Artificial Intelligence 254 (2018) 44–63

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

A quality assuring, cost optimal multi-armed bandit mechanism for expertsourcing

Shweta Jain [a,*], Sujit Gujar [b,1], Satyanath Bhat [c,2], Onno Zoeter [d,3], Y. Narahari [a]

[a] Indian Institute of Science, Bangalore, India
[b] International Institute of Information Technology, Hyderabad, India
[c] Institute of Operations Research and Analytics, NUS Business School, Singapore, Singapore
[d] Booking.com, Amsterdam, Netherlands

ARTICLE INFO

Article history:
Received 20 March 2017
Received in revised form 6 October 2017
Accepted 29 October 2017

Keywords:
Crowdsourcing
Multi-armed bandit
Mechanism design

ABSTRACT

There are numerous situations when a service requester wishes to *expertsource* a series of identical but non-trivial tasks from a pool of experts so as to achieve an assured accuracy level for each task, in a cost optimal way. The experts available are typically heterogeneous with unknown but fixed qualities and different service costs. The service costs are usually private to the experts and the experts could be strategic about their costs. The problem is to select for each task an optimal subset of experts so that the outcome obtained after aggregating the opinions from the selected experts guarantees a target level of accuracy. The problem is a challenging one even in a non-strategic setting since the accuracy of an aggregated outcome depends on unknown qualities. We develop a novel multi-armed bandit (MAB) mechanism for solving this problem. First, we propose a framework, *Assured Accuracy Bandit (AAB)* framework, which leads to a MAB algorithm, *Constrained Confidence Bound for Non-Strategic Setting (CCB-NS)*. We derive an upper bound on the number of time steps this algorithm chooses a sub-optimal set, which depends on the target accuracy and true qualities. A more challenging situation arises when the requester not only has to learn the qualities of the experts but has to elicit their true service costs as well. We modify the CCB-NS algorithm to obtain an adaptive exploration separated algorithm *Constrained Confidence Bound for Strategic Setting (CCB-S)*. The CCB-S algorithm produces an ex-post monotone allocation rule that can then be transformed into an ex-post incentive compatible and ex-post individually rational mechanism. This mechanism learns the qualities of the experts and guarantees a given target accuracy level in a cost optimal way. We also provide a lower bound on the number of times any algorithm must select a sub-optimal set and we see that the lower bound matches our upper bound up to a constant factor. We provide insights on a practical implementation of this framework through an illustrative example and demonstrate the efficacy of our algorithms through simulations.

* Corresponding author.
E-mail address: shwetajains20@gmail.com (S. Jain).

[1] This work was done when the author was post doctoral researcher at EPFL, Switzerland.
[2] This work was done when the author was Ph.D. student at Indian Institute of Science, Bangalore.
[3] This work was done when the author was working with Xerox Research Center, Europe.

https://doi.org/10.1016/j.artint.2017.10.001
0004-3702/

1. Introduction

Nowadays, crowdsourcing has emerged as an attractive tool for various organizations to obtain the services or ideas from the Internet. These organizations post their jobs on an online platform where they can hire anonymous workers online to complete the tasks and compensate the workers suitably. Though these types of crowdsourcing practices seem to be cost effective, it is often noticed that in many situations like building of wikipedia pages, hiring of anonymous workers results in extraordinary costs. In such cases, "anti-credentialism and anonymity result in uncertainty, irresponsibility, the development of cliques and the growing importance of pseudo-legal competencies for conflict resolution" [1]. Thus, it is prudent to employ professional experts, especially for specialist tasks, rather than employ anonymous workers online.

This work considers the problem of expertsourcing where experts are required to perform a task like developing customized software applications, technology forecasting, financial consultancy, medical image labeling and diagnosis, labeling legal documents and getting legal advice, patent search services, etc. In such expertsourcing problems, the requester (one who posts the tasks) faces many challenges. Many of these expertsourced projects are needed to be performed with high accuracy. Thus, unlike the crowdsourcing of micro-tasks, these tasks require highly skilled people in the specific domain of expertise. These experts further demand different prices for their services. These prices may depend on many parameters like their skill level, reputation, experience, or perhaps mere individual expectations.

There are many practical examples where expertsourcing find its use. The problem of Technology Forecasting (TF) is "concerned with the investigation of new trends, radically new technologies, and new forces which could arise from the interplay of factors such as new public concerns, national policies and scientific discoveries" [2]. TF is used to take many decisions such as product development, competition, or technology investments. These forecasts are particularly helpful for end users in predicting product development, anticipate competitors' technical capabilities, and to avoid technology surprises. In such technology forecasts, accuracy plays an important role and it becomes important for the end users to know how accurate the forecasts are. Fye et al. [3], compare the expert sourcing solution with other quantitative methods for technology forecast where experts were hired to provide their opinion on an event in technology forecast. Among these opinions, an opinion with majority voting is considered. The authors found that the experts are best at predicting if an event will occur or not while other quantitative methods are best at predicting when an event will occur.

Another example where expertsourcing finds its use is for a company which provides financial advice to its clients on whether or not to invest in a particular security. In order to provide such advice to each client, the company has a pool of financial consultants or experts. The company has two conflicting business requirements, first to keep the costs low, and second, to provide an advice that meets a minimum threshold accuracy level.

Expertsourcing is also emerging as a useful tool in the area of healthcare services like medical diagnosis [4] or public health surveillance [5]. Medical diagnosis is an area that requires a high level of accuracy. An experiment conducted by Mavandadi et al. [4] revealed that even highly trained medical experts are not always self-consistent in their diagnostic decisions and various experts may disagree among themselves even for binary decisions (like whether a cell is infected or not). Thus, in order to improve the accuracy of diagnosis, it is often required to consult multiple medical experts and combine their decisions together using an efficient algorithm like the one proposed in [4].

Motivated by the above examples, we consider an expertsourcing problem where a requester has a sequence of homogeneous tasks. For example, identifying whether or not human red blood cells are infected by malaria. By homogeneous tasks, we mean that the quality of any given individual expert does not change from one task to another. There is a pool of experts; each expert has different quality that is fixed but unknown. These qualities denote the skill sets of an expert, for example, in medical diagnosis, the quality depicts the probability with which an expert will provide the correct diagnosis. Each expert incurs a fixed cost to perform a task that is his private information. The quality and cost of each expert depend on the expertise and other exogenous parameters. Mavandadi et al. [4] demonstrated that combined opinion of a group of experts can significantly boost the accuracy of the final diagnostic decision as compared to the best individual of the group. Motivated by this, we aim to select a subset of the strategic experts incurring minimum total cost to achieve a desired accuracy for each task by aggregating the opinions from the selected experts, at the same time giving the correct incentives to the experts so that they report their costs truthfully. The target accuracy level parameter provides a handle on the trade-off between cost and accuracy. A high value of the target accuracy enhances the probability of getting an accurate opinion but at the same time could call for a larger number of experts, leading to increased costs. Therefore, one can choose a suitable target accuracy level depending on the task sensitivity.

In the absence of strategic play (with known costs), the setting reduces to a machine learning problem. Though the requester can learn the qualities of the experts over a period of time by observing their performance on similar tasks, selecting a set of low quality experts repeatedly may incur significant costs. Thus, the requester faces a dilemma of *exploration* (learning the qualities of the experts) versus *exploitation* (choosing the experts optimally based on the learnt qualities). A natural solution to this problem can be explored using techniques developed for the multi-armed bandit (MAB) problems [6]. Many existing works have considered learning qualities in non-strategic crowdsourcing settings via MAB framework [7, 8]. In [9], authors have solved the problem of non-strategic expertsourcing with limited budget using techniques from MAB algorithms. However, an important new challenge in our setting is the need to ensure the accuracy constraint which in turn depends on unknown qualities. Thus, there is a need to develop a new framework to address the accuracy constraint.

An additional challenge arises when the costs of the experts are private and they could manipulate the learning algorithm by misreporting their costs so as to benefit themselves. Thus, we have the additional task of eliciting the true costs using

a suitable mechanism. Since the qualities are unknown, classical mechanism such as VCG (Vickery–Clarke–Groves) cannot be applied directly [10]. In short, we need to blend techniques from machine learning and game theory (in particular, mechanism design) that would ensure honest behavior of the experts while the requester learns the qualities. Often such mechanisms are referred to as *Multi-Armed Bandit Mechanisms or simply MAB mechanisms* [10–13]. Previous works on MAB mechanisms, however, are not designed to achieve a target accuracy level. The MAB mechanism proposed in this paper also achieves required target accuracy level.

1.1. Contributions

The above discussion brings forth the need to design a new approach to solve the problem of selecting a subset of strategic experts to achieve a target accuracy level in a cost optimal way. We model this problem in the MAB mechanism framework by considering two versions: (1) a *non-strategic version*: the costs are known and the qualities have to be learnt and (2) a *strategic version*: the costs have to be truthfully elicited as well. Our specific contributions are:

- We propose a novel framework, *Assured Accuracy Bandit* (AAB) where we formulate an optimization problem to select a cost optimal subset of experts subject to the assured accuracy constraint for each task (Section 3).
- We provide a lower bound on the regret that any MAB algorithm in the AAB framework will suffer (Theorem 3.1).

Non-strategic version

- We design a novel algorithm, the Non-Strategic Constraint Confidence Bound (CCB-NS) for the AAB framework (Algorithm 1). Though the true qualities are not known, our algorithm guarantees that the accuracy constraint is satisfied with high probability (Theorem 4.1).
- We provide an upper bound on the number of times the algorithm selects a suboptimal expert set for a given problem, which depends on the target accuracy level and the true qualities (Theorem 4.2). This upper bound matches the lower bound up to a constant factor.

Strategic version

- In strategic version, the experts may not report their costs truthfully; we modify the CCB-NS algorithm to an adaptive exploration separated algorithm, which we call the Strategic Constrained Confidence Bound (CCB-S) and prove that the allocation rule provided by the CCB-S is ex-post monotone (Theorem 5.2) in terms of the cost. The ex-post monotonicity facilitates existing techniques [11] to be adopted to obtain an ex-post truthful and ex-post individually rational mechanism (Corollary 5.3).
- We extend the CCB-S algorithm as a non-exploration separated algorithm by exploiting the specific structure of a particular optimization problem. We also show the efficacy of our algorithms and compare the performance of the algorithm with that of a variant of ε_t-greedy algorithm [14], through simulations (Section 6). To the best of our knowledge, the proposed mechanism is the first of its kind that has all the following features: (a) learns the qualities of strategic experts (b) elicits cost information from the experts truthfully, and (c) guarantees a specified target accuracy level for each task in a cost optimal way.

2. Related work

Learning in crowdsourcing

A similar setting with an assured quality is considered in [15]. However, they dealt with a specific error probability function with a uniform and known cost of the experts. With multiple homogeneous quality experts, the problem of selecting an optimal cluster to satisfy accuracy constraint for a micro-task is considered in [16]. In a general setting, assumption of a cluster having sufficient number of homogeneous quality experts may not hold. Our setting is more general where an optimal subset of experts, with heterogeneous qualities, needs to be selected at one go for a given micro-task. A heterogeneous setting with experts having different costs and qualities is considered in [9], where authors designed an MAB algorithm for efficient selection of capacitated experts in the expertsourcing setting. For each task, a single non-strategic expert is selected as opposed to the subset selection of strategic experts. A model with different quality experts for each variety of task is considered in [17]. Improving the quality of answers while minimizing the cost is considered in [18]. Work in [19] considers learning a classifier while learning the qualities of the experts [20]. A Bayesian approach to learn the class label with noisy observations from experts is considered in [21]. Though, the models proposed [21,20,19] work well experimentally, there are no analytic guarantees on the predicted outcome. None of the above papers addresses the challenge in meeting the target accuracy level on each task in a heterogeneous, strategic cost model.

Mechanism design in crowdsourcing

A majority of the literature on mechanism design in crowdsourcing involves design of pricing strategies with online experts. An MAB mechanism to determine an optimal pricing mechanism for a crowdsourcing problem having homogeneous

qualities within a specified budget is considered in [22]. When costs are private information, [23] proposes a posted price mechanism to elicit the true costs from the users using MAB mechanisms while maintaining a budget constraint. Mechanism design in online procurement auctions [22–25] considers homogeneous quality experts. In our setting, an auction mechanism is considered to elicit the true costs from the experts with heterogeneous qualities.

Private and strategic qualities with public costs model is considered in [26,27]. Another line of work involves incentivizing people to work with their true qualities, when the qualities are privately held by the experts [28] in peer prediction markets. Cavallo and Jain [29] analyze crowdsourcing tasks as winner take it all auctions in game theoretic settings. They assume that only one expert gets paid and do not try to learn the qualities over period. [30,31] adopt techniques from online mechanism design for eliciting the expert preferences but do not address the task accuracy problem.

Mechanism design theory in crowdsourcing either elicit the costs of the experts where qualities are homogeneous and known or elicit the qualities of the experts assuming the costs to be known. Our work addresses the setting where the qualities of heterogeneous experts are to be learnt and the heterogeneous costs are to be elicited.

MAB algorithms

Our problem belongs to the stochastic MAB setting, where the reward of each arm is fixed but unknown. A recent survey by Bubeck and Cesa-Bianchi [32] compiles several variations on stochastic and non-stochastic MAB problems. The setting that is closest to ours is considered in [33] where a general bandit problem with concave rewards and convex constraints is solved. A specific case of this problem with linear rewards and global constraints is considered in [34]. Our problem setting is a further generalization, as the constraints in AAB are not convex. Moreover, the constraints need to be satisfied cumulatively across multiple rounds in [33] as opposed to our work, where the constraint needs to be satisfied at each round. Our learning algorithm may appear closely related to the PAC learning setting [35–37] but it differs in a subtle but important way. The solution obtained from any PAC algorithm is approximately correct with high probability after arms are pulled for a certain number of rounds, which depends on the provided approximation factor and the confidence. In our setting, the goal is to select an optimal set with high probability since a constraint needs to be satisfied with respect to stochastic qualities. Moreover, the number of exploration steps are adaptive that depends on the true qualities and the target accuracy level as opposed to the fixed number of exploration rounds in the PAC setting. The combinatorial MAB problem is further considered in many works [38–42] where the objective is to identify an optimal subset from given feasible subsets. However, in our setting collection of feasible sets is not given and has to be learnt over time and this makes our work different in the non-strategic setting. Problem with constraints in each round is considered in [43] but the problem of choosing a distribution over arms is considered instead of subset selection.

Stochastic MAB mechanisms

Multi-armed Bandit mechanisms in the forward setting as applied to sponsored search auction are recent advancements. Any deterministic truthful MAB mechanism must be exploration separated and thus the regret of any such algorithm is at least $O(T^{2/3})$ with T being the number of rounds [10,12]. The results are also extended to multiple pull multi-armed bandits [44,13]. However, these works do not consider combinatorial, constrained multi-armed bandit setting. A general transformation which outputs a randomized truthful mechanism with any monotone allocation rule is designed in [11]. As an application of this transformation, an MAB mechanism that is ex-post incentive compatible and ex-post individual rational with regret of $O(T^{1/2})$ is proposed. We use this transformation and propose an ex-post monotone allocation rule in the case of a reverse auction in a constrained multi-armed bandit setting. The mechanism in [45] can be translated in this setting. However, the authors considered single worker selection and do not cater to the final accuracy of the task. When the strategic agents are asked to chose the arms instead of a learning algorithm, then the problem of incentivizing the agents to explore the arms inorder to maximize the expected reward is considered in [46,47].

Our preliminary results in [48] considered only a certain type of error probability function. This current paper represents a significant improvement over our previous paper and the techniques developed in this paper are applicable to a general class of error probability functions.

3. The model

Let $\mathcal{N}$ be a set of n experts available for working on T homogeneous expertsourcing tasks. Each expert i has an associated quality q_i, which is the probability that the opinion given by him is correct. The quality of any expert i is assumed to be independent of the qualities of other experts. An expert i incurs a cost $c_i \in \mathbb{R}$ which is privately held and can be reported strategically. Let $1-\alpha$ be the target accuracy level (α is the threshold level) provided by the requester. This parameter determines the trade-off between the cost and the accuracy to be achieved for a particular task. The error on a task with inputs from the experts depends on the qualities of the experts and the rule to aggregate these opinions. We abstract this as error probability function which we describe in the following subsection.

3.1. Error probability function

Let $f_S(q) : [0,1]^n \to [0,1]$ be any error probability function (hence $(1 - f_S(q))$ is the accuracy) when a set S is selected with quality profile $q = (q_1, q_2, \ldots, q_n)$. Our goal is to minimize the cost and at the same time satisfy the constraint that

$f_S(q) < \alpha$ where $(1-\alpha)$ is the target accuracy. Depending on the aggregation rule and the requester requirements, different error probability functions could be defined. Our framework and the solution approach are general and work for any error probability function that satisfies the following properties:

- *Monotonicity:* $f_S(q)$ is monotone if for all quality profiles q and q' such that if $\forall i \in \mathcal{N},\ q'_i \leq q_i,$ we have,

  $$f_S(q') < \alpha \implies f_S(q) < \alpha,\ \forall S \subseteq \mathcal{N},\ \forall \alpha \in [0,1]\,.$$

  That is, an increase in quality of each expert can only increase the accuracy or decrease the error probability.
- *Bounded smoothness:* $f_S(q)$ satisfies bounded smoothness if there exists a strictly increasing, continuous (hence, invertible) function h such that if

  $$\max_i |q_i - q'_i| \leq \delta \implies |f_S(q) - f_S(q')| \leq h(\delta),\ \forall S \subseteq \mathcal{N}, \forall q, q' \in [0,1]\,.$$

  That is, the difference in error probability function with respect to close quality profiles is bounded by a monotone continuous function h.

These properties are similar to the ones in [38] for reward function and are satisfied by various error probability functions.

Example 3.1. With majority voting rule as the aggregation rule, the average probability of error is [49]:

$$\mathbb{P}(E_{S(q)}) \leq f_S(q) = \exp\left(\frac{-\left(\sum_{i=1}^{s}(2q_i-1)\right)^2}{2\sum_{i=1}^{s}1}\right)$$

Here, S is the selected set with players $\{1,2,\ldots,s\}$ with the quality profile q. For binary labeling tasks, let $\tilde{y}_i \in \{-1,1\}$ are the reported labels that we get from the expert $i \in \{1,2,\ldots,s\}$ and $\tilde{y}(S) = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_s)$ are the vector of reported labels from the experts set S. Then, the predicted label $\hat{y}$ with majority voting rule is:

$$\hat{y} = \begin{cases} 1 \text{ if} \sum_{i \in S} \tilde{y}_i > 0, \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

One can verify that $f_S(q)$ is monotone and satisfies bounded smoothness property. If the qualities satisfy $\frac{1}{2} + \epsilon \leq q_i \leq 1\ \forall i$, then:

$$\mathbb{P}(E_{S(q)}) \leq \exp\left(-\sum_{i=1}^{s}(2q_i-1)\epsilon\right) = f_S(q).$$

One can also define similar error probability function with other types of aggregation rules such as weighted majority voting rule which also satisfies monotonicity and bounded smoothness. In the example of software development, one can define the error probability function as the probability with which at least one code will be correct from the hired experts. It is easy to see that even this error probability function satisfies monotonicity and bounded smoothness properties. We now describe the framework with optimization problem.

3.2. Assured Accuracy Bandit (AAB) framework

Recall that a task $t \in \{1,\ldots,T\}$ needs to be completed with an assured accuracy with the optimal cost in a sequential fashion. Hence for each task t, the following optimization problem needs to be solved.

$$\min_{X_i^t \in \{0,1\}} \sum_i c_i X_i^t,\ \text{s.t.},\ f_{\{i:X_i^t=1\}}(q) < \alpha. \tag{2}$$

Here, the qualities of the experts are not known a priori and hence need to be learnt by giving tasks repeatedly to the experts. Also, solving the optimization problem, the requester has to make sure that the constraint in (2) is satisfied with respect to the true qualities with high confidence. We refer to this novel framework as Assured Accuracy Bandits (AAB).

Note that in the above optimization problem we use error probability function $f_S(q)$. In Example 3.1 function $f_S(q)$ gives an upper bound on the error but not the real aggregation error. Since, in order to solve the optimization problem we need a functional form on the error, we have formulated the optimization problem with respect to this upper bound and we refer to the solution of this optimization problem with respect to true qualities as the optimal solution in our paper.

3.2.1. Regret in AAB framework

Regret in an MAB framework is defined to be the reward difference between the learning algorithm and the optimal algorithm. We will see later that our algorithm satisfies the constraint given by (2) for each task t with probability $(1-\mu)$, where μ is the confidence parameter with which constraint is satisfied. Let S^* and S^t denote the optimal set and set selected at time t respectively. Then the regret of an algorithm $\mathcal{A}$ if the constraint is satisfied is given as:

$$\mathcal{R}(\mathcal{A}) = \sum_{t=1}^{T} \sum_{i \in S^t} c_i - T \sum_{i \in S^*} c_i. \tag{3}$$

Since the constraint is satisfied with probability $(1-\mu)$:

$$\mathbb{E}[\mathcal{R}(\mathcal{A})] = (1-\mu)\left(\sum_{t=1}^{T} \sum_{i \in S^t} c_i - T \sum_{i \in S^*} c_i\right) + \mu L T\,, \tag{4}$$

where L is the cost that is incurred by the requester if the constraint fails to satisfy. We consider a setting with large value of L, and the requester would not want to violate the constraint. However, due to stochasticity involved in learning the qualities, there is a small probability (μ) with which the constraint can be violated. With $\mu = 1/T$, we get:

$$\mathbb{E}[\mathcal{R}(\mathcal{A})] = (1-\frac{1}{T})\left(\sum_{t=1}^{T} \sum_{i \in S^t} c_i - T \sum_{i \in S^*} c_i\right) + L\,.$$

3.3. Lower bound on the regret

We first start with an important property called as Δ-separated property that we assume any quality profile q satisfies. The property is given as follows:

Definition 3.1 *(Δ-Separated property).* We say that q is Δ-*Separated* with respect to the threshold α if $\exists \Delta > 0$ such that, $\Delta = \inf_{S \subseteq \mathcal{N}} |f_S(q) - \alpha|$. That is, no set of experts, S, has probability of error $f_S(q) \in (\alpha - \Delta, \alpha + \Delta)$.

Given a quality profile q that satisfies Δ-separated property, we now provide a lower bound on the regret that any algorithm in AAB framework has to suffer.

Theorem 3.1. *Let $n_S(\mathcal{A})$ denotes the number of times an expert set S is selected till time T by an algorithm $\mathcal{A}$. Consider any algorithm that solves the optimization problem in Equation (2) and satisfies $E[n_S(\mathcal{A})] = o(T^a)\ \forall a > 0$ for all subset of experts S which is not optimal. Then:*

$$\liminf_{T \to \infty} \frac{\mathbb{E}[\mathcal{R}(\mathcal{A})]}{\ln T} \geq \frac{1}{(h^{-1}(\Delta))^2},$$

where, $\Delta = \inf_{S \subseteq \mathcal{N}} |f_S(q) - \alpha|$ and $h(.)$ is the bounded smooth function.

Proof. The proof follows similar steps to the proof of the lower bound for classical MAB problem [32,6]. For $p_1, p_2 \in [0,1]$, denote $kl(p_1, p_2)$ the Kullback–Leibler divergence between a Bernoulli of parameter p_1 and a Bernoulli of parameter p_2 defined as:

$$kl(p_1, p_2) = p_1 \ln\left(\frac{p_1}{p_2}\right) + (1-p_1)\ln\left(\frac{1-p_1}{1-p_2}\right).$$

It is easy to see that the function $x \mapsto kl(p_1, x)$ is a continuous function.

- Consider two experts with quality profile, $q = (q_1, q_2)$ with $f_{\{1\}}(q) < \alpha < f_{\{2\}}(q)$ and $c_1 > c_2$. Since, kl divergence is a continuous function and error probability function f is monotone, for any $\epsilon > 0$, one can find quality profile $q' = (q_1, q_2')$ such that $f_{\{1\}}(q') < f_{\{2\}}(q') < \alpha$ and $kl(q_2, q_2') \leq (1+\epsilon) kl(q_2, 1-\alpha)$. Thus, expert 1 is optimal with quality profile q but expert 2 is optimal with quality profile q'. Denote $\mathbb{P}, \mathbb{E}$ as the probability, expectation taken with respect to random variables generated with quality profile q and $\mathbb{P}', \mathbb{E}'$ as the probability, expectation taken with respect to random variables generated with quality profile q'.
- Denote $X_{2,1}, X_{2,2}, \ldots, X_{2,T}$ as the sequence of successes obtained when allocating tasks to expert 2 where successes are coming from quality profile q. For any $t \in \{1, 2, \ldots, T\}$, let

$$\hat{kl}_t = \sum_{i=1}^{t} \ln \frac{q_2 X_{2,i} + (1-q_2)(1-X_{2,i})}{q_2' X_{2,i} + (1-q_2')(1-X_{2,i})}.$$

Also, $\mathbb{E}[\hat{kl}_{n_2(\mathcal{A})}] = n_2(\mathcal{A})kl(q_2, q_2')$. We also have the following identity for any event B in the σ-algebra generated by $X_{2,1}, \ldots, X_{2,T}$:

$$\mathbb{P}'(B) = \mathbb{E}[\mathbb{1}_B \exp(-\hat{kl}_{n_2(\mathcal{A})})]. \tag{5}$$

- Now, consider the event $C_T = \{n_2(\mathcal{A}) < \frac{1-\epsilon}{kl(q_2,q_2')} \ln(T)$ and $\hat{kl}_{n_2(\mathcal{A})} \leq (1 - \frac{\epsilon}{2}) \ln(T)\}$. We will prove that $\mathbb{P}(C_T) \to 0$ as $T \to \infty$. From Equation (5):

$$\mathbb{P}'(C_T) = \mathbb{E}[\mathbb{1}_{C_T} exp(-\hat{kl}_{n_2(\mathcal{A})})] \geq e^{-(1-\epsilon/2)\ln(T)} \mathbb{P}(C_T).$$

Let $f_T = \frac{1-\epsilon}{kl(q_2,q_2')} \ln(T)$. Then using Markov's inequality we have,

$$\begin{aligned}\mathbb{P}(C_T) &\leq T^{(1-\epsilon/2)} \mathbb{P}'(C_T) \\ &\leq T^{(1-\epsilon/2)} \mathbb{P}'(n_2(\mathcal{A}) < f_T) \\ &\leq T^{(1-\epsilon/2)} \frac{\mathbb{E}'(T - n_2(\mathcal{A}))}{T - f_T}.\end{aligned}$$

Since there are only two experts here, $T - n_2(\mathcal{A}) = n_1(\mathcal{A})$. With respect to quality profile q' since expert 1 is sub-optimal, we have:

$$\mathbb{P}(C_T) \leq T^{(1-\epsilon/2)} \frac{T^a}{T - f_T}, \ \forall a > 0.$$

Consider $a < \epsilon/2$ then we get $\mathbb{P}(C_T) \to 0$ as $T \to \infty$.

- Now, we will prove that $\mathbb{P}(n_2(\mathcal{A}) < f_T) \to 0$ as $T \to \infty$. We have,

$$\begin{aligned}&\mathbb{P}(C_T) \\ &\geq \mathbb{P}\left(n_2(\mathcal{A}) < f_T \text{ and } \max_{t \leq f_T} \hat{kl}_t \leq \left(1 - \frac{\epsilon}{2}\right) \ln(T)\right) \\ &= \mathbb{P}\left(n_2(\mathcal{A}) < f_T \text{ and } \frac{kl(q_2, q_2')}{(1-\epsilon)\ln(T)} \max_{t \leq f_T} \hat{kl}_t \leq \frac{1 - \epsilon/2}{1-\epsilon} kl(q_2, q_2')\right).\end{aligned}$$

Using the maximal version of the strong law of large numbers and since $kl(q_2, q_2') > 0$ and $\frac{1-\epsilon/2}{1-\epsilon} > 1$, we get:

$$\lim_{T \to \infty} \mathbb{P}\left(\frac{kl(q_2, q_2')}{(1-\epsilon)\ln(T)} \max_{t \leq f_T} \hat{kl}_t \leq \frac{1 - \epsilon/2}{(1-\epsilon)} kl(q_2, q_2')\right) = 1.$$

Thus, from the previous point we get $\mathbb{P}(n_2(\mathcal{A}) < f_T) \to 0$ as $T \to \infty$. Thus, we get,

$$\mathbb{E}[n_2(\mathcal{A})] > \frac{1-\epsilon}{1+\epsilon} \frac{\ln(T)}{kl(q_2, 1-\alpha)}.$$

Using the fact that $kl(p_1, p_2) \leq \frac{(p_1-p_2)^2}{p_2(1-p_2)}$ and expert 2 is suboptimal with quality profile q we get:

$$\begin{aligned}\mathbb{E}[\mathcal{R}(\mathcal{A})] &> \left(\frac{1-\epsilon}{1+\epsilon}\right) \frac{\ln(T)}{kl(q_2, 1-\alpha)} (c_2 - c_1) \\ &\geq \left(\frac{1-\epsilon}{1+\epsilon}\right) \frac{\alpha(1-\alpha)\ln(T)}{(q_2 - (1-\alpha))^2} (c_2 - c_1).\end{aligned}$$

- Since, $(1-\alpha)$ is the target accuracy, $f_S(1-\alpha) = \alpha$, for any subset S. From the bounded smoothness property, $|f_{\{2\}}(q) - f_{\{2\}}(1-\alpha)| \leq h(q_2 - (1-\alpha)) \implies (q_2 - (1-\alpha))^2 \geq (h^{-1}(|f_{\{2\}}(q) - \alpha|))^2$. If we choose q_1 and q_2 such that $\Delta = |f_{\{2\}}(q) - \alpha|$ then we get $\mathbb{E}[\mathcal{R}](\mathcal{A})] > \left(\frac{1-\epsilon}{1+\epsilon}\right) \frac{\alpha(1-\alpha)\ln(T)}{(h^{-1}(\Delta))^2} (c_2 - c_1)$, yielding the lower bound. □

The above theorem proves that, in order to reach to the optimal solution in AAB framework, it is required to pull a sub-optimal arm at least $O\left(\frac{\ln(T)}{(h^{-1}(\Delta))^2}\right)$ number of times. Here, Δ depends on the problem instance.

4. Non-strategic version

In this setting, we solve the optimization problem in Equation (2) with unknown qualities and known costs. Since the experts give their opinions according to the true and unknown qualities, the constraint has to be satisfied with respect to the true qualities with high probability. Note that our algorithm works in a general setting and uses the aggregation rule as a black box.

Definition 4.1 *(Aggregate).* An aggregate function takes the noisy opinions of the selected set of experts as input and produces an opinion which best captures the opinion of the selected set. The aggregate function should ensure that the resulting error probability function satisfies the properties of monotonicity and bounded smoothness. For example, if the majority voting rule is used for binary labeling tasks, the aggregated label $\hat{y}$ is computed using the equation (1).

4.1. CCB-NS algorithm

The CCB-NS algorithm (presented in Algorithm 1) works on the principle of the UCB algorithm [14] and ensures that the constraint in (2) is satisfied with high confidence μ. Input to the algorithm is parameter α, the target accuracy (which is assumed to be same for all the tasks), the number of tasks T, the number of experts n, and confidence level μ with which the constraint in (2) is required to be satisfied. The output of the algorithm will be the subset S^t and predicted opinion $\hat{y}^t$ for each task t. The predicted opinion $\hat{y}^t$ is decided based on an aggregation function (AGGREGATE) in Definition 4.1 with noisy opinions $\tilde{y}(S^t)$ collected from the expert set S^t as input.

Initially all the experts are selected to have some estimate about the qualities (Step 2). Their reported opinions are aggregated and an opinion is predicted. We assume that if the complete set of experts is selected then the accuracy is always met. This assumption implies that we have enough number of good quality experts. Next, the algorithm updates the mean quality estimates, the upper and lower confidence bounds. In certain examples like intraday trading market or in a coding project, the correct opinion or the correctness of the code is revealed immediately. In such cases, the mean quality updates can be made based on the true opinion observed. However, when there is no known ground truth, then

Algorithm 1: CCB-NS algorithm.

Input: Set of experts $\mathcal{N}$, number of tasks T, parameter α, confidence level μ, cost vector $c = (c_1, \ldots, c_i, \ldots, c_n)$
Output: Expert selection set S^t, opinion $\hat{y}^t$ for all tasks $t \in \{1, 2, \ldots, T\}$

1 $\forall i \in \mathcal{N}$, $\hat{q}_i^+ = 1$, $\hat{q}_i^- = 0$ // Initialize UCB and LCB on qualities
2 $S^1 = \mathcal{N}$ // Select all experts initially
3 Observe $\tilde{y}(S^1)$ and $\hat{y}^1 = \text{AGGREGATE}(\tilde{y}(S^1))$ (Definition 4.1)
4 $\forall i \in \mathcal{N}$, $n_i(1) = 1$, update $\hat{q}_i$
5 $t = 2$
6 $S^t = \arg\min_{S \subseteq \mathcal{N}} \sum_{i \in S} c_i$ s.t. $f_S(\hat{q}^+) < \alpha$
7 **while** $f_{S^t}(\hat{q}^-) > \alpha$ **do**
8 // Explore (not the optimal set, add more experts to satisfy the constraint)
9 $S^t = S^t \cup \text{MINIMAL}(S^t, \mathcal{N} \setminus S^t, \hat{q}^-)$
10 Observe opinions of selected experts $\tilde{y}(S^t)$
11 $\hat{y}^t = \text{AGGREGATE}(\tilde{y}(S^t))$
12 **for** $i \in S^t$ **do**
13 $n_i(t) = n_i(t-1) + 1$
14 Update $\hat{q}_i$, $\hat{q}_i^+ = \hat{q}_i + \sqrt{\frac{1}{2n_i(t)} \ln(\frac{2n}{\mu})}$, $\hat{q}_i^- = \hat{q}_i - \sqrt{\frac{1}{2n_i(t)} \ln(\frac{2n}{\mu})}$
15 $t = t + 1$
16 $S^t = \arg\min_{S \subseteq \mathcal{N}} \sum_{i \in S} c_i$ s.t. $f_S(\hat{q}^+) < \alpha$
17 $t^* = t$
18 $S^{t^*} = S^t$
19 **for** $t = t^* + 1$ *to* T **do**
20 // Exploit (optimal set with high probability)
21 $S^t = S^{t^*}$
22 Observe opinions of selected experts $\tilde{y}(S^t)$
23 $\hat{y}^t = \text{AGGREGATE}(\tilde{y}(S^t))$

24 Subroutine: $\text{MINIMAL}(S^t, S, q)$
25 Return a cost minimal set $S' \subseteq S$ of experts such that $f_{S^t \cup S'}(q) < \alpha$
26 If no such set S' exists then return S

the aggregated opinion can be assumed as the true opinion since we are assuring a high accuracy level. Note that in cases like weighted majority voting rule where a bias is used for the aggregated opinion, the error probability function is also a function of weights [49]. If the weights, in turn, depend on the unknown qualities of the experts then for our algorithm to work, the error probability function should be monotone with respect to the weights. If the error probability function is monotone with respect to the weights (which in turn depends on the qualities), then for aggregation, lower confidence bound of qualities should be used as the weights. This would ensure that the predicted label satisfies the accuracy constraint with respect to true qualities as well with high probability (due to monotonicity).

Let $n_i(t)$ denote the number of times the ith expert is assigned the task and $\hat{q}_i(t)$ denote the estimate on quality parameter q_i. Similar to the UCB algorithm [14], the algorithm maintains upper confidence and lower confidence bounds on qualities. These bounds are given as:

$$\hat{q}_i^+(t) = \hat{q}_i(t) + \sqrt{\frac{1}{2n_i(t)} \ln\left(\frac{2n}{\mu}\right)},\ \hat{q}_i^-(t) = \hat{q}_i(t) - \sqrt{\frac{1}{2n_i(t)} \ln\left(\frac{2n}{\mu}\right)}.$$

By Hoeffding's inequality, one can prove that the true quality q_i lies between $\hat{q}_i^-(t)$ and $\hat{q}_i^+(t)$ with probability $1 - \frac{\mu}{n}$ for any task t and for any expert i. The bounds used by UCB1 algorithm given in [14] is given by: $\hat{q}_i^+(t) = \hat{q}_i(t) + \sqrt{\frac{2 \ln t}{n_i(t)}}$, $\hat{q}_i^-(t) = \hat{q}_i(t) - \sqrt{\frac{2 \ln t}{n_i(t)}}$. Since, we want to extend the algorithm to the strategic setting, we are using a constant term $\frac{2n}{\mu}$ in the bounds instead of t. We initialize $\hat{q}_i^+(t)$ and $\hat{q}_i^-(t)$ by 1 and 0 respectively as the true qualities of the experts lie between $[0, 1]$. In the algorithm, we represent $\hat{q}_i$, $\hat{q}_i^+$ and $\hat{q}_i^-$ to be the estimates till t number of tasks. Since, the true qualities lie between upper and lower confidence bound with high probability, one can think of an algorithm that solves the optimization problem given in Equation (2) with lower confidence bound on qualities. However, this strategy will not work as high quality experts may not be explored in this approach. The key idea is, till we have identified the optimal subset of experts, we solve the optimization problem using the upper confidence bound on the qualities which gives a cost effective subset. However, this subset need not meet the desired accuracy. Hence we add another subset of the experts from the remaining experts (using subroutine MINIMAL) that combined together ensures that the target accuracy is met even when we use the lower estimates, that is $\hat{q}^-$ in the constraints. The fact that $q_i \geq \hat{q}_i^-$ with probability at least $1 - \frac{\mu}{n}$ and the monotonicity of the error function ensures that the target accuracy level is achieved in each round with high probability. Once the algorithm finds a subset that is optimal with respect to the upper confidence bound and achieves the target accuracy even when using the lower confidence on qualities, the algorithm stops learning and uses this set for the remaining tasks. We prove (Lemma 4.1) that this is the required optimal set with high probability. Note that, in Step 9 if the MINIMAL function cannot find a set satisfying the target accuracy level using the lower confidence bound, then it simply returns $\mathcal{N}$ which meets the target accuracy level by our assumption.

We first see that the algorithm CCB-NS satisfies the constraint at each round with high probability. Note that by Hoeffding's inequality, for each i, $\hat{q}_i^- \leq q_i \leq \hat{q}_i^+$ with probability $1 - \frac{\mu}{n}$. Since the experts make error independently, we have, $\forall i \in \mathcal{N}\ \hat{q}_i^- \leq q_i \leq \hat{q}_i^+$ with probability $\left(1 - \frac{\mu}{n}\right)^n \geq (1 - \mu)$ by Bernoulli's inequality. Thus, $\forall i \in \mathcal{N}\ \hat{q}_i^- \leq q_i \leq \hat{q}_i^+$ with probability greater then $1 - \mu$. For brevity of notation, in the rest of the paper we will use $\hat{q}^- \leq q \leq \hat{q}^+$ to represent $\hat{q}_i^- \leq q_i \leq \hat{q}_i^+\ \forall i \in \mathcal{N}$. Thus from monotonicity of $f(.)$,

$$\text{w.p. at least } (1 - \mu),\ f_S(\hat{q}^+) < f_S(q) < f_S(\hat{q}^-)\ \forall S \subseteq \mathcal{N} \tag{6}$$

Theorem 4.1. *The CCB-NS algorithm satisfies the accuracy constraint with probability at least* $(1 - \mu)$ *at every round* t.

Proof. If all the experts are selected, the constraint is always satisfied (by assumption). Now, if set S^t is returned by CCB-NS, then, $f_{S^t}(\hat{q}^-) < \alpha \implies f_{S^t}(q) < \alpha$ with probability $1 - \mu$ (from Equation (6)). □

We now show that if the algorithm exits the while loop in Step 17 then the set $S^{t^*} = S^*$ (the optimal set) with probability at least $1 - \mu$. For simplicity, we assume that there exists a unique optimal set S^*, though the results can be easily generalized when there are multiple optimal sets.

Lemma 4.1. *Set* S^{t^*} *returned by the CCB-NS algorithm is an optimal set with probability (w.p.) at least* $1 - \mu$. *That is,* $C(S^{t^*}) = C(S^*)$ *w.p.* $1 - \mu$.

Proof. Since $f_{S^*}(q) < \alpha$, we have $f_{S^*}(\hat{q}^+) < \alpha$ w.p. $1 - \mu$ (from Equation (6)). As CCB-NS returns the solution S^{t^*} when the constraint is satisfied with lower confidence bound, $C(S^{t^*}) \leq C(S^*)$ and $f_{S^{t^*}}(\hat{q}^-) < \alpha$. Thus, $f_{S^{t^*}}(q) \leq f_{S^{t^*}}(\hat{q}^-) < \alpha$ with probability at least $1 - \mu$. Thus, $C(S^{t^*}) = C(S^*)$. □

4.2. Regret analysis of CCB-NS

Definition 4.2 *(Non-optimal subset).* At round t, a set S^t selected by the algorithm is a *non-optimal subset*, if $S^t \neq S^*$.

Definition 4.3 *(Non-optimal round).* A round t is a *non-optimal round* if the selected set S^t is not the optimal set S^*.

Since the algorithm selects a set which satisfies the constraint for each task with high probability, we can bound the overall regret by bounding the number of rounds in which the algorithm selects a sub-optimal set S^t i.e. $C(S^t) > C(S^*)$. If $C(S^t) = C(S^*)$, then we get zero regret for those rounds with probability $(1-\mu)$. We will show that the number of non-optimal rounds depends on the value of Δ where $\Delta = \inf_{S \subseteq \mathcal{N}} |f_S(q) - \alpha|$. The value of Δ is typically unknown to the requester since qualities are unknown but our algorithm does not require the value of Δ beforehand and thus, CCB-NS is adaptive in nature.

Lemma 4.2. *If $\forall i \in \mathcal{N}$, number of times an expert i is selected till tasks t, $n_i(t) \geq \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$, then $\forall t$,*

1. $\forall S \subseteq \mathcal{N},\ S \neq S^*,\ f_S(q) > \alpha \implies f_S(\hat{q}^+) > \alpha$ *with probability* $1-\mu$.
2. $f_{S^*}(\hat{q}^-) < \alpha$ *with probability* $1-\mu$.

Proof. Let $l = \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$. By Hoeffding's inequality, $\hat{q}_i^+ - q_i \leq 2\sqrt{\frac{1}{2n_i(t)} \ln(\frac{2n}{\mu})} \leq 2\sqrt{\frac{1}{2l} \ln(\frac{2n}{\mu})}, \forall n_i(t) \geq l,$ w.p. $1 - \frac{\mu}{n}$. Thus, $\hat{q}_i^+ - q_i \leq h^{-1}(\Delta)$ w.p. $1 - \frac{\mu}{n}$. Thus, $\hat{q}^+ - q \leq h^{-1}(\Delta)$ with probability $1-\mu$. From bounded smoothness and monotonicity, $\forall S \subseteq \mathcal{N},\ f_S(q) - f_S(\hat{q}^+) \leq h(h^{-1}(\Delta)) \leq \Delta$ and $f_S(\hat{q}^-) - f_S(q) \leq \Delta$ with probability $1-\mu$. Thus, $f_S(\hat{q}^+) \geq f_S(q) - \Delta$ and $f_{S^*}(\hat{q}^-) \leq f_{S^*}(q) + \Delta$. The proof statements then follows from Δ-separated property. □

Lemma 4.3. *If a non-optimal set S^t is selected for the task t then there exists an expert $i \in S^t$ such that $n_i(t) \leq \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$ with probability $1-\mu$.*

Proof. A non-optimal subset S^t could be selected in two ways: (a) $f_{S^t}(\hat{q}^+) < \alpha$ but $f_{S^t}(q) > \alpha$ or (b) $f_{S^*}(\hat{q}^-) > \alpha$.

From Lemma 4.2, if $n_i(t) \geq \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})\ \forall i \in S^t$, then, both the conditions are violated and thus a non-optimal subset is not selected. □

Theorem 4.2. *The number of non-optimal rounds by the CCB-NS algorithm is bounded by $\frac{2n}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$ with probability $1-\mu$.*

Proof. Lemma 4.1 shows that the CCB-NS exploitation rounds are optimal rounds. A new parameter $u_i(t)$ is associated with each expert. Whenever a set S^t is selected then, $u_i(t) = u_i(t) + 1$ s.t $i \in S^t$ and $i = \arg\min_{j \in S^t} u_j(t)$. Every time a non-optimal subset S^t is selected, $u_i(t)$ of only one expert is updated with the lowest value of $u_i(t)$ so far, such that $i \in S^t$. Thus, $u_i(t) \leq n_i(t)\ \forall i \in \mathcal{N}\ \forall t \in \{1, \ldots, T\}$. Thus, from Lemma 4.3, the number of exploration rounds is bounded by $\frac{2n}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$ with probability $1-\mu$. □

Corollary 4.3. *The total expected regret is bounded by*

$$\left(1 - \frac{1}{T}\right) \frac{2n}{(h^{-1}(\Delta))^2} \ln(2nT) C(\mathcal{N}) + L,$$

where L is the loss incurred by the requester if the constraint is not satisfied.

The above corollary can be obtained by substituting $\mu = 1/T$. We see that the regret by CCB-NS algorithm matches the lower bound in AAB framework up to a constant factor. We next address the strategic version.

5. Strategic version

5.1. The model

Denote the true cost of an expert i by c_i and the reported cost by $\hat{c}_i$. Thus, the valuation of an expert i is $v_i = -c_i$. We denote the requester as expert 0 and his valuation when the task is allocated to the expert set S, by:

$$v_0(S) = \begin{cases} R & \text{if } f_S(q) < \alpha, \\ -L & \text{otherwise.} \end{cases}$$

R denotes the reward that the requester gets for satisfying the constraint and L is the loss he incurs if the accuracy constraint is not satisfied. Note that the requester is not considered to be strategic. Social welfare $W(S)$ is given by:

$$W(S) = \begin{cases} R - \sum_{i \in S} c_i & \text{if } f_S(q) < \alpha, \\ -L - \sum_{i \in S} c_i & \text{otherwise.} \end{cases}$$

A mechanism $\mathcal{M}$ is denoted by the pair $(\mathcal{A}, \mathcal{P})$, where $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n)$ is the allocation vector where $\mathcal{A}_i$ represents number of tasks allocated to expert i and $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n)$ is the payment vector where $\mathcal{P}_i$ denotes the total payment made to the expert i which depends on the reported cost profile $\hat{c}$. We work in a quasi-linear setting where the utility of every expert is given by:

$$u_i(c_i, \hat{c}; q) = -c_i.\mathcal{A}_i(\hat{c}; q) + \mathcal{P}_i(\hat{c}; q) \, .$$

We consider the problem where a heavy penalty is incurred for providing the wrong answer and thus, the parameter L is large.

An important characterization for truthful mechanisms provided by Myerson [50] states that for a mechanism to be truthful, the allocation rule should be monotone in terms of reported bids by the players. Babaioff et al. [11] provide a generic transformation that takes any monotone allocation rule and outputs a mechanism which is truthful and individually rational. We can use this generic transformation to design the mechanism in our setting.

Since there is randomness involved due to learnt qualities, let us first define every possible random seed. The random variables are the opinions provided by the experts and can affect the learnt qualities and thus the allocation rule.

Definition 5.1 *(Success realization).* A success realization is a matrix $\rho \in \{0, 1\}^{n \times T}$, where each parameter ρ_{it} is an independent Bernoulli random variable with parameter q_i. Thus, for any time t,

$$\rho_{it} = \begin{cases} 1 \text{ with probability } q_i, \\ 0 \text{ with probability } 1 - q_i. \end{cases}$$

Note that depending on the allocation rule, only a part of success realization is observed, for example, if task t is given to expert i then $\rho_{it} = 1$ if his opinion matches with the aggregated opinion and is 0 otherwise. We now define desirable game theoretic notions in our setting. Note that, the allocation and payment rule will depend on success realizations when the true qualities are not known.

Definition 5.2 *(Ex-post incentive compatibility).* A mechanism is ex-post incentive compatible if all the bidders are truthful for every success realization irrespective of the bids of other experts, i.e., $\forall i \in \mathcal{N}, \ \forall \rho \in \{0, 1\}^{n \times T}, \ \forall \hat{c}_i \in [0, 1], \ \forall \hat{c}_{-i} \in [0, 1]^{n-1}$

$$-c_i \mathcal{A}_i(c_i, \hat{c}_{-i}, \rho) + \mathcal{P}_i(c_i, \hat{c}_{-i}, \rho) \geq -c_i \mathcal{A}_i(\hat{c}_i, \hat{c}_{-i}, \rho) + \mathcal{P}_i(\hat{c}_i, \hat{c}_{-i}, \rho) \, .$$

Definition 5.3 *(Ex-post individual rationality).* A mechanism is ex-post individual rational if for every success realization, truth telling does not give negative utility to any player corresponding to any bids of other players,

$$-\hat{c}_i \mathcal{A}_i(\hat{c}_i, c_{-i}, \rho) + \mathcal{P}_i(\hat{c}_i, c_{-i}, \rho) \geq 0 \ \forall \hat{c}_i \in [0, 1], c_{-i} \in [0, 1]^{n-1}, \rho \in \{0, 1\}^{n \times T} \, .$$

Definition 5.4 *(Ex-post monotone allocation rule).* If the allocation rule is monotone with respect to every success realization then we say that it is ex-post monotone, i.e., $\forall i \in \mathcal{N}, \ \forall t \in \{1, 2, \ldots, T\}, \ \forall \hat{c}_i \geq c_i, \ \forall \rho$

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}; \rho) \leq \mathcal{A}_i^t(c_i, c_{-i}; \rho). \tag{7}$$

5.2. The CCB-S algorithm

From [50], monotonicity of an allocation rule is required for incentive compatibility. In Step 7 of CCB-NS, if the set S^t does not satisfy the constraint with $\hat{q}^-$, in Step 9, we add experts to satisfy the constraint. This step does not consider strategic costs and may lead to a violation of monotonicity.

In order to ensure truthfulness, we modify the CCB-NS algorithm to select all the experts instead of minimal set of experts, if the constraint is not satisfied with respect to the lower confidence bound (Step 9). We show that the allocation rule then becomes ex-post monotone. Thus, we can apply results from [11] to achieve an ex-post incentive compatible and ex-post individual rational mechanism. Before going to the formal analysis of CCB-S, we first formally present an important result in [11], which is relevant in our setting:

Theorem 5.1. *[11] Let $\mathcal{A}$ be an ex-post monotone MAB allocation rule. There exists a transformation such that the mechanism $\mathcal{M}$ obtained by applying the transformation to the allocation rule $\mathcal{A}$ satisfies the following properties: (a) $\mathcal{M}$ is ex-post truthful, and ex-post individually rational. (b) For each success realization, the difference in expected welfare between $\mathcal{A}$ and $\mathcal{M}$ is at most γn where $0 < \gamma < 1$ is the parameter provided to the transformation.*

5.3. Analysis of CCB-S

Note that, we cannot apply the VCG payment scheme in this algorithm as computing VCG payments requires the computation of an allocation rule in the absence of expert i which cannot be determined by the algorithm since learning stops after computing the optimal set. In order to design an ex-post incentive compatible and ex-post individual rational mechanism, we apply transformation from Theorem 5.1 and design an ex-post monotone allocation rule. We first present this transformation in the next subsection.

5.3.1. Generic transformation for truthful mechanisms

Note that to get the truthfulness it is sufficient to design the following payment rule for any success realization ρ if the allocation rule is monotone [50]:

$$p_i(\hat{c}_i, \hat{c}_{-i}; \rho) = \hat{c}_i a_i(\hat{c}_i, \hat{c}_{-i}; \rho) - \int_{-\infty}^{\hat{c}_i} a_i(u, \hat{c}_{-i}; \rho) du \ . \tag{8}$$

The challenge here is to compute the integral as the allocation depends on how the successes are observed. To compute this integral, a sampling procedure is used. The sampling procedure takes the bids and produces two random vectors χ and ψ. The vector χ is used for determining the allocation rule and the payments are derived using the vector ψ. For deriving truthful mechanisms, we need sampling procedure to satisfy certain properties which we describe below:

Definition 5.5 *(Self-resampling procedure).* A self-resampling procedure with support $I = [\underline{c}_i, \overline{c}_i]$ and resampling probability $\mu \in (0, 1)$ is a randomized algorithm that outputs random vectors $\chi \in I$ and $\psi \in I$ given the input bid vector $\hat{c} \in I$ and satisfies the following properties, $\forall i \in \mathcal{K}$:

1. $\chi_i(\hat{c}_i)$ and $\psi_i(\hat{c}_i)$ are non-decreasing functions of $\hat{c}_i$.
2. *(A)* With probability $(1 - \mu)$, $\chi_i(\hat{c}_i) = \psi_i(\hat{c}_i) = \hat{c}_i$.
   (B) With probability μ, $\underline{c}_i \geq \chi_i(\hat{c}_i) \geq \psi_i(\hat{c}_i) > \hat{c}_i$.
3. $\mathbb{P}[\chi_i(\hat{c}_i) > a_i | \psi_i(\hat{c}_i) = \hat{c}'_i] = \mathbb{P}[\chi_i(\hat{c}'_i) > a_i] \ \ \forall a_i \geq \hat{c}'_i > \hat{c}_i$.
4. The function $\mathcal{F}(a_i, \hat{c}_i) = \mathbb{P}[\psi_i(\hat{c}_i) < a_i | \psi_i(\hat{c}_i) > \hat{c}_i]$ is called the distribution function of the self resampling procedure. For each $\hat{c}_i$, the function $F(., \hat{c}_i)$ is differentiable and strictly increasing on the interval $I \cap (-\infty, \hat{c}_i)$.

We use the sampling procedure given in Algorithm 2.

Algorithm 2: Self-resampling procedure.

Input: bid $\hat{c}_i \in [\underline{c}_i, \overline{c}_i]$, parameter $\mu \in (0, 1)$
Output: (χ_i, ψ_i) such that $\underline{c}_i \geq \chi_i \geq \psi_i \geq \hat{c}_i$

- **with probability** $(1 - \mu)$
  - $\chi_i \leftarrow \hat{c}_i$, $\psi_i \leftarrow \hat{c}_i$
- **with probability** μ
  - Pick $\hat{c}'_i \in [\hat{c}_i, \underline{c}_i]$ uniformly at random.
  - $\chi_i \leftarrow recursive(\hat{c}'_i)$, $\psi_i \leftarrow \hat{c}'_i$

function Recursive($\hat{c}_i$)

- **with probability** $(1 - \mu)$
  - return $\hat{c}_i$
- **with probability** μ
  - Pick $\hat{c}'_i \in [\hat{c}_i, \underline{c}_i]$ uniformly at random.
  - return Recursive($\hat{c}'_i$)

Lemma 5.1. *[51] The procedure in Algorithm 2 is a self-resampling procedure with distribution* $F(a_i, \hat{c}_i) = \frac{a_i - \hat{c}_i}{\overline{c}_i - \hat{c}_i}$.

The mechanism that outputs the transformed allocation and the payment is now described in Algorithm 3.
We will now show that our algorithm produces an ex-post monotone allocation rule.

Theorem 5.2. *The allocation rule given by the CCB-S algorithm* ($\mathcal{A}^{CCB-S}$) *is ex-post monotone.*

Algorithm 3: Transformation mechanism.

Input: $\forall i$, bids $\hat{c}_i \in [\underline{b}, \overline{b}]$, parameter $\mu \in (0,1)$, allocation rule a
Output: Allocation rule $\tilde{a}$ and the payment rule $\tilde{p}$

- Obtain modified bids as $(\chi, \psi) = ((\chi_1(\hat{c}_1), \psi_1(\hat{c}_1)), (\chi_2(\hat{c}_2), \psi_2(\hat{c}_2)), \ldots, (\chi_n(\hat{c}_n), \psi_n(\hat{c}_n))$ from Algorithm 2
- For each task t, return the allocation $\tilde{a}^t(\hat{c})$ to be the subset of experts in accordance with CCB-S algorithm using bids $\chi(\hat{c})$ i.e. $\tilde{a}_i^t(\hat{c}) = 1$ if expert i is allocated by CCB-S algorithm and is 0 otherwise
- For each task t, ask payment from each expert i, $\tilde{p}_i^t(\hat{c}) = \hat{c}_i \tilde{a}_i(\hat{c}) - R_i$, where,

$$R_i = \begin{cases} \frac{1}{\mu} \frac{a_i(\chi(\hat{c}))}{F_i'(\psi_i(\hat{c}_i), \hat{c}_i)}, & \text{if } \psi_i(\hat{c}_i) < \hat{c}_i \\ 0, & \text{otherwise,} \end{cases}$$

where, $F_i'(\psi_i(\hat{c}_i), \hat{c}_i) = \frac{1}{\overline{c}_i - \hat{c}_i}$

Proof. For notation brevity, let us denote $\mathcal{A}^{CCB-S}$ by $\mathcal{A}$. In order to prove monotonicity, we need to prove the following $\forall \hat{c}_i \geq c_i,\ \forall \rho$:

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}; \rho) \leq \mathcal{A}_i^t(c_i, c_{-i}; \rho),\ \ \forall i \in \mathcal{N},\ \forall t \in \{1, 2, \ldots, T\}.$$

For a fixed success realization ρ, let us denote $\mathcal{A}_i^t(\hat{c}_i, c_{-i}; \rho)$ by $\mathcal{A}_i^t(\hat{c}_i, c_{-i})$ for notation brevity. Since task $t = 1$ is given to all the experts irrespective of their bids, we have $\mathcal{A}_j^1(\hat{c}_i, c_{-i}) = \mathcal{A}_j^1(c_i, c_{-i}) = 1\ \forall j \in \mathcal{N}$. Let t be the largest time step such that, $\forall j$, $\mathcal{A}_j^{t-1}(\hat{c}_i, c_{-i}) = \mathcal{A}_j^{t-1}(c_i, c_{-i}) = t - 1$ (exploration round with $\hat{c}_i$ and c_i) and $\exists i$ such that, $\mathcal{A}_i^t(\hat{c}_i, c_{-i}) \neq \mathcal{A}_i^t(c_i, c_{-i})$.

Since other costs and quality estimates are the same, this can happen only when in one case expert i is selected, while in the other case expert i is not selected. Let the two sets of experts selected with c_i and $\hat{c}_i$ be $S(c_i)$ and $S(\hat{c}_i)$ respectively. Since the optimization problem involves cost minimization and quality updates are the same:

$$\mathcal{A}_i^t(\hat{c}_i, c_{-i}) = t - 1 \text{ which implies } i \notin S(\hat{c}_i), \text{ or}$$

$$\mathcal{A}_i^t(c_i, c_{-i}) = t \text{ which implies } i \in S(c_i)\ .$$

Since $i \notin S(\hat{c}_i)$, the selected set $S(\hat{c}_i)$ satisfies the lower confidence bound too (exploitation round with bid $\hat{c}_i$) and thus for the rest of the tasks, only $S(\hat{c}_i)$ is selected and thus we have, $\mathcal{A}_i^t(\hat{c}_i, c_{-i}) \leq \mathcal{A}_i^t(c_i, c_{-i})$. □

Corollary 5.3. *The transformation mechanism in Algorithm 3 is ex-post incentive compatible and ex-post individual rational mechanism [45].*

Regret analysis

The proposed algorithm is adaptive exploration separated and the number of suboptimal rounds for CCB-S is bounded by $\frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$. In Lemma 5.2, we prove that after $l = \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$ steps, there is no set S which satisfies the constraint with respect to upper confidence bound and its cost is less then the optimal cost. Moreover, after $l = \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$ steps, we have $f_{S^*}(\hat{q}^-) < \alpha$ with probability $1 - \mu$.

Lemma 5.2. *A.1 After $l = \frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$ number of uniform exploration rounds,*

1. *for all sets $S \neq S^*$, $f_S(q) > \alpha \implies f_S(\hat{q}^+) > \alpha$ with probability $1 - \mu$*
2. *$f_{S^*}(\hat{q}^-) < \alpha$ with probability $1 - \mu$.*

The proof follows from Lemma 4.2 as after l uniform exploration rounds, we have $n_i(t) \geq l,\ \forall i \in \mathcal{N}$. □

As a result of Lemmas 4.1 and 5.2, we have the following theorem which gives us the bound on the number of non-optimal rounds:

Theorem 5.4. *The number of non-optimal rounds of the CCB-S algorithm is bounded by $\frac{2}{(h^{-1}(\Delta))^2} \ln(\frac{2n}{\mu})$ with probability $1 - \mu$.*

Proof.

- From Lemma 4.1, the CCB-S exploitation rounds are optimal rounds.
- From Lemma 5.2, the number of exploration rounds is bounded by $\frac{2}{(h^{-1}(\Delta))^2}\ln(\frac{2n}{\mu})$ with probability $1-\mu$.

Hence the theorem follows. □

Remark (1): The number of exploration steps in algorithm CCB-S is adaptive unlike the algorithms presented in [12,44] and bounds on the number of exploration steps depend on the parameters Δ and μ.
Remark (2): When the value of Δ is very small compared to T i.e. $(h^{-1}(\Delta))^2 < \frac{1}{T}\ln\left(\frac{2n}{\mu}\right)$, then the algorithm might not converge before T time steps. In a classical MAB algorithm, for example UCB1, there is an inverse dependence on Δ_* (difference between sub-optimal arm and optimal arm). If Δ_* is low, then UCB1 suffers a large regret. For practical situations, where Δ is very low, the requester could provide a range for target accuracy to circumvent high regret. More details are given in Section 6.3.

Corollary 5.5. *The total expected regret is bounded by*

$$\left(1-\frac{1}{T}\right)\frac{2}{(h^{-1}(\Delta))^2}\ln(2nT)C(\mathcal{N})+L,$$

where L is the loss incurred by the requester if the constraint is not satisfied.

6. Practical aspects and experimental results

Until now, our focus was on a combinatorial framework that solves a general optimization problem. A naive implementation of the CCB-NS algorithm and the CCB-S algorithm could have two problems: 1) high computational complexity of the underlying optimization problem 2) high cost of exploration for the CCB-S algorithm.

Often, the underlying optimization problems are well studied combinatorial problems. Due to this, we may still be able to use the AAB framework to address the complexity concerns through efficient approximation algorithms that satisfy monotonicity that we define later. In this section, we consider the majority rule as the aggregation rule and solve Example 3.1 by formulating it as a minimum knapsack problem. The minimum knapsack problem is NP-hard, however, there exists polynomial time greedy approximate algorithm that yields a factor of 2 approximation for this problem.

To ensure truthfulness, CCB-S selects all the experts in the exploration steps and this might result in very high cost when n is large. In general, it is difficult to eliminate the low quality or high cost experts due to the combinatorial nature of the problem. However, if there exists a structure to the optimization problem, it is often possible to eliminate the experts. In the approximate solution of the minimum knapsack problem, we show that it is possible to identify early and eliminate experts of high cost and low quality in the CCB-S algorithm. This elimination avoids high cost of exploration.

6.1. Working with approximate solutions

The key to incorporating an approximate algorithm in the AAB framework is to show its monotonicity with respect to cost. We show that the CCB-S algorithm that uses the solution returned by the monotone approximate algorithm gives a monotone allocation rule.

Definition 6.1 *(Monotone algorithm).* An algorithm is said to be monotone if the allocation $\mathcal{A}$ returned by the algorithm is monotone in cost i.e. if two input instances are (c,q) and (c^+,q) such that $c_i < c_i^+$, for some i and $c_j = c_j^+ \ \forall j \neq i$, then $\mathcal{A}_i(c^+,q)=1 \Rightarrow \mathcal{A}_i(c,q)=1$.

Definition 6.2 *((β,γ) Approximate algorithm).* An algorithm is said to be a (β,γ) approximate algorithm, if for some $\beta \geq 1$ and $\gamma \leq 1$, the solution set S returned by the algorithm is such that $\mathbb{P}[C(S) \leq \beta C(S^*)] \geq \gamma$. Here, S^* is the solution returned by optimal algorithm.

Proposition 6.1. *If there exists a (β,γ) approximation algorithm that is monotone, then, incorporating that (β,γ) approximation scheme in the CCB-S algorithm will result in an ex-post monotone allocation rule.*

This is easy to see from Theorem 5.2. Note that, all the experts are selected in the exploration rounds, and in exploitation rounds, if an expert i is selected with a certain cost, he will also be selected with a lower cost due to the monotonicity property of the approximate algorithm.
Note: One can define the approximate notion of regret in this setting by incorporating the approximation factors β and γ similar to the notion defined in [52]. We define this notion for the specific example of minimum knapsack problem in the next section.

We now present an example of the minimum knapsack optimization problem and a greedy solution of the problem. In the greedy solution, it is possible to eliminate the experts without violating monotonicity condition, thus, avoiding high cost of exploration. We present the elimination strategy for this example.

6.2. An illustrative example with low regret

Algorithm 4: Greedy algorithm (GA).

Input: Set of experts $\mathcal{N}$, parameter M, cost vector $c = (c_1, \ldots, c_i, \ldots, c_n)$, quality parameters $a = (a_1, \ldots, a_i, \ldots, a_n)$
Output: $(2, 1)$-approximate optimal set S

1. Arrange the experts in ascending order of their c_i/a_i ratio. Without loss of generality, let us assume that the experts are indexed such that $\frac{c_1}{a_1} \leq \frac{c_2}{a_2} \leq \ldots \leq \frac{c_n}{a_n}$.
2. Let k_1 be the index of the expert such that $\sum_i^{k_1} a_i < M$ but $\sum_i^{k_1} a_i + a_{k_1+1} \geq M$. Let $S_0 = \{1, 2, \ldots, k_1\}$.
3. Let k_2 be the index of the expert such that $\sum_{i=1}^{k_1} a_i + a_j \geq M\ \forall k_1 + 1 \leq j \leq k_2 - 1$, but $\sum_{i=1}^{k_1} a_i + a_{k_2} < M$. Let $B_0 = \{k_1 + 1, k_1 + 2, \ldots, k_2 - 1\}$.
4. Let k_3 be the index of the expert such that $\sum_i^{k_1} a_i + \sum_{k_2}^{k_3} a_i < M$ but $\sum_i^{k_1} a_i + \sum_{k_2}^{k_3} a_i + a_{k_3+1} \geq M$. Let $S_1 = \{k_2, k_2 + 1, \ldots, k_3\}$.
5. In general let $S_l = \{k_{2l}, k_{2l} + 1, \ldots, k_{2l+1}\}$ and $B_l = \{k_{2l+1} + 1, k_{2l+1} + 2, \ldots k_{2l+2} - 1\}$, where $k_0 = 1$ and k_{2l} is such that: $\sum_{j=0}^{l} \sum_{i=k_{2j}}^{k_{2j+1}} a_i < M$ but $\sum_{j=0}^{l} \sum_{i=k_{2j}}^{k_{2j+1}} a_i + a_m \geq M\ \forall k_{2l+1} + 1 \leq m \leq k_{2l+2} - 1$.
6. Among the sets, $S_1 \cup S_2 \cup \ldots \cup S_l \cup \{j\}$ s.t. $j \in B_l$, pick the set which has the minimum cost and output that as the solution.

From Example 3.1, if all the experts have qualities of at least $\frac{2}{3}$, i.e. $q_i > 2/3$ and $\epsilon = 1/6$, then the optimization problem of minimizing cost and satisfying the accuracy constraint of α can be formulated as follows:

$$\min_{S \in \mathcal{N}} C(S) \text{ s.t. } \sum_{i \in S} (2q_i - 1) \geq 6 \ln\left(\frac{1}{\alpha}\right)$$

This turns out to be the minimum knapsack problem when $c_i \geq 0$ and $2q_i - 1 \geq 0\ \forall i$. Denote $a_i = 2q_i - 1$ and $M = 6 \ln\left(\frac{1}{\alpha}\right)$, we have the following optimization problem:

$$\min_{S \in \mathcal{N}} C(S) \text{ s.t. } \sum_{i \in S} a_i \geq M$$

6.2.1. Greedy algorithm (GA)

The minimum knapsack problem has a greedy deterministic algorithm which gives a $(2, 1)$-approximate solution [53]. The algorithm denoted by GA is described in Algorithm 4:

Lemma 6.1. *[53]. Let, S^{GA} and S^* be the solutions returned by the algorithm GA and optimal algorithm respectively. The greedy algorithm GA gives a solution which is $(2, 1)$-approximate i.e. $C(S^{\text{GA}}) \leq 2C(S^*)$*

Lemma 6.2. *The allocation rule given by greedy algorithm is monotone in cost i.e. if expert i gets a task with cost c_i, he also gets a task with cost c_i^- when the costs and the qualities of the other experts are fixed and $c_i^- < c_i$.*

Proof. Let expert i be selected with the cost c_i. Call the sets $S_0, S_1, \ldots, S_q$ as small sets and the elements of these sets as small elements since the constraint is not satisfied with these elements. Similarly, the sets $B_0, B_1, \ldots, B_q$ are called as big sets and the elements of these sets are called as big elements. Let the set returned by the algorithm be $S_0 \cup S_1 \cup \ldots \cup S_q \cup \{j\}$ where, $j \in B_q$ with cost c_i. Now, consider following cases:

1. With cost c_i, expert i belongs to set S_l where $l \leq q$: Now consider the following cases when the cost of expert i is decreased from c_i to c_i^-:
   (a) Worker i remains in S_l but can appear before some other experts in S_l. In this case, nothing will change as no other expert has changed positions. Thus, an optimal solution will still be $S_0 \cup S_1 \cup \ldots \cup S_q \cup \{j\}$ since cost has only reduced and expert i will get selected.
   (b) Worker i moves to some S_m with $m \leq l$. Since i was already in the small set, S_m will remain small. Moreover, all the other experts from small sets till S_q remains small. Thus, the optimal solution will not change and i will get selected.
   (c) Note that the expert i can never become a big element by reducing cost.
2. With cost c_i, expert $j = i$. Thus, i is a big element with cost c_i. Again, consider the following cases when expert i changes his bid to c_i^-:
   (a) Worker i becomes big element such that $i \in B_m$ with $m \leq q$. The optimal set will be $S_0 \cup S_1 \cup \ldots \cup S_m \cup \{i\}$ and hence i remains in the solution.

(b) Worker i becomes small such that $i \in S_m$ with $m \leq q$. Since i was big till set S_q, an expert k from some small set S_l with $l \leq q$ will become big. Then, the optimal set will be $S_0 \cup S_1 \cup \ldots S_l \cup \{k\}$ will become optimal and hence i will be selected. □

6.2.2. Elimination strategy with greedy algorithm GA

We now provide an algorithm with elimination strategy where all the experts need not be selected in exploration rounds. We call this strategy as CCB-SE and is provided in Algorithm 5.

Algorithm 5: CCB-SE algorithm.

Input: Set of experts $\mathcal{N}$, number of tasks T, parameter α, confidence level μ, bid vector $\hat{c} = (\hat{c}_1, \ldots, \hat{c}_i, \ldots, \hat{c}_n)$
Output: Expert selection set S^t, opinion $\hat{y}^t$ for all tasks $t \in \{1, 2, \ldots, T\}$

$\forall i \in \mathcal{N}$, $\hat{q}_i^+ = 1$, $\hat{q}_i^- = 0$ // Initialize UCB and LCB on qualities
$S^1 = \mathcal{N}$ // Select all experts initially
Observe $\tilde{y}(S^1)$ and $\hat{y}^1 = \text{AGGREGATE}(\tilde{y}(S^1))$ (Definition 4.1)
$\forall i \in \mathcal{N}$, $n_i(1) = 1$, update $\hat{q}_i$
$\hat{q}_i^+ = \hat{q}_i + \sqrt{\frac{1}{2n_i(t)} \ln(\frac{2n}{\mu})}$, $\hat{q}_i^- = \hat{q}_i - \sqrt{\frac{1}{2n_i(t)} \ln(\frac{2n}{\mu})}$
$\hat{a}_i^- = (2\hat{q}_i^- - 1)$ and $\hat{a}_i^+ = (2\hat{q}_i^+ - 1)$, $M = 6\ln(\frac{1}{\alpha})$, $S^{\text{ACTIVE}} = \mathcal{N}$
for $t = 2$ *to* T **do**
  $S^t = GA(S^{\text{ACTIVE}}, M, \hat{c}, \hat{a}^+)$
  if $\sum_{i \in S^t} \hat{a}_i^- \geq M$ **then**
    return S^t for all $t' \in \{t, t+1, \ldots T\}$ // Exploitation Starts
    exit;
  else
    $S^{\text{ACTIVE}} = GA(S^{\text{ACTIVE}}, M, \hat{c}, \hat{a}^-)$ // Exploration Phase
    Let $S^{\text{ACTIVE}} = \{1, 2, \ldots, k\}$ and $\frac{\hat{c}_1}{\hat{a}_1^-} \leq \frac{\hat{c}_2}{\hat{a}_2^-} \leq \ldots \leq \frac{\hat{c}_k}{\hat{a}_k^-}$
    for $i \notin S^{\text{ACTIVE}}$ **do**
      if $\frac{\hat{c}_i}{\hat{a}_i^+} < \frac{\hat{c}_k}{\hat{a}_k^-}$ *OR* $c_i < c_l$ *for some* $l \in \{1, 2, \ldots, k\}$ **then**
        $S^{\text{ACTIVE}} = S^{\text{ACTIVE}} \cup \{i\}$
    $S^t = S^{\text{ACTIVE}}$
    Observe judgements of selected labelers $\tilde{y}(S^t)$
    $\hat{y}^t = \text{AGGREGATE}(\tilde{y}(S^t))$
    for $i \in S^t$ **do**
      $n_i(t) = n_i(t-1) + 1$
      Update $\hat{q}_i$, $\hat{q}_i^+ = \hat{q}_i + \sqrt{\frac{1}{2n_i(t)} \ln(\frac{2n}{\mu})}$, $\hat{q}_i^- = \hat{q}_i - \sqrt{\frac{1}{2n_i(t)} \ln(\frac{2n}{\mu})}$
      $\hat{a}_i^- = (2\hat{q}_i^- - 1)$ and $\hat{a}_i^+ = (2\hat{q}_i^+ - 1)$

The algorithm CCB-SE maintains an active set denoted by S^{ACTIVE}. For a task t, we first solve the optimization problem with respect to upper confidence bound by calling the Greedy algorithm. We then check if the constraint is satisfied with respect to lower confidence bound or not. If the constraint is satisfied with LCB then we return the selected set for all the future rounds. However, if the constraint is not satisfied then we call the greedy algorithm with respect to lower confidence bounds and eliminate experts intelligently. If for some task t, the set S^t is selected by the algorithm, then the set S^t satisfies the constraint with respect to the lower confidence bounds (since the algorithm GA is called with respect to lower confidence bounds). Let us assume that the set returned by algorithm GA with lower confidence bound is denoted by $S^{\text{GA}} = \{1, 2, \ldots, k\}$. Further, if there exists an expert $r \notin S^{\text{GA}}$ such that $\frac{c_k}{\hat{a}_k^-} \leq \frac{c_r}{\hat{a}_r^+}$ and $c_r \geq c_i, \forall i \in S^{\text{GA}}$, then expert r can be discarded "safely". By safely, we mean that with qualities known perfectly, GA algorithm has a candidate solution of cost less than or equal to any candidate solution containing r with probability $(1 - \mu)$ and there is no need to consider expert r for the future tasks. In the next Lemma, we prove the correctness of CCB-SE.

Lemma 6.3. *If an expert r does not belong to S^{ACTIVE} for some task t in CCB-SE, then, with known qualities, GA algorithm has a candidate solution of cost less than or equal to any candidate solution containing r with probability $(1 - \mu)$.*

Proof. With true qualities, in GA algorithm, the elements $1, \ldots, k$ precedes r due to $\frac{c_k}{\hat{a}_k^-} \leq \frac{c_r}{\hat{a}_r^+}$ with probability $(1 - \mu)$.

As $\{1, \ldots, k\}$ meets the accuracy constraint with LCB, they meet it with true qualities also (with high probability). Therefore, there exists a $p \in \{1, \ldots, k\}$, which belongs to a big set in the run of GA with true qualities. Any candidate set with

p in the run will be of the form $\cup_{i=1}^{q} S_i \cup \{p\}$. Any candidate set with r will be of the form $\cup_{i=1}^{l} S_i \cup \{r\}$ with $l \geq q$. Therefore, in the run of GA, we can ignore any candidate solutions with r and hence r can be dropped safely. This is because $C(\cup_{i=1}^{q} S_i \cup \{p\}) \leq C(\cup_{j=1}^{l} S_j \cup \{r\})$ as $\cup_{i=1}^{q} S_i \subseteq \cup_{j=1}^{l} S_j$ and $c_p \leq c_r$. □

Lemma 6.4. *CCB-SE produces monotone allocation rule.*

Proof. In the exploration phase, if the expert i reduces his cost, he can be eliminated at a later stage only. Thus the number of allocations in the exploration phase increases. In exploitation phase, the monotonicity is immediate from Lemma 6.2. □

6.2.3. Regret analysis of CCB-SE

Since, with known qualities also we will solve the optimization problem in the approximate sense, we first define the approximate regret in this setting.

Definition 6.3 *(Approximate regret of CCB-SE).* The approximate regret of CCB-SE when the accuracy constraint is satisfied with probability at least μ is given as:

$$\mathbb{E}[\mathcal{R}(\mathcal{A})] = (1-\mu)\left(\sum_{t=1}^{T}\sum_{i\in S^t} c_i - 2T\sum_{i\in S^*} c_i\right) + \mu LT\,, \tag{9}$$

where S^* is the optimal set with minimum cost and L is the loss incurred by the requester if the constraint is not satisfied.

We now have the following theorem that bounds the approximate regret of CCB-SE

Theorem 6.1. *The approximate regret of CCB-SE is bounded by*

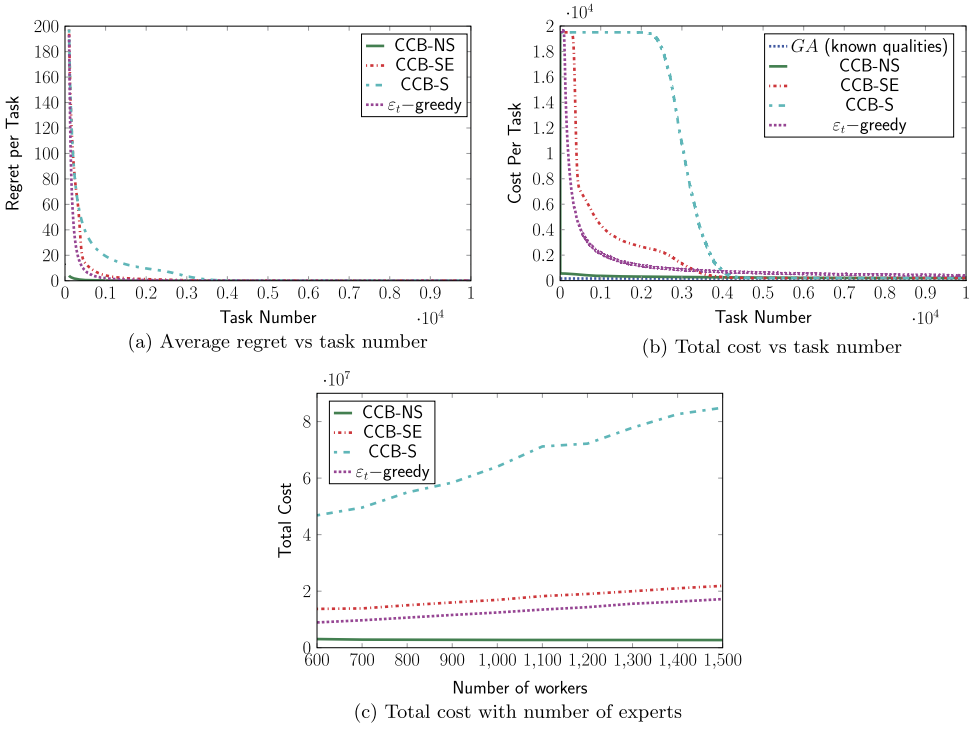
$$\left(1-\frac{1}{T}\right)\frac{2}{(h^{-1}(\Delta))^2}\ln(2nT)C(\mathcal{N}) + L.$$

Proof. Let the subset selected with quality profile q by GA algorithm be denoted by $S^{GA}(q)$ and the optimal subset with quality profile $\hat{q}^+$ be $S^*(\hat{q}^+)$. Let the optimal subset with true qualities be denoted by S^*. If the algorithm goes inside the If loop for task t^* in Step 9, then we prove that $C(S^{t^*}) \leq 2C(S^*)$. From, Lemma 6.3, we never eliminate the worker that can be a part of GA algorithm with respect to true qualities i.e. $S^{GA}(q) \subset S^{\text{ACTIVE}}\ \forall t$. We ensure this by ensuring that we never eliminate the worker that is a part of GA algorithm with respect to UCB on qualities. Thus we also have $S^{GA}(\hat{q}^+) \subset S^{\text{ACTIVE}}\ \forall t$. By the property of GA algorithm, we also have $C(S^{GA}(\hat{q}^+)) \leq 2C(S^*(\hat{q}^+))$. And by monotonicity of error probability function we have $C((S^*(\hat{q}^+)) \leq C(S^*)$. Since the constraint is satisfied with respect to UCB by S^{t^*}, we have $S^{t^*} = S^{GA}(\hat{q}^+)$. Thus, $C(S^{t^*}) \leq 2C(S^*(\hat{q}^+)) \leq 2C(S^*)$. Thus, it is enough to bound the number of tasks t^*. One can bound t^* using the similar steps as is done in Theorem 4.2. Note that since we are eliminating the experts, the CCB-SE is uniform exploration algorithm with elimination. Thus, similar to strategic setting, in the regret expression n does not appear in the numerator. □

6.3. Simulation results

We now compare the efficacy of the proposed algorithms via simulations. We use the minimum knapsack problem described in the previous section solved using GA algorithm. We compare the regret of four algorithms namely, CCB-NS, CCB-S, CCB-SE and a variant of the ε_t-greedy algorithm. The ε_t-greedy algorithm [14] solves the classical multi-armed bandit problem which involves the selection of the single best arm. In the classical version, a random arm is explored with probability ε_t and the optimal arm (with the highest empirical mean) is selected with probability $1-\varepsilon_t$. We extend the algorithm to the AAB setting by exploring all the experts with probability ε_t and selecting minimum cost expert subset meeting the constraint with empirically estimated qualities with probability $(1-\varepsilon_t)$. The parameter $\varepsilon_t = \min\{1, \frac{100}{t}\}$ decreases with time to give more weight to exploitation than exploration. Note that, the ε_t algorithm is not strategyproof.

In the simulations, we have selected the number of experts to be 1100. To emphasize the fact that the CCB-SE algorithm identifies bad experts early, out of the 1100 experts, 600 experts are chosen with cost as 20 and quality as $2/3$ whereas, the other 500 experts are chosen with the costs uniformly drawn between 10 and 20 and the quality uniformly drawn between $2/3$ and 1. The required target accuracy is chosen to be 0.9 with $\alpha = 0.1$. Since the value of Δ can be arbitrarily low, we adopt the following strategy for the implementation. We solve the optimization problem with UCB for a target accuracy of 0.95 but check the lower confidence bound with target accuracy 0.9. This ensures that the constraint is never violated, however, it may result in extra cost of experts for the rest of the rounds. Since the costs of the experts are not adversarially chosen, the expected difference between the optimal set with accuracy 0.9 and 0.95 is not large. In general, if the requester gives a target accuracy range of $(1-\alpha, 1-\alpha+\xi)$ such that the upper confidence bound is solved using accuracy $1-\alpha+\xi$ but the lower bound is checked with accuracy $1-\alpha$, then it is possible to control the number of non-optimal rounds



(a) Average regret vs task number

(b) Total cost vs task number

(c) Total cost with number of experts

Fig. 1. The plots show averages over 1000 samples. Note that CCB-NS and ε_t-greedy algorithms are not ex-post incentive compatible whereas CCB-S and CCB-SE are.

and it can be shown that the number of non-optimal rounds is at most $\min\left(\frac{1}{16(h^{-1}(\xi))^2} ln\left(\frac{2n}{\mu}\right), \frac{2}{(h^{-1}(\Delta))^2} ln\left(\frac{2n}{\mu}\right)\right)$. In the ε_t-greedy algorithm, the expert set is chosen such that the target accuracy of 0.9 is achieved with respect to the estimated qualities. For simulations, we have chosen T to be 10^4. However, if T is large, one can choose a smaller value of ξ. Over 1200 runs of simulations, we observed that none of the four algorithms violated the stochastic constraint with respect to the true qualities. With $\xi = 0.05$, the comparison of the average regret and the negative social welfare is given in Figs. 1a and 1b respectively. The regret is compared against the greedy solution returned by GA algorithm with true qualities. We ran 1000 samples to generate the graphs. We see that the algorithm CCB-NS converges much faster when compared to the ε_t-greedy algorithm. We also see that the cost of CCB-SE algorithm reduces significantly in a few iterations only. We also compare the total cost between CCB-NS algorithm and ε_t-greedy algorithm with change in the number of experts (Fig. 1c). The simulations show that the CCB-NS algorithm outperforms the ε_t-greedy algorithm even when there are fewer number of experts.

7. Summary and future work

We considered the problem of selecting an optimal subset of the experts so as to ensure a certain target accuracy for each task. We proposed a novel framework, Assured Accuracy Bandit (AAB) and developed an algorithm, Strategic Constrained Confidence Bound (CCB-S) for the same, which also leads to an ex-post incentive compatible and ex-post individually rational mechanism. We have provided bounds on the number of exploration steps that depends on the target accuracy level and the true qualities. Often, the optimization problem to be solved for each task inherently has exponential time complexity. In most cases, there exist efficient approximate algorithms for solving the optimization problem. If these algorithms are monotone, then the algorithms can be combined with CCB-S algorithm to provide a truthful, IR mechanism.

An interesting line of future research could be to improve the convergence rate of CCB-S. The slow convergence of CCB-S can be attributed to the algorithm being exploration separated. If there exists a structure to the algorithm for solving the combinatorial optimization problem, then some strategy for eliminating experts in the strategic setting can be adapted. We have seen a strategy in one example. A generalization of this to all possible optimization problem may require more assumption on the function $f_S(q)$ and forms an interesting future direction. Working with soft constraint formulation of this problem forms another extension for the future.

References

[1] M. O'Neil, Shirky and sanger, or the costs of crowdsourcing, J. Sci. Commun. 9 (1) (2010) 1–6.
[2] Technological forecasting, URL http://www.innovation-portal.info/toolkits/technological-forecasting/.

[3] S.R. Fye, S.M. Charbonneau, J.W. Hay, C.A. Mullins, An examination of factors affecting accuracy in technology forecasts, Technol. Forecast. Soc. Change 80 (6) (2013) 1222–1231.
[4] S. Mavandadi, S. Feng, F. Yu, S. Dimitrov, K. Nielsen-Saines, W.R. Prescott, A. Ozcan, A mathematical framework for combining decisions of multiple experts toward accurate and remote diagnosis of malaria using tele-microscopy, PLoS ONE 7 (10) (2012) e46192.
[5] L. Berrang-Ford, K. Garton, Expert knowledge sourcing for public health surveillance: national tsetse mapping in Uganda, Soc. Sci. Med. 91 (2013) 246–255.
[6] T. Lai, H. Robbins, Asymptotically efficient adaptive allocation rules, Adv. Appl. Math. 6 (1) (1985) 4–22.
[7] L. Tran-Thanh, S. Stein, A. Rogers, N.R. Jennings, Efficient crowdsourcing of unknown experts using multi-armed bandits, in: Proceedings of IEEE 3rd International Conference of Social Computing, 2012, pp. 768–773.
[8] L. Tran-Thanh, M. Venanzi, A. Rogers, N.R. Jennings, Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks, in: Twelfth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13), 2013, pp. 901–908.
[9] L. Tran-Thanh, S. Stein, A. Rogers, N.R. Jennings, Efficient crowdsourcing of unknown experts using bounded multi-armed bandits, Artif. Intell. 214 (2014) 89–111.
[10] M. Babaioff, Y. Sharma, A. Slivkins, Characterizing truthful multi-armed bandit mechanisms: extended abstract, in: Tenth ACM Conference on Electronic Commerce, EC'09, ACM, 2009, pp. 79–88.
[11] M. Babaioff, R.D. Kleinberg, A. Slivkins, Truthful mechanisms with implicit payment computation, in: Eleventh ACM Conference on Electronic Commerce, EC'10, ACM, 2010, pp. 43–52.
[12] N.R. Devanur, S.M. Kakade, The price of truthfulness for pay-per-click auctions, in: Tenth ACM Conference on Electronic Commerce, EC'09, 2009, pp. 99–106.
[13] A. Das Sarma, S. Gujar, Y. Narahari, Truthful multi-armed bandit mechanisms for multi-slot sponsored search auctions, Curr. Sci. 103 (9) (2012) 1064–1077.
[14] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, Mach. Learn. 47 (2–3) (2002) 235–256.
[15] C.-J. Ho, S. Jabbari, J.W. Vaughan, Adaptive task assignment for crowdsourced classification, in: International Conference on Machine Learning, ICML'13, vol. 28, 2013, pp. 534–542.
[16] I. Abraham, O. Alonso, V. Kandylas, A. Slivkins, Adaptive crowdsourcing algorithms for the bandit survey problem (colt'13), in: S. Shalev-Shwartz, I. Steinwart (Eds.), Conference on Learning Theory, in: JMLR Proceedings, vol. 30, 2013, pp. 882–910, JMLR.org.
[17] J. Fan, G. Li, B.C. Ooi, K.-l. Tan, J. Feng, Icrowd: an adaptive crowdsourcing framework, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, ACM, 2015, pp. 1015–1030.
[18] O.S. David R. Karger, S. Devavrat, Budget-optimal crowdsourcing using low-rank matrix approximations, in: 49th Annual Conference on Communication, Control, and Computing, Allerton, 2011, pp. 284–291.
[19] V.C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, J. Mach. Learn. Res. 11 (2010) 1297–1322.
[20] A.P. Dawid, A.M. Skene, Maximum likelihood estimation of observer error-rates using the EM algorithm, J. R. Stat. Soc., Ser. C, Appl. Stat. 28 (1) (1979) 20–28.
[21] P. Viappiani, S. Zilles, H. Hamilton, C. Boutilier, Learning complex concepts using crowdsourcing: a Bayesian approach, in: Algorithmic Decision Theory, vol. 6992, 2011, pp. 277–291.
[22] M. Babaioff, S. Dughmi, R. Kleinberg, A. Slivkins, Dynamic pricing with limited supply, in: Thirteenth ACM Conference on Electronic Commerce, EC'12, ACM, 2012, pp. 74–91.
[23] A. Singla, A. Krause, Truthful incentives in crowdsourcing tasks using regret minimization mechanisms, in: Twenty Second International World Wide Web Conference, WWW'13, 2013, pp. 1167–1178.
[24] A. Badanidiyuru, R. Kleinberg, Y. Singer, Learning on a budget: posted price mechanisms for online procurement, in: Thirteenth ACM Conference on Electronic Commerce, EC'12, ACM, 2012, pp. 128–145.
[25] Y. Singer, M. Mittal, Pricing mechanisms for crowdsourcing markets, in: Twenty Second International World Wide Web Conference, WWW'13, 2013, pp. 1157–1166.
[26] D. Garg, S. Bhattacharya, S. Sundararajan, S.K. Shevade, Mechanism design for cost optimal PAC learning in the presence of strategic noisy annotators, in: Twenty Eighth Conference on Uncertainty in Artificial Intelligence, UAI'12, 2012, pp. 275–285.
[27] S. Bhat, S. Nath, O. Zoeter, S. Gujar, Y. Narahari, C. Dance, A mechanism to optimally balance cost and quality of labeling tasks outsourced to strategic agents, in: Thirteenth International Conference on Autonomous Agents and Multiagent Systems, AAMAS'14, 2014, pp. 917–924.
[28] J. Witkowski, Y. Bachrach, P. Key, D.C. Parkes, Dwelling on the negative: incentivizing effort in peer prediction, in: Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing, HCOMP'13, 2013, pp. 1–8.
[29] R. Cavallo, S. Jain, Winner-take-all crowdsourcing contests with stochastic production, in: Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing, HCOMP'13, 2013.
[30] S. Gujar, B. Faltings, Dynamic task assignments: an online two sided matching approach, in: Proceedings of the 3rd International Workshop on Matching Under Preferences, MATCHUP'15, 2015.
[31] S. Gujar, B. Faltings, Auction based mechanisms for dynamic task assignments in expert crowdsourcing, in: Proceedings of the International Workshop on Agent Mediated E-Commerce and Trading Agent Design and Analysis, AMEC/TADA'15, 2015.
[32] S. Bubeck, N. Cesa-Bianchi, Regret analysis of stochastic and nonstochastic multi-armed bandit problems, Found. Trends Mach. Learn. 5 (1) (2012) 1–122.
[33] S. Agrawal, N.R. Devanur, Bandits with concave rewards and convex knapsacks, in: Fifteenth ACM Conference on Economics and Computation, EC'14, 2014, pp. 989–1006.
[34] A. Badanidiyuru, R. Kleinberg, A. Slivkins, Bandits with knapsacks, in: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS, IEEE, 2013, pp. 207–216.
[35] E. Even-Dar, S. Mannor, Y. Mansour, Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems, J. Mach. Learn. Res. 7 (2006) 1079–1105.
[36] S. Kalyanakrishnan, P. Stone, Efficient selection of multiple bandit arms: theory and practice, in: International Conference on Machine Learning, ICML'10, 2010, pp. 511–518.
[37] Y. Zhou, X. Chen, J. Li, Optimal pac multiple arm identification with applications to crowdsourcing, in: Proceedings of the 31st International Conference on Machine Learning, ICML-14, 2014, pp. 217–225.
[38] W. Chen, Y. Wang, Y. Yuan, Combinatorial multi-armed bandit: general framework and applications, in: International Conference on Machine Learning, ICML'13, vol. 28, 2013, pp. 151–159.
[39] S. Chen, T. Lin, I. King, M.R. Lyu, W. Chen, Combinatorial pure exploration of multi-armed bandits, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 27, Curran Associates, Inc., 2014, pp. 379–387.
[40] S. Kale, L. Reyzin, R.E. Schapire, Non-stochastic bandit slate problems, in: Advances in Neural Information Processing Systems, 2010, pp. 1054–1062.
[41] B. Kveton, Z. Wen, A. Ashkan, C. Szepesvari, Tight regret bounds for stochastic combinatorial semi-bandits, in: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, 2015, pp. 535–543.

[42] Z. Wen, B. Kveton, A. Ashkan, Efficient learning in large-scale combinatorial semi-bandits, in: Proceedings of the 32nd International Conference on Machine Learning, ICML-15, 2015, pp. 1113–1122.
[43] T. Lattimore, K. Crammer, C. Szepesvári, Optimal resource allocation with semi-bandit feedback, in: 30th Conference on Uncertainty in Artificial Intelligence, 2014.
[44] N. Gatti, A. Lazaric, M. Rocco, F. Trovò, Truthful learning mechanisms for multi-slot sponsored search auctions with externalities, Artif. Intell. 227 (2015) 93–139.
[45] S. Bhat, S. Jain, S. Gujar, Y. Narahari, An optimal bidimensional multi-armed bandit auction for multi-unit procurement, in: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS'15, 2015, pp. 1789–1790.
[46] I. Kremer, Y. Mansour, M. Perry, Implementing the "wisdom of the crowd", J. Polit. Econ. 122 (5) (2014) 988–1012.
[47] Y. Mansour, A. Slivkins, V. Syrgkanis, Bayesian incentive-compatible bandit exploration, in: Proceedings of the Sixteenth ACM Conference on Economics and Computation, ACM, 2015, pp. 565–582.
[48] S. Jain, S. Gujar, O. Zoeter, Y. Narahari, A quality assuring multi-armed bandit crowdsourcing mechanism with incentive compatible learning, in: Thirtheenth International Conference on Autonomous Agents and Multiagent Systems, AAMAS'14, 2014, pp. 1609–1610.
[49] H. Li, B. Yu, D. Zhou, Error rate bounds in crowdsourcing models, arXiv preprint arXiv:1307.2674, 2013.
[50] R.B. Myerson, Optimal auction design, Math. Oper. Res. 6 (1) (1981) 58–73.
[51] S. Jain, S. Bhat, G. Ghalme, D. Padmanabhan, Y. Narahari, Mechanisms with learning for stochastic multi-armed bandit problems, Indian J. Pure Appl. Math. 47 (2) (2016) 229–272, https://doi.org/10.1007/s13226-016-0186-3.
[52] M. Streeter, D. Golovin, An online algorithm for maximizing submodular functions, in: Proceedings of the 21st International Conference on Neural Information Processing Systems, NIPS'08, Curran Associates Inc., USA, 2008, pp. 1577–1584. URL http://dl.acm.org/citation.cfm?id=2981780.2981977.
[53] J. Csirik, J.B.G. Frenk, M. Labbé, S. Zhang, Heuristics for the 0–1 min-knapsack problem, Acta Cybern. 10 (1–2) (1991) 15–20.