Auction Based Mechanisms for Dynamic Task Assignments in Expert Crowdsourcing

Sujit Gujar and Boi Faltings

International Institute of Information Technology, Hyderabad, India**
sujit.gujar@iiit.ac.in
Artifitial Intelligence Laboratory, EPFL, Lausanne, Switzerland boi.faltings@epfl.ch

Abstract. Crowdsourcing marketplaces link large populations of workers to an even larger number of tasks. Thus, it is necessary to have mechanisms for matching workers with interesting and suitable tasks. Earlier work has addressed the problem of finding optimal workers for a given set of tasks. However, workers also have preferences and will stay with a platform only if it gives them interesting tasks. We therefore analyze several matching mechanisms that take into account workers' preferences as well. We propose that the workers pay premiums to get preferred matches and auction-based models where preferences are expressed through variations of the payment for a task. We analyze the properties of two matching different mechanisms: Split Dynamic VCG (SDV) and e-Auction. We compare both the mechanisms with Arrival Priority Serial Dictatorship (APSD) empirically for efficiency.

1 Introduction

Crowdsourcing has emerged as a new paradigm in getting work done, where human agents solve tasks that are difficult to solve by software agents. Crowdsourcing is used in numerous applications. For example, hand written character recognition is easy for humans, but it is difficult for software. Real-time applications like VizWiz [3] leverage crowdsourcing to gather specific information from an image.

In the crowdsourcing market, there are two types of users of the platform, a *requester*, the one who posts and a *worker*, the one who seeks to work on the tasks. Typically, the term crowdsourcing refers to settings where the requesters post simple microtasks which can be performed quickly by human workers. With the success in crowdsourcing, it is becoming prevalent to crowdsource complex macro tasks which is referred to as *expert crowdsourcing* [22]. For example, oDesk, topcoder are expert crowdsourcing platforms. Consider the following scenario of an expert crowdsourcing as shown in Fig. 1.

Example 1. On a Monday morning three requesters login to a crowdsourcing platform with their tasks. These tasks are to develop software modules and are having deadlines in two weeks. w_1, w_2 and w_3 are eligible and interested workers for these tasks. The worker w_3 prefers to work on r_1 then r_2, r_3. Similarly the other workers have preferences over the tasks, as shown in Fig. 1. w_1 is available from Monday morning till

** This work was carried out when the first author was a post-doctoral researcher at EPFL

Tuesday evening for the task assignment, where as w_2 and w_3 are present only on Monday and Tuesday respectively. The goal is to optimally assign the tasks to the dynamic workers.

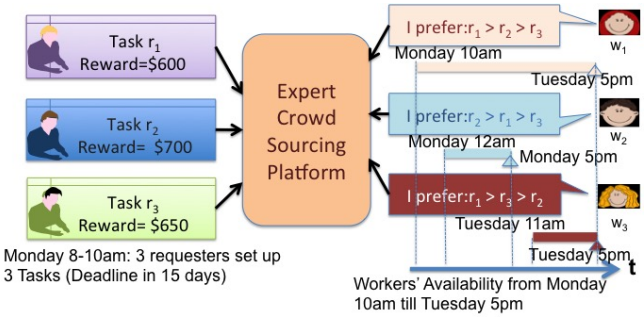

Fig. 1. An expert crowdsourcing scenario with three tasks and three workers

The prior work for the task assignments in crowdsourcing is mainly focused on catering to the requesters needs, i.e., addresses concerns with quality of the workers and the answer aggregation. It takes into account the workers' reputation and the requesters' requirements and budgets. However, one of the important advantages for the workers to work on the platform is to select tasks of their own choice. As seen from the above example, workers have preferences over the tasks they are assigned. In this paper, we address the task assignment problem in crowdsourcing catering to the workers' preferences.

Task Assignments With The Workers' Preferences. Difallah *et. al.* [5] proposed to push the tasks to the workers based on their preferences. These are only categorical preferences and not the workers' preferences for the requesters. The authors do not address strategic nature of the workers. Gujar and Faltings [9] addressed the strategic and dynamic workers. The authors model the problem as a two sided matching without money to achieve stability. In this paper, we consider task assignment to dynamically arriving workers. The dynamic workers are of two types, namely *exogenous* and *endogenous*. (i) The exogenous workers do not lie about their arrival-departure. (ii) The endogenous workers can report late arrival or early departure if its beneficial for them. [1] The goal in this paper is to improve *efficiency*, the valuation that workers assign to their assigned tasks, in task assignments to dynamic workers of an expert crowdsourcing platform.

We propose that workers pay the platform a *premium* to obtain a preferable match. We refer to such task assignment with side payments as a *matching mechanism*. A matching mechanism is designed to induce a truthful behaviour among the workers and achieve efficiency.

[1] The workers has to be logged into the system for his availability and hence cannot report early arrival or late departure.

To validate the hypothesis about side payments in task assignments, we conducted a survey with crowd workers. 92.6% of the workers were positive about the tasks assignments with side payments. Section 6 provides more details about the survey.

With such side payments, we design auction based matching mechanisms for the task assignment problem in an expert crowdsourcing platform. As users of these mechanisms may not be experts in game theory/mechanism design, the goal is to design mechanisms that are simple and avoid any complex models such as dynamic optimization using Markov Decision Processes and prior on the workers preferences. Secretary problem [1] based solutions are useful. However, the difference between classical secretary problem and the settings addressed here is, ours is a combinatorial problem with multiple recruiters interested in the workers who have high valuation for their tasks and ensuring truthfulness of reported preferences. Our contributions are as follows.

Contributions. We allow the workers to pay a premium to obtain a better match. We propose two dynamic matching mechanisms for the task assignment problem. For exogenous settings, we first develop a matching mechanism generalizing Vickrey-Clarke-Groves (VCG) mechanism, namely *Split Dynamic VCG* (SDV). We prove that SDV is strategyproof and k-competitive where k is the number of tasks. We show empirically that in SDV, the efficiency of matching improves by 5-10% over Arrival Priority Serial Dictatorship (APSD). In endogenous settings, we propose a strategyproof matching mechanism, e-Auction . It is adapted from e-competitive, single item dynamic auction proposed in [19]. The workers cannot manipulate their arrival departure periods in e-Auction . We prove that it is e^2-competitive for efficiency, as against SDV and Arrival Priority Serial Dictatorship (APSD) are k-competitive. However, empirically APSD and SDV perform better than e-Auction .

Organization. In the next section, we review the related literature. We describe our model, notation and assumptions in Section 3. We explain how to design matching mechanisms based on auctions in Section 4.With simulations, we analyze empirical efficiency of all the mechanisms in Section 5. In Section 6, we compare SDV, e-Auction and APSD mechanisms and discuss how to relax the assumptions made in the paper. We conclude the paper in Section 7.

2 Related Work

The term *crowdsourcing* was introduced by Howe [13]. Since then it has attracted researchers, practitioners, entrepreneurs, industrialists etc. Currently there are thousands of websites available for crowdsourcing.[2] It is a pull model of work where workers decide what to work on and when to work as against the traditional push model where management distributes the work among employees and monitors the progress. As it is an uncontrolled manner of getting work done, there is always a concern about the quality of the work done. Lots of research has been carried out to ensure the quality of the received answer [2, 4, 12, 11, 15–17, 20, 22]. Most of them use machine learning for the task assignments. For example, [15, 20] proposed EM based algorithms for quality

[2] http://crowdsourcing.org

management in the crowdsourcing. [22] proposed the multi-armed-bandit based algorithm to learn the qualities of the workers and analyzed it for regret. [14] discussed how to design the tasks so as to improve the quality of answers. Note that all this work is focused mainly for the requesters.

Kittur *et. al.* [18] studied worker's perspective on crowdsourcing. The authors conducted user studies on AMT and proposed different techniques to improve performance of the workers on the tasks by intelligent task design. [16, 8] considered bidding based task allocation, to elicit the costs incurred by the workers. Our paper is different from all the above as focus is on the task assignments catering to the qualified but strategic, dynamically arriving workers' preferences over the tasks and the bids are not for the costs incurred by the workers.

If side payments are not feasible and workers are static with ordinal preferences, the task assignment problem is same as assignment problem studied in economics. Sönmez and Ünver [21] is a survey that summarizes the results for the static assignment problem. For the dynamic endogenous workers with ordinal preferences, APSD is the only strategyproof mechanism [23]. In this paper, we use side payments to increase efficiency of the task assignment and address the problem as a mechanism design problem in auctions. For more about mechanism design theory, the interested readers are referred to [6, 7] and the references cite therein.

3 The Model and Notation

In this paper our focus is on the task assignments in an *expert crowdsourcing* market where workers are skilled, tasks are complex and rewards are relatively high as seen in Example (1). For designing matching mechanisms in this setting, we make the following assumptions.[3]

Assumptions:

- The number of tasks and the corresponding qualified workers is not so large that the workers will have difficulty in reporting their preferences. For example, on oDesk, on a particular day only a small portion of the tasks may be relevant for a software professional with specific skills.
- All the workers satisfy pre-qualification for the tasks.
- We focus on a time window during which each worker takes up only one task. We discuss how to relax this assumption in Section 6.
- The workers may strategize their preferences to get preferable tasks.

There are k tasks with i^{th} task denoted as r_i.$W = \{w_1, \ldots, w_k\}$ are eligible aspirants for these tasks who arrive dynamically to the market. Upon arrival in the time slot arr_j, a worker w_j observes the tasks in the system and reports his preferences over the tasks and a deadline dep_j until when he can accept a task. If an impatient worker needs a task immediately, he can indicate $dep_j = arr_j$. We capture the preferences of a worker by a valuation he assigns to the task he receives. Let b_{ij} be the value that w_j assigns to

[3] It should be noted that, the settings of expert crowdsourcing are different than microtasking where the workers finish the task quickly and move on to a next task immediately.

the matching where he is matched with the task r_i. We also interpret these numbers as a *bid*, maximum premium a worker is willing to pay, for the task. If a worker does not have any preferences over the tasks, he can set the preference to be a zero vector. The notation used in the paper is described in Table 1. With this notation, we now define

Table 1. Notation

k	Total number of tasks (workers)
R	The set of tasks
r_i	i^{th} Task
w_j	j^{th} worker
arr_j, dep_j	Arrival time and departure time for w_j
$R(t)$	The tasks not assigned till t.
$W(t)$	$\{w_j \ni arr_j \leq t \leq dep_j$ andw_j is not been assigned any task.$\}$
$AW(t)$	$\{w_j \vert arr_j = t\}$. Set of workers arriving in time slot t
$DW(t)$	$\{w_j \vert dep_j = t\}$. Set of workers departing in time slot t
$\mathbf{b_j}$	=(b_{ij}) Preference of w_j over the tasks
μ	Matching algorithm
p_j	Payment made by worker w_j to the platform for the matching.
$\mathbf{p}$	=$(p_j)_{j \in W}$

matching mechanism and desirable properties.

Matching Mechanisms. The task assignment problem is divided into two parts: (i) *A matching algorithm*(μ) - It produces (w, r) pairs, where the worker w is assigned with the task r[4]. A task should be assigned to a worker before he leaves the system. (ii) *Payment*: It decides a payment to be paid by the users for receiving preferable matches. We refer to a matching algorithm along with payments as a *matching mechanism*.

Desirable game theoretic properties of matching mechanisms. Let $\mathcal{M} = (\mu, \mathbf{p})$ be a given mechanism.
Strategyproof Let $\mathbf{b}_{-j} = (\mathbf{b}_1, \ldots, \mathbf{b}_{j-1}, \mathbf{b}_{j+1}, \ldots, \mathbf{b}_k)$ and w_j be assigned with $r_{i_1} = \mu(w_j)$ with payment p_j when he reports his preference as $\mathbf{b}_j$ and remaining workers report $\mathbf{b}_{-j}$. Let $r_{i_2} = \mu(w_j)$ when he reports $\mathbf{b}'_j$ while remaining workers report $\mathbf{b}_{-j}$ and his payment be p'_j. We say $\mathcal{M}$ is strategyproof, if $\forall w_j$,

$$b_{i_1 j} - p_j \geq b_{i_2 j} - p'_j \qquad \forall \mathbf{b}_{-j}.$$

That is, by misreporting the preferences over matching, the workers cannot gain.
Efficiency Let V^μ be the valuation that all the workers assign to their match. I.e., $V^\mu = \sum_j b_{\mu(w_j)j}$. We say μ is efficient if μ selects a matching that maximizes V^μ. If all the preferences are known before-hand, we can always find an optimal matching. This

[4] μ takes $\mathbf{b_j}$s, arr_j, dep_j as inputs and produces a bipartite matching. However to simplify notation, we just refer to μ as a bipartite matching.

is called as off-line optimal (OFF-OPT) solution and $V^{OFF-OPT}$ is valuation of the off-line optimal solution.

Competitive Ratio No matching mechanism can predict preferences of the workers yet to arrive and make perfect decisions for the departing workers. Hence in dynamic (on-line) settings, it is impossible to achieve efficiency. The *Competitive ratio* is a widely used measure of the performance of on-line algorithms. It measures how bad the solution found by the algorithm can be as compared to an optimal solution. An on-line algorithm is c-competitive if,

$$\mathbb{E}[V^{\mu}] \geq \frac{1}{c}\mathbb{E}[V^{OFF-OPT}] \quad \forall(\mathbf{b}_1, \ldots, \mathbf{b}_k)$$

The expectation is with respect to random orderings of the workers. Instead of expectation if we consider this ratio for each instance, in adversarial settings, it will be arbitrarily bad. Hence we measure it under a random hypothesis where all orders in which the workers arrive are equally likely.

We model task assignment as a dynamic auction. In the next section, we propose two matching mechanisms having different competitive ratios. We refer to the first matching mechanism, which is VCG based, as *SDV*. We also propose a matching mechanism e-Auction to improve efficiency. In both the mechanisms, matching algorithms are induced by the underlying auction's allocation rules.

4 Matching Mechanisms: Dynamic Auctions

In this section, we first describe APSD. We then propose our mechanisms.

4.1 Arrival Priority Serial Dictatorship (APSD)

Zou *et. al.* [23] proved that in endogenous settings, APSD is the only strategyproof mechanism for assignment problem when the preferences are ordinal. In this mechanism, the workers are assigned a priority based on their arrival time and the tasks are assigned according to the priority of the workers. It is k-competitive for efficiency and in general, it can lead to inefficient task assignments.

4.2 SDV: Dynamic VCG AUCTION

Let the system have tick events which are a collection of time slots. We say a tick event occurs when the system time matches with one of the time given in the list. Whenever a tick event occurs, the matching of the unassigned and available workers happens. At tick events, the platform solves an optimization problem described in (1). Let $\{t'\}$ be the tick events defined by the system, x_{ij} be an indicator variable indicating whether the task r_i is assigned to w_j or not, and $\alpha_{ij} = b_{ij}$.

$$\begin{array}{c} \max \sum_{i \in R(t'), j \in W(t')} \alpha_{ij} x_{ij} \\ s.t. \ \sum_i x_{ij} \leq 1 \ \ \forall i \ \in R(t') \\ \sum_j x_{ij} \leq 1 \ \ \forall j \ \in W(t') \\ x_{ij} \in \{0, 1\} \end{array} \tag{1}$$

For the matching mechanism SDV = (μ^{SDV}, P^{SDV}), μ^{SDV} is defined in Algorithm 1.

ALGORITHM 1: Matching Algorithm μ^{SDV}

Input: Workers' preferences ($\mathbf{b_j}$s)
Output: A matching

$t = 1, R(1) = R, W(1) = AW(1)$
if $t \in \{t'\}$ **then**
    Solve Optimization Problem (1)
    if $x_{ij} = 1$ **then**
        $\mu(w_j) = r_i$ and $\mu(r_i) = w_j$.
        Determine p_j^{SDV}
$t \leftarrow t + 1$
$R(t) \leftarrow R(t-1) \setminus \{r_i : \exists j \ni x_{ij} = 1\}$;
$W(t) \leftarrow \{W(t-1) \setminus \{w_j : \exists i \ni x_{ij} = 1\}\} \cup AW(t)$
if $W(t) == \emptyset$ *OR* $R(t) == \emptyset$ **then**
    STOP.
else
    Go to Step 2

Payment P^{SDV}: At the tick event t', let $OPT^{t'}$ be the value of the above optimization problem (1). VCG payment for w_j who is matched at t':

$$p_j^{SDV} = V_{-j}^* - OPT_{-j}^{t'} \tag{2}$$

where, $V_{-j}^* = OPT^{t'} - b_{i^*j}$. $OPT_{-j}^{t'}$ is the optimal value obtained by solving the above optimization problem with workers $W(t) \setminus w_j$.

□

Note that (i) the duration between two consecutive tick events is chosen independent of the users' preferences. (ii) It can be a minute or can be hours depending upon the task complexity. While defining tick events, the system ensures that every worker is present at least for one tick event. For example, the system can add all the departure periods of the workers in the list of tick events.

Even though integer programs are NP-hard, this particular optimization problem can be solved in polynomial time as this is a maximum weight bipartite matching between the tasks and the workers having edges from each task r_i to each worker w_j with weight $\alpha_{ij} = b_{ij}$. We illustrate SDV mechanism with an example.

Example 2. Consider the same scenario as depicted in Fig. 1 with bids as indicated in Table 2. Let the tick events t' be Monday evening 5pm (t=1) and Tuesday evening 5pm (t=2). At $t = 1$, the platform solves a maximum weight bipartite matching and assigns r_1 to w_1 and r_2 to w_2. VCG payments are 0 for both of them. At $t = 2$, r_3 is assigned to w_3 and his payment is also 0. Instead of $(5, 12, 1)$ if w_2 has preference $(12, 5, 1)$, he is assigned to task r_1 and needs to pay 1 whereas w_1 gets r_2 at no cost.

Table 2. SDV: Example Preferences

w_1	$r_1 \succ r_2 \succ r_3$	$\mathbf{b_1} = (10, 9, 0)$
w_2	$r_2 \succ r_1 \succ r_3$	$\mathbf{b_2} = (5, 12, 1)$
w_3	$r_1 \succ r_3 \succ r_2$	$\mathbf{b_3} = (15, 5, 10)$

From the above example, it is clear that the workers may not have to pay if their interests are not conflicting. As our goal is not to make money out of such matching mechanisms, low payments are acceptable. We now see the analysis of SDV.

Proposition 1. *SDV is strategyproof when workers are exogenous.*

Proof: We partition the workers using tick events such that in each subgroup, all the workers are available simultaneously and treat each of the subgroups as an independent problem. Each sub-problem is solved using VCG auction. No worker can manipulate SDV because, his preference cannot choose which VCG auction to be part of and each VCG auction is strategyproof. Thus, SDV is strategyproof.

□

Proposition 2. *SDV is k-competitive for efficiency of the matching.*

Proof: Let $V^{\mu^{SDV}}$ be the total valuation of the matching in SDV. For a given preference profile, let r_{i^*} be a task of a worker w_j in an optimal assignment and let r_i be a task assigned to him by SDV. The expected valuation of the matching to him be $\mathbb{E}[b_{ij}]$ where expectation is with respect to orderings of the workers. With the random ordering hypothesis, each agent is first to arrive with probability $\frac{1}{k}$.

$$
\begin{aligned}
\Rightarrow \quad & b_{ij} \geq b_{i^*j} \quad \text{with probability} \frac{1}{k} \\
\Rightarrow \quad & \mathbb{E}[b_{ij}] \geq \tfrac{1}{k} b_{i^*j} \\
\Rightarrow \quad & \textstyle\sum_{w_j} \mathbb{E}[b_{ij}] \geq \sum_{w_j} \frac{1}{k} b_{i^*j} \\
\Rightarrow \quad & \mathbb{E}[V^{\mu^{SDV}}] \geq \tfrac{1}{k} V^{OFF-OPT}
\end{aligned}
$$

The above holds true for any preference profile and hence SDV is k-competitive.

This bound is tight up to an additive constant 1 from the following preference. w_1 has valuation $\mathbf{b_1} = (k^2, 0, \ldots, 0)$. All the other workers have valuation $(\epsilon, 0, \ldots, 0)$, where is ϵ is very small positive real number. The optimal solution has value k^2 and SDV will achieve this with probability $\frac{1}{k}$. For all the instances where w_1 does not arrive before the first tick event where matching happens, SDV has valuation of ϵ leading to competitive ratio arbitrarily close to k.

□

This efficiency is based on the valuations that the workers assign to the matching and not based on payments or costs incurred by workers. To improve on the high competitive ratio of SDV, we propose e-Auction matching mechanism.

4.3 e-Auction

In [19], the following strategyproof auction was proposed for selling a single item to dynamically arriving k agents. We explain this for single task, k workers settings.

Single Task Dynamic Auction. The platform waits until it receives $\frac{k}{\mathrm{e}}$ bids. p, q be the two highest bids received so far. If the worker with bid p is available, allocate the task to him at price q. Otherwise, whenever a worker with bid higher than p arrives, allocate the task to that agent at price p. It is shown that the above auction is e-competitive for efficiency.

Multi Task Dynamic Auction. We adopt the above auction to our setting which is combinatorial. We have k tasks to be assigned to the k workers each requiring only one. We call the new matching mechanism e-*Auction* $= (\mu^{eA}, P^{eA})$. The algorithm μ^{eA} is described in Algorithm 2. Recall that the workers not having preferences for a task(s),

ALGORITHM 2: Matching Algorithm μ^{eA}

Input: Users Preferences ($\mathbf{b_j}s$)
Output: A Matching

1 Wait until $\frac{k}{\mathrm{e}}$ bids for each task have been received.
2 After this, for each task, if the highest bidder is present, and not assigned any task, allocate him the task at second highest bid received so far for the task.
3 If a worker is winner at more than one task, he is assigned with the task having highest pay-off (his bid $-$ payment).
4 This worker is marked as absent.
5 For all the tasks which are not assigned in the above step, the highest bid received for the task is marked as a reserve price.
6 The first worker who submits a bid higher than the reserve price and is not assigned to any of the other tasks is assigned the task.

put a bid of '0' and k being the number of workers, step 1 of the above algorithm will not wait indefinitely.

Payment P^{eA}: Each worker who wins the task in the first $\frac{k}{\mathrm{e}}$ bids, has to pay the second highest bid received up to the first $\frac{k}{\mathrm{e}}$ bids for the task. Other workers, if they receive a task, pay the highest bid received up to the first $\frac{k}{\mathrm{e}}$ bids for that task.

Proposition 3. *e-Auction is strategyproof.*

Proof: In e-Auction payment, for all $w_j \in W$, p_j is independent of $\mathbf{b}_j$ and if a worker is winner in multiple tasks, he is assigned a task with the highest $b_{ij} - p_j$. Hence no worker has any incentive to misreport his bid. If an worker reports the late arrival than true arrival, it does not increase his chance of winning on any of the tasks. If the worker tries to report departure before $\frac{k}{e}$ workers have arrived, he will not get the task. After that, does not matter when is his departure. Thus, no worker can gain anything by late-arrival or early departure.

□

Proposition 4. *e-Auction is at-most e^2-competitive for efficiency of the matching.*

Proof: Let R' be the set of tasks assigned to the workers by e-Auction and W' be the set of workers who receive a task. Let $r_i = \mu^{eA}(w_j)$. $V^{\mu^{eA}} = \sum_{j \in W'} b_{ij} = \sum_{i \in R'} b_{ij}$.

For a single task case, from the classic secretary problem analysis, with probability $\frac{1}{\mathrm{e}}$, each task will be assigned to the worker having highest valuation for that task. Say $\mu^{eA}(r_{i*}) = w_{j*}$. With probability $\frac{1}{\mathrm{e}}$, $b_{i*j*} \geq b_{i*j}$, In particular, $b_{i*j*} \geq b_{i*j'}$ where $w_{j'}$ is the OFF-OPT assignment of the task r_{i*}.

In our settings, each worker can take up only one task. Say for the task r_{i*}, w_{j*} is the highest bidder. It may happen that a worker w_j* is the highest bidder at multiple tasks, and this may lead to the task r_{i*} being not assigned. If a winner for r_{i*} is not a winner at any other task, then definitely, the task assigned. If the valuations of the workers are independent and are identically distributed, each worker is equally likely to be winner at all the tasks. Hence, probability that a worker is winner exactly at on task is $(1 - \frac{1}{k})^{k-1}$ which is $\frac{1}{\mathrm{e}}$ for large k.

$$
\begin{aligned}
&\Rightarrow \text{Probability that a task is assigned } \geq \tfrac{1}{\mathrm{e}} \\
&\quad \mathbf{E}[V^{\mu^{eA}}] = \textstyle\sum_{i \in R'} \mathbf{E}[b_{ij}] \\
&\Rightarrow \mathbf{E}[V^{\mu^{eA}}] \geq \textstyle\sum_{i \in R} \frac{1}{\mathrm{e}} \mathbf{E}[b_{ij}] \\
&\Rightarrow \mathbf{E}[V^{\mu^{eA}}] \geq \textstyle\sum_{i \in R} \frac{1}{\mathrm{e}^2} \max_j b_{ij} \\
&\Rightarrow \mathbf{E}[V^{\mu^{eA}}] \geq \textstyle\sum_{i \in R} \frac{1}{\mathrm{e}^2} V^{OFF-OPT}
\end{aligned}
$$

This proves the claim.[5]

□

e-Auction is an interesting auction as on worst case analysis, it has a much lower competitive ratio as compared to SDV. The disadvantage of e-Auction is that some tasks may not get assigned in e-Auction .

5 Evaluation of the Mechanisms

The proposed mechanisms in this paper inherently hypothesize that workers will participate in bidding for the tasks. To evaluate this hypothesis, we conducted a survey with crowd workers.

5.1 Survey: Bidding based Task Assignments

We conducted the survey on Amazon Mechanical Turk (AMT).[6] To safeguard against spammers, only workers with high acceptance over at least 5000 HITs were allowed to participate in the survey. The survey included java questions to ensure that the participant has java programming knowledge. We told workers that there are java programming tasks having a reward of \$200[7] and we are researching about possibility of

[5] Note that this is upper bound on competitive ratio.

[6] http://mturk.com

[7] Note that we are referring to expert crowdsourcing tasks and not the microtasks. Hence, such rewards are feasible.

bidding based task allocation for high paying tasks. We asked the workers whether they had worked on crowdsourced programming tasks and will they be willing to bid to the platform in such task assignments and how much. The workers were paid \$0.1 for participation and bonus of \$0.9 to those who did well on java questions. 56 different workers participated. 75% of the participants claimed that they had worked on crowdsourced programming tasks. 45% were proficient in Java. 92.8% of all the participants and all of the java proficient responded positively for participating in the bidding based task assignments. We observed that the workers are interested in bidding aggressively on a task where their chance of getting the task is higher over the task they actually prefer.

This survey supports the notion of premium to be charged in the form of bids for the task-assignments in expert crowdsourcing environment. With this positive feedback from the workers, we further evaluate the mechanisms for empirical efficiency. We perform the empirical analysis by simulations.

5.2 Empirical Evaluation

We simulated the mechanisms by generating random preference profiles and arrival-departure for the users. For arrival of the workers, we assume the workers arrive in the system according to a Poisson process with mean λ and wait in the system according to an exponential distribution with mean $\mu = 2$. We used the following three generative models for the preferences of the workers.

- I *Uniform*: For each task, all the workers' bids are generated uniformly at random from interval [0,1].
- II *Single Peaked*: Each worker's ordinal preference over the tasks is generated using uniform distribution. His bid for the most desirable task is drawn uniformly at random from [1,2]. However, his bid for i^{th} ranked tasks is $\frac{1}{i}$ of his the most desirable task.
- III *Single Peaked with Popularity* In this model, we assume certain tasks are more desirable than the others. Here, the workers ordinal preferences are drawn according to the popularity. And these preferences are converted to bids in the same manner as in Single Peaked case. The ordinal preferences with popularity $(\phi_1, \phi_2, \ldots, \phi_k)$ are generated as described below.
  - $R_1 = \{1, 2, \ldots, k\}$ A task r_i is selected as the most preferred tasks from R_1 with probability $\frac{\phi_i}{\sum_{l \in R_1} \phi_l}$. Let it be, r_{j_1}
  - $R_2 = R_1 \setminus \{r_{j_1}\}$. Now the next best task is sampled from R_2 with probabilities proportional to $\frac{\phi_i}{\sum_{l \in R_2} \phi_l}$ and so on.

For each of these three generative models, with $k = 30$, we generated 10,000 different preference profiles and studied empirical efficiency by considering the average valuation of the matching per worker per task by varying λ. In SDV, we used tick events generated by workers arrival. That is, every worker is assigned a task as soon as he arrives.

Figure 2 shows empirical efficiency SDV, APSD, e-Auction and OFF-OPT. It is the average valuation of a matching per work normalized to OFF-OPT=1. Clearly, SDV

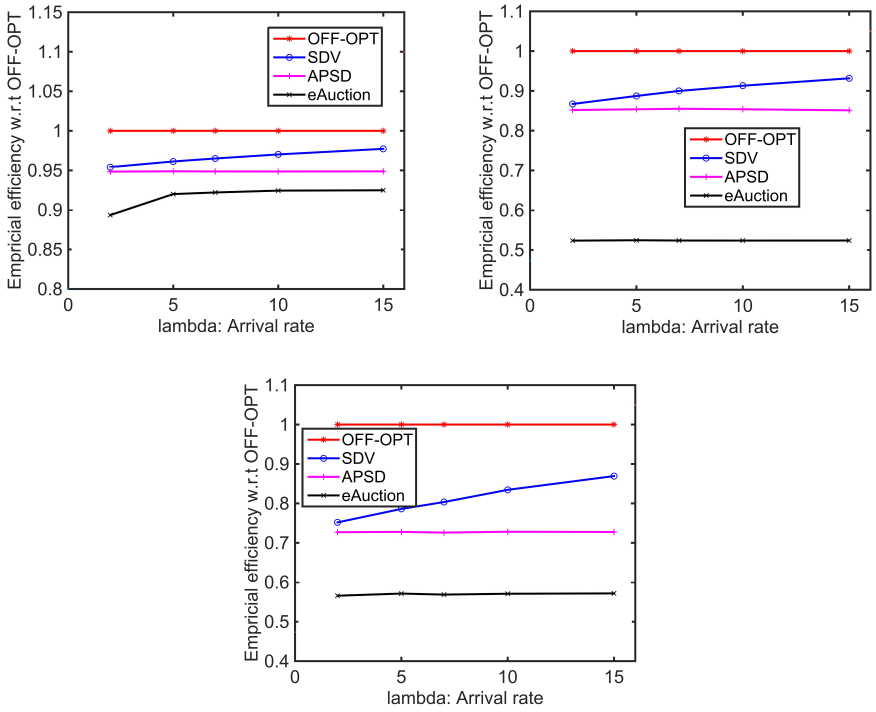

Fig. 2. Empirical Comparison of SDV, e-Auction , APSD and OFF-OPT for efficiency

improves over APSD by 2-3% for $\lambda \in (\frac{k}{6}, \frac{k}{4})$ when the workers are of type I. For workers of type II and III, this improvement is 5% and 10% respectively. This arrival rate matches on average λ workers together. Hence, if on average all the workers are willing to wait till λ workers to arrive, they clearly see an improvement in quality of matching by SDV. From simulations, it is clear that, though e-Auction has better worst case guarantees, it is within only 2-competitive (50% of the off-line optimal) where as SDV is 1.25-competitive (that is, within 80% of the off-line optimal).

In the next section, we provide a unified comparison of the mechanisms discussed in the paper. We also describe how to relax some of the assumptions.

6 Discussion

The crowdsourcing platform needs to propose/assign tasks to the workers. To keep users interested in the platform, it needs to achieve efficiency in such task assignments. Strategic workers may manipulate the task assignment by misreporting their preferences. So we need strategyproofness. Towards this, we focused on expert crowdsourcing and proposed two matching mechanisms namely, SDV and e-Auction . Table 3 compares both the proposed mechanisms for strategyproofness, efficiency and empirical efficiency for type III workers described in the previous Section.

Table 3. Comparison of mechanisms SDV, e-Auction and APSD

	SDV	eA	APSD
StrategyProofness	Y	Y	Y
Endogenous	N	Y	Y
Competitive Ratio	k	e^2	k
Expected Competitive Ratio	1.25	1.76	1.38
Is market cleared?	Y	N	Y

APSD does not need any side payments as well as works in endogenous settings. SDV assumes exogenous settings. Both the mechanisms have same guarantee on competitive ratio. However, empirically SDV performs better than APSD.
- In endogenous settings, we proposed e-Auction which has better competitive ratio. However, we observe in experiments, it performs poor for average case efficiency.
Thus, if we can assume exogenous settings, we propose to use SDV. If side payments are not desired or in endogenous settings, one can use APSD proposed by Zou *et. al.*[23]. If stronger worst case guarantee on competitive ratio is needed, we propose to use e-Auction . However, empirically, we observe that e-Auction does not perform that well. Note that, typically to best of our knowledge there is no prior work in online mechanism with competitive ratio guarantees when private information is multi-dimensional.

***e*-Auction : Empirical Analysis.** In APSD, the mechanism does not wait to gather any information about the valuations of the workers yet to arrive, but assigns the best possible task to the worker upon arrival. In worst case, such mechanism can lead to a very poor performance. In SDV, the mechanism waits certain duration defined by tick events and optimizes the sum of valuations of the workers present in the system at tick events. This improves the efficiency, but still on worst case, it may lead to poor performance. As opposed to these two matching mechanisms, e-Auction is designed to improve guarantees on the worst performance. It waits for $\frac{k}{\mathrm{e}}$ workers to arrive and learns about valuations of the workers and then matches workers and tasks. This improves the worst case guarantees. However, typically it loses opportunity to assign some tasks to the initial workers who might have left the platform before the assignments happen. This leads to a poor performance on average case analysis, as observed in the experiments (Table 3). To overcome this, in follow up work, we proposed Online Ranked Competition Auction (ORCA) [10]. ORCA waits for different thresholds for each task rather than uniform $\frac{k}{\mathrm{e}}$ that of e-Auction before matching the task to a worker. This should improve worst case performance further. Such analysis is still open. We observed, empirically ORCA improves over e-Auction . For more details, please refer to [10].

The expert crowdsourcing model. Note that the expert crowdsourcing settings in this paper are different from the widely referred crowdsourcing of micro tasks. Hence the assumptions made in the paper, may not model the microtasking environment well. We demonstrate how to relax some of the assumptions.
(a) If the workers want the tasks without waiting, all the mechanisms are valid. Either workers can set, $arr_j = dep_j$ or the tick events can occur at faster rates. As the analysis

is for worst case, it is still valid. However, in the long run, users may realize that quality of matching is better if everybody is patient.
(b) If a worker is unable to determine preferences or is not interested in reporting preferences, he can indicate it by $b_{ij} = 0$. However, if workers indicate a preference, they get preferable tasks as compared to not indicating preferences.
(c) The tasks in expert crowdsourcing are typically complex and require more time to finish them. We focus on a time-window during which only one such task can be performed. Hence, we make the assumption that each worker takes up one task. If we relax that, all mechanisms can adopt with the claimed properties. Say each worker takes up l tasks:

- For e-Auction , we can still set prices with the same rule and let the worker select up to l most preferred tasks if he is winner at multiple tasks. In fact, the competitive ratio improves with increased competition. For example, if workers can take up up to k tasks, the competitive ratio will be e.
- For, SDV, each instance being static VCG, the strategyproofness will hold true. (The optimization problem will change accordingly).

(d) All the mechanisms are valid even of the number of tasks is not same as the number of workers. However the competitive analysis will change.

7 Conclusion

In this paper, we addressed task assignments to dynamic workers in expert crowdsourcing platforms through matching mechanisms. We introduced the notion of a premium to be paid by the workers to get preferable matches. The monetary transfers help in achieving strategyproofness and efficiency. We proposed two dynamic matching mechanisms, SDV for exogenous workers and e-Auction for endogenous workers. We proved various properties of these mechanisms and in the previous section we summarized our results.

References

1. Babaioff, M., Immorlica, N., Kempe, D., Kleinberg, R.: Online auctions and generalized secretary problems. ACM SIGecom Exchanges 7(2), 7 (2008)
2. Bhat, S., Nath, S., Zoeter, O., Gujar, S., Narahari, Y., Dance, C.: A mechanism to optimally balance cost and quality of labeling tasks outsourced to strategic agents. In: Thirtheenth International Conference on Autonomous Agents and Multiagent Systems. pp. 917–924 (2014)
3. Bigham, J.P., Jayant, C., Ji, H., Little, G., Miller, A., Miller, R.C., Miller, R., Tatarowicz, A., White, B., White, S., et al.: Vizwiz: nearly real-time answers to visual questions. In: Proceedings of the 23nd annual ACM symposium on User interface software and technology. pp. 333–342. ACM (2010)
4. Chen, X., Lin, Q., Zhou, D.: Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13). pp. 64–72 (2013)
5. Difallah, D.E., Demartini, G., Cudré-Mauroux, P.: Pick-a-crowd: tell me what you like, and i'll tell you what to do. In: Proceedings of the 22nd international conference on World Wide Web. pp. 367–374 (2013)

6. Garg, D., Narahari, Y., Gujar, S.: Foundations of mechanism design: A tutorial - part 1: Key concepts and classical results. Sadhana - Indian Academy Proceedings in Engineering Sciences 33(Part 2), 83–130 (April 2008)
7. Garg, D., Narahari, Y., Gujar, S.: Foundations of mechanism design: A tutorial - part 2: Advanced concepts and results. Sadhana - Indian Academy Proceedings in Engineering Sciences 33(Part 2), 131–174 (April 2008)
8. Goel, G., Nikzad, A., Singla, A.: Allocating tasks to workers with matching constraints: truthful mechanisms for crowdsourcing markets. In: Proceedings of the companion publication of the 23rd international conference on World wide web companion. pp. 279–280 (2014)
9. Gujar, S., Faltings, B.: Dynamic task assignments: An online two sided matching approach. In: 3rd International Workshop on Matching Under Preferences, MATCHUP (2015)
10. Gujar, S., Faltings, B.: Online auctions for dynamic assignment: Theory and empirical evaluation. In: ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016). pp. 1035–1043 (2016)
11. Ho, C.J., Jabbari, S., Vaughan, J.W.: Adaptive task assignment for crowdsourced classification. In: Proceedings of the 30th International Conference on Machine Learning (ICML-13). pp. 534–542 (2013)
12. Ho, C.J., Vaughan, J.W.: Online task assignment in crowdsourcing markets. In: AAAI (2012)
13. Howe, J.: The rise of crowdsourcing. Wired magazine 14(6), 1–4 (2006)
14. Huang, E., Zhang, H., Parkes, D.C., Gajos, K.Z., Chen, Y.: Toward automatic task design: a progress report. In: Proceedings of the ACM SIGKDD workshop on human computation. pp. 77–85. ACM (2010)
15. Ipeirotis, P.G., Provost, F., Wang, J.: Quality management on amazon mechanical turk. In: Proceedings of the ACM SIGKDD Workshop on Human Computation. pp. 64–67. HCOMP '10, ACM, New York, NY, USA (2010)
16. Jain, S., Gujar, S., Zoeter, O., Narahari, Y.: A quality assuring multi-armed bandit crowdsourcing mechanism with incentive compatible learning. In: Thirtheenth International Conference on Autonomous Agents and Multiagent Systems. pp. 1609–1610 (2014)
17. Karger, D.R., Oh, S., Shah, D.: Budget-optimal crowdsourcing using low-rank matrix approximations. In: Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on. pp. 284–291. IEEE (2011)
18. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical turk. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 453–456. CHI '08, ACM, New York, NY, USA (2008)
19. Parkes, D.C.: Online mechanisms (2007)
20. Raykar, V.C., Yu, S., Zhao, L.H., Valadez, G.H., Florin, C., Bogoni, L., Moy, L.: Learning from crowds. J. Mach. Learn. Res. 11, 1297–1322 (Aug 2010)
21. Sönmez, T., Ünver, M.U.: Matching, allocation, and exchange of discrete resources. Handbook of Social Economics 1, 781–852 (2011)
22. Tran-Thanh, L., Stein, S., Rogers, A., Jennings, N.R.: Efficient crowdsourcing of unknown experts using multi-armed bandits. In: European Conference on Artificial Intelligence. pp. 768–773 (2012)
23. Zou, J.Y., Gujar, S., Parkes, D.C.: Tolerable manipulability in dynamic assignment without money. In: AAAI (2010)