monotonicity of the allocation rules embedded in these algorithms.
- **Multiple pull variants of MAB mechanisms:** Much of the work on multiple pull variants of MAB mechanisms address the application to multiple slot sponsored search auction. The problem of multiple slot sponsored search auction with l slots reduces to finding the l best advertisers. However, this problem is not of combinatorial nature. In a more recent work [14], the authors tried to address a combinatorial problem in the context of crowdsourcing. Providing a characterization for combinatorial MAB mechanisms is another interesting future direction.

References

1. S. Agrawal and N. Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. *25th annual Conference on Learning Theory, COLT'12*, 23:39.1–39.26, 2012.
2. P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Journal of Machine Learning*, 47(2-3):235–256, 2002.
3. M. Babaioff, S. Dughmi, R. Kleinberg, and A. Slivkins. Dynamic pricing with limited supply. In *Thirteenth ACM Conference on Electronic Commerce EC'12*, pages 74–91. ACM, 2012.
4. M. Babaioff, R. D. Kleinberg, and A. Slivkins. Truthful mechanisms with implicit payment computation. In *Eleventh ACM Conference on Electronic Commerce EC'10*, pages 43–52. ACM, 2010.
5. M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms: extended abstract. In *Tenth ACM Conference on Electronic Commerce, EC'09*, pages 79–88. ACM, 2009.
6. S. Bhat, S. Jain, S. Gujar, and Y. Narahari. An optimal bidimensional multi-armed bandit auction for multi-unit procurement. In *Fourteenth International Conference on Autonomous Agents and Multiagent Systems, AAMAS'15*, pages 1789–1790, 2015.
7. A. Biswas, S. Jain, D. Mandal, and Y. Narahari. A truthful budget feasible multi-armed bandit mechanism for crowdsourcing time critical tasks. In *Fourteenth International Conference on Autonomous Agents and Multiagent Systems, AAMAS'15*, pages 1101–1109, 2015.
8. S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.
9. Y. N. D. Garg and S. Gujar. Foundations of mechanism design: A tutorial - Part 2: Advanced Concepts and Results. *Sadhana - Indian Academy Proceedings in Engineering Sciences*, 33(2):121–174, 2008.
10. N. R. Devanur and S. M. Kakade. The price of truthfulness for pay-per-click auctions. In *Tenth ACM Conference on Electronic Commerce, EC'09*, pages 99–106, 2009.
11. A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *COLT 2011 - The 24th Annual Conference on Learning Theory, June 9-11, 2011, Budapest, Hungary*, pages 359–376, 2011.
12. N. Gatti, A. Lazaric, and F. Trovò. A truthful learning mechanism for contextual multi-slot sponsored search auctions with externalities. In *Thirteenth ACM Conference on Electronic Commerce, EC'12*, pages 605–622, 2012.
13. S. Jain, S. Gujar, S. Bhat, O. Zoeter, and Y. Narahari. An incentive compatible multi-armed-bandit crowdsourcing mechanism with quality assurance. *CoRR*, abs/1406.7157, 2014.
14. S. Jain, S. Gujar, O. Xoeter, and Y. Narahari. A quality assuring multi-armed bandit crowdsourcing mechanism with incentive compatible learning. In *Thirtheenth International Conference on Autonomous Agents and Multiagent Systems*, pages 1609–1610, 2014.
15. S. Jain, B. Narayanaswamy, and Y. Narahari. A multiarmed bandit incentive mechanism for crowdsourcing demand response in smart grids. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 721–727, 2014.
16. E. Kaufmann, N. Korda, and R. Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory - 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings*, pages 199–213, 2012.
17. T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.
18. D. Mandal and Y. Narahari. A novel ex-post truthful mechanism for multi-slot sponsored search auctions. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 1555–1556, 2014.

19. R. B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.
20. Y. Narahari. *Game Theory and Mechanism Design*. World Scientific Publishing Company, 2014.
21. H. Nazerzadeh, A. Saberi, and R. Vohra. Dynamic cost-per-action mechanisms and applications to online advertising. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 179–188, New York, NY, USA, 2008. ACM.
22. H. Nazerzadeh, A. Saberi, and R. Vohra. Dynamic pay-per-action mechanisms and applications to online advertising. *Operations Research*, 61(1):98–111, 2013.
23. H. Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 09 1952.
24. A. D. Sharma, S. Gujar, and Y. Narahari. Truthful multi-armed bandit mechanisms for multi-slot sponsored search auctions. *Current Science*, 103(9):1064–1077, November 2012.
25. A. Singla and A. Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Twenty Second International World Wide Web Conference, WWW'13*, pages 1167–1178, 2013.
26. W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):pp. 285–294, 1933.