A Distributed Utility-Maximizing Algorithm for Data Collection in Mobile Crowd Sensing

Yang Han, Yanmin Zhu, Jiadi Yu
Department of Computer Science and Engineering, Shanghai Jiao Tong University
hyhx_2008@sjtu.edu.cn, {yzhu, jdyu}@cs.sjtu.edu.cn

***Abstract*—Mobile crowd sensing harnesses the data sensing capability of individual smartphones, underpinning a variety of valuable knowledge discovery, environment monitoring and decision making applications. It is a central issue for a mobile crowd sensing system to maximize the utility of sensing data collection at a given cost of resource consumption at each smartphone. However, it is particularly challenging. On the one hand, the utility of sensing data from a smartphone is usually dependent on its context which is random and varies over time. On the other hand, because of the marginal effect, the sensing decision of a smartphone is also dependent on decisions of other smartphones. Little work has explored the utility maximization problem of sensing data collection. This paper proposes a distributed algorithm for maximizing the utility of sensing data collection when the smartphone cost is constrained. The design of the algorithm is inspired by stochastic network optimization technique and distributed correlated scheduling. It does not require any priori knowledge of smartphone contexts in the future, and hence sensing decisions can be made by individual smartphone. Rigorous theoretical analysis show that the proposed algorithm can achieve a time average utility that is within $O(1/V)$ of the theoretical optimum.**

***Index Terms*—Mobile crowd sensing, utility maximization, smartphone, online algorithm, distributed algorithm, cost constraint.**

I. Introduction

Over the past decades, mobile phones have become an indispensable part of the daily life of almost everyone. Most of smartphones embed a rich set of built-in sensors, such as accelerometer, gyroscope, microphone, GPS, and camera [1]. As a consequence, it is unprecedentedly easier for one to collect sensing information around surroundings and share such sensing information. As a new compelling paradigm for large-scale sensing data collection and sharing, *mobile crowd sensing* [2] harnesses the data collection capability of individual smartphones, underpinning a variety of valuable knowledge discovery, environment monitoring and decision making applications. A number of exciting applications based on mobile crowd sensing have been explored, e.g., noise mapping [3], and personal environmental impact analysis [4].

There are two types of mobile crowd sensing, depending on the way of node participation, i.e., *participatory sensing* and *opportunistic sensing* [1][5]. Participatory sensing requires participants to actively engage in sensing activities by manually determining how, when, what and where to sense. In opportunistic sensing, however, sensing activities are typically automated, without requiring user intervention to actively and consciously perform sensing tasks. In practice, opportunistic sensing applications may run in the background and the phone users may not be aware of active execution of sensing applications. In other words, opportunistic applications are usually transparent to phone users. The benefit of opportunistic sensing is that it significantly lowers the burden of phone users, allowing higher participation, which is crucial for wide adoption of mobile crowd sensing.

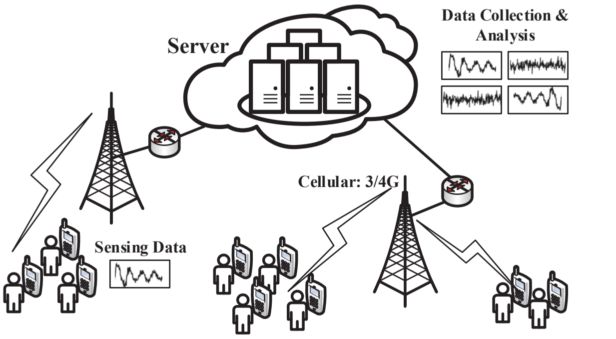

Fig. 1. An illustration of mobile crowd sensing systems. Smartphones perform sensing tasks and then reports sensing data to the data collection server via cellular networks.

This paper concentrates on opportunistic sensing based mobile crowd sensing. As illustrated in Fig. 1, a mobile crowd sensing system consists of a central data collection server and a number of smartphones. Each smartphone opportunistically collects sensing data around its vicinity and reports the sensing data to the central collection server, which then applies data analytics algorithms for monitoring or decision making purposes. The objectives of such a mobile crowd sensing system include larger sensing data volume, higher data quality, and lower cost incurred at smartphones for sensing data collection. We do not consider strategic behaviors of smartphone users and assume that smartphones are cooperative in participating sensing data collection. Such mobile crowd sensing systems are practical in the real world, e.g., when smartphones are volunteers or members of the same organization. Mobile crowd sensing systems with strategic smartphones are beyond the scope of this work.

It is a central issue for a mobile crowd sensing system to gather high quality sensing data with low resource consumption at smartphones. We observe that the utility of sensing data collected by a smartphone may be dependent on the phone context under which it collects the data [1]. The phone context typically varies over time and can be random in nature.

In a large noise detection and monitoring application, for example, the utility of acoustic sensing data is larger when the smartphone is out of the pocket. In a road traffic monitoring application that is time-sensitive, for another example, the utility of sensed road traffic condition is smaller when the smartphone has a poor network connection as it incurs long delay. In the meanwhile, we should emphasize that it costs a smartphone non-negligible resources (e.g., energy, CPU, and bandwidth) as it performs sensing and reporting sensing data to the system. A smartphone is driven by a battery, and the computing power is typically limited. As a result, it is important for smartphones to decide at appropriate time for better data collection at lower cost. More importantly, a mobile crowd sensing system can gather sensing data from many smartphones. It is easy to understand that there is redundancy with sensing data from different smartphones which leads to the *marginal effect* [6]. Therefore, a smartphone decision of data sensing and reporting should also take decisions of other smartphones into account.

There are several great challenges for the mobile crowd sensing system to maximize the utility of sensing data collection at a given cost of resource consumption at each smartphone. *First*, the context of each smartphone is random and varies over time, which is difficult, if not impossible, to predict for future contexts. *Second*, a mobile crowd sensing system may have a large number of smartphones. A centralized solution for deciding the sensing decision for each individual smartphone may introduce prohibitive computational and commutation cost. Moreover, it would introduce the single point of failure problem. *Finally*, because of the marginal effect, the sensing decision of a smartphone is also dependent on decisions of other smartphones.

Mobile crowd sensing has received increasingly extensive research study. Unfortunately, little work has been done on maximizing utility of sensing data collection from smartphones as the cost of smartphones is constrained. In particular, little work has noticed the dependence of sensing data utility on the actual context of a smartphone which is random and changes over time. In addition, most of existing work ignores the marginal effect of sensing data. As a consequence, most existing mobile crowd sensing systems and applications [3][7] blindly make smartphones to collect sensing data, either periodically or randomly.

In this paper we propose a distributed algorithm for maximizing the utility of sensing data collection in a mobile crowd sensing system. To tackle the aforementioned challenges, we take advantage of the *stochastic network optimization technique* developed in [8] and the idea of *distributed correlated scheduling* [9] to design a *distributed online scheduling algorithm*. It does not require any priori knowledge of smartphone contexts in the future, and hence sensing decisions can be made by individual smartphones. The algorithm first transforms the satisfaction of cost constraints to the stability of virtual queues. By defining a quadratic Lyapunov function, the algorithm continuously minimizes a *drift-minus-utility* expression to make sensing decisions.

Our major contributions are summarized as follows:

- It is the first attempt, to the best of our knowledge, to explore the crucial problem of utility maximization of sensing data collection in a mobile crowd sensing system when the cost of smartphones is constrained.
- We formulate the cost-constrained utility maximization problem as an online optimization problem in which the sensing action of individual smartphones is the online decision. We propose a distributed algorithm for solving the online optimization problem which allows each smartphone to make its own sensing decisions.
- We perform rigorous theoretical analysis to show that our algorithm can achieve a time average utility that is within $O(1/V)$ of the optimum with tradeoffs on the time required to converge to the cost constraints, for any $V > 0$ and can adapt to the mobility of mobile smartphones very well.

The remainder of this paper is organized as follows. We formulate the system model and define the problem formally in Section II. In Section III, we present the details of our distributed optimal online scheduling algorithm. We evaluate the performance of the algorithm based on simulations in Section IV. Related work is discussed in Section V. Finally, a brief conclusion of this work is given in Section VI.

II. Problem Definition

First, we summarize the key notations in Table I.

TABLE I
KEY NOTATIONS

N	The number of smartphones in the target region
$s_i(t)$	Phone context of the i-th smartphone in time slot t
c_i	Time average cost constraint on the i-th smartphone
$a_i(t)$	Sensing decision for the i-th smartphone in time slot t
$p_i(t)$	Cost of the i-th smartphone in time slot t
$u(t)$	Utility produced for the target region in time slot t
R_i	The trust of the i-th smartphone

We consider a typical opportunistic sensing scenario in which each smartphone automatically performs sensing tasks and reports sensing data to a remote server without user involvement. In large-scale sensing applications, smartphones are usually organized into target regions according to their geographic locations [3][4][7], for efficient data management. A target region is the area around a sensing target. For example, in the noise mapping application Ear-Phone [3], a physical area is divided into small regions with size of $100\text{m} \times 100\text{m}$. Sensing targets are the noise of each region and smartphones in the same region sense the noise of that region together. For another example, in road traffic monitoring applications, sensing targets are the traffic conditions of each road. Then the target region is the road. All smartphones on the road sense the traffic condition of that road. It is easy to understand that there is redundancy with sensing data from different smartphones in the same target region since they all collect sensing data for same sensing target. Since the scheduling problems in different regions are similar, we only need to focus on the scheduling algorithm in one target region which can be easily extended to the others.

Consider that the mobile crowd sensing system operates over discrete time with unit time slots $t \in \{0, 1, 2, ...\}$. There are N smartphones in the target region. Let $s_i(t) \in S$ denote the phone context of the i-th smartphone in time slot t, where S is the set of possible phone context. Suppose $s_i(t)$ is independent and identically distributed over time slots *. As explained in Section I, the phone context is random and can impact the utility of the sensing data. We use a large value of $s_i(t)$ to indicate that the phone context leads to a higher utility of the sensing data. Take the noise map application as an example: The application wants to take a sound sample when the phone is out of the pocket. Then $S = \{0, 1\}$ and $s_i(t)$ can be a binary value that $s_i(t) = 1$ means the phone is out of the pocket and $s_i(t) = 0$ represents the phone is in the pocket.

Suppose that the phone context can be detected automatically by sensors (e.g., accelerometer and gyroscope) [1]. In every slot, each smartphone detects the current phone context automatically and decides whether or not to perform a sensing task and report to the remote server. We use binary variable $a_i(t) \in \{0, 1\}$ to represent the sensing decision for the i-th smartphone in time slot t. Then, $a_i(t) = 1$ if the i-th smartphone performs a sensing task in slot t, and $a_i(t) = 0$ otherwise. Define the vectors $\mathbf{s}(t) = (s_1(t), s_2(t), ..., s_N(t))$ and $\mathbf{a}(\mathrm{t}) = (a_1(t), a_2(t), ..., a_N(t))$. Then, the utility produced by smartphones in the target region in slot t is denoted by $u(t)$:

$$u(t) = \hat{u}(\mathbf{s}(t), \mathbf{a}(t)) = \min[\sum_{i=1}^{N} s_i(t)a_i(t)R_i, U^*] \quad (1)$$

where U^* is a constant and R_i represents how much the system trusts in the i-th smartphone according to its hardware level. Such utility function is a special case of marginal effect and can model the realistic scenario of *information saturation* which means once a certain amount of utility U^* (e.g., 1) is achieved by one or more smartphones on slot t, there is no advantage in having other smartphones perform sensing tasks and report for the target region on that slot. Suppose each sensing task and report incurs one unit of cost (e.g., power and data traffic consumption) at smartphones. Let $p_i(t)$ be the cost of the i-th smartphone on slot t, being 1 if it performs a sensing task and report, and 0 otherwise. Then the cost for smartphone $i \in \{1, 2, ..., N\}$ in slot t is:

$$p_i(t) = a_i(t). \quad (2)$$

Each smartphone can choose not to sense and report in order to save cost. The time average expected utility and cost are denoted by $\overline{u}$ and $\overline{p}_i$:

$$\overline{u} = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[u(\tau)]$$

$$\overline{p}_i = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}[p_i(\tau)]$$

We then define the cost-constrained utility maximization problem as follows:

$$\text{Maximize}: \quad \overline{u} \quad (3)$$

$$\text{s.t.}: \quad \overline{p}_i \leq c_i, \forall i \in \{1, 2, ..., N\} \quad (4)$$

where c_i are a given set of real numbers which specify constraints on time average cost of smartphones.

We see that it is challenging to achieve the maximal time average utility considering the time average cost constraint at each smartphone, since the phone context is random and time-varying which makes it infeasible to precisely calculate optimal solution in an offline manner. And the current decision is coupled with future decision by the constraint. What's more, it is significantly more challenging to solve in a distributed method. The difficulty is that neither smartphone knows the phone context of others in the target region. Thus, a distributed scheduling algorithm may have redundant smartphones to sense and send reports which incurs costs without increasing utility. In the next section, we will provide a distributed online algorithm which is able to make the optimal sensing decision by each smartphone.

III. Online Scheduling Algorithm

The problem (3)-(4) is a stochastic network optimization problem which can be solved by the *Lyapunov optimization technique* [8] in a centralized manner. Such a centralized method requires the remote server as the coordinator to make sensing decisions for all smartphones in each region based on a full knowledge of phone contexts, in every time slot. This method is not scalable when the number of small regions becomes larger, since the server needs to make sensing decisions for every small region. Therefore, in this section, we propose a distributed approach that enables sensing decisions to be made by each smartphone, based on the idea of *distributed correlated scheduling* [9].

A. Distributed Optimal Scheduling Algorithm

In each time slot, each smartphone detects its phone context automatically and decides whether or not to perform a sensing task and report sensing data. Let $\hat{a}_i(s_i) \in \{0, 1\}, s_i \in \mathrm{S}$ denote the pure strategy of the i-th smartphone. And define a vector-valued function $\hat{\mathbf{a}}(\mathbf{s}) = (\hat{a}_1(s_1), \hat{a}_2(s_2), ..., \hat{a}_N(s_N))$ specifying a distributed decision rule where each smartphone i chooses sensing decision a_i as a deterministic function of s_i, that is $a_i = \hat{a}_i(s_i)$. The total number of pure strategy functions $\hat{\mathbf{a}}(\mathbf{s})$ is $\prod_{\mathrm{i}=1}^{N} 2^{|S|}$. Actually, the set of pure strategy functions can be pruned to a smaller set. Intuitively, most of the strategies are not efficient, since they may choose $a_i = 1$

*Context of each smartphone is possibly correlated in each time slot. The assumption is realistic if the size of time slot is appropriate. We will show that our algorithm doesn't require any knowledge of the probabilities and can adapt if they change.

if s_i is small but choose $a_i = 0$ if s_i is large. Therefore, the strategy function of each smartphone i can be restricted to the following threshold form:

$$\hat{a}_i(s_i) = \begin{cases} 0, & \text{if } s_i \le s_i^* \\ 1, & \text{if } s_i > s_i^* \end{cases} \tag{5}$$

for some thresholds $s_i^* \in S$. Since there are $|S|$ such threshold functions for each smartphone i, the number of pure strategy functions $\hat{\mathbf{a}}(\mathbf{s})$ is reduced to $M = \prod_{\mathrm{i}=1}^{N} |S|$ †. It can be proved by [9] that only considering the smaller set of strategy functions will not incur loss of optimality. Enumerate these functions using $\hat{\mathbf{a}}^{[m]}(\mathbf{s})$ for $m \in \{1, 2, ..., M\}$. The idea of distributed correlated scheduling is that, in each time slot, smartphones in the target region choose a strategy function in the set $\{\hat{\mathbf{a}}^{[1]}(\mathbf{s}), \hat{\mathbf{a}}^{[2]}(\mathbf{s}), ..., \hat{\mathbf{a}}^{[M]}(\mathbf{s})\}$, via a distributed but correlative approach.

Suppose all smartphones receive feedback message specifying the values of $s_1(t), s_2(t), ..., s_N(t)$ and $p_1(t), p_2(t), ..., p_N(t)$ before the end of time slot $t + D$, where D represents a system delay of at least one time slot. This assumption is realistic for distributed implementation and any mechanism for delivering this feedback message can be utilized, e.g, through piggybacking. Then virtual queue $Q_i(t)$ is defined and updated by:

$$Q_i(t + 1) = \max[Q_i(t) + p_i(t - D) - c_i, 0] \tag{6}$$

for each slot $t \in \{0, 1, 2, ...\}$ and $i \in \{1, 2, ..., N\}$, where $Q_i(0) = 0$ and $p_i(-1) = p_i(-2) = ... = p_i(-D) = 0$. Each smartphone can repeat updating the above virtual queues based on information available at the end of each time slot t. Therefore, all smartphones in the target region know the value of $Q_i(t)$ at the beginning of each time slot t. It can be proved according to [8] that stabilizing all virtual queues guarantees the time average cost constraints (4) are satisfied. Define the virtual queue vector $\mathbf{Q}(t) = (Q_1(t), Q_2(t), ..., Q_N(t))$.

First, we define the *Lyapunov function* as follows:

$$L(t) \triangleq \frac{1}{2}\sum_{i=1}^{N} Q_i(t)^2 \tag{7}$$

Then define Lyapunov drift as $\Delta(t) \triangleq L(t + 1) - L(t)$. Based on the techniques in [8] and [9], the algorithm is to choose strategy function in each time slot to greedily minimize an upper bound of the *drift-minus-utility* expression $\mathbb{E}\{\Delta(t + D) - V u(t)|\mathbf{Q}(t)\}$. The control parameter $V \ge 0$ represents an *importance weight* on how much we emphasize the utility maximization compared to cost constraints satisfaction at smartphones. The term $\Delta(t + D)$ differs from the standard Lyapunov optimization technique [8] and is utilized because the virtual queues are updated by the delayed feedback message by (6). A natural explanation of the algorithm is that we make $\Delta(t + D)$ small to maintain queue stability while adding the weighted utility to make decisions towards a large utility. We have the following lemma regarding the drift-minus-utility expression:

† This number is acceptable because, generally, there will be only a small number of smartphones participating in one target region.

***Lemma** 1*: In each time slot t, we have:

$$\mathbb{E}\{\Delta(t + D) - V u(t)|\mathbf{Q}(t)\} \le B(1 + 2D) - \sum_{i=1}^{N} c_i Q_i(t)$$
$$+\mathbb{E}\{\sum_{i=1}^{N} Q_i(t)p_i(t) - V u(t)|\mathbf{Q}(t)\} \tag{8}$$

where $B = \frac{1}{2}\sum_{i=1}^{N} {c_i}^2$ is a finite constant.

Proof: First, squaring both sides of (6), and using the fact that $\max[x, 0]^2 \le x^2$, we have:

$$Q_i(t + D + 1)^2 - Q_i(t + D)^2 \le (p_i(t) - c_i)^2$$
$$+2Q_i(t + D)(p_i(t) - c_i)$$

Summing over $i \in \{1, 2, ..., N\}$ and dividing by 2 yields:

$$\begin{aligned} \Delta(t + D) \;\le\; & \frac{1}{2}\sum_{i=1}^{N} (p_i(t) - c_i)^2 \\ & + \sum_{i=1}^{N} Q_i(t + D)(p_i(t) - c_i) \\ =\; & \frac{1}{2}\sum_{i=1}^{N} (p_i(t) - c_i)^2 + \sum_{i=1}^{N} Q_i(t)(p_i(t) - c_i) \\ & + \sum_{i=1}^{N} (Q_i(t + D) - Q_i(t))(p_i(t) - c_i) \end{aligned} \tag{9}$$

Moreover, by defining $B = \frac{1}{2}\sum_{i=1}^{N} {c_i}^2$ we have:

$$\frac{1}{2}\sum_{i=1}^{N} \mathbb{E}\{(p_i(t) - c_i)^2|\mathbf{Q}(t)\} \le B \tag{10}$$

From (6), we can obtain:

$$|Q_i(t + 1) - Q_i(t)| \le |p_i(t - D) - c_i|, \forall i, \forall t$$

Therefore,

$$\begin{aligned} |Q_i(t + D) - Q_i(t)| \;\le\; & \sum_{d=1}^{D} |Q_i(t + d) - Q_i(t + d - 1)| \\ \le\; & \sum_{d=1}^{D} |p_i(t + d - 1 - D) - c_i| \end{aligned}$$

Thus,

$$\sum_{i=1}^{N} (Q_i(t + D) - Q_i(t))(p_i(t) - c_i) \quad \le$$
$$\sum_{i=1}^{N}\sum_{d=1}^{D} |p_i(t_d) - c_i||p_i(t) - c_i|$$

where $t_d = t - d - 1 - D$. Taking expectations and using the

Cauchy-Schwartz inequality leads to:

$$\begin{aligned}
&\mathbb{E}\{\sum_{i=1}^{N}(Q_i(t+D)-Q_i(t))(p_i(t)-c_i)\} \\
\leq\ & \sum_{i=1}^{N}\sum_{d=1}^{D}\sqrt{\mathbb{E}[|p_i(t_d)-c_i|^2]}\sqrt{\mathbb{E}[|p_i(t)-c_i|^2]} \\
\leq\ & \sum_{d=1}^{D}\sqrt{\sum_{i=1}^{N}\mathbb{E}[|p_i(t_d)-c_i|^2]}\sqrt{\sum_{i=1}^{N}\mathbb{E}[|p_i(t)-c_i|^2]} \\
\leq\ & 2BD
\end{aligned}$$

Taking conditional expectations on both sides of (9), applying the above inequality and (10), we can see that lemma 1 holds.

The *drift-minus-utility* algorithm is to choose a pure strategy function $\hat{\mathbf{a}}^{[m]}(\mathbf{s})$ from the set $\{\hat{\mathbf{a}}^{[1]}(\mathbf{s}), \hat{\mathbf{a}}^{[2]}(\mathbf{s}), ..., \hat{\mathbf{a}}^{[M]}(\mathbf{s})\}$ to greedily minimize an upper bound of the expression $\mathbb{E}\{\Delta(t+D) - Vu(t)|\mathbf{Q}(t)\}$, that is to minimize term (8), in each time slot t. Since each smartphone doesn't have the knowledge of phone contexts of others in slot t (i.e., **s**(t)), the value of term (8) under a certain candidate strategy $\hat{\mathbf{a}}^{[m]}(\mathbf{s})$ cannot be calculated. But the delayed information $\mathbf{s}(t-D)$ is available at the end of time slot t. Based on the idea in [10], the expectations of $p_i(t)$ and $u(t)$ under strategy $\hat{\mathbf{a}}^{[m]}(\mathbf{s})$ can be approximated as follows:

$$\tilde{p}_i^{[m]}(t) = \frac{1}{W}\sum_{w=1}^{W}\hat{a}_i^{[m]}(s_i(t-D-w))$$

$$\tilde{u}^{[m]}(t) = \frac{1}{W}\sum_{w=1}^{W}\hat{u}(\mathbf{s}(t-D-w), \hat{\mathbf{a}}^{[m]}(\mathbf{s}(t-D-w)))$$

where W is a positive integer which represents a moving average window size.

Then we can derive the distributed scheduling algorithm for each smartphone $i \in \{1, 2, ..., N\}$, as illustrated by Algorithm 1 in details.

Algorithm 1 Distributed Optimal Scheduling Algorithm

Initialization: Set the parameters V and W. Initialize the virtual queue vector $\mathbf{Q}(0) = 0$.

In each time slot t:

1: Smartphone i detects its phone context $s_i(t)$ and observes the queue vector $\mathbf{Q}(t)$;

2: Smartphone i chooses the pure strategy function $\hat{\mathbf{a}}^{[m]}(\mathbf{s})$ from the set $\{\hat{\mathbf{a}}^{[1]}(\mathbf{s}), \hat{\mathbf{a}}^{[2]}(\mathbf{s}), ..., \hat{\mathbf{a}}^{[M]}(\mathbf{s})\}$ that minimizes the following expression:

$$\sum_{i=1}^{N} Q_i(t)\tilde{p}_i^{[m]}(t) - V\tilde{u}^{[m]}(t) \tag{11}$$

3: Smartphone i applies the sensing decision $a_i(t) = \hat{a}_i^{[m]}(s_i(t))$;

4: Receive the delayed feedback specifying the values of $s_1(t-D), s_2(t-D), ..., s_N(t-D)$ and $p_1(t-D), p_2(t-D), ..., p_N(t-D)$ and update all virtual queues by (6).

B. Performance Analysis

We analyze the performance of the distributed optimal scheduling algorithm by the following theorem 1.

***Theorem** 1:*For arbitrary phone contexts $s_1(t), s_2(t), ..., s_N(t)$, under Algorithm 1 with $V \geq 0$ and $W > 0$, we have:

a) The the gap between time average utility achieved by Algorithm 1 and the optimal time average utility that can be achieved by any other distributed algorithms is within $O(1/V)$:

$$\begin{aligned}
\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}[u(t)] \quad &\geq \quad u^{OPT} - O(1/\sqrt{W}) \\
&- \quad \{\frac{B(1+2D)}{V} + \frac{\mathbb{E}[L(D)]}{Vt}\}
\end{aligned} \tag{12}$$

b) The time average cost constraint on each smartphone $i \in \{1, 2, ..., N\}$ satisfies:

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}[p_i(t)] \leq c_i + O(\sqrt{\frac{V}{t}}) \tag{13}$$

Proof: It is proved in [9] that the optimal utility can be achieved by correlated scheduling with a knowledge of the probability distribution of phone contexts. The optimal algorithm satisfies $\mathbb{E}\{p_i^{OPT}(t)\} \leq c_i, \forall i$. The drift-minus-utility algorithm is to minimize the upper bound of drift-minus-utility expression in every time slot. Therefore:

$$\mathbb{E}\{\Delta(\tau+D) - Vu(\tau)|\mathbf{Q}(\tau)\} \leq B(1+2D) - Vu^{OPT}$$

Taking expectations of both sides and using iterated expectations leads to:

$$\mathbb{E}\{\Delta(\tau+D)\} - V\mathbb{E}\{u(\tau)\} \leq B(1+2D) - Vu^{OPT}$$

Summing the above over time slots $\tau \in \{0, 1, ..., t-1\}$, we have:

$$\begin{aligned}
\mathbb{E}\{L(t+D)\} - \mathbb{E}\{L(D)\} - V\sum_{\tau=0}^{t-1}\mathbb{E}\{u(\tau)\} \\
\leq B(1+2D)t - Vu^{OPT}t
\end{aligned} \tag{14}$$

Dividing both sides by Vt, considering the fact that $\mathbb{E}\{L(t+D)\} \geq 0$ yields:

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}[u(t)] \geq u^{OPT} - \{\frac{B(1+2D)}{V} + \frac{\mathbb{E}[L(D)]}{Vt}\}$$

It can be proved by [10] that the gap between Algorithm 1, which utilizes approximations without the knowledge of probability distribution of phone contexts, and the exact drift-minus-utility algorithm is within $O(1/\sqrt{W})$. Thus, (12) holds. This completes the proof of part a).

From (14), we also have:

$$\mathbb{E}\{L(t+D)\} \leq \mathbb{E}\{L(D)\} + [B(1+2D) + VU]t$$

where U is a constant that stratifies $U \geq \mathbb{E}\{u(\tau)\} - u^{OPT}, \forall t$. Using the definition of $L(t+D)$ and Jensen's inequality leads

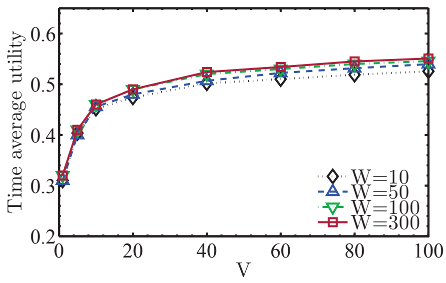

Fig. 2. Time average utility vs. V

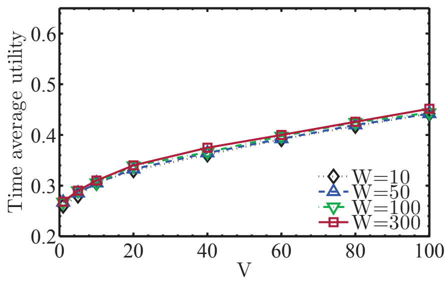

Fig. 3. Time average utility vs. V $(D = 100)$

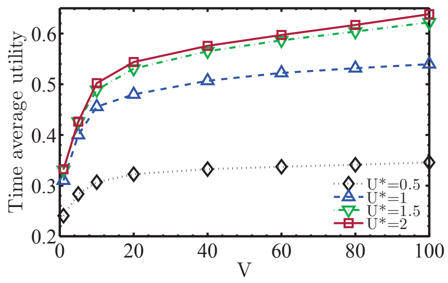

Fig. 4. Time average utility vs. V $(W = 50)$

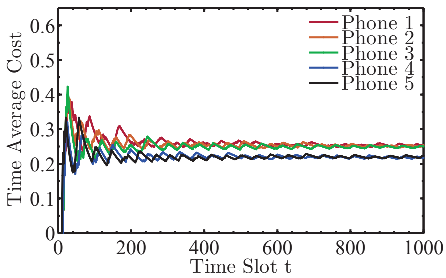

Fig. 5. Time average cost up to t ($\overline{p}_i(t)$) vs. time $(V = 10, W = 50)$

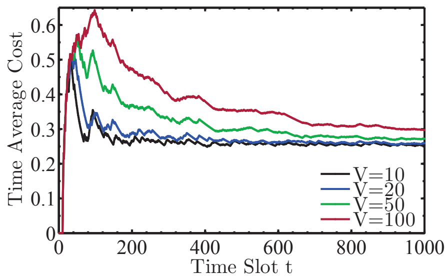

Fig. 6. $\max[\overline{p}_1(t), \overline{p}_2(t), \overline{p}_3(t), \overline{p}_4(t), \overline{p}_5(t)]$ vs. time $(W = 50)$

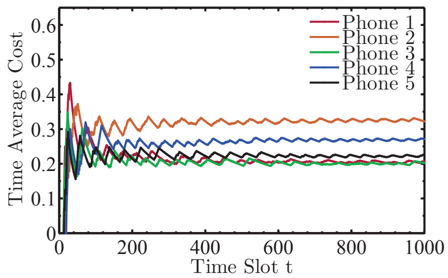

Fig. 7. Time average cost up to t ($\overline{p}_i(t)$) vs. time $(V = 10, W = 50)$

to:

$$\frac{\mathbb{E}\{||\mathbf{Q}(t+D)||\}}{t} \leq \sqrt{\frac{2\mathbb{E}\{L(D)\}}{t^2} + 2\frac{B(1+2D)+VU}{t}} \tag{15}$$

It can be easily proved by the idea in [8] that the following inequality holds:

$$\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{p_i(\tau - D)\} \leq c_i + \frac{\mathbb{E}\{Q_i(t)\}}{t} - \frac{\mathbb{E}\{Q_i(0)\}}{t}$$

Combining with (15), we see that (13) holds. This completes the proof of part b).

Theorem 1 shows that, fixing the window size W, our distributed scheduling algorithm can achieve a time average utility that is within $O(1/V)$ of the optimal value. Larger values of V will push the time average utility closer to the optimum. But the tradeoff is that more time is needed for the time average cost of each smartphone to get close to the required cost constraint. What's more, large value of W can increase the time average utility. But it will result in long computation time and large storage on smartphones.

IV. Performance Evaluation

In this section, we conduct simulations to evaluate our distributed online scheduling algorithm for mobile crowd sensing. Consider a target region which has $N = 5$ smartphones in it. The random phone context $s_i(t)$ of each smartphone $i \in \{1,2,3,4,5\}$ takes values from the set $S = \{0,1,2\}$. That is to say each smartphone has three possible contexts: $s_i(t) = 2$ means that smartphone i meets the sensing application's request very well in time slot t while $s_i(t) = 0$ means that it doesn't meet the request. Assume $s_i(t), i \in \{1,2,3,4,5\}$ are uniformly and randomly distributed over S. The trust of each smartphone R_i is set to $[0.4, 0.3, 0.2, 0.1, 0.1]$ respectively. The time average cost constraints c_i are set to $1/4$ for all smartphones. We use $U^* = 1$ for the utility function (1). The default system delay for the feedback messages is $D = 10$. And the default value of W is 50. Each simulation is run for $1,000$ time slots.

A. Verification of Optimality

First, we verify the utility optimality achieved by our algorithm. Fig. 2 shows how the parameter V affects the time average utility with different values of W. We see that the utility improves significantly and converges quickly towards the optimum as the value of V increases. The impact of W is not such obvious. The utility just improves a little when W varies from 10 to 50. The improvement can even be negligible when W is further increased. Fig. 3 shows similar results with a much larger system delay $D = 100$. Compared to Fig. 2, we see that the time average utility may decrease if the system suffers a large delay for delivering feedback messages. Fig. 4 illustrates the impact of U^* in the utility function (1) on time average utility. It means that more utility can be achieved if the problem of information saturation is not very serious. The curves for $U^* = 1.5$ and $U^* = 2$ look identical because there are cost constraints at smartphones so that they can not perform more sensing tasks.

Second, we verify whether the cost constraints at smartphones are satisfied. In Fig. 5, the curves plot time average cost up to time slot t of each smartphone (1-5). We can see that the time average cost of each smartphone satisfies the constraint $c_i \leq 1/4$. The time average cost of smartphone 1 and 2 is larger than that of others because they have higher trust so that the algorithm tends to schedule them to perform sensing task if their phone contexts meet the application's request. Fig. 6 demonstrates how the parameter V affects the time required to converge to the desired constraints. The curves plot the maximal time average cost among smartphones. This verifies the fact that larger values of V push the time average utility closer to the optimum with the tradeoff in the amount of time required for the time average cost to converge to the required constraint. Fig. 7 shows that the constraints are still satisfied when we reduce the constraints of smartphone 1 and 3 to $1/5$ and increase the constraints of smartphone 2 and 4 to $1/3$.

The simulation results above verify Theorem 1.

B. Adaption to Changes

Next, we demonstrate that our algorithm can adapt to changes robustly. The simulation time is increased to $1,500$ slots which is divided into three phases. Each phase is of 500 time slots. Note that the phone context processes $s_i(t)$ are uniformly distributed over $\{0, 1, 2\}$ for all smartphones in the above simulations. We keep that probability distribution in phase 1 and phase 3, but abruptly change the probabilities for smartphone 1-4 in phase 2, according to the following table.

	Pr[$s_i(t) = 0$]	Pr[$s_i(t) = 1$]	Pr[$s_i(t) = 2$]
i=1,3	0.8	0.1	0.1
i=2,4	0.5	0.5	0

Fig. 8 and Fig. 9 show the average utility and the average cost of smartphone 1 over $1,500$ time slots. Values at each time slot t are obtained by averaging the utility and cost in that slot over 300 independent simulation runs. We see that the system can adapt to the changes in probability distribution of smpartphone context quickly by adjusting to the new optimal average utility. And the cost constraint is still stratified with only small disturbance in a short time.

We also demonstrate adaption to smartphone's mobility. The mobile smartphones may leave or enter a target region over time. We simulate the mobility by making smartphone 1 and smartphone 2 leave the target region in phase 2, and have another smartphone with a trust of 0.4 join in phase 3. Fig. 10 and Fig. 11 show the results. We see that the algorithm can quickly adapt to the changes incurred by mobility. The average utility adjusts fast to the new optimal value when changes occur. The average cost of smartphone 3 is always satisfied despite small disturbance.

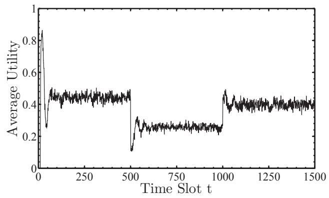

Fig. 8. Average utility vs. time

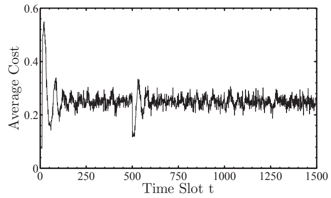

Fig. 9. Average cost of smartphone 1 vs. time

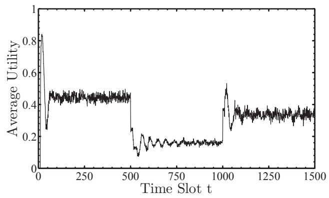

Fig. 10. Average utility vs. time

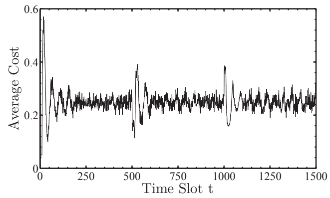

Fig. 11. Average cost of smartphone 3 vs. time

V. Related Work

Due to the fast increasing of usage of smartphones, mobile crowd sensing is becoming more and more popular in recent years and has attracted extensive research attention from both academia and industry. The research trend started with the notion of *participatory sensing* which requires the user interaction to sense particular events. Then research evolved into *opportunistic sensing* which enlarges the vision of mobile crowd sensing by allowing the cooperation of multiple smartphones without requiring the explicit interaction with users.

A great number of opportunistic sensing applications have been designed and implemented. For example, GeoServ [7] is a scalable sensor networking platform where millions of users can participate in urban sensing and share location-aware information using always-on cellular data connections. Nericell [11] is a system that performs rich sensing by piggybacking on mobile phones that users carry with them in normal course. The system could be used to annotate traditional traffic maps with information such as the bumpiness of roads, and the noisiness and level of chaos in traffic. The recent work [12] presents Sensor Mobile Enablement (SME), which is a lightweight standard for efficiently identifying, coding and decoding heterogeneous sensing information on mobile devices. More examples can be found in a recent survey paper [13]. But most of these applications don't consider the problem of limited mobile phone resources or information saturation.

Little existing work has studied the problem of efficient scheduling to achieve optimal utility considering the marginal effect with limited resource of smartphones. And most of the related work requires sufficient statistical knowledge and perform in offline manner or prediction-based approach. For example, in [14], the authors study an energy efficient problem in mobile crowd sensing and propose prediction-based algorithms to minimize the energy consumptions at smartphones. In [15], the authors develops a novel smartphone based vehicular crowd sensing system that achieves efficient utilization of limited 3G budgets to improve system performance. They propose heuristic algorithm based on the statistic data to estimate whether a WiFi encounter is approaching so as to make decisions. Their feasibility heavily depends on the the accuracy of the prediction of future patterns and can not guarantee the optimal performance. In comparison, our optimal online scheduling algorithm does not require any priori knowledge of the future patterns and can achieve a time average utility that could be arbitrarily close to the optimum, in a distributed manner.

VI. CONCLUSION

This paper has presented a distributed algorithm for maximizing the utility of sensing data collection in a mobile crowd sensing system. The algorithm leverages both the stochastic network optimization technique and distributed correlated scheduling. It does not require any priori knowledge of smartphone contexts in the future, and supports individual smartphones to make their own sensing decisions. We have performed rigorous theoretical analysis to show that the proposed algorithm can achieve a time average utility that is within $O(1/V)$ of the optimum. Extensive simulations have been carried out, and the results show that the proposed algorithm achieves high time average utility of collected sensing data.

REFERENCES

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 140–150, 2010.

[2] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 32–39, 2011.

[3] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 2010, pp. 105–116.

[4] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 55–68.

[5] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell, "Urban sensing systems: opportunistic or participatory?" in *Proceedings of the 9th workshop on Mobile computing systems and applications*. ACM, 2008, pp. 11–16.

[6] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 173–184.

[7] J. H. Ahnn, U. Lee, and H. J. Moon, "Geoserv: A distributed urban sensing platform," in *Cluster, Cloud and Grid Computing (CCGrid), 2011 11th IEEE/ACM International Symposium on*. IEEE, 2011, pp. 164–173.

[8] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[9] ——, "Distributed stochastic optimization via correlated scheduling," *arXiv preprint arXiv:1304.7727*, 2013.

[10] M. J. Neely, S. T. Rager, and T. F. La Porta, "Max weight learning algorithms for scheduling in unknown environments," *Automatic Control, IEEE Transactions on*, vol. 57, no. 5, pp. 1179–1191, 2012.

[11] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 323–336.

[12] V. Arnaboldi, M. Conti, F. Delmastro, G. Minutiello, and L. Ricci, "Sensor mobile enablement (sme): A light-weight standard for opportunistic sensing services," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*. IEEE, 2013, pp. 236–241.

[13] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: a survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 1, pp. 402–427, 2013.

[14] X. Sheng, J. Tang, and W. Zhang, "Energy-efficient collaborative sensing with mobile phones," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1916–1924.

[15] H. Liu, S. Hu, W. Zheng, Z. Xie, S. Wang, P. Hui, and T. Abdelzaher, "Efficient 3g budget utilization in mobile participatory sensing applications," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 1411–1419.