Dynamic Task Assignment in Crowdsensing with Location Awareness and Location Diversity

Xiong Wang, Riheng Jia, Xiaohua Tian, Xiaoying Gan
Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China
Email:{wangxiongsjtu, jiariheng, xtian, ganxiaoying}@sjtu.edu.cn

***Abstract*—Crowdsensing paradigm facilitates a wide range of data collection, where great efforts have been made to address its fundamental issue of matching workers to their assigned tasks. In this paper, we reexamine this issue by considering the spatio-temporal worker mobility and task arrivals, which more fits the actual situation. Specifically, we study the location-aware and location diversity based dynamic crowdsensing system, where workers move over time and tasks arrive stochastically. We first exploit offline crowdsensing by proposing a combinatorial algorithm, for efficiently distributing tasks to workers. After that, we mainly study the online crowdsensing, and further consider an indispensable aspect of worker's fair allocation. Apart from the stochastic characteristics and discontinuous coverage, the non-linear expectation is incurred as a new challenge concerning fairness issue. Based on Lyapunov optimization with perturbation parameters, we propose online control policy to overcome those challenges. Hereby we can maintain system stability and achieve a time average sensing utility arbitrarily close to the optimum. Performance evaluation on real data set validates the proposed algorithm, where 116% gain of fairness is achieved at the expense of 12% loss of sensing value on average.**

I. INTRODUCTION

Recent years have witnessed the unprecedented development of mobile devices which are embedded with powerful processers and plentiful sensors (e.g., GPS, thermometer, microphone, camera). A newly-emerged crowdsensing paradigm is recognized to outsource a collection of sensing tasks to workers that carry mobile devices. Thereafter, numerous crowdsensing systems have been implemented to collect and process large-scale sensory data, such as *SmartRoad* for traffic detection [1], *TransitLabel* for transit stations labeling [2], *Aircloud* for air quality monitoring [3], and *Ear-phone* for noise map construction [4].

Considering the sensing process in practical crowdsensing systems [1]-[4], gathering data at specific locations or regions is of central importance. The reason behind is that data collected at irrelevant locations are meaningless and useless. In line with this observation, *location awareness* requires that workers can perform those spatial sensing tasks if they happen to be in the vicinities.

Regarding location awareness, location diversity is proposed to characterize the spatial influence on workers and tasks. Workers are unevenly distributed in space and their mobility patterns also vary dramatically, thus task's priority and worker's sensing cost are heterogeneous with respect to locations [5][6]. Location diversity can bridge the gap between physical context and worker (task) heterogeneity [7].

Corresponding author: Xiaoying Gan

Few works have considered location information in crowdsensing system. Different from most studies without location awareness [10][11], location consideration will introduce more spatial constraints [6][8], which increase the problem complexity of task assignment but meanwhile improve the matching precision between task requirement and work ability. However, these location-aware studies only focus on offline crowdsensing, in which the information of workers and tasks is given in prior.

A more realistic situation is the online crowdsensing, where the system state is dynamic with time varying information of workers and tasks. As it is challenging to predict the future system state, a feasible way is to use the current information for online decision-making [12][13]. However, none of these works consider the location awareness in task assignment due to the hardness of handling dynamic spatial constraints. Another indispensable element in online crowdsensing is worker's fair allocation, which guarantees adequate completion for even low value tasks in a long period.

There is an imperious demand to study the spatio-temporal factors in crowdsensing system. On the one hand, workers and tasks are specified at different locations with different location diversities. On the other hand, workers will move over time while tasks will arrive dynamically. Therefore, online crowdsensing with location awareness and location diversity can capture the features of both spatio-temporal aspects, which also more conforms to the actual situation. Unfortunately, most of the existing works have ignored these features in crowdsensing system.

In this paper, we are motivated to address the task assignment problem considering location awareness and location diversity in both offline and online crowdsensing. However, it is particularly challenging due to the following three-fold reasons. *First*, location awareness makes the matching between workers and tasks further constrained by their geo-positions, thus increasing the problem complexity. *Second*, discontinuous coverage is incurred by worker movement in online crowdsensing, hence dynamic task assignment is sensitive to both location and time. *Third*, worker's fair allocation introduces the non-linear expectation problem which is hard to handle.

To overcome these challenges, we first carefully characterize an entropy based metric to measure location diversity. After that, we formulate location-aware models for offline

978-1-5386-4128-6/18/$31.00 ©2018 IEEE

and online crowdsensing with location diversity. For offline crowdsensing, we design a combinatorial algorithm to decouple the offline problem into each subproblem, which is optimally solved through an efficient searching method. We mainly focus on the online crowdsensing, where we harness Lyapunov optimization to bypass the problem of uncertain future information. We capture the features of mobile worker movement and dynamic task arrival in online crowdsensing, and ensure no-underflow [16][17] of tasks to prevent a waste of worker resources. Aside from optimizing the sensing value in online crowdsensing, we extend the model to maximize the proportional fairness that achieves a fair allocation of worker resources. The performance evaluated on real data set proves the stability and efficiency of the crowdsensing system. The main contributions are summarized as follows:

- We comprehensively study the location-aware and location diversity based offline and online crowdsensing systems, where the rarely considered spatio-temporal features of both workers and tasks are captured. To the best of our knowledge, we are the first to study such a crucial but non-trivial problem in crowdsensing.
- In offline crowdsensing, we design a combinatorial algorithm to decompose the task assignment problem into multiple subproblems which are optimally solved through branch-and-bound. The obtained result is proved to be 2-approximate compared with the optimal solution.
- In online crowdsensing, we propose a perturbed Lyapunov function to deal with discontinuous coverage and worker's fair allocation. Auxiliary variable is introduced to handle the non-linear expectation incurred by fairness consideration. We can maintain system stability and achieve a time average utility within $O(1/V)$ of the optimum for any $V > 0$.

In the rest of this paper, we first formulate the offline and online crowdsensing models in Section II. Section III provides the combinatorial algorithm for offline crowdsensing. Followed by the main focus on Lyapunov optimization for online crowdsensing in Section IV. Evaluations are performed in Section V. We finally discuss the related work in Section VI and draw the conclusion in Section VII.

II. System Model And Problem Formulation

A. Location Diversity

Two types of location diversity are frequently recognized: Task Location Diversity (TLD) and Worker Location Diversity (WLD), which are measured in terms of location entropy [7].

A visit to a location is identified by worker ID, location index (such as latitude and longitude), and timestamp. For a given location l in the set of locations $\mathbb{L}$, O_l is the total count of the visits by the set of workers $\mathbb{U}$. Let $O_u, u \in \mathbb{U}$ be the set of location visits of worker u with $O = \cup_{u\in\mathbb{U}} O_u$. The set of workers visiting location l is denoted as $\mathbb{U}_l$, similarly, the set of locations visited by worker u is noted as $\mathbb{L}_u$. Let $O_{u,l}$ be the visits to location l by worker u, namely $O_{u,l} = O_l \cap O_u$. The probability of a random drawn from O_l falls in $O_{u,l}$ is $P_l(u) = \frac{|O_{u,l}|}{|O_l|}$. Analogously, the probability of a random draw

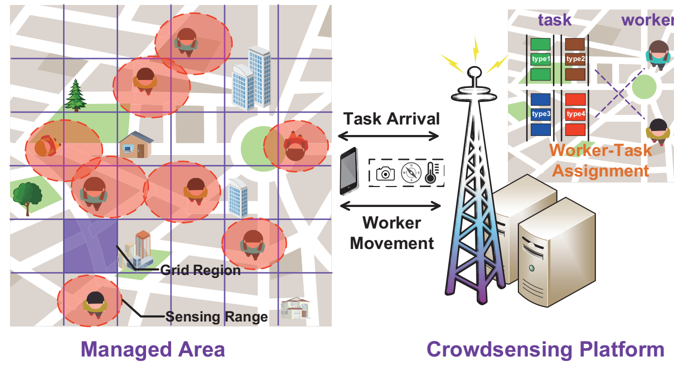

Fig. 1. Location-aware crowdsensing system. Workers walk around with sensing ranges and tasks dynamically arrive. The area is divided into grids

from O_u belongs to $O_{u,l}$ is $P_u(l) = \frac{|O_{u,l}|}{|O_u|}$. TLD and WLD are defined:

$$\begin{aligned} \text{TLD}_l &= -\sum_{u\in\mathbb{U}_l} P_l(u) \times \log P_l(u), \\ \text{WLD}_u &= -\sum_{l\in\mathbb{L}_u} P_u(l) \times \log P_u(l). \end{aligned} \tag{1}$$

Remark. High TLD implies frequent workers' visits with equal proportions, hence tasks are more likely to be completed, and vice versa. Meanwhile high WLD denotes worker visiting many locations with uniform distribution, thus the worker is inclined to walk around, and vice versa. As a consequence, task with high TLD has low priority (value), while worker with high WLD gets low moving cost.

B. System Overview

In the crowdsensing system shown in Fig. 1, there are a set of participating workers $\mathbb{W} = \{w_1, w_2, ..., w_N\}, \mathbb{W} \subseteq \mathbb{U}$, and M types of served sensing tasks $\mathbb{M} = \{1, 2, ..., M\}$ (measuring noise, taking temperature and etc.). Each worker (task) is located within a grid region of the managed area by the crowdsensing platform, which is denoted by $\mathbb{L} = \{l_1, l_2, ..., l_K\}$. Considering location awareness, worker w_i can only complete a sensing task if the location is covered by w_i's sensing range $R_i(l_{i_k})$, where l_{i_k} is w_i's located region and $R_i(l_{i_k})$ is assumed to be greater than a location region. A task of type j in region l_k is associated with a value v_{jk} composed by two parts: original value ov_j and diversity value dv_k. ov_j is the inherent value of a type j task, while dv_k represents the task priority that inversely relates to TLD_{l_k}. Analogously, worker w_i undertaking a type j task incurs a cost c_{ij} also consisting of two components: original cost oc_j and diversity cost dc_i. oc_j is the consumption of sensing data for type j task, and dc_i which is inversely related to WLD_{w_i} measures the moving cost within the sensing range $R_i(l_{i_k})$. Due to the ability heterogeneity, each worker w_i is associated with an expertise vector $\mathbf{E}_i = [e_{i1}, e_{i2}, ..., e_{iM}]$ with $e_{ij} \in [0, 1]$ denoting w_i' expertise to type j task. We assume the expertise of each worker is qualified, and task values are distinctly higher than worker costs.

C. Problem Formulation

1) Offline Crowdsensing: The information of both workers and tasks is given in offline crowdsensing. Suppose the crowdsensing platform publishes a set of sensing tasks Γ of M types. For any task $\tau_n \in \Gamma$ with identified task type n_j and located region l_{n_k}, the task value is specified as $v_{n_j n_k}$. Worker w_i has a sensing budget B_i due to the constraints of mobility pattern and sensing ability (such as battery volume, sensor configuration), then w_i's total cost is restricted by B_i. Let $b_{in} \in \{0, 1\}$ indicate the assignment between worker w_i and task τ_n. $b_{in} = 1$ means w_i undertaking τ_n if $l_{n_k} \in R_i(l_{i_k})$, otherwise $b_{in} = 0$.

The crowdsensing platform aims to optimize the overall value of sensing tasks performed by the workers under both the budget and sensing range constraints. Therefore, the *OFFlIne CrowdsEnsing of Value Maximization (OFFICE-VM)* problem is formulated as follows:

$$\begin{aligned}
\max & \sum_{w_i \in \mathbb{W}} \sum_{\tau_n \in \Gamma} b_{in} e_{in_j} v_{n_j n_k} \\
\text{s.t.} & \sum_{\tau_n \in \Gamma} b_{in} c_{in_j} \leq B_i, \forall w_i \in \mathbb{W} \\
& b_{in} \in \{0, 1\}, \text{if } b_{in} = 1, l_{n_k} \in R_i(l_{i_k}),\ \forall w_i \in \mathbb{W}, \forall \tau_n \in \Gamma \\
& \sum_{w_i \in \mathbb{W}} b_{in} \leq 1, \forall \tau_n \in \Gamma,
\end{aligned} \tag{2}$$

where e_{in_j} and c_{in_j} are w_i's expertise and cost to τ_n, respectively. The last formula denotes each task τ_n can be assigned to at most one worker. OFFICE-VM contains a set of coupled NP-hard subproblems (task assignment for each worker), which makes it much tougher than only solving every subproblem individually.

2) Online Crowdsensing: In online crowdsensing, workers continuously move around in the managed area, and tasks dynamically arrive over time. We consider the crowdsensing system runs in a slotted time $t \in \{0, 1, 2, ...\}$, which varies from minutes to hours on the basis of sensing requirements.

Let $A_{jk}(t), 0 \leq A_{jk}(t) \leq A_{jk}^{\max}$ denote the new arrival tasks of type j at region l_k in time slot t, with $A_{jk}(t)$ i.i.d over time slots. Note that l_k is a region, and the arriving tasks are specified at some locations within l_k, hence those tasks are considered different. $Q_{jk}(t)$ represents the task queue of type j at region l_k, with $a_{jk}(t), 0 \leq a_{jk}(t) \leq A_{jk}(t)$ tasks admitted into $Q_{jk}(t)$ in time slot t. Since workers are moving around, they can only complete the assigned tasks in current time slot, otherwise the assigned tasks may be out of their sensing ranges due to the movement, which is referred to as *discontinuous coverage*. Considering this feature, worker w_i can perform at most k_i tasks in each time slot because time slot length is limited. Let $b_{ijk}(t)$ be the assigned tasks in $Q_{jk}(t)$ to w_i based on sensing range $R_i(l_{i_k}(t))$ where $l_{i_k}(t)$ is w_i's current location. Therefore, the cost $c_i(t)$ of worker w_i is:

$$c_i(t) = \sum_{j=1}^{M} \sum_{k=1}^{K} b_{ijk}(t) c_{ij}, \tag{3}$$

with the time average expected cost $\overline{c_i} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[c_i(t)]$. Analogously, $\overline{c_i} \leq B_i$ since sensing ability (like battery volume) is restricted. The sensing value $u_{jk}(t)$ of task queue $Q_{jk}(t)$ is computed:

$$u_{jk}(t) = \sum_{i=1}^{N} b_{ijk}(t) e_{ij} v_{jk}, \tag{4}$$

and the time average expected sensing value is $\overline{u_{jk}} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[u_{jk}(t)]$.

Consequently, the whole time average sensing utility is:

$$f(\overline{u}) = \sum_{j=1}^{M} \sum_{k=1}^{K} f(\overline{u_{jk}}), \tag{5}$$

in which $f(\overline{u_{jk}})$ function has two forms. 1) $f(\overline{u_{jk}}) = \overline{u_{jk}}$, then $f(\overline{u}) = \overline{u} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{j=1}^{M} \sum_{k=1}^{K} \mathbb{E}[u_{jk}(t)]$ which is the whole time average sensing value, similar to the offline crowdsensing; 2) $f(\overline{u_{jk}}) = \log(1 + \beta \overline{u_{jk}})$, then $f(\overline{u}) = \lim_{T\to\infty} \sum_{j=1}^{M} \sum_{k=1}^{K} \log\{1 + \beta \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[u_{jk}(t)]\}$ that considers fairness issue in sensing utility [18]. Overall, the *ONline CrowdsEnsing of Utility Maximization (ONCE-UM)* problem is formulated as follows:

$$\begin{aligned}
\max\ & f(\overline{u}) \\
\text{s.t.}\ & \overline{c_i} \leq B_i, \forall w_i \in \mathbb{W} \\
& \sum_{j=1}^{M} \sum_{k=1}^{K} b_{ijk}(t) \leq k_i\ \forall w_i \in \mathbb{W} \\
& b_{ijk}(t) \in \{0, 1, ..., k_i\}, \forall w_i \in \mathbb{W}, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L} \\
& \text{if } b_{ijk}(t) > 0, l_k \in R_i(l_{i_k}(t)), \forall w_i \in \mathbb{W}, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L} \\
& Q_{jk}(t) \text{ is stable } \forall j \in \mathbb{M},\ \forall l_k \in \mathbb{L}.
\end{aligned} \tag{6}$$

The stability of $Q_{jk}(t)$ in the last formula means:

$$\limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[Q_{jk}(t)] < \infty. \tag{7}$$

Aside from the online characteristics, ONCE-UM has to consider worker's discontinuous coverage of tasks, and the fair allocation of worker resources. Both of them require much more extra efforts to deal with.

III. Location-aware And Location Diversity Based Offline Crowdsensing

Offline crowdsensing describes the scenario where the crowdsensing platform has received a set of sensing tasks await to be completed by enlisted workers. Therefore, the information of workers and tasks is known in advance. Offline scenario can be viewed as the case of static workers and tasks without dynamic changes.

A. Complexity Analysis of OFFICE-VM

Recall OFFICE-VM in Eq. (2). For any single worker $w_i \in \mathbb{W}$, the subproblem becomes:

$$\begin{aligned}
\max & \sum_{\tau_n \in \Gamma} b_{in} e_{in_j} v_{n_j n_k} \\
\text{s.t.} & \sum_{\tau_n \in \Gamma} b_{in} c_{in_j} \leq B_i \\
& b_{in} \in \{0, 1\}, \text{if } b_{in} = 1, l_{n_k} \in R_i(l_{i_k}), \forall \tau_n \in \Gamma,
\end{aligned} \tag{8}$$

which is a typical knapsack problem. However, solving the OFFICE-VM is no longer solving the knapsack problem for each worker separately. Because each task $\tau_n \in \Gamma$ can be assigned to at most one worker, that is the results of worker w_i can influence that of worker $w_{i'}$ if $R_i(l_{i_k})$ and $R_{i'}(l_{i'_k})$ overlap. As a consequence, the knapsack problems for different workers are coupled together.

OFFICE-VM is more like a Generalized Assignment Problem (GAP) [14] which has been proved to be APX-hard. A problem of APX-hardness is a set of NP optimization problems (the knapsack problem in this paper) that are addressed with polynomial-time approximation algorithms. However, different from traditional GAP, OFFICE-VM has a unique feature: each worker only needs to consider the sensing tasks covered by its sensing range. Hence, when handling each knapsack problem, we can seek for the solution in a much smaller subset of task set Γ. This feature allows us to design an efficient searching algorithm to find the optimal result for the subproblem in Eq. (8).

B. *Offline Combinatorial Algorithm*

As our former analysis, we can harness the methods of GAP to facilitate the settlement of OFFICE-VM, while also combing its specific feature. Due to APX-hardness, we resort to approximation algorithms to obtain a near optimal result.

Definition 1. *Let $u(x)$ be a utility function and F be the feasibility constraints. A feasible solution x is said to be $r-$approximate if $ru(x) \geq u(x^*)$, where x^* is the optimal solution.*

Intuitively, any approximation algorithm has $r > 1$, thus our goal is to acquire a lower value of r. We learn from the idea in [14] to design an *Offline Combinatorial Algorithm (OCA)*, in which an efficient searching algorithm is implemented.

The principle behind OCA is to solve the subproblem in Eq. (8) for each worker sequentially, and meanwhile utilizing an indicator vector $\mathbf{T} = [T_1, ..., T_{|\Gamma|}]$ to record the assigned worker of each task. The task indicators enable the decoupling function among subproblems for overlapping workers. For any task $\tau_n \in \Gamma$, indicator T_n is initialized to 0. If τ_n is momentarily assigned to worker w_i, let $T_n = i$. Denote $v_{n_j n_k}(i)$ as the transient value of τ_n when assigned to w_i, which is calculated:

$$v_{n_j n_k}(i) = \begin{cases} e_{in_j} v_{n_j n_k} & \text{if } T_n = 0 \\ e_{in_j} v_{n_j n_k} - e_{i'n_j} v_{n_j n_k} & \text{if } T_n = i'. \end{cases} \tag{9}$$

For instance, when processing the subproblem for w_i, if $l_{n_k} \in R_i(l_{i_k})$ and $T_n = i'$, then $v_{n_j n_k}(i) = e_{in_j} v_{n_j n_k} - e_{i'n_j} v_{n_j n_k}$. If w_i undertakes τ_n, update $T_n = i$. Through this operation, the overall sensing value is certainly increased since only if $v_{n_j n_k}(i) > 0$, namely $e_{in_j} > e_{i'n_j}$, τ_n can be selected by w_i.

Now, we shed light on the solution to the subproblem described in Eq. (8), which is proved to be a knapsack problem. A shared approach is the greedy algorithm that can achieve $\frac{e}{e-1}$ approximation ratio for each subproblem. However, since the covered task set for each worker is highly

Algorithm 1: OCA for OFFICE-VM

Input: Workers $\mathbb{W}$, costs $\{c_{ij}\}$, budgets $\{B_i\}$, expertise $\{\mathbf{E}_i\}$, sensing ranges $\{R_i(l_{i_k})\}$, tasks Γ, task locations $\mathbb{L}_\Gamma$, values $\{v_{jk}\}$

Output: Worker-Task Assignment

1 $\mathbf{T} = \mathbf{0}$;
2 **for** $w_i \in \mathbb{W}$ **do**
3     $\Gamma_i = \emptyset$;
4   **for** $\tau_n \in \Gamma$ **do**
5   **if** $l_{n_k} \in R_i(l_{i_k})$ **then**
6             $\Gamma_i = \Gamma_i \cup \tau_n$;
7   calculate $v_{n_j n_k}(i)$ according to (9);
8   revoke branch-and-bound for w_i;
9   **for** $\tau_n \in \Gamma_i$ **do**
10   **if** τ_n *is assigned to* w_i **then**
11             $T_n = i$;
12 **return** $\mathbf{T}$;

likely to be much smaller than Γ, we can appeal to an efficient searching algorithm to obtain the optimal result. Considering Eq. (8), we propose branch-and-bound algorithm to obtain the optimal solution.

The OCA for OFFICE-VM is depicted in Algorithm 1. We first solve the subproblem for each worker (Lines 1-8), and then update the task indicators to record the transient assigned workers (Lines 9-11). Note that we utilize branch-and-bound (Line 8) to obtain the optimal solution for each subproblem under the sensing range constraint (Lines 5-7). A detailed description for branch-and-bound is also presented in technical report [20] due to space limit.

Lemma 1. *OCA in Algorithm 1 is a 2-approximation for OFFICE-VM.*

Proof: See Appendix A in technical report [20]. ■

IV. Location-aware and Location Diversity Based Online Crowdsensing

Online crowdsensing represents a more practical situation, where tasks dynamically arrive and workers continuously walk around over time. The future information of workers and tasks is unknown and unpredictable in prior, thus making it much more difficult to address ONCE-UM in Eq (6). Considering these features, we exploit Lyapunov optimization [15] to circumvent the challenging problems.

Different from previous studies of Lyapunov optimization [12][13], the discontinuous coverage caused by worker movement makes it impossible to accumulate tasks in each worker, thus the task assignment is sensitive to both location and time. Apart from this characteristic, fairness issue in long-term online crowdsensing should also be further investigated. To comprehensively study online crowdsensing, we first consider to maximize the sensing value, and then extend the model to optimize the proportional fairness of worker resources.

A. Online Crowdsensing of Value Maximization

According to Eq. (5), when maximizing the time average sensing value, $f(\overline{u_{jk}}) = \overline{u_{jk}}$. ONCE-UM problem in Eq. (6) becomes *ONline CrowdsEnsing of Value Maximization (ONCE-VM)* problem with the objective function $f(\overline{u}) = \overline{u} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}[u_{jk}(t)]$.

Combining the value $u_{jk}(t)$ of task queue $Q_{jk}(t)$ in Eq. (4), the whole sensing value $u(t)$ in time slot t is:

$$u(t) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K} b_{ijk}(t)e_{ij}v_{jk}. \tag{10}$$

1) Queue Dynamics: In time slot t, $b_{ijk}(t)$ tasks in task queue $Q_{jk}(t)$ are completed by worker w_i, and $a_{jk}(t)$ new tasks are admitted into $Q_{jk}(t)$. Denote $b_{jk}(t) = \sum_{i=1}^{N} b_{ijk}(t)$ as the tasks assigned to all the workers $\mathbb{W}$, thus $Q_{jk}(t)$ evolves according to:

$$Q_{jk}(t+1) = \max[Q_{jk}(t) - b_{jk}(t), 0] + a_{jk}(t). \tag{11}$$

Note that $b_{jk}(t) = \sum_{i=1}^{N} b_{ijk}(t) \leq \sum_{i=1}^{N} k_i$ and $a_{jk}(t) \leq A_{jk}(t) \leq A_{jk}^{\max}$. For notation convenience, let $\mathbf{Q}(t) = \{Q_{jk}(t) : \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L}\}$.

Each worker w_i has a cost budget B_i with $\overline{c_i} \leq B_i$, hence we need to maintain a (virtual) cost queue $Z_i(t)$ for w_i. The queue dynamics of $Z_i(t)$ is:

$$Z_i(t+1) = \max[Z_i(t) + c_i(t) - B_i, 0], \tag{12}$$

where $c_i(t) = \sum_{j=1}^{M}\sum_{k=1}^{K} b_{ijk}(t)c_{ij} \leq k_i \max_j c_{ij}$ according to Eq. (3). To satisfy the budget constraint, $Z_i(t)$ should be stable, which is similar to Eq. (7). Denote $\mathbf{Z}(t) = \{Z_i(t) : w_i \in \mathbb{W}\}$, and $\mathbf{\Theta}(t) = (\mathbf{Q}(t), \mathbf{Z}(t))$.

2) Lyapunov Optimization: In each time slot, we are supposed to make the decision based on the current available information. Lyapunov optimization dispenses us with the unreliable prediction of future system state. Define *perturbed Lyapunov function* in terms of $Q_{jk}(t)$ and $Z_i(t)$ as follows:

$$\begin{aligned} L(\mathbf{\Theta}(t)) &= \frac{1}{2}||\mathbf{Q}(t) - \boldsymbol{\theta}|| + \frac{1}{2}||\mathbf{Z}(t)|| \\ &= \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{K}(Q_{jk}(t) - \theta_{jk})^2 + \frac{1}{2}\sum_{i=1}^{N} Z_i^2(t). \end{aligned} \tag{13}$$

θ_{jk} is a perturbation parameter which can guarantee *no-underflow* in task queue to avoid a waste of worker resources [16][17]. No-underflow constraint prevents the situation where workers are left unallocated for a long time. If we push Lyapunov function to a small value, each queue backlog will stay in a low level, thus maintaining queue stability.

To achieve queue stability, we utilize one slot (conditional) *Lyapunov drift*:

$$\Delta(\mathbf{\Theta}(t)) = \mathbb{E}[L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)], \tag{14}$$

where the expectation is due to the randomness of dynamics in workers and tasks. Meanwhile, we also need to balance the maximization of the sensing value. Therefore, *drift-minus-utility* is considered: $\Delta(\mathbf{\Theta}(t)) - V\mathbb{E}[u(t)|\mathbf{\Theta}(t)]$. $V > 0$ is a tunable parameter that represents an "importance weight" on how much we stress maximizing the sensing value.

Lemma 2. *drift-minus-utility* $\Delta(\mathbf{\Theta}(t)) - V\mathbb{E}[u(t)|\mathbf{\Theta}(t)]$ *satisfies:*

$$\begin{aligned} &\Delta(\mathbf{\Theta}(t)) - V\mathbb{E}[u(t)|\mathbf{\Theta}(t)] \\ &\leq D - \sum_{i=1}^{N}\mathbb{E}[Z_i(t)B_i|\mathbf{\Theta}(t)] + \sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}[a_{jk}(t)(Q_{jk}(t) - \theta_{jk})|\mathbf{\Theta}(t)] \\ &+ \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}\{b_{ijk}(t)[Z_i(t)c_{ij} - (Q_{jk}(t) - \theta_{jk}) - Ve_{ij}v_{jk}]|\mathbf{\Theta}(t)\}, \end{aligned} \tag{15}$$

where $D = \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{K}\max[(A_{jk}^{\max})^2, (\sum_{i=1}^{N} k_i)^2] + \frac{1}{2}\sum_{i=1}^{N}\max[B_i^2, (k_i \max_j c_{ij})^2]$.

Proof: See Appendix B in technical report [20]. ■

We are interested in **minimizing the upper bound of drift-minus-utility** in Lemma 2 to enable the tradeoff between queue stability and value maximization.

3) Online Control Policy: We design an *Online Control Policy (OCP)* to minimize the upper bound of drift-minus-utility only based on current available information, that can achieve $O(1/V)$ of the maximum time average sensing value. Specifically, when observing queues $Q_{jk}(t)$ and $Z_i(t)$, OCP will decide $a_{jk}(t)$ and $b_{ijk}(t)$ accordingly. From Eq. (15), the first and second terms D and $\sum_{i=1}^{N}\mathbb{E}[Z_i(t)B_i|\mathbf{\Theta}(t)]$ on the right hand are given in each time slot t, thus OCP is going to deal with the third and fourth terms.

a) Task Admission Control: For each task queue $Q_{jk}(t)$, the new arrival tasks are $A_{jk}(t)$, among which $a_{jk}(t)$ are admitted into $Q_{jk}(t)$. We choose $a_{jk}(t)$ in consistent with the following problem:

$$\begin{aligned} &\min\ a_{jk}(t)(Q_{jk}(t) - \theta_{jk}) \\ &\text{s.t.}\ \ 0 \leq a_{jk}(t) \leq A_{jk}(t). \end{aligned} \tag{16}$$

The decision of $a_{jk}(t)$ reduces to a simple threshold rule:

$$a_{jk}(t) = \begin{cases} 0 & \text{if } Q_{jk}(t) \geq \theta_{jk} \\ A_{jk}(t) & \text{if } Q_{jk}(t) < \theta_{jk}. \end{cases} \tag{17}$$

Intuitively, task admission of OCP can be carried out in a distributed manner for each $Q_{jk}(t)$.

b) Worker-Task Assignment: Worker w_i can complete $b_{ijk}(t)$ tasks out of $Q_{jk}(t)$, which is restricted by sensing range $R_i(l_{i_k}(t))$. Besides, $\sum_{j=1}^{M}\sum_{k=1}^{K} b_{ijk}(t) \leq k_i$ due to the limited time slot length. Therefore, the worker-task assignment of OCP is performed according to the following problem:

$$\begin{aligned} &\min \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K} b_{ijk}(t)[Z_i(t)c_{ij} - (Q_{jk}(t) - \theta_{jk}) - Ve_{ij}v_{jk}] \\ &\text{s.t.} \sum_{j=1}^{M}\sum_{k=1}^{K} b_{ijk}(t) \leq k_i\ \forall w_i \in \mathbb{W} \\ &\quad b_{ijk}(t) \in \{0, 1, ..., k_i\}, \forall w_i \in \mathbb{W}, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L} \\ &\quad \text{if } b_{ijk}(t) > 0, l_k \in R_i(l_{i_k}(t)), \forall w_i \in \mathbb{W}, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L}. \end{aligned} \tag{18}$$

Decisions of $b_{ijk}(t)$ for different workers are coupled together in underflow scenario, thus making it more arduous to solve

Eq. (18). Here we consider no-underflow constraint, and also present a detailed discussion about underflow situation in Section IV-C of technical report [20].

Now we address the problem in Eq. (18). For w_i, the problem becomes to decide $b_{ijk}(t)$ that minimizes $\sum_{j=1}^{M}\sum_{k=1}^{K} b_{ijk}(t)[Z_i(t)c_{ij} - (Q_{jk}(t) - \theta_{jk}) - Ve_{ij}v_{jk}]$. Since $Q_{jk}(t), Z_i(t)$ are known, OCP checks all task queues in $R_i(l_{i_k}(t))$, and finds $j^* \in \mathbb{M}, l_{k^*} \in R_i(l_{i_k}(t))$ that minimizes $Z_i(t)c_{ij} - (Q_{jk}(t) - \theta_{jk}) - Ve_{ij}v_{jk}$. Therefore, when $Z_i(t)c_{ij^*} - (Q_{j^*k^*}(t) - \theta_{j^*k^*}) - Ve_{ij^*}v_{j^*k^*} \geq 0$, let $b_{ij^*k^*}(t) = 0$. When $Z_i(t)c_{ij^*} - (Q_{j^*k^*}(t) - \theta_{j^*k^*}) - Ve_{ij^*}v_{j^*k^*} < 0$, assign $b_{ij^*k^*}(t) = k_i$. For any other $Q_{jk}(t)$, $b_{ijk}(t) = 0$. From another perspective, if we regard $-[Z_i(t)c_{ij} - (Q_{jk}(t) - \theta_{jk}) - Ve_{ij}v_{jk}]$ as the *regulated value*, OCP allocates all the worker resources to the task which can gain the maximum and positive regulated value. Worker-task assignment can be implemented concurrently for each w_i under no-underflow constraint.

c) Queue Update: After OCP, update $Q_{jk}(t)$ and $Z_i(t)$ according to Eq. (11) and Eq. (12), respectively.

4) Performance Analysis for OCP: We theoretically analyze the performance of the designed OCP in this part.

Theorem 1. *Suppose $Q_{jk}(0) = \theta_{jk}, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L}$, $Z_i(0) = 0, \forall w_i \in \mathbb{W}$, for any parameter $V > 0$, we have the following properties of OCP.*

a) The queue backlog of each task queue $Q_{jk}(t)$ for all t is bounded by:

$$0 \leq Q_{jk}(t) \leq \theta_{jk} + A_{jk}^{\max}. \tag{19}$$

b) The queue backlog of each cost queue $Z_i(t)$ for all t satisfies:

$$Z_i(t) \leq \max\{Z_i^{\max}, Z_i^{\max} + k_i \max_j c_{ij} - B_i\} \tag{20}$$

where $Z_i^{\max} = \max_{j,l_k} \frac{A_{jk}^{\max} + Ve_{ij}v_{jk}}{c_{ij}}$.

c) The no-underflow condition for every $Q_{jk}(t)$ is that perturbation parameter θ_{jk} satisfies:

$$\theta_{jk} \geq V(\max_{w_i} e_{ij})v_{jk} + 2\sum_{i=1}^{N} k_i, \tag{21}$$

d) Denote u^* as the optimal time average sensing value for ONCE-VM in Eq. (6). The time average sensing value achieved by OCP satisfies:

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}[u(t)] \geq u^* - \frac{D}{V}. \tag{22}$$

Proof: See Appendix C in technical report [20]. ∎

B. Online Crowdsensing of Fairness Maximization

Maximizing the sensing value only will lead to severe starvation for low value task queues with no workers allocated. Therefore, fair distribution of worker resources among different task queues is crucial for long-term online crowdsensing. To this end, we further extend value maximization to fairness maximization, which is also validated in Section V in terms of *coverage ratio*.

Proportional fairness is a common fairness metric, where the allocation of resources is required to be proportional to $\overline{u_{jk}}$ of each $Q_{jk}(t)$. Maximizing the logarithms $\log(\overline{u_{jk}})$ has been shown to satisfy proportional fairness [18]. Frequently, $\log(1+\beta\overline{u_{jk}})$ is utilized to enable the same function. Hence $f(\overline{u_{jk}}) = \log(1 + \beta\overline{u_{jk}})$ in Eq. (5) when we aim to maximize the time average proportional fairness. Consequently, ONCE-UM problem in Eq. (6) becomes *ONline CrowdsEnsing of Fairness Maximization (ONCE-FM)* problem with the objective $f(\overline{u}) = \lim_{T\to\infty}\sum_{j=1}^{M}\sum_{k=1}^{K}\log\{1 + \beta\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[u_{jk}(t)]\}$.

1) Auxiliary Variable: By introducing auxiliary variable $\gamma_{jk}(t)$ with $0 \leq \gamma_{jk}(t) \leq u_{jk}^{\max}$, we can transform maximizing logarithm function of the time average into maximizing the time average logarithm function. From Eq. (4), we have $\gamma_{jk}(t) \leq u_{jk}^{\max} \leq \sum_{i=1}^{N} k_i e_{ij} v_{jk}$. Jensen's inequality yields:

$$\begin{aligned} &\frac{1}{T}\sum_{t=0}^{T-1} f(\gamma_{jk}(t)) \leq f(\frac{1}{T}\sum_{t=0}^{T-1}\gamma_{jk}(t)), \\ &\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[f(\gamma_{jk}(t))] \leq f(\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{\gamma_{jk}(t)\}). \end{aligned} \tag{23}$$

According to [15], instead of solving ONCE-FM in Eq. (6), we handle the following problem:

$$\max \sum_{j=1}^{M}\sum_{k=1}^{K}\overline{f(\gamma_{jk})} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}[f(\gamma_{jk}(t))]$$

s.t. constraints in Eq. (6)

$$\overline{\gamma_{jk}} \leq \overline{u_{jk}}, 0 \leq \gamma_{jk}(t) \leq \sum_{i=1}^{N} k_i e_{ij} v_{jk}, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L}. \tag{24}$$

Eq. (24) implies that the transformed problem has an objective of the time average logarithm function, and more constraints for auxiliary variables compared to ONCE-FM.

2) Queue Dynamics: In addition to task queue $Q_{jk}(t)$ and cost queue $Z_i(t)$, there is one more (virtual) auxiliary queue $G_{jk}(t)$, which evolves obeying the following rule:

$$G_{jk}(t+1) = \max[G_{jk}(t) + \gamma_{jk}(t) - u_{jk}(t), 0]. \tag{25}$$

$G_{jk}(t)$ also should be stable. The dynamics of $Q_{jk}(t)$ and $Z_i(t)$ are the same as Eq. (11) and Eq. (12), respectively. Analogously, denote $\mathbf{G}(t) = \{G_{jk}(t) : \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L}\}$, and $\mathbf{\Theta}(t) = (\mathbf{Q}(t), \mathbf{Z}(t), \mathbf{G}(t))$.

3) Lyapunov Optimization: We still consider the no-underflow constraint in ONCE-FM. Define a perturbed Lyapunov function as:

$$\begin{aligned} L(\mathbf{\Theta}(t)) &= \frac{1}{2}||\mathbf{Q}(t) - \boldsymbol{\kappa}|| + \frac{1}{2}||\mathbf{Z}(t)|| + \frac{1}{2}||\mathbf{G}(t)|| \\ &= \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{K}(Q_{jk}(t) - \kappa_{jk})^2 + \frac{1}{2}\sum_{i=1}^{N} Z_i^2(t) + \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{K} G_{jk}^2(t), \end{aligned} \tag{26}$$

where κ_{jk} is a perturbation parameter. Hence, we get Lyapunov drift: $\Delta(\mathbf{\Theta}(t)) = \mathbb{E}[L(\mathbf{\Theta}(t+1)) - L(\mathbf{\Theta}(t))|\mathbf{\Theta}(t)]$, and drift-minus-utility: $\Delta(\mathbf{\Theta}(t)) - V\sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}[f(\gamma_{jk}(t))|\mathbf{\Theta}(t)]$.

Lemma 3. *drift-minus-utility satisfies:*

$$\Delta(\boldsymbol{\Theta}(t)) - V\sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}[f(\gamma_{jk}(t))|\boldsymbol{\Theta}(t)] \le D'$$
$$- \sum_{i=1}^{N}\mathbb{E}[Z_i(t)B_i|\boldsymbol{\Theta}(t)] + \sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}[a_{jk}(t)(Q_{jk}(t)-\kappa_{jk})|\boldsymbol{\Theta}(t)]$$
$$+ \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}\{b_{ijk}(t)[Z_i(t)c_{ij} - (Q_{jk}(t)-\kappa_{jk})]|\boldsymbol{\Theta}(t)\}$$
$$- \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}[b_{ijk}(t)e_{ij}v_{jk}G_{jk}(t)|\boldsymbol{\Theta}(t)]$$
$$+ \sum_{j=1}^{M}\sum_{k=1}^{K}\mathbb{E}[\gamma_{jk}(t)G_{jk}(t) - Vf(\gamma_{jk}(t))|\boldsymbol{\Theta}(t)]. \tag{27}$$

where $D' = \frac{1}{2}\sum_{j=1}^{M}\sum_{k=1}^{K}\{\max[(A_{jk}^{\max})^2, (\sum_{i=1}^{N}k_i)^2] + (\sum_{i=1}^{N}k_ie_{ij}v_{jk})^2\} + \frac{1}{2}\sum_{i=1}^{N}\max[B_i^2, (k_i\max_j c_{ij})^2]$.

Proof: See Appendix D in technical report [20]. ∎

4) Fair Online Control Policy: We design a *Fair Online Control Policy (FOCP)* to **minimize the upper bound of drift-minus-utility** in Eq. (27). In each time slot, with observed queues $Q_{jk}(t), Z_i(t), G_{jk}(t)$, FOCP determines $a_{jk}(t), b_{ijk}(t), \gamma_{jk}(t)$.

a) Fair Auxiliary Variable Decision: To compute $\gamma_{jk}(t)$, FOCP solves the following problem:

$$\begin{aligned} &\min\ \gamma_{jk}(t)G_{jk}(t) - Vf(\gamma_{jk}(t)) \\ &\text{s.t.}\ \ 0 \le \gamma_{jk}(t) \le \sum_{i=1}^{N}k_ie_{ij}v_{jk}, \end{aligned} \tag{28}$$

where $f(\gamma_{jk}(t)) = \log(1+\beta\gamma_{jk}(t))$. Let $g(\gamma_{jk}(t)) = \gamma_{jk}(t)G_{jk}(t) - V\log(1+\beta\gamma_{jk}(t))$ and derive g with respect to $\gamma_{jk}(t)$, we have:

$$g'(\gamma_{jk}(t)) = G_{jk}(t) - \frac{V\beta}{1+\beta\gamma_{jk}(t)}. \tag{29}$$

If $g'(\gamma_{jk}(t)) = 0$, then $\gamma_{jk}(t) = \frac{V}{G_{jk}(t)} - \frac{1}{\beta}$. Thus, FOCP decides $\gamma_{jk}(t)$ in line with the following rule:

$$\gamma_{jk}(t) = \begin{cases} 0 & \text{if } G_{jk}(t) > V\beta \\ \dfrac{V}{G_{jk}(t)} - \dfrac{1}{\beta} & \text{if } G_{jk}(t) \in [\dfrac{V\beta}{1+\beta\sum_{i=1}^{N}k_ie_{ij}v_{jk}}, V\beta] \\ \displaystyle\sum_{i=1}^{N}k_ie_{ij}v_{jk} & \text{if } G_{jk}(t) < \dfrac{V\beta}{1+\beta\sum_{i=1}^{N}k_ie_{ij}v_{jk}}. \end{cases} \tag{30}$$

b) Fair Task Admission Control: Similar to the task admission control in OCP, FOCP admits $a_{jk}(t)$ into $Q_{jk}(t)$ following a simple threshold rule:

$$a_{jk}(t) = \begin{cases} 0 & \text{if } Q_{jk}(t) \ge \kappa_{jk} \\ A_{jk}(t) & \text{if } Q_{jk}(t) < \kappa_{jk}. \end{cases} \tag{31}$$

c) Fair Worker-Task Assignment: In fair worker-task assignment, FOCP decides $b_{ijk}(t)$ for each worker w_i with respect to task queue $Q_{jk}(t)$ covered by $R_i(l_{i_k}(t))$. The decision is through solving the problem below:

$$\begin{aligned} &\min \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}b_{ijk}(t)[Z_i(t)c_{ij} - (Q_{jk}(t)-\kappa_{jk}) - e_{ij}v_{jk}G_{jk}(t)] \\ &\text{s.t.}\ \sum_{j=1}^{M}\sum_{k=1}^{K}b_{ijk}(t) \le k_i\ \forall w_i \in \mathbb{W} \\ &\quad b_{ijk}(t) \in \{0,1,...,k_i\}, \forall w_i \in \mathbb{W}, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L} \\ &\quad \text{if } b_{ijk}(t) > 0, l_k \in R_i(l_{i_k}(t)), \forall w_i \in \mathbb{W}, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L}. \end{aligned} \tag{32}$$

Under no-underflow constraint, the computation for $b_{ijk}(t)$ is effective. Homoplastically, FOCP traverses all task queues in $R_i(l_{w_k}(t))$, and finds $j' \in \mathbb{M}, l_{k'} \in R_i(l_{i_k}(t))$ that minimizes $Z_i(t)c_{ij} - (Q_{jk}(t)-\kappa_{jk}) - e_{ij}v_{jk}G_{jk}(t)$. If $Z_i(t)c_{ij'} - (Q_{j'k'}(t)-\kappa_{j'k'}) - e_{ij'}v_{j'k'}G_{j'k'}(t) < 0$, let $b_{ij'k'}(t) = k_i$, otherwise $b_{ij'k'}(t) = 0$. As for any other $j \in \mathbb{M}, l_k \in \mathbb{L}$, let $b_{ijk}(t) = 0$. The underflow situation is also discussed in technical report [20] Section IV-C.

d) Queue Update: After FOCP, queues $Q_{jk}(t)$, $Z_i(t)$ and $G_{jk}(t)$ are updated according to Eq. (11), Eq. (12) and Eq. (25), respectively.

5) Performance Analysis for FOCP: We will illustrate the performance of FOCP in the following theorem.

Theorem 2. *Suppose* $Q_{jk}(0) = \kappa_{jk}, G_{jk}(0) = 0, \forall j \in \mathbb{M}, \forall l_k \in \mathbb{L}$, $Z_i(0) = 0, \forall w_i \in \mathbb{W}$, *for any parameter* $V > 0$, *we have the following properties for FOCP.*

a) The queue backlog of each task queue $Q_{jk}(t)$ *for all* t *is bounded by:*

$$0 \le Q_{jk}(t) \le \kappa_{jk} + A_{jk}^{\max}. \tag{33}$$

b) The queue backlog of each auxiliary queue $G_{jk}(t)$ *satisfies* $G_{jk}(t) \le G_{jk}^{\max}$ *for all* t, *where:*

$$G_{jk}^{\max} = \max[V\beta, \frac{V\beta}{1+\beta\sum_{i=1}^{N}k_ie_{ij}v_{jk}} + \sum_{i=1}^{N}k_ie_{ij}v_{jk}]. \tag{34}$$

c) The queue backlog of each cost queue $Z_i(t)$ *for all* t *is bounded by:*

$$Z_i(t) \le \max\{ZG_i^{\max}, ZG_i^{\max} + k_i\max_{j\in\mathbb{M}}c_{ij} - B_i\} \tag{35}$$

where $ZG_i^{\max} = \max_{j,l_k}\frac{A_{jk}^{\max} + e_{ij}v_{jk}G_{jk}^{\max}}{c_{ij}}$, *and* $G_{jk}^{\max}$ *is defined in Eq.* (34).

d) For any perturbation parameter κ_{jk}, *if it satisfies:*

$$\kappa_{jk} \ge (\max_{w_i} e_{ij})v_{jk}G_{jk}^{\max} + 2\sum_{i=1}^{N}k_i, \tag{36}$$

with $G_{jk}^{\max}$ *defined in Eq.* (34), *task queue* $Q_{jk}(t)$ *will not suffer from underflow.*

e) Denote $f(u')$ *as the optimal time average fairness for ONCE-FM in Eq.* (6). *The time average fairness produced by FOCP satisfies:*

$$\sum_{j=1}^{M}\sum_{k=1}^{K}f(\overline{u_{jk}}) \ge f(u') - \frac{D'}{V}, \tag{37}$$

where $\overline{u_{jk}} = \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[u_{jk}(t)]$.

Proof: See Appendix E in technical report [20]. ∎

V. Performance Evaluation

In this section, we evaluate the performance of the crowdsensing system, as well as comparing the results of the proposed methods to baseline methods.

A. Data Set and Parameter Settings

We leverage a real data set which includes contributors' trajectories in a campus [19] composed by over 130 thousand location records. We divide the area into $300m \times 300m$ small grid regions. After that, we filter out abnormal contributors' trajectories, and obtain trajectories in $K = 86$ regions.

First calculate the location diversity TLD and WLD. Let diversity value $dv_k = \frac{1}{\text{TLD}_{l_k}+1}$ and diversity cost $dc_i = \frac{1}{\text{WLD}_{w_i}+1}$. The maximum task arrivals $A_{jk}^{\max}$ is proportional to TLD_{l_k}. ov_j uniformly distributes in $(1, 3)$, and oc_i is uniformly sampled in $(0.2, 0.4)$. Budget B_i is randomly valued in $(1.2, 1.5)$, while constraint of limited time slot length k_i is a random integer in $[4, 6]$. Qualified expertise $\mathbf{E}_i$ takes values in $(0.5, 1)$. Sensing range $R_i(l_{i_k})$ covers 9 regions including the located region and the surrounding 8 regions. Two mobility models are considered: *Random Walk (RW)*-workers randomly stay in one of the covered 9 regions in next time slot; *Real Trace (RT)*-workers follow the traces in the real data set. We compare our proposed algorithms with two baseline algorithms:

- *Random Algorithm (RA)*: Workers undertake tasks randomly under the constraints in offline crowdsensing and online crowdsensing, respectively.
- *Greedy Algorithm (GA)*: Workers myopically undertake the maximum value tasks under the constraints in offline crowdsensing and online crowdsensing, respectively.

B. Results for Offline Crowdsensing

The number of tasks for each type is uniformly in $[20, 30]$. Let task type $M = 5$ with all tasks randomly distributed in the managed area, we obtain the sensing values with respect to the number of workers N in Fig. 2. We can see that OCA outperforms RA and GA, since OCA optimally assign tasks to each worker in Eq. (8). All the sensing value tends to increase over N due to more tasks completed by the increased workers. One interesting phenomenon is that RA is better than GA. Because workers are prone to selecting the same high value tasks in GA, but a task is only assigned to one worker, thus causing a severe waste of worker resources.

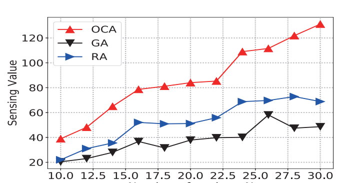

Fig. 2. Sensing value vs. number of workers N

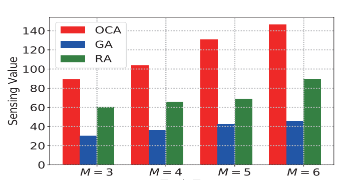

Fig. 3. Sensing value vs. task type M

When $N = 10$ and task type M varies from 3 to 6, the impacts of M are drawn in Fig. 3 which shows OCA has a higher sensing value than GA and RA. Besides, the sensing value increases with M because more tasks are available to be completed.

C. Results for Online Crowdsensing

We evaluate the performance for overall 2000 time slots, which can fully depict the system characteristics.

1) Time Average Sensing Value: First let task type $M = 5$, importance weight $V = 30$. Vary N from 10 to 30 with an interval 2, the sensing values in RW and RT are demonstrated in Fig. 4. We can see that the time average sensing values of designed OCP and FOCP, as well as baselines GA and RA increase with N since more tasks are undertaken by workers. Besides, the performance order is OCP $>$ FOCP $>$ GA $>$ RA. GA is better than RA due to no-underflow constraint which can fully utilize worker resources.

We draw the time average sensing value when task type M changes, with $N = 10, V = 30$ in Fig. 5. The bars with dotted black edges denote the results in RW, while the bars below signify the results in RT. It shows that the time average sensing value rises with M due to more diverse task values. The performance order is OCP $>$ FCOP $>$ GA $>$ RA as well.

Finally, we provide how importance weight V influences the time average sensing value. Let $N = 10, M = 5$, and vary V from 30 to 100. The results in Fig. 6 show that the time average sensing value increases with V for OCP and FOCP, but fluctuates for GA and RA. The reason is that V only affects the "weight" on utility maximization in OCP and FOCP. Besides, OCP and FOCP go beyond GA and RA.

2) Fairness Demonstration: Coverage ratio denotes how many task queues are allocated with workers during the 2000 time slots. If coverage ratio is low, only a small portion of task queues are allocated with workers, which implies that worker resources are unfairly distributed. Let $M = 5, V = 30$, we provide coverage ratio influenced by N in Fig. 7. It illustrates that FOCP is much fairer than OCP. Besides, FOCP can frequently achieve full task coverage (coverage ratio is 1). Thus, FOCP sacrifices the sensing value (**4%-16%** with average **12%**) for the fairness improvement (**80%-162%** with average **116%**), since worker's fair allocation means more attention paid to low value tasks. The impact of V is also investigated in Fig. 8 when $N = 10, M = 5$. Intuitively, FOCP still far outperforms OCP. Furthermore, coverage ratio in RW is a bit higher than that in RT because workers in RW mobility can cover wider regions.

3) Queue Stability: We illustrate typical backlogs of task queue, cost queue and auxiliary queue in Figs. 9-11, which show that all the queues will not go infinity, that is queues maintain stability.

VI. Related Work

The emerging crowdsensing paradigm has proliferated a broad range of mobile applications [1]-[4], where the fundamental issue is to determine how workers undertaking tasks

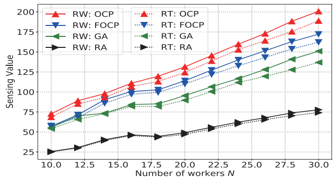

Fig. 4. Time average sensing value vs. number of workers N

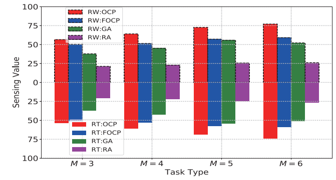

Fig. 5. Time average sensing value vs. task type M

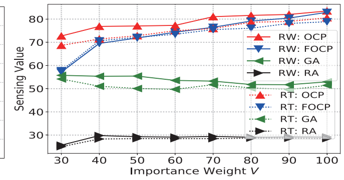

Fig. 6. Time average sensing value vs. importance weight V

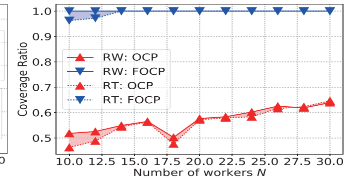

Fig. 7. Coverage ratio (fairness) vs. number of workers N

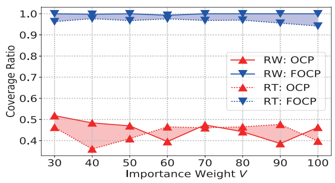

Fig. 8. Coverage ratio (fairness) vs. importance weight V

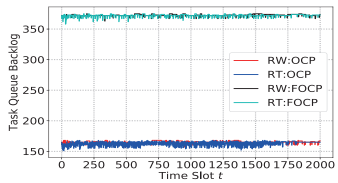

Fig. 9. Task queue backlog vs. time slot t

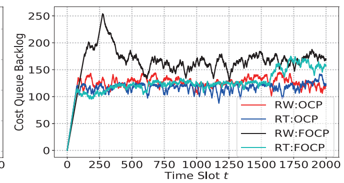

Fig. 10. Cost queue backlog vs. time slot t

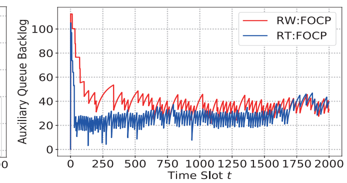

Fig. 11. Auxiliary queue backlog vs. time slot t

based on the locations. Besides, diversity issue is also an important factor in crowdsensing [9].

Most previous works are devoted to offline crowdsensing. Cheng *et al.* consider a spatial crowdsensing system, which aims to maximize the reliability and diversity of enlisted workers, however only the worker's diversity is considered [5]. Kazemi *et al.* consider a simple location-aware and diversity based crowdsourcing system, but only location diversity of tasks is studied [6].

There are also a few studies for online crowdsensing. Han *et al.* propose an online policy to maximize the sensing value [12], but location awareness and worker mobility are neglected. Gao *et al.* design an online incentive to maintain adequate workers [13], but they do not decide how to assign tasks. Moreover, hardly any works have considered the fair allocation of worker resources in online crowdsensing.

We provide a comprehensive analysis for the crowdsensing system in this paper. An entropy based location diversity is used for both workers and tasks [7]. Our main efforts lie in the online crowdsensing, where we harness Lyapunov optimization [15] to handle the system dynamics, and further ensure the proportional fairness [18] of worker allocation.

VII. Conclusion

In this paper, we study the location-aware and location diversity based offline and online crowdsensing systems. We first investigate offline crowdsensing, where a combinatorial algorithm is proposed to assign tasks to workers. After that, we mainly consider online crowdsensing with dynamic workers and tasks. We innovatively implement Lyapunov optimization to handle the stochastic characteristics, and further take into account the fair allocation of worker resources. Both theoretical analysis and performance evaluation on real data set demonstrate the system efficiency.

Acknowledgement

This work was supported by NSF China (No. 61325012, 61532012, 61672342, 61572319, 61671478, U1405251) and Science and Technology Innovation Program of Shanghai (Grant 17511105103).

References

[1] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "SmartRoad: Smartphone-Based Crowd Sensing for Traffic Regulator Detection and Identification," *ACM TOSN*, vol. 11, no. 4, 2015.

[2] M. Elhamshary, M. Youssef, A. Uchiyama, H. Yamaguchi, and T. Higashino, "TransitLabel: A Crowd-Sensing System for Automatic Labeling of Transit Stations Semantics," in *Proc. ACM MobiSys*, 2016.

[3] Y. Cheng, X. Li, Z. Li, S. Jiang, Y. Li, J. Jia, and X. Jiang, "Aircloud: A cloud-based air-quality monitoring system for everyone," in *Proc. ACM SenSys*, 2014, pp. 251-265.

[4] 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 *Proc. ACM IPSN*, 2010, pp. 105-116.

[5] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, J. Han, and J. Zhao, "Reliable Diversity-Based Spatial Crowdsourcing by Moving Workers," in *Proc. ACM VLDB*, vol. 8, no. 10, pp. 1022-1033, 2015.

[6] L. Kazemi, and C. Shahabi, "GeoCrowd: Enabling Query Answering with Spatial Crowdsourcing," in *Proc. ACM SIGSPATIAL*, 2012, pp. 189-198.

[7] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, "Bridging the Gap Between Physical Location and Online Social Networks," in *Proc. ACM UbiComp*, 2010, pp. 119-128.

[8] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "TRAC: Truthful auction for location-aware collaborative sensing in mobile crowd-sourcing," in *Proc. IEEE INFOCOM*, 2014.

[9] T. Wu, L. Chen, P. Hui, C. J. Zhang, and W. Li, "Hear the Whole Story: Towards the Diversity of Opinion in Crowdsourcing Markets," in *Proc. ACM VLDB*, vol. 8, no. 5, pp. 485-496, 2015.

[10] 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.

[11] H. Jin, L. Su, and K. Nahrstedt, "CENTURION: Incentivizing Multi-Requester Mobile Crowd Sensing," in *Proc. IEEE INFORCOM*, 2017.

[12] Y. Han, and Y. Zhu, "Profit-Maximizing Stochastic Control for Mobile Crowd Sensing Platforms," in *Proc. IEEE MASS*, 2014.

[13] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *Proc. IEEE INFOCOM*, 2015.

[14] R. Cohen, L. Katzir, and D. Raz, "An Efficient Approximation for the Generalized Assignment Problem," *Information Processing Letters*, vol. 100, no. 4, pp. 162-166, 2006.

[15] M. J. Neely, "Stochastic Network Optimization with Application to Communication and Queueing Systems," *Morgan & Claypool*, 2010.

[16] L. Huang, and M. J. Neely, "Utility optimal scheduling in processing networks," *Performance Evaluation*, vol. 68, no. 11, pp. 1002-1021, 2011.

[17] L. Huang, "The Value-of-Information in Matching With Queues," *IEEE/ACM TON*, vol. 25, no. 1, pp. 29-42, 2017.

[18] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33-37, 1997.

[19] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "CRAWDAD dataset ncsu/mobilitymodels (v. 2009-07-23)," http://crawdad.org/ncsu/mobilitymodels/20090723.

[20] "Technical Report," [Online]. Avaliable: https://1drv.ms/b/s!AucSx6OjIO3thQqsRH5O4daiuBdH.