A Truthful Budget Feasible Multi-Armed Bandit Mechanism for Crowdsourcing Time Critical Tasks

Arpita Biswas*
Xerox Research Centre India
Bangalore, India
arpita.biswas@xerox.com

Shweta Jain†
Indian Institute of Science
Bangalore, India, 560012
jainshweta@csa.iisc.ernet.in

Debmalya Mandal‡
School of Engineering and Applied Sciences
Harvard University, Cambridge, MA - 02138
dmandal@g.harvard.edu

Y. Narahari
Indian Institute of Science
Bangalore, India, 560012
hari@csa.iisc.ernet.in

ABSTRACT

Motivated by allocation and pricing problems faced by service requesters on modern crowdsourcing platforms, we study a multi-armed bandit (MAB) problem with several real-world features: (a) the requester wishes to crowdsource a number of tasks but has a fixed budget which leads to a trade-off between cost and quality while allocating tasks to workers; (b) each task has a fixed deadline and a worker who is allocated a task is not available until this deadline; (c) the qualities (probability of completing a task successfully within deadline) of crowd workers are not known; and (d) the crowd workers are strategic about their costs. We propose a mechanism that maximizes the expected number of successfully completed tasks, assuring budget feasibility, incentive compatibility, and individual rationality. We establish an upper bound of $O(B^{2/3}(K\ln(KB))^{1/3})$ on the expected regret of the proposed mechanism with respect to an appropriate benchmark algorithm, where B is the total budget and K is the number of workers. Next, we provide a characterization of any deterministic truthful mechanism that solves the above class of problems and use this characterization to establish a lower bound of $\Omega(B^{2/3}K^{1/3})$ on the expected regret for any budgeted MAB mechanism satisfying the above properties.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; I.2.6 [**Learning**]: Parameter learning

*This work was done when the author was a student at Indian Institute of Science

†This work was partially supported by a Microsoft Research doctoral fellowship grant to the author.

‡This work was done when the author was a research associate at Indian Institute of Science

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

Keywords

Multi-Armed Bandits; Mechanism Design; Online Learning; Crowdsourcing; Regret Bound; Rational Agents

1. INTRODUCTION

Over the past decade, crowdsourcing has received significant attention for its utility in solving problems that require intelligence and effort of human beings. In a typical crowdsourcing platform, the requesters submit their tasks to the platform and expect the tasks to be completed with high quality. The requesters are required to make online decisions about the tasks to be assigned to specific workers and the payments to be offered. Often these decisions are complex, requiring the use of algorithms that learn different attributes of the workers over time, such as quality of the workers, time taken by a worker to complete a task, etc., besides taking onto account the strategic behavior of the workers.

In this paper, we study a *budgeted multi-armed bandit mechanism*, motivated by online crowdsourcing platforms like *guru.com, elance.com, rent-acoder.com etc.*, where each requester posts tasks and workers bid for the desired tasks. In *rent-acoder.com*, for example, a requester posts a project that has to be completed within a budget and has a fixed deadline. Once the project is posted, various registered workers can bid for the project. The requester then assigns the project to one or more workers depending on their bids and his past experience about the quality of the workers. However, if the requester is completely unaware of the quality of the workers bidding on the project, the requester would like to learn the qualities of these workers while ensuring that the project is completed within the project deadline. This could be done by dividing the project into smaller tasks and giving these tasks to the workers in a sequential manner. Deadlines of these tasks should be planned in accordance with the overall project deadline. Submissions made by the allocated workers could be evaluated for quality. For example, if a task is completed with high quality by a worker, the same worker can be given another task. Also, for posting a new task, the requester has to choose among only the available workers.

Motivated by such crowdsourcing platforms, we consider a model where a requester has a set of homogeneous tasks that need to be completed within a fixed budget. Each task has a fixed deadline which is assumed to be the same for all the tasks. Each worker on the other hand has a fixed quality which is unknown to the requester and cost for completing a task is private information of the worker. In order to maximize the quality, the requester assigns the tasks to the workers one by one so as to learn their qualities. We assume that the success or failure of a task can be verified instantly as soon as the task deadline elapses, and thus the quality of the workers can be estimated. The estimated quality is useful to determine future allocations. No task is allocated to the worker until the deadline for the allocated task elapses. In order to utilize the budget in the best possible way, the requester has to incentivize the workers to bid their true cost of effort, at the same time, ensure voluntary participation. Thus, the requester seeks to maximize the expected number of tasks completed successfully within the fixed budget, besides ensuring truthfulness and voluntary participation of the strategic workers.

The above situation is an example of sequential decision making in an uncertain environment. Here, the requester seeks to optimize his allocation and payment decisions while continuously gathering more information about the qualities of the workers. This leads to a trade-off between exploration (allocating tasks to all workers sufficiently often to obtain better estimates of the quality of each worker) and exploitation (allocating each task to the best available worker). These kind of problems naturally fall into multi-armed bandit (MAB) problems. The presence of strategic agents in multi-armed bandit problems leads to multi-armed bandit (MAB) mechanism design problems. The budget constraint leads to *budgeted MAB mechanism design* problems. We have, in addition, deadlines for tasks which have to be honored.

1.1 Our Contributions

There exist several papers in the literature (we provide a review in the next section) that deal with budgeted multi-armed bandit problems. However, there is no existing work that additionally captures the task deadlines and strategic nature of workers over their costs. The main contributions of our work are as follows.

- We propose a MAB mechanism that takes into account limited budget, task deadlines, unknown qualities, and strategic workers (strategic about their costs). Note that the quality of a worker refers to the probability of the worker completing a task successfully within the given deadline. Our mechanism maximizes the expected number of tasks completed successfully subject to budget feasibility, incentive compatibility, and individual rationality. We believe this is the first effort in designing a mechanism for this class of problems.

- We establish an upper bound of $\mathrm{O}(B^{2/3}(K\ln(KB))^{1/3})$ (Theorem 2) on the expected regret of the proposed mechanism with respect to an appropriate benchmark algorithm, where B is the total budget and K is the number of workers. The uncertainty in the availability of a worker, the budget constraint, and the strategic nature of the workers render the regret analysis challenging.

- We provide a characterization of any deterministic truthful mechanism that solves the above class of problems. We establish a lower bound $\Omega(B^{2/3}K^{1/3})$ on the expected regret (Theorem 6).

2. RELATED WORK

Multi-armed bandit problems have been studied extensively for solving problems in different domains [1, 4, 7]. The budgeted multi-armed bandit that is close to our setting is considered by Tran-Thanh et al. [18, 17], without strategic agents and task deadlines. The algorithms in [18, 17] achieves regret of $\mathrm{O}(B^{2/3})$ and $\mathrm{O}(\ln(B))$ respectively. A more general formulation is considered by Agrawal and Devanur [2] where the authors attempt to maximize a concave objective function with convex constraints via multiarmed bandit algorithm. However, these algorithms do not consider the strategic behavior of the agents and assume that the costs are private knowledge [18, 17] or stochastic [2].

Budgeted MAB problems have also been widely studied for pricing tasks (or items) in crowdsourcing (or dynamic procurement) problems. With workers arriving online with a fixed and known distribution, Singer et al. [15] considered a budgeted setting with a goal to maximize the total number of allocated tasks to the workers. The workers were assumed to complete the task successfully if allocated and thus, the goal was to design a pricing mechanism to complete the tasks within a budget. In our setting, workers complete the allocated task with a fixed probability which is unknown and we wish to design an auction mechanism by incentivizing the workers to bid their true cost of effort. Badanidiyuru et al. [6] and Ho et al. [11] modeled dynamic procurement and crowdsourcing problems as MAB problems where arms corresponded to feasible posted-prices. However, they did not captured task deadlines and strategic nature of the workers. The survey paper by Slivkins et al. [16] lists results for various crowdsourcing problems and provides insights for possible directions of research in this area.

In MAB problems, the need for mechanism design arises when each arm holds some private valuations. Thus, in addition to learning the unknown parameters, the mechanism is also required to elicit the private valuations of the arms truthfully. Most of the research in this area deals with forward auction, for example, auction of ad-slots on a webpage among advertisers, where the click probabilities of the advertisers are to be learnt and the value of an advertisement is held privately by the corresponding advertiser. In the online advertising context, MAB mechanisms aim at maximizing the revenue of the platform or the social welfare as a whole. Devanur et al. [9] showed that the truthful restriction on pay-per-click online advertising problem imposes statistical limits on achievable regret in terms of revenue and thus the achievable regret is very high ($\Theta(n^{2/3})$). Babaioff et al. [5] proved that any truthful mechanisms for forward auction must separate exploration and exploitation, and the regret in terms of social welfare is $\Omega(n^{2/3}K^{1/3})$. Our work, when compared to the existing MAB mechanisms, is novel due to the extension to the case of limited budget and non-availability of workers due to task deadlines.

In the absence of learning, Singer et al. [14] considered the problem of budget feasible truthful mechanism for non-decreasing submodular valuation functions. We wish to point out here that while we adopt allocation techniques from standard MAB variations like budgeted MAB by Guha et al. [10] and sleeping bandits due to unavailability by Kleinberg et al. [12], however, the presence of strategic agents requires a carefully designed payment rule that makes regret analysis interesting.

3. THE MODEL

In this section, we formalize the *budgeted MAB mechanism design* problem for crowdsourcing scenario. We consider a requester and a fixed set of K workers denoted by $N = \{1, \ldots, K\}$. The requester has a set of homogeneous tasks to be completed within a budget B. In addition to the budget, the requester also has a fixed deadline and he wishes to complete all the tasks within the deadline. On the other hand, each worker $i \in N$ is associated with a quality q_i and incurs a cost c_i for completing a task. The quality q_i represents the probability with which the worker i successfully completes the allocated task within the specified deadline. The qualities are initially unknown to the requester as well as the workers, whereas, the costs of the workers are the private information held by the respective workers. We consider a general version of this problem wherein the costs and qualities do not depend on each other. The requester's objective is to design a mechanism that maximizes the expected number of successfully completed tasks within the budget B, in the presence of strategic workers. The model is described in Figure 1.

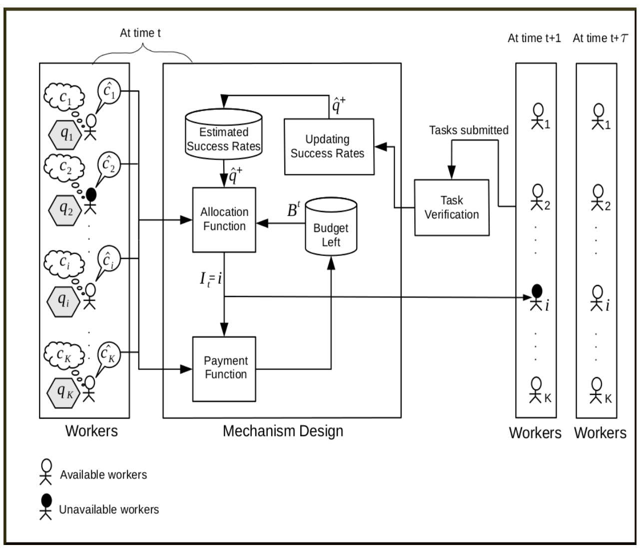

Figure 1: A pictorial representation of the model

Since the qualities of the workers are stochastic and unknown, the requester is required to learn the qualities. So, the requester posts one task per time step, allocates the task to a worker, observes the success of the task, and thus, gains a better estimate of the quality of the worker after each time step. In this paper, we assume that the requester gives a fixed deadline of τ time steps for each task posted, which causes an additional complexity to the *budgeted MAB mechanism design* problem. When a task is allocated to a worker i, the worker becomes unavailable for next τ time steps as the worker is busy executing the task. Thus, no further tasks can be assigned to the worker until the next τ time steps. The tasks are homogeneous, that is, each worker i completes any task successfully with a probability q_i, incurs a fixed cost c_i and each task has a fixed deadline of τ. Thus, these tasks are time critical tasks.

Symbol	Description
K	Number of workers
N	Set of workers $\{1, 2, \ldots, K\}$
B	Total budget available
q_i	Quality (probability of success) of worker i
$\hat{q}_i$	Estimated quality of worker i
$\hat{q}_i^+$	Upper confidence bound on $\hat{q}_i$
c_i	True cost of worker i
c	Vector of true costs $c = (c_1, \ldots, c_K)$
$\hat{c}_i$	Bid of worker i
$\hat{c}_{-i}$	Bid vector of all the workers except i
$[\underline{c}, \overline{c}]$	Minimum and maximum bids
τ	Fixed deadline for executing each task
B^t	Budget remaining with the requester after $t-1$ time steps
s	Success realization where, $s_{i,t}$ is the indicator variable denoting success of the task submitted at time step t by worker i
$\mathcal{A}_i^t(\hat{c}; s; B^t)$	Indicator function denoting whether the worker i is allocated for a task at time step t
$\mathcal{P}_i^t(\hat{c}; s; B^t)$	Payment for a task to an allocated worker i at time step t
$\mathcal{U}_i^t(\hat{c}; c_i; s; B^t)$	Utility of a worker i for task at time step t
$\mathbb{E}[R]$	Expected regret
$\underset{k \in N}{argmax^n}[a]$	Gives the index of n^{th} maximum value in an array $[a]$

Table 1: Table of Notations

Let s be a success realization matrix, where $s_{i,t} \in \{0, 1\}$ denotes whether or not (1 or 0) a task submitted by a worker i at time step t is successfully completed within deadline. The worker i submits the allocated task at $(t+\tau)^{th}$ time, becomes available, and the success bit $s_{i,(t+\tau)} \in \{0, 1\}$ is observed at $(t+\tau)^{th}$ time step for estimating the worker's quality, where $s_{i,(t+\tau)} = 1$ with probability q_i and $s_{i,(t+\tau)} = 0$ with probability $1-q_i$. Thus, a task allocated at time step t can be unsuccessful in two ways, first, if the allocated worker fails to submit the task at time step $t+\tau$, and second, an incorrect submission (e.g. code submitted to *rentacoder.com* might fail to work). A task is allocated only to one of the available workers at each time step.

When a task is allocated to a worker, the requester has to make payment to the worker irrespective of the task being successfully completed. In order to decide the payment, the requester conducts a reverse auction where the workers bid for taking up the task. We assume that the workers are allowed to bid only at the starting of the auction and their costs remain same for all the tasks as the tasks are homogeneous. The requester also provides a lower limit and upper limit on the bids, $\underline{c}$ and $\overline{c}$ respectively. Lower limit $\underline{c}$ can be 0 and upper limit $\overline{c}$ indicates that the requester is not willing to pay more than $\overline{c}$ per task. The bids for all the workers are denoted by a vector $\hat{c} = (\hat{c}_1, \ldots, \hat{c}_K) \in [\underline{c}, \overline{c}]^K$.

The requester needs to define an allocation rule $\mathcal{A}$ and a payment rule $\mathcal{P}$ to ensure that the workers bid their true costs, which leads to *budgeted MAB mechanism design*. Let $s^t = (s_{i,t'})^{i \in N}_{t' \in \{1,\ldots,t-1\}}$ and B^t denote the success realization and the budget left respectively till $t-1$ time step. An allocation rule $\mathcal{A}$ is a function that maps the bid vector $\hat{c}$, the success realization s^t, and the budget left B^t to a worker $i \in N$ for each time t. For each task a worker i is allocated, a payment $\mathcal{P}_i^t$ is given to the worker. The requester has a budget B, and the total payment given to the workers cannot exceed the budget B. Also, not all the entries in s^t are known at a time step t, since only the success of the allocated workers is only revealed. Thus, the allocation $\mathcal{A}$ and payment $\mathcal{P}$ depend only on the observed entries of success realization s^t.

Now, ideally, the bid $\hat{c}_i$ must be equal to the true cost c_i for each worker i, however, the workers, being strategic agents, may bid a value different from their true costs to maximize their own utilities. The utility of a worker i at time step t is given as

$$\mathcal{U}_i^t(\hat{c}_i; c_i; s^t; B^t) = (\mathcal{P}_i^t(\hat{c}; s^t; B^t) - c_i)\mathcal{A}_i^t(\hat{c}; s^t; B^t) \quad (1)$$

where,

$$\mathcal{A}_i^t(\hat{c}; s^t; B^t) = \begin{cases} 1 & \text{if worker } i \text{ is allocated task at time } t \\ 0 & \text{otherwise} \end{cases}$$

We now present some essential properties that our mechanism should satisfy.

Definition 1 ***(Truthful)*** *A mechanism is* truthful *if bidding true costs maximizes the utility of any worker i irrespective of the bids of other workers. Formally,* $\forall \hat{c}_{-i}, \hat{c}_i, c_i, s^t, B^t$,

$$\mathcal{U}_i^t(c_i, \hat{c}_{-i}; c_i; s^t; B^t) \geq \mathcal{U}_i^t(\hat{c}_i, \hat{c}_{-i}; c_i; s^t; B^t). \quad (2)$$

In this work, we consider the strongest notion of truthfulness also known as Dominant Incentive Strategy Compatible (DSIC) where no worker has incentive to misreport his bid irrespective of the bids of other workers.

Definition 2 ***(Individually Rational)*** *A mechanism is* individually rational *if every worker i derives a non negative utility by participating in the auction. Formally,* $\forall c, s^t, B^t$,

$$\mathcal{U}_i^t(c; c_i; s^t; B^t) \geq 0. \quad (3)$$

Note that, even if the workers are bidding only once, the truthfulness is defined based on the utility, a worker achieves at every time. This is because, workers are unaware of the budget of the requester and thus will wish to maximize the utility at every time step. As in the later rounds, budget might get over. The non-strategic version of our problem can be mapped to a budgeted MAB problem [18] where workers represent the arms, tasks represent time steps and allocating a task to the worker corresponds to pulling an arm. Table 1 provides the notation that we will be using throughout.

4. BUDGETED MAB MECHANISM WITH TIME CRITICAL TASKS

4.1 A Benchmark Mechanism

In order to compare the performance of our mechanism, we consider a benchmark mechanism that knows the quality q_i for each worker i, but still has to incentivize the workers to bid truthfully. Thus, the benchmark algorithm is not required to learn the qualities of the workers, however, it should still satisfy properties like budget feasibility, truthfulness, and individual rationality.

Let $T_c(B)$ denote the total number of tasks that can be executed with budget B and cost vector c. Note that the total number of tasks also depend on the payment which in turn depends on the costs and qualities. We have dropped this dependence for notational brevity. Let us denote $\mathcal{A}_i(\hat{c}; q; B) = \sum_{t=1}^{T_{\hat{c}}(B)} \mathcal{A}_i^t(\hat{c}; q; B)$ to be the number of tasks allocated to a worker i when the bid profile is $\hat{c}$ and the known quality vector is q. Let the payment given to a worker i is denoted by $\mathcal{P}_i(\hat{c}; q; B) = \sum_{t=1}^{T_{\hat{c}}(B)} \mathcal{P}_i^t(\hat{c}; q; B)$. If $\tau = 0$, i.e., all the workers are available for all tasks, we have the following optimization problem:

$$\begin{aligned} \underset{(\mathcal{A}(\hat{c};q;B), \mathcal{P}(\hat{c};q;B))}{\text{maximize}} \quad & \sum_{i=1}^{K} \mathcal{A}_i(\hat{c}; q; B) q_i \\ \text{subject to} \quad & \sum_{i=1}^{K} \mathcal{P}_i(\hat{c}; q; B) \leq B \\ & \mathcal{A}_i(\hat{c}; q; B) \in \mathbb{Z} \quad \forall i \in N \end{aligned} \quad (4)$$

where, worker i's payment $\mathcal{P}_i^t(\hat{c}; q; B)$ and the number of allocations $\mathcal{A}_i^t(\hat{c}; q; B)$ satisfy the property of truthfulness and individual rationality given in Definitions 1 and 2 respectively. If we assume that the costs are not private information then this optimization problem reduces to the unbounded knapsack problem which is NP-hard and is considered in [18] when costs are public knowledge. Thus, we adopt a similar benchmark where the best worker is identified according to highest quality by cost ratio. However, there is a non-zero task deadline, $\tau > 0$, allocating a task to a worker implies that the worker remains unavailable for the next τ time steps. The idea is to allocate a task to the most efficient available worker at each time step. Without loss of generality, let us assume that all the workers are ranked according to their quality by cost ratio, that is, $\frac{q_1}{\hat{c}_1} \geq \frac{q_2}{\hat{c}_2} \geq \ldots \geq \frac{q_K}{\hat{c}_K}$. Workers are allocated greedily according to this ranking among all the available workers. By the end of the first τ time steps, the best worker is available, and the best worker gets the task at time step τ. So, instead of selecting one optimal worker, the allocation rule has to select best τ workers. Thus, the best τ workers are allocated one by one sequentially. The benchmark algorithm is given in Algorithm 1 where N_τ is the set of τ best workers, P denotes the sum of payments given to the best τ workers for taking one task each, and T denotes the number of tasks given to each worker.

ALGORITHM 1: *Benchmark mechanism*

Input: Bids $\hat{c}_1, \ldots, \hat{c}_K$, Qualities $q_1, \ldots, q_K$, Deadline τ, Maximum possible bid $\bar{c}$, and Budget B
Output: Mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$
Assumption: $\frac{q_1}{\hat{c}_1} \geq \frac{q_2}{\hat{c}_2} \geq \ldots \geq \frac{q_K}{\hat{c}_K}$ and $\tau < K$;
$N_\tau = \{1, \ldots, \tau\}$;
for *each i in* N_τ **do**
    Set $p_i = min\{\frac{q_i}{q_{\tau+1}} c_{\tau+1}, \bar{c}\}$;
$P = \sum_{i \in N_\tau} p_i$;
$T = \lfloor B/P \rfloor$;
for $t = 1$ *to* T **do**
    for *each i in* N_τ **do**
        $\mathcal{A}_i^{t*i-1} = 1$;
        $\mathcal{P}_i^{t*i-1} = p_i$;

Now, we show that the benchmark mechanism satisfies budget feasibility, truthfulness and individual rationality. The mechanism is budget feasible as $T = \lfloor \frac{B}{P} \rfloor$ ensures that the total payment is less than or equal to B. For ensuring truthfulness and individual rationality, a property called monotone allocation rule is required, and hence, we define monotone allocation.

Definition 3 ***(Monotone Allocation)*** *An allocation rule $\mathcal{A}$ is **monotone** if for any quality vector q, Budget B, $\hat{c}_i \leq c_i$,*

$$\mathcal{A}_i^t(c_i, c_{-i}; q; B) \leq \mathcal{A}_i^t(\hat{c}_i, c_{-i}; q; B). \tag{5}$$

Each worker i that belongs to the set of best τ workers ($i \in N_\tau$), gets the allocation only if his bid is less than $\frac{c_{\tau+1} q_i}{q_{\tau+1}}$, keeping all the other bids same. Thus, the allocation rule for the benchmark mechanism is monotone.

Also, the payment rule should satisfy some criteria to make the benchmark mechanism truthful and individually rational, according to the following Theorem.

Theorem 1 *[3, 13] A mechanism is incentive compatible (truthful) and individually rational if and only if for each agent i and bid vector $\hat{c}$, the allocation rule $\mathcal{A}_i^t$ is monotone, $\int_0^\infty \mathcal{A}_i^t(\hat{c}; q; B) d\hat{c} < \infty$ and the payment is given by*

$$\mathcal{P}_i^t(\hat{c}; q; B) = \hat{c}_i \mathcal{A}_i^t(\hat{c}; q; B) + \int_{\hat{c}_i}^{\overline{c}} \mathcal{A}_i^t(z, \hat{c}_{-i}; q; B) dz \tag{6}$$

*This payment is called **critical payment**.*

The critical value which is paid to each of the best τ workers is given by Equation (6).

From Algorithm 1, the payment for any worker $i \in N_\tau$, is given by $\mathcal{P}_i^t = \frac{c_{\tau+1} q_i}{q_{\tau+1}}$. To see that the payment satisfies Equation (6), we observe that the worker $i \in N_\tau$ lose allocation if $\hat{c}_i > \frac{c_{\tau+1} q_i}{q_{\tau+1}}$. Thus, the value of integration $\int_{\hat{c}_i}^{\overline{c}} \mathcal{A}_i^t(z, \hat{c}_{-i}; q; B) dz = \left(\frac{c_{\tau+1} q_i}{q_{\tau+1}} - \hat{c}_i \right) \mathcal{A}_i^t(\hat{c}_i, \hat{c}_{-i}; q; B)$ and thus, satisfies Equation (6). Note that, if all the best τ workers are not allocated the same number of tasks, it is hard to define the critical payments. Thus, we consider that $B - \lfloor \frac{B q_{\tau+1}}{c_{\tau+1} \sum_i q_i} \rfloor \frac{c_{\tau+1} \sum_i q_i}{q_{\tau+1}}$ budget remains unallocated to ensure truthfulness. Also, for worker $i \in N_\tau$, the payment for each task is $\frac{c_{\tau+1} q_i}{q_{\tau+1}}$ which is greater than or equal to c_i making the mechanism individually rational. Thus, the benchmark mechanism ensures budget feasibility, truthfulness and individual rationality. Note that, generalization of the above algorithm is difficult to the case when the task deadlines are heterogeneous i.e. each task has a different deadline, because the allocation is no longer uniform and this requires complicated payments to design a truthful mechanism.

The total expected reward (expected number of successfully completed tasks) accumulated by the benchmark algorithm is

$$\frac{B}{\sum_{k=1}^{\tau} \frac{q_k}{q_{\tau+1}} c_{\tau+1}} \sum_{k=1}^{\tau} q_k = B \frac{q_{\tau+1}}{c_{\tau+1}} \tag{7}$$

The *expected regret* for an algorithm with unknown success rates is given by

$$\mathbb{E}[R] = B \frac{q_{\tau+1}}{c_{\tau+1}} - \sum_{t=1}^{T_{\hat{c}}(B)} q_{I_t} \tag{8}$$

where, $T_{\hat{c}}(B)$ is the total number of tasks, and I_t denotes the worker allocated for task t. We bound the regret with respect to benchmark greedy algorithm, since it is hard to compute the optimal solution even when the qualities are known. Moreover, benchmark algorithm gives us an approximate solution with a factor of two.

4.2 Proposed Mechanism

In this section, we provide a mechanism for budgeted MAB problem with task deadlines given in Algorithm 2. The input parameters to the algorithm are bid vector $\hat{c}$, task deadline τ, budget B and the maximum allowed bid $\overline{c}$. We call our mechanism exploration-separated as the mechanism divides the budget B into exploration budget and exploitation budget. The value of exploration budget B_1 is given in Step 1 and is calculated in a way that minimizes the expected regret 8 (calculation provided in Section 4.4). In Section 4.5, we prove that the exploration separated property is necessary for any truthful and IR mechanism.

ALGORITHM 2: *Budgeted MAB mechanism* with task deadline

Input: Bids $\hat{c}_1, \ldots, \hat{c}_K$, Deadline τ, Maximum bid $\overline{c}$, Budget B
Output: Mechanism $\mathcal{M} = (\mathcal{A}, \mathcal{P})$

1 Initialize $t = 1; \hat{q}_k = 0$ and $n_k = 0 \ \forall k \in N$;
2 Set $B_1 = \frac{1}{(2)^{1/3}} (\overline{c} K \ ln(KB))^{1/3} B^{2/3}$;
3 **for** $l = 1, 2, \ldots, \lfloor \frac{B_1}{K \ \overline{c}} \rfloor$ **do**
4   **for** $i = 1, 2, \ldots, K$ **do**
5   **if** $t > \tau$ **then**
6   Let k= worker allocated at time $(t - \tau)$;
7   Observe reward $s_{k,t}$;
8   Update $\hat{q}_k = (\hat{q}_k n_k + s_{k,t})/(n_k + 1)$;
9   Update $n_k = n_k + 1$;
10   Allocate worker i, $\mathcal{A}_i^t = 1$, $\mathcal{P}_i^t = \overline{c}$;
11   Update $t = t + 1$;
12 **for** *each k in N* **do**
13   Update $\hat{q}_k^+ = \hat{q}_k + \sqrt{\frac{K \ \overline{c} \ ln(KB)}{2B_1}}$;
14 Set $N_\tau^{'} = \{[1], \ldots, [\tau]\}$;
15 Set $j = \underset{k \in N}{argmax}^{\tau+1} \ \frac{\hat{q}_k^+}{\hat{c}_k}$;
16 Set $B^t = B - \lfloor \frac{B_1}{K \ \overline{c}} \rfloor K \overline{c}$;
17 **while** $B^t > \sum_i \min\{\frac{\hat{q}_i^+}{\hat{q}_j^+} c_j, \overline{c}\}$ **do**
18   **for** *each i in $N_\tau^{'}$* **do**
19   Set $\mathcal{A}_i^t = 1$, $\mathcal{P}_i^t = min\{\frac{\hat{q}_i^+}{\hat{q}_j^+} c_j, \overline{c}\}$;
20   Allocate worker i, $\mathcal{A}_i^t = 1$ and pay $\mathcal{P}_i^t$;
21   Update $B^{t+1} = B^t - \mathcal{P}_i^t$;
22   Update $t = t + 1$;

The algorithm first explores the workers by allocating the tasks to the workers in a round robin fashion till budget B_1 is exhausted. Such rounds are known as exploration rounds. The algorithm maintains the running average of quality $\hat{q}_k$ obtained from each worker k in the exploration round. The per allocation payment for each worker during exploration phase is $\overline{c}$. After exploration rounds, the best τ workers, according to decreasing order of $\frac{\hat{q}_k^+}{\hat{c}_k}$ $\forall$ $k \in N$, are chosen to be played sequentially one by one in each round, where $\hat{q}_k^+ = \hat{q}_k + \sqrt{\frac{K\ \bar{c}\ ln(KB)}{2B_1}}$ is the upper bound quality estimate for worker k.

Let $[i]$ denote the i^{th} ranked worker and $N_\tau^{'}$ denote the best τ workers according to the ratio $\frac{\hat{q}_k^+}{\hat{c}_k}$, that is, $[i] = \underset{k\in N}{argmax}^i\ \frac{\hat{q}_k^+}{\hat{c}_k}$ and $N_\tau^{'} = \{[1]\ ,\ldots,\ [\tau]\}$. We use the notation $argmax^i$ to denote the parameter value that gives i^{th} maximum value of the corresponding array. The per allocation payment made to a worker k among the best τ workers is $\frac{\hat{q}_k^+\hat{c}_{[\tau+1]}}{\hat{q}_{[\tau+1]}^+}$, where $[\tau+1] = \underset{k\in N}{argmax}^{\ \tau+1}\ \frac{\hat{q}_k^+}{\hat{c}_k}$.

4.3 Properties of the Proposed Mechanism

Our mechanism satisfies the following desirable properties:

- **Truthfulness:** 1. *Monotone Allocation*: The allocation function during the exploration rounds is independent of the bids, so $\mathcal{A}_i^t(\hat{c}_i, \hat{c}_{-i}; s^t; B^t)$ does not change for a different bid $c_i^- < \hat{c}_i$. During the exploitation phase, the workers are allocated according to the ratio $\frac{\hat{q}_i^+}{\hat{c}_i}$. Since the qualities does not change in exploitation rounds, the allocation rule of the proposed mechanism is monotone. 2. *Critical Payment*: The critical payment is given by Equation (6). In exploration phase, the allocation is bid independent, that is, the allocation remains same for any $c_i' \in [c_i, \bar{c}]$. Thus, payment to an allocated worker during exploitation phase at each time is $\bar{c}$. During exploitation phase, any worker i among best τ workers lose allocation when $\hat{c} > \frac{\hat{q}_i^+\hat{c}_{[\tau+1]}}{\hat{q}_{[\tau+1]}^+}$. Thus, the payment using Equation (6) can be shown to be $\frac{\hat{q}_i^+\hat{c}_{[\tau+1]}}{\hat{q}_{[\tau+1]}^+}$ per round. As the payment rule of the mechanism matches the payment given by (6) with monotone allocation rule, the mechanism is truthful, that is $\hat{c}_i = c_i\ \forall i \in N$.

- **Individually Rationality:** The payment $\bar{c}$ during exploration phase and $\frac{\hat{q}_i^+\hat{c}_{[\tau+1]}}{\hat{q}_{[\tau+1]}^+}$ during exploitation phase to each worker i is always greater than or equal to $\hat{c}_i$, and hence, the mechanism is individually rational.

- **Budget Feasibility:** The algorithm stops when there is no more budget left to be paid to the workers in exploitation phase. Thus, the total payment given to all the workers does not exceed the given budget B. Thus, the mechanism is budget feasible.

- **Computationally Efficient:** Initially, the mechanism invites bids from all the workers, and using the observed qualities and bids, find best $\tau+1$ workers according to their quality per cost ratio. This takes $O(K)$ time [8].

The following section provides an upper bound on the expected regret and proposes the optimal value of B_1 for the algorithm.

4.4 Upper Bound Analysis

The performance of any mechanism depends upon the regret accumulated by the mechanism. The expected regret of the proposed algorithm (as given by Equation 8) is given as:

$$\frac{Bq_{\tau+1}}{c_{\tau+1}} - \frac{B_1}{K\ \bar{c}}\sum_{k\in N} q_k - \frac{B - B_1}{\frac{c_{[\tau+1]}}{q_{[\tau+1]}}\sum_{i\in N_\tau'} \hat{q}_{[i]}^+}\ \sum_{i\in N_\tau'} q_{[i]} \tag{9}$$

here, $[i]$ denotes i^{th} ranked worker according to the ratio $\frac{\hat{q}_k^+}{c_k}$.

The following theorem provides an upper bound on the regret by taking an optimal value of B_1.

Theorem 2 *The expected regret for the proposed algorithm is* $O(B^{\frac{2}{3}}\ K^{\frac{1}{3}}\ (\ln(KB))^{\frac{1}{3}})$.

We first provide bounds on the learnt quality after exploration phase in the following two lemmas.

Lemma 3 *After the exploration rounds, for each worker* $k \in N$, $q_k < \hat{q}_k^+$ *with probability at least* $1 - \frac{1}{KB}$.

Proof: Let each worker be allocated n times in the exploration phase, i.e. $n = \lfloor \frac{B_1}{K\ \bar{c}} \rfloor$. After n allocations we have,

$$\begin{aligned}\mathbb{P}\left(\hat{q}_k^+ \le q_i\right) &= \mathbb{P}\left(\hat{q}_k + \sqrt{\frac{\ln(KB)}{2n}} \le q_i\right)\\ &= \mathbb{P}\left(\hat{q}_k \le q_i - \sqrt{\frac{\ln(KB)}{2n}}\right)\end{aligned}$$

Applying Chernoff-Hoeffding's bound, we get:

$$\mathbb{P}\left(\hat{q}_k \le q_i - \sqrt{\frac{\ln(KB)}{2n}}\right) \le\quad e^{\left\{-2n\left(\sqrt{\frac{ln(KB)}{2n}}\right)^2\right\}}\quad = \quad \frac{1}{KB}$$

Therefore, with probability at least $(1 - \frac{1}{KB})$, for a worker k, $q_k < \hat{q}_k^+$. ■

Lemma 4 *After the exploration rounds, with probability at least* $1 - \frac{1}{B}$, $\frac{q_r}{c_r} \le \frac{\hat{q}_{[r]}^+}{c_{[r]}}$ *where,* $[r] = \underset{k\in N}{argmax}^r\ \frac{\hat{q}_k^+}{c_k}$

Proof: The probability that $\hat{q}_k^+$ is greater than q_k for every worker $k \in N$ is

$$\begin{aligned}&\mathbb{P}\left(\bigcap_{k\in N}\ (\hat{q}_k^+\ >\ q_k)\right)\\ =&1 - \mathbb{P}\left(\bigcup_{k\in N}\ (\hat{q}_k^+\ \le\ q_k)\right)\\ \ge&1 - \sum_{k\in N}\mathbb{P}(\hat{q}_k^+\ \le\ q_k)\\ \ge&1 - \sum_{k\in N}\frac{1}{KB} \qquad\qquad \text{(Using lemma 3)}\\ =&1 - \frac{1}{B}\end{aligned}$$

Hence, r^{th} max value of the set $\left\{\frac{q_k}{c_k} : k \in N\right\}$ is less than or equal to the r^{th} max value of the set $\left\{\frac{\hat{q}_k^+}{c_k} : k \in N\right\}$ with probability $(1 - \frac{1}{B})$. ■

Proof of Theorem 2: The expected regret of the algorithm can be upper bounded by

$$(1-\frac{1}{B})\left(\frac{B\ q_{\tau+1}}{c_{\tau+1}}-\frac{B_1}{K\ \bar{c}}\sum_{k\in N}q_k-\frac{(B-B_1)}{\frac{c_{[\tau+1]}}{q_{[\tau+1]}}\sum_{i\in N'_\tau}\hat{q}^+_{[i]}}\sum_{i\in N'_\tau}q_{[i]}\right)$$

$$+\frac{1}{B}\frac{B\ q_{\tau+1}}{c_{\tau+1}}$$

$$=(1-\frac{1}{B})\left(B_1\left(\frac{q_{\tau+1}}{c_{\tau+1}}-\frac{1}{K\ \bar{c}}\sum_{k\in N}q_k\right)\right.$$

$$\left.+(B-B_1)\left(\frac{q_{\tau+1}}{c_{\tau+1}}-\frac{\sum_{i\in N'_\tau}q_{[i]}}{\frac{c_{[\tau+1]}}{\hat{q}^+_{[\tau+1]}}\sum_{i\in N'_\tau}\hat{q}^+_{[i]}}\right)\right)+\frac{q_{\tau+1}}{c_{\tau+1}}$$

$$\leq(1-\frac{1}{B})\left(\frac{B_1q_{\tau+1}}{c_{\tau+1}}+(B-B_1)\left(\frac{q_{\tau+1}}{c_{\tau+1}}-\frac{\sum_{i\in N'_\tau}q_{[i]}}{\frac{c_{[\tau+1]}}{\hat{q}^+_{[\tau+1]}}\sum_{i\in N'_\tau}\hat{q}^+_{[i]}}\right)\right)$$

$$+\frac{q_{\tau+1}}{c_{\tau+1}}$$

$$\leq(1-\frac{1}{B})\left(\frac{B_1q_{\tau+1}}{c_{\tau+1}}+\frac{(B-B_1)}{\frac{c_{[\tau+1]}}{\hat{q}^+_{[\tau+1]}}\sum_{i\in N'_\tau}\hat{q}^+_{[i]}}\left(\frac{\frac{q_{\tau+1}}{c_{\tau+1}}}{\frac{\hat{q}^+_{[\tau+1]}}{c_{[\tau+1]}}}\sum_{i\in N'_\tau}\hat{q}^+_{[i]}-\sum_{i\in N'_\tau}q_{[i]}\right)\right)$$

$$+\frac{q_{\tau+1}}{c_{\tau+1}}$$

$$\leq(1-\frac{1}{B})\left(\frac{B_1q_{\tau+1}}{c_{\tau+1}}+\frac{(B-B_1)}{\frac{c_{[\tau+1]}}{\hat{q}^+_{[\tau+1]}}\sum_{i\in N'_\tau}\hat{q}^+_{[i]}}\left(\sum_{i\in N'_\tau}\hat{q}^+_{[i]}-\sum_{i\in N'_\tau}q_{[i]}\right)\right)$$

$$+\frac{q_{\tau+1}}{c_{\tau+1}} \qquad \text{(By Lemma 4)}$$

$$=(1-\frac{1}{B})\left(\frac{B_1q_{\tau+1}}{c_{\tau+1}}+\frac{(B-B_1)2\tau}{\frac{c_{[\tau+1]}}{\hat{q}^+_{[\tau+1]}}\sum_{i\in N'_\tau}\hat{q}^+_{[i]}}\sqrt{\frac{K\ \bar{c}\ ln(KB)}{2B_1}}\right)+\frac{q_{\tau+1}}{c_{\tau+1}}$$

$$\leq\left(\frac{B_1q_{\tau+1}}{c_{\tau+1}}+\frac{(B-B_1)2\tau}{\frac{c_{[\tau+1]}}{\hat{q}^+_{[\tau+1]}}\sum_{i\in N'_\tau}\hat{q}^+_{[i]}}\left(\sqrt{\frac{K\ \bar{c}\ ln(KB)}{2B_1}}\right)\right)+\frac{q_{\tau+1}}{c_{\tau+1}}$$

$$\leq\left(B_1\frac{q_{\tau+1}}{c_{\tau+1}}+\frac{(B-B_1)2\tau}{\sum_{i\in N'_\tau}c_{[i]}}\left(\sqrt{\frac{K\ \bar{c}\ ln(KB)}{2B_1}}\right)\right)+\frac{q_{\tau+1}}{c_{\tau+1}}$$

$$\leq\left(B_1\frac{1}{\underline{c}}+\frac{2B}{\underline{c}}\left(\sqrt{\frac{K\ \bar{c}\ ln(KB)}{2B_1}}\right)\right)+\frac{q_{\tau+1}}{c_{\tau+1}}$$

It can be shown that the above expression attains minimum value when the value of B_1 is

$$B_1=\frac{1}{(2)^{1/3}}\ (\bar{c}K\ ln(KB))^{1/3}\ B^{2/3} \qquad (10)$$

Substituting the value of B_1, we get the upper bound of expected regret as

$$\mathbb{E}[R]\leq\frac{1.587}{\underline{c}}\ (\bar{c}K\ ln(KB))^{1/3}\ B^{2/3}+\frac{q_{\tau+1}}{c_{\tau+1}}$$

Thus it is proved that the expected regret is

$$O(B^{2/3}\ K^{1/3}\ (ln(KB))^{1/3})$$

■

Note that the upper bound on regret does not depend on τ directly, though the presence of $\frac{q_{\tau+1}}{c_{\tau+1}}$ in the expression of upper bound on regret makes the regret bound decrease as τ increases. Also, our regret bounds matches with the existing literature [18]. We now show that this is the best achievable regret up to a constant factor.

4.5 Lower Bound Analysis

In this section, we prove that the regret bound for the proposed algorithm is tight by providing lower bound analysis. Lower bound on regret exists for MAB mechanism in the context of sponsored search auction without any budget constraint [5]. However, these results cannot be extended in a straight forward way for budgeted MAB mechanism. The presence of budget causes an inherent dependency between payment and the number of rounds. Given a fixed budget, the number of rounds of the algorithm may vary as the bid changes or observed success rate changes. This makes the lower bound proof in the setting more challenging. Let $T_c(B)$ denotes the total number of tasks that are executed with cost profile c and budget B. Note that the total number of tasks will also depend on the success realization and more importantly on allocation and payment rule. However, for notational simplicity we do not show this dependence explicitly. We prove the lower bounds for the special case of $\tau = 0$, i.e., the task is verified instantly and thus all the workers are available for all the tasks. Since, the problem with task deadline $\tau = 0$ is a special case, the lower bounds are applicable to the general class of problems. We start with some definitions that will be used in proofs and Theorem 6 provides the final lower bound.

Definition 4 ***(Similar success realization)*** *Two success realizations s_1 corresponding to bid profile c and s_2 corresponding to bid profile $\hat{c}$ are said to be similar if $s_1^t = s_2^t\ \forall t \leq \min(T_c(B), T_{\hat{c}}(B))$. Similarity between two success realizations s_1 and s_2 is denoted by $s_1 \sim s_2$.*

We now redefine monotonicity of allocation rule for learning setting:

Definition 5 ***(Monotone)*** *An allocation $\mathcal{A}$ is called **monotone** if for any $\hat{c}_i \leq c_i$, $\mathcal{A}_i^t(c_i, c_{-i}; s_1) = 1$, then $\mathcal{A}_i^t(\hat{c}_i, c_{-i}; s_2) = 1$, $\forall s_1 \sim s_2$ and $\forall t \leq \min(T_{(c_i,c_{-i})}(B), T_{(\hat{c}_i,c_{-i})}(B))$.*

To provide lower bounds, we will assume that the mechanism knows full success realization and hence allocation rule at any time t depends on complete success realization s instead of s^t. An allocation rule at time t may or may not depend on the bid. We denote such rounds as bid independent rounds.

Definition 6 ***(Bid Independent Round)*** *A round t is called bid independent if the allocation $\mathcal{A}^t(\hat{c})$ remains same for all $\hat{c} \in [\underline{c}, \bar{c}]^K$.*

It is easy to see from the critical payment that if a round t is bid independent, then the payment to the allocated worker at that round is $\bar{c}$. If the requester chooses to make allocations of all rounds bid independent, then only $\frac{B}{\bar{c}}$ rounds will be played. However, such allocation rule severely reduces the expected number of successfully completed jobs. Ideally, there should be no bid-independent allocation rounds. On the other hand, we show that bid independent rounds are necessary for incentive compatibility. Hence, the total number of bid independent rounds should be properly balanced.

We say a round t is used for exploration, when the success observed in that round is used for deciding future allocations. We denote such a round t as *influential round*, and the round t' whose allocation depends on the success observed at round t is called *influenced round*. Such a pair (t, t') is called *influential pair.*

Definition 7 ***(Influential Round)*** *A round t is called an influential round if the success observed in round t, is used for deciding future allocation in round t' and round t' is called influenced round.*

Definition 8 ***(Influential Pair)*** *A pair (t, t') is called an influential pair if $\mathcal{A}^{t'}(c; s) \neq \mathcal{A}^{t'}(c; s')$ for $t' > t$, where $\mathcal{A}^t(c; s) = j$ and $s' = s \oplus \mathbb{I}\{j, t\}$ for some bid profile c, and some success realization s.*

Here, $s' = s \oplus \mathbb{I}\{j, t\}$ denotes success realization with only the success bit for worker j at time t is flipped in s.

From Theorem 1, any truthful and individually rational mechanism should satisfy the property of monotonicity. Number of influential rounds indicates how many rounds are required to learn the qualities. We now show that any influential pair (t, t') should be bid independent for truthful mechanism and hence, payment in these influential rounds should be $\bar{c}$.

Theorem 5 *If there exists an influential pair (t, t') for some bid profile c, and success realization s, where $t < t' \leq \frac{B}{\bar{c}}$ then a deterministic mechanism is truthful only if the payment at time t is bid independent.*

Proof: Let us choose $K = 2$ with workers 1 and 2. Let (t, t') be an influential pair for some bid profile $\hat{c}$ and some success realization s such that $\mathcal{A}^t(\hat{c}; s) = 2$ and $\mathcal{A}^{t'}(\hat{c}; s) \neq \mathcal{A}^{t'}(\hat{c}; s')$, where, $s' = s \oplus \mathbb{I}\{2, t\}$. We assume t' is the least time step which is influenced by time step t. Any algorithm plays for at least $\frac{B}{\bar{c}}$ rounds since the payment to any worker for any task does not exceed $\bar{c}$. Let us assume that round t is bid dependent, hence for some $\hat{c}_1^- < \hat{c}_1$, $\mathcal{A}^t(\hat{c}_1^-, \hat{c}_2; s_1) = 1$ where $s_1 \sim s$. Since $s_1^x = s^x \ \forall x \leq \frac{B}{\bar{c}}$, we replace s_1 with s in rest of the proof (as $t, t' \leq \frac{B}{\bar{c}}$). As the success of arm 2 at round t is not observed by the algorithm, so the allocation and payment at t' remains same for s and s', and thus, $\mathcal{A}^{t'}(\hat{c}_1^-, \hat{c}_2; s) = \mathcal{A}^{t'}(\hat{c}_1^-, \hat{c}_2; s') = i$ and $\mathcal{P}_i^{t'}(\hat{c}_1^-, \hat{c}_2; s) = \mathcal{P}_i^{t'}(\hat{c}_1^-, \hat{c}_2; s')$.

Let us assume that $\mathcal{A}^{t'}(\hat{c}; s) = 1$ and $\mathcal{A}^{t'}(\hat{c}; s') = 2$, otherwise we can swap s with s'. Due to monotonicity, $\mathcal{A}^{t'}(\hat{c}; s) = 1 \Rightarrow \mathcal{A}^{t'}(\hat{c}_1^-, \hat{c}_2; s) = 1$. Thus, $\mathcal{A}^{t'}(\hat{c}_1^-, \hat{c}_2; s) = \mathcal{A}^{t'}(\hat{c}_1^-, \hat{c}_2; s') = 1$. Now, $\mathcal{A}^{t'}(\hat{c}; s) = 1$ and he is paid an amount up to which he could have increased his bid and still get the allocation, thus the payment $\mathcal{P}_1^{t'}(\hat{c}; s) \geq \hat{c}_1$.

(1) Since $\mathcal{A}^{t'}(\hat{c}_1^-, \hat{c}_2; s) = 1$, he is paid an amount up to which he could increase his bid and still get the allocation. Now, as we increase the cost from $\hat{c}_1^-$ to $\hat{c}_1$, arm 1 still gets the allocation, so the payment $\mathcal{P}_1^{t'}(\hat{c}_1^-, \hat{c}_2; s) \geq \hat{c}_1$.

(2) Again, $\mathcal{A}^{t'}(\hat{c}; s') = 2$ and $\mathcal{A}^{t'}(\hat{c}_1^-, \hat{c}_2; s') = 1$, so the amount up to which he can raise his bid and still get the allocation is less than or equal to $\hat{c}_1$, as he did not get an allocation at round t with bid $\hat{c}_1$, so $\mathcal{P}_1^{t'}(\hat{c}_1^-, \hat{c}_2; s) \leq \hat{c}_1$.

By arguments (1) and (2), it can be seen that the payment $\mathcal{P}_1^{t'}(\hat{c}_1^-, \hat{c}_2; s) = \mathcal{P}_1^{t'}(\hat{c}_1^-, \hat{c}_2; s') = \hat{c}_1$. If payment to a worker is equal to its bid then the mechanism can not be truthful. Thus, the payment at time t has to be bid independent. ■

According to Theorem 5, the allocation at an influential or exploration round t does not depend on the bids, and thus the payment at any influential round t is $\bar{c}$.

Theorem 6 *Any deterministic truthful mechanism for budgeted MAB setting with strategic costs has an expected regret of $\Omega(B^{\frac{2}{3}} \ K^{\frac{1}{3}})$.*

Proof: To prove lower bounds, we assume that there is an adversary that provides the qualities and costs vector as input to the algorithm that tries to maximize the regret. Since the upper bound on the payment to any worker is given by $\bar{c}$, any mechanism in this setting run for at least $\frac{B}{\bar{c}}$ rounds. If there are no influential rounds that means allocation to any worker does not depend on his quality (since the quality is being learnt). In this case the best any mechanism can do is to allocate equal number of tasks to all the workers. Otherwise the adversary can chose the quality of the worker that has been assigned lesser number of tasks to be of that of 1 and rest of quality 0. Thus, incurring the regret of at least $\frac{B(K-1)}{\bar{c}K}$ if quality profile is $q = (1, 0, \ldots, 0)$.

Now, let us assume that there are some influential pairs in the first $\frac{B}{\bar{c}}$ rounds, and let the corresponding influential rounds be denoted by set $\mathcal{I}$. By Theorem 5, these influential rounds should be bid independent. Now consider two cases:

1. If $|\mathcal{I}| \geq \beta B^{2/3} K^{1/3}$, where β is any constant. These rounds are bid independent, by (Theorem 5) payment made in these rounds is $\bar{c}$. Now, consider the quality vector, $q = (1, 1, \ldots, 1)$ and cost vector, $c = (\underline{c}, \underline{c}, \ldots, \underline{c})$. Regret for this profile is at least $\frac{\beta B^{2/3} K^{1/3}}{\underline{c}} - \frac{\beta B^{2/3} K^{1/3}}{\bar{c}}$. As $\bar{c} > \underline{c}$, we get the regret of $\Omega(B^{2/3} \ K^{1/3})$.

2. If $|\mathcal{I}| \leq \beta B^{2/3} K^{1/3}$, then using the result in [5], one can prove that there exists success realizations which achieves regret of at least $\beta B^{2/3} K^{1/3}$.

■

Thus, upper bound regret is tight up to a logarithmic factor.

5. CONCLUSION AND FUTURE WORK

We studied a budgeted MAB problem with strategic arms and task deadlines. We proposed a budget feasible, truthful, and individually rational mechanism to solve the problem. We provided an upper bound of $O(B^{2/3} K^{1/3} \ln(KB)^{1/3})$ on the regret of the proposed algorithm with respect to an appropriate benchmark. We also showed that any deterministic truthful algorithm that solved *budgeted MAB mechanism design* problem would suffer an expected regret of $\Omega(B^{2/3} K^{1/3})$. The questions left open by this work concern results for the following problems: (a) A randomized truthful mechanism for the problem with better regret bound. (b) A truthful mechanism for the problem with tasks having different deadlines. (c) A truthful mechanism for minimizing total payment where the total reward obtained is higher than a fixed threshold.

REFERENCES

[1] R. Agrawal. Sample mean based index policies with $O(logn)$ regret for the multi-armed bandit problem. *Advances in Applied Probability*, pages 1054–1078, 1995.

[2] S. Agrawal and N. R. Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM Conference on Economics and Computation (EC'14)*, pages 989–1006. ACM, 2014.

[3] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science, 2001*, pages 482–491. IEEE, 2001.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[5] M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, pages 79–88. ACM, 2009.

[6] A. Badanidiyuru, R. Kleinberg, and Y. Singer. Learning on a budget: posted price mechanisms for online procurement. In *Proceedings of the Thirteenth ACM Conference on Electronic Commerce (EC'12)*, pages 128–145. ACM, 2012.

[7] 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.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2 edition, 2001.

[9] N. R. Devanur and S. M. Kakade. The price of truthfulness for pay-per-click auctions. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, pages 99–106. ACM, 2009.

[10] S. Guha and K. Munagala. Approximation algorithms for budgeted learning problems. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pages 104–113. ACM, 2007.

[11] C.-J. Ho, A. Slivkins, and J. W. Vaughan. Adaptive contract design for crowdsourcing markets: bandit algorithms for repeated principal-agent problems. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 359–376. ACM, 2014.

[12] R. Kleinberg, A. Niculescu-Mizil, and Y. Sharma. Regret bounds for sleeping experts and bandits. *Machine Learning*, 80(2-3):245–272, 2010.

[13] R. B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[14] Y. Singer. Budget feasible mechanisms. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 765–774. IEEE, 2010.

[15] Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1157–1166, 2013.

[16] A. Slivkins and J. W. Vaughan. Online decision making in crowdsourcing markets: Theoretical challenges (position paper). *arXiv preprint arXiv:1308.1746*, 2013.

[17] L. Tran-Thanh, A. C. Chapman, A. Rogers, and N. R. Jennings. Knapsack based optimal policies for budget-limited multi-armed bandits. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1134–1140, 2012.

[18] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *European Conference on Artificial Intelligence*, pages 768–773, 2012.