Received June 14, 2019, accepted July 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Stackelberg Game based Dynamic Admission and Scheduling in Mobile Crowdsensing

Zhifei Wang¹, Haohao Zhou¹, Yongpeng Zhao², Yahui Wu¹, Su Deng¹, and Hongbin Huang¹

¹Science and Technology on Information System Engineering Laboratory, National University of Defense Technology, Changsha 410073, China
²Strategic Planning Office, Beijing 100091, China.

Corresponding author: Yahui Wu (e-mail: wuyahui@nudt.edu.cn ).

This work was supported by the NSFC under Grant 61401482 and 61871388, and the NSF of Hunan under Grant 2018JJ2477.

ABSTRACT Mobile crowdsensing is a crowdsourcing-based paradigm, where the platform executes the sensing requests with the help of many common peoples’ handheld devices (typically smartphones). In this paper, we mainly address the dynamic sensing request admission and smartphone scheduling problem to maximize the long-term profit, taking into account the competitive interaction procedure between the platform and smartphones, the queue backlog, and the location of sensing requests and smartphones. First, formulate this problem as a discrete time model and the interaction procedure between the platform and smartphones as a Stackelberg game. Then we introduce the Lyapunov optimization technique and design a Stackelberg game based dynamic Admission and Scheduling algorithm (SAS). In SAS, all control decisions are made only based on the currently available information and none of the stakeholders, including the platform and smartphones, can improve his utility by unilaterally changing its current strategy. Next, we design an online Cooperative dynamic Admission and Scheduling algorithm (CAS) for the situation where the platform and smartphones work in cooperative way. Theoretical analysis shows that under any control parameter \( V > 0 \), both SAS and CAS algorithm can achieve \( O(1/V) \)-optimal average profit, while the sensing request backlog is bounded by \( O(V) \). Extensive numerical results based on both synthetic and real trace demonstrate the Stackelberg equilibrium of SAS. CAS always outperforms SAS, and in some certain situations, the profit of SAS is very close to that of CAS.

INDEX TERMS Dynamic admission and scheduling, Lyapunov optimization, mobile crowdsensing, Stackelberg game.

I. INTRODUCTION

Mobile crowdsensing is a novel paradigm where an immense number of mobile devices (typically smartphones) collectively provide sensing and computing services related to a certain phenomenon of interest [1], [2]. Recent years have witnessed the rapid and explosive growth of mobile devices equipped a rich set of cheap yet powerful resources (e.g., accelerometer, digital compass, GPS, camera microphone) [3], [4]. These advances promote the low cost, ubiquitous sampling, large-scale coverage and application diversity of mobile crowdsensing, which is not possible with traditional sensor networks [5], [6]. Thereafter, mobile crowdsensing is becoming applicable in a large range of applications such as traffic monitoring [7], environment monitoring [8], parking availabilities [9], urban safety [10], and so on.

A typical mobile crowdsensing system consists of a service platform resided in the cloud and a collection of mobile smartphones registered to the system [11], [12]. The sensing requests arrive at the platform dynamically and the platform schedules smartphones to serve these requests. The platform needs to make two important control decisions: (1) admission control: how many requests to be admitted at any given time. This control decision ensures that the admitted requests will not exceed the capacity of the system. (2) scheduling control: how to schedule smartphones to process the sensing requests at any given time to maximize the profit of the platform.

In this paper, we are trying to address the following problems: Given location aware dynamical arrival of requests and volatile mobility of smartphones, how can the platform efficiently decide the amount of admitted requests, the unit price of processing each request at each time instant, in order...
to maximize long-term profit while guaranteeing the average queue backlog within finite length? Given the price offered by the platform, how should each smartphone determine the amount of requests to be processed, in order to achieve its own profit maximization? Actually, the interaction procedure between the platform and smartphones is a Stackelberg game [13]. In the first stage, the platform determines the amount of admitted requests and announces the unit price of requests to each smartphone that can maximize its profit. In the second stage, each smartphone responds with the optimal number of requests that it is willing to serve.

It is of great challenge to address the above profit-maximizing admission and scheduling control problem. First, it is challenging to predict the future information of sensing requests’ arrival and the mobility of each smartphone. We can only use the current information to make online decisions to achieve long-term profit maximization. Second, the requests are location aware and the location of each smartphone is dynamic. Location consideration brings some difficulties and complexities to tackle the problem. Moreover, both the platform and smartphones want to achieve profit maximization on their own and they make decisions in a competitive way, which increases the complexity of online admission and scheduling.

In this paper, we are trying to tackle these challenges. To our best knowledge, we are the first to take the competitive interaction procedure between the platform and smartphones into consideration, and simultaneously and efficiently tackle these challenges. The major contributions of this paper are summarized as follows:

- Novel model structure. We formally formulate the location aware admission and scheduling problem in mobile crowdsensing by a simple and general discrete time model. For each time slot, we model the interaction procedure between the platform and smartphones as a Stackelberg game. This is a novel model structure for mobile crowdsensing to the literature.
- Online Stackelberg game based dynamic Admission and Scheduling algorithm(SAS). We design the SAS algorithm based on backward induction analysis and Lyapunov optimization theory. In SAS, all control decisions are made only based on the currently available information and none of the stakeholders, including the platform and smartphones, can improve his utility by unilaterally changing its current strategy.
- Online Cooperative dynamic Admission and Scheduling algorithm(CAS). We design CAS by putting the competition between the platform and smartphones aside. CAS only depends on the currently available information.
- Performance evaluation. Theoretical analysis characterizes the bound of time average profit and backlog of SAS and CAS. Both algorithms can achieve optimal average profit asymptotically while guaranteeing the stability constraint. Extensive numerical results based on both synthetic and real trace demonstrate the Stackelberg Equilibrium of SAS. Then, we compare the performance of SAS with that of CAS.

The rest of this paper is organized as follows. We discuss related work in Section II. Section III presents the system model formulation. The SAS and CAS algorithms are developed in Section IV. Performance of our algorithms is analyzed and evaluated in Section V and Section VI, respectively. Section VII concludes this paper.

II. RELATED WORK

Recently, a large amount of prior works have been focused on the scheduling mechanism in mobile crowdsensing [14]-[16]. These literature can be broadly classified into three categories according to the interaction type between the platform and smartphones. In the first category of works [17]-[22], the objective of scheduling mechanism is to maximize the profit or minimize the cost from the perspective of the platform. Correspondingly, smartphones follow the mechanism and work in a cooperative way. For example, the works in [17] and [18] aim at optimizing the overall utility of multiple tasks in mobile crowdsensing while keeping a total budget constraint for the platform. In [19], the authors propose a dynamic task bundling mechanism that minimizes movement cost and the variance of participation, while maximizing the number of potential smartphones. The authors in [20], [21] and [22] study the profit-maximizing dynamic mobile crowdsensing system, where the state of smartphones changes over time and sensing tasks arrive stochastically.

Auction based approaches [23]-[29] constitute the second category, where smartphones work in a competitive way. Auctions have extensively been used in mobile crowdsensing, and have been effective in modeling the economic interactions between the platform and the smartphones. Specifically, the smartphones report their bids reflecting their processing and communication costs to the platform. The platform selects the smartphones and determines the corresponding payments, aiming to minimize the total payment or to maximize the platform utility. For example, the authors in [23] and [24] consider the user-centric model where each smartphone can ask for reserve price, and design a revenue maximizing reverse auction mechanism, which is which is computationally efficient, individually-rational, profitable, and truthful. In [25], the authors propose a novel privacy-preserving reverse auction-based mechanism to protect users' true bids against the honest-but-curious platform while minimizing the social cost. Other types of auction including Vickrey auction [26], VCG auction [27], [28] and random auction [29] and so on.

Another competitive category is based on the game theory [13]. Different from auction based scheduling mechanisms, the platform has more control over the payment to smartphones and smartphones tailor their actions to cater for the platform and maximize their profit. Recently, game based mechanisms have drawn extensive research attention [23],
[30]-[39]. References [30]-[33] model mobile crowdsensing system as a Stackelberg game, where the platform chooses a reward to attract enough smartphones and maximize its expected profit. Thereafter, smartphones make participation decision to maximize their utility. Whereas in [23] and [34]-[37], a smartphone compete for the reward and participates in the task by setting its participation level. All the above Stackelberg game based approaches also assume that there only exists one sensing task or multiple dependent tasks in the system. Luo et al [38] focus on scenario with multiple collaborative tasks, where each sensing task requires a group of smartphones to perform collaboratively. Specifically, smartphones would consider task priority and the platform would design suitable reward functions to allocate the total reward. Reference [39] points out that due to the arrival and departure of sensing tasks, resource should be allocated and released dynamically. This paper designs a game theoretic approach based mechanism to encourage the “best” neighbor mobile devices to share their own resource for sensing, and an auction based task migration to adjust resource among mobile devices for the better crowdsensing response.

The model used in this paper falls into the class of game based mechanism. Compared to the researches mentioned above, the SAS mechanism designed in this paper not only considers the competitive interaction procedure between the platform and smartphones, but also emphasizes the dynamic arrival of location aware sensing tasks. Furthermore, SAS designs admission mechanism to ensure that the admitted requests should not exceed the processing capacity of the mobile crowdsensing system. More importantly, SAS can achieve the long-term profit maximization of the platform.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first present the system model of the location aware mobile crowdsensing system with dynamic admission and scheduling controls. Then, we compare the interaction procedure of the Stackelberg game based approach with that of the cooperative approach. Finally, we give the problem formulation of these two approaches. Table I lists the frequently used notations.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{M}$</td>
<td>Set of all grids</td>
</tr>
<tr>
<td>$\mathcal{N}$</td>
<td>Set of all smartphones</td>
</tr>
<tr>
<td>$O(t)$</td>
<td>Amount of sensing requests arriving at grid $i$ at time $t$</td>
</tr>
<tr>
<td>$O^\infty$</td>
<td>Maximum amount of sensing requests arriving at grid $i$ for each time slot</td>
</tr>
<tr>
<td>$o_i(t)$</td>
<td>Amount of sensing requests admitted at grid $i$ at time $t$</td>
</tr>
<tr>
<td>$s_j(t)$</td>
<td>Whether smartphone $j$ can serve grid $i$ at time $t$</td>
</tr>
<tr>
<td>$a_i(t)$</td>
<td>Service rate of smartphone $j$ at grid $i$ for each time slot</td>
</tr>
<tr>
<td>$a^\infty_j$</td>
<td>Maximum service rate of smartphone $j$</td>
</tr>
<tr>
<td>$r(t)$</td>
<td>Service rate of grid $i$ at time $t$</td>
</tr>
<tr>
<td>$p_j(t)$</td>
<td>Unit price offered to smartphone $j$ at time $t$</td>
</tr>
<tr>
<td>$Q(t)$</td>
<td>Queue backlog of grid $i$ at time $t$</td>
</tr>
<tr>
<td>$U_j(t)$</td>
<td>Profit of smartphone $j$ at time $t$ under SAS</td>
</tr>
<tr>
<td>$U_i(t)$</td>
<td>Profit of the platform at time $t$ under SAS</td>
</tr>
<tr>
<td>$U_2(t)$</td>
<td>Profit of the system at time $t$ under CAS</td>
</tr>
</tbody>
</table>

A. System Model

As shown in Fig. 1, we consider a location aware mobile crowdsensing system with two control decisions, including admission control and scheduling control, where the whole sensing area is divided into a set $\mathcal{M} = \{1, 2, ... , m\}$ of grids. The platform operates in slotted time and serves $m$ grids of sensing requests with diverse arrival rates. A set $\mathcal{N} = \{1, 2, ... , n\}$ of smartphones with heterogeneous processing capacities and sensing regions register to the platform. They move around the sensing area according to certain mobility patterns.

1) DYNAMIC ADMISSION CONTROL

The first decision of the platform is to determine the amount of admitted requests. Suppose that each grid is associated with a sensing request queue. The sensing requests arrive at each grid dynamically. Specifically, we use $O_i(t)$ to denote the amount of sensing requests arriving at grid $i$ at time $t$. Assume that $O_i(t)$ is independent and identically distributed over time and bounded by $O^\infty_i$. Let $o_i(t)$ be the amount of sensing requests admitted at grid $i$ at time $t$. Obviously, we have $0 \leq o_i(t) \leq O_i(t)$, $i \in \mathcal{M}$. We call the vector $\{o_i(t), i \in \mathcal{M}\}$ as admission control of the mobile crowdsensing system.

2) DYNAMIC SCHEDULING CONTROL

The second decision of the platform is to schedule smartphones to serve sensing requests. As shown in Fig. 1, each smartphone has the potential to sense a specific region in a certain time slot according to his mobility pattern and sensing range. In particular, we use the symbol $s_j(t) \in \{0, 1\}$ to denote whether smartphone $j$ can serve the sensing requests at grid $i$ at time $t$. Thereafter, the sensing region of smartphone $j$ at time $t$ can be expressed as $s_j(t) = \{s_j(i), i \in \mathcal{M}\}$, which is marked by the shadow area around it.

We use $a_i(t)$ to denote the service rate of smartphone $j$ at time $t$. We assume that $0 < a_i(t) \leq a^\infty_j$, $i \in \mathcal{M}$. This assumption is quite practical as the resources of each smartphone are limited. If grid $i$ is located in the sensing region of
smartphone \( j \), then it can serve the sensing requests of grid \( i \) at the rate of \( \alpha_i(t) \). Denote \( r_i(t) \) as the service rate of grid \( i \) at time \( t \). \( r_i(t) \) is determined by the service rate of each smartphone and its sensing region,

\[
r_i(t) = \sum_{j \in \mathcal{N}} \alpha_j(t) s_j(t)
\]

(1)

3) QUEUE DYNAMIC
The platform buffers location aware sensing requests at each grid in its associated queue. Let \( Q_i(t) \) denote the sensing request backlog of grid \( i \) at time \( t \). Recall the service rate \( r_i(t) \) and admitted rate \( \alpha_i(t) \), we have the following queue dynamic,

\[
Q_i(t+1) = \max\{Q_i(t) - r_i(t), 0\} + \alpha_i(t)
\]

(2)

with \( \{Q_i(0), i \in \mathcal{M}\} = 0 \). We say \( Q_i(t) \) is stable if,

\[
\limsup_{t \to \infty} \frac{1}{t} \sum_{t=1}^{t} \mathbb{E}(Q_i(t)) < \infty
\]

(3)

Furthermore, we say that the mobile crowdsensing system is stable if all queues \( \{Q_i(t), i \in \mathcal{M}\} \) are stable.

4) PROCEEDS AND COST
The proceeds of the system depends on the amount of the admitted sensing requests. Specifically, the platform can earn \( \gamma \) units of utility by processing one unit of sensing requests.

Smartphones will incur extra costs when processing sensing tasks. Let \( c_j(t) \) and \( \beta_j(t) \) be the total cost and unit cost of smartphone \( j \) at time \( t \), respectively. Here, the unit cost \( \beta_j(t) \) is a function of the service rate \( \alpha_j(t) \). Suppose that the unit cost \( \beta_j(t) \) firstly decreases with \( \alpha_j(t) \) because of high volume production before a certain threshold, after which it increases as the large amount of sensing tasks becomes a burden. So, the unit cost \( \beta_j(t) \) is modelled as a quadratic function of service rate \( \alpha_j(t) \) [40],

\[
\beta_j(t) = c_j - b_j \alpha_j(t) + a_j \alpha_j^2(t)
\]

(4)

Where \( \{a_j, b_j, c_j, j \in \mathcal{N}\} \) are smartphone-specific positive constants. Therefore, the total cost \( c_j(t) \) of smartphone \( j \) can be computed as follows,

\[
c_j(t) = \alpha_j(t)\beta_j(t) \sum_{i \in \mathcal{M}} s_i(t)
\]

(5)

B. Interaction Procedure
In the Stackelberg game based approach, we regard the interaction procedure between the platform and smartphones as a two-stage Stackelberg game with the platform-leader and smartphone-follower structure. In Stage I, the platform determines the amount of admitted requests of each grid and unit price \( p_j(t) \) of sensing tasks for each smartphone \( j \) to maximize the long-term average profit. Then, the platform announces \( p_j(t) \) to each smartphone \( j \). In stage II, given the unit price \( p_j(t) \), each smartphone \( j \) decides its service rate \( \alpha_j(t) \) to maximize its individual profit, subject to the limited processing capacity. We call the vector \( \{p_j(t), \alpha_j(t), j \in \mathcal{N}\} \) as scheduling control of the mobile crowdsensing system.

Finally, each smartphone \( j \) processes the tasks at the rate of \( \alpha_j(t) \) and upload the results to the platform.

In the cooperative approach, the platform and smartphones work in cooperative way to achieve system profit maximization. For the scheduling control, the service rate \( \alpha_j(t) \) of each smartphone \( j \) is determined by the platform. Fig. 2 shows the difference between this approach and the Stackelberg game based approach.

C. Stackelberg Game based Approach Formulation
In this section, we formulate the Stackelberg game based approach and present the details of the two stages in a backward manner.

1) STAGE II: SERVICE RATE DETERMINATION
At time \( t \), the profit of smartphone \( j \) equals to the payment from the platform minus its sensing cost, which is given by,

\[
U_j(t) = \alpha_j(t) (p_j(t) - \beta_j(t)) \sum_{i \in \mathcal{M}} s_i(t)
\]

(6)

The smartphone won’t participate in the mobile crowdsensing system when it has negative profit. That is, each smartphone must have a non-negative profit and the strategy must satisfy the property of individual rationality,

\[
U_j(t) \geq 0
\]

(7)

Given the unit price \( p_j(t) \), each smartphone calculates its optimal strategy for service rate determination:

\[
\max U_j(t)
\]

s.t. \( 0 \leq \alpha_j(t) \leq \alpha_j^{\text{max}} \)

constraints (4)(6)(7)

2) STAGE I: ADMISSION CONTROL AND PRICE DETERMINATION
At time \( t \), the profit of the platform equals to the proceeds by serving the sensing requests minus the payment to smartphones, which is given by,
\[ U_p(t) = \gamma \sum_{i \in \mathcal{M}} o_i(t) - \sum_{j \in \mathcal{N}} \alpha_j(t)p_j(t) \sum_{i \in \mathcal{M}} s_i(t) \]  

(9)

The platform calculates its optimal strategy for the amount of admitted sensing requests of each grid and unit price for each smartphone to achieve long-term profit maximization,

\[
\max \ U_p = \lim_{t \to \infty} U_p(t) = \lim_{t \to \infty} \left( \frac{1}{t} \sum_{i \in \mathcal{M}} U_p(t) \right) \\
\text{s.t.} \ p_j(t) \geq 0, \ j \in \mathcal{N} \\
0 \leq o_i(t) \leq O_i(t), \ i \in \mathcal{M} \\
\alpha_j(t) = \arg \max_{0 \leq \alpha_j \leq \alpha_j^{\text{max}}} \ U_j, \ j \in \mathcal{N}
\]

(10)

The last constraint indicates the optimal service rate determined by each smartphone through solving the optimization problem in Stage I and affected by unit price offered by the platform.

D. Cooperative Approach Formulation

In the Stackelberg game based approach, the platform and smartphones make decisions sequentially and selfishly maximize the long-term profit on their own. Therefore, the total profit of the platform may not be the largest as compared to the cooperative situation. To reveal the gap from the optimal system profit, we further formulate a cooperative system profit maximization problem, as follows. At time \( t \), the profit of the system equals to the proceeds by serving the sensing requests minus the sensing cost of all smartphones, which is given by,

\[ U_S(t) = \gamma \sum_{i \in \mathcal{M}} o_i(t) - \sum_{j \in \mathcal{N}} c_j(t) \]

(11)

The system calculates its optimal strategy for the amount of admitted sensing requests of each grid and service rate of each smartphone to achieve long-term profit maximization,

\[
\max \ U_S = \lim_{t \to \infty} U_S(t) = \lim_{t \to \infty} \left( \frac{1}{t} \sum_{i \in \mathcal{M}} U_S(t) \right) \\
\text{s.t.} \ 0 \leq o_i(t) \leq O_i(t), \ i \in \mathcal{M} \\
0 \leq \alpha_j(t) \leq \alpha_j^{\text{max}}, \ j \in \mathcal{N} \\
\text{constraints} \ (1) (2) (3) (4) (5) (6)
\]

(12)

IV. ONLINE DYNAMIC ADMISSION AND SCHEDULING ALGORITHM

In this section, we present the SAS and CAS algorithms in detail. For the convenience of the following analysis, we define some symbols as follows,

\[ p_j^i = c_j - \frac{b^2}{3a_j} \]

(13)

\[ p_j^i = c_j - \frac{b^2}{4a_j} \]

\[ p_j^i = c_j - b_j \alpha_j^{\text{max}} + a_j (\alpha_j^{\text{max}})^2 \]

\[ p_j^i = c_j - 2b_j \alpha_j^{\text{max}} + 3a_j (\alpha_j^{\text{max}})^2 \]

A. Stackelberg Game based Online Algorithm

According to the above description, the Stackelberg game in this paper can be seen as multiple one-to-one game between the platform and each smartphone. There doesn’t exist coupling relationship between smartphones and each smartphone does not need to consider the strategies of other smartphones. Next, we analyze each stage of the Stackelberg game systematically through backward induction.

1) STAGE II: SERVICE RATE DETERMINATION

From (6), the first derivative of \( U_j(t) \) to \( \alpha_j(t) \) is

\[ \frac{\partial U_j(t)}{\partial \alpha_j(t)} = (p_j(t) - c_j + 2b_j \alpha_j(t) - 3a_j \alpha_j(t)^2) \sum_{i \in \mathcal{M}} s_i(t) \]

(14)

This is a quadratic function of \( \alpha_j(t) \) and its discriminant is

\[ \Delta^2_{\alpha_j} = (4b_j^2 + 12a_j(p_j(t) - c_j)) \sum_{i \in \mathcal{M}} s_i(t)^2 \]

(15)

The value of \( \Delta^2_{\alpha_j} \) can determine the number of roots for \( \frac{\partial U_j(t)}{\partial \alpha_j(t)} = 0 \). If \( \Delta^2_{\alpha_j} \leq 0 \), that is, \( p_j(t) \leq p_j^i \), then the value of (14) is always non-negative and \( U_j(t) \) is monotone decreasing in the whole domain of definition. Therefore, the optimal strategy \( \alpha_j(t) = 0 \).

If \( \Delta^2_{\alpha_j} > 0 \), that is, \( p_j(t) > p_j^i \), then there exists two roots for \( \frac{\partial U_j(t)}{\partial \alpha_j(t)} = 0 \) and they are,

\[ x_{1,2} = \frac{-b_j \mp \sqrt{b_j^2 + 3a_j(p_j(t) - c_j)}}{3a_j} \]

(16)

The quadratic coefficient of (14) is negative. The value of (14) is non-positive in \( (-\infty, x_1] \) and \( [x_2, +\infty) \), and \( U_j(t) \) is decreasing. On the other hand, the value of (14) is non-negative in \( [x_1, x_2] \), and \( U_j(t) \) is increasing. Therefore, \( \alpha_j(t) = x_2 \) is a local maximum point of \( U_j(t) \). \( U_j(t) \) is continuous on a closed interval \( [0, \alpha_j^{\text{max}}] \), then by the extreme value theorem, global maximum point of \( U_j(t) \) exists. Furthermore, the global maximum point either must be the local maximum point \( x_2 \) (if \( x_2 \in [0, \alpha_j^{\text{max}}] \)) in the interior of the domain \( [0, \alpha_j^{\text{max}}] \), or must lie on the boundary of the domain \( [0, \alpha_j^{\text{max}}] \). The following lemma gives the condition when \( x_2 \in [0, \alpha_j^{\text{max}}] \).

Lemma 1. For \( x_2 \) defined in (16), if \( p_j(t) > p_j^i \), then \( x_2 \) exists and it satisfies that \( x_2 > b_j / 3a_j \). Furthermore, we have,

\[ \begin{cases} 
  x_2 \in [0, \alpha_j^{\text{max}}], & \alpha_j^{\text{max}} > b_j / 3a_j & \text{and} & p_j^i < p_j(t) \leq p_j^i \\
  x_2 \in [0, \alpha_j^{\text{max}}], & \text{otherwise} \end{cases} \]

(17)

Proof. When \( p_j(t) > p_j^i \), we have,
\[
x_2 = \frac{b_j + \sqrt{b^2_j + 3a_j(p_j(t) - c_j)}}{3a_j} > \frac{b_j}{3a_j}
\] (18)

Therefore, \(a^{\text{max}}_j > b_j/3a_j\), and \(p_j(t) > p^*_j\), is a necessary condition for \(x_2 \in [0, a^{\text{max}}_j]\). Combining \(x_2 \leq a^{\text{max}}_j\) with (16) and rearranging the items, we can obtain \(p_j(t) \leq p^*_j\) and further obtain (17). □

Based on the above analysis, we can obtain the optimal service rate of each smartphone in Theorem 1.

**Theorem 1**: For the Stackelberg game based approach, given the unit price \(p_j(t)\) and sensing region \(s_j(t)\) of each smartphone \(j\), the optimal service rate \(\alpha_j(t)\) satisfies,

a) When \(a^{\text{max}}_j < b_j/2a_j\), we have

\[
\alpha_j(t) = \begin{cases} 0, & p_j(t) < p^*_j \\ a^{\text{max}}_j, & \text{otherwise} \end{cases}
\] (19)

b) When \(a^{\text{max}}_j \geq b_j/2a_j\), we have

\[
\alpha_j(t) = \begin{cases} 0, & p_j(t) < p^*_j \\ a^{\text{max}}_j, & p_j(t) > p^*_j \\ x_2, & \text{otherwise} \end{cases}
\] (20)

**Proof.** Part a). For the situation where \(a^{\text{max}}_j < b_j/2a_j\) and \(x_2 \in [0, a^{\text{max}}_j]\), by rearranging the items of (16), we have,

\[
p_j(t) = c_j - 2b_jx_j + 3a_jx_j^2
\] (21)

Plugging (21) into (6), we can obtain,

\[
U_j(t)_{\alpha_j(t)} = x_j(-b_j + 2a_jx_j + 3a_jx_j^2)
\]

\[
> 0 = U_j(t)_{\alpha_j(t)}
\] (22)

Where \(U_j(t)_{\alpha_j(t)}\) is corresponding value \(U(t)\) with respect to \(\alpha_j(t)\). We can conclude from (22) that the function value \(U(t)\) under local maximum point is smaller than the corresponding value \(U(t)\) with respect to \(\alpha_j(t)=0\). Therefore, the global maximum point must lie on the boundary of the domain \([0, a^{\text{max}}_j]\). Furthermore, for the situation where \(a^{\text{max}}_j < b_j/2a_j\), we can easily obtain,

\[
\begin{align*}
U_j(t)_{\alpha_j(t)} &> U_j(t)_{\alpha_j(t)}^{\text{max}}, & p_j(t) < p^*_j \\
U_j(t)_{\alpha_j(t)} &\leq U_j(t)_{\alpha_j(t)}^{\text{max}}, & \text{otherwise}
\end{align*}
\] (23)

This demonstrates that for the situation where \(a^{\text{max}}_j < b_j/2a_j\), (19) holds. We complete the proof of part a) of Theorem 1.

Part b). Considering the situation where \(x_2 \in [0, a^{\text{max}}_j]\), then \(U_j(t)\) is decreasing in the interval \([x_2, a^{\text{max}}_j]\). Therefore, the global maximum point must be \(x_2\) or \(0\). Suppose that the global maximum point is \(x_2\). Then according to (22), we have,

\[
U_j(t)_{\alpha_j(t)} = x_2(-b_j + 2a_jx_2 + 3a_jx_2^2)
\]

\[
> 0 = U_j(t)_{\alpha_j(t)}
\] (24)

\[
x_2 \geq \frac{b_j}{2a_j}
\]

Combining (16)-(17) with (24), we can obtain the condition when \(x_2\) is global maximum point as follows,

\[
p_j^* < p_j(t) < p^*_j
\] (25)

Furthermore, for the situation where \(x_2 \in [0, a^{\text{max}}_j]\) and the global maximum point is 0, \(p_j(t)\) satisfies,

\[
p_j^* < p_j(t) < p^*_j
\] (26)

Recall that when \(p_j(t) \leq p^*_j\), \(U_j(t)\) is monotone decreasing in the whole domain \([0, a^{\text{max}}_j]\) and the optimal service rate \(\alpha_j(t)\). So if \(p_j(t) < p^*_j\), then the optimal service rate \(\alpha_j(t)\) is 0.

With regard to the situation where \(p_j(t) > p^*_j\), \(x_2\) is outside the domain \([0, a^{\text{max}}_j]\) and the global maximum point must lie on the boundary of the domain \([0, a^{\text{max}}_j]\). In this situation, we have,

\[
U_j(t)_{\alpha_j(t)} = x_2(-b_j + 2a_jx_2 + 3a_jx_2^2)
\]

\[
> (p_j^* - p_j(t)) \sum_{i \in M} s_j(t)
\]

\[
= \alpha_j^{\text{max}}x_2\sum_{i \in M} s_j(t)
\]

\[
\geq 0 = U_j(t)_{\alpha_j(t)}
\] (27)

This indicates that the global maximum point is \(a^{\text{max}}_j\) when \(p_j(t) > p^*_j\). □

According to Theorem 1, to obtain the optimal service rate, each smartphone only needs to make several very simple calculations and comparisons. Furthermore, each smartphone can make decisions on its own. After reforming (19)-(20), we can obtain the relationship between the unit price \(p_j(t)\) and the service rate \(\alpha_j(t)\). To see the service determination more clearly, we analyze the values of a specific smartphone at a specific time slot by taking some typical parameters \(a, b, c\) and \(a^{\text{max}}\) numerically. As shown in Fig.3 and Fig.4, the solid lines plot the unit price \(p\) and cost \(\beta\) with respect to the corresponding optimal service rate \(\alpha\), while the service rate of dashed lines will never be an optimal value.

We find from Fig.3 that for the situation where \(a^{\text{max}} < b/2a\) and \(p < p_3\), whatever the value of the service rate is, the price is always smaller than the cost. Therefore, the smartphone has to set the service rate to zero to guarantee the individual rationality. That is, the smartphone will not serve any sensing request. When \(p > p_3\), the larger the service rate \(\alpha\) is, the larger the gap between the price and the cost is. The smartphone serves the sensing requests at maximum service rate to maximize its utility. In a word, all the values of optimal service rate are located at the bound of the range of \(\alpha\).
As shown in Fig. 4, when \( \alpha_{\text{max}} > b/2a \) and \( p < p_2 \), the optimal service rate is zero as the price is always smaller than the cost no matter how many sensing requests the smartphone serves. Consider a specific point \((\alpha^*, p)\) on the solid line of \( p \) under the situation where \( p \geq p_2 \). If the smartphone executes the sensing requests at a lower service rate than \( \alpha^* \), it will earn more by executing one unit of sensing requests. By contrast, if the smartphone serves at a higher service rate than \( \alpha^* \), it will earn less by serving one unit of sensing requests. However, in both situations, the overall utility of the smartphone is smaller than the situation where \( \alpha = \alpha^* \). Moreover, \( \alpha \in (0, b/2a) \) will never be an optimal solution.

The one-step conditional Lyapunov drift is
\[
\Delta^L_c(t) \leq \mathbb{E}[L(t+1) - L(t) | Q(t)]
\]

Where \( Q(t) \) represents the vector of sensing request queue. Furthermore, we define the drift-minus-profit expression of the platform as,
\[
\Delta^L_p(t) = \Delta^L_c(t) - V\mathbb{E}[U_p(t) | Q(t)]
\]

Where \( V \) is a non-negative constant that allows tradeoff between sensing request backlog and profit.

The following lemma gives an upper bound of drift-minus-profit expression of the platform.

**Lemma 2.** For each time slot \( t \), under any feasible admission control and price determination algorithm, we have,
\[
\Delta^L_p(t) \leq B + \mathbb{E}\left[ \sum_{i \in M} \alpha_i(t)(Q_i(t) - V\gamma) | Q(t) \right] 
+ \mathbb{E}\left[ \sum_{j \in V} (V_{ij}(t)\alpha_j(t) + \sum_{i \in M} s_i(t) - \alpha_j(t)) \sum_{i \in M} Q_j(t)s_i(t) | Q(t) \right]
\]

Where \( B \) is defined as,
\[
B = \frac{1}{2} \sum_{i,j \in M} ((O_{ij}^{\text{max}})^2 + \sum_{j \in V} (\alpha_j^{\text{max}})^2)
\]

**Proof.** Squaring both sides of (2) and using the facts that \( \max[Q_i(t)-r_i(t),0]^2 \leq (Q_i(t) - r_i(t))^2 \) yields,
\[
Q_i^2(t) + 2Q_i(t)(Q_i(t) - r_i(t)) + (Q_i(t)-r_i(t))^2 \leq Q_i^2(t) + 2Q_i(t)(Q_i(t) - r_i(t)) + (\sum_{j \in V} \alpha_j^{\text{max}})^2 + \sum_{j \in V} (\alpha_j^{\text{max}})^2
\]

Summing the above inequality over \( i \in M \) and combining it with (30), we can further obtain,
\[
\Delta^L_p(t) \leq B + \mathbb{E}\left[ \sum_{i \in M} Q_i(t)(\alpha_i(t) - r_i(t)) | Q(t) \right] 
- V\mathbb{E}[U_p(t) | Q(t)]
\]

Plugging (1) and (9) into (34), and rearranging the terms, we can see that Lemma 2 holds. □

The right-hand-side of (31) is an upper bound of drift-minus-profit expression of the platform and minimizing it is equivalent to solving the following problems:

\[
\min \quad \alpha_i(t)(Q_i(t) - V\gamma)
\]
\[s.t. \quad 0 \leq \alpha_i(t) \leq O_i(t)
\]

and,
\[
\begin{align*}
\min Y_j(t) &= Vp_j(t)\alpha_j(t) \sum_{i \in M} s_{ij}(t) - \alpha_j(t) \sum_{i \in M} Q_i(t)s_{ij}(t) \\
\text{s.t.} \quad p_j(t) &\geq 0 \quad \text{constraint \ (8)} \\
\alpha_j(t) &= \arg \max \bar{U}_j \\
\end{align*}
\]

(36)

It is very easy to get the optimal solution of (35),
\[
o_j(t) = \begin{cases} O_j(t), & Q_j(t) - V\gamma < 0 \\ 0, & \text{otherwise} \end{cases}
\]

(37)

This means that the platform can make admission control for each grid independently and easily. When the sensing requests queue backlog of grid \(i\) keeps below the threshold \(V_j\), all the sensing requests will be admitted by the platform. Otherwise, none of sensing requests are admitted. This admission control guarantees the stability constraint. More analysis will come in Section V.

The optimal solution of the optimization problem in (36) is presented in Theorem 2.

**Theorem 2**: For the Stackelberg game based approach, given the sensing region \(s_j(t)\) of each smartphone \(j\) and queue backlog \(Q_i(t)\), the optimal price \(p_j(t)\) offered by the platform satisfies,

a) When \(\alpha_{j}^{\max} < b_j/2a_j\), we have
\[
p_j(t) = \arg \min_{p_j(t) \in [0, p_j^1]} \{Y_j(t) | \alpha_j(t)\} \quad (38)
\]

b) When \(\alpha_{j}^{\max} \geq b_j/2a_j\), we have
\[
p_j(t) = \arg \min_{p_j(t) \in [0, p_j^1]} \{Y_j(t) | \alpha_j(t)\} \quad (39)
\]

Where,
\[
p_j^1 = c_j - 2b_j\alpha_j^d + 3a_j(\alpha_j^d)^2 \\
\alpha_j^d = \max\left(\min\left(y_j, \alpha_j^{\max}\right), b_j / 2a_j\right) \\
Y_{1,2} = \begin{cases} 4Vb_j \sum_{i \in M} s_{ij}(t) + \sqrt{\Delta_{i,j}^R}, & \Delta_{i,j}^R > 0 \\ -18Va_j \sum_{i \in M} s_{ij}(t), & \Delta_{i,j}^R = 0 \\ -\infty, & \text{otherwise} \end{cases}
\]

(42)

(43)

Proof. Part a). When \(\alpha_{j}^{\max} < b_j/2a_j\) and \(p_j(t) < p_j^1\), we have \(a_j(t) = 0\) and \(Y_j(t) = 0\). That is, no matter how high the price is, the corresponding service rate is always zero. Therefore, we set \(p_j(t)\) to zero. When \(\alpha_{j}^{\max} < b_j/2a_j\) and \(p_j(t) \geq p_j^1\), we have \(a_j(t) = \alpha_{j}^{\max}\) and
\[
Y_j(t) = Vp_j(t)\alpha_{j}^{\max} \sum_{i \in M} s_{ij}(t) - \alpha_{j}^{\max} \sum_{i \in M} Q_i(t)s_{ij}(t) \quad (44)
\]

\(Y_j(t)\) is a linear function of \(p_j(t)\) and its minimum point is \(p_j^1\). Combining these two situations yields (38).

Part b). Similar to the situation where \(\alpha_{j}^{\max} < b_j/2a_j\) and \(p_j(t) < p_j^1\), we have \(Y_j(t) = 0\) and \(p_j(t) = 0\) when \(\alpha_{j}^{\max} \geq b_j/2a_j\) and \(p_j(t) \geq p_j^1\). Similar to the situation where \(\alpha_{j}^{\max} < b_j/2a_j\) and \(p_j(t) \geq p_j^1\), we can obtain the optimal price \(p_j(t) = p_j^1\) when \(\alpha_{j}^{\max} \geq b_j/2a_j\) and \(p_j(t) \geq p_j^1\). The corresponding value of \(Y_j(t)\),
\[
Y_j(t) = Vp_j(t)\alpha_{j}^{\max} \sum_{i \in M} s_{ij}(t) - \alpha_{j}^{\max} \sum_{i \in M} Q_i(t)s_{ij}(t) \quad (45)
\]

As for the situation where \(\alpha_{j}^{\max} \geq b_j/2a_j\) and \(p_j(t) < p_j^1\), \(a_j(t)\) is a monotone increasing function of \(p_j(t)\) and its domain is \([b_j/2a_j, \alpha_{j}^{\max}])\). We can derive from (20) and (36),
\[
Y_j(t) = Vc_j\alpha_j(t) - 2b_j\alpha_j^d(t) + 3a_j\alpha_j^d(t) \sum_{i \in M} s_{ij}(t) - \alpha_j(t) \sum_{i \in M} Q_i(t)s_{ij}(t) \quad (46)
\]

The first derivative of \(Y_j(t)\) to \(a_j(t)\) is
\[
\frac{\partial Y_j(t)}{\partial a_j(t)} = Vc_j \sum_{i \in M} s_{ij}(t) - \sum_{i \in M} Q_i(t)s_{ij}(t) \quad (47)
\]

This is a quadratic function and its quadratic coefficient is positive. The discriminant \(\Delta_{c_j}^R\) is shown in (43) and the value of \(\Delta_{c_j}^R\) can determine the number of roots for \(\partial Y_j(t)/\partial a_j(t) = 0\). If \(\Delta_{c_j}^R \leq 0\), then the value of (47) is always no smaller than zero and \(Y_j(t)\) is monotone increasing in the domain \([b_j/2a_j, \alpha_{j}^{\max}])\). Therefore, the optimal strategy \(a_j(t) = b_j/2a_j\). Let \(y_j\) be \(-\infty\), then \(a_j(t) = a_j^d\).

If \(\Delta_{c_j}^R > 0\), then there exists two roots for \(\partial Y_j(t)/\partial a_j(t) = 0\), as shown in (42). Obviously, \(y_j < b_j/2a_j\). The value of (47) is non-negative in \((-\infty, y_j]\) and \([y_j, +\infty)\), and \(Y_j(t)\) is increasing. On the other hand, the value of (47) is non-positive in \([y_j, +\infty)\), and \(Y_j(t)\) is decreasing. We divide this situation into the following three situations: (1) If \(\alpha_{j}^{\max} \leq y_j\), then \(Y_j(t)\) is decreasing in \([b_j/2a_j, \alpha_{j}^{\max}])\) and \(a_j(t) = \alpha_{j}^{\max}\). (2) If \(b_j/2a_j < y_j \leq \alpha_{j}^{\max}\), then \(Y_j(t)\) is decreasing in \([b_j/2a_j, y_j]\) and \(Y_j(t)\) is increasing in \([y_j, \alpha_{j}^{\max}])\). Under this condition, \(a_j(t) = y_j\). (3) If \(y_j < b_j/2a_j\), then \(Y_j(t)\) is increasing in \([b_j/2a_j, \alpha_{j}^{\max}])\) and \(a_j(t) = b_j/2a_j\). In summary, we have \(a_j(t) = a_j^d\).

Finally, we compare the value of \(Y_j(t)\) when \(a_j(t) = 0\) with the value of \(Y_j(t)\) when \(a_j(t) = a_j^d\), and yield (39).

We can see from Theorem 2 that for each smartphone \(j\), the price \(p_j(t)\) is determined by the sensing region \(s_j(t)\), the backlogs in its sensing region and the control parameter \(V\). The platform can obtain price determination \(p_j(t)\) for each smartphone \(j\) through several simple calculations and comparisons. For example, when \(\alpha_{j}^{\max} < b_j/2a_j\), we only need to compare \(Vp_j(t)\alpha_{j}^{\max} \sum_{i \in M} s_{ij}(t) - \alpha_{j}^{\max} \sum_{i \in M} Q_i(t)s_{ij}(t)\) with zero.

In summary, at the beginning of each time slot \(t\), the platform makes admission control according to (37) and determines the amount of admitted sensing requests for each grid. The platform calculates the price \(p_j(t)\) according to (38-
(39) and offers it to each smartphone. Upon receiving the price, each smartphone responds with the service rate to maximize its own profit according to (19)-(20). Finally, the platform updates queue backlogs according to (2). The Stackelberg game based online algorithm is summarized in Algorithm 1. We will show in Section VI that this online algorithm achieves a close-to-offline-optimum time-averaged profit for the platform, while guaranteeing the stability constraint.

**Algorithm 1:** SAS: online Stackelberg game based dynamic Admission and Scheduling algorithm

1. Set \( Q(0) = 0 \), \( i \in M \)
2. for each time slot \( t \) do
3. \( i \in M \) do
4. The platform determines \( a_i(t) \) according to (37)
5. end for
6. for \( j \in N \) do
7. if \( \alpha_j^\infty < \beta_j/20 \) do
8. The platform determines \( p_j(t) \) according to (38)
9. Smartphone \( j \) determines \( a_j(t) \) according to (19)
10. else
11. The platform determines \( p_j(t) \) according to (39)
12. Smartphone \( j \) determines \( a_j(t) \) according to (20)
13. end if
14. end for
15. for \( i \in M \) do
16. The platform calculates \( r_i(t) \) according to (1)
17. The platform updates \( Q(t) \) according to (2)
18. end for
19. end for

**B. Cooperative Online Algorithm**

Similarly, for the cooperative approach, we define the drift-minus-profit expression of the system as,

\[
\Delta_c^L(t) \triangleq \Delta^L(t) - \mathbb{V} \mathbb{E}[U_j(t) | Q(t)]
\]

Similar to Lemma 2, we can obtain an upper bound of the drift-minus-profit expression of the system,

\[
\Delta_c^L(t) \leq B + \mathbb{E} \left\{ \sum_{i \in M} o_i(t)(Q(t) - V \gamma) | Q(t) \right\} + \mathbb{E} \left\{ \sum_{j \in V} (V \beta_j(t) \alpha_j(t) \sum_{i \in M} s_j(t) - \alpha_j(t) \sum_{i \in M} Q(t)s_j(t)) | Q(t) \right\}
\]

Minimizing this upper bound is equivalent to solving the following problems:

\[
\begin{align*}
\text{min} & \quad o_i(t)(Q(t) - V \gamma) \\
\text{s.t.} & \quad 0 \leq o_i(t) \leq O_i(t) \\
\end{align*}
\]

and,

\[
\begin{align*}
\text{min} & \quad Z_j(t) = V \beta_j(t) \alpha_j(t) \sum_{i \in M} s_j(t) - \alpha_j(t) \sum_{i \in M} Q(t)s_j(t) \\
\text{s.t.} & \quad 0 \leq \alpha_j(t) \leq \alpha_j^{\max} \\
\end{align*}
\]

The solution of (50) is the same as (35). The optimal solution of the problem in (51) is presented in Theorem 3.

**Theorem 3:** For the cooperative approach, given the sensing region \( s_j(t) \) of each smartphone \( j \) and queue backlog \( Q(t) \), the optimal service rate \( \alpha_j(t) \) satisfies,

- a) When \( \Delta_c^L(t) \leq 0 \), we have,
  \[
  \alpha_j(t) = 0
  \]
- b) When \( \Delta_c^L(t) > 0 \), we have,
  \[
  \alpha_j(t) = \arg \min \left( Z_j(t) \left| \alpha_j(t) \in [0, \alpha_j^{\infty}] \right. \right)
  \]

Where,

\[
Z_i^d = \min\{z_i^d, \alpha_j^{\max}\}
\]

This is a quadratic function and its quadratic coefficient is positive. The discriminant \( \Delta_c^L \) is presented in (56) and the value of \( \Delta_c^L \) can determine the number of roots for \( \frac{\partial Z_i(t)}{\partial \alpha_i(t)} = 0 \).

Part a). If \( \Delta_c^L \leq 0 \), then the value of (52) is always no smaller than zero and \( Z_i(t) \) is monotone increasing in the whole domain. Therefore, the optimal strategy \( \alpha_i(t) = 0 \).

Part b). If \( \Delta_c^L > 0 \), then there exists two roots for \( \frac{\partial Z_i(t)}{\partial \alpha_i(t)} = 0 \), as shown in (55). The value of (52) is non-negative in \((-\infty, z_2)\) and \([z_2, +\infty)\), and \( Z_i(t) \) is increasing. On the other hand, the value of (52) is non-positive in \([z_1, z_2]\), and \( Z_i(t) \) is decreasing. Therefore, \( \alpha_i(t) = z_2 \) is a local minimum point of \( Z_i(t) \). \( Z_i(t) \) is continuous on a closed interval \([0, \alpha_j^{\infty}]\), then by the extreme value theorem, global minimum point of \( Z_i(t) \) exists. Furthermore, the global minimum point of either must be a local minimum point \( z \) (if \( z \in [0, \alpha_j^{\infty}] \)) in the interior of the domain \([0, \alpha_j^{\infty}]\), or must lie on the boundary of the domain. Then, we have: (1) If \( z > \alpha_j^{\infty} \), then the global minimum point must lie on the boundary of \([0, \alpha_j^{\infty}]\). (2) If \( z \leq \alpha_j^{\infty} \), then \( Z_i(t) \) is increasing in \([z_2, \alpha_j^{\max}]\) and the global minimum point must be 0 or \( z_2 \). Combining these situations yields (53). \( \square \)

In cooperative online algorithm, the admission control and scheduling control are non-coupled. At each time slot, the platform calculates the admission decision according to (37), and the scheduling decision according to (52)-(53). The cooperative online algorithm is summarized in Algorithm 2. The solution is very easy to obtain and suitable for online implementation. More analysis will be presented in Section V.
Algorithm 2 CAS: online Cooperative dynamic Admission and Scheduling algorithm

1. Set $Q(0) = 0$, $i \in M$
2. for each time slot $t$ do
3. for $i \in M$ do
4. The platform determines $o_i(t)$ according to (37)
5. end for
6. for $j \in N$ do
7. if $\Delta_j \leq 0$ do
8. The platform determines $o_j(t)$ according to (52)
9. else
10. The platform determines $o_j(t)$ according to (53)
11. end if
12. end for
13. for $i \in M$ do
14. The platform calculates $r_i(t)$ according to (1)
15. The platform updates $Q(t)$ according to (2)
16. end for
17. end for

V. PERFORMANCE ANALYSIS

The following theorem shows how SAS stabilizes the queues and achieves the optimal profit.

**Theorem 4.** For the location aware admission and scheduling problem formulated in Section III, we have:

a) Our proposed SAS algorithm with any $V > 0$ stabilizes the system, with a resulting backlog bound given by,

$$0 \leq Q(t) \leq V \gamma + O_{max}^t$$  \hfill (58)

b) The gap between the time average profit $\bar{U}_{p, opt}^S$ of the platform achieved by SAS and the optimal profit $\bar{U}_{p, max}^\opt$ is smaller than $B/V$,

$$\bar{U}_{p, max}^\opt \geq \bar{U}_{p, max}^\opt - \frac{B}{V}$$  \hfill (59)

**Proof.** Part a). We prove part a) by mathematical induction.

**Induction Basis:** For $t=0$ and any $i \in M$, (58) is true,

$$0 \leq Q(0) = 0 \leq V \gamma + O_{max}^0$$

**Induction Steps:** Suppose that for some $t=r$, we already have $0 \leq Q(t+1) \leq V \gamma + O_{max}^t$. If $0 \leq Q(t) \leq V \gamma$, then according to (2), we have $0 \leq Q(t) \leq V \gamma + O_{max}^t$. If $V \gamma \leq Q(t) \leq V \gamma + O_{max}^t$, then there is no sensing request admitted in the system at grid $i$, that is, $o_i(t) = 0$. So $Q(t+1)$ is no bigger than $Q(t)$. $0 \leq Q(t+1) \leq Q(t) \leq V \gamma + O_{max}^t$.

**Conclusion:** We can conclude that (58) holds for any $t$ and any $i \in M$.

Based on (58), taking expectations, summing over time, dividing $t$ and taking limits $t \rightarrow \infty$ yields,

$$0 \leq \lim_{t \rightarrow \infty} \sup_{t \geq 0} \frac{1}{t} \sum_{t=0}^{t} \mathbb{E}Q(t) \leq V \gamma + O_{max}^t < \infty$$  \hfill (60)

We can see that all queues of the mobile crowdsensing system are stable and the system is stable.

Part b). The proof of (59) requires the following lemma.

**Lemma 2.** If the problem formulated in Section III is feasible, then for any $\delta > 0$, for arbitrary sensing request arrival rate of each grid and available smartphones in the system, there exists a randomized stationary control algorithm $\sigma$ that yields the following steady values:

$$\mathbb{E}[\bar{U}_p^\sigma(t)] \geq \bar{U}_p^\opt - \delta$$  \hfill (61)

$$\mathbb{E}[o_i^\sigma(t)] \leq \mathbb{E}[r_i^\sigma(t)] + \delta$$  \hfill (62)

Where the randomized stationary control algorithm is the algorithm that makes decisions independent of the current sensing request backlogs $Q(t)$. This lemma derives from the Theorem 4.5 of [41] and we omit the proof for brevity. □

Now we fix $\delta > 0$, and consider the randomized stationary control algorithm $\sigma$ that yields (61)-(62). Since the resulting values $\bar{U}_p^\sigma(t)$, $o_i^\sigma(t)$, $r_i^\sigma(t)$ of algorithm $\sigma$ are independent of current backlogs, we can obtain,

$$\mathbb{E}[\bar{U}_p^\sigma(t) | Q(t)] = \mathbb{E}[\bar{U}_p^\sigma(t)]$$  \hfill (63)

$$\mathbb{E}[o_i^\sigma(t) | Q(t)] = \mathbb{E}[o_i^\sigma(t)]$$  \hfill (64)

$$\mathbb{E}[r_i^\sigma(t) | Q(t)] = \mathbb{E}[r_i^\sigma(t)]$$  \hfill (65)

For every time slot $t$, SAS minimizes right-hand-side of (31). So we have,

$$\Delta_p^t(t) \leq B + \mathbb{E}[\sum_{i \in M} Q(t) o_i^\sigma(t) - r_i^\sigma(t) | Q(t)] - V \mathbb{E}[\bar{U}_p^\sigma(t) | Q(t)]$$  \hfill (66)

Combining (61)-(62) with (66) and taking a limit as $\delta \rightarrow 0$ yields,

$$\Delta_p^t(t) \leq B - V \bar{U}_p^\opt$$  \hfill (67)

Now fix any time slot $t$. Since (67) holds for time slot $t$, we can take expectations over $Q(t)$ to yield,

$$\mathbb{E}[L(t+1) - L(t)] - V \mathbb{E}[U_p(t)] \leq B - V \bar{U}_p^\opt$$  \hfill (68)

Summing over $t \in \{0, 1, \ldots, t \}$ for some $t > 0$ yields,

$$\mathbb{E}[L(t)] - \mathbb{E}[L(0)] - \sum_{t=0}^{t} \mathbb{E}[U_p(t)] \leq t(B - V \bar{U}_p^\opt)$$  \hfill (69)

Dividing by $Vt$, rearranging terms and using the fact that $L(0) = 0$, it is easy to show that (69) directly implies the following inequality for all $t > 0$:

$$- \frac{1}{t} \sum_{t=0}^{t} \mathbb{E}[U_p(t)] \geq P^opt - \frac{B}{V} + \frac{\mathbb{E}[L(t)]}{Vt} \geq \bar{U}_p^\opt - \frac{B}{V}$$  \hfill (70)

Taking a limit of the above as $t \rightarrow 0$ proves (59). □

Furthermore, we can obtain how CAS stabilizes the queues and achieves the optimal profit.

**Theorem 5.** For the location aware admission and scheduling problem formulated in Section III, we have:

a) Our proposed CAS algorithm with any $V > 0$ stabilizes the system, with a resulting backlog bound given by,

$$0 \leq Q(t) \leq V \gamma + O_{max}^t$$  \hfill (71)
b) The gap between the time average profit $\bar{U}_{P}^{\text{CAS}}$ of the platform achieved by CAS and the optimal profit $\bar{U}_{S}^{\text{opt}}$ is smaller than $B/V$.

$$\bar{U}_{P}^{\text{CAS}} \geq \bar{U}_{S}^{\text{opt}} - \frac{B}{V} \quad (72)$$

The proof of Theorem 5 is similar to Theorem 4 and we omit the proof for brevity. □

VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed SAS and CAS algorithms through simulations based on both synthetic and real trace.

A. Simulation Settings

In our simulations, we consider two scenarios. In the first scenario, each smartphone moves according to the classical random walk (RW) movement model. In particular, the platform serves a virtual city with size $10\text{km} \times 10\text{km}$. A hotspot is located at the center of the city. The grids near to the center of the hotspot have larger arrival rate than those far from it. The simulation in the second scenario uses the Cabspotting dataset, which traces San Francisco’s taxi cabs as they travel throughout the Bay Area. Suppose that each cab owns a smartphone participating in the system. The platform serves San Francisco, Daly City, San Bruno and the area surrounded by these three cities. There are two hotspots, one located at downtown area of San Francisco and the other located at San Francisco International Airport. The former hotspot has larger arrival rate.

In both scenarios, each grid size is set as $200\text{m} \times 200\text{m}$. The sensing requests arrive at grid $i$ according to the uniform distribution with average rate $\bar{O}_i$. The average value of all $\{\bar{O}_i, i \in M\}$ is set as $2$. The average sensing request arrival distribution of these two scenarios is shown in Fig.5 and Fig.6.

There are totally 500 smartphones participating in the system. We simply define the sensing region of each smartphone $j$ as a circle $C_j$ with the center the location of $j$ and the radius randomly picked from $[400\text{m}, 800\text{m}]$. The smartphone-specific constant $\alpha_{j}^{\text{max}}$ is randomly selected from $[0.8, 1.2]$. The smartphone-specific constants $a_j$, $b_j$ and $c_j$ are uniformly and randomly selected from $[0.4, 0.6]$. The remaining parameters are set as follows: $\gamma = 1$, $V \in \{20, 25, 30, 35, 40, 50, 60, 70, 80, 100\}$. All the following results are obtained by running simulations for $10000$ time slots.

B. Profit-backlog Tradeoff

Theorem 4 and Theorem 5 show the tradeoffs between the profit and queue backlog. By adjusting the value of parameter $V$, the SAS and CAS algorithms can achieve $O(1/V)$-optimal average profit, while the sensing request backlog is bounded by $O(V)$. In this subsection, we verify the conclusions of Theorem 4 and Theorem 5 through simulations. We plot the time average sensing request backlog of all grids and the time average profit with the growth of $V$ in Fig. 7 and Fig. 8 respectively. On one hand, we can see that the time average sensing request backlog increases linearly with growth of $V$ and is no more than $\gamma + O_{\text{max}}$, where $O_{\text{max}}$ is the average value of all $\{O^{\text{max}}_i, i \in M\}$. This is consistent with part a) of Theorem 4 and Theorem 5. On the other hand, as $V$ increases, the time average profit achieved by SAS and CAS algorithms improves sharply at first and then converges quickly to very close to the optimal profit. This verifies the part b) of Theorem 4 and Theorem 5.

FIGURE 6. Average sensing request arrival distribution with Cabspotting dataset.

FIGURE 5. Average sensing request arrival distribution with RW model.

FIGURE 7. Time average sensing request backlog vs. $V$. 
C. Stackelberg Equilibrium

In SAS, none of the stakeholders, including the platform and smartphones, can improve his utility by unilaterally changing its current strategy.

1) STAGE II: PROFIT MAXIMIZATION OF EACH SMARTPHONE

Given the unit price $p_j$, the strategy in (19)-(20) maximizes the individual profit of each smartphone. We compare our strategy with the following three strategies: (i) $\alpha_j = 0.2$; (ii) $\alpha_j = 0.5$; (iii) $\alpha_j = 0.8$, $j \in \mathcal{N}$. As shown in Fig.9, we plot the results when the given price varies from 0.3 to 0.5. We can see that our strategy always outperforms other strategies. Furthermore, the profit of our strategy is always no smaller than zero and satisfies the property of individual rationality. However, the profit of some smartphones under the other strategies has fallen below zero.

2) STAGE I: PROFIT MAXIMIZATION OF THE PLATFORM

The results in Fig.10 and Fig.11 illustrate the comparison between non-optimal profits and the optimal Stackelberg profit of the platform. In the simulation, the platform announces a fixed price ranging from 0.3 to 0.5 and each smartphone responds with the service rate according to (19)-(20). We can see that the optimal Stackelberg strategy always outperforms other strategies. For example, when $V=30$, the profit of optimal Stackelberg strategy is 2325 while the profit for $p=0.4$ is 1643 with RW model. Hence, as both stakeholders have reached their maximum profits using Stackelberg game, they will perform the mobile crowdsensing activity according to these optimal strategies and thus achieve a Stackelberg Equilibrium.

We also find that the profit of the platform increases with the growth of the fixed price before a certain threshold. When it exceeds this threshold, the corresponding profit begins to fall. We can see this phenomenon more clearly in Fig.12. This is because higher price leads to larger service rate, thereby yielding more profit. However, when the price exceeds a certain threshold, the payment becomes a burden.
D. Competition Loss

In this section, we compare the performance of SAS with that of CAS, and explore the impact of sensing capacity on the profit difference between these two algorithms. Define the competition loss as the difference between the system profit under CAS and the aggregate profit under SAS,

$$U_t = \frac{1}{T} \sum_{t=0}^{T-1} (U_i(t) - U_R(t) - \sum_{j \in V} U_j(t))$$

(72)

As shown in Fig.13 and Fig.14, we plot the competition loss with different ratios to the default sensing capacity. We find that the CAS always outperforms SAS. This is because the competition between the platform and smartphones may lead to the profit loss of each stakeholder. On the other hand, the competition loss becomes smaller as the sensing capacity becomes smaller. Further in some certain situations, the profit of SAS is even very close to the profit of CAS. For example, the competition loss is only 8.2 with Cabspotting and $V=20$. In these situations, SAS considers the competition between the platform and smartphones without much loss of the profit.

VII. CONCLUSION

In this paper, we explore the dynamic admission and scheduling control problem in mobile crowdsensing system with stability constraint. In particular, the arrival of sensing requests and the location of each smartphone are dynamic. We first use the Stackelberg game to model the interaction procedure between the platform and smartphones. Then we introduce the Lyapunov optimization to address this problem and propose SAS algorithm. SAS can be implemented based on the currently available information and reach Stackelberg equilibrium. None of the stakeholders, including the platform and smartphones, can improve his utility by unilaterally changing its current strategy. Next, we design CAS algorithm to consider the situation where the smartphones operate in a cooperative way. Theoretical analysis and extensive results based on synthetic and real trace demonstrate that the SAS and CAS algorithms can achieve $O(1/V)$-optimal average profit, while the sensing request backlog is bounded by $O(V)$. Finally, we introduce the concept of competition loss to compare the performance of SAS with that of CAS. CAS always outperforms SAS, and in some certain situations, the profit of SAS is very close to that of CAS.

REFERENCES

REFERENCES


Zhifei Wang was born in Hebei, China, in 1990. He received the B. S. and M. S. degrees in information system engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2013 and 2015 respectively. He is currently studying for a PhD degree in management science and engineering in NUDT. His research interests include incentive mechanism design and resource allocation for participatory sensing, mobile crowdsourcing, and people-centric sensing.

Haohao Zhou was born in Henan, China, in 1988. He received B.Sc. and M.Sc. degrees from the National University of Defense Technology, Hunan, China, and later received a Ph.D. degree in management science in 2016. He is currently a researcher in Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology. His research fields include data engineering and information management.

Yongpeng Zhao was born in 1978. He received a Ph.D. degree in 2011. He is currently a Senior Engineer in Strategic Planning Office, Beijing, China. His research fields include data engineering and information system.
Yahui Wu was born in Henan, China, in 1983. He received the B. S. and M. S. degrees in information system engineering from National University of Defense Technology (NUDT), Changsha, China, in 2006 and 2008, respectively. Then, he received his PhD degree from NUDT in 2013. He is currently a lecturer in the Science and Technology on Information Systems Engineering Laboratory of NUDT. His research interests include mobile crowdsourcing, opportunistic sensing and opportunistic network.

Su Deng was born in Hunan, China, in 1963. He received the B. S. degree from Naval University of Engineering, Wuhan, China, in 1983. Then he received the PhD degree from National University of Defense Technology (NUDT), in 2001. He is currently a professor in Science and Technology on Information Systems Engineering Laboratory of NUDT. His research interests include mobile crowdsourcing system, cyber-physical system, and information management.

Hongbin Huang was born in Jiangsu, China, in 1975. He received the B. S. and Ph. D. degrees in information system engineering from National University of Defense Technology (NUDT), Changsha, China, in 1995 and 2006, respectively. He is currently a professor in the Science and Technology on Information Systems Engineering Laboratory of NUDT. His research interests include crowdsourcing service, focusing service and decision making.