A Fair Task Assignment Strategy for Minimizing Cost in Mobile Crowdsensing

Yujun Liu†, Yongjian Yang†, En Wang†*, Wenbin Liu†, Dongming Luan†, Xiaoying Sun†, Jie Wu‡
†Jilin University, Changchun, China
‡Temple University, Philadelphia, USA
*Corresponding Author
{yujun18, luandm17, liuwb16}@mails.jlu.edu.cn, {yyj, wangen, sunxy}@jlu.edu.cn, jiewu@temple.edu

***Abstract*—Mobile CrowdSensing (MCS) is a promising paradigm that recruits mobile users to cooperatively perform various sensing tasks. When assigning tasks to users, most existing works only consider the fairness of users, i.e., the user's processing ability, with the goal of minimizing the assignment cost. However, in this paper, we argue that it is necessary to not only give full use of all the users' ability to process the tasks (e.g., not exceeding the maximum capacity of each user while also not letting any user idle too long), but also satisfy the assignment frequency of all corresponding tasks (e.g., how many times each task should be assigned within the whole system time) to ensure a long-term, double-fair and stable participatory sensing system. Hence, to solve the task assignment problem which aims to reasonably assign tasks to users with limited task processing ability while ensuring the assignment frequency, we first model the two fairness constraints simultaneously by converting them to user processing queues and task virtual queues, respectively. Then, we propose a *Fair Task Assignment Strategy* (FTAS) utilizing Lyapunov optimization and we provide the proof of the optimality for the proposed assignment strategy to ensure that there is an upper bound to the total assignment cost and queue backlog. Finally, extensive simulations have been conducted over three real-life mobility traces: *Changchun/taxi*, *Epfl/mobility*, and *Feeder*. The simulation results prove that the proposed strategy can achieve a trade-off between the objective of minimizing the cost and the fairness of tasks and users compared with other baseline approaches.**

***Keywords*-Task assignment, Lyapunov optimization, Fairness, Mobile CrowdSensing**

I. INTRODUCTION

With the rapid development of smart devices embedded with various sensors, Mobile CrowdSensing (MCS) [1] has become a hot sensing paradigm in recent years and has facilitated large-scale data collection [2]. In MCS, the mobile users with smart devices (e.g., smartphones) [3] can ubiquitously collect information and perform various sensing tasks, such as monitoring traffic, air quality, and wireless signal strengths [4] [5] [6]. In most cases, in order to complete more tasks with less cost, we should assign tasks to suitable users, which raises the fundamental task assignment problem in MCS.

So far, there has been a large amount of research [7] [8] [9] devoted to task assignment with the optimization objective of maximizing the amount of assigned tasks or minimizing assignment cost. Obviously, assigning more tasks leads to a higher assignment cost. Hence, some studies focus on making a trade-off between assignment cost and the amount of assigned tasks [10] [11] [12]. However, the task assignment approaches in these studies only pay attention to total assignment cost and the total amount of assigned tasks, and do not consider each task's required assignment frequency or each user's processing ability, which we regard is unfair to both the task and the user. In other words, the user's processing ability is usually limited which means that too many tasks assigned to him in a certain period may cause overload and of course is unfair to the user. On the other hand, the large number of assigned tasks does not indicate that each task has good assignment frequency, which means that these approaches also cannot guarantee the fairness of the task. In order to address the above problems, some researches consider the fairness in task assignment [13] [14] [15] [16], however, they only consider the fairness of users such as cost-fair and participation-fair and ignore that of tasks.

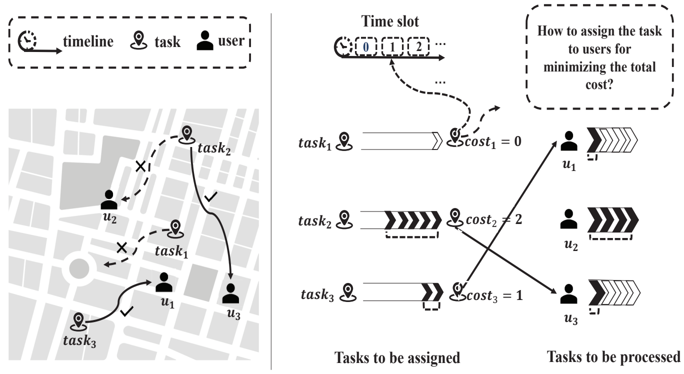


Fig. 1: An example to illustrate the task assignment problem with the purpose of minimizing total assignment cost while taking both task's and user's fairness into consideration.

Consider a crowdsensing scenario as shown in Fig. 1. The platform publishes some location-based tasks and we regard the distance between user and task as the cost of completing a task. The whole system time is divided into t time slots, and each task needs to be assigned a certain number of times within the whole system time in order to collect enough data to satisfy each task's assignment frequency. At the same time, in a time slot, each user can be assigned a limited number of tasks, which cannot exceed user's processing ability. In order to minimize the assignment cost, an easy idea is that we could assign tasks (such as $task_1$) which are nearest to users in each time slot. However, in this way, some tasks may be assigned too many times, while the other tasks have no chance to be assigned, which breaks the fairness of tasks. Similarly, some users may be continuously assigned tasks and overloaded soon. This also breaks the fairness of users. Fig. 1 provides an example for the fairness of users and tasks, instead of assigning $task_1$, we first assign $task_2$ and $task_3$ because they have more tasks in the assignment queue (task fairness). When assigning $task_2$, we prefer to assign it to $user_3$ rather than $user_2$, even though $user_2$ is closer to $task_2$ than $user_3$. This is because that the number of tasks assigned to $user_2$ almost exceeds his processing ability (user fairness). Above all, the problem to be addressed in this paper is to propose a fair task assignment strategy in each time slot with the purpose of minimizing the total assignment cost while taking both the user and task fairness into consideration.

To deal with the proposed fair task assignment problem, we need to address the following challenges. Firstly, considering the objective of minimizing the total assignment cost, it is hard to *model the fairness of users and tasks simultaneously*. Second, our problem is actually a *three-objective optimization problem*, i.e., a trade-off between the assignment cost minimization and the fairness of users and tasks. It is not easy to find a solution to minimize the cost while maintaining fairness. Furthermore, it is more challenging to *find a bound* for the three-objective optimization problem between our approach and the optimal solution.

In this paper, to model the two fairness constraints simultaneously while minimizing the cost, we convert the fairness of users and tasks into user processing queues and task virtual queues, respectively. Then we change the fairness problem of assigning the tasks to users with fairness into assigning the tasks while maintaining the double queues stability. Thus, we formulate a *Fair Task Assignment Problem* (FTAP), which aims to minimize the cost and stabilize the double queues. Then, to solve the above three-objective optimization problem, we introduce Lyapunov optimization technical, which can make control decisions in a dynamic system with no prior knowledge of the platform and can achieve the objective while maintaining the queue stability. We jointly consider the double queues by expanding the traditional single queue based on original optimization model and propose a *Fair Task Assignment Strategy* (FTAS) to achieve a trade-off between minimizing the total assignment cost while maintaining the queue stability. Finally, to show the gap between our approach and the optimal solution, we achieve an upper bound between them through a rigorous mathematical proof, which also illustrates that the approach is effective at minimizing the assignment cost as well as maintaining the queue stability.

Our contributions can be summarized as follows:

- We model the two fairness constraints of users and tasks simultaneously through converting them to user processing queues and task virtual queues, respectively. We formulate a *Fair Task Assignment Problem* (FTAP) which aims at minimizing the objective of assignment cost and maintaining the double queues stability simultaneously. As far as we know, this is the first work taking both fairness of users and tasks into consideration.
- We propose a *Fair Task Assignment Strategy* (FTAS) utilizing Lyapunov optimization to achieve a trade-off between minimizing the assignment cost and maintaining the queue stability by jointly addressing the double queues.
- We analyze the performance of the proposed fair task assignment strategy. We also find the upper bound of the total cost and queue backlog.
- Extensive simulations are conducted on three realistic data sets. Compared with the baseline approaches, the proposed approach is proven to achieve cost minimization while maintaining queue stability.

The remainder of this paper is organized as follows: in Section II, we review some related work. We introduce the system model and formulate the problem in Section III. In Section IV, we design a Fair Task Assignment Strategy (FTAS) to address the problem with the objective of minimizing the assignment cost. Section V illustrates the performance of the proposed strategy and simulations are evaluated in Section VI. Finally, a brief conclusion is given in Section VII.

II. RELATED WORK

A. Fairness in mobile crowdsensing

There has been some effort to solve the fairness problem for task assignment in mobile crowdsensing and mobile social networks. Peng *et al.* [13] presented a novel fair energy-efficient assignment framework which focused on both energy efficiency and fairness by making one control decision in each time slot. Sun *et al.* [14] concentrated on the cost-fair task assignment (CFA) problem in nondeterministic MCS scenarios, and designed two algorithms to balance the sensing cost of users and satisfy the data reliability requirement. In order to provide the long-term participation incentive, Gao *et al.* [15] proposed a Lyapunov-based VCG auction policy for the on-line sensor selection. Sooksatra *et al.* [16] formulated the sustainability problem as an optimization problem maximizing providers proportionally fair utilities with respect to their multi-dimensional fairness factors, and designed a fairness-aware auction mechanism to incentive sensory-data providers. However, these studies only consider unilateral fairness, either the platform or users. Different from the above research, we consider the fairness for both users and tasks. Furthermore, we also utilize Lyapunov optimization technique in a novel way in which we deal with the problem of fair task assignment.

B. Lyapunov-optimization-based resource assignment strategy

In recent years, there has been much research which aims at solving resource assignment problems by utilizing Lyapunov optimization. Combining Lyapunov optimization technique with the weight perturbation, Fang *et al.* [17] introduced a stochastic control algorithm to achieve both profit optimality and system stability. Zhang *et al.* [18] proposed a Top-Down

TABLE I: Main Notations Throughout The Paper

Symbol	Meaning
S, U	the sets of tasks and users, respectively.
k	the number of tasks.
u	the number of users.
t	the t-th time slot.
$P_i(t), Q_j(t)$	the task virtual queue backlog of task s_i and the user processing queue backlog of user u_j in time slot t.
s_i, u_j	the i-th task , the j-th user.
Y_i	invalid threshold for s_i.
$x_{ij}[t]$	whether to assign task s_i to user u_j.
$d_j(t)$	the processing ability of user u_j.
$\mathbf{Z}(t)$	the vector of all queue backlogs in time slot t.
$c_{ij}(t)$	the distance between the location of task s_i and mobile user u_i in time slot t.
$u(t)$	the total assignment cost quantified by distance in time slot t.
$L(t)$	Lyapunov function to measure the size of the joint queue backlog.
$\Delta L(t)$	the one-slot conditional Lyapunov drift function.

Optimal Control Module (TDOC) based on Lyapunov optimization to address the task assignment problem for Quality of Service (QoS) objective of the application in edge computing systems. Tian *et al*. [19] used a Lyapunov optimization framework for their online control policy in crowdsensing system to optimize the time average sensing utility as well as system stability. Yao *et al*. [20] proposed *a two time scale* control algorithm based on Lyapunov optimization, which aims at reducing power cost and optimizing the trade-off between the power cost and delay in geographically distributed data centers. Liu *et al*. [21] utilized a schedule algorithm based on Lyapunov drift -plus-penalty framework to maximize the system throughput of the secondary user (SU), while satisfying different kinds of constraints to be involved under the time-varying channel and traffic conditions. To solve the problem of mobile users' workload offloading, Liu *et al*. [22] proposed a Lyapunov optimization framework to maximize the offloading utility while ensuring the queue stability. Fang *et al*. [23] designed an online control algorithm based on Lyapunov optimization framework to optimize the trade-off between system throughput and energy consumption. Han *et al*. [24] proposed a profit maximizing algorithm for crowdsensing platforms by utilizing Lyapunov optimization technique. It achieves a time average profit which is arbitrarily close to the optimum and maintains strong stability. The above research either only considered just one queue, or even if there were double queues, the control strategy was made independently for each queue. Different from the above research, we consider double queues, i.e., the user processing queue and the task virtual queue. We make one control decision in each time slot to simultaneously keep both queues stable.

III. SYSTEM OVERVIEW AND PROBLEM FORMULATION

A. System overview

First of all, we consider a system model in mobile crowdsensing which aims at task assignment and also takes the

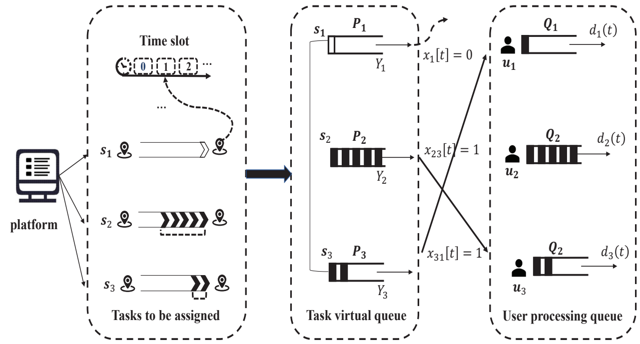


Fig. 2: An example of the system model in our fair MCS scenario.

concept of fairness into consideration. We assume that the platform produces a set of tasks which are denoted by $S = \{s_1, s_2, \ldots, s_k\}$, and these tasks will be geographically mapped into different locations and there are a set of users with mobile devices $U = \{u_1, u_2, \ldots, u_n\}$.

As shown in the left part of Fig. 2, we divide the whole system into t time slots, $t \in \{0, 1, 2, \ldots\}$. In the system, the platform continuously generates tasks and each task can only be assigned to one user in each time slot t. We let $x_{ij}[t]$ denote whether to assign task s_i to user u_j in time slot t, if the task is assigned, $x_{ij}[t] = 1$, otherwise 0. $c_{ij}(t)$ denotes the cost that user u_j processes task s_i which is quantified by the distance between user u_j's location and task s_i's location. Notably, the cost can be measured in various ways, such as the length of time a user takes to process a task, user's expertise for a task, etc. We just use the distance between tasks and users as a measurement for simplicity to prove the universality of our theory. First, we define the total cost of the task assignment for mobile users in time slot t as follows:

$$u(t) = \sum_{i=1}^{k} \sum_{j=1}^{n} c_{ij}(t) \cdot x_{ij}[t]. \tag{1}$$

As discussed above, we concentrate on the fairness of tasks and users. For tasks, with the objective of minimizing the assignment cost, a task may suffer from the scenario where it is not assigned in time slot t because of its high cost. If this scenario continues, the tasks's assignment frequency will not be satisfied and it thus causes unfairness. In this paper, we use an indicator called assignment probability to reflet the task assignment frequency, which is the probability of each task being assigned. Specifically, there exists a threshold Y_i called the *invalid threshold* of task s_i. If task s_i's assignment probability is smaller than Y_i, we consider the collected data of task s_i is invalid. Therefore, to ensure the assignment frequency of each task, the assignment probability of each task should be no smaller than the *invalid threshold*, which is described as follows:

$$Y_i \leq D_i(\mathbf{X}_i) = \frac{1}{T} \sum_{t \in T} \sum_{j=1}^{n} x_{ij}[t], \forall i \in k, \tag{2}$$

where $x_{ij}[t] \in \{1, 0\}$ and $\mathbf{X}_i$ represents the whole assignment distribution of task s_i in all time slots and $D_i(\mathbf{X}_i)$ denotes the time average assignment probability of task s_i.

For users, as shown in the right part of Fig. 2, we use $Q_j(t)$ to denote the processing queue backlog of user u_j in time slot t. The queue dynamic is defined as follows:

$$Q_j(t+1) = max[Q_j(t) - d_j(t), 0] + \sum_{i=1}^{k} x_{ij}[t], \quad (3)$$

where $d_j(t)$ denotes the processing ability of user u_j in time slot t, which is the amount of tasks data that the user u_j can deal with in time slot t. It is worth noting that although the number of processed tasks in time slot t for each user may be different, due to the computing restriction of the same devices and the assumption that each task's amount of data is the same, we consider $d_j(t)$ is a fixed value for each user. As $d_j(t)$ is limited, there exists d_{max} so that $d_j(t) \leq d_{max}$ for all users and all time slots. $\sum_{i=1}^{k} x_{ij}[t]$ denotes the amount of tasks assigned to user u_j in time slot t. By the queue stability theorem [25], the queue $Q_j(t)$ is stable only if the amount of assigned tasks of user u_j is less than or equal to his processing ability in each time slot. This establishes an equivalence between the constraint of users' fairness and the user queue stability. Thus, to maintain the fairness of users, we need to ensure that the corresponding user processing queue is stable under the proposed strategy. The main notations used throughout this paper are illustrated in Table I.

B. Conversion of assignment frequency

As Lyapunov optimization is a widely used technique for solving stochastic optimization problems with time average constraints, we can utilize it to solve the problem in this paper, i.e., the assignment cost minimization problem is a stochastic optimization problem and the task assignment frequency constraint is a time average constraint. Therefore, similar to the user processing queue, we use Lyapunov optimization technique to convert the constraint of task assignment frequency into the *task virtual queue*. Furthermore, by maintaining the stability of the task virtual queue, the constraint of task assignment frequency is satisfied.

Firstly, we introduce the *virtual queue* P_i for task s_i. As shown in the middle part of Fig. 2, the virtual queue is used to buffer the virtual assignment request of each task. Here, the virtual request is not actually initiated by tasks, it is used to represent the requirement of the task assignment frequency constraint. That is to say, one virtual request represents that "to satisfy the constraint of task assignment frequency, the task should be assigned in one additional time slot". Thus, the backlog of a task virtual queue denotes the total amount of virtual requests in the queue (which may not actually be an integer), which is also the total number of additional time slots that the task should be assigned to users.

As mentioned above, each virtual request of task s_i will come into the queue with a constant arrival rate of Y_i. We let $x[i]$ denote whether task s_i is assigned or not in time slot t, and its value is either 1 or 0. As previously designed, the average assignment probability of task s_i is $D_i(\mathbf{X}_i) = \frac{1}{T}\sum_{t\in T}\sum_{j=1}^{n} x_{ij}[t]$. Once a request of task s_i leaves the queue in time slot t, $x[i] = 1$, otherwise, $x[i] = 0$ and we let $D_i(\mathbf{X}_i)$ denote the average departure rate.

Therefore, we convert the task assignment frequency constraint into a task virtual queue based on Lyapunov optimization technique and let $P_i(t)$ denote the queue backlog of task s_i in time slot t. For task s_i, we combine the virtual request (also called arrival rate) and the result of assignment (departure) in each time slot t, so we have the following task virtual queue dynamic:

$$P_i(t+1) = max[P_i(t) - \sum_{j=1}^{n} x_{ij}[t], 0] + Y_i. \quad (4)$$

Similar to the user processing queue, the queue $P_i(t)$ is stable only if the task arrival rate is less than or equal to the departure rate, i.e., $Y_i \leq D_i(\mathbf{X}_i)$. This establishes the equivalence between the task assignment frequency constraint and the queue stability. That is to say, to satisfy the task assignment frequency constraint, we need to ensure the task virtual queue is stable under our strategy. Besides, we use $\mathbf{P}_i(t) = \{P_i(t), \forall i \in k\}$ to represent the queue vector backlog of all tasks.

Next, we improve the traditional Lyapunov optimization technique by taking both the task virtual queue and the user processing queue into consideration in this paper. We use $P_i(t)$ and $Q_j(t)$ to denote the queue backlog of task s_i and user u_j in time slot t, respectively. Let $\mathbf{Z}(t) = (P_i(t), Q_j(t), i = 1, \ldots, k, j = 1, \ldots, n), t = 0, 1, \ldots,$ denote all the queue backlogs in time slot t. We use the following definition of queue stability:

$$\overline{Z} \triangleq \lim_{T\to\infty} sup \frac{1}{T}\sum_{t=0}^{T-1}(\sum_{i=1}^{k}\mathbb{E}\{P_i(t)\} + \sum_{j=1}^{n}\mathbb{E}\{Q_j(t)\}) < \infty. \quad (5)$$

Obviously, there is a necessary condition to maintain the stability of the queue. According to the *queue stability theorem* [25], the queue is stable if and only if the time average arrival rates of all the tasks are no larger than the time average task processing ability of all the users. If not, no matter what the strategy is, the queue will never be stable. Thus, we have the following condition:

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{i=1}^{k} Y_i \leq \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1}\sum_{j=1}^{n} d_j(t). \quad (6)$$

In view of the previous description just consider the three parts of Fig. 2 separately, we now describe the overall process of it to make the structure of the entire system more clear. As the platform generates tasks in each time slot t, there are two fairness constraints for tasks (task assignment frequency) and users (user processing ability) in the system. We first convert these two fairness constraints to double queues, namely, the task virtual queue and the user processing queue, according to Lyapunov optimization. Then, we transform the problem of satisfying the two fairness constraints into the problem of satisfying the stability of the double queues, i.e., the task virtual queue and the user processing queue, and we further demonstrate the rationality of this conversion. At last, we combine the double queues to form a joint queue, and we propose the optimization problem in next part.

C. Problem formulation

After introducing the system model and converting the two constraints of fairness to a joint queue, we focus on the task assignment problem in fair mobile crowdsensing. First, we define a time average assignment cost function:

$$\overline{u} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{u(t)\}. \tag{7}$$

$\overline{u}$ is the optimization objective which should be minimized with the constraint of the joint queue stability as mentioned above, and we formulate a *Task Fair Assignment Problem* (FTAP) as follows:

$$\begin{aligned} Min: \ & \overline{u} \\ s.t. \ & x_{ij}[t] \in \{1, 0\}, \ \forall i, \forall t \\ & \overline{Z} \le \infty. \end{aligned} \tag{8}$$

Thus, according to the above optimization objective function, our primary goal is to find a task assignment strategy by determining all the values of $x_{ij}[t]$ in each time slot to minimize the time average assignment cost, subject to the above two constraints. The first constraint shows that the values of $x_{ij}[t]$ can only be 0 or 1. The second constraint guarantees the stability of the joint queue which consists of the task virtual queue and the user processing queue.

IV. THE LYAPUNOV OPTIMIZATION BASED METHOD

After defining the problem above, in this section, we give a Fair Task Assignment Strategy (FTAS) based on Lyapunov optimization in detail. As previously described, Lyapunov optimization [25] is widely used to solve problems which consider both the system stability and the optimization objective. In reality, it just makes the decision according to the current queue backlog. To achieve the optimization objective of minimizing the assignment cost while maintaining the queue stability, in brief, the main approach is to first construct a *drift-plus-penalty* function which consists of the queue backlog and the optimization objective function. Then, we find its upper bound theoretically. Lastly, this paper makes the assignment decision to minimize the upper bound in each time slot.

Firstly, we define a Lyapunov function $L(t)$ as follows:

$$L(t) \triangleq \frac{1}{2}(\sum_{i=1}^{k}[P_i(t)]^2 + \sum_{j=1}^{n}[Q_j(t)]^2). \tag{9}$$

This function represents not only a scalar measure of the task virtual queue congestion, but also the user processing queue. Then, we introduce a *one-slot conditional Lyapunov drift*, which represents the change in Lyapunov function from one slot to the next, as follows:

$$\Delta L(t) \triangleq \mathbb{E}\{L(t+1) - L(t)|\mathbf{Z}(t)\}. \tag{10}$$

The expectation here is taken over by the randomness of the task request's arrival rate and the assignment control action in each time slot. Next, according to Lyapunov optimization, we combine the Lyapunov drift with the objective function to achieve a *drift-plus-penalty* term in time slot t:

$$\Delta L(t) + V \cdot \mathbb{E}\{u(t)|\mathbf{Z}(t)\}. \tag{11}$$

The control parameter $V \ge 0$ is an important weight on how much we emphasize the assignment cost minimization compared to the system stability, which enables various trade-offs between queue backlog stability and assignment cost minimization. Intuitively, the smaller the $\Delta L(t)$ and $\mathbb{E}\{u(t)|\mathbf{Z}(t)\}$ is, the better our expectations are. In the following part of this section, we prove the upper bound and have the following theorem according to the *drift-plus-penalty* function:

Theorem 1. ***(Drift-plus-penalty bound)****. In each time slot t, for a given parameter $V > 0$, under any feasible control decisions, the* drift-plus-penalty *expression has the following upper bound :*

$$\begin{aligned} &\Delta L(t) + V \cdot \mathbb{E}\{\sum_{i=1}^{k}\sum_{j=1}^{n} c_{ij}(t) \cdot x_{ij}[t]|\mathbf{Z}(t)\} \\ &\le D + \mathbb{E}\{\sum_{i=1}^{k} P_i(t)Y_i|\mathbf{Z}(t)\} - \mathbb{E}\{\sum_{j=1}^{n} Q_j(t)d_j(t)|\mathbf{Z}(t)\} \\ &+ \mathbb{E}\{\sum_{i=1}^{k}\sum_{j=1}^{n}(Q_j(t) - P_i(t) + V \cdot c_{ij}(t)) \cdot x_{ij}[t]|\mathbf{Z}(t)\}, \end{aligned} \tag{12}$$

where $D = \frac{k}{2} \cdot \{(R_i^{max})^2 + (Y_i)^2\} + \frac{n}{2} \cdot \{(X_{max})^2 + (d_{max})^2\}$.

Proof: It is a ground truth in math that for any $x \ge 0, y \ge 0, z \ge 0$, $(max[x-y, 0] + z)^2 \le x^2 + y^2 + z^2 - 2x(y-z)$. according to this, we square both sizes of (4), and then we can get:

$$\begin{aligned} &P_i^2(t+1) - P_i^2(t) \\ &= (max[P_i(t) - \sum_{j=1}^{n} x_{ij}[t], 0] + Y_i)^2 - P_i^2(t) \\ &\le (\sum_{j=1}^{n} x_{ij}[t])^2 + (Y_i)^2 - 2 \cdot P_i(t)(\sum_{j=1}^{n} x_{ij}[t] - Y_i) \\ &\le (R_i^{max})^2 + (Y_i)^2 - 2 \cdot P_i(t)(\sum_{j=1}^{n} x_{ij}[t] - Y_i). \end{aligned} \tag{13}$$

It is worth noting that $\sum_{j=1}^{n} x_{ij}[t] \le R_i^{max} = 1$ and we assume Y_i is a constant. Then, similar to $P_i(t)$, according to (3) and the assumptions that $0 \le \sum_{i=1}^{k} x_{ij}[t] \le X_{max}$ and $0 \le d_j(t) \le d_{max}$, we have:

$$
\begin{aligned}
&[Q_j(t+1)]^2 - [Q_j(t)]^2 \\
&= (max[Q_j(t) - d_j(t), 0] + \sum_{i=1}^{k} x_{ij}[t])^2 - Q_j^2(t) \\
&\leq (d_j(t))^2 + (\sum_{i=1}^{k} x_{ij}[t])^2 - 2Q_j(t)(d_j(t) - \sum_{i=1}^{k} x_{ij}[t]) \\
&\leq (X_{max})^2 + (d_{max})^2 - 2Q_j(t)(d_j(t) - \sum_{i=1}^{k} x_{ij}[t])). \quad (14)
\end{aligned}
$$

Combining the above two equations, we can have the following result:

$$
\begin{aligned}
&\{\Delta L(t)|\mathbf{Z}(t)\} \leq D + \mathbb{E}\{\sum_{j=1}^{n} Q_j(t)(\sum_{i=1}^{k} x_{ij}[t] - d_j(t))|\mathbf{Z}(t)\} \\
&+ \mathbb{E}\{\sum_{i=1}^{n} P_i(t)(Y_i - \sum_{j=1}^{n} x_{ij}(t)|\mathbf{Z}(t)\}. \quad (15)
\end{aligned}
$$

$$
D = \frac{k}{2} \cdot \{(R_i^{max})^2 + (Y_i)^2\} + \frac{n}{2} \cdot \{(X_{max})^2 + (d_{max})^2\}. \quad (16)
$$

Then, we take expectations on both sides of (15), and add the term $V \cdot \mathbb{E}\{\sum_{i=1}^{k} \sum_{j=1}^{n} c_{ij}(t) \cdot x_{ij}[t]|\mathbf{Z}(t)\}$ to both sides and simplify the terms. Theorem 1 is proven. ■

In order to minimize the time average cost while maintaining the system stability, the objective is now to minimize the upper bound of Theorem 1. The controller can observe the current queue backlog of task virtual queues and user processing queues, and Y_i and $d_j(t)$ are known. We just need to take the control action to minimize the last function to minimize the upper bound:

$$
\mathbb{E}\{\sum_{i=1}^{k} \sum_{j=1}^{n} (Q_j(t) - P_i(t) + V \cdot c_{ij}(t)) \cdot x_{ij}[t]|\mathbf{Z}(t)\}. \quad (17)
$$

We design a strategy called *Fair Task Assignment Strategy* (FTAS) to minimize (17) as shown in Algorithm 1 and here, we give a detailed illustration of it. For each task $i \in T$, we need to find $j^* = argmin_{j \in U}(Q_j(t) - P_i(t) + V \cdot c_{ij}(t))$ in time slot t. If $(Q_{j^*}(t) - P_i(t) + V \cdot c_{ij^*}(t)) \leq 0$, we set $x_{ij^*}[t] = 1$, which means that the task s_i will be assigned to user u_{j^*} in time slot t, otherwise, $x_{ij^*}[t] = 0$. Meanwhile, for other users $j \in U_{-j^*}$ except j^*, we set $x_{ij}[t] = 0$.

This task assignment approach makes the control decision in each time slot to guide the platform to assign tasks to the most suitable users to minimize the time average assignment cost while meeting the condition of the system stability. After the assignment of all the tasks in time slot t, we update the task virtual queues and the user processing queues.

Algorithm 1 Fair Task Assignment Strategy (FTAS)

Input: t: current time slot, S: a set of tasks, U: a set of users with their mobility, $P_i(t)$: the task virtual queue in current time slot, Y_i: the request rate in task virtual queue, $Q_j(t)$: the user processing queue in current time slot.
Output: j^*: the user we selected to assign the task in time slot t, $x_{ij^*}[t]$: whether task i is assigned in time slot t.

```
for i ∈ T do
    Find j* ∈ U which makes j* = argmin_{j∈U}(Q_j(t) − P_i(t) + V · c_ij(t));
    if (Q_j*(t) − P_i(t) + V · c_ij*(t)) ≤ 0 then
        Set x_ij*[t] = 1;
    else
        Set x_ij*[t] = 0;
    For other users j ∈ U_{−j*}, we set x_ij[t] = 0;
Update P_i(t) and Q_j(t);
return x_ij*[t]
```

V. OPTIMAL PERFORMANCE ANALYSIS

In this section, the performance analysis analyzes the gap between the solution obtained by the proposed drift-plus-penalty algorithm and the optimal solution (optimality gap). Thus, to demonstrate the performance analysis, we first describe the optimal solution of the problem. Next, we analyze our strategy's performance in depth from the perspective of the objective function of *time average assignment* cost and the *stability of double queues*. To start, we define a strategy for the problem:

Lemma 1. ***(U-only policy)****. This policy is also called the* stationary randomized algorithm*, which makes the optimal control action only based on the system current state and hence is independent of the queue backlog [25]. In other words, the stationary randomized algorithm in this paper determines the value of $x_{ij}[t]$ according to a conditional probability distribution which depends on the task producing rate, but is independent of the queue backlog $\mathbf{Z}(t)$.*

Thus, it is obvious that the following theorem exists, which can obtain the optimal time average assignment cost based on **u-only policy**.

Theorem 2. ***(Optimal u-only policy)****. The* optimal u-only policy *is a* u-only policy*, which means there is a policy θ that can determine the values of all the $x_{ij}[t]$ for $i \in S$ and $j \in U$ in time slot t, while meeting the following conditions, where $\boldsymbol{\alpha}$ is a given task producing rate:*

$$
\mathbb{E}\left\{u^{\theta}(t)\right\} = u^*(\boldsymbol{\alpha}), \quad (18)
$$

$$
\mathbb{E}\{\sum_{j=1}^{n} x_{ij}^{\theta}[t]\} \geq \mathbb{E}\left\{Y_i\right\}, \quad (19)
$$

$$
\mathbb{E}\left\{d_j(t)\right\} \geq \mathbb{E}\{\sum_{i=1}^{k} x_{ij}^{\theta}[t]\}. \quad (20)
$$

In reality, we assume that there exists an *optimal u-only policy* for problem (8). This is a crucial assumption which at least means that this problem has an optimal solution, and the *optimal u-only policy* is a construction method that provides the optimal solution. Actually, M.J.Neely [25] has demonstrated that if the problem (8) has a solution, there must be an *optimal u-only policy*. Here, we omit the proof for brevity.

It is worth noting that in Theorem 2, (19) denotes that the invalid threshold of the task should be less than or equal to the assigned times, and (20) shows that the number of assigned tasks for user u_j should be less than or equal to user u_j's processing ability. In addition, there also exists an *arbitrary u-only policy* which is demonstrated in the lemma below:

Lemma 2. ***(Arbitrary u-only policy)***. *We assume that there exists an* arbitrary u-only policy *σ, which does not require optimally that can determine the values of all the $x_{ij}[t]$ for $i \in S$ and $j \in U$ in time slot t, while meeting the following conditions:*

$$\mathbb{E}\left\{u^{\sigma}(t)\right\} = u^{\star}(\boldsymbol{\alpha}) \geq u^{*}(\boldsymbol{\alpha}), \tag{21}$$

$$\mathbb{E}\{\sum_{j=1}^{n} x_{ij}^{\sigma}[t]\} - \mathbb{E}\{Y_i\} \geq \epsilon, \tag{22}$$

$$\mathbb{E}\{d_j(t)\} - \mathbb{E}\{\sum_{i=1}^{k} x_{ij}^{\sigma}[t]\} \geq \epsilon, \tag{23}$$

where ϵ is a small positive number, $u^{\star}(\boldsymbol{\alpha})$ is the total assignment cost under the arbitrary u-only policy *σ, and $u^{\sigma}(t)$ is a bounded function which satisfies that $u_{min} \leq u^{\star}(\boldsymbol{\alpha}) \leq u_{max}$.*

Theorem 3. ***(Queue stability performance of the proposed strategy)***. *If we are given a task producing rate vector $\boldsymbol{\alpha}$ and an* arbitrary u-only policy *σ, then, we have the following bounds of the queue stability, where $u^{\star}$ is the time average cost that can be achieved by* arbitrary u-only policy *that stabilizes the queue:*

$$\overline{Z} \triangleq \lim_{T\to\infty} \frac{1}{T}(\sum_{i=1}^{k}\mathbb{E}\{P_i(t)\} + \sum_{j=1}^{n}\mathbb{E}\{Q_j(t)\}) \leq \frac{D'}{\epsilon}, \tag{24}$$

where $D' = D + V \cdot (u_{max} - u_{min})$.

Proof: According to Lemma 2, for $\boldsymbol{\alpha}$, there exists an *arbitrary u-only policy* σ that satisfies the following:

$$\begin{aligned}&\Delta L(t) + V\cdot\mathbb{E}\{u(t)|\mathbf{Z}(t)\}\\ &\leq D + \mathbb{E}\{\sum_{i=1}^{k} P_i(t)Y_i|\mathbf{Z}(t)\} - \mathbb{E}\{\sum_{j=1}^{n} Q_j(t)d_j(t)|\mathbf{Z}(t)\}\\ &+\mathbb{E}\{\sum_{i=1}^{k}\sum_{j=1}^{n}(Q_j(t) - P_i(t) + V\cdot c_{ij})\cdot x_{ij}^{\sigma}[t]|\mathbf{Q}(t)\}.\end{aligned} \tag{25}$$

Substituting the corresponding parts in (25) with (21), (22) and (23), we have:

$$\begin{aligned}&\Delta L(t) + V\cdot\mathbb{E}\{u|\mathbf{Z}(t)\} \leq D + V\cdot\mathbb{E}\{u^{\star}(\boldsymbol{\alpha})|\mathbf{Z}(t)\}\\ &-\epsilon\mathbb{E}\{\sum_{i=1}^{k} P_i(t)|\mathbf{Z}(t)\} - \epsilon\mathbb{E}\{\sum_{j=1}^{n} Q_j(t)|\mathbf{Z}(t)\}.\end{aligned} \tag{26}$$

As mentioned above that $u_{min} \leq u^{\star}(\boldsymbol{\alpha}) \leq u_{max}$, we rearrange the terms and get:

$$\begin{aligned}&\mathbb{E}\{L(t+1) - L(t)\} + \epsilon\mathbb{E}\{\sum_{i=1}^{k} P_i(t)\} + \epsilon\mathbb{E}\{\sum_{j=1}^{n} Q_j(t)\}\\ &\leq D + V\cdot(u_{max} - u_{min}).\end{aligned} \tag{27}$$

Here, we set $D' = D + V\cdot(u_{max} - u_{min})$. $D + V\cdot(u_{max} - u_{min})$ is a constant. So we have:

$$\begin{aligned}&\mathbb{E}\left\{L(t+1) - L(t)\right\}\\ &\leq D' - \epsilon\mathbb{E}\{\sum_{j=1}^{n} Q_j(t)\} - \epsilon\mathbb{E}\{\sum_{i=1}^{k} P_i(t)\}.\end{aligned} \tag{28}$$

Summing (28) from $t = 0$ to $T-1$, we get:

$$\begin{aligned}&\mathbb{E}\left\{L(T)\right\} - \mathbb{E}\left\{L(0)\right\}\\ &\leq T\cdot D' - \epsilon\sum_{t=0}^{T-1}\mathbb{E}\{\sum_{i=1}^{k} P_i(t)\} - \epsilon\sum_{t=0}^{T-1}\mathbb{E}\{\sum_{j=1}^{n} Q_j(t)\}.\end{aligned} \tag{29}$$

Due to the fact that $L(0) = 0$ and $L(T) \geq 0$, we rearrange (29) and get:

$$\epsilon\sum_{t=0}^{T-1}\mathbb{E}\{\sum_{i=1}^{k} P_i(t)\} + \epsilon\sum_{t=0}^{T-1}\mathbb{E}\{\sum_{j=1}^{n} Q_j(t)\} \leq T\cdot D'. \tag{30}$$

By dividing both sides of (30) by $\epsilon\cdot T$ and taking a limit as $T \longrightarrow \infty$, we can obtain (24).

■

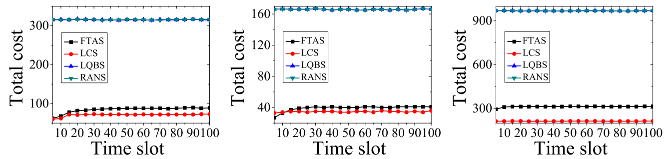
Theorem 4. ***(Cost performance of the proposed strategy)***. *If we are given a task producing rate vector $\boldsymbol{\alpha}$ and an* optimal u-only policy *θ, then, we have the following bounds of the time average assignment cost, where u^{*} is the optimal time average cost that can be achieved by* optimal u-only policy *that stabilizes the queue:*

$$\overline{u} \triangleq \lim_{T\to\infty} sup\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\{u(t)\} \leq u^{*} + \frac{D}{V}. \tag{31}$$

Proof:

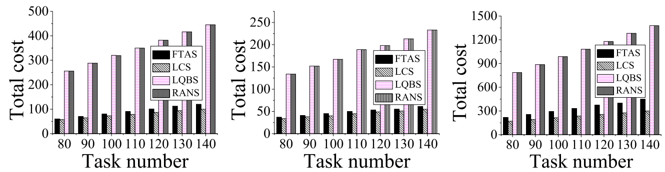
Similarly, using Theorem 1, Theorem 2 and the fact that $\mathbb{E}\left\{Q_j(t) \geq 0\right\}$ for all $j \in U$ and $\mathbb{E}\left\{P_i(t) \geq 0\right\}$ for all $i \in K$, we have:

$$\begin{aligned}&\mathbb{E}\left\{\Delta L(t) + V\cdot u(t)|\mathbf{Z}(t)\right\} \leq D + V\cdot\mathbb{E}\left\{u^{\theta}(t)|\mathbf{Z}(t)\right\}\\ &+\mathbb{E}\{\sum_{i=1}^{k}\sum_{j=1}^{n} x_{ij}^{\theta}[t]\cdot Q_j(t)|\mathbf{Z}(t)\} - \mathbb{E}\{\sum_{j=1}^{n} d_j(t)\cdot Q_j(t)|\mathbf{Z}(t)\}\\ &+\mathbb{E}\{\sum_{i=1}^{k} P_i(t)\cdot Y_n^i|\mathbf{Z}(t)\} - \mathbb{E}\{\sum_{i=1}^{k}\sum_{j=1}^{n} x_{ij}^{\theta}[t]\cdot P_i(t)|\mathbf{Z}(t)\}.\end{aligned} \tag{32}$$



(a) *Changchun/taxi* (b) *Epfl/mobility* (c) *Feeder*

Fig. 3: Total cost with time slot on three data sets.



(a) *Changchun/taxi* (b) *Epfl/mobility* (c) *Feeder*

Fig. 4: Total cost with task number on three data sets.

Furthermore, since the strategy does not depend on the queue status, we have $\mathbb{E}\left\{u^{\theta}(t)|\mathbf{Z}(t)\right\} = u^{*}(\boldsymbol{\alpha})$, and based on (19) and (20), we can get:

$$\mathbb{E}\left\{\Delta L(t) + V \cdot u(t)|\mathbf{Z}(t)\right\} \leq D + V \cdot u^{*}(\boldsymbol{\alpha}). \tag{33}$$

Summing (33) from $t = 0$ to $T-1$, using the fact that $L(t) \geq 0$ for all t and dividing both sides by $T \cdot V$, we have:

$$\sum_{t=0}^{T-1} \mathbb{E}\left\{u(t)|\mathbf{Z}(t)\right\} \leq \frac{D}{V} + u^{*}(\alpha). \tag{34}$$

Taking a limit as $T \to \infty$ for both sides of (34), we obtain (31). Hence, Theorem 4 is proven. ■

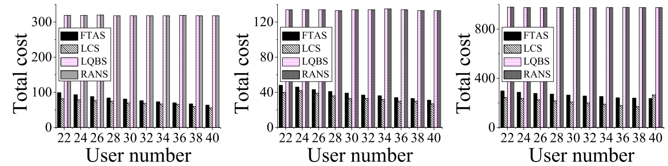
By proving Theorem 3 and Theorem 4, we achieve the upper bounds of the time average assignment cost and the time average queue backlog. In addition, these two theorems also illustrate that there is $[O(1/V), O(V)]$ trade-off between the assignment cost and the queue backlog. Intuitively, with a large enough V, the time average assignment cost achieved by the proposed strategy can become infinitely close to the optimum, but the queue backlog will be large.

VI. PERFORMANCE EVALUATION

A. Data sets

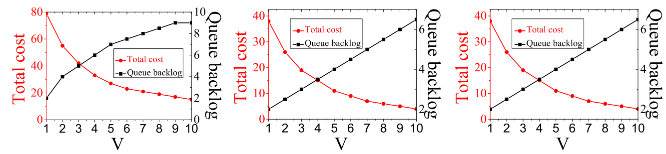
In this paper, we conduct simulations to evaluate the performance of the task assignment strategies over three widely-used real-world data sets: *Changchun/taxi*, *Epfl/mobility*, and *Feeder*. For each user, we randomly select one GPS point from his/her trajectory as the starting point in each time slot. Moreover, we randomly select POI locations as the task locations.

- *Changchun/taxi* contains the GPS data collected from taxis in Chang Chun, China. We select 250 traces as used data in our paper, each of which was collected from 8:00 to 20:00 in one day.
- *Epfl/mobility* [26] is a trace set of mobility data from about 500 taxi cabs collected over 30 days in the San Francisco Bay Area, USA.



(a) *Changchun/taxi* (b) *Epfl/mobility* (c) *Feeder*

Fig. 5: Total cost with user number on three data sets.



(a) *Changchun/taxi* (b) *Epfl/mobility* (c) *Feeder*

Fig. 6: Total cost with the change of V on three data sets.

- *Feeder* [27] contains four kinds of data: the smartphone CDR data, smartcard data, taxicab GPS data, and bus GPS data collected from Shen Zhen, China. We select 200 taxi traces as the data in this paper, each of which has the continuous GPS records collected from the same period of time, i.e., 8:00-18:00, for two days.

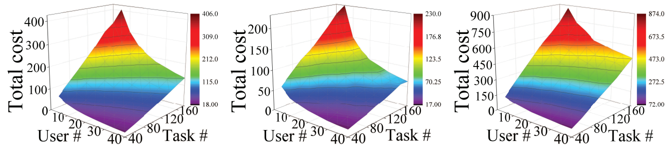
B. Baselines

To prove the effectiveness of our strategy, we utilize the other following three task assignment strategies besides the proposed strategy in the simulation:

- Fair Task Assignment Strategy (FTAS): The proposed task assignment strategy in the paper, which is based on Lyapunov optimization framework to guide the platform to allocate the tasks in each time slot.
- Random Strategy (RANS): The platform randomly assigns each task to any user in each time slot.
- Lowest Queue Backlog Strategy (LQBS): The platform assigns each task to the user with the lowest queue backlog in each time slot.
- Lowest Cost Strategy (LCS): The platform assigns each task to the nearest user in each time slot .

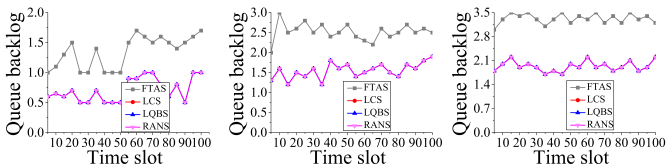
C. Simulation results

First, we analyze the indicator of the time average assignment cost of the proposed strategy along with the time slot t, the number of users, the number of tasks, and the growth of V in Figs. 3-6. Fig. 3 shows that assignment cost changes with the growth of time slot on three data sets. We observe the queue backlog value in each of the 5 time slots. The simulation results illustrate that LCS achieves the minimum assignment cost as it guides the platform to assign the tasks to their nearest users. Our proposed approach, $FTAS$, achieves a higher cost than that of LCS, but much lower than that of $LQBS$ and $RANS$. Hence, $FTAS$ maintains the system stability with a low time average assignment cost. Next, we investigate the impact of the number of tasks in Fig. 4 on the time average assignment cost. Fig. 4(a) demonstrates the result of the simulation conducted on *Changchun/taxi*, which shows



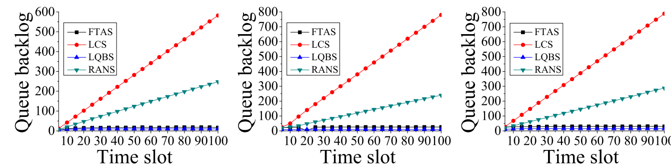
(a) *Changchun/taxi* (b) *Epfl/mobility* (c) *Feeder*

Fig. 7: Total cost with user number and task number.



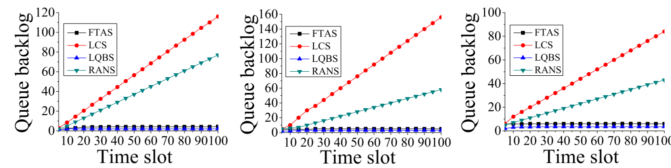
(a) *Changchun/taxi* (b) *Epfl/mobility* (c) *Feeder*

Fig. 8: Task queue backlog with time slot on three data sets.



(a) *Changchun/taxi* (b) *Epfl/mobility* (c) *Feeder*

Fig. 9: User queue backlog with time slot on three data sets.



(a) *Changchun/taxi* (b) *Epfl/mobility* (c) *Feeder*

Fig. 10: Average queue backlog with time slot.

that the time average assignment costs of four task assignment strategies increase along with the number of tasks. Similarly, LCS achieves the lowest time average cost and $FTAS$ incurs more cost than that of LCS, but performs better than $LQBS$ and $RANS$, which aligns with our theoretical analysis. Since the simulation results in Fig. 4(b) and Fig. 4(c) are similar to that in Fig. 4(a), we do not give additional description here. Finally, we give a description of the impact of the user number on the time average assignment cost on three data sets. Taking Fig. 5(a), which is conducted on *Changchun/taxi* as example, we can observe that the time average assignment costs of $FTAS$ and LCS decrease with the growth of the number of users. This is because as the number of users increases, there are more chances for the platform to assign the tasks to nearer users. LCS achieves the lowest time average assignment cost. Because $LQBS$ guides the platform to assign tasks to users who have the lowest queue backlog, the assignment cost of it changes slightly with the change of user number. The results of Fig. 5(b) and Fig. 5(c) are similar to Fig. 5(a).

We also investigate the trend of the total assignment cost of $FTAS$ with respect to the number of users and the number of tasks jointly on three data sets in Fig. 7 by using three-dimensional figures. The number of tasks varies from 40 to 180 and the number of users varies from 5 to 40. Fig. 7 shows that the total assignment cost increases with the growth of task number and the decrease of user number, which corresponds to the theoretical analysis.

As we can see in Fig. 6, with the value of V increasing from 1 to 10, all the time average assignment costs on the three data sets decrease. As mentioned, V is an important weight reflecting how much we emphasize minimizing the primary objective (assignment cost), and because we find a trade-off between system stability and the assignment cost minimization, as the V value increases, the system will make more contributions to minimizing the assignment cost. This means that the platform would like to assign tasks to the nearest users and thus, the assignment cost decreases.

Immediately after, we study the change of the average queue backlog with the value of V. The average queue backlog here represents the queue backlog value of all queues, including both the task virtual queues and the user processing queues. Similarly, the value of V varies from 1 to 10 and with each V, we observe the average queue backlog with the time slot from 1 to 100. Fig. 6 shows that the average queue backlogs increase with the growth of the V value. Combining the description of the assignment cost above, as V increases, the platform tries its best to assign tasks to the nearest users. Thus, the platform may buffer the task in the task virtual queue until the nearest user is ideal, and the scene in Fig. 6 appears.

Next, we investigate the change of the task virtual queue backlog, the user processing queue backlog, and the average backlog with the growth of the time slot t in Figs. 8-10. We observe the queue backlog every 5 time slots between the 1st and 100th time slot. Fig. 8 shows that the task virtual queue backlog of $FTAS$ is higher than the other three strategies, and the task virtual queue backlog of other three methods are identical. This is because in the cases of LCS, $LQBS$, and $RANS$, the platform assigns tasks that meet the assignment frequency requirements in each time slot, therefore, the task assignment rates in each time slot of all cases are identical. By contrast, $FTAS$ controls the amount of the tasks which should be assigned to users to ease their burden. This means that $FTAS$ may not assign any tasks in some time slots if there are too many tasks in the users' queues. Hence, the task virtual queue backlog of $FTAS$ is higher. It is worth noting that due to the fact that the assignment rate is stochastic in our system, the task virtual queue backlogs of all the strategies fluctuate in different time slots.

In Fig. 9, the user processing queue backlogs of the four task assignment strategies on three data sets are ranked as follows: $LQBS < FTAS < RANS < LTCS$. Since $LQBS$ guides the platform to assign tasks to the users who have the lowest queue backlogs, the user processing queue backlog of $LQBS$ is the lowest. Although the user processing queue backlog of $FTAS$ is larger than that of $LQBS$, these two results are very close and they are much smaller than that of $RANS$ and LCS. This proves that the user processing queue backlog of

$LQBS$ and $FTAS$ will not reach infinity over time, which implies that our proposed approach can maintain the queue stability. As for LCS strategy, due to the fact that a user may be close to many tasks, the platform may assign different tasks to him over time. Therefore, the performance of LCS is worse than that of $RANS$. In Fig. 10, we study the average queue backlog in relation to the change of the time slot. Since the user queue backlog is much more than the task virtual queue backlog, the trend of the average queue backlog is similar to that of the user processing queue backlog.

VII. CONCLUSIONS

In this paper, we investigate the task assignment problem in an MCS scenario which considers the fairness for both tasks and users, where each task should be assigned by a assignment frequency and each user's processing ability is limited. With the objective of minimizing the assignment cost, to satisfy the fairness constraints, we first model each user's processing ability as a user processing queue and convert the constraint of task assignment frequency to the task virtual queues. Then a *Fair Task Assignment Problem* (FTAP), which aims at minimizing the assignment cost and maintaining the double queues' stability simultaneously, is formulated. To solve the problem, we propose a *Fair Task Assignment Strategy* (FTAS) by utilizing Lyapunov optimization, which can make control actions in each time slot to minimize the assignment cost while maintaining the system stability. To analyze the performance of the proposed assignment strategy, we find an upper bound of the total assignment cost and the queue backlog. Finally, extensive evaluations on *Changchun/taxi*, *Epfl/mobility*, and *Feeder* verify the effectiveness of the proposed strategy. The results of our simulation show that the proposed strategy outperforms the other three baselines by simultaneously minimizing the assignment cost and maintaining the system stability.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundations of China under Grant No. 61772230, No. 61972450 and No. 61772230, National Natural Science Foundations of China for Young Scholars No. 61702215, Changchun Science and Technology Development Project No. 18DY005, Innovation Capacity Building Project of Jilin Provincial Development and Reform Commission No. 2020C017-2, National Defense Science and Technology Key Laboratory Fund Project No. 61421010418Natural Science Foundations of Jilin Province No. 20190201022JC, National Science Key Lab Fund project No. 61421010418, and this research is also supported in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

REFERENCES

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

[2] B. Guo, C. Chen, D. Zhang, Z. Yu, and A. Chin, "Mobile crowd sensing and computing: when participatory sensing meets participatory social media," *IEEE Communications Magazine*, vol. 54, no. 2, 2016.

[3] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications magazine*, vol. 48, no. 9, 2010.

[4] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. of ACM SenSys 2009*, pp. 85–98.

[5] 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. of ACM SenSys 2014*, pp. 251–265.

[6] X. Fan, P. Yang, C. Xiang, and L. Shi, "imap: A crowdsensing based system for outdoor radio signal strength map," in *Proc. of IEEE ISPA 2016*, pp. 1442–1447.

[7] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, 2017.

[8] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *Proc. of IEEE INFOCOM 2014*, pp. 745–753.

[9] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, 2016.

[10] H. Li, T. Li, W. Wang, and Y. Wang, "Dynamic participant selection for large-scale mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, 2018.

[11] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: Multi-task allocation in mobile crowd sensing," in *Proc. of ACM UbiComp 2016*, pp. 403–414.

[12] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, 2018.

[13] Q. Zhao, Y. Zhu, H. Zhu, J. Cao, G. Xue, and B. Li, "Fair energy-efficient sensing task allocation in participatory sensing with smartphones," in *Proc. of IEEE INFOCOM 2014*, pp. 1366–1374.

[14] G. Sun, Y. Wang, X. Ding, and R. Hu, "Cost-fair task allocation in mobile crowd sensing with probabilistic users," *IEEE Transactions on Mobile Computing*, 2019, in press.

[15] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *Proc. of IEEE INFOCOM 2015*, pp. 2803–2811.

[16] K. Sooksatra, R. Li, Y. Li, X. Guan, and W. Li, "Fairness-aware auction mechanism for sustainable mobile crowdsensing," in *Proc. of Springer WASA 2019*, pp. 310–321.

[17] W. Fang, X. Yao, X. Zhao, J. Yin, and N. Xiong, "A stochastic control approach to maximize profit on service provisioning for mobile cloudlet platforms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, 2016.

[18] D. Y. Zhang and D. Wang, "An integrated top-down and bottom-up task allocation approach in social sensing based edge computing systems," in *Proc. of IEEE INFOCOM 2019*, pp. 766–774.

[19] X. Wang, R. Jia, X. Tian, and X. Gan, "Dynamic task assignment in crowdsensing with location awareness and location diversity," in *Proc. of IEEE INFOCOM 2018*, pp. 2420–2428.

[20] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. of IEEE INFOCOM 2012*, pp. 1431–1439.

[21] J.-S. Liu and W.-L. Chang, "Flow control and resource allocation in cognitive radio networks," in *Proc. of ACM ICACS 2018*, pp. 190–194.

[22] Y. Li, S. Xia, B. Cao, Q. Liu *et al.*, "Lyapunov optimization based trade-off policy for mobile cloud offloading in heterogeneous wireless networks," *IEEE Transactions on Cloud Computing*, 2019, in press.

[23] W. Fang, Y. Li, H. Zhang, N. Xiong, J. Lai, and A. V. Vasilakos, "On the throughput-energy tradeoff for data transmission between cloud and mobile devices," *Information Sciences*, vol. 283, 2014.

[24] Y. Han and Y. Zhu, "Profit-maximizing stochastic control for mobile crowd sensing platforms," in *Proc. of IEEE MASS 2014*, pp. 145–153.

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

[26] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "Crawdad data set epfl/mobility," *2009-02-24 Downloaded from.*

[27] D. Zhang, J. Zhao, F. Zhang, R. Jiang, and T. He, "Feeder: supporting last-mile transit with extreme-scale urban infrastructure data," in *Proc. of IEEE/ACM IPSN 2015*, pp. 226–237.