

Deliverable D.2.1: Report on the Relevant Literature, Current Practices, Main Challenges and Research Questions

Evangelos Markakis Thanasis Papaioannou Georgios Papasotiropoulos
Georgios D. Stamoulis Panagiotis Tsamopoulos

Abstract

Reverse auctions have been applied successfully for assigning a project, a task, or a service, commissioned by a central entity, to one or more candidate providers. In the recent years, we have experienced the emergence of several new market environments, e.g., in crowdsourcing or participatory sensing platforms, that bring forward several challenges to the design of reverse auctions. The purpose of this deliverable is to provide an overview of mathematical models that have been proposed in the relevant literature, along with the progress that has been made on the design of mechanisms in this context. At the same time, we wish to identify current research challenges and open problems, with an emphasis on crowdsourcing and labor matching markets.

1 Introduction

Reverse auctions, also known as procurement auctions, form a popular class of mechanisms for assigning a project, a task, or a service, commissioned by a central entity, to one or more candidate providers. In the more traditional applications of reverse auctions, the central entity may be a governmental organization (e.g., for assigning a public project to a construction firm), or a company looking for partners who can perform specialized tasks. In the private sector, reverse auctions have been successfully used for supply management, where the central entity is a firm interested in outsourcing certain services, such as the supply of logistics services, transport services, electronic equipment, or retail services, see [11]. Going further, and given the explosion and monetization of the Internet, in more modern applications, the central entity may be even a single individual, as is often the case in online labour markets and crowdsourcing systems, e.g., Upwork, Freelancer, Amazon Mechanical Turk, and Clickworker are some representative platforms, where the posted tasks can range from very elementary assignments as image labeling, or survey completion to more specialized services like translating documents. Participatory crowdsensing systems also fall under this framework, where some entity is interested in collecting sensing information from smartphones or other devices. Hereinafter, we will refer to the central entity as the "auctioneer", irrespective of the application at hand.

The emergence of such online labour markets, has caught the increasing attention of the research community, which has focused on properly formulating and addressing relevant research questions. As expected, this has led to a fusion of ideas from Computer Science and Economics. In particular, the main high level challenges that have been brought forward have to do with the design of market mechanisms that achieve

- computational efficiency (so that a mechanism is simple, easy to understand for the users, and with scalable running time),

- economic efficiency (which is evaluated by metrics such as the derived social welfare, the level of satisfaction for the final outcome, or the total expenditure),
- incentives for honest participation (so that the involved entities cannot attempt to manipulate the outcome by misreporting their preferences).

In the case of auctions for a single task, most critical issues are well understood, and we can have relatively simple mechanisms with desirable properties. Unfortunately, when we have combinatorial reverse auctions with several tasks to be allocated at the same time, much more complications arise, since the above criteria are often in conflict with each other. To elaborate more on this, it is already known that optimizing the final utility of the auctioneer or the derived social welfare is a computationally hard (NP-complete) problem due to the combinatorial explosion on the number of possible solutions, when there are many tasks to be auctioned. Some of the economic criteria are also often incompatible, e.g., optimizing the auctioneer’s profit versus optimizing the overall welfare (the sum of everyone’s utility). Finally, the design of incentive compatible mechanisms often results in computational intractability. Therefore, apart from very few notable cases, we cannot hope to have computationally and economically efficient, incentive compatible mechanisms with provable performance guarantees.

Driven by the above considerations, several works have explored, to a certain extent, the types of trade-offs that can be achieved by relaxing one or more of the desirable properties mentioned earlier. As an example, finding approximately optimal solutions, can alleviate the computational intractability burden in some cases. This approach however still presents challenges in richer or more expressive models. As a second example, when there is a strict budget constraint for the auctioneer, one might need to relax either the incentives guarantees (as truthful mechanisms have a tendency to produce relatively high payments), or the optimization guarantee (as budgeted versions of many optimization problems tend to be intractable).

The existing literature has already made significant progress on examining the effects of different aspects and modeling parameters on the performance of relevant mechanisms. At the same time, there are still many directions that remain underexplored and where the design of better mechanisms has been elusive. The purpose of this survey is to report on mathematical models and results that have been obtained within the last decade for reverse auctions, with an emphasis on mechanism design aspects for crowdsourcing and labor matching markets. We also wish to identify the current challenges and open problems, that can serve as a guide for future researchers in the area.

The rest of the survey is structured as follows: In Section 2, we provide some basic definitions that are useful for all the models that we study later on. In Section 3, we study some of the most elementary, yet non-trivial, models that have been proposed. We believe these models help the reader understand what are the essential challenges and the compromises one has to settle with in the design of efficient mechanisms. In Section 4, our goal is to focus on richer models that take further aspects and parameters into account and present some generalizations of results from Section 3. Moving on, in Section 5, we focus on mechanisms that arise in the context of spatial crowdsourcing, which is a prominent paradigm for task allocation. Finally, in Section 6, we pay special attention to reputation-based mechanisms, which maintain information about the quality of the participating agents. We conclude in Section 7 with a discussion on directions for future work.

2 Basic Definitions

Given the plethora of works for procurement auctions and crowdsourcing platforms, this survey will focus on mathematical models that are most related to the COMRADE project. In particular, we are interested in presenting models of constrained optimization with the goal of designing mechanisms that achieve high economic performance and guarantee incentive properties in computationally efficient ways.

We start with a generic model for crowdsourcing, which we will specialize in subsequent sections. We consider a procurement (reverse) auction setting, where an entity that we will refer to as the auctioneer is interested in purchasing goods or assigning a set of tasks to interested agents. For reasons of uniformity, we will stick to the scenario of task assignment. We let $A = [n] = \{1, 2, \dots, n\}$, denote a set of n agents who are willing to participate in the reverse auction.

There are various possibilities for modeling the capabilities of each agent, the requirements of the auctioneer, and the bidding interface for making offers. E.g., the auctioneers may simply want to assign each task to a single agent, or assign tasks multiple times (for quality assurance). Each agent may be capable of performing a single task out of the tasks posted by the auctioneer, or she may be able to bid for performing a subset of the tasks. The interface may also allow agents to make multiple bids on different combinations of tasks, or to make bids only for single tasks. The choices made for these aspects can lead to a variety of different models for task assignment. From Section 3 onwards, we explicitly describe the specific models that have attracted significant attention in the related works of the literature.

Regardless of which model we use, the auctioneer collects the offers made by the agents and eventually needs to make two decisions: the first is to decide how to allocate the tasks to the agents (it is allowed to also leave tasks unallocated), and the second is to issue a payment to every agent (naturally, no payment is made to agents who were not assigned anything). Finally, to evaluate the selected outcome, the auctioneer is also associated with a valuation function $v : P(A) \rightarrow \mathbb{R}$, where $P(A)$ is the powerset of the set of agents¹. The valuation function determines the total value (level of happiness) derived by the selection made. Namely, for $S \subseteq A$, $v(S)$ is the value derived by the auctioneer if the set S is selected for performing the tasks. For notational convenience, we will use $v(\cdot)$ instead of $v(\{\cdot\})$ for singletons.

2.1 Desirable Properties in Mechanism Design

In most of the settings that we will consider here, the agents are strategic, in the sense that they may try to misreport their true costs for performing the tasks in order to enforce a higher payment for them. To address this concern, we need to resort to ideas from the field of mechanism design.

More formally, let θ_i be the actual type of agent i , so θ is the type profile, i.e., the vector containing the type θ_i of each agent i . The type can be either a single parameter (e.g., the actual cost of performing a task) or can be made up of multiple parameters (the costs for performing different combinations of tasks, arrival time of an agent, departure time, etc). A mechanism $\mathcal{M} = (f, p)$ takes as input a bidding vector $\mathbf{b} = (b_1, \dots, b_n)$, where b_i may differ from the real type θ_i , and consists of an allocation rule f and a payment rule

¹In some settings, the valuation function may not only depend on the set of workers, e.g., when the workers may have a different quality level for each task, and there are multiple ways to split the tasks among the workers.

p . Hence, $f(\mathbf{b})$ specifies the task assignment to every agent, and $p(\mathbf{b}) = (p_i(\mathbf{b}))_{i \in A}$ is the payment vector, where $p_i(\mathbf{b})$ denotes the payment issued to agent i .

To avoid the strategic behavior of agents, the dominant solution concept that has been employed and extensively studied is the notion of truthfulness.

Definition 2.1. *A mechanism $\mathcal{M} = (f, p)$ is truthful, if reporting the actual type θ_i is a dominant strategy for every agent i , or equivalently, is the optimal move regardless of the actions of other players.*

For randomized mechanisms, we use the notion of *universal truthfulness*, which means that the mechanism is a probability distribution over deterministic truthful mechanisms, and such mechanisms can be more powerful than their deterministic counterparts [31].

Truthfulness on its own can be achieved easily (e.g., never allocate any tasks). The main challenge in mechanism design is to design truthful mechanisms that also achieve other desirable properties with regard to their economic performance. We list below some of the properties that are the focus of the relevant literature.

Definition 2.2. *A mechanism $\mathcal{M} = (f, p)$ is*

1. *individually rational, if for every agent, her payment is never lower than the declared cost for performing the tasks assigned to her,*
2. *welfare maximizing if the outcome maximizes the social welfare, for every type vector θ ,*
3. *no-deficit (or profitable), if the final utility of the auctioneer is never negative or equivalently, the sum of payments issued by the auctioneer never exceed the value derived by the assignment selected,*
4. *budget feasible, if whenever the auctioneer has a budget constraint of B , the total amount paid to the participants by the auctioneer does not exceed B . The optimization variant that aims to the minimization of the total amount paid by the auctioneer has been also of interest.*

2.2 Classes of Valuation functions

We proceed with giving some relevant definitions on how the auctioneer evaluates an outcome of the mechanism. For the more basic models that will be considered in this survey, it is assumed that the auctioneer is associated with a valuation function $v : P(A) \rightarrow \mathbb{R}$, defined on all possible subsets of A , i.e., the auctioneer derives value for any subset of agents she may select as the winners. In richer models, the exact value may be determined by further parameters of the model under consideration (i.e., dependent on what the agents are expected to deliver to the auctioneer if selected as winners).

We say that a valuation function $v : P(A) \rightarrow \mathbb{R}$, is **additive** over the set A , if for each $S \subseteq A$ we have

$$v(S) = \sum_{k \in S} v_k$$

where v_k is the value added to the auctioneer by selecting only agent k .

A valuation function $v : P(A) \rightarrow \mathbb{R}$ is said to be **submodular** over A , if for each $S \subseteq T \subseteq A$ and for each item $j \notin T$ we have

$$v(T \cup \{j\}) - v(T) \leq v(S \cup \{j\}) - v(S)$$

Submodular valuation functions are essentially the discrete analogous of concave functions. An alternative characterization is the following:

Theorem 2.3 (Nemhauser et al. [30]). *A set function v is submodular if and only if for all $S, T \subseteq A$ we have $v(T) \leq v(S) + \sum_{i \in T \setminus S} (v(S \cup \{i\}) - v(S)) - \sum_{i \in S \setminus T} (v(S \cup T) - v(S \cup T \setminus \{i\}))$.*

A valuation function $v : P(A) \rightarrow \mathbb{R}$ is said to be ***fractionally subadditive*** (or *XOS*), if there exists a set $\{f_1, f_2, \dots, f_m\}$ of additive functions for some finite m , such that

$$v(S) = \max\{f_1(S), f_2(S), \dots, f_m(S)\}.$$

Finally, a valuation function $v : P(A) \rightarrow \mathbb{R}$ is said to be ***subadditive***, if for each $S, T \subseteq A$, with $S \cap T = \emptyset$, we have

$$v(S \cup T) \leq v(S) + v(T)$$

Throughout this work, we also make the natural assumption that $v(\emptyset) = 0$. All the functions described above have the property of *substitutability*. That is, some items or subsets of items are similar in value or can be thought of as substitutes of each other, and therefore, they should be procured separately. This is a quite common phenomenon in such markets, and these families of functions have been used extensively in the related literature.

Before concluding the current subsection, we also note that it is easy to establish some containment results concerning these classes of valuation functions, by carefully inspecting their definitions. More specifically, the following relations hold:

$$\text{Additive} \subseteq \text{Submodular} \subseteq \text{XOS} \subseteq \text{Subadditive}$$

2.3 Representation of valuation functions.

We end this section with a discussion on how an algorithm accesses the valuation function of the auctioneer. Throughout the execution of a mechanism, the allocation algorithm needs to use the value of the auctioneer for several different subsets of agents. When the function $v(\cdot)$ can be given in a closed form, then one can compute $v(S)$ for any $S \subseteq A$, by applying a formula, as is the case for example with additive valuation functions. More generally, we say that a function has a *succinct representation*, when it is possible to compute $v(S)$ by applying a polynomial time algorithm. If no such representation is available, the description of the valuation function might be exponentially large. In such cases (as in Section 3.2.1 or 3.2.3), we assume that the algorithm has oracle access to the valuation function via queries. The most natural query model is the so called *value query* model, where the algorithm can query the value $v(S)$, for a set $S \subseteq A$. Clearly, the focus then is on algorithms that can make only a small number of queries, e.g., polynomially many value queries can be considered acceptable. Another common approach is the use of *demand queries*, where a demand oracle receives a vector of prices $p_1, p_2, \dots, p_n$, representing candidate payments to the agents, and returns a maximal subset S such that $S \in \argmax_{T \subseteq A} (V(T) - \sum_{t \in T} p_t)$. It turns out that a value query can be simulated by a polynomial number of demand queries, hence, an algorithm that works with polynomially many value queries, can also be executed with polynomially many demand queries. The reverse however is not always true, which means that the demand query model is more powerful. In the coming sections, most of the results presented concern functions with a succinct representation or with access via value queries. If this is not the case, we state explicitly that the result holds under the demand query model.

3 Basic Models: Offline Optimization for Single-parameter Agents

In the current section, we will analyze first a basic class of models, studied in a big part of related works (right from the early literature and up until recently), that are relatively simple to describe, but still contain all the essential challenges within the crowdsourcing context.

3.1 Main Features

Let $A = \{1, \dots, n\}$ be the set of potential workers. We will be using the term agent and worker interchangeably in the sequel. Consider also an auctioneer who is associated with a set of tasks that he would like to see completed (not necessarily all of them, but enough so that he gains a sufficient benefit from the final outcome).

Although there are differences in the precise models used in the papers discussed in this section, they all share the following features:

- **Atomic tasks.** Each task under consideration is distinct, not decomposable into sub-tasks, and not dependent on the completion of any other tasks.
- **Offline optimization.** We assume that we are only examining a certain time window or snapshot of a system and that all the necessary information for determining how to assign the tasks is available right at the beginning of this time window. This means that all the workers under consideration are available for the whole duration of this period, and that the tasks are posted from the beginning, and can be executed at any time during the snapshot.
- **Single-parameter agents.** This involves the amount of private information that an agent is asked to reveal to the auctioneer. By single-parameter agents, we refer to workers who submit only one parameter to the mechanism, typically their cost for executing the assigned tasks. This assumption can capture several different scenarios. E.g., for an agent i , her cost c_i could simply be the cost incurred to her for executing a single task. In other cases, it can denote the cost for executing a specific collection of tasks that the worker is capable of. Regardless of the interpretation of the cost c_i , all these settings can be treated in a unified manner from the viewpoint of mechanism design.

Under the assumption of single-parameter agents, there is a general methodology for obtaining truthful mechanisms due to the celebrated result of Myerson [29]. In more detail, in order to design a truthful mechanism $\mathcal{M} = (f, p)$, one needs to focus only on enforcing a certain property on the allocation rule f , as defined below.

Definition 3.1. *We say that an allocation rule f is monotone, if for every agent $i \in A$, and any vector of cost declarations $\mathbf{b}$, if $i \in f(\mathbf{b})$, then $i \in f(b'_i, \mathbf{b}_{-i})$ for $b'_i \leq b_i$.*

In other words, Definition 3.1 simply means that if an agent i is selected in the outcome by declaring cost b_i , then by declaring a lower cost for offering the same service, he should still be selected. Myerson's lemma below implies that monotone allocation algorithms admit truthful payment schemes.

Lemma 3.2 (Myerson’s lemma [29]). *Given a monotone algorithm f , there is a unique payment scheme p such that (f, p) is a truthful and individually rational mechanism, given by*

$$p_i(b) = \begin{cases} \sup_{b_i \in [c_i, \infty)} \{b_i : i \in f(b_i, b_{-i})\}, & \text{if } i \in f(b) \\ 0, & \text{otherwise} \end{cases}$$

The payments in Lemma 3.2 are often referred to as *threshold payments*, since they indicate the threshold at which an agent stops being selected. Myerson’s lemma greatly simplifies the design of truthful mechanisms by focusing only on constructing monotone algorithms and not having to worry about the payment scheme. Once a monotone allocation algorithm has been established, the corresponding payments to the agents can be computed either by a direct formula (in simpler settings) or by an appropriate binary search. For all the algorithms presented within Section 3, we assume that the underlying payment scheme is given by Myerson’s lemma.

3.2 Mechanism Design under Budget Constraints

In this subsection, we discuss a family of models that were originally introduced by Singer [35], and whose common aspect is the existence of a strict upper bound for the payments made by the auctioneer. In a budget-constrained reverse-auction, the auctioneer has a budget $B > 0$, denoting the maximum amount that can be spent for payments to all agents. Each participating agent i is single-parameter, and has a private cost $c_i \geq 0$ associated with the service that she can offer to the auctioneer (which can correspond to a single task or a collection of tasks). Moreover, each agent i is asked to submit an offer b_i to the auctioneer, which may or may not be equal to the actual private cost c_i .

As already described in Section 2, a valuation function is associated with the auctioneer, in the form $v : P(A) \rightarrow \mathbb{R}$, where $P(A)$ is the powerset of the set of agents, A . Hence, $v(S)$ denotes the value (which can be thought of as the level of profit or happiness) derived by the auctioneer if he selects the set S of agents as the auction winners. It is also assumed that when a set S of agents is selected as the winners, these agents are committed to executing the tasks corresponding to their offer. To summarize, the auctioneer collects the offers of the agents, and then needs to select a set $S \subseteq A$, as the winning agents, along with a payment to each winner. Let p_i denote the payment issued to agent i . The property of individual rationality dictates that $p_i \geq b_i$ for every winning agent, i.e., a winner cannot be paid less than her declared cost for the tasks. At the same time, the constraint of budget feasibility requires that the total sum of all payments cannot exceed the budget B .

Therefore, a natural goal for the auctioneer under this model, is to design a truthful mechanism for maximizing his attained value under the given constraints, which results in solving the following optimization problem:

$$\begin{aligned} & \text{maximize } v(S) \\ \text{s. t. } & S \subseteq A \\ & \sum_{i \in S} p_i \leq B \\ & p_i \geq b_i, & \forall i \in S \\ & p_i = 0, & \forall i \in A \setminus S \end{aligned}$$

We stress that budget feasibility makes the problem more challenging, with respect to truthfulness, as it already rules out some standard approaches, as the VCG mechanism. It is well known that the algorithmic versions of the problems of interest here, often admit

constant factor approximation algorithms, but it is not clear how to appropriately convert them into truthful budget feasible mechanisms. What is even more interesting is that the problem is nontrivial even if we allow exponential time algorithms, since computational power does not necessarily make the problem easier (see the discussion in [14]). This is in stark contrast to the case where there are no budget constraints, where in exponential worst-case time, we can always have a truthful mechanism (VCG). All these issues create an intriguing landscape, where one needs to strike a balance between the incentives of the agents, the budget constraints, and the computational requirements.

Unfortunately, even if we drop the requirement for truthfulness and even for simple additive valuation functions, the problem is NP-hard and cannot be solved efficiently. To briefly justify why this is the case, when $v(S)$ is additive, one can observe that the problem simply degenerates to a Knapsack problem. Since computational efficiency is not compatible with optimizing the auctioneer's valuation function, most of the literature has focused on finding mechanisms with provable approximation guarantees with respect to the optimal budget-feasible solution. To be more precise about the notion of approximation that will be examined, let $OPT(A, B)$ denote the optimal solution for the auctioneer's value under the budget B , when the set of available workers is A , and when all workers are truthful. We use the following, quite standard notion of multiplicative approximation for evaluating a task assignment.

Definition 3.3. *We say that a mechanism achieves a ρ -approximation for $\rho \geq 1$, if for any instance, it selects a set $S \subseteq A$ such that $v(S) \geq 1/\rho \cdot OPT(A, B)$.*

Hence, eventually the goal of the auctioneer is to design a truthful, individually rational, budget-feasible mechanism that achieves the best possible approximation guarantee in polynomial time. The results that have been obtained so far along this direction in the literature are dependent on the assumption that is made about the auctioneer's valuation function. We find it appropriate to follow this categorization in the sequel as well.

3.2.1 Additive and Submodular Valuations

Among the related works, we stress that additive and submodular valuation functions are the most well studied families due to their applicability in modeling procurement problems. We therefore start with highlighting the most important ideas that have been employed in these cases.

For such functions, most of the literature has used a greedy approach for constructing truthful mechanisms. We should note however that one cannot use a completely naive greedy algorithm as this may affect both the approximation guarantee and the incentives of the mechanism. Hence one needs to be slightly more careful in order to enforce the desirable properties. We describe now the general mechanism design framework that has been used, which was first demonstrated by Singer [35], and by Chen et al. [10]. Namely, we present the mechanisms RAND-MECH-SM and MECH-SM, a randomized mechanism and its corresponding deterministic version. For these mechanisms, we have chosen to present here the improved analysis by [23] for their approximation ratio, but before stating these mechanisms and their analysis it is important to focus on a greedy procedure which plays a crucial role in both of them.

GREEDY-SM($A, v, \mathbf{c}, B/2$) [10]

```

1 Let  $k = 1$  and  $S = \emptyset$ 
2 while  $k \leq |A|$  and  $v(S \cup \{k\}) > v(S)$  and  $c_k \leq \frac{B}{2} \cdot \frac{v(S \cup \{k\}) - v(S)}{v(S \cup \{k\})}$  do
3   |  $S = S \cup \{k\}$ 
4   |  $k = k + 1$ 
5 return  $S$ 

```

The greedy mechanism GREEDY-SM is a greedy algorithm that picks agents according to their ratio of marginal value over cost, given that this cost is not too large. For the sake of presentation, we assume the agents are sorted in descending order with respect to this ratio. The marginal value of each agent is calculated with respect to the previous agents in the ordering, i.e., $1 = \arg \max_{j \in A} \frac{v(j)}{c_j}$ and $i = \arg \max_{j \in A \setminus [i-1]} \frac{v([j]) - v([j-1])}{c_j}$ for $i \geq 2$. To analyze RAND-MECH-SM and its deterministic counterpart, the crucial step is to establish the guarantee that is provided by GREEDY-SM, which is stated below.

Lemma 3.4 (inferred from [10] and [23]). *GREEDY-SM is monotone. Assuming an additive or more generally, a (non-decreasing) submodular function v and the payments of Myerson's lemma, GREEDY-SM($A, v, \mathbf{c}, B/2$) is budget-feasible and outputs a set S such that $v(S) \geq \frac{1}{3} \cdot \text{OPT}(A, B) - \frac{2}{3} \cdot v(i^*)$.*

Moving on to the RAND-MECH-SM, the main idea is that it either accepts only the agent that provides the highest value, or it runs a greedy procedure with an appropriately reduced budget.

RAND-MECH-SM($A, v, \mathbf{c}, B$) [10]

```

1 Set  $A' = \{i \mid c_i \leq B\}$  and  $i^* \in \arg \max_{i \in A'} v(i)$ 
2 return  $i^*$  with probability  $\frac{2}{5}$ , and GREEDY-SM( $A, v, \mathbf{c}, B/2$ ) with probability  $\frac{3}{5}$ 

```

A derandomized version of RAND-MECH-SM is also provided by [10]. It has all the desired properties, while suffering a small loss on the approximation factor. Following the improved analysis of [23], we fine-tune this derandomized mechanism to obtain the deterministic mechanism MECH-SM below.

MECH-SM($A, v, \mathbf{c}, B$)

```

1 Set  $A' = \{i \mid c_i \leq B\}$  and  $i^* \in \arg \max_{i \in A'} v(i)$ 
2 if  $(2 + \sqrt{6}) \cdot v(i^*) \geq \text{OPT}(A \setminus \{i^*\}, B)$  then
3   | return  $i^*$ 
4 else
5   | return GREEDY-SM( $A, v, \mathbf{c}, B/2$ )

```

Using Lemma 3.4, one can then analyze the mechanisms and establish their worst-case approximation guarantees. The next theorem summarizes the properties of RAND-MECH-SM and MECH-SM.

Theorem 3.5 (inferred from [10], [23]). *RAND-MECH-SM is a randomized mechanism that runs in polynomial time, it is universally truthful, individually rational, budget-feasible, and has approximation ratio 5. MECH-SM is a deterministic mechanism, that is truthful, individually rational, budget-feasible, and has approximation ratio $3 + \sqrt{6}$.*

For submodular functions, the guarantees of Theorem 3.5 are the currently best known. Coming now to the special case of additive functions, the state of the art was recently

improved by [19]. The approach of [19] is to use first a greedy scheme in order to exclude low *value-per-cost* agents, as before. In the second phase of their algorithms, a type of posted-pricing scheme is used for the remaining agents. The currently best approximation guarantees for additive valuations are summarized as follows.

Theorem 3.6 ([19]). *For additive valuations, there exists a truthful, randomized polynomial time, budget-feasible mechanism achieving a 2-approximation ratio. Furthermore, there exists a truthful, deterministic, polynomial time and budget-feasible mechanism that achieves a 3-approximation.*

As discussed later in Section 3.2.4, the randomized mechanism above cannot be improved upon, and there is only small room for improvement for deterministic mechanisms. Hence, we can conclude that for additive valuations, the truthful maximization of the auctioneer’s valuation is essentially resolved.

Further assumptions in the input can be made so as to achieve better approximation guarantees. A representative example is the so-called, *large markets* assumption and crowdsourcing markets often form a good representative example of this situation. Informally speaking, a market is considered to be large if no single agent can have a significant effect on the final outcome. Mathematically, the precise definition may depend on the model, but the intuition remains the same. For the single-parameter, budget-constrained models that we study in this section, the most typical *large market assumption* is that $c_{\max}/B \rightarrow 0$, where $c_{\max} = \max_{i \in A} c_i$, and B is the budget. This means that the cost of any worker that the mechanism may select cannot be a significant fraction of the total budget available to the auctioneer. This assumption has been utilized in other examples of big markets, such as the auctions for online advertizing, e.g., see [28] (where they refer to it as the *small bid-to-budget ratio* assumption).

In our context, the large market assumption can help in achieving better approximation guarantees for truthful, budget feasible mechanisms. Namely, in [4], it was proved that with additive valuation functions, one can have mechanisms with an approximation of $1 - 1/e$, which was also shown to be the best possible. Hence, we can come much closer to the absolute optimal solution, compared to what we can guarantee without the assumption. In the same work, improved results were given for submodular valuations as well.

An alternative way to exploit large market effects is to make an assumption regarding the obtained value of the auctioneer instead of the costs incurred. For additive valuations, the work of [4] considers the assumption $v_{\max}/OPT(A, B) \rightarrow 0$, where $v_{\max} = \max_{i \in A} v(i)$. This means that no worker can be responsible for a significant fraction of the auctioneer’s optimal benefit. The definition can also be used for submodular valuations, since $v(i)$ is the best marginal value that an agent can contribute to a solution. In a similar fashion as done with the previous assumption, it has been shown in [4] that better results can be obtained under large markets for additive and submodular valuations. Additionally, this version of the large market assumption has also been used in a richer model with additive valuations by [18], where the auctioneer does not only need to decide on the set of winners, but also needs to determine which task is allocated to each agent. This is applicable in cases where each agent can have different skill sets and the final solution needs to be a matching (in the natural, graph-theoretic modeling of the problem as a bipartite graph), where precisely one task is allocated to each of the winners. Constant factor approximation guarantees are established for this model under the large market assumption, which are also applicable when agents can be assigned more than one task.

Non-monotone submodular valuations

We switch now to a variant of the problem, which presents an additional challenge. All the results mentioned so far have assumed that the valuation function of the auctioneer is a monotone function, i.e., $v(S) \geq v(T)$, whenever $T \subseteq S$. This however may not always be true. Imagine for example a situation where 2 agents might be incompatible or not well-fitted with each other. Some agents may also have negative influence on the completion of a task. To further motivate the study of non-monotone submodular objectives, consider the following well-studied sensor placement problem [8, 12, 24]: assume that we want to monitor some spatial phenomenon (e.g., the temperature of a specific environment), modeled as a Gaussian process. We may place sensing devices on some of the prespecified locations, but each location has an associated cost. A criterion for finding an optimal such placement, suggested by Caselton and Zidek [8] for the unit cost case, is to maximize the *mutual information* between chosen and non chosen locations, i.e., we search for the subset of locations that minimizes the uncertainty about the estimates in the remaining space. Such mutual information objectives are submodular but not monotone. In addition, it is straightforward to modify this problem to model participatory crowdsensing scenarios where users have incentives to lie about the true cost of installing a sensor.

Unfortunately, none of the results mentioned so far can be directly generalized for non-monotone functions. A first approach for deriving mechanisms for non-monotone objectives was given by [14]. There they focused on the so-called *cut functions*, a prominent class of non-monotone submodular functions, as discussed in [33]. By a cut function, it is meant that $v(S)$ equals the value of the cut between S and $A \setminus S$ in an appropriately defined graph, where agents correspond to vertices. The work of [14] presents a randomized greedy scheme, which is then turned into a deterministic mechanism. Although this is a technically interesting work, that manages to adapt the use of greedy schemes for non-monotone functions, the final approximation guarantees are prohibitively high.

Later on, [2] considered a more general class of valuation functions that contains cut functions. In particular, the focus of their work was on symmetric² submodular functions. Cut functions are a special case and are considered to be a canonical example of this class. The main results of [2] include significant improvements on the approximation ratio established for symmetric submodular valuation functions, and provide some partial answers to the open questions discussed in [14]. Although their mechanisms do not run in polynomial time in general, for every such function (the running time depends on further properties of the function itself), they do so for cut functions and their weighted generalizations. The results can be summarized as follows:

Theorem 3.7 ([2]). *For symmetric submodular functions, there exists a truthful, randomized 10-approximation budget feasible mechanism. For cut functions, the mechanism runs in polynomial time, and a deterministic analogue achieves a 27.25-approximation, also in polynomial time. The latter applies to weighted cut functions as well.*

Regarding the approach of [2], the core idea is to exploit a combination of local search with mechanisms for monotone submodular functions. The reason local search is convenient for symmetric submodular functions is that it produces two local optima, and one can then prove that the function $v(\cdot)$ is non-decreasing when restricted to subsets of each local optimum. By utilizing mechanisms for monotone submodular functions on

²In some works on mechanism design, symmetric submodular functions have a different meaning and refer to the case where $v(S)$ depends only on $|S|$. Here we have adopted the terminology of earlier literature on submodular optimization, e.g., [17].

the two subsets, the final step boils down to proving that one of the two solutions will attain a good approximation. The running time is still an issue under this approach, since finding an exact local optimum is not guaranteed to terminate fast. However, by finding *approximate* local optima, one can appropriately adjust the initial framework and obtain provable approximation guarantees.

Finally, in a recent breakthrough [3], a new mechanism was proposed for arbitrary non-monotone submodular valuations. The mechanism uses a carefully designed and fine-tuned greedy approach, which tries to build two candidate solutions simultaneously. To handle the incentives of the agents, the mechanism needs to prevent them to jump from one solution to the other by changing their cost. This is done with offering each agent a take-it or leave-it price. We note that this is a randomized mechanism, and uses random sampling as an essential building block, to estimate the optimal value. The use of random sampling is not surprising since it has been common in many other randomized auction mechanisms. Finding a deterministic mechanism with similar guarantees seems a technically challenging problem, as one would need to replace the random sampling approach by some other polynomial-time procedure for estimating the optimal value. The main result of [3] is summarized below.

Theorem 3.8 ([3]). *For arbitrary, non-monotone submodular valuation functions, there exists a truthful, randomized, budget-feasible mechanism achieving a constant factor approximation.*

We note that although the approximation factor in the above theorem is a large constant, the mechanism is valuable since it is the first one achieving an approximation ratio that is independent of the input variables.

3.2.2 Fractionally subadditive (XOS) valuations

We continue with the case of XOS valuations. This forms a strict superclass of submodular valuation functions, defined as follows. The value for any subset $S \subseteq A$ of agents is taken as the maximum among a collection of additive valuations. Hence, an XOS function $v(\cdot)$ is defined by a set of additive valuation functions $\alpha_1, \alpha_2, \dots, \alpha_r$, for some $r \in \mathbb{N}$, such that $v(S) = \max\{\alpha_1(S), \alpha_2(S), \dots, \alpha_r(S)\}$ for any $S \subseteq A$. We note that other equivalent definitions also exist, however this one is technically convenient in certain proof arguments.

The class of XOS functions is very appealing from a mathematical viewpoint, as it lies between submodular and subadditive functions, and it has already been observed that in several occasions the analysis of problems concerning submodular optimization is generalizable to this class.

We will first discuss some results (a deterministic and a randomized mechanism) for a special class of XOS functions, motivated by matroid theory. Both of these mechanisms were introduced by Amanatidis et al. [1]. To begin with, we should acquaint ourselves with the definition of an independence system.

Definition 3.9. *An independence system is a pair (U, I) , where U is an arbitrary finite set and $I \subseteq P(U)$ is a family of subsets, whose members are called independent sets of U and satisfy:*

1. $\emptyset \in I$
2. If $B \in I$ and $A \subseteq B$ then $A \in I$.

Note that if in the above definition, we additionally require that the *exchange property* must be satisfied (which is: if $A, B \in I$ with $|A| < |B|$, then $\exists x \in B \setminus A : A \cup \{x\} \in I$), we have precisely the definition of a *matroid* (U, I) . Since Knapsack problems with matroid constraints have been well studied, it is now natural to define a variant of the Knapsack problem where the feasible solutions are constrained to an independence system. Let *Independence System Knapsack* (ISK) be the computational problem for which, given an independence system (U, I) with costs c_i and values v_i on the elements of U , as well as a budget B , the goal is to find $S \in I$ that maximizes $\sum_{i \in S} v_i$ subject to $\sum_{i \in S} c_i \leq B$.

The following observations illustrate the significance of Independence System Knapsack (ISK from now on), as a wide variety of common problems are in fact special cases of ISK. Firstly, it is obvious that problems like Budgeted Max Weighted Matroid Member (where (U, I) is a matroid) can be modeled as ISK problems, since any matroid is also an independence system. Secondly, if we let $U = [n]$ and I be its powerset, then ISK reduces to the plain Knapsack problem. Moving on, if U is the set of edges of a given graph G , and I is the set of all matchings of G or the set of all its acyclic subgraphs, then ISK corresponds to the well known Budgeted Max Weighted Matching problem or the Budgeted Max Weighted Forest problem respectively. On the other hand, if U is the set of vertices of a given graph G , and I is the set of the independent sets of G , then ISK coincides with the problem of Budgeted Max Independent Set.

It can be easily proven that every problem which can be formulated as an ISK problem belongs to the XOS class [1]. Out of these, the problem of Budgeted Max Weighted Matching, turns out to be a good representative ISK example. We remind the reader that Budgeted Max Weighted Matching is defined as the problem of finding a matching of maximum value, subject to a budget constraint B in a given graph, where each edge e has a cost c_e and a value v_e . The mechanism design version of the problem suggests that each edge can be viewed as a single-parameter strategic agent whose cost is private information, whereas the value of each edge is known to the mechanism. For a set of agents/edges S , if we define as $v(S)$ to be the maximum matching that can be derived by using only the edges of S , then we have a mechanism design version of Budgeted Max Weighted Matching with an XOS valuation function. We note that $v(S)$ cannot be expressed as a submodular function, and therefore none of the results from Section 3.2.1 are directly applicable here.

Regarding the mechanisms, both the deterministic as well as the randomized algorithm are based on a greedy procedure, referred to as Greedy-ISK in [1], which however differs from the greedy schemes presented in Section 3.2.1. This procedure is parameterized by a deterministic exact algorithm f for the unbudgeted Max Weighted Matching problem (e.g., Edmond's algorithm). The main idea behind the mechanism is that in each iteration there is an implicit common upper bound on the rate that determines the payment of each winner in the candidate outcome of the iteration. If the candidate matching is budget feasible the algorithm stops, otherwise it moves to the next iteration with a smaller subset of agents. More specifically, it was suggested that having the agents sorted in decreasing order of cost per value, the common payment per value for every winner provided that iteration i , producing a candidate matching M with total value $v(M)$, is upper bounded by $\min\{\frac{B}{v(M)}, \frac{c_{i-1}}{v_{i-1}}\}$ and this bound decreases (at the same time the set of agents also decreases by one in every step) as the procedure continues, until the set of remaining agents form a budget feasible solution. Greedy-ISK is proven as monotone (hence truthful and individually rational) and budget feasible.

To complete the mechanism, one more corrective step is needed for the cases when

there is a single agent with a very high value. A good compromise can be achieved first in a randomized way; returning the solution produced by the above described procedure with probability $2/3$ and simply the agent with the maximum value (as a matching with 1 edge, and paying him exactly B), with probability $1/3$, yields a 3-approximation randomized mechanism. That algorithm can be derandomized achieving a slightly worse (but still far better than any existing result for the general XOS case) approximation ratio of 4. The derived deterministic mechanism returns the best solution between the agent that achieves the maximum value, say i^* , and the solution produced by Greedy-ISK when agent i^* is not present.

The main results can now be summarized as follows:

Theorem 3.10 ([1]). *There exists a universally truthful, budget feasible, polynomial time randomized mechanism, achieving a 3-approximation in expectation for the Budgeted Max Weighted Matching problem.*

Theorem 3.11 ([1]). *There exists a truthful, budget feasible, polynomial time deterministic mechanism, achieving a 4-approximation for the Budgeted Max Weighted Matching problem.*

Using the same ideas as in the Budgeted Max Weighted Matching problem, one can generalize there results for other problems that can be formulated as ISK problems. The main insight is that we can parameterize the above two mechanisms by replacing the matching algorithm f in the description of Greedy-ISK, by a $\rho(n)$ -approximation algorithm for the unbudgeted version of any other ISK problem. This also requires an adjustment of the relevant probabilities from $2/3$ to $\frac{2\rho(n)}{2\rho(n)+1}$ and from $1/3$ to $\frac{1}{2\rho(n)+1}$. This generalization yields the following results:

Theorem 3.12 ([1]). *If a deterministic $\rho(n)$ -approximation algorithm f for the unbudgeted version of an Independence System Knapsack problem is given as an auxiliary input to Greedy-ISK, then there exists a $(2\rho(n)+1)$ -approximate randomized (resp. $(2\rho(n)+2)$ -approximate deterministic), truthful, budget feasible mechanism. Moreover, if f runs in polynomial time, so does the overall mechanism.*

We can now move to the general case of arbitrary XOS valuations. This is a much harder problem, and the first mechanism that worked for any XOS function was a randomized one provided by [7] achieving a 768-approximation, which does not even run in polynomial time. It also needs to utilize the more powerful demand query model, as explained in Section 2.3, for accessing the valuation function. The mechanism picks uniformly at random either the most valuable agent as the unique winner paying him exactly the budget or the winning set of an auxiliary procedure. The procedure samples in expectation half the agents and computes the optimal solution of the restricted instance which roughly indicates the optimal value and thus can be used to compute a threshold for accepting or rejecting as possible winners, sets formed by the remaining agents. It is stressed that the mechanism runs in polynomial time for certain classic XOS problems but in exponential time for general XOS functions.

A better fine-tuning of the above algorithm was later suggested by [26], which achieves an approximation factor of 436. Yet another modification of the algorithm by [7], was presented in [2], achieving a factor of 244, which is still, upon our knowledge, the currently best ratio. The latter mechanism has also a similar structure with the one presented above for Budgeted Max Weighted Matching, in the sense that with a small constant probability p , it picks the most valuable agent, paying him with the amount of B , whereas with the remaining probability, $1 - p$, it runs a sampling procedure.

Theorem 3.13 ([2]). *Under the demand query model, there exists a truthful, budget feasible, randomized mechanism, achieving a 244-approximation for XOS valuation functions.*

3.2.3 Subadditive valuations

Last but not least, we also briefly examine the case of subadditive valuations. This is a richer class where results have been rather scarce. The only general results that are known are two mechanisms (one deterministic and one randomized), both introduced by Dobzinski et al. [14], where access to a demand oracle is assumed for the valuations, given that there are already known inapproximability results under the value oracle model [35].

We will now present the main idea behind the randomized algorithm of [14], which only needs a polynomial amount of demand queries (and it actually can be derandomized). One of the key components is a procedure for finding a bundle S_v of size t such that the value of any bundle of the same size would be at most $4v(S_v)$. The randomized budget feasible approximation mechanism is as follows. Firstly it selects the agent $i^* = \arg\max_k v(k)$. It then creates sets of agents S_t for each $t \in \{1, 2, \dots, 2^{\lceil \log n \rceil}\}$, using the above procedure (for a carefully chosen subset of items such that it does not contain i^* and the cost is upper bounded by $\frac{B}{2t^{\lceil \log n \rceil}}$). Finally it chooses universally at random between i^* and $\cup_t S_t$. The derandomization of that mechanism uses a *monotone estimator* for the value of the union, and it then selects the option achieving the highest value between the solution provided by the estimator and agent i^* . We note that in both algorithms, the winning participants are paid the threshold payments according to Myerson's lemma [29], and now we summarize the results as follows:

Theorem 3.14 ([14]). *There exists a randomized mechanism with a polynomial number of demand queries, that is universally truthful, budget feasible and provides an expected approximation ratio of $O(\log^2 n)$.*

The same work also provided a deterministic mechanism, albeit with a slightly worse guarantee of $O(\log^3 n)$. However, by exploiting the results of the follow up work by [5], one can get an improvement that matches the same approximation ratio for randomized mechanisms.

Theorem 3.15 ([14] and [5]). *There exists a deterministic mechanism with a polynomial number of demand queries, that is truthful, budget feasible, and its approximation ratio is $O(\log^3 n)$.*

3.2.4 Impossibility Results under Budget Constraints

In the subsections 3.2.1-3.2.3, we presented positive results, i.e., mechanisms where we could provide an upper bound on the approximation guarantee that is achievable in worst case. In this section, we present results on the opposite direction, that demonstrate the limitations of truthful mechanisms. Hence, the central question of this section is:

For a given class of valuation functions, can we provide lower bounds (inapproximability results) on the worst case guarantee for any truthful and budget feasible mechanism?

The first impossibility result was provided in [35] for additive valuations functions. Namely, it was shown that no deterministic, truthful and budget feasible mechanism can have an approximation guarantee better than 2 by the following argument: consider an instance with two agents, both of unit value (thus, the value of selecting both agents together is 2). If their costs are $(B - \epsilon, B - \epsilon)$, then at least one agent should be selected (and fits in the

budget), otherwise the approximation ratio is infinite. Without loss of generality, we can assume that the first agent wins, and then her payment is at least $B - \epsilon$. Now consider another profile where the costs are $(\epsilon, B - \epsilon)$. By truthfulness, the first agent should win again, and receive a payment of at least $B - \epsilon$. Then, the second agent cannot be selected, because of the budget constraint and individual rationality. Therefore, the mechanism can only achieve a total value of 1 for this instance, whereas the optimal solution is 2. This gives us the lower bound of 2 for deterministic mechanisms.

In [10], the above lower bound was improved to $1 + \sqrt{2}$ for deterministic mechanisms, and additionally, a lower bound of 2 was given for randomized truthful mechanisms. What is interesting about these results is that they are unconditional, i.e., they are not dependent on any complexity assumption, like $P \neq NP$, or any assumption on the running time of the mechanism. Hence, these results do not come from complexity reductions, but instead they are based on constructing certain families of instances that show why truthfulness and budget feasibility cannot be compatible with the given approximation guarantees.

Finally, we also note that these impossibility results are given for additive valuations, which implies that they also carry over to submodular, XOS and subadditive functions. Below we summarize the known results.

Theorem 3.16 ([10]). *There is no deterministic, truthful, budget feasible mechanism that can achieve an approximation ratio better than $1 + \sqrt{2}$, even if there are only three agents. Furthermore, for randomized truthful mechanisms, the approximation ratio cannot be better than 2, even in the case of two agents.*

As we move to richer classes of valuation functions, such as XOS or subadditive functions, it is natural to expect that we can get stronger lower bounds. Indeed, with XOS functions, the underlying problem is hard even if we drop the requirement for truthfulness. To begin with, suppose we want to find an algorithm to approximate the optimal value of an XOS function subject to a budget constraint, under the value query model. The following result shows the severe limitations of the value query model.

Theorem 3.17 ([35]). *For any constant $\epsilon > 0$, let A be any (possibly randomized) $n^{1/2-\epsilon}$ -approximation algorithm for maximizing an XOS function subject to a budget constraint. Then A requires exponentially many value queries (in expectation if A is randomized).*

Lately, this has been strengthened when the valuation function is not monotone to $n^{1-\epsilon}$ by [3]. This means that when we allow non-monotone valuations, then essentially we cannot get any non-trivial approximation with value queries.

Theorem 3.17 shows the necessity of using the more powerful demand query model, as we have already demonstrated in the earlier subsections. Under demand queries, the impossibility results for the purely algorithmic version are less severe. The bound shown in the following theorem by [5], is valid for both randomized and deterministic algorithms for XOS and subadditive functions (it suffices to establish it for XOS, the smallest of the two classes).

Theorem 3.18 ([5]). *Fix some constant $\epsilon > 0$. Let A be a (possibly randomized) $(2 - \epsilon)$ -approximation algorithm for maximizing an XOS function subject to a budget constraint. Then A makes exponentially many demand queries.*

It is yet unclear if Theorem 3.18 can be strengthened when we add the truthfulness requirement. Hence, in combination with the positive results under demand queries from Sections 3.2.2 and 3.2.3, there is still a large gap with ample room for improvement,

for the best approximation guarantees of truthful, budget feasible mechanisms (which is between $2 - \epsilon$ and 244 for XOS functions, and between $2 - \epsilon$ and $O(\log^2 m)$ for subadditive functions) under demand queries.

3.3 Models without Budget Constraints

Section 3.2 reveals that a significant part of the literature has focused on mechanism design under budget constraints for the total payments issued by the auctioneer. A complementary direction, which is also very natural in several applications is to drop this requirement, since the auctioneer may not always have a hard limit on his expenditure.

In the absence of budgets, the main objective presented in Section 3.2 of maximizing the auctioneer's valuation function is no longer meaningful. Instead, it is more appropriate to focus on other objectives, which incorporate either the true costs of the agents or the sum of payments by the auctioneer into the objective function. If we still keep truthfulness as a hard-wired requirement, there have been two major approaches that have been followed. The former concerns the design of truthful, no-deficit mechanisms, i.e., mechanisms where the auctioneer never has to pay more than his value for the outcome (the auctioneer's final utility is non-negative). The latter involves the (possibly approximate) maximization of the produced social welfare, which equals the auctioneer's value minus the total costs of the agents for carrying out the tasks. We will provide a brief overview over both approaches within this section.

3.3.1 Design of No-deficit Mechanisms

One of the first works on this topic was by [45]. The model considered there assumes a submodular valuation function for the auctioneer. The final utility of the auctioneer, if he selects a set S of workers and pays each worker i an amount of p_i is

$$u(S) = v(S) - \sum_{i \in S} p_i$$

The mechanism that is proposed in [45] is once again based on a greedy scheme, by trying to select at each step the worker that has the best marginal effect on the total welfare. However this time, in contrast to Section 3.2, one needs to worry about enforcing the no-deficit property, hence one cannot accept right away the choices made by the algorithm. This is done by one additional control check, as illustrated in the mechanism below. For brevity, we only present the allocation part of the mechanism, since we assume as before that the payments are the threshold payments, determined by Myerson's lemma.

GREEDY-NODEFICIT($A, v, \mathbf{c}$) [45]

```

1 Set  $S = \emptyset$ , and  $i = \arg \max_{j \in A} [v(S \cup j) - v(S) - c_j]$ 
2 while  $c_i < v(S \cup i) - v(S)$  do
3    $S = S \cup \{i\}$ 
4   if  $S \neq A$  then
5      $i = \arg \max_{j \in A \setminus S} [v(S \cup j) - v(S) - c_j]$ 
6   else
7     exit
8 return  $S$ 
```

The mechanism stops right when it realizes that the agent with the currently largest marginal contribution has a high cost (higher than its contribution). This enforces that

the threshold payments will never exceed the total value derived for the auctioneer, and hence establishes the no-deficit property. If the mechanism admitted such high cost agents to the final set of selected workers, then by individual rationality it would have to assign a payment that would exceed the agent's marginal contribution and therefore the auctioneer might have been in trouble of deriving negative utility. As for truthfulness, it is quite easy to see that the allocation algorithm is monotone, therefore Myerson's lemma is applicable. To summarize, we can have the following result.

Theorem 3.19 ([45]). *The mechanism GREEDY-NODEFICIT is truthful, individually rational, no-deficit, and runs in polynomial time.*

3.3.2 Mechanisms for Social Welfare Maximization

A different approach to the design of no-deficit mechanisms, and somewhat complementary, is the maximization of the derived social welfare. We recall that the (utilitarian) social welfare equals the auctioneer's derived value minus the costs incurred to the agents. This is a well established criterion of economic efficiency in auctions and it is natural to consider it in this context as well.

One of the first works that focused on this metric is [16] for a simple setting where the auctioneer has an additive valuation function and even further obtains the same value v for every completed task. Each worker is capable of performing any of the available tasks but may only be allocated one task. For this simple case, the welfare can be maximized by a very simple greedy algorithm, of essentially picking the cheapest workers.

A richer model was studied in [34], where an additive valuation is also assumed for the auctioneer, but there is a possibly different value derived for each task. Regarding the workers' capabilities, the authors adopt the model of [45], so that each agent i can perform a set of tasks Γ_i at a cost c_i . Compared to [16], the problem in this case becomes NP-hard, and the main result of [34] is a heuristic algorithm that is based on an Integer Programming formulation of the problem. In particular, the maximization of social welfare is equivalent to solving the following integer program.

$$\begin{aligned}
& \max && \sum_{j \in M} v_j x_j - \sum_{i \in A} b_i y_i \\
\text{s. t.} &&& x_j \leq \sum_{i: \tau_j \in \Gamma_i} y_i && \forall j \in M \\
&&& x_j \in \{0, 1\} && \forall j \in M \\
&&& y_i \in \{0, 1\} && \forall i \in A
\end{aligned}$$

In the above formulation, v_j is the value derived by the auctioneer when the task j is completed, for every $j \in M$. Given that the integer program cannot be solved in polynomial time, the authors solve instead the LP relaxation of the program and then use the fractional solution to propose a greedy algorithm. Namely, the workers are sorted according to the fractional value y_i from the optimal LP solution, since the larger their value, the more likely it is to increase the profit of the platform, and hence the more contribution that user can provide. Then each worker in turn is selected as long as she has a positive marginal effect on the current social welfare. Their algorithm is called

ProMoT and its pseudocode follows.

ProMoT($M, v, \{\Gamma_i, b_i\}_{i \in A}$) [34]

```

1 Initialize  $W = \emptyset$ 
2 Solve LP relaxation of the problem to obtain optimal  $x^*, y^*$  solutions
3 Sort workers in non increasing order of their value in  $y^*$  and form their index list
4 for  $w \in A$  according to the index list do
5   | Select the corresponding worker  $w$ 
6   | if  $v(W \cup \{w\}) - v(W) - b(w) > 0$  then
7   |   | Add  $w$  to  $W$ 
8 return  $W$ 

```

Interestingly, ProMoT turns out to be monotone, and hence we can use Myerson's lemma to turn this into a truthful mechanism. Finally, although the authors do not provide an approximation guarantee for the produced social welfare, the experimental evaluation shows that the algorithm behaves quite well and obtains close to optimal solutions.

4 Generalizations and Extensions

In this section, we will examine some generalizations made to the basic models that have been discussed so far.

4.1 Decomposable Tasks

Most of the studies we have referred to, assume that each crowdsourcing task is an atomic task that can be completed by one service provider and cannot be further decomposed into other smaller sub-tasks. However, in some settings, a single service provider may not be able to finish a task, as it may require the cooperation of more than one service providers.

Here we consider such scenarios with a set of jobs $\mathcal{J} = \{j_1, j_2, \dots, j_l\}$, where the l jobs are distinct and each job is indivisible (i.e., not further decomposed into smaller jobs). The auctioneer announces the set of crowdsourcing tasks $\mathcal{T} = \{t_1, t_2, \dots, t_m\}$, where each of the tasks is a multi-subset of $\mathcal{J}$, showing how the task decomposes into jobs. The valuation function of the auctioneer is assumed to be additive with respect to the set of tasks completed. Finally, it is assumed that each agent i is capable of working on a set of jobs J_i (which is again a multi-subset of $\mathcal{J}$). The cost c_i for the agent i to work on J_i , is private.

The above model is discussed by Zhang et al. [47] and is referred to as Single-requester Single-bid (SS-Model). Viewed as a reverse auction, each worker i submits its bid $B_i = (J_i, b_i)$ to the platform, where b_i is the payment player i asks in order to complete J_i . Here, it is assumed that only b_i is private information to worker i , whereas the skill set J_i is known. In [47], one of the main contributions is a no-deficit, individually rational, truthful and polynomial time mechanism. The mechanism operates by first selecting a subset of tasks that satisfies two conditions: 1) selected tasks can be finished by all of the providers and 2) no *monopoly provider* exists over the selected tasks. A provider is called a *monopoly provider* if its elimination would make the rest of the providers unable to finish the tasks. The tasks are selected iteratively in a greedy manner based on which one has the largest value.

Next, the mechanism determines the allocation of the winning providers for completing the selected tasks, and their payments. The selection proceeds iteratively. In each iteration a provider is selected according to the minimum ratio of ask to the number of jobs that can be used to complete a task. Finally, once the greedy allocation rule has finished, the payments are calculated according to Myerson's Theorem [29], since the allocation algorithm is proved to be monotone.

One extra step is needed though, to check that the total valuation of the tasks completed is greater than the sum of the payments, in order to ensure that this is indeed a no-deficit mechanism.

4.2 Multi-parameter Agents

The models that have been discussed can be further generalized by adding more parameters to the problem. One way to do this is by introducing temporal information regarding the availability of agents. Such a model was presented in the work of Feng et al. [16], where the tasks as well as the agents arrive dynamically to the system. In this model, time is divided in slots of equal size, and tasks arrive at random and dynamically to the system. Each task can be completed within a single slot, it is assigned to at most one user and can be processed by any available user. The model uses the reverse auction framework already described.

The reverse auction is executed round by round. Each round is of equal size, containing m time slots. Within each round, tasks and agents can join dynamically. We will describe within this subsection the offline case that concerns each round, and where the future availability of the agents can be known in advance. In particular, each agent i is asked to submit a bid in the form $B_i = (a_i, d_i, b_i)$, where a_i denotes the begin of the active time for agent i (arrival time), d_i is the end of his active time (departure time), and b_i is the claimed cost for providing the service. All three parameters are assumed to be private information and can be misreported in the aim of profit maximization. The system then determines the allocation rule and the payments for the winning agents, and this analysis is applied to each round separately. As for the auctioneer, the valuation function is taken to be symmetric additive, meaning that for a given set S , $v(S) = v \cdot |S|$, for some constant v . Hence, the auctioneer derives a value of v for every task completed.

Under this setup, a truthful, individually rational and computationally efficient algorithm for the problem is presented in [16]. In the first step the problem is transformed to a maximum matching problem. For a bipartite graph G with two distinct sets of vertices X, Y , each task is a vertex in X and each agent a vertex in Y . Each pair of vertices share an edge with weight equal to the value of the task minus the posted cost from the agent, if the agent is available in the corresponding slot, otherwise, the weight is 0. In the next step the algorithm finds the maximum weighted matching M by employing the Hungarian algorithm [25]. Based on M , the task allocation is determined as follows, for each edge (u, v) in M the task represented by u is assigned to the agent represented by v . In the final step, the algorithm determines the payments of the winning bids, by running the well known VCG mechanism [32]. To summarize, we can have the following result.

Theorem 4.1 ([16]). *There is a truthful, individually rational and computationally efficient mechanism for the maximization of social welfare, under the multi-parameter model of [16], for symmetric additive valuation functions.*

We now move on to a different multi-parameter model, by allowing agents to submit multiple offers on different subsets of tasks. Such a model was presented in [47], and was

referred to as Single requester-Multiple bid (SM-model), with the additional assumption that each agent can be assigned to work on at most one of the subsets of tasks he bids for. The model is almost identical with the SS-model described in Section 4.1, the only difference being that each agent submits a set of bids $A_i = \{(J_i^1, a_i^1), (J_i^2, a_i^2), \dots, (J_i^{\sigma_i}, a_i^{\sigma_i})\}$, where J_i^k is a multi-subset of $\mathcal{J}$ and a_i^k is the declared payment asked in order to complete J_i^k . For each $i \in \{1, 2, \dots, n\}$ and each $k \in \{1, 2, \dots, \sigma_i\}$, there is an actual cost $c_i^k \geq 0$, which is private information, only known to agent i . Hence, each agent i has σ_i private parameters.

The proposed algorithm works just like the SS-model. The platform announces the set of tasks $\mathcal{T}$, and each provider P_i submits its bids $B_i = \{(J_i^1, b_i^1), (J_i^2, b_i^2), \dots, (J_i^{\sigma_i}, b_i^{\sigma_i})\}$ to the platform. Upon receiving the bids, the platform selects the set of winning providers $W \subseteq P$, assigns each $i \in W$ to work on exactly one of the offers made by i , say $J_i^{\delta(i)}$ where $\delta(i) \in \{1, \dots, \sigma_i\}$. The allocation algorithm is a greedy algorithm, extending essentially the idea described in Section 4.1. In the SM-model, each provider can submit multiple bids with multiple sets of jobs that he is willing to work on. Therefore, in the allocation algorithm, the mechanism not only selects the winning agents, but also determines the specific job set that each winner will work on. More specifically, the selection algorithm proceeds iteratively. In each iteration, the algorithm selects the job set $J_{i^*}^{k^*}$ with the minimum ask value and takes its owner i^* as a winner. Player i^* is assigned to work on $J_{i^*}^{k^*}$ and will not be assigned to any other jobs throughout the algorithm.

Regarding the payments, the problem now is that one cannot apply directly Myerson's lemma, since we are not in a single-parameter setting. However, by adjusting carefully, the ideas behind 2nd price auctions, it can be established that there exist payments so that no deficit is created and for each i , we have $p_i \geq a_i^{\delta(i)}$. Hence, at the end, each winning worker i is assigned to complete precisely the tasks in $J_i^{\delta(i)}$, and receives the payment p_i from the platform. This results in a polynomial time, no-deficit, individually rational, and truthful mechanism, as summarized below.

Theorem 4.2 ([47]). *There is a truthful, no-deficit, individually rational and computationally efficient mechanism for additive valuation functions in both the single and the multi-parameter model, under the condition that for the multi-parameter setting the winning bidders are assigned to only one of their bids.*

Finally, in [47], an even more general model is presented, referred to as the MM-model. Here there are multiple service requesters and workers as well. The platform working as the auctioneer, receives bids both from requesters and providers. By using similar algorithms as described in the previous models, we can have truthful algorithms for selecting the set of winning requesters and winning providers.

4.3 Online Models for Crowdsourcing Markets

In Section 3, we assumed that all the information needed by the mechanism to decide which agents to select is available right from the beginning. This is a reasonable assumption if one wants to study an isolated snapshot of the problem at a given time. However, if one wants to investigate how does a mechanism perform over a period of time, it is very natural to resort to online models, where agents may have different arrival and departure times which are not known in advance, and the auctioneer needs to make decisions based on the input seen so far. In the sequel we will study two main research directions that have been explored.

4.3.1 Online Truthful Mechanisms

Even in dynamic settings, where agents or tasks arrive at different times, one may still consider the question of designing a truthful mechanism. Again, the appropriate use of greedy algorithms is crucial for enforcing the right incentives. Regarding performance guarantees, in online algorithms, one cannot expect to be able to always find the optimal solution, given the uncertainty about the input. Therefore, in a similar spirit as in offline approximation guarantees, one is interested in algorithms whose solution is within some bound from the exact offline optimal solution, i.e., using the rather strict benchmark of the best solution if we knew in advance all the input. The minimum such bound is referred to as the *competitive ratio* of an online algorithm.

Starting with the budget-constrained model from Section 3.2, one of the first works to consider an online version of the problem was [36]. In their setting, called *random arrival model*, the auctioneer has a set of tasks with a common completion deadline T , and workers are i.i.d., arriving randomly according to some distribution on the possible time slots $t \in \{1, \dots, T\}$. The valuation function of the auctioneer is symmetric additive, i.e., a special case of an additive function, namely $v(S) = |S|$. This means that the auctioneer cares for maximizing just the number of completed tasks. Hence, the overall goal is to design a truthful mechanism that respects the budget constraint and is competitive with respect to the value derived for the auctioneer. The main result in [36] applies one of the typical approaches in online algorithms, where the input is sampled during the first half of the time slots, in order to estimate a threshold price. This is then used for selecting the winning workers. This approach yields a constant approximation to the optimal. The constant is rather large, but still, this is one of the first positive results, in the context of online algorithms, and provides some optimism that one can also tackle more involved valuation functions.

Theorem 4.3 ([36]). *Under the random arrival model of [36], and for symmetric additive valuations, there exists a truthful, budget feasible, and $O(1)$ -competitive mechanism.*

Some extensions to richer valuation functions have been obtained in [6], which focused on the performance of *posted price* mechanisms for procurement auctions. Posted price mechanisms form a special case of simple truthful mechanisms where agents are offered a payment not dependent on their bid (a different payment may be offered to each agent), who need in turn to decide whether to accept or reject the offer. Posted-price mechanisms may have worse guarantees than other more refined truthful mechanisms. However, it is a simple and intuitive solution that can be competitive in certain special cases.

In [6], it is assumed that the workers arrive in an online fashion and 3 different variations are examined with respect to the order of arrival: the *adversarial* model, the *secretary* model, and the *i.i.d.* model. In the adversarial model, the worst case costs and arrival order are controlled by an adversary, whereas in the secretary model, the arrival order is assumed to be a random permutation (but the costs of the agents may still be controlled by an adversary). Finally, in the i.i.d. model, the cost of each arriving agent is drawn independently from a common distribution. Naturally, the adversarial case is the hardest to tackle, whereas the i.i.d. model provides more information that could be exploited by a mechanism.

The main results of [6] are two posted-price mechanisms, where the first one achieves $O(1)$ -competitiveness for symmetric submodular valuation functions under the i.i.d. model, and the second one is $O(\log n)$ -competitive for general monotone submodular functions under the secretary model. The main message that can be extracted from these results

is that posted price mechanisms can be a suitable approach for designing truthful mechanisms in online models.

Moving on, [6] also proposed a non-posted price mechanism for submodular valuations. This is based on a similar 2-stage process as in [36], where the input is sampled to estimate an approximation to the optimal solution and a threshold value that determines the final allocation. The main drawback of such approaches is that in order to ensure truthfulness, the set of agents sampled in the first stage has to be rejected, i.e., these agents are only used for estimating the price to be paid by the other agents. If they are not rejected, there may exist incentives for misreporting and truthfulness cannot be guaranteed. To alleviate this, the work of [49] has addressed the need for the *Consumer Sovereignty* property, saying that any agent can be among the selected workers, if he submits a low enough offer, given the offers of other agents. This is a property that can be more easily enforced in offline mechanisms, but is more challenging in online models where some agents are used only for sampling purposes. To deal with this, [49] proposed a multi-stage sampling process, dynamically increasing the sampling size and keeping track of not exhausting the budget. The following summarizes the established results for submodular valuations.

Theorem 4.4 ([6] and [49]). *For submodular valuations, there exists a truthful, budget feasible mechanism that is $O(1)$ -competitive under the secretary model. Furthermore, the same holds under the i.i.d. model and under the Consumer Sovereignty requirement.*

Yet more progress was recently made in [3] for the secretary model. In particular, the approach of [3] for the offline case, explained in Section 3.2.1, was shown to go through under random arrivals of the agents. This implies that we can have $O(1)$ -competitive mechanisms even for non-monotone submodular valuations. Although the constants that are hidden in the $O(1)$ are rather large in all the results mentioned in this subsection, it is still important that one can get truthful mechanisms whose competitive ratio does not depend on the number of agents or tasks.

Finally, apart from the budget-constrained setting, online models have also been considered for the case where there is no hard constraint on the expenditure of the auctioneer. In the absence of budgets, one of the main objectives is to optimize the generated social welfare. This is addressed in [16], which studies a model with both tasks and agents arriving at random within a given period of time. The authors focus on the special case of symmetric additive valuations, where for every set S , $v(S) = v \cdot |S|$, for some constant v . The main result in [16] is a simple greedy algorithm, showing that under the i.i.d. model, one can have a truthful, $1/2$ -competitive mechanism with respect to social welfare maximization. Obtaining such small constant approximation guarantees for richer valuation functions is still not known.

4.3.2 Learning-based Mechanisms

A related approach that is appropriate for online models is the use of ideas from machine learning, such as multi-armed bandit methodologies, the exploration-exploitation framework and no-regret algorithms. Under such mechanisms, one tries to strike a balance between exploring further the agents under consideration, and exploiting the knowledge the algorithm has already obtained from previous rounds. As a result, one may relax the requirement of truthfulness, but still ensure that agents will have negligible regret for being truthful.

In [37], the i.i.d. model is analyzed under this framework, where agents are assumed to arrive one at a time, and their costs are drawn from the same distribution independently.

The paper proposes a greedy posted-price mechanism, where the price slightly increases from one round to the next. The algorithm fits under the multi-armed bandit framework, where the workers correspond to the "arms" that the auctioneer can pull in each round in order to explore the available options. The main theoretical results of [37] is that for the case of additive valuations under budget constraints, their algorithm achieves an average regret that tends to 0. Furthermore, the performance of the algorithm is examined experimentally, and is compared against various benchmarks. Among them, the two major benchmarks are the best offline optimal solution, as well as the best offline fixed price solution (offering the same reward to all selected agents). It is demonstrated that the no-regret algorithm performs very well and comes near these two benchmarks on average.

Another approach was taken by [22], where they also look at additive valuations and propose an algorithm based on the primal-dual formulation of the problem, inspired by a similar analysis for auctions in online advertizing. The online primal-dual framework yields a two-stage exploration-exploitation framework, which outperforms greedy algorithms in this setting. By using the dual program (which is equivalent to optimizing the primal program), the first phase consists of a number of exploration rounds where the algorithm learns skill levels of the workers. The remaining rounds simply allocate tasks so as to maximize the marginal utility. Apart from the experimental evaluation, it is also proven that the algorithm achieves a competitive ratio that is dependent on the fraction of rounds that is spent on the sampling (exploration) phase.

Finally, yet another work along a similar spirit is [40]. Again, a multi-armed bandit type algorithm was proposed for additive valuations under budget constraints, where the exploration phase is completely distinct from exploitation. Namely, the proposed algorithm is referred to as an ϵ -*first policy*, where an ϵ fraction of the budget is first spent for uniform exploration, and then a greedy knapsack-based algorithm is used for exploitation. The work establishes theoretical bounds on the regret of such classes of algorithms, based on the relevant parameters, while also showing a promising experimental performance of the overall approach.

5 Spatial Models

In this section, we consider types of crowdsourcing markets having spatial attributes and, mainly, where there are explicit requirements for completing all (or a range of) the given tasks. Such a special form of crowdsourcing systems has emerged from applications where the tasks are located in specific positions or require from the workers to physically move to certain locations. This is usually referred to as *spatial crowdsourcing*, and one can even further generalize this by adding the dimension of time and have spatio-temporal constraints. Several scenarios from mobile crowdsourcing or from participatory sensing can fall under this framework, since they require the workers to be in specific locations in order to transmit sensor information from the geographic areas under consideration. Most models in spatial crowdsourcing need first to specify the type of tasks and the location-dependent information. In some scenarios, the task is simply located in a certain position within a metric space, whereas in other scenarios, a task may consist of an origin and a destination, and the assigned worker needs to travel from the former to the latter position. In addition, it is common to have a travel budget, meaning that there can be time limits on the travel time or generally on the cost of traveling from an origin to a destination (if there are deadlines on the completion of the tasks). The workers can also have their own arrival and departure times and in general, all the extensions of the most

basic models presented in Section 4 are applicable here as well.

At what follows, we will mainly focus at the single-dimensional case where, additionally, coverage constraints occur. For more general models, very recently, two relevant surveys have been published in [39] and [20], which cover to a big extent the literature on spatial crowdsourcing. Therefore, at the end of the section, we will only briefly comment on some of the defining features in such systems and we refer the reader to these two sources for a more complete exposition on related research.

To motivate the setting that we discuss in the sequel, we note that many real-world crowdsourcing applications have strong requirements of data integrity. Suppose that an authority needs to allocate tasks of data retrieval to potential workers. In practice, it is natural to only accept a solution as feasible if it meets a coverage requirement meaning that all tasks publicized by the planner are being accomplished, since otherwise the value of fragmented data may be significantly lower than expected. We will now present some works that define and investigate models with coverage constraints from an algorithmic (non-strategic) and game-theoretic (strategic) point of view. We refer to the corresponding papers for further discussions on the wider applicability of such models.

A scenario with strategic crowdworkers who wish to perform tasks that share a natural linear ordering is being considered in [13]. In the examined model, a planner wishes to hire some workers to perform a set of similar binary labelling tasks which (unlike the tasks on platforms like Amazon Mechanical Turk) require significant cost from the workers' part. It is also assumed that the tasks are ordered on a line (which can correspond either to some sort of geographic proximity, or any other form of linear ordering). Each worker has a (known to the planner) quality level, which comes from the probability of identifying the true label of any task, which can be computed possibly by previous task assignments or review ratings. Furthermore, each agent i is *single-minded* in the sense that she is interested in executing precisely a set S_i of contiguous tasks (according to the linear ordering). This means that S_i forms an interval of tasks and if agent i is selected, then she will execute all the tasks of S_i . Because of the potential existence of low-quality workers there may also be a need of assigning more than a single worker per task so as to successfully execute it (at least up to certain extent, given the specified threshold on inference quality for each task, i.e. a covering constraint). Each worker may act strategically and misreport either her true cost for performing the declared tasks or the set of tasks she can execute. The goal of the proposed interval-cover mechanism is to allocate the tasks to a set of crowdworkers, so as to minimize the overall payment cost, in a way that the quality of the allocated workers to each task meet a predefined threshold.

Starting with the case where all the workers have the same quality (a scenario that is called homogeneous) and considering the problem from its algorithmic viewpoint, it is natural to inspect the Integer Program of the problem as given below. Concerning the notation, let A be the set of workers, T be the set of tasks, b_i be the claimed bid of worker i and for every worker i and task j it is true that $i \in s(j)$ if task j is located in the interval that the worker i is capable of executing. Additionally let n_j be the threshold that should be reached so as task j can be considered as implemented and which can be efficiently computed taking into account the contribution of the workers that belong to $s(j)$.

$$\begin{aligned} \min \quad & \sum_{i \in A} b_i x_i \\ \text{s.t.} \quad & \sum_{i \in A \cap s(j)} x_i \geq n_j, \forall j \in T \end{aligned}$$

$$x_i \in \{0, 1\}, \forall i \in A$$

Interestingly, when the first constraint is written in matrix form, i.e. $Mx \geq n$, and because of the contiguous structure of the workers' bids, it can be observed that M satisfies a property known as *total unimodularity*, which is that in each column, the 1's appear in a contiguous sequence. Linear relaxations of Integer Programs having a constraint matrix M that is totally unimodular admit polynomial time algorithms that yield an integer solution. For the case of workers with different quality (heterogeneous scenario), total unimodularity does not hold anymore and the problem becomes computationally hard even for a single interval since Knapsack can be reduced to it. In that case, a polynomial algorithm that approximates the optimal solution exists since the problem is equivalent to a known optimization problem (referred to as (0,1)-RESALL in [9]), which admits a primal-dual 4-approximation algorithm.

Moving on to the strategic setting, since the homogeneous case can be optimally solved in polynomial time, the VCG payment rule results to a truthful and individually rational mechanism that minimizes the total cost of the workers. Hence, this completely resolves the mechanism design problem for the homogeneous scenario. At what concerns the strategic, heterogeneous setting, the work of [13] provides a monotone allocation algorithm, and thus resulting in a truthful mechanism by using Myerson's payment formula. At the same time, the mechanism also provides an approximation to the optimum total payment cost, but by a factor that is non-constant in the input size. In more detail, the approximation factor is roughly a multiple of the ratio between the maximum and the minimum quality of workers. To our knowledge, no improvements of this result have been proved. The following theorem summarizes this discussion:

Theorem 5.1 ([13]). *The task allocation problem under the interval coverage constraints is*

- *optimally solvable in polynomial time in the non-strategic setting of the homogeneous scenario and, as a consequence, an optimal truthful, individually rational mechanism exists for its strategic counterpart.*
- *4-approximable in polynomial time in the non-strategic setting of the heterogeneous scenario and its strategic counterpart is $2^{\frac{p_{\max}-1}{p_{\min}-1}}$ -approximable using a computationally efficient, individually rational, truthful mechanism, where $p_{\max}(p_{\min})$ is the maximum (resp. minimum) quality of any crowdworker.*

A similar problem but with a different optimization objective has been considered in [44]. In that work there is a time-interval S , that contains one task per time unit in it, that needs to be covered with no cardinality or quality constraints. Each crowdworker declares a bid for executing the tasks included in either one or multiple intervals (called time-windows) intersecting with S . Supposing that the workers can only misreport their true costs and not their time intervals (since the interval-truthfulness can be achieved using a verifying protocol), the objective is, as before, to minimize the social cost, or equivalently, the sum of true costs of the selected workers. For the case where every worker bids on a single interval, the algorithmic problem can be efficiently solved using a dynamic programming procedure and thus, the VCG mechanism works for the corresponding strategic setting. On the other hand, allowing workers to work on more than one interval makes the algorithmic problem NP-hard, but a polynomial, individually rational and truthful mechanism that achieves an approximation ratio asymptotically bounded by the natural logarithm of the number of tasks in S exists. More precisely, the main results of this work are the following:

Theorem 5.2 ([44]). *The task allocation problem with the interval coverage constraints, admits*

- *an optimal algorithm in polynomial time in the non-strategic setting of the single time-windows scenario and, as a consequence, an optimal truthful, individually rational mechanism exists for its strategic counterpart.*
- *a computationally efficient, individually rational, truthful mechanism for the multiple time-windows scenario, that is $(\ln |S| + 1)$ -approximate, where $|S|$ denotes the length of the overall time interval, i.e. the number of time units contained in S .*

Moving a step away of the intervals and regarding the optimization objectives, in a similar spirit as before, there is often a coverage requirement for completing all tasks pertaining to a specific Region of Interest (RoI), e.g., collecting sensor information for environmental purposes in a given region, or collecting traffic information from the downtown area of a city. As a result, and in contrast to previous sections, it may not be meaningful to maximize the auctioneer’s valuation function, but instead it is more natural to minimize the total cost for satisfying the coverage requirements and for completing all the tasks. In parallel to this, when there are no strict coverage requirements, then one can still utilize the performance metrics that we saw in Section 3, such as the maximization of the auctioneer’s utility or the derived social welfare, see among others [21].

Finally, regarding incentive mechanisms, one could insist as before on truthful mechanisms, such as the work of [15] for location aware tasks. However, we stress that in contrast to Sections 3 and 4, incentive compatibility has not been always a strong requirement in related works. One of the reasons is that many of these problems are highly related to well known geometric or graph-theoretic optimization problems (including covering and/or routing problems) and the focus of many previous works has been on appropriately adjusting these models to work for crowdsourcing as well. Another reason is that due to the geometric nature, travel times and costs are often verifiable, and hence the workers may not be able to misreport their costs. Some pricing mechanisms however have been explored, and though not truthful in worst case, they try to ensure individual rationality or other guarantees, as demonstrated in [39] and [20].

6 Reputation-based Mechanisms

So far, the workers have been assumed strategic in their cost declaration for winning the different auction mechanisms; at the same time, the workers were always assumed to deliver a service of acceptable quality. In this section, the quality of contribution is determined based on the hidden type of the worker or based on the hidden behaviour (i.e., costly effort) exerted by the worker while performing a task.

6.1 Reputation-based worker selection and isolation

The paper in [48] considers strategic workers of hidden behavior that may deliver or not services of acceptable quality, while being paid upfront in a *moral hazard* context.

The paper in [48] aims to social-welfare maximization in a setting of strategic workers. Truthfulness of workers is approached as a Pareto-efficient equilibrium strategy in the presence of a social norm based on a reputation scheme for their performance. The model is the common one in online reputation mechanisms, where the requester pays the worker upfront to perform a task and the reputation of the latter depends on its effort in the

transaction. A threshold reputation policy is employed, whereas workers with reputation lower than a certain value are excluded from transactions for a number of rounds.

Different from the traditional models in the design of pricing schemes where the interactions between workers and requesters are usually formalized as a one-shot game, repeated interactions between requesters and workers are captured within the formalism of repeated games. To formalize the reputation scheme, social norms, which consist of a social strategy and a reputation scheme, are introduced to regulate the behavior of workers. An appropriate social norm is designed to punish workers' deviations from the selected social strategy (i.e., exerting high effort), and the workers' equilibrium behaviors based the benefit and cost of tasks as well as their patience are rigorously analysed. Subsequently, the sufficient and necessary conditions under which the workers comply with the prescribed social strategy are quantified.

Formally, the payment p of the requester to the worker is ex-ante, i.e., given up-front, while a portion $1 - \lambda$ is kept by the website as a transaction fee. In a transaction, workers can exert a binary effort $A = \{H, L\}$, where H stands for "High Effort" and L stands for "Low Effort". High effort has a cost c for the worker, while low effort has zero cost. The effort exerted by a worker can be erroneously observed with probability $\alpha \ll 1$. A social norm κ , which is chosen by the protocol designer, is composed of a social strategy σ , a reputation scheme τ , and a reputation set Θ . A social strategy σ determines a service threshold $h(\sigma) \in \{1, \dots, K_\kappa\}$ upon social norm κ , such that workers with lower reputation values are excluded from transactions for $h(\sigma)$ time slots. For a worker of reputation θ that selected an action $a \in A$, the reputation scheme adopted is as follows:

$$\tau(\theta, a) = \begin{cases} \min\{K_\kappa, \theta + 1\} & \text{if } a = H \text{ and } \theta \geq h_\kappa \\ \theta - 1 & \text{if } a = L \text{ and } \theta \geq h_\kappa + 1 \\ 0 & \text{if } a = L \text{ and } \theta = h_\kappa \\ \theta + 1 & \text{if } \theta < h_\kappa \end{cases} \quad (1)$$

As defined in [48], a social norm $\kappa = (\sigma, \tau)$ is an *equilibrium one* if and only if the worker cannot benefit from a single deviation from the social strategy. Also, as defined in [48], a protocol (κ, λ) is *sustainable* if and only if κ is a social norm equilibrium.

Theorem 6.1 (Theorem 1 in [48]). *Any social norm that does not satisfy the following equations delivers a social welfare that is lower than the optimal value of (10) in [48].*

$$\begin{aligned} 0 &\leq \delta(1 - 2\alpha) [v_\kappa^\infty(\min\{K_\kappa, \theta + 1\}) - v_\kappa^\infty(\theta - 1)] - c, \text{ if } \theta \geq h_\kappa + 1, \\ 0 &\leq \delta(1 - 2\alpha) [v_\kappa^\infty(\min\{K_\kappa, h_\kappa + 1\}) - v_\kappa^\infty(0)] - c, \text{ if } \theta = h_\kappa, \end{aligned} \quad (2)$$

$v_\kappa^\infty(\theta^{(t_0)})$ is a compliant's worker expected long-term utility in the repeated game starting from any period t_0 , while δ is the discount factor. The condition in 2 means that a positive incentive guarantees a worker's compliance to the social norm, and a protocol is sustainable if and only if workers have positive incentives at any $\theta \geq h_\kappa$.

Moreover, according to Proposition 1 in [48], given a social norm κ , choosing $\lambda^* = 1$ maximizes the social welfare of the workers.

Also, as proved in Theorem 2 of [48], given p , c , δ , and α , a social norm $k = (\sigma, \tau)$ can be sustained as an equilibrium if and only if its social threshold h_κ is larger than a specific constant $\bar{h}(p, c, \delta, \alpha)$ and the highest reputation K_κ is smaller than a specific constant $\bar{K}(p, c, \delta, \alpha)$.

Theorem 6.2. *Sustainable protocols exist if and only if (1) Given δ and α , the cost-to-price ratio $r \triangleq c/p$ of the worker is lower than a threshold $\bar{r}(\delta, \alpha)$; (2) Given the*

cost-to-price ratio r and α , the discount factor δ is higher than a threshold $\underline{d}(r, \alpha)$; (3) Given r and δ , the error probability α is lower than a threshold $\bar{a}(r, \delta)$, and higher than a threshold $\underline{a}(r, \delta)$.

Another setting were the delivered quality of the worker(s) is uncertain, while the payment per task from the requester is pre-announced is considered in [42]. In this paper, one requester assigns several workers per task. A worker can be an *expert*, i.e., a worker having sufficient skill to complete a specific task, or *novice*, i.e., a worker of insufficient skill to complete a specific task. When the reward is split among workers, free-riding arises on behalf of novice users. A reputation mechanism is proposed where the requester gives a binary rating to all workers of a task and the task reward is split among those getting a positive vote, unless all workers get negative votes when they all split the reward. A reputation system is proposed to alleviate novice free-riding and establish an equilibrium where only expert users accept tasks, while novice users reject them.

More formally, while multiple task types can be considered, without loss of generality, the analysis in [42] focuses at a certain task type. Considering a task of a certain type, a requester pays a transaction fee of $T > 0$ to the crowdsourcing system, and pays a reward of $r > 0$ to be distributed to contributing workers as well. In total, a requester pays $T + r$ in order to have the task completed.

A prior distribution of expertise in the worker community is assumed to be common (public) knowledge, according to which in each task assignment, a task is assigned to a novice with probability $\beta \in [0, 1]$, and it is assigned to an expert, with probability $1 - \beta$. The model in [42] considers that $m \geq 1$ work on a task denoted by $w_1, \dots, w_m$. Let $\theta_i \in \{0, 1\}$ denote the skill type of w_i , where $\theta_i = 1$ implies that w_i is an expert, and $\theta_i = 0$ implies that w_i is a novice. When a worker receives a task assignment, he can refuse (D) to solve it, or commit to solve this task by a high effort (H) or a low effort (L). Hence the action set of a worker is $\{H, L, D\}$. If a worker refuses a task, then this task will be reassigned until m workers commit to solve it. An expert worker exerting a high effort H is with cost c , and a novice exerting a high effort H is with cost c' , where $c' > c > 0$.

According to the analysis in [42], for $m \geq 2$ workers, for expert workers, H is a strictly dominant strategy, if and only if $r > mc$. For novice workers, L is a strictly dominant strategy and gives a utility higher than refusing a task, no matter of reward set.

A reputation system is proposed in [42] to alleviate the *novice free-riding* problem. The provision of low quality is considered as a mistake and the proposed reputation system penalizes a worker when its mistakes exceed a certain threshold $W_g \geq 1$ by disabling the worker's account for W_p time slots. Let s^* denote a desirable strategy for workers where expert workers exert high effort (H), and novice workers refuse tasks (D).

Theorem 6.3. (Theorem 4.2 in [42]) Consider the proposed incentive mechanism (with the reputation system) in [42] that allows $m \geq 2$ workers to work on a task, and a (W_g, W_p) -reputation system. If $W_g = 1$ and $W_p \geq 1$, each player will not unilaterally deviate from s^* , iff $r > mc$.

Also, according to Theorem 4.3 in [42], when $\delta > 1 / \left(1 + \frac{1-\beta}{\beta^{m-1}}\right)$, $W_g = 1$, and

$$W_p \geq \left\lceil \ln \left(1 - \frac{\beta^{m-1}}{1-\beta} \frac{1-\delta}{\delta}\right) / \ln \delta \right\rceil$$

and

$$r > mc \left(1 - \frac{1-\delta}{(1-\delta^{W_p})} \frac{\beta^{m-1}}{\delta} \frac{1}{1-\beta}\right)^{-1},$$

then the subgame perfect equilibrium of the previous theorem is unique.

The work of [42] is significantly extended and generalized in [43], where multiple expertise levels $\{1, \dots, M\}$ and multiple effort levels $\{0, 1, \dots, K\}$ of workers are considered, and a class of incentive and reputation mechanisms is proposed and analyzed. Moreover, a trade-off between system efficiency (i.e., the number of tasks that can be solved for a given reward) and revenue (i.e., the amount of transaction fees) is studied.

In the proposed incentive mechanism in [43], upon posting a task, a requester must submit its associated reward r and transaction fee T to the crowdsourcing system administrator. The task is solved by exactly n workers and solutions are submitted to the administrator, and all are forwarded to the requester. Upon receiving these n solutions, a requester evaluates their quality and selects the one with the highest quality. It is assumed that the requester can only be satisfied if at least one solution has the highest possible quality. If there is a tie, each solution within the tie is selected with equal probability. We refer to the highest-quality solutions as winner(s) and the other solutions as losers. Finally, the requester notifies the system administrator as to the identity of the winners, and the system administrator distributes the reward r evenly to those winners. In this incentive scheme, the requester cannot refuse to pay workers, as the reward r and the transaction fee T are held by the administrator, while a worker with zero effort receives a reward r/n when all n workers exert no effort. This model applies to macrotasking crowdsourcing systems, which serve as platforms to outsource innovative and challenging tasks that require special skills (e.g., develop a computer program), e.g., UpWork, as opposed to microtasking platforms, e.g., Amazon Mechanical Turk.

The probability that a worker of type m is assigned a task is denoted by β_m , while worker w_i 's strategy is a map $s_i : M \rightarrow \Sigma$ prescribing an action for each possible skill type; e.g., $s_i(M) = (0, \dots, 0, 1)$ means that when w_i has skill type M , he always exerts level K effort. Exerting effort level k for a worker of skill type m costs $c_{m,k}$.

As proved by Lemma 3.5 in [43], considering the proposed incentive mechanism with $n \geq 2$ workers assigned a task at least one Bayesian Nash equilibrium exists; given a skill type m , if

$$r > \frac{n\beta_m c_{m,K}}{\left(\sum_{l=1}^m \beta_l\right)^n - \left(\sum_{l=1}^{m-1} \beta_l\right)^n - n\beta_m \left(\sum_{l=1}^{m-1} \beta_l\right)^{n-1}}, \quad (3)$$

then each such equilibrium $(s_1^*(\cdot), \dots, s_n^*(\cdot))$ satisfies $s_i(m) = (0, \dots, 0, 1)$ for all $i = 1, \dots, n$. This result means that there is a minimum bound of task reward, specific per worker of type m , which can motivate that worker to exert the highest possible effort. However, still the required quality by the requester may not be achievable for the n workers, unless at least a worker with maximum expertise is among them.

To alleviate this, a (N_e, N_b) -reputation system is proposed that allows a worker to remain active for N_e "errors", i.e., maximum number of time slots where requested quality is not offered by the worker, and blocks him for N_b time slots when N_e is reached. Under this reputation system, Theorem 4.3 in [43] establishes that when $N_e = 1$ and for properly lower bounded values of the discount factor δ , the blocked window threshold N_b and the task reward r , there is a unique subgame perfect equilibrium where each worker of type M exerts the maximum effort, while all workers of skill types lower than M refuse the task (this strategy is referred to as *desired strategy*). From Theorem 4.3 ([43]), it is suggested that a crowdsourcing provider should set N_b as large as possible, provided that there are no errors in reputation updating. Theorem 5.1 ([43]) suggests that a crowdsourcing provider should decrease N_e (or increase N_b) so as to decrease the reward paid out by requesters, while N_e is upper bounded in order for the desired strategy to be a unique

subgame perfect equilibrium upon noisy ratings. However, the side effect is a decrease in the number of active workers (cf. Theorem 5.3 in [43]).

Moreover, the parameters of the reputation system are found in [43], so that the revenue of the system is maximized. For example, when noisy ratings arise with probability $\epsilon_{M,K} = 0.05$, the solution is $N_e = 6$ and $N_b = 1$, which achieves a revenue of $0.994\tilde{R}$, where $\tilde{R}$ is the theoretically maximum achievable revenue. Finally, in [43], they find parameters N_e, N_b that maximize system efficiency for a minimum acceptable revenue. For example, for a 5% decrease in the acceptable revenue, they find that when $\epsilon_{M,K} = 0.05$, the corresponding reputation system design is $N_e = 5, N_b = 7$, and a reward $r = 3.41c_{M,K}$ is obtained, which attains 0.8976 of the theoretical maximum system efficiency.

6.2 Crowdsourcing with task subdelegation

The paper in [46] introduces another interesting direction of modeling, namely subdelegation of tasks. In particular, each worker that has already been assigned a workload of tasks can subdelegate part of this to other workers on the basis of their reputation. Thus, it may happen that a certain task is executed by a chain of workers. This fact is also taken into account in the mechanism for updating the workers' reputation. In particular, decreasing proportions of weight are applied to consecutive agents in such a chain. The reputation of each worker participating in the chain is updated on the basis of the relevant proportion of the collective outcome of the task. Each task, if completed within its deadline, induces a positive reward at each time to the assigned agent. Moreover, at each time t , each agent is characterized by an instantaneous price $p_i(t)$ for exerting effort. The overall objective is that tasks are subdelegated in such a way that the social welfare, ie. time average of all agent's instantaneous reward is maximized. The trade-off faced by each worker is between the increase of his unfinished workload versus his cost for subdelegating part of this load to other workers. The authors of the article develop a queueing model for dealing with this trade-off. In particular, they introduce an appropriate Lyapunov drift, and subsequently a drift-plus-cost expression, in order to analyze how to control this queueing system in a way the latter drift is minimized. Moreover, when selecting others to subdelegate tasks, workers employ a reputation-threshold policy whereby only workers with a high enough reputation and a low enough price are considered. The performance of this so-called RTS (Reputation-aware Task Subdelegation) approach is initially investigated theoretically. In particular, the policy prescribes that if $q_i(t) - \rho_i p_i(t) r_i(t) \leq 0$, then worker i can accept new tasks requiring up to a certain amount of extra effort, where $q_i(t)$ is his remaining work-load and ρ_i is a parameter indicating the worker's eagerness to work. This condition implies that new tasks are accepted if they are expected to lead to higher rewards and/or if the worker has an appropriately light load. Moreover, this worker i decides whether to subdelegate tasks or not on the basis of another condition involving a bound on the average price ϕ_i charged by the other workers known to him. The authors subsequently establish two performance bounds. Namely, that the social welfare attained by means of this policy is within $O(1/\rho)$ of the optimal, where $\rho = \frac{1}{T} \sum_{t=0}^{T-1} / \sum_i \rho_i$ is the time-average of the collective willingness to work of all workers, while at the same time the pending workload is bounded by $O(\rho)$. The RTS policy is also evaluated experimentally, whereby it is seen that it is significantly more efficient than other related policies. The authors also present briefly certain directions for future research such as the inclusion of policies for the adjustment of each worker's price $p_i(t)$, and priorities and dependencies among tasks.

6.3 Worker selection via endorsement

The paper in [27] proposes an incentive mechanism for high quality contributions and high participation in the participatory sensing context. A participant who contributes sensor data earns a reward in terms of redeemable contribution power (CP) gained based on the quality of its contributions and redeemable endorsement power (EP) which is given by its endorsers. Endorsement power (more) and contribution power (less) can be considered as reputation values employed in this incentive mechanism. Part of the reward proportional to the fraction of borrowed power (BP) over the total power of a contributor is shared back with its endorsers. As a result, endorsers become beneficiaries of the contributor. Contributions require costly effort proportional to their quality. According to [27], if (very likely) a contributor is endorsed by some of his/her “nepotic” social connections, a new incentive is created for the contributor: “work for your cared (or loved) ones” (besides yourself). The requester seeks to maximize its net profit from the aggregated data, while dealing with individually rational, self-interested and potential nepotic contributors. The problem is formulated as a Stackelberg game, where the requester (i.e., the leader) selects redemption rates for CP and EP, and the participants (i.e., the followers) select their contribution qualities, so as to maximize their utilities based on globally available CP and EP values for all participants.

Also, [27] formulates the problem of an endorser to select the set of contributors to endorse, so as to maximize its expected payoff, subject to the social constraint that it will endorse its socially preferred acquaintances. The problem is solved based on efficient partial sorting algorithm with linearithmic complexity on contributors (c.f. Proposition 1 in [27]). Similarly, the problem of a contributor to select its endorsers is formulated so as to maximize its total EP (and consequently its reward).

However, the three optimization problems in [27] are not jointly pursued and equilibrium analysis has not been performed. Current CP and EP values are observed by participants to determine their quality values, which are employed for selecting their contributions and endorsements in the next period. It is unclear whether the proposed incentive mechanism in [27] enhances quality of contributions at equilibrium. It is numerically shown in [27] that requester’s profit and the quality of contributions are increasing in CP and decreasing in EP values. Moreover, it has been numerically demonstrated that participants should be selective in their endorsements.

7 Conclusions and Discussion on Open Problems for Future Research

In this survey, we have attempted to provide an overview of recent research on mechanism design and optimization aspects in crowdsourcing markets. Our main focus has been on research that has established mathematically provable guarantees on the design and evaluation of proposed mechanisms. As a conclusion of this overview, we can say that this has been a very active research field within the last decade and that it will continue to be active, given the emergence of more and more applications and challenges. In the rest of this section, we summarize some of the open research problems and directions for future research.

To begin with, despite the significant progress that has already been made on the basic models and their variants in Sections 3 and 4, it is still an open question to provide tighter approximation algorithms for the relevant classes of valuation functions. This applies mostly for the cases where the valuation function of the auctioneer is XOS or

subadditive, where there are still relatively large gaps between the known approximability and hardness results. It is also possible that some special cases of submodular valuations admit even better approximations than the ones presented in Section 3. Making progress on this front would require a better understanding or even more, a characterization result, for truthful mechanisms in this setting.

A second and very important direction, which is also among the biggest challenges for the markets under consideration, is to understand better the design of mechanisms for multi-parameter agents (where simultaneous offers on different combinations of tasks can be allowed by each worker). As already explained in Section 4.2, there has been a very limited number of works dealing with such types of workers and most models assume that each worker only has a single parameter as private information (corresponding to the cost for performing a specific task or a specific set of tasks). As we see new labour matching platforms arising with the need to express a bigger variety of preferences, we expect that multi-parameter mechanism design will be one of the main research directions within this field. Allowing the workers to express multiple offers on different combinations of tasks leads to more difficult optimization problems, and at the same time, to more demanding considerations regarding the incentives for truthful behavior. One cannot apply any more the characterization by Myerson’s lemma, and therefore, new methodologies are needed for designing mechanisms and enforcing a desirable behavior to the workers.

The study of Bayesian models under this setting, is also worth further investigation. A big part of the works that have been presented here deal with models of complete information. Some generalizations have been provided on mechanism design under the assumption that the workers’ costs are drawn from a distribution, see e.g., [7]. However, we feel that it is a natural next step to study whether we can have the same approximation guarantees under uncertainty, or whether Bayesian models impose more impossibility results.

Moving on, and given the fact that one cannot always satisfy all desirable criteria defined in Section 2.1, yet another interesting direction is to define and examine relaxations of the problem requirements. For example, under budget constraints, one could seek for a model where instead of a hard budget limit, the solution cost is allowed to exceed the initial budget by some factor. Along the same spirit, a well-motivated question is to study relaxations of truthfulness. It has been well observed that truthfulness may be too stringent as a hard-wired property in practice. One can therefore study some alternative options, such as randomized mechanisms that are truthful only in expectation, or mechanisms that are almost-truthful or partially truthful, i.e., they provide incentives to agents to misreport their value only by a tiny fraction or only with respect to certain classes of bids - which may eventually not even occur, such as overbidding. Works on verification mechanisms that identify whether a bid is truthful or how far from the real value it is, can also be proved to be useful for discouraging agents from misreporting their preferences in mechanism design problems.

Finally, with the exception of Section 6, we have not addressed much here the issue of verifying that the winning agents perform their tasks correctly. Most of the works that have been discussed in this survey have assumed that once a worker is assigned a task, she will also complete it as promised. In cases where the auctioneer is not able to choose specific agents, as in MTurk, there have been works that try to minimize the estimation error (e.g., for classification tasks). This was not the focus of our survey, but it is an active topic, which is also in a sense complementary to the goal of truthful behavior in mechanism design. For more on this, we refer to [41]. Also, a more behavioral approach is used in [38], where the auctioneer’s utility depends on the effort level of the workers.

To conclude, we believe that the study and design of crowdsourcing markets is a fascinating area, requiring the interplay between algorithmic and game-theoretic approaches. The field provides ample space for future research and for the emergence of new techniques that could be applicable in other domains of mechanism design as well. We hope that this survey can serve as a guide for the state of the art especially with respect to computational complexity and game-theoretic questions on the design of efficient mechanisms.

References

- [1] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. Coverage, matching, and beyond: New results on budgeted mechanism design. In *Proceedings of the 12th International Conference on Web and Internet Economics, WINE 2016*, pages 414–428, 2016.
- [2] Georgios Amanatidis, Georgios Birmpas, and Evangelos Markakis. On budget-feasible mechanism design for symmetric submodular objectives. In *International Conference on Web and Internet Economics (WINE 2017)*, pages 1–15, 2017.
- [3] Georgios Amanatidis, Pieter Kleer, and Guido Schaefer. Budget-feasible mechanism design for non-monotone submodular objectives: Offline and online. In *Proceedings of the 20th ACM Conference on Economics and Computation, EC 2019*, pages 901–919, 2019.
- [4] Nima Anari, Gagan Goel, and Afshin Nikzad. Mechanism design for crowdsourcing: An optimal $1-1/e$ competitive budget-feasible mechanism for large markets. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014*, pages 266–275, 2014.
- [5] Ashwinkumar Badanidiyuru, Shahar Dobzinski, and Sigal Oren. Optimization with demand oracles. In *Proceedings of the 13th ACM conference on electronic commerce*, pages 110–127, 2012.
- [6] Ashwinkumar Badanidiyuru, Robert Kleinberg, and Yaron Singer. Learning on a budget: posted price mechanisms for online procurement. In *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012*, pages 128–145, 2012.
- [7] Xiaohui Bei, Ning Chen, Nick Gravin, and Pinyan Lu. Budget feasible mechanism design: from prior-free to bayesian. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012*, pages 449–458, 2012.
- [8] William F Caselton and James V Zidek. Optimal monitoring network designs. *Statistics & Probability Letters*, 2(4):223–227, 1984.
- [9] Venkatesan T Chakaravarthy, Amit Kumar, Sambuddha Roy, and Yogish Sabharwal. Resource allocation for covering time varying demands. In *European Symposium on Algorithms*, pages 543–554. Springer, 2011.
- [10] Ning Chen, Nick Gravin, and Pinyan Lu. On the approximability of budget feasible mechanisms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 685–699, 2011.
- [11] Peter Cramton, Yoav Shoham, and Richard Steinberg. *Combinatorial Auctions*. MIT Press, 2007.
- [12] Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.
- [13] Pankaj Dayama, Balakrishnan Narayanaswamy, Dinesh Garg, and Y. Narahari. Truthful interval cover mechanisms for crowdsourcing applications. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1091–1099, 2015.
- [14] Shahar Dobzinski, Christos H. Papadimitriou, and Yaron Singer. Mechanisms for complement-free procurement. In *Proceedings 12th ACM Conference on Electronic Commerce (EC-2011)*, pages 273–282, 2011.

- [15] Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M. Ni, and Athanasios V. Vasilakos. TRAC: truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2014*, pages 1231–1239, 2014.
- [16] Zhenni Feng, Yanmin Zhu, Qian Zhang, Hongzi Zhu, Jiadi Yu, Jian Cao, and Lionel Ni. Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones. In *Proceedings of the IEEE 34th International Conference on Distributed Computing Systems, ICDCS 2014*, pages 11–20, 2014.
- [17] S. Fujishige. Canonical decompositions of symmetric submodular systems. *Discrete Applied Mathematics*, 5:175–190, 1983.
- [18] Gagan Goel, Afshin Nikzad, and Adish Singla. Mechanism design for crowdsourcing markets with heterogeneous tasks. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014*, pages 77–86, 2014.
- [19] Nick Gravin, Yaonan Jin, Pinyan Lu, and Chenhao Zhang. Optimal budget-feasible mechanisms for additive valuations. In *Proceedings of the 20th ACM Conference on Economics and Computation, EC 2019*, pages 887–900, 2019.
- [20] Srinivasa Raghavendra Bhuvan Gummidi, Xike Xie, and Torben Bach Pedersen. A survey of spatial crowdsourcing. *ACM Transactions on Database Systems*, 44(2):8:1–8:46, 2019.
- [21] Shibo He, Dong-Hoon Shin, Junshan Zhang, and Jiming Chen. Toward optimal allocation of location dependent tasks in crowdsensing. In *Proceedings of the IEEE Conference on Computer Communications, INFOCOM 2014*, pages 745–753, 2014.
- [22] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2012*, 2012.
- [23] Pooya Jalaly Khalilabadi and Éva Tardos. Simple and efficient budget feasible mechanisms for monotone submodular valuations. In *14th International Conference on Web and Internet Economics, WINE 2018*, pages 246–263, 2018.
- [24] Andreas Krause, Ajit Paul Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.
- [25] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.
- [26] Stefano Leonardi, Gianpiero Monaco, Piotr Sankowski, and Qiang Zhang. Budget feasible mechanisms on matroids. In *International Conference on Integer Programming and Combinatorial Optimization, IPCO 2017*, pages 368–379, 2017.
- [27] T. Luo, S. S. Kanhere, and H. Tan. Sew-ing a simple endorsement web to incentivize trustworthy participatory sensing. In *2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 636–644, 2014.
- [28] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM*, 54(5):22:1–22:19, 2007.
- [29] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1), 1981.
- [30] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978.
- [31] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic behavior*, 35(1-2):166–196, 2001.
- [32] Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

- [33] M. Queyranne. Minimizing symmetric submodular functions. *Mathematical Programming*, 82(1-2):3–12, 1998.
- [34] Hamed Shah-Mansouri and Vincent W. S. Wong. Profit maximization in mobile crowdsourcing: A truthful auction mechanism. In *Proceedings of the IEEE International Conference on Communications, ICC 2015*, pages 3216–3221, 2015.
- [35] Yaron Singer. Budget feasible mechanisms. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010*,, pages 765–774, 2010.
- [36] Yaron Singer and Manas Mittal. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd international conference on World Wide Web, WWW 2013*, pages 1157–1166, 2013.
- [37] Adish Singla and Andreas Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd International World Wide Web Conference, WWW 2013*, pages 1167–1178, 2013.
- [38] Jiajun Sun and Huadong Ma. A behavior-based incentive mechanism for crowd sensing with budget constraints. In *2014 IEEE International Conference on Communications (ICC)*, pages 1314–1319. IEEE, 2014.
- [39] Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and Cyrus Shahabi. Spatial crowdsourcing: a survey. *The VLDB Journal*, 29(1):217–250, 2020.
- [40] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R. Jennings. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Journal of Artificial Intelligence Research JAIR*, 214:89–111, 2014.
- [41] Long Tran-Thanh, Matteo Venzani, Alex Rogers, and Nicholas R Jennings. Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks. 2013.
- [42] H. Xie, J. C. S. Lui, J. W. Jiang, and W. Chen. Incentive mechanism and protocol design for crowdsourcing systems. In *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 140–147, 2014.
- [43] Hong Xie, John C. S. Lui, and Don Towsley. Design and analysis of incentive and reputation mechanisms for online crowdsourcing systems. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 1(3), May 2016.
- [44] Jia Xu, Jinxin Xiang, and Dejun Yang. Incentive mechanisms for time window dependent tasks in mobile crowdsensing. *IEEE Transactions on Wireless Communications*, 14(11):6353–6364, 2015.
- [45] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones. *IEEE/ACM Transactions on Networking*, 24(3):1732–1743, 2016.
- [46] Han Yu, Chunyan Miao, Zhiqi Shen, Cyril Leung, Yiqiang Chen, and Qiang Yang. Efficient task sub-delegation for crowdsourcing. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI’15*, page 1305–1311. AAAI Press, 2015.
- [47] Xiang Zhang, Guoliang Xue, Ruozhou Yu, Dejun Yang, and Jian Tang. Truthful incentive mechanisms for crowdsourcing. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2830–2838. IEEE, 2015.
- [48] Y. Zhang and M. van der Schaar. Reputation-based incentive protocols in crowdsourcing applications. In *2012 Proceedings IEEE INFOCOM*, pages 2140–2148, 2012.
- [49] Dong Zhao, Xiang-Yang Li, and Huadong Ma. Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully. *IEEE/ACM Transactions on Networking*, 24(2):647–661, 2016.