BLISS: Budget LImited robuSt crowdSensing through Online Learning

Kai Han*† Chi Zhang* Jun Luo*
*School of Computer Engineering, Nanyang Technological University, Singapore
†School of Computer Science, Zhongyuan University of Technology, China
Email: hankai@gmail.com, {czhang8, junluo}@ntu.edu.sg

***Abstract*—Mobile crowdsensing has been intensively explored recently due to its flexible and pervasive sensing ability. Although many crowdsensing platforms have been built for various applications, the general issue of how to manage such systems intelligently remains largely open. While recent investigations mostly focus on incentivizing crowdsensing, the robustness of crowdsensing toward uncontrollable sensing quality, another important issue, has been widely neglected. Due to the non-professional personnel and devices, the quality of crowdsensing data cannot be fully guaranteed, hence the revenue gained from mobile crowdsensing is generally uncertain. Moreover, the need for compensating the sensing costs under a limited budget has exacerbated the situation: one does not enjoy an infinite horizon to learn the sensing ability of the crowd and hence to make decisions based on sufficient statistics. In this paper, we present a novel framework, Budget LImited robuSt crowdSensing (BLISS), to handle this problem through an online learning approach. Our approach aims to minimize the difference on average sense (a.k.a. *regret*) between the achieved total sensing revenue and the (unknown) optimal one, and our BLISS sensing policy is shown to be asymptotically optimal. Finally, we use extensive simulations to demonstrate the effectiveness of BLISS.**

I. Introduction

Given the pervasive availability of hand-held mobile devices (in particular the increasingly powerful smart phones), the concept of *Mobile Crowdsensing* [1] has started a new sensing paradigm, where human crowds (along with their mobile devices) are not only consumers of the sensed data but also their producers. Thanks to the huge number of pervasively available mobile sensors (those embedded in smart phones) and their virtually unlimited spatial-temporal coverage, the efficiency (in gathering a sufficient amount of data) and the ubiquity (in capturing relevant events) are the major strengths of this new sensing paradigm. Consequently, there have recently emerged many interesting mobile crowdsensing applications across a wide variety of research and application domains [2].

However, compared with the traditional remote sensing systems, the mobile crowdsensing paradigm has posed several unique challenges. While an *owner* of a crowdsensing task can save the expenditures of buying and deploying specialized sensors, substantial (preferably monetary) compensation is necessary to drive mobile crowdsensing [3]–[7]. This is so because a *participant* to a mobile crowdsensing task needs to i) move to specific areas where sensing is required, 2) consume his/her smart phone, mostly in terms of the embedded sensors and battery, and iii) probably pay for the 3G access to upload sensing data [6]. Furthermore, as the crowdsensing participants are usually unprofessional (hence resulting in high data missing rate and low sensing quality), the data readings acquired from a single participant may be noisy and of poor data quality [8], [9]. This makes it necessary to require a minimum number of participants for improving sensing robustness. Actually, such a requirement is essential in a lot of crowdsensing applications [3].

Based on the above observations, an astute sensing task owner has to seriously set up a *budget*, and to carefully choose participants so that it can harvest the most from information-gathering under that budget. Whereas this problem seems to fall in a conventional combinatorial optimization framework, the uncertainness of data quality in mobile crowdsensing makes it much more complicated. As neither the involved sensors nor their operators (the crowdsensing participants) are professional, the quality of sensing data cannot be perfectly guaranteed at a certain level. As one typical example, the amount of useful (or qualified) data gathered by a certain participant during a given time span may well be a random number instead of a deterministic function of the sampling rate [9]–[12]. Consequently, the value of the sensing data to the owner can be random, and the owner would certainly need to seek robustness against such an uncertainty subject to the budget limit. To the best of our knowledge, this issue has never been tackled in the literature by far.

We study in this paper a novel robust sensing problem imposed by mobile crowdsensing: an owner aims to repetitively conduct a sensing task under a limited budget, by choosing from a set of available participants whose individual sensing values are random with unknown probabilistic distributions. The problem is combinatorial in nature due to the budget limited selection process, but it is made far more challenging due to the non-deterministic sensing values of individual participants. To this end, we propose *Budget LImited robuSt crowdSensing* (BLISS) as a general framework to tackle the problem. To keep the selection process robust to the uncertainty, we adopt an online learning approach to acquire the statistical information about the sensing values throughout the selection process. Due to the uncontrollable sensing quality, the objective of our robust crowdsensing is to minimize the difference on average sense (a.k.a. *regret*) between the achieved total sensing revenue and the optimal one computed by a genie. In summary, we make the following major contributions in this paper:

- We introduce the robust crowdsensing problem, a realistic yet open issue for many crowdsensing scenarios.
- We propose Budget LImited robuSt crowdSensing (BLISS) as a general framework to tackle this problem.
- We propose an online learning algorithm which has a logarithmic regret bound on the expected total sensing revenue, and we have proven its asymptotical optimality.
- We perform extensive simulations to demonstrate the effectiveness of BLISS.

The remaining of our paper is organized as follows. We introduce the background and models in Sec. II, where we also formulate the BLISS framework. Then we present our BLISS online learning algorithm in Sec. III, and analyze its theoretical performance in Sec. IV. We report the results of our extensive simulations in Sec. V. We finally discuss the related work in Sec. VI, before concluding our paper in Sec. VII. In order to maintain fluency, we postpone all the (sketched) proofs to the Appendix.

II. Modeling and Problem Formulation

We shall first give a brief discussion on the application scenarios, before diving into the mathematical formulations.

A. Background and Scenarios

One of the major revolutions brought by mobile crowdsensing is *urban scale information gathering* [13]–[15]. Traditionally, these information gathering procedures always rely on professional operators and specialized (high-end) sensors (e.g., traffic cameras) that have limited coverage. Mobile crowdsensing, on the contrary, makes use of the pervasive availability of human participants, so it can be made more scalable both spatially and temporally. Nevertheless, the sensing data quality (in terms of data timeliness, relevancy, coverage, etc. [10]) in crowdsensing cannot be perfectly guaranteed due to the unprofessional sensors and the casual behaviours of the participants. This problem gets even more prominent given that many crowdsensing systems are designed to involve the least user intervention (a.k.a. opportunistic sensing) [1], [11]. The sensing in these systems is autonomously activated if predefined conditions have been satisfied, which can result in uncertain data quality and high data-missing rate because sufficient exposure time for sensors may not be guaranteed [9], [11]. Therefore, we cannot fully benefit from mobile crowdsensing in revolutionizing our urban living and working without handling this uncertainty on sensing data quality.

Let us take the dust level sensing task as an example, for which a certain number of participants are chosen in a city to gather information on the dust levels, and finally get remunerated for the data gathered by them. One typical constraint is that a minimum number of participants have to be chosen. This requirement is meaningful to many crowdsensing tasks, for example, the dust level sensing should involve a sufficient amount of participants to improve the overall value of the collected data and/or to cover a sensing area. While the participants may always deliver a sequence of readings either online or offline to claim their remunerations, it is highly possible that not all the readings are qualified due to, for example, a wrong placement of the sensing device. Although other sensors can be used to identify and remove unqualified data (e.g., accelerometers can be used to detect if a smartphone is put into pocket [16] where the dust level readings are not qualified), the amount of qualified data from each participant (hence the sensing value of this participant to the owner) becomes uncontrollable and random.

At the meantime, the task owner has at its disposal a budget and a set of participants to recruit [17]. The owner bears the wish that the total sensing revenue got from all participants is maximized, but it cannot achieve this by a one-time participant selection due to the randomness in data quality. Given the amount of budget to support its mobile crowdsensing task for a certain period of time (e.g., tens of days), a reasonable strategy taken by the owner is to learn the sensing values gradually, smartly reshuffle the selected participants every day, and aim to minimize the gap on average sense between the achieved total sensing revenue and the one obtained by a genie. To summarize, the robust sensing problem raised by the realistic crowdsensing applications share the following features:

F1: At least a minimum number of participants have to be involved for a crowdsensing task.
F2: The sensing values of individual participants are random.
F3: The owner of a crowdsensing task has a limited budget to recruit participants.

B. Models and Assumptions

Suppose that the *owner* of a crowdsensing task $\mathcal{T}$ has a budget G to conduct its task that often lasts for a certain amount of time slots (e.g., tens of days). The owner also has a set of *participants* indexed by $[\![d]\!] = \{1, 2, ..., d\}$[1] at its disposal to actually perform the sensing.

For each participant, we define his/her *sensing value* as the amount of qualified data that he/she collected within one time slot (e.g., a day). According to the discussions in Sec. II-A, this quantity for participant i during the r-th time slot is obviously a random variable, which we denote by $X_{i,r} : r \in \mathbb{Z}^+$. The value of $X_{i,r}$ can only be revealed after participant i is selected and has finished his/her sensing for the r-th time slot. Without loss of generality, we assume that the sequence $\{X_{i,1}, X_{i,2}...\}$ are i.i.d. non-negative random variables following an **unknown** distribution with an unknown expectation τ_i, but the distributions followed by $X_{i,1}$ and $X_{j,1}$ can be different if $j \neq i$. We also assume that $X_{i,r}, \forall i, r \in \mathbb{Z}^+$ has normalized supports in $[0, 1]$, but our results can be easily extended to the case of arbitrary supports of $X_{i,r}$. To evaluate the total sensing revenue, the owner has, for each participant i, a weight ω_i. This weight may represent several factors related to the prior information on this participant's ability of performing the sensing task, such as the types and sampling resolution of the participant's sensors. As a result, the sensing revenue obtained by the owner from participant i in the r-th time slot can be expressed as $\omega_i X_{i,r}$.

[1]We use $[\![z]\!]$ to denote the set $\{1, 2, ..., z\}$ for any $z \in \mathbb{Z}^+$.

During each time slot, the owner selects a certain number of participants to conduct the crowdsensing task. We assume that selecting participant i for one time slot costs the owner p_i, which includes the cost of rewarding i and processing the collected data. Moreover, to accomplish a meaningful sensing task, the number of participants selected by the owner during each time slot must be no less than a predefined positive integer $m : m \leq d$ (see F1 of Sec. II-A). For this requirement, we formally introduce the concept of *Feasible Sensing Engagement* in ***Definition*** 1:

***Definition** 1 (FSE):* A *Sensing Engagement* (SE) is a vector $\mathbf{v} = (v_1, v_2, \cdots, v_d)$, where $v_i \in \{0, 1\}$ indicates whether participant i is selected for sensing. If $\sum_{i=1}^{d} v_i \geq m$, then we term $\mathbf{v}$ a *Feasible Sensing Engagement* (FSE). The set of all FSEs is denoted by $\mathcal{V}$.

If a FSE $\mathbf{v} \in \mathcal{V}$ is selected in the r-th time slot, the owner would get a sensing revenue $\sum_{i=1}^{d} \omega_i X_{i,r} v_i$, but at a cost of $p(\mathbf{v}) = \sum_{i=1}^{d} p_i v_i$. As the owner is not sure about the random sensing values of individual participants, it would adaptively select different FSEs at different points in time. However, as the owner has a budget G, the total time span it can play this "trial-and-error" procedure is limited. We formally define this procedure as a *Robust Sensing Policy*:

***Definition** 2 (RSP):* A *Robust Sensing Policy* of the owner is a sequence $\Phi = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \cdots)$, where $\boldsymbol{\phi}_r = (\phi_{1,r}, \phi_{2,r}, \cdots, \phi_{d,r}) \in \mathcal{V}, \forall r \in \mathbb{Z}^+$ is the FSE selected for the r-th time slot. Also, the policy satisfies $\sum_{r=1}^{\infty} p(\boldsymbol{\phi}_r) \leq G$. The total sensing revenue of Φ is $W_G(\Phi) = \sum_{r=1}^{\infty} \sum_{i=1}^{d} \omega_i X_{i,r} \phi_{i,r}$.

Now the *expected* total revenue of a RSP Φ becomes:

$$R_G(\Phi) = \mathbb{E}\{W_G(\Phi)\} = \sum_{r=1}^{\infty} \sum_{i=1}^{d} \omega_i \tau_i \phi_{i,r} \quad (1)$$

Let $\boldsymbol{\tau}$ denote the vector $(\tau_1, \tau_2, ..., \tau_d)$. For $\mathbf{v} \in \mathcal{V}$, we denote its expected revenue by $f(\mathbf{v}, \boldsymbol{\tau}) = \sum_{i=1}^{d} \omega_i \tau_i v_i$ and the set of selected participants in $\mathbf{v}$ by $\mathcal{H}(\mathbf{v}) = \{i | v_i \neq 0 \wedge i \in [\![d]\!]\}$. The minimum and maximum total costs of an FSE are denoted by p_{min} and p_{max}, respectively, where p_{min} is the sum of m smallest p_i's and $p_{max} = \sum_{i=1}^{d} p_i$.

C. Problem Formulation

Under our Budget LimIted robuSt crowdSensing (BLISS) framework described in Sec. II-B, a desired objective is to find an RSP Φ^* such that $R_G(\Phi^*)$ is maximized. In terms of combinatorial optimization, Φ^* is an optimal solution to the following *integer linear programming* (ILP) problem:

$$\begin{aligned}
&[\text{BLISS-ILP}] \quad \text{Maximize} \quad R_G(\Phi) && (2)\\
&\text{s.t.} \quad \sum_r \sum_{i=1}^{d} p_i \phi_{i,r} \leq G && (3)\\
&\sum_{i=1}^{d} \phi_{i,r} \geq m \cdot y_r && (4)\\
&\phi_{i,r} \leq y_r && (5)\\
&y_r \geq y_{r+1} && (6)\\
&y_r \in \{0,1\}; \quad \phi_{i,r} \in \{0,1\}; \quad \forall r \in \mathbb{Z}^+; \quad \forall i \in [\![d]\!]
\end{aligned}$$

Constraints (3) and (4) are due to the above definitions of FSE and RSP. The variable $y_r \in \{0, 1\}$ denotes whether the sensing task is performed for the r-th time slot. Obviously, the number of non-zero y_r is bounded by $\lfloor G/p_{min} \rfloor$. Constraint (6) is artificially introduced to force these non-zero elements appearing only at the beginning of the time sequence: it confines the problem dimension without sacrificing generality. Constraint (5) states that participants are chosen only when the task is performed. Assuming that the expected sensing values $\{\tau_i\}$ are known, BLISS-ILP can be proven as NP-hard.

***Theorem** 1:* BLISS-ILP is NP-hard when $\boldsymbol{\tau}$ is known.

However, in the BLISS problem, the expected sensing values are not known to us. Therefore, it is not possible to solve BLISS-ILP either optimally or approximately, and it only serves as a benchmark in our performance evaluation.

Based on the above discussions, we will instead adopt an online leaning approach to tackle the robust sensing problem, i.e., the owner repetitively learns the participants' sensing values and chooses the next FSE accordingly until running out of budget. The goal is then to minimize the difference with respect to the optimal solution computed by a genie, essentially a standard optimization objective in the field of online learning [18]. Formally speaking, we aim at an RSP Φ such that the *regret* $R_G(\Phi^*) - R_G(\Phi)$ is minimized.

III. BLISS Online Learning Algorithm

We shall first briefly motivate our algorithm, before presenting its details.

A. Motivations

According to Sec. II-C, we are confronting an "exploration vs. exploitation" dilemma under the BLISS framework, i.e., balancing revenue maximization based on the already acquired empirical knowledge of the sensing values with attempting new FSEs to acquire further knowledge. A popular model for solving such a kind of dilemma is the *Multi-Armed Bandit* (MAB) problem in the area of reinforcement learning [19]. In [19], Auer *et.al.* study the problem of regret minimization for pulling a row of slot machines (or *one-armed bandits*) with unknown i.i.d. rewards over time, and the rule is to pull exactly one arm each time. The UCB algorithm proposed in [19] for multi-armed bandits achieves a storage and regret bound both grow linearly in the number of arms.

Directly applying UCB to our problem faces two major obstacles. Firstly, we have to model each FSE as an arm, hence resulting in $\sum_{i=m}^{d} \binom{d}{i} = \mathcal{O}(2^d)$ arms in total. Consequently, the regret bound and required storage can grow exponentially in the number of participants. Secondly, pulling arms is assumed to be free in UCB (hence the arms can be pulled for ever), whereas our BLISS framework has a budget limit in selecting FSEs. Actually, in the BLISS problem, we aim to achieve a small regret bound while simultaneously respecting the budget limit and the combinatorial nature of FSE selection, which makes the problem extremely challenging.

B. Algorithm Details

To conquer the above difficulties, we propose a BLISS online learning algorithm shown by **Algorithm 1**. In this

Algorithm 1: BLISS Online Learning

Input: $G, m, d, \boldsymbol{\omega} = (\omega_1, ..., \omega_d), \mathbf{p} = (p_1, ..., p_d)$
Output: Φ, N
1 **for** $i = 1$ **to** d **do** $\lambda_{i,0} \leftarrow 0; \quad k_{i,0} \leftarrow 0$
2 $r \leftarrow 1; \quad \boldsymbol{\phi}_r \leftarrow \{1\}^d$
3 $(\boldsymbol{\lambda}_r, \mathbf{k}_r, G) \leftarrow \mathsf{Update}(\boldsymbol{\lambda}_{r-1}, \mathbf{k}_{r-1}, d, r, \boldsymbol{\phi}_r, G)$
4 **while** $true$ **do**
5 $\quad r \leftarrow r + 1$
6 $\quad$ **for** $i = 1$ **to** d **do** $\bar{\lambda}_{i,r} \leftarrow \lambda_{i,r-1} + \sqrt{\frac{5 \ln r}{2k_{i,r-1}}}$
7 $\quad$ Call **Algorithm 2** to find an FSE $\bar{\mathbf{v}}_r$
8 $\quad$ **if** $p(\bar{\mathbf{v}}_r) \le G$ **then**
9 $\quad\quad \boldsymbol{\phi}_r \leftarrow \bar{\mathbf{v}}_r$
10 $\quad\quad (\boldsymbol{\lambda}_r, \mathbf{k}_r, G) \leftarrow \mathsf{Update}(\boldsymbol{\lambda}_{r-1}, \mathbf{k}_{r-1}, d, r, \boldsymbol{\phi}_r, G)$
11 $\quad$ **else break**
12 $N \leftarrow r - 1$
13 **return** (Φ, N)
14 **Function** $\mathsf{Update}(\boldsymbol{\lambda}_{r-1}, \mathbf{k}_{r-1}, d, r, \boldsymbol{\phi}_r, G)$
15 Perform crowdsensing for the r-th time slot based on $\boldsymbol{\phi}_r$
16 **for** $i = 1$ **to** d **do**
17 $\quad$ **if** $\phi_{i,r} > 0$ **then**
18 $\quad\quad \lambda_{i,r} \leftarrow \frac{\lambda_{i,r-1} k_{i,r-1} + X_{i,r}}{k_{i,r-1} + 1}; \quad k_{i,r} \leftarrow k_{i,r-1} + 1$
19 $\quad$ **else**
20 $\quad\quad \lambda_{i,r} \leftarrow \lambda_{i,r-1}; \quad k_{i,r} \leftarrow k_{i,r-1}$
21 $G \leftarrow G - p(\boldsymbol{\phi}_r)$
22 **return** $(\boldsymbol{\lambda}_r, \mathbf{k}_r, G)$

Algorithm 2: Selecting an FSE for the rth time slot

Input: $\bar{\boldsymbol{\lambda}}_r = (\bar{\lambda}_{1,r}, ..., \bar{\lambda}_{d,r}), m, \boldsymbol{\omega}, d, \mathbf{p}, r$
Output: $\bar{\mathbf{v}}_r$
1 $a \leftarrow 0; \quad b \leftarrow \left(\sum_{i=1}^{d} \omega_i \bar{\lambda}_{i,r}\right) / p_{min}$
2 **for** $i = 1$ **to** d **do** $h[i] \leftarrow i; \quad \bar{v}_{i,r} \leftarrow 0$
3 **for** $j = d$ **to** 1 **do**
4 $\quad$ **for** $i = 1$ **to** $j - 1$ **do**
5 $\quad\quad (a, b, z) \leftarrow \mathsf{Max}(a, b, h[i], h[i+1], \bar{\boldsymbol{\lambda}}_r, m, \boldsymbol{\omega}, d, \mathbf{p})$
6 $\quad\quad$ **if** $z = h[i+1]$ **then** $h[i] \leftrightarrow h[i+1]$
7 $Y \leftarrow \left\{ \left\langle \frac{\sum_{1 \le t \le i} \omega_{h[t]} \bar{\lambda}_{h[t],r}}{\sum_{1 \le t \le i} p_{h[t]}}, i \right\rangle \middle| m \le i \le d \right\}$
8 **forall the** $\langle s, i \rangle \in Y \wedge s \in [a, b]$ **do**
9 $\quad u \leftarrow \max \left\{ \sum_{t=1}^{j} (\omega_{h[t]} \bar{\lambda}_{h[t],r} - p_{h[t]} s) \middle| m \le j \le d \right\}$
$\quad$ **if** $u = 0$ **then break**
10 **for** $j = 1$ **to** i **do** $\bar{v}_{h[j],r} \leftarrow 1$
11 **return** $\bar{\mathbf{v}}_r = (\bar{v}_{1,r}, \bar{v}_{2,r}, ..., \bar{v}_{d,r})$
12 **Function** $\mathsf{Max}(a, b, i', j', \boldsymbol{\lambda}_r, m, \boldsymbol{\omega}, d, \mathbf{p})$
13 $i \leftarrow i'; \quad j \leftarrow j'; \quad$ **if** $p_{i'} < p_{j'}$ **then** $i \leftrightarrow j$
14 $o \leftarrow \omega_i \bar{\lambda}_{i,r} - \omega_j \bar{\lambda}_{j,r}$
15 **if** $o \ge b(p_i - p_j)$ **then return** (a, b, i)
16 **if** $o \le a(p_i - p_j)$ **then return** (a, b, j)
17 $l \leftarrow o/(p_i - p_j); \quad U \leftarrow \{\omega_t \bar{\lambda}_{t,r} - l \cdot p_t | 1 \le t \le d\}$
18 $S \leftarrow \arg\max_{S \subseteq U \wedge |S| \ge m} \sum_{s \in S} s$
19 **if** $\sum_{s \in S} s > 0$ **then return** (l, b, j)
20 **else return** (a, l, i)

algorithm, we maintain two vectors $\boldsymbol{\lambda}_r = (\lambda_{1,r}, ..., \lambda_{d,r})$ and $\mathbf{k}_r = (k_{1,r}, ..., k_{d,r})$ as the empirical knowledge learnt from the history. More specifically, $\lambda_{i,r}$ is the sample mean of participant i's sensing value at the end of the r-th time slot and $k_{i,r}$ is the number of time slots that i is selected (sampled) by then. At the initialization stage (lines 1-3), the algorithm selects all participants to acquire the initial information $\boldsymbol{\lambda}_1$ and $\mathbf{k}_1$. Then **Algorithm 2** is invoked for each of the later time slots to select FSEs based on current $\boldsymbol{\lambda}_r$ and $\mathbf{k}_r$ (line 7). Instead of directly using the sample means, we introduce a new vector $\bar{\boldsymbol{\lambda}}_r$ by amending each $\lambda_{i,r}$ with an additive factor (line 6); it serves as improved estimations on the expected sensing values and is used for actually selecting $\bar{\mathbf{v}}_r$.

Algorithm 2 adopts a greedy strategy that selects an FSE with the maximum *Revenue-Cost Ratio* (RCR henceforth) parameterized by $\bar{\boldsymbol{\lambda}}_r$. In other words, **Algorithm 2** should return $\bar{\mathbf{v}}_r = \mathbf{v}^*(\bar{\boldsymbol{\lambda}}_r)$ so that $\theta_r^* = f(\mathbf{v}^*(\bar{\boldsymbol{\lambda}}_r), \bar{\boldsymbol{\lambda}}_r)/p\left(\mathbf{v}^*(\bar{\boldsymbol{\lambda}}_r)\right)$ is maximized. This is essentially a fractional programming [20] problem and we solve it optimally in polynomial time by **Algorithm 2** based on a *parametric sorting* method. More detailed analysis of **Algorithm 2** is given in Sec IV-A.

After identifying $\bar{\mathbf{v}}_r$ (lines 8–11), **Algorithm 1** checks if its cost is bigger than the current leftover budget. If so, the algorithm stops and returns the FSEs selected so far as well as the number of time slots during which the crowdsensing task has been performed. Otherwise, it employs the participants indicated by $\boldsymbol{\phi}_r = \bar{\mathbf{v}}_r$ to perform sensing for the r-th time slot and subtracts $p(\boldsymbol{\phi}_r)$ from the current budget. The sensing values learnt during this time slot are then used to update the empirical knowledge.

IV. Performance Analysis

In this section, we provide theoretical performance analysis for the BLISS online learning algorithm proposed in Sec. III. We will first prove the optimality of **Algorithm 2** in Sec. IV-A, and then prove the regret bound of **Algorithm 1** in Sec. IV-B.

A. Optimality of the FSE Selection

As we mentioned before, the objective of **Algorithm 2** is to find an FSE $\bar{\mathbf{v}}_r$ that maximizes RCR based on the estimated sensing value vector $\bar{\boldsymbol{\lambda}}_r$. A critical building block of **Algorithm 2** is the function Max, which is a parametric comparison function called by the sorting process in **Algorithm 2**. To prove the optimality of $\bar{\mathbf{v}}_r$, we first reveal two important features of the function Max, as shown by ***Lemma 1***:

Lemma 1: Suppose that the function $\mathsf{Max}(a, b, i', j', \bar{\boldsymbol{\lambda}}_r, m, \boldsymbol{\omega}, d, \mathbf{p})$ returns (a', b', z'). If $\theta_r^* \in [a, b]$. Then we must have

i) $\theta_r^* \in [a', b']$.
ii) $\forall \mu \in [a', b'] : z' = \arg\max_{t \in \{i,j\}} (\omega_t \bar{\lambda}_{t,r} - \mu \cdot p_t)$

Clearly $0 \le \theta_r^* \le \sum_{i=1}^{d} \omega_i \bar{\lambda}_{i,r} / p_{min}$, hence θ_r^* is guaranteed to be in $[a, b]$ after line 1 of **Algorithm 2** is executed. As the parametric (bubble) sorting process in lines 3–6 of

Algorithm 2 repetitively calls the function Max to compare participants while at the same time to shrink $[a, b]$, we know that θ_r^* is never excluded from $[a, b]$ due to i) of ***Lemma** 1*. Moreover, after the sorting is completed, the participants must be in a decreasing order that satisfies the following property due to ii) of ***Lemma** 1*:

$$\forall \mu \in [a, b] : \omega_i \bar{\lambda}_{i,r} - \omega_j \bar{\lambda}_{j,r} \geq \mu \cdot (p_i - p_j), \tag{7}$$

where (i, j) is any pair of participants such that i is "greater" than j in the sorting result. Based on these, lines 7–10 of **Algorithm 2** then finds the optimal $\bar{\mathbf{v}}_r$, and the correctness of lines 7–10 is proved by ***Theorem** 2*:

***Theorem** 2:* $f(\bar{\mathbf{v}}_r, \bar{\boldsymbol{\lambda}}_r)/p(\bar{\mathbf{v}}_r) = \theta_r^*$.

We also show that **Algorithm 2** is a polynomial-time algorithm.

***Theorem** 3:* The average time complexity of **Algorithm 2** is $\mathcal{O}(d^2 \log^2 d)$.

B. Regret Bound for BLISS

Now we are ready to prove the regret bound of **Algorithm 1**. The overall idea of the proofs is the following: we show that the expected participant sensing values learned by **Algorithm 1** do not deviate much from the real values, so we do not suffer a big loss by using them for selecting FSEs, compared with using the real sensing value expectations for selection. Note that the number of time slots during which **Algorithm 1** runs (i.e., N) is a random variable, hence most of our proofs are based on conditional probabilities with respect to N.

Let $q(r, s) = \sqrt{\frac{5 \ln r}{2s}}$ for any $r, s > 0$. Using the Chernoff-Hoeffding bound [21], we can show that if a participant i is selected for a sufficient number of times in the history (i.e., $k_{i,r-1}$ is sufficiently large), then the sample mean of i's sensing value $\lambda_{i,r-1}$ will be close to the real expected sensing value τ_i with high probability. In other words,

$$\mathbf{Pr}\left\{|\lambda_{i,r-1} - \tau_i| > q(r, k_{i,r-1})\big|N\right\} \leq 2r^{-4} \tag{8}$$

Let $\eta^* = \max_{\mathbf{v} \in \mathcal{V}} \frac{f(\mathbf{v}, \boldsymbol{\tau})}{p(\mathbf{v})}$ and $\mathcal{A} = \left\{\mathbf{v} \in \mathcal{V} \big| \frac{f(\mathbf{v}, \boldsymbol{\tau})}{p(\mathbf{v})} < \eta^*\right\}$, i.e., $\mathcal{A}$ is the set of FSEs with sub-optimal RCRs computed by the real expected participant sensing values. If $\mathcal{A} \neq \emptyset$, then let γ and ξ be the smallest and largest discrepancy between η^* and the RCR of any FSE in $\mathcal{A}$, i.e., $\gamma = \min\{\eta^* - f(\mathbf{v}, \boldsymbol{\tau})/p(\mathbf{v}) | \mathbf{v} \in \mathcal{A}\}$ and $\xi = \max\{\eta^* - f(\mathbf{v}, \boldsymbol{\tau})/p(\mathbf{v}) | \mathbf{v} \in \mathcal{A}\}$. Let $\beta = \max\{\sum_{i \in \mathcal{H}(\mathbf{v})} \omega_i / p(\mathbf{v}) | \mathbf{v} \in \mathcal{V}\}$. Let the event $\mathcal{E}_r = \left\{\forall i \in \mathcal{H}(\boldsymbol{\phi}_r) : k_{i,r-1} > \frac{10\beta^2}{\gamma^2} \ln r\right\}$. Based on (8) and ***Theorem** 2*, the following lemma reveals that, if all the selected participants for the r-th time slots have been selected in the past for a sufficient number of times (greater than $\frac{10\beta^2}{\gamma^2} \ln r$), then with high probability the FSE chosen for the r-th time slot would maximize the RCR with respect to the real expected sensing values.

***Lemma** 2:* For any $1 < r \leq N$, we have $\mathbf{Pr}\{\mathcal{E}_r \wedge (\boldsymbol{\phi}_r \in \mathcal{A}) | N\} \leq 2dr^{-4}$

Using ***Lemma** 2*, we can bound the total number of non-optimal FSEs that have been chosen; these FSEs do not maximize the RCR with respected to the real sensing value expectations, and are selected due to the deviations of our empirically learned sensing values. Fortunately, ***Lemma** 3* shows that the expected number of such FSEs is no more than $\mathcal{O}(d \log N)$.

***Lemma** 3:* Let $\mathcal{D}_N = \{\boldsymbol{\phi}_r | r \in [\![N]\!] \wedge \boldsymbol{\phi}_r \in \mathcal{A}\}$. We have

$$\mathbb{E}\left\{|\mathcal{D}_N| \big| N\right\} \leq 1 + \frac{10\beta^2}{\gamma^2} d \ln N + \frac{d\pi^4}{45}$$

Bounding the number of sub-optimal FSEs chosen by **Algorithm 1** through ***Lemma** 3* allows us to further bound the total regret of choosing these FSEs. Moreover, repetitively choosing optimal FSEs is an approximation policy to BLISS-ILP, whose regret can also be derived. Based on these ideas, we prove the regret bound of **Algorithm 1** in ***Theorem** 4*:

***Theorem** 4:* The regret of **Algorithm 1** is no more than $p_{max}\left(\eta^* + \xi + \frac{10\beta^2 \xi d}{\gamma^2} \ln\left(\frac{G}{p_{min}}\right) + \frac{\xi d \pi^4}{45}\right) = \mathcal{O}(d \log G)$.

As a special case, note that if $\forall \mathbf{v} \in \mathcal{V} : \eta^* = f(\mathbf{v}, \boldsymbol{\tau})/c(\mathbf{v})$, then we have $\mathcal{A} = \emptyset$ and $\mathcal{D}_N = \emptyset$, hence the regret bound shown in ***Theorem** 4* would be a constant $p_{max}\eta^*$.

The asymptotical optimality of this regret bound is further proven as follows.

***Theorem** 5:* Any algorithm for BLISS has a regret of at least $\Omega(d \log G)$.

Note that when $G \to \infty$, the average regret of **Algorithm 1** per time slot goes to 0. This implies that **Algorithm 1** is Hannan consistent [18].

V. Simulations

In this section we evaluate the performance of our online learning algorithm through extensive simulations. The simulations focus on the effect of various crowdsensing conditions on the performance of sensing policies generated by **Algorithm 1** (denoted by BLISS in the simulations) and other related algorithms. To the best of our knowledge, the closest algorithm that can be adapted to our scenario is LLR proposed by [22]: it solves network optimization problems (e.g., maximum weighted matching) under a stochastic MAB model. However, as the costs for pulling arms are neglected in LLR, its policy behaves in a "myopic" way in our scenario by selecting all the participants in every time slot to maximize the short-term sensing revenue. We also implement a straightforward policy, RANDOM, that randomly selects an FSE in each time slot.

In all our simulations, the weight of any participant i (i.e. ω_i) is randomly generated from the uniform distribution $U[0.1, 1.1]$, and p_i is generated by the same method. The sensing value $X_{i,r}, \forall i, r \in \mathbb{Z}^+$ of participant i in any time slot is randomly sampled from two candidate distributions: the first one is the truncated Gaussian distribution with mean τ_i, standard deviation $\frac{\tau_i}{2}$, and support $[0, 2\tau_i]$, and the second one is the uniform distribution with support $[0, 2\tau_i]$.

A. On Regret

We first compare the regrets of different algorithms. Unlike the other MAB algorithms such as [19], a major difficulty for us to evaluate the regret of BLISS is the NP-hardness of computing the optimal solution (see ***Theorem** 1*). Therefore, we only compare the regrets of BLISS, LLR and RANDOM under a small case where $d = 6$, $m = 3$, and the budget G scales from 10 to 300 with an increment of 10. For any participant i, the expected sensing value τ_i is generated randomly such that the support of $X_{i,r}, \forall r \in \mathbb{Z}^+$ belongs to $[0, 1]$. In such a small case, BLISS-ILP can be solved optimally using an ILP solver (e.g., CPLEX [23]) in reasonable time, and we can compute the optimal solution and hence the regrets of all algorithms. The results are shown in Fig. 1, where each algorithm's regret is normalized with respect to the logarithm of n–the number of time slots during which the algorithm performs crowdsensing.

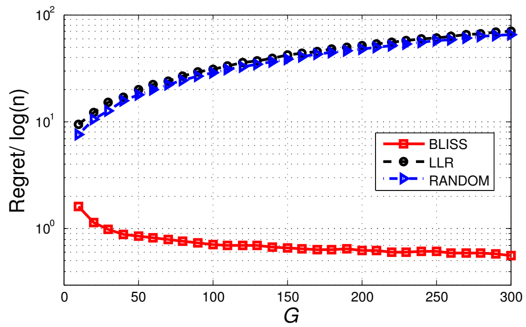


Fig. 1. Regret Performance of BLISS and other algorithms

Obviously, the regret of BLISS is much lower than those of LLR and RANDOM (note the logarithmic scale of the y-axis). Actually, the normalized regrets of both LLR and RANDOM grow linearly with respect to G, whereas that of BLISS levels off to a constant. Since $n = \Theta(G)$ (as $\frac{G}{p_{max}} \le n \le \frac{G}{p_{min}}$), the results in Fig. 1 also strongly corroborate the theoretical regret bound proved in Sec. IV-B.

B. On Sensing Revenue

We then study the performance of different algorithms in terms of the total sensing revenue, which is the actual benefit the owner gains in practice. The results are shown in Fig. 2 and Fig. 3, where τ_i is randomly generated from the uniform distribution $U[500, 1500]$ for any participant i. In Fig. 2, all participants' sensing values are sampled from Gaussian distributions, whereas they are sampled from both Gaussian and uniform distributions with equal chance in Fig. 3. All the figures show the statistical summaries (i.e., means and standard deviations) of 100 simulation results.

We study the impact of budget on the sensing revenue in Fig. 2(a) and 3(a), where we set $d = 100$, $m = 40$ and scale the budget G from 1000 to 10000 with an increment of 1000. The sensing revenue of all the algorithms increasing with the budget can be easily understood: more participants can be employed for sensing under a larger budget.

In Fig. 2(b) and 3(b), we set $G = 10000$, $m = 40$ and scale d from 100 to 1000 with an increment of 100. In this case, the revenue of BLISS exhibits an uptrend with the increasing of d, whereas those of LLR and RANDOM do not change much. This can be explained by the reason that BLISS intelligently selects participants based on their sensing values and costs, so a larger group of participants brings a larger space for selection and hence a higher revenue. On the contrary, LLR and RANDOM either myopically or blindly select participants, which makes their revenue insensitive to the enlargement of participant groups.

In Fig. 2(c) and 3(c), we study the relation between the sensing revenue and m by setting $G = 10000$, $d = 200$ and scale m from 10 to 100 with a step of 10. The results show that LLR and RANDOM give similar revenue under all the values of m, while the revenue obtained by BLISS drops with the increment of m. The reasons for this phenomenon is that a larger m results in a smaller selection space for BLISS. Actually, the revenue got by all algorithms would be similar when m is very close to d, because all of them have to select all the participants at each time slot under the extreme case where $m = d$.

We can further make the following observations by comparing the three algorithms and contrasting Fig. 2 and 3.

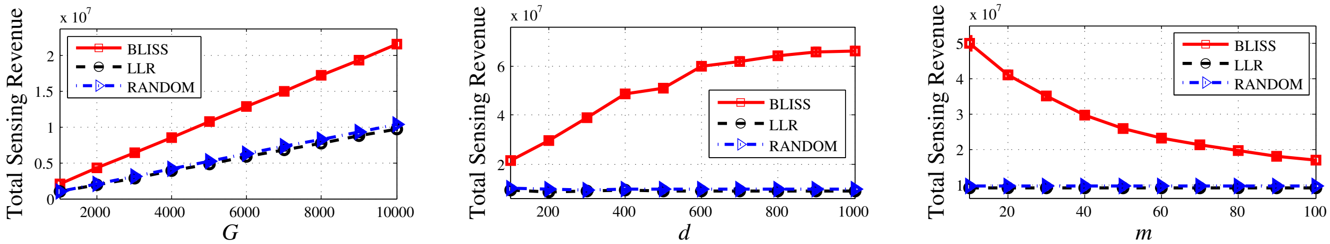
- The sensing revenue obtained by BLISS is significantly larger than those obtained by LLR and RANDOM under all the cases, demonstrating the superiority of BLISS under various crowdsensing conditions.
- BLISS is insensitive to the distributions of the sensing values: it outperforms other algorithms without affected by the specific sensing value distributions that the participants follows.

VI. RELATED WORK

Recently, there has been a substantial growth on designing crowdsensing systems for various applications, such as (vehicle) traffic monitoring/prediction [13], localization [11], parking space allocation/searching [14], and ambient (e.g.,dust level) surveillance [15]. At the same time, theoretical investigations on managing crowdsensing has also been conducted, but most of them concentrate on the incentive problems of crowdsensing [3]–[6]. Although certain data-quality related problems for crowdsensing have been raised in [8]–[10], [12], none of them has considered the problem of handling data-quality uncertainty by intelligently recruiting the participants, as we have done in this paper.

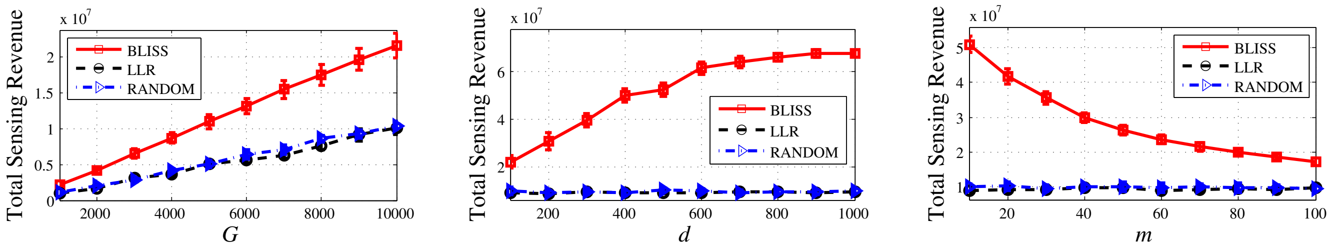
The study on stochastic MAB problems is pioneered by Lai *et.al.* [24] and Auer *et.al.* [19], who provide algorithms with regret bounds growing logarithmically with respect to the number of arm-pullings. Following them, extensive proposals on MAB problems have been proposed, and an excellent survey can be found in [25]. However, all these proposals assume that playing arms is free, and the problem of considering arm-playing costs for MAB only starts to attract attention very recently [26].

We note that all the proposals in [24], [19] and [26] adopt the classical MAB model, i.e., exactly one single arm can be played at each step. This playing rule is revised by a recent work [22], where multiple arms with certain combinatorial structures can be played at the same time. The authors of [22] indicate that traditional MAB algorithms that play one arm



(a) Scaling G (b) Scaling d (c) Scaling m

Fig. 2. Performance comparisons when the participants' random sensing values are drawn from homogeneous distributions.



(a) Scaling G (b) Scaling d (c) Scaling m

Fig. 3. Performance comparisons when the participants' random sensing values are drawn from heterogeneous distributions.

at each step (e.g., [19]) perform poorly in such a scenario and propose new algorithms with provable regret bounds. However, since [22] still assumes that playing arms is free (the same as [24] [19]), the arms can be played perpetually to acquire sufficient knowledge about them. Consequently, none of the existing work in [19], [22], [24], [26] fits the BLISS problem studied in this paper. Actually, to the best of our knowledge, we are the first to consider a stochastic MAB model where multiple arms (participants) with costs can be played simultaneously under a combinatorial structure and the regret should be minimized subject to a budget limit.

VII. Conclusion

We have raised and considered a novel but practical robust crowdsensing problem, where the quality of sensing data acquired by the participants are uncertain and a crowdsensing task owner aims to maximize its expected total sensing revenue under a limited budget for compensating the sensing costs. To efficiently tackle such a problem, we have formulated the Budget LImited robuSt crowdSensing (BLISS) as a high-level general framework to characterize it, and we have further proposed an online learning algorithm whose regret bound is proven to be asymptotically optimal. We also have conducted extensive simulations and the simulation results have strongly demonstrated the effectiveness of our approach.

Acknowledgment

This work was supported in part by the AcRF Tier 1 Grant RGC5/13 and the National Natural Science Foundation of China (No. 61103007).

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile Crowdsensing: Current State and Future Challenges," *IEEE Communications Magazine*, vol. 6, no. 11, pp. 32–39, 2011.

[2] W. Khan, Y. Xiang, M. Aalsalem, and Q. Arshad, "Mobile Phone Sensing Systems: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 402 –427, 2013.

[3] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive Mechanisms for Smartphone Collaboration in Data Acquisition and Distributed Computing," in *Proc. IEEE INFOCOM*, 2012, pp. 1701–1709.

[4] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing," in *Proc. ACM MobiCom*, 2012, pp. 173–184.

[5] I. Koutsopoulos, "Optimal Incentive-driven Design of Participatory Sensing Systems," in *Proc. IEEE INFOCOM*, 2013.

[6] Q. Li and G. Cao, "Providing Privacy-Aware Incentives for Mobile Sensing," in *Proc. IEEE PerCom*, 2013, pp. 76–84.

[7] K. Han, C. Zhang, and J. Luo, "Truthful scheduling mechanisms for powering mobile crowdsensing," 2013. [Online]. Available: http://arxiv.org/abs/1308.4501

[8] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On Truth Discovery in Social Sensing: a Maximum Likelihood Estimation Approach," in *Proc. ACM/IEEE IPSN*, 2012, pp. 233–244.

[9] H. Min, P. Scheuermann, and J. Heo, "A Hybrid Approach for Improving the Data Quality of Mobile Phone Sensing," *International Journal of Distributed Sensor Networks*, 2013, DOI:10.1155/2013/786594.

[10] S. Reddy, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "A Framework for Data Quality and Feedback in Participatory Sensing," in *Proc. ACM SenSys*, 2007, pp. 417–418.

[11] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically Characterizing Places with Opportunistic Crowdsensing Using Smartphones," in *Proc. ACM UbiComp*, 2012, pp. 481–490.

[12] C.-K. Tham and T. Luo, "Quality of Contributed Service and Market Equilibrium for Participatory Sensing," in *Proc. IEEE DCOSS*, 2013.

[13] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "GreenGPS: A Participatory Sensing Fuel-Efficient Maps Application," in *Proc. ACM MobiSys*, 2010, pp. 151–164.

[14] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrashekharan, W. Xue, M. Gruteser, and W. Trappe, "ParkNet: Drive-by Sensing of Road-Side Parking Statistics," in *Proc. ACM MobiSys*, 2010, pp. 123–136.

[15] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song, "CoMon: Cooperative Ambience Monitoring Platform with Continuity and Benefit Awareness," in *Proc. ACM MobiSys*, 2012, pp. 43–56.

[16] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity Recognition Using Cell Phone Accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[17] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment Framework for Participatory Sensing Data Collections," in *Proc. Pervasive*, 2010, pp. 138–155.

[18] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[19] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-Time Analysis of the Multiarmed Bandit Problem," *Machine Learning*, vol. 47, no. 2–3, pp. 235–256, 2002.

[20] N. Hadjisavvas, S. Komlósi, and S. S. Schaible, *Handbook of Generalized Convexity and Generalized Monotonicity*. Springer Verlag, 2005.

[21] W. Hoeffding, "Probability Inequalities for Sums of Bounded Random Variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 13–30, 1963.

[22] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial Network Optimization with Unknown Variables: Multi-Armed Bandits with Linear Rewards and Individual Observations," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1466–1478, 2012.

[23] "IBM ILOG CPLEX Optimizer." [Online]. Available: http://www.ibm.com/software/integration/optimization/cplex-optimizer/

[24] T. L. Lai and H. Robbins, "Asymptotically Efficient Adaptive Allocation Rules," *Advances in Applied Mathematics*, vol. 6, no. 1, pp. 4–22, 1985.

[25] S. Bubeck and N. Cesa-Bianchi, "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[26] L. Tran-Thanh, A. C. Chapman, A. Rogers, and N. R. Jennings, "Knapsack Based Optimal Policies for Budget-Limited Multi-Armed Bandits," in *Proc. AAAI*, 2012, pp. 1134–1140.

[27] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[28] F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, *NIST Handbook of Mathematical Functions*. Cambridge University Press, 2010.

APPENDIX

***Proof of Theorem** 1:* We prove the NP-hardness of BLISS-ILP by a reduction from the NP-complete Partition problem [27]. Given a set of ℓ positive integers $S = \{s_1, s_2, \cdots, s_\ell\}$, the Partition problem is to decide whether S can be partitioned into two subsets such that the sum of the numbers in one subset equals that in another. Suppose that there are $d = 2\ell$ participants in BLISS-ILP, and let $m = 1$. Let $\omega_i = 1$, $p_i = \tau_i = (2^{\ell+1} + 2^i)\ell \sum_{j=1}^{\ell} s_j$ and $p_{\ell+i} = \tau_{\ell+i} = \tau_i + s_i$ for $\forall i \in [\![\ell]\!]$. Let the budget $G = \sum_{i=1}^{\ell} s_i \left[\ell^2 2^{\ell+1} + \ell \sum_{i=1}^{\ell} 2^i + 2^{-1}\right]$. The decision version of this special BLISS-ILP asks if there exists a solution such that the value of the objective function (2) is no less than G. It can be verified that this decision problem is equivalent to the Partition problem on the set S (the detailed verification is omitted due to page limit). Hence the theorem follows. □

***Proof of Lemma** 1:* We first prove i). Note that only lines 19-20 of **Algorithm 2** can make $[a', b'] \neq [a, b]$. If line 19 is executed, then there must exist $\hat{\mathbf{v}} \in \mathcal{V}$ such that

$$\sum\nolimits_{t \in \mathcal{H}(\hat{\mathbf{v}})} (\omega_t \bar{\lambda}_{t,r} - l \cdot p_t) = \sum\nolimits_{s \in S} s > 0$$

If $\theta_r^* \leq l$ in this case, then we must have

$$0 < \sum_{t \in \mathcal{H}(\hat{\mathbf{v}})} (\omega_t \bar{\lambda}_{t,r} - l \cdot p_t) \leq \sum_{t \in \mathcal{H}(\hat{\mathbf{v}})} (\omega_t \bar{\lambda}_{t,r} - \theta_r^* \cdot p_t)$$

which yields $f(\hat{\mathbf{v}}, \bar{\boldsymbol{\lambda}}_r)/p(\hat{\mathbf{v}}) > \theta_r^*$; a contradiction. Hence $\theta_r^* \in [l, b] = [a', b']$. Similarly, we can prove that the execution of line 20 also guarantees $\theta_r^* \in [a', b'] = [a, l]$. This completes the proof for i).

Now we prove ii). For simplicity, we assume $p_{i'} \geq p_{j'}$ and the case of $p_{i'} < p_{j'}$ can be proved by symmetry. Due to line 13 of **Algorithm 2**, we have $i = i'$, $j = j'$, and $p_i \geq p_j$. When $b(p_i - p_j) \leq \omega_i \bar{\lambda}_{i,r} - \omega_j \bar{\lambda}_{j,r}$, we get $z' = i$ according to line 15. Actually, in this case for any $\mu \in [a', b'] = [a, b]$ we have

$$(\omega_i \bar{\lambda}_{i,r} - \mu \cdot p_i) - (\omega_j \bar{\lambda}_{j,r} - \mu \cdot p_j) \geq (b - \mu)(p_i - p_j) \geq 0$$

Hence ii) holds. Similarly, we can also prove that ii) is true for $\omega_i \bar{\lambda}_{i,r} - \omega_j \bar{\lambda}_{j,r} \leq a(p_i - p_j)$ (line 16). If neither line 15 nor line 16 is executed, we must have $l \in [a, b]$ according to line 17. In this case, if line 19 is executed and $[a', b', z'] = [l, b, j]$, we must have

$$\begin{aligned} \forall \mu \in [l, b]: \quad & (\omega_i \bar{\lambda}_{i,r} - \mu \cdot p_i) - (\omega_j \bar{\lambda}_{j,r} - \mu \cdot p_j) \\ \leq \quad & (\omega_i \bar{\lambda}_{i,r} - \omega_j \bar{\lambda}_{j,r}) - l(p_i - p_j) = 0 \end{aligned}$$

hence ii) still holds. Similarly, the execution of line 20 allows ii) to hold for $[a', b', z'] = [a, l, i]$. □

***Proof of Theorem** 2:* According to ***Lemma** 1*, after completing the parametric sorting process, we have $\theta_r^* \in [a, b]$. Moreover, for $1 \leq i < j \leq d$ and any $\mu \in [a, b]$, we have

$$\omega_{h[i]} \bar{\lambda}_{h[i],r} - \mu \cdot p_{h[i]} \geq \omega_j \bar{\lambda}_{h[j],r} - \mu \cdot p_{h[j]} \tag{9}$$

Let $g(i, \mu) = \sum_{t=1}^{i} (\omega_{h[t]} \bar{\lambda}_{h[t],r} - \mu \cdot p_{h[t]})$. Using (9) we know that, for any $\mu \in [a, b]$:

$$\begin{aligned} & \max\nolimits_{\mathbf{v} \in \mathcal{V}} \left\{ \sum\nolimits_{i \in \mathcal{H}(\mathbf{v})} \omega_i \bar{\lambda}_{i,r} - \mu \sum\nolimits_{i \in \mathcal{H}(\mathbf{v})} p_i \right\} \\ = \quad & \max\{g(i, \mu) | m \leq i \leq d\} \end{aligned} \tag{10}$$

Clearly, equation (10) equals 0 when $\mu = \theta_r^*$. On the other hand, if equation (10) equals 0 and $\mu < \theta_r^*$, then we have

$$\sum_{i \in \mathcal{H}(\mathbf{v}^*(\bar{\boldsymbol{\lambda}}_r))} (\omega_i \bar{\lambda}_{i,r} - \theta_r^* p_i) < \sum_{i \in \mathcal{H}(\mathbf{v}^*(\bar{\boldsymbol{\lambda}}_r))} (\omega_i \bar{\lambda}_{i,r} - \mu \cdot p_i) \leq 0$$

and hence $\theta_r^* > f(\mathbf{v}^*(\bar{\boldsymbol{\lambda}}_r), \bar{\boldsymbol{\lambda}}_r)/p(\mathbf{v}^*(\bar{\boldsymbol{\lambda}}_r))$, a contradiction. Similarly we can prove that $\mu > \theta_r^*$ does not hold if equation (10) equals 0. In other words, equation (10) is equal to 0 iff $\mu = \theta_r^*$. Now the theorem follows from lines 7-10. □

***Proof of Theorem** 3:* The dominant running time of the function Max is spent on line 18, done by a quick sort in $\mathcal{O}(d \log d)$ time. Therefore, the loop in lines 3-6 runs in $\mathcal{O}(d^3 \log^2 d)$ time. The time spent on lines 7-10 is $\mathcal{O}(d^2)$. So the overall time complexity of **Algorithm 2** is $\mathcal{O}(d^3 \log^2 d)$. However, if we replace the bubble sorting framework (used to simplify our presentation) in lines 3-6 by a quick sorting framework, then **Algorithm 2** can be implemented in $\mathcal{O}(d^2 \log^2 d)$ time on average. □

***Proof of Lemma** 2:* Suppose that $\mathcal{E}_r \wedge (\phi_r \in \mathcal{A})$ holds. Let $\Gamma_{max} = \max\{\bar{\lambda}_{i,r} - \tau_i | i \in \mathcal{H}(\phi_r)\}$. Using ***Theorem** 2* we get $f(\boldsymbol{\phi}_r, \bar{\boldsymbol{\lambda}}_r)/p(\boldsymbol{\phi}_r) \geq f(\mathbf{v}^*(\boldsymbol{\tau}), \bar{\boldsymbol{\lambda}}_r)/p(\mathbf{v}^*(\boldsymbol{\tau}))$. Besides,

$$\frac{f(\boldsymbol{\phi}_r, \bar{\boldsymbol{\lambda}}_r)}{p(\boldsymbol{\phi}_r)} - \frac{f(\boldsymbol{\phi}_r, \boldsymbol{\tau})}{p(\boldsymbol{\phi}_r)} \leq \frac{\sum_{i\in\mathcal{H}(\boldsymbol{\phi}_r)} \omega_i}{p(\boldsymbol{\phi}_r)} \cdot \Gamma_{max} \leq \beta \cdot \Gamma_{max}$$

So we have

$$f(\mathbf{v}^*(\boldsymbol{\tau}), \bar{\boldsymbol{\lambda}}_r)/p(\mathbf{v}^*(\boldsymbol{\tau})) - f(\boldsymbol{\phi}_r, \boldsymbol{\tau})/p(\boldsymbol{\phi}_r) \leq \beta \cdot \Gamma_{max} \quad (11)$$

Note that when $\boldsymbol{\phi}_r \in \mathcal{A}$, we must have

$$f(\mathbf{v}^*(\boldsymbol{\tau}), \boldsymbol{\tau})/p(\mathbf{v}^*(\boldsymbol{\tau})) - f(\boldsymbol{\phi}_r, \boldsymbol{\tau})/p(\boldsymbol{\phi}_r) \geq \gamma \quad (12)$$

Combining (11) and (12) yields

$$\frac{f(\mathbf{v}^*(\boldsymbol{\tau}), \bar{\boldsymbol{\lambda}}_r)}{p(\mathbf{v}^*(\boldsymbol{\tau}))} - \frac{f(\mathbf{v}^*(\boldsymbol{\tau}), \boldsymbol{\tau})}{p(\mathbf{v}^*(\boldsymbol{\tau}))} \leq \beta \cdot \Gamma_{max} - \gamma \quad (13)$$

Case 1: $\forall i \in [\![d]\!] : \bar{\lambda}_{i,r} - \tau_i \geq 0$;

In this case, we must have $\Gamma_{max} \geq \frac{\gamma}{\beta}$ according to equation (13). Since $\forall i \in \mathcal{H}(\boldsymbol{\phi}_r) : k_{i,r-1} > \frac{10\beta^2}{\gamma^2} \ln r$, we have $2q(r, k_{i,r-1}) < \frac{\gamma}{\beta}$ and hence $\exists i \in \mathcal{H}(\boldsymbol{\phi}_r) : \bar{\lambda}_{i,r} - \tau_i > 2q(r, k_{i,r-1})$. As $\bar{\lambda}_{i,r} = \lambda_{i,r-1} + q(r, k_{i,r-1})$, we get $\exists i \in [\![d]\!] : \lambda_{i,r-1} - \tau_i > q(r, k_{i,r-1})$.

Case 2: $\exists i \in [\![d]\!] : \bar{\lambda}_{i,r} - \tau_i < 0$;

In this case, we have $\exists i \in [\![d]\!] : \lambda_{i,r-1} - \tau_i < -q(r, k_{i,r-1})$. Synthesizing Case 1 and Case 2, we have

$$\begin{aligned} & \mathbf{Pr}\{\mathcal{E}_r \wedge (\phi_r \in \mathcal{A})|N\} \\ \leq\ & \mathbf{Pr}\left\{\exists i \in [\![d]\!] : |\lambda_{i,r-1} - \tau_i| > q(r, k_{i,r-1}) \middle| N\right\} \\ \leq\ & \sum_{i=1}^{d} \mathbf{Pr}\left\{|\lambda_{i,r-1} - \tau_i| > q(r, k_{i,r-1}) \middle| N\right\} \quad (14) \\ \leq\ & 2dr^{-4} \quad (15) \end{aligned}$$

where (14) holds because of the union bound and (15) is due to (8). So the lemma follows. □

Proof of Lemma 3: We define a set of random variables $\{\delta_{i,r} | i \in [\![d]\!], r \in [\![N]\!]\}$ as follows. For any $i \in [\![d]\!]$, let $\delta_{i,1} = 1$. For any $2 \leq r \leq N$, if $\boldsymbol{\phi}_r \in \mathcal{A}$, then we find $j = \arg\min_{i\in\mathcal{H}(\boldsymbol{\phi}_r)} \delta_{i,r-1}$ (breaking ties arbitrarily) and set $\delta_{j,r} = \delta_{j,r-1} + 1$. Any $\delta_{i,r} (2 \leq r \leq N)$ not involved in the aforementioned rule remains the same with $\delta_{i,r-1}$. Let $\varepsilon = \frac{10\beta^2}{\gamma^2} \ln N$. We have

$$\begin{aligned} & \sum_{r=2}^{N} \mathbf{Pr}\{\phi_r \in \mathcal{A}|N\} \\ \leq\ & \sum_{r=2}^{N} \sum_{i=1}^{d} \mathbf{Pr}\{\phi_r \in \mathcal{A} \wedge \delta_{i,r} > \delta_{i,r-1}|N\} \\ \leq\ & d\varepsilon + \sum_{r=2}^{N} \sum_{i=1}^{d} \mathbf{Pr}\{\phi_r \in \mathcal{A} \wedge \delta_{i,r} > \delta_{i,r-1} > \varepsilon|N\} \end{aligned}$$

Notice that if $\delta_{i,r} > \delta_{i,r-1} > \varepsilon$, we must have $i = \arg\min_{j\in\mathcal{H}(\boldsymbol{\phi}_r)} \delta_{j,r-1}$, which implies $\forall j \in \mathcal{H}(\boldsymbol{\phi}_r) : \delta_{j,r-1} > \varepsilon$. Since $\varepsilon \geq \frac{10\beta^2}{\gamma^2} \ln r$ and $\forall i \in [\![d]\!] : k_{i,r-1} \geq \delta_{i,r-1}$, we have

$$\begin{aligned} & \sum_{i=1}^{d} \mathbf{Pr}\{\boldsymbol{\phi}_r \in \mathcal{A} \wedge \delta_{i,r} > \delta_{i,r-1} > \varepsilon|N\} \\ \leq\ & \mathbf{Pr}\left\{\boldsymbol{\phi}_r \in \mathcal{A} \wedge (\forall j \in \mathcal{H}(\boldsymbol{\phi}_r) : \delta_{j,r-1} > \varepsilon) \middle| N\right\} \\ \leq\ & \mathbf{Pr}\left\{\boldsymbol{\phi}_r \in \mathcal{A} \wedge \left(\forall j \in \mathcal{H}(\boldsymbol{\phi}_r) : k_{j,r-1} > \frac{10\beta^2 \ln r}{\gamma^2}\right) \middle| N\right\} \\ =\ & \mathbf{Pr}\{\mathcal{E}_r \wedge (\boldsymbol{\phi}_r \in \mathcal{A})|N\} \leq 2dr^{-4} \end{aligned}$$

Hence

$$\begin{aligned} \mathbb{E}\{|\mathcal{D}_N||N\} &\leq 1 + \sum_{r=2}^{N} \mathbf{Pr}\{\boldsymbol{\phi}_r \in \mathcal{A}|N\} \\ &\leq 1 + \frac{10\beta^2}{\gamma^2} d \ln N + \sum_{r=2}^{N} 2dr^{-4} \\ &\leq 1 + \frac{10\beta^2}{\gamma^2} d \ln N + \frac{d\pi^4}{45} \quad (16) \end{aligned}$$

where (16) holds due to the Riemann zeta function $\sum_{i=1}^{\infty} i^{-4} = \frac{\pi^4}{90}$ [28]. □

Proof of Theorem 4: Let $\overline{\mathcal{D}}_N = \{\boldsymbol{\phi}_r | 1 \leq r \leq N\} - \mathcal{D}_N$. Since $\mathbf{Pr}\left\{\sum_{r=1}^{N} p(\boldsymbol{\phi}_r) > G - p_{max}\right\} = 1$, we have

$$\begin{aligned} & G - p_{max} \leq \mathbb{E}_N \left\{\mathbb{E}\left\{\sum_{r=1}^{N} p(\boldsymbol{\phi}_r) \middle| N\right\}\right\} \\ \leq\ & \mathbb{E}_N \left\{\mathbb{E}\left\{\sum_{\mathbf{v}\in\mathcal{D}_N} p(\mathbf{v}) \middle| N\right\}\right\} \\ & + \mathbb{E}_N \left\{\mathbb{E}\left\{\sum_{\mathbf{v}\in\overline{\mathcal{D}}_N} p(\mathbf{v}) \middle| N\right\}\right\} \quad (17) \end{aligned}$$

Note that for any $\mathbf{v} \in \overline{\mathcal{D}}_N$, we have $\frac{f(\mathbf{v},\boldsymbol{\tau})}{p(\mathbf{v})} = \eta^*$, hence

$$\mathbb{E}\left\{\sum_{\mathbf{v}\in\overline{\mathcal{D}}_N} p(\mathbf{v}) \middle| N\right\} = \frac{1}{\eta^*} \mathbb{E}\left\{\sum_{\mathbf{v}\in\overline{\mathcal{D}}_N} f(\mathbf{v}, \boldsymbol{\tau}) \middle| N\right\} \quad (18)$$

On the other hand,

$$\begin{aligned} R_G(\Phi) &= \mathbb{E}_N \left\{\mathbb{E}\left\{\sum_{\mathbf{v}\in\mathcal{D}_N} f(\mathbf{v}, \boldsymbol{\tau}) \middle| N\right\}\right\} \\ &\quad + \mathbb{E}_N \left\{\mathbb{E}\left\{\sum_{\mathbf{v}\in\overline{\mathcal{D}}_N} f(\mathbf{v}, \boldsymbol{\tau}) \middle| N\right\}\right\} \quad (19) \end{aligned}$$

Combining (17),(18) and (19) we get

$$\begin{aligned} R_G(\Phi) &\geq \mathbb{E}_N \left\{\mathbb{E}\left\{\sum_{\mathbf{v}\in\mathcal{D}_N} (f(\mathbf{v}, \boldsymbol{\tau}) - \eta^* p(\mathbf{v})) \middle| N\right\}\right\} \\ &\quad + \eta^* G - p_{max}\eta^* \quad (20) \end{aligned}$$

Note that $R_G(\Phi^*) \leq G\eta^*$ and

$$\forall \mathbf{v} \in \mathcal{V} : \eta^* p(\mathbf{v}) - f(\mathbf{v}, \boldsymbol{\tau}) \leq \xi p(\mathbf{v}) \quad (21)$$

Therefore, using (20), (21) and ***Lemma 3*** we get

$$\begin{aligned} & R_G(\Phi^*) - R_G(\Phi) \\ \leq\ & p_{max}\eta^* + \xi \cdot \mathbb{E}_N \left\{\mathbb{E}\left\{\sum_{\mathbf{v}\in\mathcal{D}_N} p(\mathbf{v}) \middle| N\right\}\right\} \\ \leq\ & p_{max}\eta^* + \xi \cdot \mathbb{E}_N \left\{p_{max} \cdot \mathbb{E}\{|\mathcal{D}_N||N\}\right\} \\ \leq\ & p_{max}\eta^* + p_{max}\xi \cdot \mathbb{E}_N \left\{1 + \frac{10\beta^2}{\gamma^2} d \ln N + \frac{d\pi^4}{45}\right\} \\ \leq\ & p_{max} \left(\eta^* + \xi + \frac{10\beta^2 \xi d}{\gamma^2} \ln\left(\frac{G}{p_{min}}\right) + \frac{\xi d\pi^4}{45}\right) \quad (22) \end{aligned}$$

where (22) holds because $\mathbf{Pr}\left\{N \leq \frac{G}{p_{min}}\right\} = 1$. Hence the theorem follows. □

Proof of Theorem 5: The proof is omitted due to the lack of space. □