A Deterministic MAB Mechanism for Crowdsourcing with Logarithmic Regret and Immediate Payments

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

Ganesh Ghalme
Indian Institute of Science
Bangalore, India
ganeshghalme@gmail.com

Satyanath Bhat
Indian Institute of Science
Bangalore, India
satya.bhat@gmail.com

Sujit Gujar*
Indian Institute of Science
Bangalore, India
sujit.gujar@gmail.com

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

ABSTRACT

We consider a general crowdsourcing setting with strategic workers whose qualities are unknown and design a multi-armed bandit (MAB) mechanism, CrowdUCB, which is deterministic, regret minimizing, and offers immediate payments to the workers. The problem involves sequentially selecting workers to process tasks in order to maximize the social welfare while learning the qualities of the strategic workers (strategic about their costs). Existing MAB mechanisms are either: (a) deterministic which potentially cause significant loss in social welfare, or (b) randomized which typically lead to high variance in payments. CrowdUCB completely addresses the above problems with the following features: (i) offers deterministic payments, (ii) achieves logarithmic regret in social welfare, (iii) renders allocations more effective by allocating blocks of tasks to a worker instead of a single task, and (iv) offers payment to a worker immediately upon completion of an assigned block of tasks. CrowdUCB is a mechanism with learning that learns the qualities of the workers while eliciting their true costs, irrespective of whether or not the workers know their own qualities. We show that CrowdUCB is ex-post individually rational (EPIR) and ex-post incentive compatible (EPIC) when the workers do not know their own qualities and when they update their beliefs in sync with the requester. When the workers know their own qualities, CrowdUCB is EPIR and $\varepsilon-$EPIC where ε is sub-linear in terms of the number of tasks.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Intelligent agents; I.2.6 [**Learning**]: Parameter learning

General Terms

Algorithms, Economics

Keywords

Crowdsourcing, Multi-armed bandit, Mechanism design

*This work was done when the author was a post doctoral fellow at LIA, EPFL, Lausanne, Switzerland.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

1. INTRODUCTION

Crowdsourcing is emerging as a powerful sourcing alternative to organizations for the execution of tasks in a timely, scalable, and cost-effective manner. The availability of a large worker pool enables crowdsourcing to be used widely to collect labels for large scale learning problems, like collecting a ground dataset for supervised learning algorithms. The rise in the number of crowdsourcing platforms in recent times indicates the usefulness of crowdsourcing in various real world applications.

On a crowdsourcing platform, a requester (one who is seeking the tasks to be performed) naturally seeks high quality and low cost workers. We focus our attention on crowdsourcing platforms where the number of tasks is large and individual tasks are simple, common, low cost tasks such as image labeling. Due to diverse, heterogeneous demographics of crowd workers and a rich variety of tasks available on a crowdsourcing platform, the quality of a worker is typically unknown to the requester and sometimes even to the worker as well. We allow workers to bid their costs for a particular task (for example, rent-a-coder.com). When there are low cost tasks at hand and auction overheads are involved, it is preferable to allocate a block of tasks to a worker rather than allocate a single task. The problem of choosing an optimal worker becomes a challenging one when these costs need to be elicited and qualities need to be learned via a suitable learning algorithm. This problem can be posed as a *multi-armed bandit* (MAB) mechanism design problem [5, 10], where, while eliciting the true costs, the requester needs to either explore the workers (modeled as arms) to learn their qualities or exploit the best worker identified so far.

There are two types of MAB mechanisms available in the literature: (a) deterministic mechanisms [5, 10, 7] which are known to incur high regret in social welfare (for example, as high as, $O(T^{2/3})$ where T is the number of tasks) but with zero variance in payments, and (b) randomized mechanisms [4, 6] where optimal (logarithmic) regret is achieved but with a high variance in payments. In this work, we propose a mechanism that (a) offers deterministic payments, (b) achieves minimal regret (logarithmic in the number of tasks), (c) allows allocation of blocks of tasks to workers, and (d) offers immediate payments. Moreover, the proposed mechanism allows workers the flexibility of modifying their bids in successive rounds. Following are the two key features that differentiate our work from extant work in the literature.

Deterministic MAB Mechanism with Logarithmic Regret

The lower bound characterization results for deterministic MAB mechanisms with dominant strategy incentive compatibility (DSIC) notion assumes that the manipulating worker has the power to know all the future events, leading to high regret [5, 10]. While such an approach is theoretically appealing, in practice, however, a worker is not sure about his own quality and thus cannot manipulate based on the knowledge of future events. Recent works [11, 13] argue against the approach of providing worst case guarantees for large games and recommend instead, simpler yet optimal strategies with pragmatic assumptions. For crowdsourcing, this approach is more suitable due to the large population of global workforce and the diversity in the nature of tasks posted by requesters.

To achieve logarithmic regret, we use the notion of *ex-post incentive compatibility* (EPIC). EPIC ensures that even after considering the future events, for every task, it is best for any worker to bid his cost truthfully when the other workers bid truthfully. Though EPIC is a weaker notion than DSIC, it is much stronger than Bayesian incentive compatibility that is often used in mechanism design [14]. Further, we make no assumptions about workers' knowledge of success realization (which future tasks will be successfully completed) and about the number of tasks requester wants to execute.

Block Allocations, Dynamic Bids, and Immediate Payments

In the context of learning of qualities, the algorithms in the literature usually allocate a single task at a time and make the next allocation following evaluation of the performance of the previous tasks. With low cost tasks, this could cause significant overheads and the workers may quit the crowdsourcing platform. Our mechanism has the enhanced feature of allocating blocks of tasks to the workers. The block sizes are dynamically updated based on the learned qualities and the bids of the workers. Our mechanism further allows the workers to bid different costs in different rounds as the time progresses and offers immediate payments to the workers.

Contributions

We propose two variants of our mechanism: (a) qualities are unknown to the workers and (b) qualities are private information of the workers. Following are the specific contributions:

- We design a MAB mechanism, *CrowdUCB* which is deterministic, regret optimal, makes block allocations, and offers immediate payments (Section 4).
- We theoretically prove that CrowdUCB suffers minimal regret that matches the lower bound in the space of stochastic MAB problems (Theorem 3).
- When workers are unaware of their qualities and learn these qualities as the mechanism progresses, we prove that CrowdUCB is ex-post individually rational (EPIR) and ex-post incentive compatible (EPIC)(Section 5).
- When workers know their qualities, we prove that CrowdUCB is $\varepsilon-$EPIC where ε is sub-linear in the number of tasks T, making CrowdUCB asymptotically EPIC in this setting (Section 6).
- We experimentally show that: (a) existing randomized mechanisms suffer from high variance as opposed to CrowdUCB and (b) the constants hidden within the regret are reasonable enough (Section 7).

2. RELATED WORK

While the MAB problem is a well studied area in machine learning community, designing MAB mechanisms is still a relatively unexplored area. Though there are optimal regret guarantees in MAB problems [8], strategic versions of the problem fail to achieve such guarantees [5, 10]. The lower bound proof in [5, 10] for any deterministic MAB mechanism makes a crucial assumption that a worker is fully aware of the complete success realization, thus giving more manipulative power to the worker. Our mechanism is more practical where we relax the above assumption and achieve logarithmic regret. In the literature, the high regret issue is alleviated via a randomized mechanism [3, 21] and it turns out that standard MAB algorithms like UCB1 [2] can be transformed into a truthful randomized mechanism. However, the payment scheme in such mechanisms involves a trade-off between the revenue obtained and the variance in the payment. It is then shown that this trade-off is necessary to achieve truthfulness [21]. In our work, we provide a mechanism that circumvents this trade-off in the specific context of crowdsourcing via realistic assumptions.

The concept of MAB based algorithms and MAB mechanisms is not new to crowdsourcing [18]. When the costs are fixed and known and qualities are unknown, the crowdsourcing setting can be formulated as a variation of a MAB problem with crowd workers acting as the arms [19, 20, 1, 12]. In another setting, When the costs of the workers are not known, but the qualities are homogeneous among workers, the problem of designing a posted price mechanism is formulated using a MAB algorithm with various prices as the arms of the bandit [3, 17]. When workers arrive online, designing a truthful mechanism for eliciting the bids from the workers is considered in [16] but the work assumes that the workers are homogeneous in terms of their qualities. With an offline worker pool, designing truthful MAB mechanisms in crowdsourcing setting for eliciting true costs and learning qualities at the same time is also considered by many authors. Authors in [7] considered an exploration separated mechanism to the budgeted crowdsourcing setting that is deterministic but incurs a high regret ($O(T^{2/3}$) with T being the number of tasks). On the other hand, [6] proposes a UCB1 [2] based algorithm for the allocation and provides a randomized mechanism similar to [4] to compute the payments which suffer high variation in payments. In this work, we design a deterministic MAB mechanism that achieves optimal regret similar to that of UCB1 with zero variance.

3. THE MODEL

There is a requester who seeks to allocate T homogeneous tasks among K crowd workers. Worker i ($i = 1, \ldots, K$) has a fixed quality $q_i \in [0, 1]$ which is the probability with which worker i executes the task successfully. Due to heterogeneity among the tasks available at crowdsourcing platforms, the quality of a worker on a particular type of task is unknown to the requester and sometimes to the worker as well. A worker i incurs a cost $c_i \in [c_{\min}, c_{\max}]$ for completing the task and c_i is a private information of worker i. We assume that the qualities and costs do not change over time for homogeneous tasks that a requester seeks to get executed.

K	Number of crowd workers
T	Number of homogeneous tasks
$t \in \{1, \ldots, T\}$	Running index of tasks
M	Reward to the requester for every successfully completed task
$q_i \in [0, 1]$	Quality of worker i
$c_i \in [c_{\min}, c_{\max}]$	Cost per task of worker i
c_{-i}	Cost vector of all workers other than i
$b_{i,t}$	Cost bid by worker i for task t
$b_{-i,t}$	Cost bid by workers other than i for task t
b_t	Cost bid vector of all the workers for task t
I_t	Worker allocated for task t
$\tau_t + 1$	Number of tasks allocated as a block to a worker from tasks t to $t + \tau_t$
$N_{i,t}$	Number of tasks allocated to worker i till tasks t
$S_{i,t}$	Number of tasks successfully completed by worker i till tasks t
h^t	History till tasks t of allocations and observed success i.e. $\{I_1, S_{I_1,1}, \ldots, I_{t-1}, S_{I_{t-1},t-1}\}$
$p_{i,t}(b_t; h^t)$	Payment to worker i for task t with history h^t and cost bid vector b_t
$u_{i,t}(b_{i,t}, b_{-i,t}; h^t; c_i)$	Utility to worker i for task t with cost bid vector $(b_{i,t}, b_{-i,t})$, history h^t, and true cost c_i

Table 1: Notation Table

For each successfully completed task, the requester derives a reward of M, hence the expected reward of the requester is given by Mq_i, when the worker i is allocated a task. The requester would like to allocate the tasks so as to maximize the total expected reward. Since the qualities are unknown, in order to learn the qualities, the requester can allocate the tasks sequentially and can observe the qualities of the workers over the allocated tasks. Based on the learnt qualities over the previous tasks, the requester can compute allocations and payments for the future.

As the costs per task are private to the workers, we propose to use an auction for allocating tasks. Though the private cost of a worker does not change over time, his beliefs about his quality get updated. Therefore, a worker can update his bid to a different value for a task if he thinks it would give him better utility. Note that, inviting bids for every task t gives more flexibility to the workers. Let c_{-i} denote the private cost vector of all the workers other than i. Let $b_{i,t}$ be the bid by a worker i for a task t and $b_{-i,t}$ is the bid vector by the workers other than i. On completion of the task, the requester pays to the worker i, an amount $p_{i,t}$, if i is selected for the task t. The utility of a worker i for a task t with bid vector b_t and true valuation c_i is given as $u_{i,t}(b_t; h^t; c_i) = \mathbb{1}\{I_t(b_t; h^t) = i\}(p_{i,t}(b_t; h^t) - c_i)$. Here, I_t represents the index of the worker allocated the task t and h^t represents the history of all the allocations and the corresponding performances of the allocated workers in the past. Let $S_{i,t}$ denotes the reward of the worker i if he is allocated task t, then the history at task t is given as $h^t = \{(I_1, S_{I_1,1}), \ldots, (I_{t-1}, S_{I_{t-1},t-1})\}$ and depends on the inherent quality of the workers.

A requester would like to update his beliefs about the qualities of the workers after completion of every task and select a worker for the next task based on updated beliefs. However, single task allocations followed by belief updates and invitation for bids involve considerable time overheads, moreover, the workers prefer receiving multiple tasks at a time. Our mechanism allows the requester to allocate a block of tasks without losing on optimality in learning. Let $0 \leq \tau_t \leq T - t$ denote the additional number of tasks allocated as a block starting from task t to $t + \tau_t$ (thus giving $\tau_t + 1$ tasks to the worker). If worker I_t is allocated a block of tasks from t to $t+\tau_t$, bids are not sought for next τ_t tasks. Thus, $b_{i,t'} = b_{i,t}$ $\forall i$, $\forall t' \in \{t, \ldots, t + \tau_t\}$. After $\tau_t + 1$ tasks, empirical quality of the worker i is updated. Notations are provided in Table 1. We make the following key deviations from existing settings:

- Most of the literature in this space [5, 4] assumes that a strategic worker is fully aware of the future successes. This assumption leads to either higher regret or variance in the payments. In this work, a strategic worker is not cognizant of the future and has knowledge of only past allocations. However, the workers are aware of how a requester is learning and can manipulate their bids to maximize their utility with this information.

- Due to anonymity in crowdsourcing platforms, a worker does not have the knowledge of total number of tasks T available. Hence, he would not like to lose the allocations in the current rounds if the allocations result in positive utility.

We now define some properties that CrowdUCB satisfies in the settings considered.

Definition 1 ***Ex-Post Individual Rationality (EPIR):*** *A mechanism is said to be ex-post individually rational if each worker receives a non-negative utility for each task, irrespective of the history prior to this task and irrespective of the bids of other workers. That is,* $\forall i$, $\forall t$, $\forall h^t$, $\forall b_{-i,t}$,

$$u_{i,t}(c_i, b_{-i,t}; h^t; c_i) \geq 0, \ \forall c_i.$$

Definition 2 ***Ex-Post Incentive Compatibility (EPIC):*** *A mechanism is EPIC if for any worker i, bidding truthfully in all the rounds results in higher total utility as compared to non-truthful bidding in any round when the other workers are assumed to bid truthfully, i.e.* $\forall i$, $\forall c_{-i}$, $\forall c_i$, $\forall h^t$, $\forall b_{i,t}$,

$$\sum_{t=1}^{T} u_{i,t}(c_i, c_{-i}; h^t; c_i) \geq \sum_{t=1}^{T} u_{i,t}(b_{i,t}, c_{-i}; h^t; c_i).$$

Note that EPIC is a weaker notion than dominant strategy incentive compatibility where the other workers can bid different values in different rounds but it is stronger than Bayesian Nash incentive compatibility where a worker is truthful with respect to the expectation of other workers' valuations.

Let $u_{i,t}(b_{i,t}, c_{-i}; h^t; c_i; q_i)$ be the utility of worker i with bid $b_{i,t}$ for task t when he knows his true quality q_i. We will show that when workers' qualities are private knowledge then CrowdUCB is $\varepsilon-$EPIC which we define as follows:

Definition 3 ***$\varepsilon-$EPIC***:
A mechanism is said to be $\varepsilon-$EPIC if by misreporting bids in any round(s), no worker can gain more than ε in his total utility that he would have obtained by bidding truthfully in all the rounds when the other workers are assumed to bid truthfully. i.e. $\forall i,\ \forall c_{-i},\ \forall c_i,\ \forall h^t$ and $\forall b_{i,t}$,

$$\mathbb{E}\left[\sum_{t=1}^{T} u_{i,t}(c_i, c_{-i}; h^t; c_i; q_i)\right] + \varepsilon \geq \mathbb{E}\left[\sum_{t=1}^{T} u_{i,t}(b_{i,t}, c_{-i}; h^t; c_i; q_i)\right].$$

Here, expectation is taken over success realizations given true qualities.

Definition 4 ***Asymptotically EPIC***:
We say that a mechanism is asymptotically EPIC when per round utility gain over truthful bidding for every worker i goes to zero with number of rounds to infinity when other workers report truthfully, i.e. $\forall i,\ \forall c_{-i},\ \forall c_i,\ \forall h^t,\ \forall b_{i,t}$.

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \left(u_{i,t}(b_{i,t}, c_{-i}; h^t; c_i) - u_{i,t}(c_i, c_{-i}; h^t; c_i)\right)\right] = 0.$$

Definition 5 ***Regret in Social Welfare:*** *Social welfare is given by the sum of the valuation of all the workers and the requester. The valuation of the requester and a worker i is given by Mq_i and $-c_i$ respectively, if the worker i is given the task t. The regret in social welfare of an algorithm A is defined as the difference in the social welfare achieved by an optimal algorithm that knows the true qualities and the social welfare by the algorithm A that is learning the qualities. When the workers bid truthfully,*

$$R(A) = T(Mq_{i^*} - c_{i^*}) - \sum_{t=1}^{T}\sum_{i=1}^{K} \mathbb{1}\{I_t = i\}(Mq_i - c_i),$$

where, $i^* \in \arg\max_i (Mq_i - c_i)$.

Here, $\mathbb{1}\{I_t = i\}$, is an indicator variable which is 1 if $I_t = i$ and 0 otherwise. For the rest of the paper, whenever we talk about regret, we will be referring to regret in social welfare. Social welfare is helpful for many socially desirable outcomes, for example, a government procuring work from crowd for creating job opportunities or government seeking help from the volunteers through crowdsourcing of disaster relief. Moreover, for revenue maximization, it is shown that the revenue of social welfare maximizing mechanism approaches to that of revenue maximizing mechanism with large number of bidders [9]. Further, mechanisms with social welfare maximization are simple and more intuitive. We say that an algorithm A has *sub-linear* regret if the regret of the algorithm is sub-linear in the number of tasks i.e. $R(A) = o(T)$. We now propose our mechanism CrowdUCB that has sub-linear regret of $O(\ln T)$. CrowdUCB is EPIR and EPIC when workers are unaware of their qualities. Further, when workers know their qualities but the mechanism still learns the qualities of the workers, CrowdUCB is EPIR and $\varepsilon-$EPIC where ε is sub-linear in the number of tasks T hence is asymptotically EPIC.

4. PROPOSED MECHANISM: CROWDUCB

A deterministic mechanism, $\mathcal{M}_t(b_t; h^t) = \langle I_t, \tau_t, p_{I_t}\rangle$, at any time t, allocates $\tau_t + 1$ number of tasks to a worker I_t, and pays an amount $p_{I_t} = \sum_{s=t}^{s=t+\tau_t} p_{I_t,s}$. The allocated worker I_t, the number of tasks τ_t+1 and the payment p_{I_t} depend on the bid vector b_t and history h^t obtained so far. In the rest of the paper, we will not show this dependence explicitly if it is clear from the context. The next auction happens after τ_t+1 tasks are completed and thus, $b_{i,s} = b_{i,t},\ \forall i,\ \forall s,$ s.t. $t \leq s \leq \tau_t + t$. A new bid vector $b_{t+\tau_t+1}$ is obtained from the workers for task $t + \tau_t + 1$. *CrowdUCB* uses some learning bounds on qualites to make allocations and payments for the current task. These bounds are shown in Table 2.

$\hat{q}_{i,t} = \frac{S_{i,t}}{N_{i,t}}$	Estimate of quality q_i till tasks t
$\hat{q}^{+}_{i,t} = \hat{q}_{i,t} + \sqrt{\frac{2\ln(t)}{N_{i,t}}}$	Upper confidence bound (UCB) on the quality of the worker i till tasks t.
$\hat{q}^{+\bar{\rightarrow}\tau}_{i,t} = \frac{S_{i,t}}{N_{i,t}+\tau} + \sqrt{\frac{2\ln(t+\tau)}{N_{i,t}+\tau}}$	UCB on the quality of worker i if he fails on all the tasks from $t, t+1, \ldots, t+\tau$
$\hat{q}^{+\rightarrow\tau}_{i,t} = \frac{S_{i,t}}{N_{i,t}} + \sqrt{\frac{2\ln(t+\tau)}{N_{i,t}}}$	UCB on the quality of worker i when he does not get any task from t to $t+\tau$
$\hat{q}^{+\tilde{\rightarrow}\tau}_{i,t} = \frac{S_{i,t}}{N_{i,t}} + \sqrt{\frac{2\ln(t+\tau)}{N_{i,t}+\tau}}$	UCB on the quality of worker i when he gets tasks from t to $t+\tau$ and he performs the tasks with same empirical estimate as at time t

Table 2: Learning Bounds on Qualities

4.1 Allocation Rule

For a task t, the allocation rule selects a worker I_t and the number of additional tasks, τ_t, to be allocated to him. The allocation rule is designed keeping in view of the following two important yet conflicting objectives.

Learning: Let, $\hat{q}^{+}_{i,t} = \hat{q}_{i,t} + \sqrt{\frac{2\ln(t)}{N_{i,t}}}$ be the upper confidence bound (UCB) on the quality of the worker i. Here, $N_{i,t}$ is the number of tasks allocated to the worker i till task t and $\hat{q}_{i,t}$ is the empirical estimate of q_i. A worker I_t for the task t is selected as:

$$I_t \in \arg\max_i (M\hat{q}^{+}_{i,t} - b_{i,t}), \tag{1}$$

We refer, $M\hat{q}^{+}_{i,t} - b_{i,t}$ as the UCB index of the worker i. Note that for the fixed values of M and c_i, UCB index on $Mq_i - c_i$ can be computed by upper bounding q_i which is the only stochastic parameter. Hence, UCB index is defined accordingly. Thus, the allocation rule selects a worker I_t having the highest UCB index. The above strategy of optimism in the face of uncertainty is crucial for learning.

Block allocations: As already stated, though it is beneficial for the requester to allocate one task at a time, in this work, we propose the block allocation of tasks to reduce auction overheads. We propose a block size of $\tau_t + 1$ while allocating the task t. Here, $\tau_t + 1$ is the smallest block of tasks such that the worker I_t remains optimal even when he fails on every task in this block. Let $S_{i,t}$ denote the number of successfully completed tasks till task t (This implies, $\hat{q}_{i,t} = \frac{S_{i,t}}{N_{i,t}}$). Let $\hat{q}^{+\bar{\rightarrow}\tau}_{i,t} = \frac{S_{i,t}}{N_{i,t}+\tau} + \sqrt{\frac{2\ln(t+\tau)}{N_{i,t}+\tau}}$, denote UCB

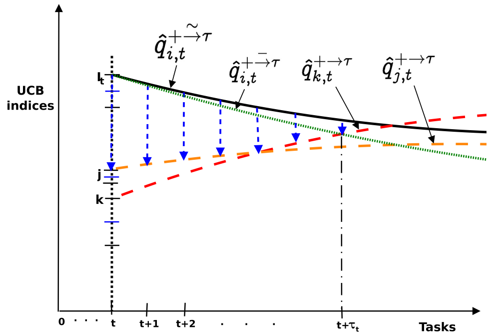

Figure 1: Allocation block given to the worker I_t from tasks t to $t+\tau_t$. The dotted arrows represent the utility to worker I_t from tasks t to $t+\tau_t$.

on quality of a worker i if he fails on all the tasks from $t, t+1, \ldots, t+\tau$. Similarly, let $\hat{q}_{i,t}^{+\rightarrow\tau} = \frac{S_{i,t}}{N_{i,t}} + \sqrt{\frac{2\ln(t+\tau)}{N_{i,t}}}$ denote UCB on quality of a worker i when he does not get any task from t to $t+\tau$. Then τ_t, the block size at the allocation of the task t is defined as:

$$\tau_t = \operatorname*{arg\,max}_{0\le\tau\le T-t} \left\{ \begin{array}{l} M\hat{q}_{i,t}^{+\rightarrow\tau} - b_{i,t} \le M\hat{q}_{I_t,t}^{+\overset{-}{\rightarrow}\tau} - b_{I_t,t}, \\ \forall i \neq I_t \end{array} \right\}. \tag{2}$$

Thus, τ_t represents the maximum number of additional tasks that can be given to worker I_t even if he fails to successfully complete the tasks in the future while UCB indices on qualities of other workers evolve according to $\hat{q}_{i,t}^{+\rightarrow\tau}$. Figure 1 demonstrate this scheme pictorially.

4.2 Properties of the Allocation Rule

We first make the following observation.

Observation 1: $\forall t > e,\ t \ge n \ge 0, \tau \ge 0,\ \frac{\ln(t)}{n} \ge \frac{\ln(t+\tau)}{n+\tau}$.

Proof. $t \ge n$ and $\ln(t) \ge 1 \implies \tau\frac{\ln(t)}{n} \ge \frac{\tau}{n} \ge \frac{\tau}{t}$. Using, $\forall x > -1,\ x \ge \ln(1+x)$, we get $\frac{\tau}{t} \ge \ln(1+\frac{\tau}{t})$. Thus,

$$\frac{\tau\ln(t)}{n} \ge \ln(1+\frac{\tau}{t}) \ge \ln(t+\tau) - \ln(t)$$
$$\implies \ln(t)(1+\frac{\tau}{n}) \ge \ln(t+\tau) \implies \frac{\ln(t)}{n} \ge \frac{\ln(t+\tau)}{n+\tau}.\square$$

Let, $\Gamma^j_{i,t,b_t}(\tau) = M\hat{q}_{j,t}^{+\rightarrow\tau} - b_{j,t} - M\hat{q}_{i,t}^{+\overset{-}{\rightarrow}\tau} + b_{i,t}$. Intuitively, this term captures minimum loss in social welfare if the mechanism allocates next $\tau+1$ tasks to a worker j instead of i with the bid vector b_t. We will drop b_t from the notation if it clear from the context. From Equation (2), $\Gamma^j_{I_t,t}(\tau) \le 0,\ \forall j \neq I_t,\ \forall\tau,\ 0 \le \tau \le \tau_t$. The following Lemma shows that $\forall j,\ \Gamma^j_{I_t,t}(\tau)$ increases monotonically with τ.

Lemma 1 *The term $\Gamma^j_{I_t,t}(\tau)$ increases with τ, $\forall j \neq I_t$. That is, $\Gamma^j_{I_t,t}(\tau+s) \ge \Gamma^j_{I_t,t}(\tau),\ \forall j \neq I_t,\ \forall s \ge 0$.*

Proof. From the definition of $\hat{q}_{i,t}^{+\rightarrow\tau}$, $\forall i$,

$$\hat{q}_{i,t}^{+\rightarrow\tau+s} \ge \hat{q}_{i,t}^{+\rightarrow\tau}. \tag{3}$$

Applying Observation 1, $\sqrt{\frac{2\ln(t+\tau+s)}{N_{I_t,t+\tau+s}}} \le \sqrt{\frac{2\ln(t+\tau)}{N_{I_t,t+\tau}}}$ and since $\frac{S_{I_t,t}}{N_{I_t,t+\tau+s}} \le \frac{S_{I_t,t}}{N_{I_t,t+\tau}}$,

$$\hat{q}_{I_t,t}^{+\overset{-}{\rightarrow}\tau+s} \le \hat{q}_{I_t,t}^{+\overset{-}{\rightarrow}\tau}. \tag{4}$$

Using the above two inequalities,

$$\Gamma^j_{I_t,t}(\tau+s) = M\hat{q}_{j,t}^{+\rightarrow\tau+s} - b_{j,t} - M\hat{q}_{I_t,t}^{+\overset{-}{\rightarrow}\tau+s} + b_{I_t,t}$$
$$\ge M\hat{q}_{j,t}^{+\rightarrow\tau} - b_{j,t} - M\hat{q}_{I_t,t}^{+\overset{-}{\rightarrow}\tau} + b_{I_t,t} \ge \Gamma^j_{I_t,t}(\tau).\square$$

Thus, $\forall\tau, s,\ \Gamma^j_{I_t,t}(\tau) > 0 \implies \Gamma^j_{I_t,t}(\tau+s) > 0$. Therefore, for deciding a block size τ_t, it is enough to take the first value of τ such that $\Gamma^j_{I_t,t}(\tau+1) > 0$ for some worker j.

To achieve incentive compatibility, it is important that the allocation rule satisfies monotonicity condition with respect to the cost [15]. The following Lemma ensures this property:

Lemma 2 *The number of allocated tasks to any worker i for any task t, monotonically decreases with his reported cost if the costs of the other workers are fixed.*

Proof. Consider two bids by a worker i, $b^+_{i,t} > b_{i,t}$ and the corresponding bid vectors be b^+_t and b_t respectively. From Equation (1), $I_t(b_t) \neq i \implies I_t(b^+_t) \neq i$, thus, satisfying monotonicity condition trivially. If $I_t(b_t) = i$ but $I_t(b^+_t) \neq i$, then there is nothing to prove. Let k and k' denote the additional number of tasks, worker i gets with bid $b_{i,t}$ and $b^+_{i,t}$ respectively. From Equation (2),

$$k = \operatorname*{arg\,max}_{0\le\tau\le T-t} \left\{ \Gamma^j_{i,t,b_t}(\tau) \le 0, \forall j \neq i \right\},$$
$$\implies \Gamma^j_{i,t,b_t}(k+1) > 0 \text{ for some } j \neq i.$$

Note that for a fixed τ and j, $\Gamma^j_{i,t,b_t}(\tau)$ monotonically increases with $b_{i,t}$ when all the other bids are fixed. Thus, $\Gamma^j_{i,t,b^+_t}(k+1) \ge \Gamma^j_{i,t,b_t}(k+1) > 0 \implies k' \le k.\quad\square$

4.3 Payment Scheme

We use externality based payment which also accounts for the loss of learning the qualities of the other workers. Let $j^\tau \in \arg\max_{j\neq I_t}(M\hat{q}_{j,t}^{+\rightarrow\tau} - b_{j,t})$, be the second best worker with the current beliefs about the qualities and the reported bids. Let, $\hat{q}_{I_t,t}^{+\overset{\sim}{\rightarrow}\tau} = \frac{S_{i,t}}{N_{i,t}} + \sqrt{\frac{2\ln(t+\tau)}{N_{i,t+\tau}}}$ be the UCB on the quality of worker i when he gets tasks from t to $t+\tau$ and he performs the tasks with same empirical estimate as at time t. Then, the payment to the worker I_t is the bid required by him to match the utility of the requester by assigning the task $t+\tau$ to the worker j^τ by performing the future tasks with the same quality. The payment for task $t+\tau$, with, $0 \le \tau \le \tau_t$ is given as:

$$p_{I_t,t+\tau}(b_{i,t}, b_{-i,t}; h^t) = M\hat{q}_{I_t,t}^{+\overset{\sim}{\rightarrow}\tau} - \left(M\hat{q}_{j^\tau,t}^{+\rightarrow\tau} - b_{j^\tau,t}\right). \tag{5}$$

To avoid notation clutter, we will use index j instead of j^τ, but it should be clear from the context that this index depends on the value of τ. We make an implicit assumption that each worker genuinely submits an assigned task and honestly performs the task. So, CrowdUCB offers a payment to a worker for each allocated task even if the submission is not successful. Not performing the task honestly, may affect the learning of the algorithm and hence more sophisticated analysis may be needed. Moreover, since we are interested in low cost tasks this assumption holds true in many practical situations. The mechanism can be easily modified to the case where a payment is not offered to a worker in the event the worker fails to submit the task successfully.

5. PROPERTIES OF CrowdUCB: UNKNOWN QUALITY SETTING

When the workers are unaware of their true qualities and learn their qualities as the mechanism progresses, we show that CrowdUCB is EPIR (Theorem 1), EPIC (Theorem 2), and achieves logarithmic regret (Theorem 3).

Theorem 1 *CrowdUCB is EPIR.*

PROOF. We need to prove that $\forall i,\ \forall t,\ \forall h^t,\ \forall b_{-i,t}$, $u_{i,t}(c_i, b_{-i,t}; h^t; c_i) \geq 0$. If $I_t \neq i$, $u_{i,t}(c_i, b_{-i,t}; h^t; c_i) = 0$ and hence nothing to prove. If $I_t = i$, $u_{i,t}(c_i, b_{-i,t}; h^t; c_i) = -c_i + p_{i,t}(c_i, b_{-i,t}; h^t)$. Hence it is enough to prove that,

$$p_{I_t,t+\tau}(c_{I_t}, b_{-I_t,t}; h^t) - c_{I_t} \geq 0\ ,\ \forall\tau,\ 0 \leq \tau \leq \tau_t.$$

CrowdUCB allocation rule is designed such that, $\forall j \neq I_t, \forall\tau \in \{0, 1, 2, \ldots, \tau_t\}, \forall b_{j,t},\ \Gamma^j_{I_t,t,b_t}(\tau) \leq 0$. This leads to the following inequalities: $\forall\, j \neq I_t, \forall\, \tau \in \{0, 1, \ldots, \tau_t\}$,

$$\begin{aligned}
& M\hat{q}_{j,t}^{+\to\tau} - b_{j,t} - M\hat{q}_{I_t,t}^{+\overset{-}{\to}\tau} + c_{I_t} \leq 0 \\
& \implies c_{I_t} \leq M\hat{q}_{I_t,t}^{+\overset{-}{\to}\tau} + b_{j,t} - M\hat{q}_{j,t}^{+\to\tau} \\
& \implies c_{I_t} \leq M\hat{q}_{I_t,t}^{+\overset{\sim}{\to}\tau} + b_{j,t} - M\hat{q}_{j,t}^{+\to\tau} \\
& (\text{As, } \hat{q}_{I_t,t}^{+\overset{\sim}{\to}\tau} = \tfrac{S_{I_t}}{N_{I_t}} + \sqrt{\tfrac{2\ln(t+\tau)}{N_{I_t}+\tau}} \geq \tfrac{S_{I_t}}{N_{I_t}+\tau} + \sqrt{\tfrac{2\ln(t+\tau)}{N_{I_t}+\tau}} = \hat{q}_{I_t,t}^{+\overset{-}{\to}\tau}) \\
& \implies c_{I_t} \leq p_{I_t,t+\tau}(c_{I_t}, b_{-i,t}; h^t) \qquad (\text{From Equation (5)}).
\end{aligned}$$

□

We first make following observation before proving that CrowdUCB is EPIC.

Observation 2: If tasks t to $t+k_t$ are allocated to worker i, then $\forall j \neq i,\ \forall\tau,\ 0 \leq \tau \leq k_t$ following holds: $\hat{q}_{j,t}^{+\to k_t} = \hat{q}_{j,t+\tau}^{+\to k_t-\tau}$

PROOF. Since tasks t to $t+k_t$ are given to worker i and not to worker j, $S_{j,t} = S_{j,t+\tau}$ and $N_{j,t} = N_{j,t+\tau}$. Thus, the observation is immediate from the definition of $\hat{q}_{j,t}^{+\to\tau}$. □

Theorem 2 *CrowdUCB is EPIC when the workers have the same beliefs about their qualities as the requester and do not know the value of T.*

PROOF. We need to prove that $\forall i,\ \forall c_{-i},\ \forall c_i,\ \forall h^t,\ \forall b_{i,t}$

$$\sum_{t=1}^{T} u_{i,t}(c_i, c_{-i}; h^t; c_i) \geq \sum_{t=1}^{T} u_{i,t}(b_{i,t}, c_{-i}; h^t; c_i). \tag{6}$$

We will show that for any task t, it is not beneficial for a worker to underbid or overbid even after considering the future utility gains. For a task t, there are two possibilities of manipulation for the worker i,

(I) *underbidding:* $b_{i,t} < c_i$

(II) *overbidding:* $b_{i,t} > c_i$.

Let $k_t, k'_t \geq 0$ denote the number of tasks allocated to the worker i with bids c_i and $b_{i,t}$ respectively. We will prove that non-truthful bidding results in lesser utility for all tasks from t to $\max(t+k_t-1, t+k'_t-1)$, thus proving Equation (6).

Underbidding

For this case, $k'_t \geq k_t$ from Lemma 2.

Case 1: $\mathbf{k_t = 0}$: That is $I_t \neq i$ if everybody bids truthfully. If $k'_t = 0$, there is nothing to prove. If $k'_t > 0$, $\forall\tau$, s.t, $0 \leq \tau \leq k'_t - 1$: $u_{i,t+\tau}(b_{i,t}, c_{-i}; h^t; c_i) = -c_i + M\hat{q}_{i,t}^{+\overset{\sim}{\to}\tau} - M\hat{q}_{j,t}^{+\to\tau} + c_j$, with $j = \arg\max_{j\neq i} M\hat{q}_{j,t}^{+\to\tau} - c_j$. Moreover, since $k_t = 0$, $\exists l$ s.t. $M\hat{q}_{i,t}^{+} - c_i \leq M\hat{q}_{l,t}^{+} - c_l \leq M\hat{q}_{l,t}^{+\to\tau} - c_l \leq M\hat{q}_{j,t}^{+\to\tau} - c_j$. From Observation 1, $\hat{q}_{i,t}^{+\overset{\sim}{\to}\tau} \leq \hat{q}_{i,t}^{+} \implies u_{i,t+\tau}(b_{i,t}, c{-}i; h^t; c_i) \leq 0 \leq u_{i,t+\tau}(c_i, c_{-i}; h^t; c_i)$. The later inequality follows from the EPIR property.

Case 2: $\mathbf{k_t > 0}$: Since the payment to the worker i is independent of his bid, $\forall\tau$, with $0 \leq \tau \leq k_t - 1$: $u_{i,t+\tau}(b_{i,t}, c_{-i}; h^t; c_i) = u_{i,t+\tau}(c_i, c_{-i}; h^t; c_i)$. If $k'_t = k_t$, Equation (6) holds true. Let $k = k'_t - k_t > 0$. In this case, we will prove that $\forall\tau,\ 0 \leq \tau \leq k-1$, $u_{i,t+k_t+\tau}(b_{i,t}, c_{-i}; h^t; c_i) = u_{i,t+k_t+\tau}(c_i, c_{-i}; h^{t+k_t}; c_i)$ if worker i beliefs over his quality is same as that of the requester. Let, $s^{k_t} = \hat{q}_{i,t} * k_t$ denotes the belief of a worker i about the number of successes he will get in alloted k_t tasks. If truthful bidding again at $t + k_t$ results positive number of allocations for worker i then,

$$\begin{aligned}
& u_{i,t+k_t+\tau}(b_{i,t}, c_{-i}; h^t; c_i) \\
& = -c_i + M\hat{q}_{i,t}^{+\overset{\sim}{\to}k_t+\tau} - M\hat{q}_{j,t}^{+\to k_t+\tau} + c_j \\
& = -c_i + M\left(\frac{S_{i,t}}{N_{i,t}} + \sqrt{\frac{2\ln(t+k_t+\tau)}{N_{i,t}+k_t+\tau}}\right) - M\hat{q}_{j,t}^{+\to k_t+\tau} + c_j \\
& = -c_i + M\left(\frac{S_{i,t}+s^{k_t}}{N_{i,t}+k_t} + \sqrt{\frac{2\ln(t+k_t+\tau)}{N_{i,t}+k_t+\tau}}\right) - M\hat{q}_{j,t}^{+\to k_t+\tau} + c_j
\end{aligned}$$

Since worker i believes that his estimated quality will remain the same as t i.e. $\hat{q}_{i,t} = \hat{q}_{i,t+k_t}$, thus:

$$\begin{aligned}
u_{i,t+k_t+\tau}(b_{i,t}, c_{-i}; h^t; c_i) &= -c_i + M\hat{q}_{i,t+k_t}^{+\overset{\sim}{\to}\tau} - M\hat{q}_{j,t}^{+\to k_t+\tau} + c_j \\
&= -c_i + M\hat{q}_{i,t+k_t}^{+\overset{\sim}{\to}\tau} - M\hat{q}_{j,t+k_t}^{+\to\tau} + c_j \\
&\qquad\qquad (\text{From Observation 2}) \\
&= u_{i,t+k_t+\tau}(c_i, c_{-i}; h^{t+k_t}; c_i).
\end{aligned}$$

However, if truthful bidding again at $t + k_t$ results in zero number of allocations for worker i then the arguments go similar to case 1.

Overbidding

For this case, $k'_t \leq k_t$ from Lemma 2.

Case 1: $\mathbf{k_t = 0}$: This implies $k'_t = 0$ and there is nothing to prove.

Case 2: $\mathbf{k_t > 0}$: If $k'_t = 0$, then from Theorem 1, $u_{i,t+\tau}(c_i, c_{-i}; h^t; c_i) \geq 0\ \forall\tau,\ 0 \leq \tau \leq k_t - 1$. However, it may happen that by overbidding, the worker i loses initial allocations which goes to another worker say j, thus decreasing the UCB index of worker j. In this case, if the payment made to the worker i when he gets allocation for later tasks depends on the UCB index of worker j then the utility of the worker i for those tasks can increase. However, since the worker i is unaware of the number of remaining tasks, he will prefer not to lose the confirmed allocations with positive utility for small gain in utility for future tasks which may not happen. Thus, overbidding does not benefit the worker. If $k'_t = k_t$, then $u_{i,t+\tau}(c_i, c_{-i}; h^t; c_i) = u_{i,t+\tau}(b_{i,t}, c_{-i}; h^t; c_i)\ \forall\tau,\ 0 \leq \tau \leq k_t - 1$, as the payment does not depend on the bid of the worker i. If $k'_t < k_t$, then $\forall\tau,\ 0 \leq \tau \leq k-1$, $u_{i,t+k_t+\tau}(b_{i,t+k_t}, c_{-i}; h^{t+k_t}; c_i) = u_{i,t+k_t+\tau}(c_i, c_{-i}; h^t; c_i)$ from Case 2 of Underbidding using similar arguments. □

Theorem 3 *The allocation induced by CrowdUCB is same as the allocation induced by UCB1. This leads to the social welfare regret of CrowdUCB to be $O(\ln T)$.*

PROOF. Since, CrowdUCB is EPIC, we assume that all the workers bid truthfully for every task. Using induction, we will prove that for any task t, an agent I_t is selected by CrowdUCB if and only if he is selected by UCB1. For task $t = 1$, the condition is trivially satisfied as both UCB1 and CrowdUCB allocates in round robin fashion to get first estimates. By induction hypothesis, workers selected by CrowdUCB and UCB1 till tasks $t-1$ are same. For task t, UCB1 selects the worker I_t with highest UCB index. Since the workers selected by UCB1 and CrowdUCB for previous tasks were same, UCB indices of all the workers at t remain same. If bidding happens for task t then CrowdUCB will also select worker with highest UCB index i.e. I_t. If bidding does not happen at task t and CrowdUCB selects agent $j \neq I_t$ then for some previous task s, we have: $M\hat{q}_{j,s}^{+\bar{\rightarrow} t-s} - c_j \geq M\hat{q}_{I_t,s}^{+\rightarrow t-s} - c_{I_t}$. But, this would mean $M\hat{q}_{j,t}^{+} - c_j \geq M\hat{q}_{I_t,t}^{+} - c_{I_t}$ (assuming there exists a single agent with highest UCB otherwise we would break the ties in similar way for both the algorithms) thus contradicting the fact that I_t has the highest UCB index at t. Thus, $j = I_t$ and the allocation induced by CrowdUCB is same as UCB1. Hence, CrowdUCB and UCB1 achieve same regret. $\square$

6. PROPERTIES OF CrowdUCB: KNOWN QUALITY SETTING

We now show that when the workers are aware of their own qualities but do not know qualities of other workers then they can manipulate the learning of the requester by non-truthful bidding. However, when the workers do not know the number of tasks a requester has, then CrowdUCB is $\varepsilon-$EPIC under the assumption that at any round t, maximum block size τ_t does not exceed some constant B. The parameter ε depends on the total number of tasks and and is sub-linear in T. As a corollary, we get that CrowdUCB is asymptotically EPIC.

Theorem 4 *CrowdUCB is $\varepsilon-$EPIC with $\varepsilon = \sqrt{2}MB(\ln(T))^{\frac{3}{2}}$.*

Proof. Let, worker i gets k_t', k_t tasks with bid $b_{i,t}$ and c_i respectively. We look at underbidding ($b_{i,t} < c_i$) and overbidding ($b_{i,t} > c_i$) cases separately and first bound the gain in utility $\varepsilon_{i,t}$ by non-truthful bidding.

Underbidding($k_t^{'} \geq k_t$)

Case 1:$k_t = 0$ **:** If $k_t^{'} = 0$ then $i \neq I_t$ in both the cases, hence utility in both the cases is zero. If $k_t^{'} > 0$ then from Theorem 2 case 1 of underbidding, worker i will get negative utility for tasks t to $t + k_t^{'} - 1$ as the utility of a worker for any task depends only on the learnt quality by the requester and not on his true quality. However, he may improve his learning by getting negative utility in initial rounds so as to increase his future utility. Since, the worker is unaware of the total number of tasks, he would not risk to incur loss in immediate utility.

Case 2:$k_t > 0$**:** If $k_t = k_t^{'}$, then the utility from tasks t to $t+k_t-1$ is same in both cases. If $k_t^{'} > k_t$, at task $t+k_t$ there would be one more auction with truthful bidding. As worker i knows his true quality, his belief over quality about task $t + k_t$ would be $\frac{S_{i,t}+q_i k_t}{N_{i,t}+k_t}$. In this case the belief of a worker i about his utility for task $(t + k_t + \tau)$ $\forall \tau,\ 0 \leq \tau \leq k_t^{'} - k_t$ is given as:

$$u_{i,t+k_t+\tau}(c_i, c_{-i}; h^t; c_i; q_i)$$
$$= M\Big(\frac{S_{i,t} + q_i k_t}{N_{i,t} + k_t} + \sqrt{\frac{2ln(t + k_t + \tau)}{N_{i,t} + k_t + \tau}}\Big) - c_i - M\hat{q}_{j,t+k_t}^{+\rightarrow\tau} - c_j.$$

Whereas, with bid $b_{i,t}$, the utility for task $(t+k_t+\tau)$ $\forall \tau,\ 0 \leq \tau \leq k_t^{'} - k_t$ is given as:

$$u_{i,t+k_t+\tau}(b_{i,t}, c_{-i}; h^t; c_i; q_i)$$
$$= M\Big(\frac{S_{i,t}}{N_{i,t}} + \sqrt{\frac{2ln(t + k_t + \tau)}{N_{i,t} + k_t + \tau}}\Big) - c_i - M\hat{q}_{j,t}^{+\rightarrow k_t+\tau} - c_j$$

Thus, gain in utility is given by:

$$u_{i,t+k_t+\tau}(b_{i,t}, c_{-i}; h^t; c_i; q_i) - u_{i,t+k_t+\tau}(c_i, c_{-i}; h^t; c_i; q_i)$$
$$= M\Big(\frac{S_{i,t}}{N_{i,t}} - \frac{S_{i,t} + q_i k_t}{N_{i,t} + k_t}\Big) \qquad \text{(From Observation 2)}$$
$$= Mk_t\Big(\frac{S_{i,t} - q_i N_{i,t}}{N_{i,t}(N_{i,t} + k_t)}\Big)$$

From Hoeffding's inequality, i.e., w.p. $\geq 1 - t^{-4}$ we have, $\frac{S_{i,t}}{N_{i,t}} - \sqrt{\frac{2\ln(t)}{N_{i,t}}} \leq q_i \leq \frac{S_{i,t}}{N_{i,t}} + \sqrt{\frac{2\ln(t)}{N_{i,t}}}$. Thus,

$$u_{i,t+k_t+\tau}(b_{i,t}, c_{-i}; h^t; c_i; q_i) - u_{i,t+k_t+\tau}(c_i, c_{-i}; h^t; c_i; q_i)$$
$$\leq Mk_t\left(\frac{S_{i,t} - \left(\frac{S_{i,t}}{N_{i,t}} - \sqrt{\frac{2\ln(t)}{N_{i,t}}}\right)N_{i,t}}{N_{i,t}(N_{i,t} + k_t)}\right) \qquad \text{(w.p. } \geq 1 - t^{-4}\text{)}$$
$$= Mk_t\left(\frac{\sqrt{2\ln(t)}}{\sqrt{N_{i,t}}(N_{i,t} + k_t)}\right) \leq Mk_t\left(\frac{\sqrt{2\ln(T)}}{N_{i,t}^{\frac{3}{2}}}\right)$$
$$\leq MB\left(\frac{\sqrt{2\ln(T)}}{N_{i,t}^{\frac{3}{2}}}\right). \qquad \text{(As, } k_t \leq B\text{)}$$

Overbidding($k_t^{'} \leq k_t$)

Case 1: $k_t^{'} = k_t = 0$**:** Nothing to prove.

Case 2: $k_t^{'} < k_t$**:** If $k_t' = 0$, then worker i loses initial allocations but can increase his future utility by affecting the learning of other workers. However, since the number of tasks is unknown to the worker, he will not risk to lose the immediate positive utility. When $k_t' > 0$, utility of worker i from t to $t + k_t^{'} - 1$ is same. At task $t + k_t^{'}$, there would be one more auction in overbidding case. With known true quality q_i, worker i's belief about his quality at the end of $t+k_t^{'}$ tasks will be, $\frac{S_{i,t}+q_i k_t^{'}}{N_{i,t}+k_t^{'}}$. Thus, his belief over his utility from tasks $t + k_t^{'} + \tau$ $\forall$ τ $s.t$ $0 < \tau < k_t - k_t^{'}$ by overbidding is given by:

$$u_{i,t+k_t'+\tau}(b_{i,t}, c_{-i}; h^t; c_i; q_i)$$
$$= M\left(\frac{S_{i,t} + q_i k_t'}{N_{i,t} + k_t'} + \sqrt{\frac{2\ln(t + k_t' + \tau)}{N_{i,t} + k_t' + \tau}}\right) - c_i - M\hat{q}_{j,t+k_t'}^{+\rightarrow\tau} - c_j$$

With truthful bidding his utility is given by:

$$u_{i,t+k_t'+\tau}(c_i, c_{-i}; h^t; c_i; q_i)$$
$$= M\left(\frac{S_{i,t}}{N_{i,t}} + \sqrt{\frac{2\ln(t + k_t' + \tau)}{N_{i,t} + k_t' + \tau}}\right) - c_i - M\hat{q}_{j,t}^{+\rightarrow\tau+k_t'} - c_j$$

Thus, the gain in utility is given by:

$$u_{i,t+k_t'+\tau}(b_{i,t}, c_{-i}; h^t; c_i; q_i) - u_{i,t+k_t'+\tau}(c_i, c_{-i}; h^t; c_i; q_i)$$
$$= M\left(\frac{S_{i,t} + q_i k_t'}{N_{i,t} + k_t'} - \frac{S_{i,t}}{N_{i,t}}\right) \qquad \text{(From Observation 2)}$$

$$\leq Mk_t^{'}\left(\frac{\sqrt{2\ln(T)}}{N_{i,t}^{\frac{3}{2}}}\right) \qquad (\text{w.p.} \geq 1 - t^{-4})$$

$$\leq MB\left(\frac{\sqrt{2\ln(T)}}{N_{i,t}^{\frac{3}{2}}}\right) \qquad (\text{As, } k_t^{'} \leq B)$$

Here, second last inequality follows from similar arguments presented in case 2 of underbidding and using other side of Hoeffding's inequality. Note that maximum gain in utility for any task t by a worker i is given by $M\hat{q}_{i,t}^{+\tilde{\rightarrow}\tau} - c_i - (M\hat{q}_{j,t}^{+\rightarrow\tau} - c_j) \leq M + c_{\max}$. As qualities are bounded above by 1 and c_j by $c_{\max}$. We now bound the expected gain in utilities over time horizon T i.e.

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(u_{i,t}(c_i, c_{-i}; h^t; c_i; q_i) - u_{i,t}(b_i, c_{-i}; h^t; c_i; q_i)\right)\right]$$

$$\leq MB\sqrt{2\ln(T)}\left(\sum_{t=1}^{T}\frac{1}{N_{i,t}^{\frac{3}{2}}}(1 - t^{-4})\right) + \sum_{t=1}^{T} t^{-4}(M + c_{\max})$$

$$\leq MB\left(\sqrt{2\ln(T)}\ln(N_{i,t})\right) + (\frac{1}{3} + \frac{1}{3T^3})(M + c_{\max})$$

$$\leq \sqrt{2}MB\ln(T)^{\frac{3}{2}} + (\frac{1}{3} + \frac{1}{3T^3})(M + c_{\max}) \qquad (N_{i,t} \leq T, \ \forall\ t\ )$$

□

Since ε is sub-linear in T, we get the following Corollary:

Corollary 1 *CrowdUCB is asymptotically EPIC under known quality setting.*

Note that CrowdUCB remains to be EPIR in this setting since the proof of Theorem 1 for worker i uses his true cost and not the bid that may change based on the knowledge of true quality.

7. SIMULATIONS

In this section, we show via simulations that CrowdUCB achieves logarithmic regret with constants matching the regret of the UCB1 algorithm [2]. Further, randomized mechanisms such as in [4] lead to very high variance in payments.

For the simulations, we fix the number of workers K to be 10. For each worker i, a fixed quality q_i and a fixed cost c_i are drawn independently and uniformly from the interval $[0, 1]$. For each successfully completed task, the requester receives a reward of $M = 2$. Figure 2 shows the regret of CrowdUCB and UCB1 with the number of tasks taken on a logarithmic scale along the X-axis. As can be seen from the figure, regrets of CrowdUCB and UCB1 match exactly and show logarithmic behavior.

As an example of a randomized mechanism, we have implemented the mechanism in [4] by modifying the mechanism for a crowdsourcing setting with the value of resampling parameter μ as 0.1. For a fixed cost vector and fixed quality vector, we first generate an instance (that is, success realization) by fixing a $K \times T$ table, where each entry (i, t) represents the reward by the worker i if the worker had been allocated task t. The entries of success realization are filled by obtaining Bernoulli rewards, that is, the entry (i, t) is 1 with probability q_i and 0 with probability $(1 - q_i)$. Figure 3 shows that, for a fixed success realization, the randomized mechanism incurs high variance in the payments whereas CrowdUCB being a deterministic mechanism yields the same payment when the mechanism is invoked several times for a fixed instance. The total payment given to the workers is plotted over 1000 calls to the mechanisms for a fixed instance and the shaded area represents the payment between maximum and minimum payments given by the randomized mechanism.

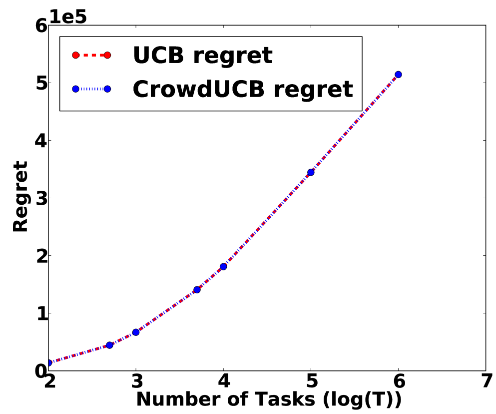

Figure 2: Regret of UCB1 and CrowdUCB on log scale of number of tasks.

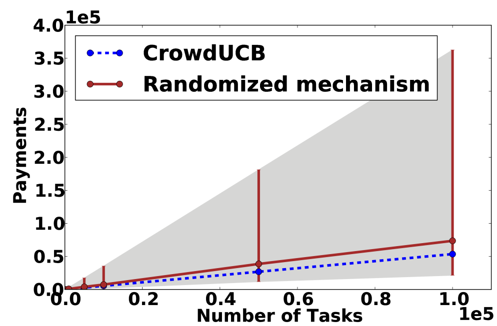

Figure 3: High variance in payments by a randomized mechanism for a fixed instance.

8. SUMMARY AND FUTURE DIRECTIONS

In this paper, we proposed a deterministic, ex-post individually rational and ex-post incentive compatible MAB mechanism that allows block allocations, and permits the workers to change their bids as the time progresses. This work is an important step towards designing a regret minimizing, deterministic MAB mechanisms for crowdsourcing and other general procurement settings. The work can be extended to a setting where workers' valuations also change over time. In this case, the optimal worker will change and hence one has to invoke contextual bandits. The other extension could be to design a deterministic mechanism with low regret that satisfies a stronger notion of truthfulness like dominant strategy incentive compatibility.

REFERENCES

[1] I. Abraham, O. Alonso, V. Kandylas, and A. Slivkins. Adaptive crowdsourcing algorithms for the bandit

survey problem. In *Conference On Learning Theory*, volume 30 of *JMLR Proceedings*, pages 882–910. JMLR.org, 2013.

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

[3] M. Babaioff, S. Dughmi, R. Kleinberg, and A. Slivkins. Dynamic pricing with limited supply. In *Thirteenth ACM Conference on Electronic Commerce*, 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*, 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*, 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*, 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*, 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] J. Bulow and P. Klemperer. Auctions versus negotiations. *The American Economic Review*, 86(1):180–194, 1996.

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

[11] M. Feldman, N. Immorlica, B. Lucier, T. Roughgarden, and V. Syrgkanis. The price of anarchy in large games. *CoRR*, abs/1503.04755, 2015.

[12] C. J. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning*, volume 28, pages 534–542, 2013.

[13] S. Li, M. Mahdian, and R. P. McAfee. Value of learning in sponsored search auctions. In *Internet and Network Economics - 6th International Workshop, WINE 2010, Stanford, CA, USA, December 13-17, 2010. Proceedings*, pages 294–305, 2010.

[14] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

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

[16] Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In *Twenty Second Internation World Wide Web Conference*, pages 1157–1166, 2013.

[17] A. Singla and A. Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Twenty Second International World Wide Web Conference*, pages 1167–1178, 2013.

[18] A. Slivkins and J. W. Vaughan. Online decision making in crowdsourcing markets: Theoretical challenges. *SIGecom Exchanges*, 12(2), December 2013. Position paper and survey.

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

[20] L. Tran-Thanh, M. Venanzi, A. Rogers, and N. R. Jennings. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. In *Twelfth International Conference on Autonomous Agents and Multiagent Systems*, pages 901–908, May 2013.

[21] C. A. Wilkens and B. Sivan. Single-call mechanisms. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 946–963, New York, NY, USA, 2012. ACM.