Truthful Interval Cover Mechanisms for Crowdsourcing Applications

Pankaj Dayama
IBM Research, India
pankajdayama@in.ibm.com

B. Narayanaswamy
Department of CSE, UCSD
muralib@cs.ucsd.edu

Dinesh Garg
IBM Research, India
garg.dinesh@in.ibm.com

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

ABSTRACT

The varying nature of qualities and costs of the crowdworkers makes task allocation a non-trivial problem in almost all crowdsourcing applications. If crowdworkers are strategic about their costs, the problem becomes even more challenging. Interestingly, in several crowdsourcing applications, for example, traffic monitoring, air pollution monitoring, digital epidemiology, smart grids operations, etc., the structure of the tasks in space or time exhibits a natural linear ordering. Motivated by the above observation, we model the problem of task allocation to strategic crowdworkers as an *interval cover mechanism design* problem. In this mechanism, a planner (or task requester) needs to crowdsource labels for a set of tasks in a cost effective manner and make a high quality inference. We consider two different scenarios in this problem: *homogeneous* and *heterogeneous*, based on the qualities of crowdworkers. We show that the task allocation problem is polynomial time solvable in the homogeneous case while it is NP-hard in the heterogeneous case. When the crowdworkers are strategic about their costs, we design truthful mechanisms for both the scenarios. In particular, for the heterogeneous case, we propose a novel approximation algorithm that is monotone, leading to a truthful interval cover mechanism via appropriate payments.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Economics, Theory, Algorithms

Keywords

Crowd Sourcing, Task Allocation, Mechanism Design, Truthful Mechanisms

1. INTRODUCTION

Online crowdsourcing has emerged as an attractive and a cost effective way to access a sizable on-demand workforce for accomplishing large tasks that are difficult to complete otherwise. Tasks for which crowdsourcing has been shown to be useful include labeling large image collections [27], environmental sensing, and searching for objects across a wide geographical area [22]. The success of crowdsourcing can be attributed to many factors, including emergence of low cost and easy to use platforms, ease of access to global workers, and lack of internal expertise and resources for large enterprises. Popular online crowdsourcing platforms include *Amazon Mechanical Turk (AMT), Rent-A-Coder*, *oDesk*, etc. AMT is one of the early crowdsourcing platforms to emerge for *Human Intelligence Tasks (HITs)*. On AMT, a planner (or task requester) posts a set of HITs and crowdworkers (or agents) complete different subsets of these tasks. AMT follows posted price mechanisms, where a planner posts a price that he is willing to pay.

The posted price mechanism is suitable for simple and commodity tasks, where the cost to the crowdworker is quite insignificant – for example, tagging a *CAPTCHA* [1] image with its string. These mechanisms, however, are not ideal for other crowdsourcing applications that involve specialized tasks for which the planner hardly has any idea about crowdworkers' costs and/or qualities, for example, tagging geographical regions with air pollution level or severity level of *Ebola* like disease. For such situations, the planner might be better off using a reverse auction mechanism instead of using a posted price mechanism. In a reverse auction mechanism, crowdworkers are asked to submit bids on their cost and/or quality for completing bundles of tasks and then specific crowdworkers are selected to execute tasks so as to minimize the total cost while ensuring a threshold level of quality.

1.1 Interval Cover Mechanisms

In this paper, we address the problem of designing a reverse auction mechanism for a family of crowdsourcing applications that have certain signature characteristics (discussed below). For reasons that would be clear soon, we call these auction mechanisms as *interval cover mechanisms*. In a subsequent section, we highlight a couple of application scenarios having these characteristics.

Imagine a crowdsourcing application, where

1. Multiple tasks need to be executed and each task involves acquiring a label about some underlying object. The label of each object is binary, say 0 or 1. The true label of each object is unknown to the planner. [2]
2. Discovering the true labels of all the tasks is either prohibitively

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

[1]http://en.wikipedia.org/wiki/CAPTCHA
[2]We use the terms tasks and objects interchangeably but the meaning should be clear from the context.

expensive or time consuming for the planner, making crowdsourcing a natural option. The planner invites crowdworkers to report the labels of chosen subsets of tasks. The crowdworkers can have different qualities. The quality of a crowdworker corresponds to the probability of the worker labeling a task correctly.

3. Unlike human intelligence tasks on Amazon Mechanical Turk, the tasks are such that the execution of these tasks involves significant cost on the part of the crowdworkers. We assume that the planner knows the quality of each crowdworker but is unsure about the true costs of the crowdworkers to execute these tasks.

4. The underlying objects (and hence corresponding tasks) have a natural linear ordering (for example, in either time or space). Therefore, each crowdworker is naturally interested in executing a subset of tasks that are contiguous on this linear scale. We refer to a subset of contiguous tasks an **interval**.

5. The crowdworkers could be strategic in terms of revealing their cost of executing an interval of tasks and also the interval itself.

6. The crowdworkers are *single minded bidders* in the sense that they submit a take-it or leave-it bid for the corresponding interval. This means that a crowdworker would either supply the labels of all the tasks in an interval or nothing.

7. Given the fact that crowdworkers may be of low quality in their labeling of the tasks, the planner provisions to acquire labels from multiple crowdworkers for each task. The planner also wants to achieve a certain threshold on inference quality for each task, so he specifies upfront a certain error tolerance limit for each task.

For any such crowdsourcing application, a compelling option for the planner is to conduct a reverse auction where he invites each crowdworker to submit a bid on any interval of tasks. We call such an auction mechanism as *interval cover mechanism* due to the bid structures. An interval cover mechanism allocates each task to a subset of crowdworkers, depending on their qualities and bids, such that the overall cost is minimized and the error tolerance limit gets satisfied for each task.

For any given set of bids (truthful or otherwise) in an interval cover mechanism, computing a cost minimizing allocation of the tasks to the crowdworkers is in general an NP-Hard problem (Theorem 2). The problem, however, becomes polynomial time solvable if the quality of all the crowdworkers is assumed to be the same (Theorem 1). This motivates us to split our investigations into two parts -*homogeneous scenario* and *heterogeneous scenario* depending on whether we assume the quality of all the crowdworkers to be the same or different.

1.2 Motivating Crowdsourcing Applications

In this section, we provide a couple of important real world applications where interval cover mechanisms make perfect sense.

Mobile Pervasive Sensing: A typical goal of mobile pervasive sensing [4, 9] is to provide real time information on urban processes such as fine grained air quality data, traffic conditions along heavily congested routes (e.g. an arterial road) in that urban area. General wisdom suggests that the cost of sensing here can be reduced significantly by means of crowdsourcing, where daily commuters who travel alongside these routes act as crowdworkers and sense the portion of the arterial road that falls in their daily commute route [13, 28]. If we were to formulate an interval cover mechanism for this application scenario, the intervals would correspond to different contiguous patches on the arterial road, and the bid of a crowdworker would primarily include the cost of installing and maintaining sensors (or applications on smartphones [23]) and using a particular (possibly sub-optimal to her) spatial commute. Each crowdworker bids for an interval of the route that he is willing to sense, and this interval depends on the intersection of the target arterial road with her daily commute route.

Demand Response in Smart Grids: In the context of smart grids, the planner faces a time varying demand profile for electricity. The planner can purchase electricity power from a wide variety of power suppliers. The planner asks these power suppliers to submit their bids for time intervals when they can meet the power demand. One can assume that each power supplier has a certain reliability to meet its promised supply and this reliability score is known to the planner based on past experience [2]. The goal of the planner is to purchase power from a mix of suppliers to meet the demand with a certain tolerance limit (or maximum probability of failure) at each time point. This problem can be modeled as an interval cover mechanism where intervals would correspond to time intervals and power suppliers would correspond to the crowdworkers. The label of a power supplier would correspond to whether it can meet the power demand for a given time point. The goal of the planner is to purchase electricity from a mix of suppliers to meet demand with a certain tolerance limit (or maximum probability of failure) at each time point under planning horizon.

In all of these (and many other) crowdsourcing situations, the goal of the planner is to select winning bids and assign the task of labeling corresponding subsets to the winning crowdworkers so that the total cost of labeling is minimized and error tolerance limit for each task is not violated.

1.3 Summary of Contributions

In this paper, we address the problem of designing interval cover auction mechanisms for crowdsourcing application as described earlier. We split our design problem along two dimensions depending on characteristics of the crowdworkers - (i) strategic versus non-strategic crowdworkers, (ii) homogeneous versus heterogeneous crowdworkers. Homogeneous crowdworkers have the same quality levels, whereas heterogeneous crowdworkers are in different in terms of their quality levels. In what follows, we highlight the design challenges and the novelty of our contributions for each of these four splits.

1. In (non-strategic, homogeneous) scenario, we show that the task allocation problem can be solved optimally in polynomial time, due to totally unimodularity (TUM) property [18] (Section 2.1).

2. For (non-strategic, heterogeneous) scenario, we first derive a bound on minimum number of crowdworkers required to achieve a certain inference quality. These bounds are based on the assumption that for each task the planner aggregates the label by majority voting method. Using these bounds, we formulate an integer programming problem to allocate tasks in a cost minimizing manner. We show that this problem is similar to the problem considered by Chakaravarthy et al. [3] and the 4-approximation factor algorithm proposed by Chakaravarthy et al. [3] can be used for this scenario.

3. In (strategic, homogeneous) scenario, we show that because the non-strategic counterpart of this scenario can be solved in polynomial time, we can use the classical VCG payment scheme [11] to design a truthful mechanism (Section 3.1).

4. We analyze the(strategic, heterogeneous) scenario in Section 3.2. It is known that an allocation algorithm leads to a truthful mechanism if it is monotone [15]. Interestingly, there is no known monotone approximation algorithm for task allocation in this scenario that runs in polynomial time. Motivated by this observation, we propose an approximation algorithm that runs in polynomial time and satisfies the monotonicity property (Section 3.2). We also provide theoretical guarantees on the approximation factor of our algorithm. Our experiments in Section 4 suggest that for most practical scenarios, our algorithm performs better than the algorithm of Chakaravarthy et al. [3] which is the best known (but non-monotone) algorithm in literature. In Section 3.2, we show that our proposed algorithm in conjunction with an appropriate payment rule yields a truthful and ex-post individual rational crowdsourcing mechanism.

We believe this is the first paper to address the problem of designing truthful interval cover auction mechanisms for crowdsourcing applications where the crowdworkers may have heterogeneous qualities. The underlying task allocation problem for this setting turns out to be NP-hard for which we have proposed a (monotone) approximation algorithm with provable approximation guarantees and a truthful mechanism.

A non-trivial generalization of our approach is to consider bids on arbitrary subsets of tasks. We emphasize that we stick to interval mechanisms here to focus on the compelling applications and obtain a firm grounding of the problem.

1.4 Related Work

The prior art on crowdsourcing can be classified into two broad categories - one addressing fundamental research issues, with the other addressing the research challenges encountered in specific application scenarios such as mobile pervasive sensing. Our paper can be classified more in the first category. While there are many fundamental research questions that lie at the core of this area, our work finds relevance to only a few of them - deriving inference out of noisy reports of the crowdworkers, cost optimal selection of crowdworkers, and strategic behavior of crowdworkers.

A vast majority of the literature in crowdsourcing area is dedicated to address the fundamental problem of inference making from noisy reports of crowdworkers. A wide variety of statistical inferencing techniques have been proposed for the same and they differ in terms of their assumptions - see [17, 19, 25] for a comprehensive survey.

Cost optimal allocation of tasks to crowdworkers is the next important issue after the previous one. Tran-Thanh et al. [24] have proposed a task allocation algorithm for crowdsourcing systems with interdependent tasks given a budget constraint. Kamar et al. [7] have focused on consensus tasks and proposed a general system that combines machine learning and decision-theoretic planning to assign tasks to the crowdworkers. For crowdsensing of spatial phenomena, Krause et al. [8] have proposed an scheme for crowdworker selection and fusion of reports from workers. Venanzi et al. [26] learn the error models for crowdworkers' reports of spatial observation. Our work differs from these approaches in the sense that we focus on addressing the computational complexity of the underlying task allocation problem while simultaneously addressing the strategic aspects of crowdworkers.

Incentivizing crowdworkers' participation and truthful reporting has garnered significant importance in recent times. Kamar and Horvitz [6] have focused on consensus tasks and proposed a payment rule that incentivizes crowdworkers to report truthfully in crowdsourcing. Work on peer prediction (Miller et al. [12]) studies the problem of incentivizing truthful reporting when all agents observe the same phenomenon. The cost minimizing task allocation problem, on the other hand, has not been addressed in these paper.

Finally, our work is related to and builds upon the work on efficient auction mechanisms, studied in modern computational economics literature. Of particular relevance is Lehmann et al. [10] which discusses truthful mechanisms when the allocation problem is solved approximately instead of exactly. They identify a set of properties to be satisfied by the approximate allocation algorithm that yield a truthful mechanism. Rothkopf et al. [20] show that combinatorial forward auctions with special structure can often be solved in polynomial time. Archer et al. [1] devise a randomized rounding technique for designing monotone approximation algorithm for combinatorial auctions with single parameter agents. It then gives a mechanism that is truthful with high probability and in expectation. Our setting is different from those addressed above since qualities are also involved. We give a monotone approximation algorithm for task allocation in our setting and design a dominant strategy incentive compatible mechanism.

1.5 Notation

To formalize the problem, we introduce some notation first. Let $T = \{1, 2, \ldots, m\}$ denote the ordered set of labeling tasks. Let $A = \{1, 2, \ldots, N\}$ denote the set of crowdworkers, where each crowdworker is capable of, and interested in, solving an interval of tasks. A bid submitted by crowdworker i is of the form $([s_i, e_i], b_i)$, where the interval $[s_i, e_i]$ corresponds to the subset of tasks $\{s_i, s_i+1, \ldots, e_i\}$ and b_i is crowdworker's reported cost for executing the tasks in this interval. When there is no confusion, we denote the bid simply by b_i. Let p_i denote the quality of the labels provided by crowdworker i. Recall that the quality of a crowdworker corresponds to the probability that the label reported by him is the correct label. We assume the quality information for all the crowdworkers in the set A to be known to the planner. Let ε_j denote the error tolerance limit for the final label inferred by the planner for the task $j \in T$. By error tolerance limit, we mean that the probability of the planner's inferred label for task j being wrong should be no more than ε_j.

2. NON-STRATEGIC SETTING

Often, in crowdsourcing scenarios, crowdworkers are motivated by altruistic motives rather than monetary rewards and hence, bid their costs truthfully for labeling the interval that interests them. This setting is referred to as *non-strategic setting*. We consider two different scenarios under this setting, based on worker quality, namely *homogeneous and heterogeneous*.

2.1 Homogeneous Scenario

In this scenario, all the crowdworkers have the same quality, i.e., $p_i = p, \forall i \in A$. In order to be within the required error tolerance limit, the planner would require to allocate each task $j \in T$ to multiple, say n_j, crowdworkers. In order to infer a final label for each task, assume that the planner aggregates the reported labels from multiple crowdworkers based on the *majority voting rule*. In such a situation, a task $j \in T$ gets labeled incorrectly if $\lceil n_j/2 \rceil$ or more reports (out of n_j) for the task are wrong. Note that the sum of n_j reports is a Binomial random variable with mean p. The probability that this sum is less than or equal to $\lfloor n_j/2 \rfloor$ is given by

$$P_{err} = \sum\nolimits_{k=0}^{\lfloor n_j/2 \rfloor} \binom{n_j}{k} p^k (1-p)^{n_j-k} \qquad (1)$$

Using Chernoff-Hoeffding bound [14], we get

$$P_{err} \leq \exp\left(-n_j\, \mathcal{D}(1/2 \parallel p)\right) \tag{2}$$

where $\mathcal{D}(\cdot \parallel \cdot)$ is the KL divergence function. Using expression (2), the planner can determine the minimum number of crowdworkers n_j that he needs to select for task j.

For this scenario, the task allocation problem for the planner becomes an integer linear program (ILP) as given below.

$$\begin{aligned} \text{Minimize} \quad & \textstyle\sum_{i \in A} b_i x_i \\ \text{subject to} \quad & \sum_{i \in A | s_i \leq j \leq e_i} x_i \geq n_j, \quad \forall j \in T \\ & x_i \in \{0,1\}, \quad \forall i \in A \end{aligned} \tag{3}$$

Here, the binary variable x_i is equal to 1 when crowdworker i is assigned his interval. Constraint (3) ensures that task j is allocated to at least n_j crowdworkers. We assume that the planner has access to a large pool of crowdworkers so that above optimization problem always has a feasible solution.

Note that constraint (3) can be written in matrix form as $\boldsymbol{M}x \geq \boldsymbol{n}$, where $\boldsymbol{n} = [n_1, \ldots, n_m]^\intercal$, $\boldsymbol{M} = [M_{ij}]$, and $M_{ij} = 1$ if task j is part of crowdworker i's interval, and is 0 otherwise. The matrix $\boldsymbol{M}$ has a property that in each column, the 1's appear in a contiguous sequence, i.e., $\boldsymbol{M}$ satisfies the consecutive-ones property and hence $\boldsymbol{M}$ is a totally unimodular matrix (TUM). Because of TUM property of matrix $\boldsymbol{M}$, the LP relaxation of the above formulation yields an optimal integer solution [18], and hence the following theorem.

THEOREM 1. *Task allocation in the homogeneous scenario can be optimally solved in polynomial time.*

2.2 Heterogeneous Scenario

Here, we assume that crowdworker i has quality p_i and as usual, task $j \in T$ has a error tolerance limit ε_j. The task allocation problem for this scenario is also an ILP - similar to the one shown in the homogeneous case - except that we cannot use the quality constraint (3) any more and instead, we need to develop its counterpart for the current scenario.

Consider a task j having error tolerance limit ε_j. Let Z_i^j be an indicator random variable representing whether the reported label from crowdworker i who is allocated task j is correct. Thus, $P(Z_i^j = 1) = p_i$ and $P(Z_i^j = 0) = (1 - p_i)$. Let $p_{min} = \min\{p_1, \ldots, p_n\}$ and $p_{max} = \max\{p_1, \ldots, p_n\}$. Let A_j be the set of crowdworkers allocated for task j. Let $n_j = |A_j|$. Let $Z = \sum_{i \in A_j} Z_i^j$. Then, the probability that the label inferred for task j is incorrect, while using majority voting rule, is: $P(Z < n_j/2)$. We want to upper bound this quantity by ε_j. We use the multiplicative form of Chernoff bound (see e.g. Theorem 4.2 in [14]), which says

$$P[Z \leq (1-\nu)\mu] \leq \exp\left(-\mu\nu^2/2\right)$$

where, $\mu = E[Z] = \sum_{i \in A_j} p_i$, and $0 < \nu < 1$. By substituting

$$\nu = \frac{1}{2\mu} \sum_{i \in A_j} (2p_i - 1)$$

we get the following bound

$$P(Z < n_j/2) \quad \leq \quad \exp\left(-\frac{1}{8\mu} \sum_{i \in A_j} (2p_i - 1)^2\right)$$

For $p_i \geq 2/3$, we can simplify this bound as follows:

$$P(Z < n_j/2) \leq \exp\left(-\textstyle\sum_{i \in A_j} (2p_i - 1)/8\right)$$

Upper bounding the above inequality by ϵ_j, we get

$$\textstyle\sum_{i \in A_j} (2p_i - 1) \geq 8 \log(1/\epsilon_j)$$

Now, we make use of this inequality to write down the task allocation ILP for the heterogeneous scenario as follows.

$$\begin{aligned} \text{Minimize} \quad & \textstyle\sum_{i \in A} b_i x_i \\ \text{subject to} \quad & \sum_{i \in A | s_i \leq j \leq e_i} r_i x_i \geq d_j, \ \forall j \in T \\ & x_i \in \{0,1\}, \quad \forall i \in A \end{aligned} \tag{4}$$

where, $r_i = (2p_i - 1)/(2p_{min} - 1)$, and $d_j = 8 \log(1/\epsilon_j)/(2p_{min} - 1)$. We refer to r_i as the contribution of crowdworker i towards task j and d_j as the total demand of task j. Further, we define $r_{max} = \max_{i \in A} r_i$. Note, in the above formulation, we have divided both the sides of the constraint by $(2p_{min} - 1)$. This does not alter the ILP in any way but it helps us make a crucial observation (stated in Lemma 1) which is used in a subsequent section for the development of the proposed algorithm.

LEMMA 1 (DEMAND COVERING (DC) LEMMA). *In the heterogeneous scenario, every time the planner recruits a new worker for task $j \in T$, the corresponding demand d_j appearing in the constraint (4) gets reduced by at least 1 unit and at most by r_{max} units.*

PROOF. This follows by observing that $1 \leq r_i \leq r_{max}\ \forall i \in A$ as defined above. □

Note, unlike previous scenario, that the ILP (4) cannot be solved in polynomial time as stated below.

THEOREM 2. *The task allocation problem for heterogeneous scenario is NP-hard even when $|T| = 1$.*

PROOF. We prove it by showing a reduction from a restricted Knapsack problem where the value of each item is bounded by 1. Consider a finite set of items $I = \{1, 2, \ldots, N\}$. Let b_i and $r_i \leq 1$ denote the weight and the value, respectively, for the item $i \in I$. Given two integers d and U, it is NP-hard to decide whether there is a subset $\tilde{I} \subseteq I$ such that $\sum_{i \in \tilde{I}} r_i \geq d$ and $\sum_{i \in \tilde{I}} b_i \leq U$ [5]. Note that the general knapsack problem can be reduced to the above restricted knapsack problem by appropriate scaling of values of the items and the knapsack size.

In our setup, we consider a special case where we have N crowdworkers and only one task. Let b_i be the bid submitted by crowdworker i for solving this task and let $(2p_i - 1)$ be the weight assigned to the crowdworker i. Let $d = 8\log(1/\epsilon)$ be the demand to cover the task. In view of the above ILP, it is easy to see that the instance of the restricted Knapsack has a solution if and only if there is a solution of cost at most U for this specific instance of our problem. □

We also make an observation that above *heterogeneous scenario* is equivalent to $(0,1)$-RESALL problem considered in [3]. The $(0,1)$-RESALL problem specifies time-varying demand for discrete time-slots and a set of resources, each available for a certain duration. The goal of $(0,1)$-RESALL problem is to decide a minimum cost allocation of resources to satisfy the demand where each resource has an associated capacity and cost. Our problem under heterogeneous scenario can be mapped to the $(0,1)$-RESALL problem in following manner. Tasks in our scenario map to the time slots, efforts required $D = \{d_1, d_2, \ldots, d_m\}$ for the tasks map to the demand profile d, and bids' intervals from crowdworkers map to the set of resources.

Given this mapping, we can always use the primal-dual approximate algorithm, proposed by [3] for solving the $(0,1)$-RESALL problem, to solve our problem as well under the heterogeneous scenario. The primal dual algorithm of [3] relies on the notion of ILP with flow-cover inequalities. Such an ILP is like an alternative formulation of our original ILP. The flow-cover inequality based primal-dual ILP of our problem are given in Appendix. We have also customized the primal-dual approximate algorithm of [3] for our scenario and have presented the details of this algorithm in Appendix. The result below directly follows from [3].

PROPOSITION 1. *There exists a polynomial time 4-factor approximation algorithm for solving the task allocation problem in the heterogeneous scenario.*

3. STRATEGIC SETTING

In this section, we consider the strategic setting where crowdworkers need not be truthful in terms of reporting their costs and/or the interval. We address the problem of the planner who needs to design a task allocation rule and a payment rule such that (i) he allocates crowdworkers to the tasks in an *allocatively efficient* manner, (ii) crowdworkers are *truthful*, and (iii) he is *individually rational*. Allocative efficiency here would mean an *allocation for which the sum of the bids of the winning crowdworkers is minimum.* We call these mechanisms as *interval cover mechanisms*. When we refer to truthfulness, we mean *Dominant Strategy Incentive Compatibility (DSIC)*. That is, bidding truthfully is a weakly dominant strategy for every crowdworker.

3.1 Homogeneous Scenario

As per Theorem 1, an efficient allocation can be determined in polynomial time for this scenario. We can use this algorithm to design VCG (Vickrey-Clarke-Groves) mechanism [11]. In fact, the VCG mechanism here would satisfy all the desired properties, that is, allocative efficiency, truthfulness, and individual rationality. As per the VCG payment rule, the planner pays marginal contribution $C^*(A \setminus \{i\}) - C^*(A)$ to every crowdworker i, where $C^*(A)$ is the sum of winning bids in efficient allocation when set of crowdworkers is A. Note that if crowdworker i is a winner then we will have $C^*(A) \leq C^*(A \setminus \{i\})$, otherwise we will have $C^*(A) = C^*(A \setminus \{i\})$. This means the planner makes a non-negative payment to the winning crowdworkers and zero payment to rest of the crowdworkers. This shows the individual rationality of this mechanism.

3.2 Heterogeneous Scenario

Now, we propose a truthful mechanism for the *heterogeneous* scenario. Here, it is not practical to appeal to a VCG mechanism because the allocation problem is NP-hard (Theorem 2) and so computation of the VCG payments could possibly involve solving as many NP-hard problems as the number of winners plus one. To overcome this, we need to design an allocation scheme which satisfies monotonocity property as defined below.

DEFINITION 1 (MONOTONICITY). *If a crowdworker i gets allocated a task when he submits a bid $([s_i, e_i], b_i)$, then he will also get allocated if he submits a bid $([\tilde{s}_i, \tilde{e}_i], \tilde{b}_i)$, for any $\tilde{b}_i \leq b_i$ and $[\tilde{s}_i, \tilde{e}_i] \supseteq [s_i, e_i]$, provided the bids of the other crowdworkers remain the same.*

It is easy to see that the Algorithm 3 (given in Appendix) does not satisfy the monotonicity property due to delete phase of the algorithm. The problem of designing a truthful mechanism for the heterogeneous scenario critically depends on designing monotone allocation algorithm.

In what follows, we propose a monotone approximation algorithm (Algorithm 1) to solve the task allocation problem and design a truthful mechanism using this algorithm. This is an iterative algorithm, where we keep checking residual demand vector $\hat{\boldsymbol{d}}_t$ in each iteration and continue iterations until $\hat{\boldsymbol{d}}_t$ hits zero at which point our solution in hand, namely S_t, would have satisfied the quality constraint (as given by the constraint in ILP (4)). In any iteration, if there is non-zero residual demand for at least one task, we call the subroutine *Single-Unit Demand Cover()* and supply the current residual demand vector to this routine. This subroutine looks at the current residual demand vector $\hat{\boldsymbol{d}}_t$ and returns an optimal cost solution for covering at least one unit of demand for each task j that has a non-zero residual demand in $\hat{\boldsymbol{d}}_t$. This subroutine is like a dynamic programming algorithm to find a solution for interval cover. After making a call to this subroutine, we update the residual demand for each task and also the set of unallocated crowdworkers. We repeat this procedure until all the demand d_j for every task j gets met.

3.2.1 Analysis of Algorithm 1

Let S^* be the optimal solution to the ILP (4). Let $S^*(1) \subseteq S^*$ be the cost minimizing set of crowdworkers such that such that each non-zero demand task is covered by atleast by one crowdworker in this set. Let $C(S^*(1))$ be the total cost of set $S^*(1)$. Let $d^*_j(1)$ be the demand covered by $S^*(1)$ for task $j \in T$.

LEMMA 2. *$d^*_1(1) \leq r_{max}, d^*_m(1) \leq r_{max}$, and $d^*_j(1) \leq 2r_{max}$; $\forall j \in \{2, 3, \ldots, m-1\}$, where $r_{max} = (2p_{max} - 1)/(2p_{min} - 1)$.*

PROOF. We make two key observations.

- *No two crowdworkers in $S^*(1)$ have the same starting point or end point.* To prove this, let us suppose two crowdworkers $i, \hat{i} \in S^*(1)$ are such that, $s_i = s_{\hat{i}}$. WLOG assume $e_i \leq e_{\hat{i}}$. Then removing crowdworker i from set $S^*(1)$ will reduce the cost of the cover. Hence, we have a contradiction.

- *For any task j, at most two crowdworkers covering it can be part of $S^*(1)$.* To prove this, let us suppose there are three crowdworkers covering task j. It is easy to see that just by keeping at most two crowdworkers (ones with earliest starting id and last ending id) one can reduce the cost of cover. Hence we have a contradiction.

Thus the start and end tasks will get covered by only one crowdworker and other intermediate tasks can get covered by at most two bids in the set $S^*(1)$. Also, maximum contribution by any single bid is r_{max}. This proves the lemma. □

LEMMA 3. *Single-Unit Demand Cover() function in Algorithm 1 returns an optimal subset of crowdworkers in the set U to cover all the tasks with non-zero demand at least once. Further, the time complexity of this step is $\mathcal{O}(m^2)$.*

PROOF. The running time can be derived in a straightforward way. To prove the lemma, we claim that Single-Unit Demand Cover() maintains following invariant - *at the end of every iteration j, the set $\tilde{S}_j$ is an optimal set of crowdworkers within U satisfying following two properties (i) it covers each task in the interval $[1, j]$ having non-zero demand at least once, (ii) the bid interval of every member in $\tilde{S}_j$ is a subset of $[i, j]$.* This invariance directly proves the lemma. We prove this invariance by induction as follows.

- **Base Case:** For $j = 1$, the invariance holds true trivially due to initialization step.

ALGORITHM 1: Monotone task allocation algorithm for heterogeneous scenario

```
1  Algorithm Main()
     Input :
       $b_i, i \in A$: bid submitted by crowdworker $i$ (includes both
       cost and interval of interest);
       $\boldsymbol{d} = [d_1, \ldots, d_m]$: Total demand vector where $d_j$ is the
       demand to be covered for task $j$;
       $r_i$: Contribution of crowdworker $i$ towards each task that lies
       in his bid interval
     Output :
       $S_t$: Set of crowdworkers allocated after iteration $t$;
       $C(S_t)$: Total cost of set $S_t$
     Initialize: $t \leftarrow 0$; $S_t \leftarrow \emptyset$; Residual demand vector after
         iteration $t := \hat{\boldsymbol{d}}_t \leftarrow \boldsymbol{d}$; Unallocated crowdworker set
         $:= U_t \leftarrow A$
2    while $\hat{\boldsymbol{d}}_t > 0$ do
3      $t \leftarrow t + 1$
4      $S_t \leftarrow$ Single-Unit Demand Cover($\hat{\boldsymbol{d}}_{t-1}, U_{t-1}$)
5      Update $U_t$ and $\hat{\boldsymbol{d}}_t$
6    end
1  Procedure Single-Unit Demand Cover($\hat{\boldsymbol{d}}, U$)
     Initialize: For each task, $l, j$ such that $l \leq j$, define
       $u_{lj}$ := A crowdworker in $U$ whose bid value is minimum
       across all the crowdworkers in $U$ having bid interval as $[l, j]$
       (if required use dummy crowdworker $D$ having sufficiently
       high bid value);
       $C(u_{lj}) \leftarrow$ Bid of crowdworker $u_{lj}$;
       If $\hat{d}_z = 0 \; \forall l \leq z \leq j$ then $u_{lj} \leftarrow \emptyset, C(u_{lj}) \leftarrow 0$;
       $\tilde{S}_j$ := Set of crowdworkers in $U$ that cover each of the tasks
       $\{1, \ldots, j\}$ at least once;
       $\tilde{S}_j \leftarrow u_{1j},\ C(\tilde{S}_j) \leftarrow C(u_{1j}) \forall j \in T$
2    for $j = 2 \rightarrow m$ do
3      for $l = 2 \rightarrow j$ do
4        if $C(\tilde{S}_{l-1}) + C(u_{lj}) \leq C(\tilde{S}_j)$ then
5          Set $\tilde{S}_j \leftarrow \tilde{S}_{l-1} \bigcup \{u_{lj}\}$
6        end
7      end
8      for $k = 1 \rightarrow (j - 1)$ do
9        if $C(\tilde{S}_k) >= C(\tilde{S}_j)$ then
10         Set $\tilde{S}_k \leftarrow \tilde{S}_j$
11       end
12     end
13   end
14   return $\tilde{S}_m$
```

- **Induction Hypothesis:** Let us assume that for some $\hat{j} < m$ the invariance holds true.

- **Induction Step:** Assume $j = (\hat{j} + 1)$ and we show that invariance holds true for this j. Note, in the initialization step, we find the crowdworker with minimum bid for interval $[1, j]$. WLOG, let us analyze any intermediate iteration where say $l = 5$ and $\hat{j} > l$. In step 4, we compare the total cost of optimal cover till $(l - 1)$ and the lowest bid for interval $[l, j]$, with optimal cover cost computed till previous iteration for the interval $[1, j]$. We choose the minimum of these two values to finally get best cover for the interval $[1, j]$ tasks till this point. At the end of execution of *for loop* over l, we will obtain optimal allocation $\tilde{S}_j$ and cost $C(\tilde{S}_j)$. The *for loop* over k, updates the cost of the subintervals $[1, k], k < j$, computed till now, in case it happens to be higher than the recently computed cost for the interval $[1, j]$. This will ensure that at the end of iteration j, the cost of any such subinterval is optimal over all crowdworkers in U whose bids lies within interval $[1, j]$.

This proves the invariance and hence the lemma. □

LEMMA 4. *Algorithm 1 produces an allocation that is monotone.*

PROOF. Consider a crowdworker i and fix the bids from all other crowdworkers. Suppose crowdworker i's bid is $([s_i, e_i], b_i)$ and he gets allocated in iteration t. Let us now consider the allocation when crowdworker i reduces his bid to $([\tilde{s}_i, \tilde{e}_i], \tilde{b}_i)$. If the crowdworker i gets allocated in iteration $t' < t$ then we are done. Let us assume the worst case where he does not get allocated till iteration t. So nothing changes till iteration $t - 1$.

In iteration t, as before, the residual demand profile and unallocated bids are the same. By optimality of *single-unit demand cover* routine in Algorithm 1 (Lemma 3), crowdworker i will get allocated when his bid is $\tilde{b}_i$. We have shown that the allocation produced satisfies monotonicity. □

THEOREM 3. *Algorithm 1 is a polynomial time $2r_{max}$ factor monotone approximation algorithm for solving the task allocation problem in the heterogeneous scenario.*

PROOF. Let $S^* \subseteq A$ be the optimal solution to the ILP (4). Set $S^*_1 \leftarrow S^*$ and $A_1 \leftarrow A$. Let $S^*(1) \subseteq S^*_1$ be the optimal set of crowdworkers that covers all the non-zero demand tasks at least once and $C(S^*(1))$ the corresponding cost.

Note, in view of Lemma 1 and Lemma 2, by making at most $2r_{max}$ successive calls to the subroutine *Single-Unit Demand Cover()* starting from set A we would get a solution $S_{2r_{max}}$ that covers as much demand for every task as covered by the solution $S^*(1)$. Let $S_{2r_{max}} = Y_1 \cup Y_2 \ldots \cup Y_{2r_{max}}$ where Y_k is the solution set returned by k^{th} call to the subroutine. Let $Y_k = X_k \cup \hat{X}_k$ be the set of crowdworkers selected in iteration k and $C(Y_k)$ be the corresponding cost, where $Y_k \cap S^*(1) = X_k$. At the end of iteration k, the demand yet to be covered will be at most the demand covered by set $S^*(1) \setminus (X_1 \cup X_2 \ldots \cup X_k)$. By Lemma 3, optimality of our algorithm at every step, we have, $C(Y_{k+1}) \leq C(S^*(1) \setminus (X_1 \cup X_2 \ldots \cup X_k)) \leq C(S^*(1))$. Thus, $C(S_{2r_{max}}) \leq 2r_{max} C(S^*(1))$. If $S_{2r_{max}}$ covers the full demand covered by S^* then we stop. If not, we remove set $S_{2r_{max}}$ from S^*_1 and A_1 to get S^*_2 and A_2 and repeat this procedure. Note that at any point we are within $2r_{max}$ of the optimal cost. Hence the claim.

Further, the time complexity of this algorithm is $\mathcal{O}(m^2 d_{max} + n \log n)$ where $d_{max} = \max\limits_{1 \leq j \leq m} d_j$. □

For this scenario, note that we have proposed a tractable, monotone algorithm for task allocation. This algorithm outputs an approximately efficient allocation. In what follows, we suggest a payment rule, which when coupled with the allocation given by Algorithm 2 provides a truthful mechanism which is $2r_{max}$-approximate allocative efficient and is also ex-post individually rational.

DEFINITION 2 (CRITICAL PAYMENT). *The critical value of a winning crowdworker is the maximum bid which he could have reported and still be allocated his interval while the bids of all other crowdworkers are kept fixed.*

The computation of the critical value depends on the allocation rule and in general it is hard to obtain a closed form expression unless the allocation rule is simple enough. Mu'Alem and Nisan [16] suggested a naive approach to compute critical value for any kind of allocation rule using binary search. We compute the critical bid value for crowdworker i, using binary search, by repeatedly running the allocation rule with his bids varying in the interval of current bid value and some appropriate high value.

```
ALGORITHM 2: A $2r_{max}$-approximate truthful mechanism
1 Allocate the tasks to the crowdworkers as per Algorithm 1.
2 Crowdworker $i$ gets payment of zero if no task is allocated to
  him
3 If crowdworker $i$ is allocated his interval, he is rewarded with
  his critical value (as per Definition 2)
```

THEOREM 4. *The interval cover mechanism given by Algorithm 2 for heterogeneous scenario is $2r_{max}$-approximately allocative efficient, truthful, and ex-post individual rational.*

PROOF. Note that the approximation factor of $2r_{max}$ follows from the fact that our allocation is given by Algorithm 1. Individual rationality is trivial by design of the mechanism itself. We just need to show truthfulness. For this, we appeal to Theorem 4 in [10], which says that if a mechanism satisfies *monotone allocation, exactness, ex-post individual rationality and critical payments* then it would be truthful. Ours is a reverse auction setting but similar claims and proofs hold good with slight modifications as shown below. Exactness here means that a crowdworker never gets allocated his interval partially. In Algorithm 1, it is clear that the interval for a crowdworker i is either completely allocated or not allocated. Thus, the algorithm satisfies *exactness* property. The critical payment property comes from the design of the payment rule for the mechanism. By Lemma 4, we know that the mechanism satisfies monotonicity.

LEMMA 5. *In a mechanism that satisfies exactness, monotonicity, ex-post individual rationality, and critical payments, a crowdworker i having type $([s_i, e_i], b_i)$ is never better off reporting $([s_i, e_i], \hat{b}_i)$ where $\hat{b}_i \neq b_i$.*

PROOF. By critical payment and ex-post individual rationality property, utility of any crowdworker who reports his true type is always non-negative. If i does not get allocated when he misreports his type then his utility is zero. Thus, in this case, he is better-off reporting his true type. Let us look at another scenario when i gets allocated in both cases. His valuation and payment is the same (critical payment) in both the cases. So the claim still holds. Last scenario is when i is not allocated by reporting true type but only when he misreports. This implies that $\hat{b}_i < \hat{p}_i^c < b_i$. Thus his utility is non-positive. Hence the claim. □

LEMMA 6. *In a mechanism that satisfies exactness, monotonicity, and critical payments, a crowdworker i having type $([s_i, e_i], b_i)$ whose bid is allocated gets a payment p_i^c that is at least the price $\hat{p}_i^c$ that he would get as payment if he had reported type $([\hat{s}_i, \hat{e}_i], b_i)$ for any $[\hat{s}_i, \hat{e}_i] \subseteq [s_i, e_i]$.*

PROOF. By monotonicity $([\hat{s}_i, \hat{e}_i], b_i)$ would have been allocated and by critical payment let $\hat{p}_i^c$ be paid to i such that: for any $x > \hat{p}_i^c$ the bid $([\hat{s}_i, \hat{e}_i], x)$ would not have been granted. By monotonicity, for any such x the bid $([s_i, e_i], x)$ would not have been granted. By critical Payment, for any x such that $x < p_i^c$, the bid $([s_i, e_i], x)$ would have been granted. This implies that $p_i^c \leq \hat{p}_i^c$. □

LEMMA 7. *If a mechanism satisfies exactness, monotonicity, ex-post individual rationality, and critical payments then it is truthful.*

PROOF. Suppose i's true type is $([s_i, e_i], b_i)$. We need to prove that he will be never interested in misreporting his type as $([\hat{s}_i, \hat{e}_i], \hat{b}_i)$. Suppose i misreports his type as $([\hat{s}_i, \hat{e}_i], \hat{b}_i)$. Assume that $[s_i, e_i] \subset [\hat{s}_i, \hat{e}_i]$ or $[s_i, e_i] \not\subseteq [\hat{s}_i, \hat{e}_i]$. In this case if i gets allocated then his valuation is $-\infty$. But his payment is bounded. So his utility will be negative. If he is not allocated then his utility is 0. So he is better off reporting his true type. Assume then $[\hat{s}_i, \hat{e}_i] \subseteq [s_i, e_i]$. Since i's cost for exploring $[\hat{s}_i, \hat{e}_i]$ is same as that for $[s_i, e_i]$, by Lemma 6 i would be better off bidding $([s_i, e_i], \hat{b}_i)$. Lemma 5 implies that bidding $([s_i, e_i], \hat{b}_i)$ cannot be better off than being truthful. □

■

4. SIMULATION EXPERIMENTS

In this section, we evaluate the performance of our proposed interval cover mechanism. We use the following metrics: *empirical approximation factor* for the task allocation algorithm and *overpayment factor* for the payment rule. Empirical approximation factor of an algorithm is the ratio of the approximate solution computed by the algorithm to the optimal solution. Overpayment factor is ratio of the total payment (to the winning crowdworkers) to the social cost where social cost is the total cost of all the winning crowdworkers. It is an indicator of the cost paid by planner to ensure truthful bids from the strategic crowdworkers.

The default values of various parameters used for simulations are given in Table 1. The bids of crowdworkers are generated based on i.i.d samples from two distributions: normal distribution (NORM) and uniform distribution (UNIF).

Table 1: Simulation Parameter Settings

Parameter name	Value
Number of crowdworkers, N	$[400, 2000]$
Number of tasks, m	$[5, 20]$
Crowdworker quality, p_i	$(0.5, 0.99]$
Mean bid value, $\mathbb{E}[b_i]$	100
Task error tolerance limit, ε_j	$[0.05, 0.5)$

4.1 Evaluation of Proposed Allocation Algorithm 1

In the previous sections, we have given theoretical upper-bounds for the task allocation algorithm for the heterogeneous scenario. Now, we also wish to examine its empirical performance. We experimentally compare the performance of our proposed algorithm 1 to two baselines. The first baseline is the primal-dual (PD) Algorithm 3 which is the best known algorithm in the literature for our scenario. The second baseline is a monotone version of the PD algorithm obtained by dropping the reverse delete phase in the algorithm. We compare the empirical approximation factor of our proposed algorithm with these two baselines.

In the simulations, we draw crowdworker qualities from a uniform discrete distribution on $[p_{min}, p_{max}]$. The values of p_{min} and p_{max} will decide the theoretical bound on the approximation factor. All the other parameters are set as per Table 1. We solve the task allocation problem optimally using the commercial software *ILOG CPLEX Optimization Studio*. We run our proposed algorithm and the two baseline algorithms to compute approximate solutions to the task allocation problem. Empirical approximation factors for all the three algorithms are plotted in Figure 1. Each data point in the plot is the average of 20 independent runs under the same parameter setting. The above procedure is repeated, for different theoretical bound on the approximation factor, by varying p_{min} and p_{max}.

Figure 1 clearly shows that in most cases our proposed algorithm performs much better than the two baseline algorithms. Also, we

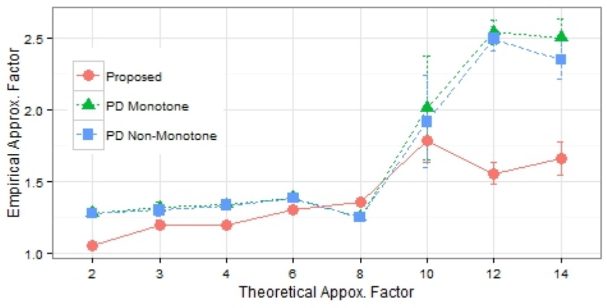

Figure 1: Comparison of performance of our proposed algorithm against benchmark algorithms

get much better empirical performance as compared to the theoretical bounds.

4.2 Evaluation of Payment Rule

In our experiments, we have considered a competitive setting where $N \gg m$, which generally holds true for crowdsourcing applications. Figure 2 plots the overpayment factor when the number of crowdworkers changes from 400 to 2000 and number of tasks varies from 5 to 20. The overpayment factor is always below 1.06 for both the distributions we have considered. We also observe that the overpayment factor remains small even when the number of tasks, m, is increased. This highlights an attractive aspect of our mechanism in ensuring truthfulness with very low levels of additional payments. These results are consistent with the quantitative bounds shown in [21] for multi-item auctions.

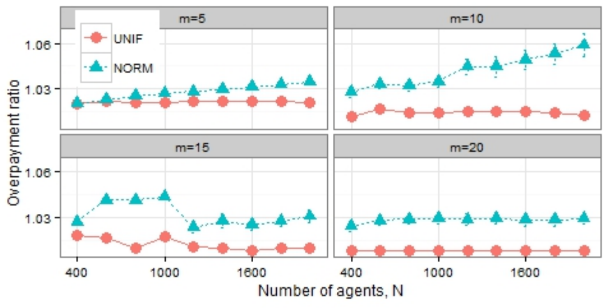

Figure 2: Overpayment factor for different cost functions

5. CONCLUSIONS AND FUTURE WORK

In this work, motivated by pervasive sensing and smart grids applications, we introduced the problem of allocating linearly ordered tasks to crowdworkers bidding for intervals of contiguous tasks. We provided tractable solutions for two different scenarios, homogeneous and heterogeneous, which are defined based on the crowdworkers qualities. Our work also investigated the strategic version of this problem where the crowdworkers have their costs and intervals as private information. We proposed truthful and individually rational mechanisms for both the scenarios. In particular, for the heterogeneous scenario, we proposed a novel approximation algorithm that is monotone, leading to a truthful interval cover mechanism via appropriate payments.

The linear ordering of the tasks, so natural to the discussed applications, comes in handy to ground the problem leading to valuable insights. Our approach here will facilitate a better understanding of the general case of combinatorial bids and this is an immediate direction for future work. The work can be also be extended to incorporate crowdworkers having different qualities for different tasks.

APPENDIX

A Primal Dual Algorithm for Heterogeneous Scenario

By making use of flow-cover inequality, the ILP (4) can be written in an alternative manner as follows.

$$\begin{aligned} \text{Minimize} \quad & \sum_{i \in A} b_i x_i \\ \text{subject to} \quad & \\ & \sum_{i \in A | s_i \le j \le e_i; i \notin S} r(i,S) x_i \ge d_j(S), \qquad (5) \\ & \qquad \forall j \in T, S \subseteq A \\ & x_i \in \{0,1\}, \quad \forall i \in A \end{aligned}$$

where, $S \subseteq A$ can be viewed as a set of crowdworkers as if they are already chosen. $d_j(S) = d_j - \sum_{i \in S | s_i \le j \le e_i} r_i$ is the residual weight to be covered and $d(i,S) = \min\{r_i, d_j(S)\}$. The dual of the LP-relaxation of the above formulation is as follows.

$$\begin{aligned} \text{Maximize} \quad & \sum_{(S,j)} d_j(S) y(S,j) \\ \text{subject to} \quad & \\ & \sum_{(S,j): s_i \le j \le e_i; i \notin S} r(i,S) y(S,j) \le b_i, \ \forall i \in A \\ & y(S,j) \ge 0, \quad \forall (S,j) : S \subseteq A \end{aligned}$$

ALGORITHM 3: Primal-dual task allocation approximation algorithm for the *heterogeneous scenario*

```
1  $S_0 \leftarrow \emptyset$
2  $t \leftarrow 0$
3  **Forward Phase:**
4      **repeat**
5          $t \leftarrow t+1$;
6          $j_t \leftarrow$ Task $j$ having maximum value of $d_j(S_{t-1})$;
7          Increase dual variable $y(S_{t-1}, j_t)$ till some dual
           constraint becomes tight;
8          Let $i_t$ be the primal variable corresponding to this
           constraint;
9          Set $x_{i_t} = 1$;
10         Set $S_t = S_{t-1} \cup \{i_t\}$;
11     **until** $\max_{j \in T} d_j(S_{t-1}) = 0$;
12 **Reverse Delete Phase:**
13     Consider the crowdworkers in reverse order from set $S_t$
14     Delete a crowdworker if it retains the primal feasibility
15 Return minimum feasible solution after appropriate deletion
```

References

[1] Aaron Archer, Christos Papadimitriou, Kunal Talwar, and Éva Tardos. “An approximate truthful mechanism for combinatorial auctions with single parameter agents”. In: *Internet Mathematics* 1.2 (2004), pp. 129–150.

[2] Eilyan Y Bitar, Ram Rajagopal, Pramod P Khargonekar, Kameshwar Poolla, and Pravin Varaiya. “Bringing wind energy to market”. In: *IEEE Transactions on Power Systems* 27.3 (2012), pp. 1225–1235.

[3] Venkatesan T Chakaravarthy, Amit Kumar, Sambuddha Roy, and Yogish Sabharwal. “Resource allocation for covering time varying demands”. In: *Algorithms–ESA 2011*. Springer, 2011, pp. 543–554.

[4] Raghu K Ganti, Fan Ye, and Hui Lei. “Mobile crowdsensing: current state and future challenges”. In: *Communications Magazine, IEEE* 49.11 (2011), pp. 32–39.

[5] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[6] Ece Kamar and Eric Horvitz. “Incentives for truthful reporting in crowdsourcing”. In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems. 2012, pp. 1329–1330.

[7] Ece Kamar, Severin Hacker, and Eric Horvitz. “Combining human and machine intelligence in large-scale crowdsourcing”. In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems. 2012, pp. 467–474.

[8] Andreas Krause, Eric Horvitz, Aman Kansal, and Feng Zhao. “Toward community sensing”. In: *Proceedings of the 7th International Conference on Information Processing in Sensor Networks*. IEEE Computer Society. 2008, pp. 481–492.

[9] Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. “A survey of mobile phone sensing”. In: *Communications Magazine, IEEE* 48.9 (2010), pp. 140–150.

[10] Daniel Lehmann, Liadan Ita Oćallaghan, and Yoav Shoham. “Truth revelation in approximately efficient combinatorial auctions”. In: *Journal of the ACM (JACM)* 49.5 (2002), pp. 577–602.

[11] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. *Microeconomic theory*. Vol. 1. Oxford university press New York, 1995.

[12] Nolan Miller, Paul Resnick, and Richard Zeckhauser. “Eliciting informative feedback: The peer-prediction method”. In: *Management Science* 51.9 (2005), pp. 1359–1373.

[13] Prashanth Mohan, Venkata N Padmanabhan, and Ramachandran Ramjee. “Nericell: rich monitoring of road and traffic conditions using mobile smartphones”. In: *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM. 2008, pp. 323–336.

[14] Rajeev Motwani. *Randomized algorithms*. Cambridge university press, 1995.

[15] Ahuva Mu’Alem and Noam Nisan. “Truthful approximation mechanisms for restricted combinatorial auctions”. In: *Games and Economic Behavior* 64.2 (2008), pp. 612–631.

[16] Ahuva Mu’Alem and Noam Nisan. “Truthful approximation mechanisms for restricted combinatorial auctions”. In: *Games and Economic Behavior* 64.2 (2008), pp. 612–631.

[17] Victor Naroditskiy, Iyad Rahwan, Manuel Cebrian, and Nicholas R Jennings. “Verification in referral-based crowdsourcing”. In: *PloS one* 7.10 (2012), e45924.

[18] George L Nemhauser and Laurence A Wolsey. *Integer and combinatorial optimization*. Vol. 18. Wiley New York, 1988.

[19] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. “Learning from crowds”. In: *The Journal of Machine Learning Research* 11 (2010), pp. 1297–1322.

[20] Michael H Rothkopf, Aleksandar Pekeč, and Ronald M Harstad. “Computationally manageable combinational auctions”. In: *Management Science* 44.8 (1998), pp. 1131–1147.

[21] Tim Roughgarden and Mukund Sundararajan. “Is efficiency expensive”. In: *Third Workshop on Sponsored Search Auctions*. 2007.

[22] John C Tang, Manuel Cebrian, Nicklaus A Giacobe, Hyun-Woo Kim, Taemie Kim, and Douglas Beaker Wickert. “Reflecting on the DARPA red balloon challenge”. In: *Communications of the ACM* 54.4 (2011), pp. 78–85.

[23] Arvind Thiagarajan, Lenin Ravindranath, Katrina LaCurts, Samuel Madden, Hari Balakrishnan, Sivan Toledo, and Jakob Eriksson. “VTrack: accurate, energy-aware road traffic delay estimation using mobile phones”. In: *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2009, pp. 85–98.

[24] Long Tran-Thanh, Trung Dong Huynh, Avi Rosenfeld, Sarvapali D Ramchurn, and Nicholas R Jennings. “BudgetFix: budget limited crowdsourcing for interdependent task allocation with quality guarantees”. In: *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2014, pp. 477–484.

[25] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. “Efficient crowdsourcing of unknown experts using multi-armed bandits”. In: *European Conference on Artificial Intelligence*. 2012, pp. 768–773.

[26] Matteo Venanzi, Alex Rogers, and Nicholas R Jennings. “Crowdsourcing Spatial Phenomena Using Trust-Based Heteroskedastic Gaussian Processes”. In: *First AAAI Conference on Human Computation and Crowdsourcing*. 2013.

[27] Peter Welinder, Steve Branson, Pietro Perona, and Serge J Belongie. “The multidimensional wisdom of crowds”. In: *Advances in Neural Information Processing Systems*. 2010, pp. 2424–2432.

[28] Celal Ziftci, Nima Nikzad, Nakul Verma, Piero Zappi, Elizabeth Bales, Ingolf Krueger, and William Griswold. “Citisense: Mobile Air Quality Sensing for Individuals and Communities”. In: *Proceedings of the 3rd Annual Conference on Systems, Programming, and Applications: Software for Humanity*. ACM, 2012, pp. 23–24.