Summer 8-7-2018

Incentive Mechanism Design in Mobile Crowdsensing Systems

Zhuojun Duan

INCENTIVE MECHANISM DESIGN IN MOBILE CROWDSENSING SYSTEMS

by

Zhuojun Duan

Under the Direction of Zhipeng Cai, Ph.D. and Wei Li, Ph.D.

ABSTRACT

In the past few years, the popularity of *Mobile Crowdsensing Systems* (MCSs) has been greatly prompted, in which sensory data can be ubiquitously collected and shared by mobile devices in a distributed fashion. Typically, a MCS consists of a cloud platform, sensing tasks, and mobile users equipped with mobile devices, in which the mobile users carry out sensing tasks and receive monetary rewards as compensation for resource consumption (*e.g.*, energy, bandwidth, and computation) and risk of privacy leakage (*e.g.*, location exposure). Compared with traditional mote-class sensor networks, MCSs can reduce the cost of deploying specialized sensing infrastructures and enable many applications that require resources and sensing modalities beyond the current mote-class sensor processes as today's mobile devices (smartphones (iPhones, Sumsung Galaxy), tablets (iPad) and vehicle-embedded sensing devices (GPS)) integrate more computing, communication, and storage resources than traditional mote-class sensors. The current applications of MCSs include traffic congestion detection, wireless indoor localization, pollution monitoring, etc . There is no doubt that one of the most significant characteristics of MCSs is the active involvement of mobile users to collect and share sensory data.

In this dissertation, we study the incentive mechanism design in mobile crowdsensing system with consideration of economic properties.

Firstly, we investigate the problem of joining sensing task assignment and scheduling in MCSs with the following three considerations: i) partial fulfillment, ii) attribute diversity, and iii) price diversity. Then, we design a distributed auction framework to allow each task owner to independently process its local auction without collecting global information in a MCS, reducing communication cost. Next, we propose a cost-preferred auction scheme (CPAS) to assign each winning mobile user one or more sub- working time durations and a time schedule-preferred auction scheme (TPAS) to allocate each winning mobile user a continuous working time duration.

Secondly, we focus on the design of an incentive mechanism for a MCS to minimize the social cost. The social cost represents the total cost of mobile devices when all tasks published by the MCS are finished. We first present the working process of a MCS, and then build an auction market for the MCS where the MCS platform acts as an auctioneer and users with mobile devices act as bidders. Depending on the different requirements of the MCS platform, we design a Vickrey-Clarke-Groves (VCG)-based auction mechanism for the continuous working pattern and a suboptimal auction mechanism for the discontinuous working pattern. Both of them can ensure that the bidding of users are processed in a truthful way and the utilities of users are maximized. Through rigorous theoretical analysis and comprehensive simulations, we can proof that these incentive mechanisms satisfy economic properties and can be implemented in reasonable time complexcity.

Next, we discuss the importance of fairness and unconsciousness of MCS surveillance applications. Then, we propose offline and online incentive mechanisms with fair task scheduling based on the *proportional share allocation rules*. Furthermore, to have more sensing tasks done over time dimension, we relax the truthfulness and unconsciousness property requirements and design a (ε, μ)-unconsciousness online incentive mechanism. Real map data are used to validate these proposed incentive mechanisms through extensive simulations.

Finally, future research topics are proposed to complete the dissertation.

INDEX WORDS: Mobile Crowdsensing System(MCS), Incentive Mechnism, Economic property, Internet of Things(IoT)

INCENTIVE MECHANISM DESIGN IN MOBILE CROWDSENSING SYSTEMS

by

Zhuojun Duan

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2018

INCENTIVE MECHANISM DESIGN IN MOBILE CROWDSENSING SYSTEMS

by

Zhuojun Duan

Committee Chair:	Zhipeng Cai
Committee:	Wei Li Anu G. Bourgeois Gengsheng(Jeff) Qin

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
August 2018

DEDICATION

This dissertation is dedicated to my family for their endless support, love and passion during my Ph.D. years. I cannot finish my Ph.D. without their love and encouragement.

ACKNOWLEDGEMENTS

There are so many individuals who have shaped my career over the past years, I wish to reiterate my thanks to them.

I thank my research advisors Dr. Zhipeng Cai, Dr. Wei Li, and all dissertation committee members Dr. Anu Bourgeois and Dr. Gengsheng Qin, for their continuous support of my Ph.D study and research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this dissertation.

I am very thankful to the professors and staffs of Computer Science Department at Gerogia State University, especially Dr. Yingshu Li, Dr. Raj Sunderraman and Dr. Yuan Long, for their help with my Ph.D. study, teaching, research, and for their great support in my job hunting.

I appreciate the wonderful group, including Dr. Mingyuan Yan, Dr. Guoliang Liu, Dr. Meng Han, Dr. Xu Zheng, Dr. Dongjing Miao, Zaobo He, Ji Li, Yi Liang, and Yan Huang, for all their help and support throughout my Ph.D study.

I would like to thank my Master advisor Dr. Xiaoming Wang and his wife Wanping Yang, for their patient guidance, encouragement and advice they have provided throughout my time as their student. I have been extremely lucky to have them who cared so much about my work and life.

Finally, my deep and sincere gratitude to my family for their continuous love, help and support.

I would like to show my great gratitude to all of them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background and Motivations

The highly distributed wireless network paradigm extends ubiquity of the Internet through integrating every terminal for interaction via embedded systems, in which all the physical terminals can collect and exchange data. In network, the new emerging techniques integrate multiple types of sensors and high-performance processors into physical terminals, *e.g.*, smartphones (iPhones, Sumsung Galaxy, etc.), tablets (iPad, etc.), and vehicle-embedded sensing terminals (GPS). These mobile terminals can be used to sense and collect data, so that become data sources. All above mentioned properties promote the development of Mobile Crowdsening System (MCS). In an MCS, a complicated sensing job is divided into several simpler tasks. Each participated mobile physical terminal can undertake one or more simpler sensing tasks. The most attractive properties of MCSs is that it aims at letting the regular mobile physical terminals work for the complicated job, while keeping the users of these mobile physical terminals unconscious. In tradition, however, the job must be done by professional experts and the sensors have to be deployed in advance. The great potential of MCSs enables numerous applications, such as Common Sense [1] for air quality information collection, Nericell [2] for traffic information acquisition, and BikeNet [3] for the cyclist experience mapping.

High quality of sensing service in MCSs is heavily reliant on the number of participated users. Monetary incentive is the most popular and effective way to attract adequate users' participation and compensate their resource consumption (e.g, battery and computing) and risk of privacy leakage (e.g., location exposure) during the participation in MCSs. Recently, to effectively incentivize mobile users to participate in sensing services, a number of auction-based incentive mechanisms have been proposed [4–13]. According to the design objective, the existing works can be classified into two major categories: (i) social welfare/utility/profit/revenue maximization for the cloud platform [4–7]; and (ii) cost/payment minimization for the cloud platform [8–11]. Meanwhile, in different scenarios, some works also consider various assignment constraints, including data quality [4, 5], buyer budget [14, 15], and data sensing area coverage [9], etc.

Firstly, we investigate *the problem of joining sensing task assignment and scheduling in MCSs with the following three considerations*: i) *partial fulfillment*, which means that a sensing task can get assigned if it can be partially completed by one or more mobile users in the time domain; for example, if a task requests the sensory data at a certain location from 9:00am to 11:00am and a mobile user who is the only user in the mobile system can collect the required data from 9:30am to 10:30am, the task will be assigned to the mobile user; ii) *attribute diversity*, which indicates that the implementation requirements of tasks and the availability of mobile users vary in task attributes, including location, starting time, ending time, and types of sensors; and iii) *price diversity*, which says that each mobile user could ask different prices for performing different tasks. Notice that the existing auction schemes [4, 9, 16–21] do not consider task scheduling in the time domain, thus they cannot be applied to solve our problem. Moreover, extending such auctions to consider partial fulfillment, attribute diversity, and price diversity is nontrivial. Therefore, designing a truthful auction for task assignment and scheduling while taking into account partial fulfillment, attribute diversity, and price diversity is very challenging.

To overcome these challenges, we first formulate the joint problem of task assignment and scheduling as a *reverse auction*, in which partial fulfillment, attribute diversity, and price diversity are considered. Next, we design a distributed auction framework, in which each task owner independently controls its local auction. Based on such a framework, we propose two distributed auction schemes, cost-preferred auction scheme (CPAS) that schedules tasks according to the non-decreasing order of mobile users' asking prices and time schedule-preferred auction scheme (TPAS) that schedules tasks according to the non-decreasing order of mobile users' arrival time. Furthermore, via rigorous theoretical proof, we show that both CPAS and TAPS can achieve computational efficiency, individual rationality, budget balance, and truthfulness. Finally, our intensive simulation results confirm the effectiveness of the proposed auction schemes CPAS and TPAS.

Secondly, we focus on the design of an incentive mechanism for a MCS to minimize the social cost. The social cost represents the total cost of mobile devices when all tasks published by the MCS are finished. To achieve the objective in the MCS, we confront several challenges: i) *True cost revelation.* The cost of each mobile device for finishing a task is private. It is difficult to encourage all participants to report their real costs; ii) *Minimal cost optimization.* Assume all users report their true costs to the MCS. Since mobile devices may vary in capacities and costs, it is hard to select the optimal set of users; iii) *Incentive mechanism.* As discussed, the MCS platform should reward each user who works for it as incentives. Within the budget, the reward should be greater than the cost of the user. Deciding a proper reward for each participant is still challenging.

These challenges lead us to investigate an auction mechanism that concentrates on the trade between the MCS platform and mobile users. This work begins with the assumption that the MCS platform publishes only one task in one round and the task consists of pieces of sub-tasks. Each user with a mobile device can work for one or more sub-tasks. The auction mechanism in our paper aims to minimize the social cost of mobile users while guaranteeing the truthful cost of bidding from each participating user.

Depending on the requirements of a MCS platform, there are two different working patterns. The first one is the *continuous* working pattern, which requires each participant to work on a set of continuous sub-tasks. We call another working pattern the *discontinuous* working pattern, where a participant can work for any set of sub-tasks. Depending on the different requirements of the MCS platform, we design a Vickrey-Clarke-Groves (VCG)-based auction mechanism for the continuous working pattern and a suboptimal auction mechanism for the discontinuous working pattern. Both of them can ensure that the bidding of users are processed in a truthful way and the utilities of users are maximized.

Thirdly, motivating the mobile users to participate in sensing services for efficient data generation and collection is one of the most critical issues in mobile crowdsensing systems. Auction based mechanisms are seen to be promising and effective solutions to incentivize mobile users. Participant's preference for different sensing tasks is a pivotal factor which should be considered in the auction mechanisms as assigning the least favorite tasks discourages them to participate in future sensing tasks. Unfortunately, participant's preference have been overlooked by all the existing works, which motivates us to fill this gap in this paper. We first propose a new concept "mutual preference degree" to capture participant's preference and then design a preference-based auction mechanism (PreAM) to simultaneously guarantee individual rationality, budget feasibility, preference truthfulness, and price truthfulness. Finally, both the theoretical analysis and simulation results demonstrate the effectiveness of PreAM.

1.2 Organization

The rest of this dissertation is organized as follows: Chapter 2 summarizes the related literature. Chapter 3 investigates the distributed auctions for task assignment and scheduling in mobile crowdsensing systems. Chapter 4 studies the problem of minimizing social cost in mobile crowdsensing systems. Chapter 5 studies the problem of preference-based auction mechanism which can simultaneously guarantee individual rationality, budget feasibility, preference truthfulness, and price truthfulness. Chapter 6 conducts the future research directions. And the last chapter concludes this dissertation.

Chapter 2

RELATED WORKS

In this chapter, existing literature related to our research are summarized.

2.1 Applications of MCS

Recently, many commercial MCS applications have been released. These applications can be installed on mobile devices carried by users. After installation, these mobile users are able to undertake computational or sensing tasks. Then, all the results or information generated by the applications will be transmitted to a service center for final process. For example, GigWalk [22] can assist users in verifying the service quality and product placement. It can also provide reports about the graffiti at bus or train stations to the government. Additionally, GigWalk could work for real estate, consumer research, travel, advertising, and so on. Field Agent [23] is an application used for businesses. It can work for two tasks: audit and research. The audit task mainly focuses on information collection, which allows manufacturers and retailers to attract customers and spread information. The research task is interested in gathering customers' feedback on products or services, so the businessmen can have a better insight of the market. In [1], the authors introduce a MCS application named Common Sense, which is used for pollution monitoring. Nericell [2] can be used to determine the average speeds or traffic delays, and DietSense [24] is proposed for health control. These applications suggest that the importance of MCS is growing in practical business fields.

2.2 Distributed incentive mechanism design

Note that an auction can be performed in a centralized way [4, 9, 16–19], in which an auctioneer gathers global information and computes the auction results. In [16], a user-centric combinatorial auction was designed, in which each mobile user bids for a set of sensing tasks with an asking price and the crowdsourcer aims to maximize its utility via user selection. Feng *et al.* [9] proposed a reverse auction for the platform to minimize its cost, in which a sensing task can be done by a smartphone if the location of the sensing task is within the service coverage of the smartphone. Jin *et al.* [4] designed a single-minded reverse combinatorial auction and a multi-mined reverse combinatorial auction by considering the quality of information of mobile users. In [17], via considering that the sensing tasks are randomly published and the mobile users dynamically arrive in an MCS, an offline auction and an online auction were proposed. Zhang *et al.* [18] studied three auction schemes respectively corresponding to the following three scenarios in MCSs: i) single-requester single-bid model; ii) single-requester multiple-bid model; and iii) multiple-requester multiple-bid model. Ji *et al.* [19] investigated the discretization in crowdsensing systems and designed two auction-based incentive mechanisms, in which each user has a uniform sensing subtask length. However, in the auction models of the above work [4, 9, 16–19], for each sensing task, the requirements of working time and types of sensors are not taken into account.

To the best of our knowledge, the only existing work on distributed incentive mechanisms for task allocation in MCSs is [20]. In [20], the authors first formulated the problem of task selection for mobile users as a non-cooperative task selection game and then investigated the equilibriums and convergence of the game. In the proposed game, the objective is to maximize each mobile user's utility by finding an order to complete one or more sensing tasks that locate at different places.

Different from all the above prior work, we propose to design distributed truthful auction schemes for task assignment and scheduling in MCSs while considering partial fulfillment, attribute diversity, and price diversity.

2.3 Truthful Incentive Mechanisms for Social Cost Minimization

Many studies have been done on task allocation in MCSs. Most of them target the maximization of system efficiency. In [25], the authors design a fair energy-efficient allocation framework and propose two sensing task allocation algorithms: one is an offline allocation algorithm and the other is an online allocation algorithm. Ho and Vaughan [26] formalize the online task assignment problem, which makes the allocation decision upon arrival of each worker. Then, a two-phase exploration–exploitation assignment algorithm is proposed. Authors of [27] investigate the problem of task assignment and label inference for heterogeneous classification tasks. They derive a probably near-optimal adaptive assignment algorithm by applying online primal-dual techniques. An architectural model using the SLURM tool for efficient management in the MCS is outlined in [28]. The authors propose a novel idea of adaptive task scheduling which is based on the feedback of customer satisfaction. However, they don't consider the incentive mechanisms.

A handful of researchers put effort on the design of incentive mechanisms for the MCS. Yang *et al.* [29] consider two types of incentive mechanisms: platform-centric incentive mechanisms and user-centric incentive mechanisms. The first one is based on the Stackelberg game, in which the MCS platform has absolute control over the total budget to users, and users can only adjust their actions to meet the requirements of the platform. The roles of the platform and users are reversed in the user-centric incentive mechanisms. Each user reports the lowest price for selling a service to the MCS platform. In [30], the authors design a reward-based collaboration mechanism. The client publishes a total reward to be shared among collaborators. The collaboration is successful when enough users are willing to collaborate. In order to attract more users to participate, [31] designs a novel Reverse Auction-based Dynamic Price (RADP) incentive mechanism. In this mechanism, users can sell their sensing data to a service provider by their claimed bid prices. Singla and Krause [32] exploit a link between procurement auctions and multi-armed bandits. Its mechanism design is budget feasible. In conclusion, most of the existing works concentrate on maximizing social efficiency and achieving fairness in MCSs.

Chapter 3

DISTRIBUTED AUCTIONS FOR TASK ASSIGNMENT AND SCHEDULING IN MOBILE CROWDSENSING SYSTEMS

3.1 Introduction

In the past few years, the popularity of *Mobile Crowdsensing Systems* (MCSs) has been greatly prompted, in which sensory data can be ubiquitously collected and shared by mobile devices in a distributed fashion. Typically, a MCS consists of a cloud platform, sensing tasks, and mobile users equipped with mobile devices, in which the mobile users carry out sensing tasks and receive monetary rewards as compensation for resource consumption (*e.g.*, energy, bandwidth, and computation) and risk of privacy leakage (*e.g.*, location exposure). Compared with traditional mote-class sensor networks, MCSs can reduce the cost of deploying specialized sensing infrastructures and enable many applications that require resources and sensing modalities beyond the current mote-class sensor processes as today's mobile devices (smartphones (iPhones, Sumsung Galaxy), tablets (iPad) and vehicle-embedded sensing devices (GPS)) integrate more computing, communication, and storage resources than traditional mote-class sensors [33]. The current applications of MCSs include traffic congestion detection, wireless indoor localization, pollution monitoring, etc [20, 33, 34]. There is no doubt that one of the most significant characteristics of MCSs is the active involvement of mobile users to collect and share sensory data. In other words, for any MCS, one of the most important problems is " *how to get mobile users involved in sensing tasks?*"

Thus, to effectively incentivize mobile users to join mobile crowdsening, auction that is a powerful game-theoretical incentive mechanism [35–40] has been widely applied to design market-based sensing task assignment schemes [4, 9, 16–21]. Unfortunately, several critical issues are ignored by most of the existing work: i) in the proposed auction models [9, 17–19, 21, 41, 42], a sensing task is assigned to a mobile user if and only if the task can be fully completed by the mobile user, which is impractical in some scenarios; in reality, a sensing task might not be fully completed by one mobile user at a time (*e.g.*, pollution monitoring within an area during a time period) as a mobile user's available working time in an MCS is limited; ii) heterogeneity in MCSs is not fully explored – sensing tasks may have different requirements in terms of location, starting time, ending time, types of sensors, *etc*, and mobile users also vary in their locations, available starting time, available ending time, set of equipped sensors, *etc*; iii) due to the aforementioned diversities of task requirement and user availability, the prices asked by a mobile user to process different tasks are also different [43–52].

Motivated by the above observations, in this work, we investigate *the problem of joining sensing task assignment and scheduling in MCSs with the following three considerations*: i) *partial fulfillment*, which means that a sensing task can get assigned if it can be partially completed by one or more mobile users in the time domain; for example, if a task requests the sensory data at a certain location from 9:00am to 11:00am and a mobile user who is the only user in the mobile system can collect the required data from 9:30am to 10:30am, the task will be assigned to the mobile user; ii) *attribute diversity*, which indicates that the implementation requirements of tasks and the availability of mobile users vary in task attributes, including location, starting time, ending time, and types of sensors; and iii) *price diversity*, which says that each mobile user could ask different prices for performing different tasks. Notice that the existing auction schemes [4,9,16–21] do not consider task scheduling in the time domain, thus they cannot be applied to solve our problem. Moreover, extending such auctions to consider partial fulfillment, attribute diversity, and price diversity is nontrivial. Therefore, designing a truthful auction for task assignment and scheduling while taking into account partial fulfillment, attribute diversity, and price diversity is very challenging.

In this work, to overcome the above challenges, we first formulate the joint problem of task assignment and scheduling as a *reverse auction*, in which partial fulfillment, attribute diversity, and price diversity are considered. Next, we design a distributed auction framework, in which each task owner independently controls its local auction. Based on such a framework, we propose two distributed auction schemes, cost-preferred auction scheme (CPAS) that schedules tasks according to the non-decreasing order of mobile users' asking prices and time schedule-preferred auction scheme (TPAS) that schedules tasks according to the non-decreasing order of mobile users' arrival time. Furthermore, via rigorous theoretical proof, we show that both CPAS and TAPS can achieve computational efficiency, individual rationality, budget balance, and truthfulness. Finally, our intensive simulation results confirm the effectiveness of the proposed auction schemes CPAS and TPAS. To sum up, our multi-fold contributions are as follows [53]:

- To the best of our knowledge, we are the first to establish a reverse auction model incorporating sensing task assignment and scheduling while considering partial fulfillment, attribute diversity, and price diversity.

- We design a distributed auction framework to allow each task owner to independently process its local auction without collecting global information in an MCS, reducing communication cost.

- We propose a cost-preferred auction scheme (CPAS) to assign each winning mobile user one or more sub-working time durations and a time schedule-preferred auction scheme (TPAS) to allocate each winning mobile user a continuous working time duration.

- We perform comprehensive theoretical analysis and prove that both CPAS and TAPS are computationally efficient, individually rational, budget balanced and truthful.

- The simulations are well conducted to validate the performance of CPAS and TPAS in terms of allocation efficiency, working time utilization, utility, and truthfulness.

The rest of this chapter is organized as follows. Then the system model and problem formulation are presented in Section 3.2. The cost-preferred auction scheme and the time schedule-preferred auction scheme are proposed in Section 3.3 and Section 3.4, respectively. After evaluating the performance of the two proposed auction schemes in Section 3.5, we conclude this paper in Section 3.6.

3.2 System Model and Problem Formulation

3.2.1 System Model

We consider an MCS consisting of a cloud platform, multiple sensing task owners (STOs), and multiple mobile users equipped with smart devices (MUDs). In an MCS, each STO acts as a buyer demanding sensing task service and each MUD acts as a seller offering sensing task service. Both the STOs and the MUDs can connect to the platform via cloud. The cloud platform allows the STOs to periodically publish their sensing requests.

Suppose that there are m STOs and each one owns a sensing task to be done. Let $\Pi = \{\pi_1, \pi_2, \cdots, \pi_m\}$ be the set of all STOs' tasks. In this work, "STO i's task" and "task π_i" are interchangeable as each STO has only one task request. Each sensing task is associated with four attributes: *locations*, *starting time*, *ending time*, and *resources (e.g., camera and gyroscope)*. These four attributes indicate the specific requirements of sensing task implementation and are determined by the STOs. Each STO i's *sensing task information*, denoted by f_i^π, includes the four task attributes and can be formally represented as follows:

$$f_i^\pi = (L_i^\pi, [\alpha_i^\pi, \beta_i^\pi], R_i^\pi),$$

where L_i^π is the required location to perform task π_i, α_i^π and β_i^π are respectively the starting time and the ending time to perform task π_i, and R_i^π presents a set of required resource to implement task π_i. In addition, STO i has budget b_i to complete task π_i per unit time slot. In other words, STO i requests that task π_i needs to be implemented at location L_i^π during time period $[\alpha_i^\pi, \beta_i^\pi]$ by consuming a set of resource R_i^π with a maximum unit payment b_i.

On the other hand, there exist n MUDs denoted by $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$. Each MUD γ_j is allowed to work for at most one STO and has its initial location L_j^γ, available starting time α_j^γ, available ending time β_j^γ, a set of resource R_j^γ embedded into its smart device, and an average moving rate λ_j^γ. Formally, each MUD γ_j's *sensing service information* is defined to be

$$f_j^\gamma = (L_j^\gamma, [\alpha_j^\gamma, \beta_j^\gamma], R_j^\gamma, \lambda_j^\gamma).$$

Obviously, to implement each task π_i, a cost of moving from location L_j^γ to location L_i^π and a cost of consuming resource in R_j^γ will be brought to each MUD γ_j. Thus, each MUD γ_j has an asking price vector $A_j =< a_{1j}, a_{2j}, \cdots, a_{mj} >$, in which every a_{ij} $(1 \leq i \leq m)$ is an asking price per unit time slot indicating the costs of movement and resource consumption to process task π_i .

Since task π_i's location L_i^π may be different from MUD γ_j's location L_j^γ, MUD γ_j should first move from L_j^γ to L_i^π and then start the required sensing task. Let $d(L_i^\pi, L_j^\gamma)$ be the Euclidean distance between locations L_i^π and L_j^γ. With moving rate λ_j^γ, MUD γ_j arrives at location L_i^π at time $t_{ij}^\alpha = \frac{d(L_i^\pi, L_j^\gamma)}{\lambda_j^\gamma} + \alpha_j^\gamma$. For simplicity, we assume that an MUD can start working as soon as it arrives at a task's required location. Indeed, this assumption does not affect the performance of our proposed model and schemes. Let T_{ij} be MUD γ_j's maximum available working time duration for π_i and $|T_{ij}|$ be the number of time slots of time duration T_{ij}. For each MUD γ_j and each task π_i, T_{ij} can be calculated via the following six cases:

1. If $t_{ij}^\alpha \geq \beta_i^\pi$, task π_i is finished when MUD γ_j arrives and thus MUD γ_j cannot perform task π_i, *i.e.*, $T_{ij} = \emptyset$.

2. If $\beta_j^\gamma \leq \alpha_i^\pi$, task π_i starts when MUD γ_j's available time ends. Therefore, MUD γ_j cannot perform task π_i and we have $T_{ij} = \emptyset$.

3. If $\alpha_i^\pi \leq t_{ij}^\alpha < \beta_j^\gamma < \beta_i^\pi$, it means that MUD γ_j arrives when/after task π_i begins and that MUD γ_j's available time ends before task π_i's ending time. Accordingly, we have $T_{ij} = [t_{ij}^\alpha, \beta_j^\gamma]$.

4. If $\alpha_i^\pi \leq t_{ij}^\alpha < \beta_i^\pi \leq \beta_j^\gamma$, MUD γ_j arrives when/after task π_i begins and MUD γ_j's available time ends when/after task π_i's required working time terminates. Thus, the maximum available working time duration is $T_{ij} = [t_{ij}^\alpha, \beta_i^\pi]$.

5. If $t_{ij}^\alpha < \alpha_i^\pi < \beta_j^\gamma < \beta_i^\pi$, MUD γ_j arrives before task π_i begins and MUD γ_j's available time ends before task π_i's ending time. In this case, $T_{ij} = [\alpha_i^\pi, \beta_j^\gamma]$.

6. If $t^{\alpha}_{ij} < \alpha^{\pi}_{i} < \beta^{\pi}_{i} \leq \beta^{\gamma}_{j}$, MUD γ_j arrives before task π_i begins and MUD γ_j's available time ends when/after task π_i's required working time terminates. As a result, we have $T_{ij} = [\alpha^{\pi}_{i}, \beta^{\gamma}_{j}]$.

3.2.2 Problem Formulation

When competing for task π_i with other MUDs, MUD γ_j is scheduled an actual working time duration T^s_{ij} by STO i. Correspondingly, denoted by $|T^s_{ij}|$ the number of time slots of T^s_{ij}. Note that T_{ij} is the maximum available working time duration of MUD γ_j for task π_i, thus we obtain the following relationships: i) $T^s_{ij} \subseteq T_{ij}$; and ii) $0 \leq |T^s_{ij}| \leq |T_{ij}|$. We use a 0-1 binary variable $x_{ij} \in \{0, 1\}$ to indicate the task assignment, *i.e.*, γ_j processes task π_i if and only if $x_{ij} = 1$. Since each MUD γ_j is allowed to work for at most one STO, we have $\sum_{i=1}^{m} x_{ij} \leq 1$. If γ_j is allocated to perform task π_i, γ_j can obtain a payment p_{ij} from task owner STO i as a reward and receive a utility U^{γ}_{j} that is computed through Eq. (3.1).

$$U^{\gamma}_{j} = \sum_{i=1}^{m} u^{\gamma}_{ij} = \sum_{i=1}^{m} x_{ij}(p_{ij} - a_{ij}|T^s_{ij}|). \tag{3.1}$$

In this work, we consider a practical scenario, in which each STO i independently controls its local task auction to determine the winning MUDs and to schedule their working time. Thus, each STO i also works as an auctioneer of its local task auction which can be formulated to be a *reverse auction* as presented in Eq. (3.2).

$$\begin{aligned}
\min \quad & \sum_{j=1}^{n} x_{ij} a_{ij} |T^s_{ij}|, && \text{(3.2a)}\\
\text{s.t.} \quad & \bigcup_{j=1}^{n} x_{ij} T^s_{ij} \subseteq [\alpha^{\pi}_{i}, \beta^{\pi}_{i}], && \text{(3.2b)}\\
& \sum_{j=1}^{n} x_{ij} |T^s_{ij}| \leq |\beta^{\pi}_{i} - \alpha^{\pi}_{i}|, && \text{(3.2c)}\\
& x_{ij} \in \{0, 1\}, 1 \leq j \leq n, && \text{(3.2d)}\\
& T^s_{ij} \subseteq T_{ij}, 1 \leq j \leq n. && \text{(3.2e)}
\end{aligned}$$

In the above reverse auction Eq. (3.2), each STO i's objective is to minimize the cost for sensing task assignment and scheduling such that the following conditions can simultaneously hold: i) condition Eq. (3.2b) requires that the union of scheduled working time durations cannot exceed the task's time duration; ii) condition Eq. (3.2c) indicates that the total allocated time slots cannot be more than the number of slots of the task's time duration; iii) conditions Eqs. (3.2d) and (3.2e) show the ranges of assignment variable x_{ij} and schedule variable T_{ij}^s, respectively.

3.2.3 Auction Economic Properties

In an auction scheme, the following economic properties are typically considered [35]:

- **Individual-rationality.** This states that no buyer/seller obtains a negative utility, *i.e.*, in this work, $U_j^\gamma \geq 0$ for all $\gamma_j \in \Gamma$.

- **Budget-balance.** In a double-side auction, the auctioneer's budget is the difference between the total charge collected from all buyers and the total payment paid to all sellers. Notice that each STO i has a budget b_i in its single-side reverse auction and works as both a buyer and an auctioneer at the same time. Therefore, in each STO i's auction, budget-balance is defined as: $\sum_{j=1}^{n} x_{ij} b_i |T_{ij}^s| - \sum_{j=1}^{n} x_{ij} p_{ij} \geq 0$ for all $1 \leq i \leq m$.

- **Incentive-compatibility.** This is also called "truthfulness" or "strategy-proof", which indicates that no bidder can improve its received utility via lying about its bid price. In each STO i's auction, incentive-compatibility ensures that each MUD $\gamma_j \in \Gamma$ can receive a maximum utility if and only if its asking price satisfies $a_{ij} = \bar{a}_{ij}$ for all $\pi_i \in \Pi$, where $\bar{a}_{ij}$ denotes the true asking price of MUD γ_j for task π_i.

If an auction can simultaneously achieve individual-rationality, budget-balance, and truthfulness, it is called *economic-robust auction.*

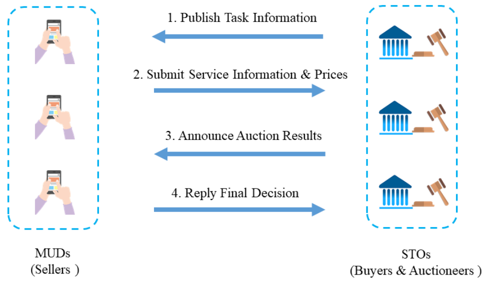


Figure 3.1. A distributed auction framework.

3.2.4 Distributed Auction Framework

In this chapter, we propose to design distributed auction schemes containing four major stages that are presented in Fig. 3.1. These four stages are briefly summarized in the following:

- **Stage 1: Publish Task Information.** At the beginning, each STO i publishes its task information f_i^{π} and the deadline of accepting bids from MUDs on the cloud platform. The bid submitted by an MUD after the deadline will be rejected.

- **Stage 2: Submit Service Information & Price.** After receiving the task information, each MUD γ_j submits its service information f_j^{γ} and asking price a_{ij} to STO i if it is interested in task π_i.

- **Stage 3: Announce Auction Results.** Each STO i collects service information and asking prices from the MUDs, schedules working time, decides the potential winners, and payments. Then, each STO i announces the auction results and a deadline of submitting final decision to the MUDs who have submitted information and prices. Each MUD should reply its final decision to the STOs who have chosen it as a potential winner before the deadline.

- **Stage 4: Reply Final Decision.** If MUD γ_j is chosen as a potential winner by one or more STOs, it should reply its final decision to the STOs who have chosen it as a potential winner.

From the above four stages, one can see that an STO may be rejected by the MUDs. Thus, to complete the sensing task, each STO continues to conduct its reverse auction to schedule the remaining unassigned working time slots in a multi-round manor until its task time duration has been fully scheduled or no potential winning MUD can be selected. Meanwhile, if an MUD is successfully scheduled to a task, it exits the auction; otherwise, it continues to compete for working until no task information is published.

Furthermore, under the proposed auction framework, two different policies can be used to perform task assignment and scheduling: i) **cost-preferred policy:** the STOs determine the potential winners according to the non-decreasing order of the MUDs' asking prices; and ii) **time schedule-preferred policy:** the STOs determine the potential winners based on a first-come-first-serve manor. The adoption of the policy is determined through MUDs' negotiation before conducting the auction. The auction schemes corresponding to the two policies are detailed in Sections 3.3 and 3.4, respectively.

3.3 Cost-Preferred Auction Scheme

In this section, a Cost-Preferred Auction Scheme termed **CPAS** is proposed, in which each STO i greedily performs sensing task assignment and scheduling according to the non-decreasing order of the MUDs' asking prices. The stages of CPAS for each STO i is outlined in Algorithm 1.

Since the auction scheme CPAS is performed in a multi-round manor and the auction process of each round is the same, we just demonstrate the auction process of a round in the following part of this section.

3.3.1 Information Publication & Collection

At the beginning of an auction, each STO i publishes its task information f_i^{π} on the cloud platform. After obtaining all the STOs' task information, each MUD γ_j submits its service information f_j^{γ} and asking price a_{ij} to STO i. Note that an MUD could be interested

Algorithm 1: Cost-Preferred Auction Scheme for STO i

input : f_i^π, $[\alpha_i^\pi, \beta_j^\pi]$
output: $\{x_{ij}\}$, $\{T_{ij}^s\}$

Set $\{x_{ij}\} = \{0\}$, $\{T_{ij}^s\} = \{\emptyset\}$, and $T_i^u = [\alpha_i^\pi, \beta_j^\pi]$;
REPEAT
Publish sensing task information f_i^π**;**
Receive sensing service information $\{f_j^\gamma\}$ **and asking prices** $\{a_{ij}\}$ **from the MUDs;**
Run Alg. 2 to determine potential winners, schedule working time, compute payments, and announce the results;
Collect replies from the MUDs, record the values of $\{x_{ij}\}$**, and update** $T_i^u = T_i^u \setminus \bigcup_{j=1}^{n} x_{ij} T_{ij}^s$**;**
UNTIL$T_i^u = \emptyset$ **or no potential winner is selected.**

in more than one sensing task and send its service information and asking price to the corresponding STOs at the same time.

3.3.2 Potential Winner Determination and Payment Calculation

When STO i receives service information $\{f_j^\gamma\}$ and asking price $\{a_{ij}\}$ from one or more MUDs, based on $\{f_j^\gamma\}$, $\{a_{ij}\}$, f_i^π, and b_i, STO i forms a set of available MUDs as

$$\Gamma^c(\pi_i) = \{\gamma_j | (T_{ij} \cap T_i^u) \neq \emptyset, R_j^\gamma \subseteq R_i^\pi, \text{ and } a_{ij} \leq b_i\},$$

in which T_i^u denotes the un-scheduled time duration for task π_i at the current round of auction. This computation is implemented in lines 2-7 of Algorithm 2.

Potential Winner Determination Initially, the set of potential winners is $W(\pi_i) = \emptyset$. To schedule working time, STO i first sorts all the available MUDs in $\Gamma^c(\pi_i)$ in a non-decreasing order in terms of their asking prices and gets a sorted set $\Gamma^{c\prime}(\pi_i)$ (see line 8 in Algorithm 2). Next, STO i scans the available MUDs in $\Gamma^{c\prime}(\pi_i)$ and allocates un-scheduled time slots in a greedy manor. Specifically speaking, if MUD γ_j's current available working time duration $(T_{ij} \cap T_i^u)$ has not been fully scheduled to other available MUDs, *i.e.*, $(T_{ij} \cap$

$T_i^u) \cap (\bigcup_{\gamma_{j'} \in W(\pi_i)} T_{ij'}^s) \neq (T_{ij} \cap T_i^u)$, MUD γ_i can be chosen as a potential winner and assigned a set of time slots that has not been allocated to current potential winners in $W(\pi_i)$, *i.e.*, $T_{ij}^s = (T_{ij} \cap T_i^u) \backslash (T_{ij} \cap T_i^u \cap (\bigcup_{\gamma_{j'} \in W(\pi_i)} T_{ij'}^s))$ (see lines 9-14 in Algorithm 2).

Payment Calculation After completing task scheduling, STO i computes the payment for each potential winning MUD γ_j via identifying γ_j's *critical neighbor*, which is defined to be the MUD γ_k in $\Gamma^c(\pi_i)$ where if a_{ij} is higher than a_{ik}, γ_j can not be scheduled. Different from the previous works [9, 17–19, 21, 41, 42] in which each winner has only one critical neighbor, each winning MUD γ_j in the auction CPAS has one or more critical neighbors because the time slots of T_{ij}^s could be scheduled to one or more other available MUDs if MUD γ_j does not join the auction (see lines 17 - 32 of Algorithm 2). Thus, the payment is calculated according to every critical neighbor of winner γ_j. In order to find the critical neighbors, STO i sorts all the MUDs in $\Gamma_{-\gamma_j}^c(\pi_i) = \Gamma^c(\pi_i) \setminus \gamma_j$ in the non-decreasing order in terms of their asking prices, selects winners again in the sorted set $\Gamma_{-\gamma_j}^{c\prime}(\pi_i)$, and schedules working time to them. Any MUD γ_k is a critical neighbor of MUD γ_j if their allocated time durations are overlapping, *i.e.*, $T_{ik}^{s\prime} \cap T_{ij}^{s\prime} \neq \emptyset$, where $T_{ik}^{s\prime}$ is the time duration assigned to MUD γ_k and $T_{ij}^{s\prime}$ records the remaining time duration in T_{ij}^s that is not allocated to others. So the corresponding critical payment is $a_{ik}|T_{ik}^{s\prime} \cap T_{ij}^{s\prime}|$. But, if no critical neighbor is found for MUD γ_j, its critical payment is STO i's budget $b_i|T_{ij}^{s\prime}|$.

Then, STO i announces the auction results $\{T_{ij}^s\}$ and $\{p_{ij}\}$ to the MUDs.

Remark: Via Algorithm 2, each potential winning MUD can receive a working time duration T_{ij}^s that contains one or more sub-time durations. For example, the time duration of task π_i is from 1:00pm to 5:00pm, winner γ_j's working time duration is $T_{ij}^s = \{[2\text{:}00\text{pm}, 3\text{:}00\text{pm}], [4\text{:}30\text{pm}, 5\text{:}00\text{pm}]\}$ containing two sub-time durations, and the number of working time slots is $|T_{ij}^s| = 90$ minutes.

Algorithm 2: Cost-Preferred Task Scheduling & Pricing for Task π_i

input : $f_i^\pi, b_i, T_i^u, \Gamma, \{f_j^\gamma\}, \{a_{ij}\}$

output: $W(\pi_i), \{T_{ij}^s\}, \{p_{ij}\}$

Set $\Gamma^c(\pi_i) = \emptyset, W(\pi_i) = \emptyset$, $\{T_{ij}^s\} = \{\emptyset\}$, and $\{p_{ij}\} = \{0\}$;

for *each* $\gamma_j \in \Gamma$ *with submitted* f_j^γ *and* a_{ij} **do**

  Calculate T_{ij};

  if $(T_{ij} \cap T_i^u) \neq \emptyset$, $R_j^\gamma \subseteq R_i^\pi$, *and* $a_{ij} \leq b_i$ **then**

    $\Gamma^c(\pi_i) = \Gamma^c(\pi_i) \cup \gamma_j$;

Sort all MUDs in $\Gamma^c(\pi_i)$ in non-decreasing order based on $\{a_{ij}\}$ and obtain the sorted set $\Gamma^{c\prime}(\pi_i)$;

for $j = 1$ *to* $|\Gamma^{c\prime}(\pi_i)|$ **do**

  if $(T_{ij} \cap T_i^u) \cap (\bigcup_{\gamma_{j'} \in W(\pi_i)} T_{ij'}^s) \neq (T_{ij} \cap T_i^u)$ **then**

    $W(\pi_i) = W(\pi_i) \cup \gamma_j$;

    $T_{ij}^s = (T_{ij} \cap T_i^u) \backslash (T_{ij} \cap T_i^u \cap (\bigcup_{\gamma_{j'} \in W(\pi_i)} T_{ij'}^s))$;

for *each* $\gamma_j \in W_i^\pi$ **do**

  Set $\{T_{ik}^{s\prime}\} = \{\emptyset\}$ and $T_{ij}^{s\prime} = T_{ij}^s$;

  Sort all the MUDs in $\Gamma^c(\pi_i) \backslash \gamma_j$ in a non-decreasing order based on $\{a_{ik}\}$ and obtain the sorted set $\Gamma_{-\gamma_j}^{c\prime}(\pi_i)$;

  Set $k = 1$ and $W_{-\gamma_j}(\pi_i) = \emptyset$;

  while $k \leq |\Gamma_{-\gamma_j}^{c\prime}(\pi_i)|$ *and* $T_{ij}^{s\prime} \neq \emptyset$ **do**

    if $(T_{ik} \cap T_i^u) \cap (\bigcup_{\gamma_{j'} \in W_{-\gamma_j}(\pi_i)} T_{ij'}^{s\prime}) \neq (T_{ik} \cap T_i^u)$ **then**

      $W_{-\gamma_j}(\pi_i) = W_{-\gamma_j}(\pi_i) \cup \gamma_k$,

      $T_{ik}^{s\prime} = (T_{ik} \cap T_i^u) \backslash (T_{ik} \cap T_i^u \cap (\bigcup_{\gamma_{j'} \in W_{-\gamma_j}(\pi_i)} T_{ij'}^{s\prime}))$;

      if $T_{ik}^{s\prime} \cap T_{ij}^{s\prime} \neq \emptyset$ **then**

        $p_{ij} = p_{ij} + a_{ik} |T_{ik}^{s\prime} \cap T_{ij}^{s\prime}|$,

        $T_{ij}^{s\prime} = T_{ij}^{s\prime} \backslash (T_{ik}^{s\prime} \cap T_{ij}^{s\prime})$;

    $k = k + 1$;

  if $T_{ij}^{s\prime} \neq \emptyset$ **then**

    $p_{ij} = p_{ij} + b_i |T_{ij}^{s\prime}|$.

3.3.3 Final Service Decision

Each MUD γ_j independently makes its service decision when it obtains the auction results from the STOs. Let $\Pi(\gamma_j)$ be the set of tasks of which their owners select MUD γ_j as a potential winner, which is defined as

$$\Pi(\gamma_j) = \{\pi_i | \gamma_j \in W(\pi_i) \text{ and } \pi_i \in \Pi\}.$$

The decision process is described as follows.

- If $|\Pi(\gamma_j)| = 0$, MUD γ_j is a loser in each STO i's local auction scheme CPAS, does not need to send a reply, and remains in the auction until no task request is published.

- If $|\Pi(\gamma_j)| = 1$, MUD γ_j is a potential winner in an STO i's location auction, accepts the service request, and exits the auction.

- If $|\Pi(\gamma_j)| > 1$, MUD γ_j is selected as a potential winner by multiple STOs and accepts the STO who brings γ_j the maximum utility. That is, the accepted task request π_i is decided as

$$\pi_i = \arg \max_{\pi_h \in \Pi(\gamma_j)} \{(p_{hj} - a_{hj}|T^s_{hj}|)\}.$$

  Then, MUD γ_j exits the auction.

Finally, each STO i sets the values of $\{x_{ij}\}$ based on $W(\pi_i)$ and the MUDs' replies.

3.3.4 Property Analysis

In this subsection, we theoretically analyze the performance of auction mechanism CPAS in terms of computational efficiency, individual-rationality, budget-balance, and truthfulness.

Lemma 1. *The cost-preferred scheduling scheme, Algorithm 2, can terminate within $O(n^2 \log(n))$.*

Proof: From line 2 to line 7, the running time of forming set $\Gamma^c(\pi_i)$ is at most n that is the number of MUDs in set Γ. In line 8, sorting the MUDs in $\Gamma^c(\pi_i)$ costs at most $n \log(n)$

time. The potential winner determination, in lines 9-14, has a time complexity of $O(n)$. Similarly, we know that the sorting process of line 17 has a time complexity of $O(n\log(n))$ and that finding critical neighbors terminates within $O(n)$. Additionally, the "for" loop from line 15 to line 33 contains at most n iterations and can end within $O(n^2\log(n))$. To sum up, the time complexity of Algorithm 2 is $O(n^2\log(n))$.

Theorem 1. *The proposed auction scheme CPAS is computationally efficient with a time complexity of* $O(n^3\log(n))$.

Proof: From Algorithm 1, one can see that each STO i stops if and only if either of the two conditions satisfies: i) $T_i^u = \emptyset$; and ii) no potential winner is selected. Let us consider the worst case for any STO i: STO i picks only one potential winner at each round but is rejected by the potential winner. Under this situation, the potential winner definitely accepts another STO's task request and then exits the auction. Thus, after at most n rounds, STO i ends its auction as no potential winner can be chosen. From Lemma 1, we obtain the conclusion that the time complexity of CPAS is $O(n^3\log(n))$.

Theorem 2. *The auction scheme CPAS ensures individual-rationality for all MUDs.*

Proof: If MUD γ_j is a loser for all the tasks in an auction, we have $\sum_{i=1}^{m} x_{ij} = 0$ and $U_j^\gamma = 0$.

If MUD γ_j is a winner for task π_i in an auction, $x_{ij} = 1$ and $T_{ij}^s > 0$. Due to the definitions of critical neighbor and set $\Gamma^c(\pi_i)$, we have $a_{ik} \geq a_{ij}$ for γ_j's every critical neighbor γ_k and $b_i \geq a_{ij}$ for STO i, indicating that $p_{ij} \geq a_{ij}|T_{ij}^s|$ (see line 24 and line 31 of Algorithm 1). Therefore, $U_j^\gamma = \sum_{i=1}^{m} u_{ij}^\gamma = \sum_{i=1}^{m} x_{ij}(p_{ij} - a_{ij}|T_{ij}^s|) \geq 0$.

Theorem 3. *The auction scheme CPAS is budget-balanced for all STOs*

Proof: From Algorithm 1, it can be seen that all the potential winners are selected from set $\Gamma^c(\pi_i)$ and $a_{ij} \leq b_i$ for all $\gamma_j \in \Gamma^c(\pi_i)$. In addition, from line 24 and line 31 of Algorithm 1, we have $b_i \geq p_{ij}$ for each winner γ_j. Thus, $\sum_{j=1}^{n} x_{ij}b_i|T_{ij}^s| - \sum_{j=1}^{n} x_{ij}p_{ij} \geq 0$, *i.e.*, CPAS achieves budget-balance for each STO i.

Lemma 2. *In each STO i's auction CPAS, if MUD γ_j is selected as a potential winner with a price a_{ij}, it can still be a potential winner with a smaller price $a_{ij}^1 < a_{ij}$ and $T_{ij}^s \subseteq T_{ij}^{s1}$, where T_{ij}^{s1} is the assigned working time duration corresponding to a_{ij}^1.*

Proof: Suppose that $pos(a_{ij}^1)$ and $pos(a_{ij})$ are the positions of MUD γ_j in the sorted set $\Gamma^{c\prime}(\pi_i)$ when bidding with a_{ij}^1 and a_{ij}, respectively. Since $a_{ij}^1 < a_{ij}$, $pos(a_{ij}^1) \leq pos(a_{ij})$. From the methods of scheduling and pricing (see Algorithm 2), it is seen that MUD γ_j submitting a_{ij}^{s1} can be successfully scheduled a time duration T_{ij}^{s1} and $T_{ij}^s \subseteq T_{ij}^{s1}$.

Theorem 4. *The auction scheme CPAS guarantees truthfulness for all MUDs.*

Proof: To prove this theorem, it is equivalent to prove that in each STO i's local auction CPAS, each MUD $\gamma_j \in \Gamma$ cannot enhance its utility by submitting an asking price $a_{ij} \neq \bar{a}_{ij}$. This can be analyzed through the following cases.

Case 1: $a_{ij} < \bar{a}_{ij}$ (or $a_{ij} > \bar{a}_{ij}$) and MUD γ_j loses the auction with both a_{ij} and $\bar{a}_{ij}$. In this case, γ_j's utility received from STO i's auction is zero.

Case 2: $a_{ij} < \bar{a}_{ij}$ and MUD γ_j can win the auction with both a_{ij} and $\bar{a}_{ij}$. According to Lemme 2, we have $\bar{T}_{ij}^s \subseteq T_{ij}^s$ and $|\bar{T}_{ij}^s| \leq |T_{ij}^s|$, where $\bar{T}_{ij}^s$ and $|\bar{T}_{ij}^s|$ respectively denote the assigned time duration and the number of time slots of $\bar{T}_{ij}^s$ corresponding to $\bar{a}_{ij}$. Accordingly, the payment p_{ij} can be re-computed via two parts: i) the payments $\bar{p}_{ij}$ paid for time duration $\bar{T}_{ij}^s$ that is the same for both $\bar{a}_{ij}$ and a_{ij}; and ii) payment Δp_{ij} paid for time duration $T_{ij}^s \setminus \bar{T}_{ij}^s$, in which $a_{ij}|T_{ij}^s \setminus \bar{T}_{ij}^s| \leq \Delta p_{ij} \leq \bar{a}_{ij}|T_{ij}^s \setminus \bar{T}_{ij}^s|$ as $a_{ij} \leq a_{ik} \leq \bar{a}_{ij}$ for γ_j's every critical neighbor γ_k. Correspondingly, the received utility is $u_{ij}^\gamma = p_{ij} - \bar{a}_{ij}|T_{ij}^s| = (\bar{p}_{ij} - \bar{a}_{ij}|\bar{T}_{ij}^s|) + (\Delta p_{ij} - \bar{a}_{ij}|T_{ij}^s \setminus \bar{T}_{ij}^s|)$. Since $a_{ij}|T_{ij}^s \setminus \bar{T}_{ij}^s| \leq \Delta p_{ij} \leq \bar{a}_{ij}|T_{ij}^s \setminus \bar{T}_{ij}^s|$, we have $(\Delta p_{ij} - \bar{a}_{ij}|T_{ij}^s \setminus \bar{T}_{ij}^s|) \leq 0$. As a result, we obtain $u_{ij}^\gamma = p_{ij} - \bar{a}_{ij}|T_{ij}^s| \leq \bar{p}_{ij} - \bar{a}_{ij}|\bar{T}_{ij}^s|$, *i.e.*, MUD γ_j cannot get a higher utility by bidding a_{ij}.

Case 3: $a_{ij} < \bar{a}_{ij}$ and MUD γ_j wins with a_{ij} but loses with $\bar{a}_{ij}$. In this case, we know that $\bar{a}_{ij}$ is higher than its critical neighbors' asking prices $\{a_{ik}\}$ or is higher than STO i's budget b_i. That is, $\bar{a}_{ij}|\bar{T}_{ij}^s| \geq p_{ij}$. Therefore, we have $u_{ij}^\gamma = p_{ij} - \bar{a}_{ij}|\bar{T}_{ij}^s| \leq 0$.

Case 4: $a_{ij} > \bar{a}_{ij}$ and MUD γ_j wins with $\bar{a}_{ij}$ but loses with a_{ij}. In this case, $u_{ij}^\gamma = 0$ which cannot be higher than the utility corresponding to $\bar{a}_{ij}$.

Case 5: $a_{ij} > \bar{a}_{ij}$ and MUD γ_j wins the auction with both $\bar{a}_{ij}$ and a_{ij}. Similar to the analysis of Case 2, we have $T_{ij}^s \subseteq \bar{T}_{ij}^s$ and $|T_{ij}^s| \leq |\bar{T}_{ij}^s|$. In addition, payment $\bar{p}_{ij}$ consists of two parts: i) the payments p_{ij} paid for time duration T_{ij}^s that is the same for both $\bar{a}_{ij}$ and a_{ij}; and ii) the payment $\Delta\bar{p}_{ij}$ paid for time duration $\bar{T}_{ij}^s \setminus T_{ij}^s$, in which $\bar{a}_{ij}|\bar{T}_{ij}^s \setminus T_{ij}^s| \leq \Delta\bar{p}_{ij} \leq a_{ij}|\bar{T}_{ij}^s \setminus T_{ij}^s|$ as $\bar{a}_{ij} \leq a_{ik} \leq a_{ij}$ for γ_j's every critical neighbor γ_k. Thus, the received utility is $u_{ij}^{\gamma} = p_{ij} - \bar{a}_{ij}|T_{ij}^s| \leq (p_{ij} - \bar{a}_{ij}|T_{ij}^s|) + (\Delta\bar{p}_{ij} - \bar{a}_{ij}|\bar{T}_{ij}^s \setminus T_{ij}^s|)$; that is, MUD γ_j's utility cannot be enhanced by submitting $a_{ij} > \bar{a}_{ij}$.

In summary, each STO i's auction CPAS is truthful for all MUDs. Furthermore, from Subsection 3.3.3, we can conclude that each MUD γ_j cannot increase the value of $\max\limits_{\pi_i \in \Pi(\gamma_j)} \{(p_{ij} - a_{ij}|T_{ij}^s|)\}$ via cheating on its asking price a_{ij} for each task π_i. Therefore, the auction scheme CPAS can achieve truthfulness for all MUDs.

3.4 Time Schedule-Preferred Auction Scheme

Notice that in the auction CPAS, a winning MUD's working time duration contains one or more sub-time durations. To allocate one single continuous time duration to each MUD, we propose a time schedule-preferred auction scheme termed **TPAS**, in which an STO first schedules the MUDs based on a first-come-first-serve manor in the time domain and then computes the payment for each winning MUD. The stages of TPAS for each STO i are outlined in Algorithm 3.

3.4.1 Information Publication & Collection

In this stage, each STO i publishes its task information f_i^{π} on the platform. Then, each MUD i submits its service information f_j^{γ} and asking price a_{ij} to STO i if the MUD is interested in task π_i.

Algorithm 3: Time Schedule-Preferred Auction Scheme for STO i

input : τ, k, g, δ_m
output: seed set S

Set $\{x_{ij}\} = \{0\}$, $\{T_{ij}^s\} = \{\emptyset\}$, and $T_i^u = [\alpha_i^\pi, \beta_j^\pi]$;
REPEAT
Publish sensing task information f_i^π;
Receive sensing service information $\{f_j^\gamma\}$ and asking prices $\{a_{ij}\}$ from the MUDs;
Run Alg. 4 to determine potential winners, schedule working time, compute payments, and announce the results;
Collect replies from the MUDs, record the values of $\{x_{ij}\}$, and update $T_i^u = T_i^u \setminus \bigcup_{j=1}^n x_{ij} T_{ij}^s$;
UNTIL $T_i^u = \emptyset$ or no potential winner is selected.

3.4.2 Potential Winner Determination & Payment Calculation

After obtaining the submitted service information, each STO i computes the set of available MUDs, $\Gamma^t(\pi_i)$, as follows:

$$\Gamma^t(\pi_i) = \{\gamma_j | (T_{ij} \cap T_i^u) \neq \emptyset \text{ and } R_j^\gamma \subseteq R_i^\pi\}.$$

Potential Winner Determination According to the first-come-first-serve policy, each STO i greedily assigns a working time duration to each available MUD γ_j according to the non-decreasing order in terms of the MUDs' arrival time $t_{ij}^\alpha = \{\frac{d(L_i^\pi, L_j^\gamma)}{\lambda_j^\gamma} + \alpha_j^\gamma\}$ until no available MUD can be selected or its unassigned working time duration T_i^u becomes empty. More specifically, in order to schedule an as long continuous working time duration as possible, STO i assigns each available MUD $\gamma_j \in \Gamma^t(\pi_i)$ a time duration from γ_j's prior MUD's ending working time to the time $\min\{\beta_i^\pi, \beta_j^\gamma\}$ if this time duration is unassigned. The pseudo-code of the scheduling scheme is presented in Algorithm 4.

Payment Calculation After completing task scheduling, each STO i first sorts all the available MUDs' asking prices in the non-decreasing order. Without loss of generality,

Algorithm 4: Time Schedule-Preferred Task Scheduling for Task π_i

input : f_i^π, T_i^u, Γ, $\{f_j^\gamma\}$
output: $\{T_{ij}^s\}$

Set $\Gamma^t(\pi_i) = \emptyset$ and $\{T_{ij}^s\} = \{\emptyset\}$ for $\forall \gamma_j \in \Gamma(\pi_i)$;
for *each* $\gamma_j \in \Gamma$ *with submitted* f_j^γ *and* a_{ij} **do**
    Calculate t_{ij}^α and T_{ij};
    if $(T_{ij} \cap T_i^u) \neq \emptyset$ *and* $R_j^\gamma \subseteq R_i^\pi$ **then**
        $\Gamma^t(\pi_i) = \Gamma^t(\pi_i) \cup \gamma_j$;
Sort all MUDs in $\Gamma^t(\pi_i)$ in the non-decreasing order based on $\{t_{ij}^\alpha\}$ and get the sorted set $\Gamma^{t\prime}(\pi_i)$;
Set $Start = \alpha_i^\pi$;
for $j = 1$ *to* $|\Gamma^{t\prime}(\pi_i)|$ **do**
    if $Start < \min\{\beta_j^\gamma, \beta_i^\pi\}$ *and* $(T_{ij} \cap T_{ij}^u) \cap (\bigcup_{j'=1}^{j-1} T_{ij'}^s) \neq (T_{ij} \cap T_{ij}^u)$ **then**
        $T_{ij}^s = [Start, \min\{\beta_j^\gamma, \beta_i^\pi\}]$;
        $Start = \min\{\beta_j^\gamma, \beta_i^\pi\}$.

in set $\Gamma^t(\pi_i)$, we simply assume that

$$a_1 \leq a_2 \leq \cdots \leq a_{|\Gamma^t(\pi_i)|}.$$

Then, each STO i searches for a maximum index k_i^π such that $a_{k_i^\pi} \leq b_i < a_{k_i^\pi+1}$, initializes the set of potential winners $W(\pi_i) = \emptyset$, and determines winners according to the following two cases.

- Case 1: $T_{ik_i^\pi}^s \neq \emptyset$. If $\gamma_j \in \Gamma^t(\pi_i)$, $1 \leq j \leq k_i^\pi$, and $T_{ij}^s \neq \emptyset$, set $W(\pi_i) = W(\pi_i) \cup \gamma_j$ and $p_{ij} = b_i|T_{ij}^s|$. Moreover, in this case, b_i is the *critical price* of all he MUDs in STO i's auction.
- Case 2: $T_{ik_i^\pi}^s = \emptyset$. If $\gamma_j \in \Gamma^t(\pi_i)$, $1 \leq j < k_i^\pi$, and $T_{ij}^s \neq \emptyset$, set $W(\pi_i) = W(\pi_i) \cup \gamma_j$ and $p_{ij} = a_{ik_i^\pi}|T_{ij}^s|$. In this case, MUD $\gamma_{k_i^\pi}$ and $a_{ik_i^\pi}$ are the *critical neighbor* and the *critical price* of all the MUDs in STO i's auction, respectively.

Next, each STO i notifies the MUDs of the auction results.

3.4.3 Final Service Decision

When learning the auctions results, each MUD γ_j makes its final decision using the method the same as that of Subsection 3.3.3. Finally, the results of $\{x_{ij}\}$ can be obtained.

3.4.4 Property Analysis

In this subsection, we theoretically prove the performance of the auction scheme TPAS in terms of computational efficiency, individual-rationality, budget-balance, and truthfulness.

Lemma 3. *The computational complexity of the scheduling scheme Algorithm 4 is $O(n\log(n))$.*

Proof: From line 4 to line 9, the construction of set $\Gamma^t(\pi_i)$ can be done within $O(n)$. In line 10, the sorting process can be completed within $O(n\log(n))$. From line 12 to line 17, the scheduling process contains at most n iterations and each iteration has a time complexity of $O(1)$. Therefore, the computational complexity of Algorithm 4 is $O(n\log(n))$.

Lemma 4. *The computational complexity of payment calculation in the auction scheme TPAS is $O(n)$.*

Proof: To compute the payments, each STO i has to search for a maximum index k_i^π by scanning set $\Gamma^t(\pi_i)$. As $|\Gamma^t(\pi_i)| \leq n$, the computation complexity of payment calculation is $O(n)$.

Theorem 5. *The proposed auction scheme TPAS achieves computational efficiency with a time complexity of $O(n^2\log(n))$.*

Proof: From Lemmas 3 and 4, and the analysis of Theorem 1, this theorem

Theorem 6. *The proposed auction scheme TPAS is individually-rational for all MUDs.*

Proof: When all STOs' auctions end, there are two cases for each MUD γ_j:

- If $\sum_{i=1}^{m} x_{ij} = 0$, we have $p_{ij} = 0$ and $|T_{ij}^s| = 0$ for each $\pi_i \in \Pi$. Thus, $U_j^\gamma = 0$.
- If $\exists \pi_i \in \Pi, x_{ij} = 1$, we have $U_j^\gamma = u_{ij}^\gamma = p_{ij} - a_{ij}|T_{ij}^s| \geq 0$ as $p_{ij} \geq a_{ij}|T_{ij}^s|$ and $|T_{ij}^s| > 0$.

Therefore, we can conclude that TPAS achieves individual rationality for all MUDs.

Theorem 7. *The proposed auction scheme TPAS is budget-balanced for all STOs.*

Proof: If sensing task π_i is successfully assigned to one or more MUDs, we have $\sum_{j=1}^{n} x_{ij} \geq 1$, $\sum_{j=1}^{n} x_{ij}|T_{ij}^s| > 0$, and $p_{ij} \leq b_i|T_{ij}^s|$ for every winning MUD γ_j. Thus, for STO i, we have $\sum_{j=1}^{n} x_{ij}b_i|T_{ij}^s| - \sum_{j=1}^{n} x_{ij}p_{ij} \geq 0$, indicating that TPAS can ensure budget-balance for all STOs.

Lemma 5. *For each STO i, the scheduling results $\{T_{ij}^s\}$ of Algorithm 4 are independent of all MUDs' asking prices $\{a_{ij}\}$.*

Proof: From line 12 to line 16 of Algorithm 4, one can see that the computation of γ_j's working time duration T_{ij}^s does not depend on its asking price a_{ij}. Therefore, this theorem holds.

Lemma 6. *In each STO i's location auction TPAS, if MUD γ_j is a potential winner with bidding a price a_j, it can also become a potential winner with a smaller price $a_{ij}^1 < a_{ij}$.*

Proof: When MUD γ_j submits a smaller price a_{ij}^1, γ_j's position in set $\Gamma^t(\pi_i)$ changes from j to j^1. Since $a_{ij}^1 < a_{ij}$, we have $j^1 \leq j \leq k_i^\pi$. In addition, according to Lemma 5, the assigned working time duration T_{ij}^s remains the same for MUD γ_j. As a result, γ_j can be still selected as a potential winner by STO i for task π_i.

Theorem 8. *The proposed auction scheme TPAS can achieve truthfulness for all MUDs.*

Proof: Proving this theorem is equivalent to prove that in each STO i's local auction TPAS, each MUD $\gamma_j \in \Gamma$ cannot improve its utility u_{ij}^γ by asking for a price $a_{ij} \neq \bar{a}_{ij}$, in which there are five cases to be considered for each MUD γ_j.

Case 1: $a_{ij} < \bar{a}_{ij}$ (or $a_{ij} > \bar{a}_{ij}$) and MUD γ_j loses the auction with both a_{ij} and $\bar{a}_{ij}$. In this case, γ_j's utility received from STO i is zero.

Case 2: $a_{ij} < \bar{a}_{ij}$ and MUD γ_j wins the auction with both a_{ij} and $\bar{a}_{ij}$. Through Lemma 5, we know that the assigned time duration is T_{ij}^s for MUD γ_j with both a_{ij} and $\bar{a}_{ij}$. In addition, from the property of the pricing method in TPAS and Lemma 6, we have

$a_{ij} < \bar{a}_{ij} \leq a_{ik_i^{\pi}} \leq b_i$, i.e., $a_{ij}|T_{ij}^s| < \bar{a}_{ij}|T_{ij}^s| \leq p_{ij}$. Therefore, the utility also remains the same, *i.e.*, $u_{ij}^{\gamma} = p_{ij} - \bar{a}_{ij}|T_{ij}^s|$.

Case 3: $a_{ij} < \bar{a}_{ij}$ and MUD γ_j wins with a_{ij} but loses with $\bar{a}_{ij}$. In this case, $\bar{a}_{ij}$ is higher than the critical price $a_{ik_i^{\pi}}$ or is higher than STO i's budget b_i. Thus, we have $\bar{a}_{ij}|T_{ij}^s| \geq p_{ij}$ according to the pricing method in Subsection 11. As a result, the utility is $u_{ij}^{\gamma} = p_{ij} - \bar{a}_{ij}|T_{ij}^s| \leq 0$.

Case 4: $a_{ij} > \bar{a}_{ij}$ and MUD γ_j wins with $\bar{a}_{ij}$ but loses with a_{ij}. In this case, $u_{ij}^{\gamma} = 0$ which cannot be higher than the utility corresponding to $\bar{a}_{ij}$.

Case 5: $a_{ij} > \bar{a}_{ij}$ and MUD γ_j wins the auction with both $\bar{a}_{ij}$ and a_{ij}. Similar to Case 2, we have the following relationships: i) T_{ij}^s for MUD γ_j with both a_{ij} and $\bar{a}_{ij}$ from Lemma 5; and ii) $\bar{a}_{ij} < a_{ij} \leq a_{ik_i^{\pi}} \leq b_i$ due to the property of the pricing method in TPAS and Lemma 6. Thus, the utility is unchanged, *i.e.*, $u_{ij}^{\gamma} = p_{ij} - \bar{a}_{ij}|T_{ij}^s|$.

The above five cases indicate that each STO i's auction is truthful for all MUDs. Moreover, from Subsection 3.3.3, one can see that each MUD γ_j cannot increase the value of $\max\limits_{\pi_i \in \Pi(\gamma_j)} \{(p_{ij} - a_{ij}|T_{ij}^s|)\}$ via cheating on its asking price a_{ij} for each task π_i. Therefore, the auction scheme TPAS can ensure truthfulness for all MUDs.

Remark: In the auction scheme TPAS, the process of task scheduling is independent of the MUDs' asking prices. As a result, an MUD that has been assigned a non-empty time duration cannot win the auction if the MUD's asking price is higher than the corresponding critical price. In fact, any price-independent scheduling algorithm can be applied in TPAS to obtain $\{T_{ij}^s\}$, without any impact on truthfulness for the MUDs.

3.5 Performance Evaluation

3.5.1 Simulation Settings

We evaluate the performance of the cost-preferred auction scheme (CPAS) and time schedule-preferred auction scheme (TPAS) based on a synthetic data set. The number of sensing tasks varies from 5 to 30, and the number of MUSs varies from 50 to 150. The locations of all sensing tasks are randomly and uniformly distributed within a rectangular area of 6km$\times$6km. The locations of all MUDs are set according to two kinds of distributions: i) **uniform:** the locations are uniformly deployed within the rectangular area of 6km $\times$ 6km at random; and ii) **hotspot:** the location of each task is viewed as a circle-centered hotspot with a radius of 0.7km, and the MUDs randomly locate within these hotspot areas. We consider 10 types of sensors and the number of each type of sensor is one. Each sensing task requests a number of different sensors, which is a random number uniformly picked from $[3, 10]$; similarly, each MUD is equipped with a number of different sensors, which is a random number uniformly chosen from $[1, 10]$. In the simulation, the unit time slot is one minute and the longest time duration is 5 hours. More specifically, each sensing task (and each MUD's available time) randomly begins at or after 1:00pm and ends at or before 5:00pm; that is, for any task (and any MUD), the working time duration is at most 5 hours containing 300 time slots. Each STO's budget (and each MUD's asking price) is an integer that is uniformly selected from $[10, 25]$ at random. Suppose that all the MUDs are pedestrians with mobile devices, so the moving rate of each MUD is randomly and uniformly chosen within [4.5, 5.4]*km/hour* [54].

We use the following metrics to evaluate the performance of the two proposed auction schemes.

- **Allocation Efficiency.** The allocation efficiency of a sensing task is defined to be the ratio of the total number of assigned working time slots to the number of requested working time slots. Formally, the average allocation efficiency of all sensing tasks is calculated as
$$\frac{1}{m}\left(\sum_{i=1}^{m}\frac{\sum_{j=1}^{n}x_{ij}|T_{ij}^{s}|}{\beta_i^{\pi}-\alpha_i^{\pi}}\right).$$
- **Working Time Utilization.** The working time utilization of an MUD is the ratio of the number of assigned working time slots to the number of available working time

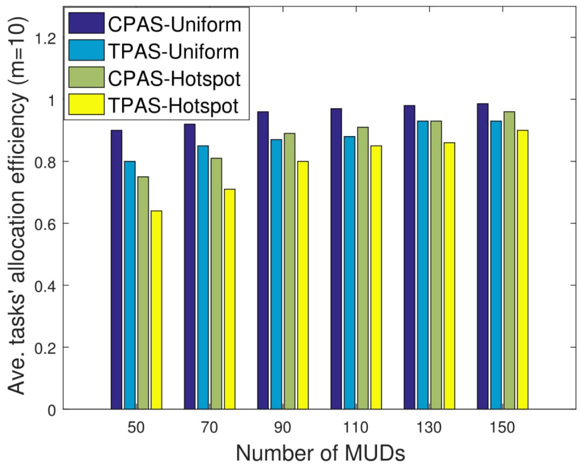


Figure 3.2. Average tasks' allocation efficiency (m=10).

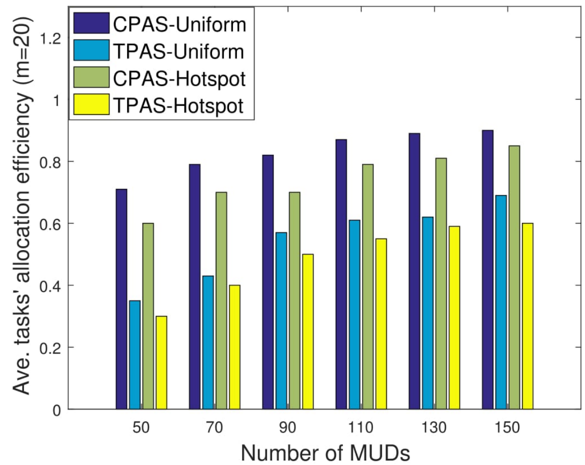


Figure 3.3. Average tasks' allocation efficiency (m=20).

slots. Thus, the average working time utilization of all MUDs is computed by

$$\frac{1}{n}\left(\sum_{j=1}^{n} \frac{\sum_{i=1}^{m} x_{ij}|T_{ij}^{s}|}{\beta_j^{\gamma} - \alpha_j^{\gamma}}\right).$$

- **Utility.** We also compare the average utility of all MUDs.
- **Truthfulness.** At each time, we randomly pick an MUD, set fake asking prices to it, and examine its received utility when biding truthfully and untruthfully.

3.5.2 Simulation Results and Analysis

We first check the average allocation efficiency of all sensing tasks with the number of STOs changing from 10 to 20 and the number of MUDs increasing from 50 to 150 under the uniform and hotspot distributions. The results are presented in Fig. 3.2 and Fig. 3.3. As shown in Fig. 3.2 and Fig. 3.3, the average allocation efficiency increases when the number of MUDs increases. This is because if more MUDs participate in the auction, each STO can find more MUDs to implement its sensing task. Besides, in both Fig. 3.2 and Fig. 3.3, CPAS performs better than TPAS. The reason lies in the following two aspects: i) CPAS assigns each MUD one or more working time durations while TPAS assigns each MUD at most one working time duration; ii) the working time scheduling of TPAS is independent of the MUDs' asking prices, leading to that some MUDs who have been scheduled a working time duration may be losers if their prices are higher than the critical price. In other words, in CPAS, a larger portion of required working time duration can be scheduled to the MUDs, getting a higher allocation efficiency for each STO. Moreover, CPAS under the uniform distribution can achieve a higher allocation efficiency than CPAS under the hotspot distribution. The same situation also occurs to TPAS. This is due to the fact that under the hotspot distribution, the number of MUDs that are near to any STO is reduced, indicating that the number of available MUDs becomes smaller for each STO.

Next, we analyze the average working time utilization and the average utility of all MUDs with 110 MUDs and the number of sensing tasks increasing from 5 to 30. The results of the average working time utilization and the average utility are plotted in Fig. 3.4 and Fig. 3.5, respectively. From Fig. 3.4 and Fig. 3.5, we obtain the following two observations. First, one can see that the average working time utilization and the average utility gradually increase as more and more sensing tasks are published, because it becomes less competitive for the MUDs to be assigned more working time slots when more tasks are available. Second, CPAS (respectively TPAS) under the uniform distribution obtains a larger average working time utilization and a higher average utility than CPAS (respectively TPAS) under the hotspot utilization. The reason is that an MUD is usually selected by an STO who is near to it but under the hotspot distribution, and the number of STOs who are near to any MUD is decreased, *i.e.*, the probability of becoming a winner is reduced for each MUD.

Furthermore, we verify truthfulness of CPAS and TPAS, in which there are 20 tasks and 110 MUDs under each distribution scenario. In the simulation, we randomly pick one MUD at a time, set fake asking prices to the picked MUD, and compare its received utilities when truthful bidding and untruthful bidding. We totally select five different MUDs under each distribution scenario and present the results in Figs.3.6-3.9. Notice that all the selected MUDs cannot receive higher utilities via cheating on asking prices. For example, in Fig.3.6, the selected MUDs are the 3rd MUD, the 25th MUD, the 36th MUD, the 89th MUD, and

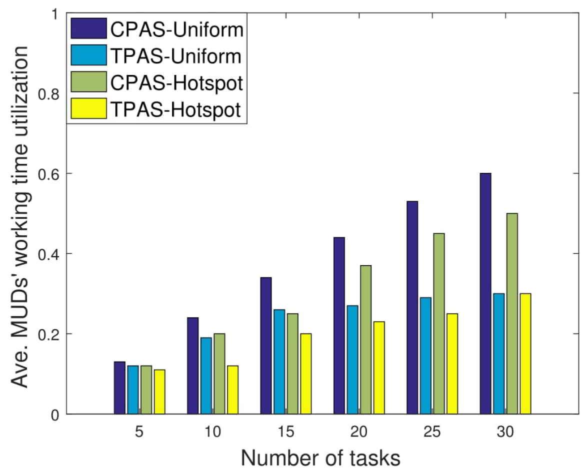


Figure 3.4. Average MUDs' working time utilization.

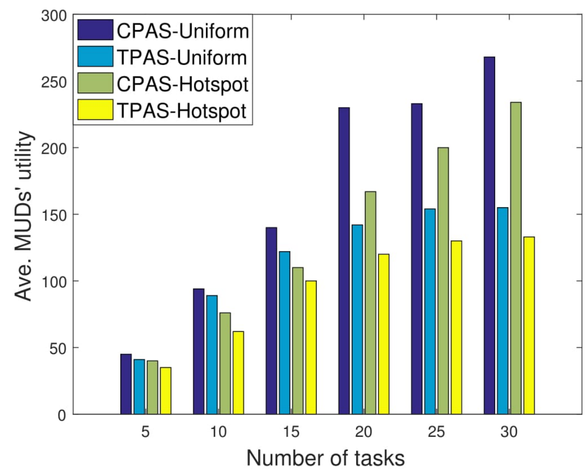


Figure 3.5. Average MUDs' utility.

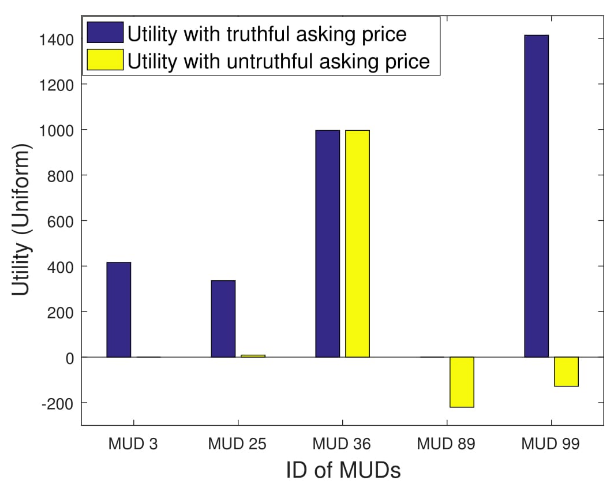


Figure 3.6. MUDs' truthfulness in CPAS under uniform distribution.

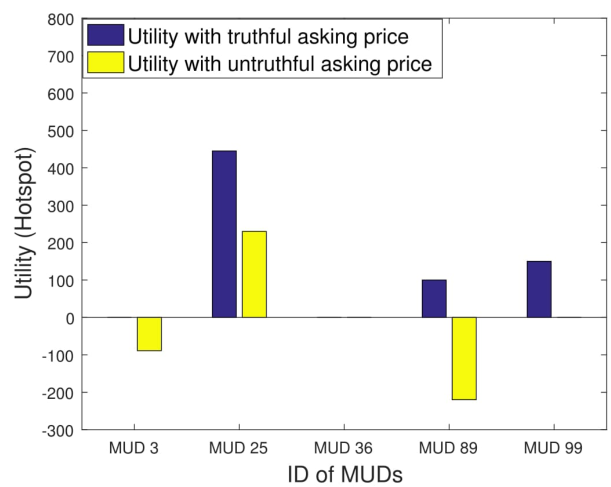


Figure 3.7. MUDs' truthfulness in CPAS under hotspot distribution.

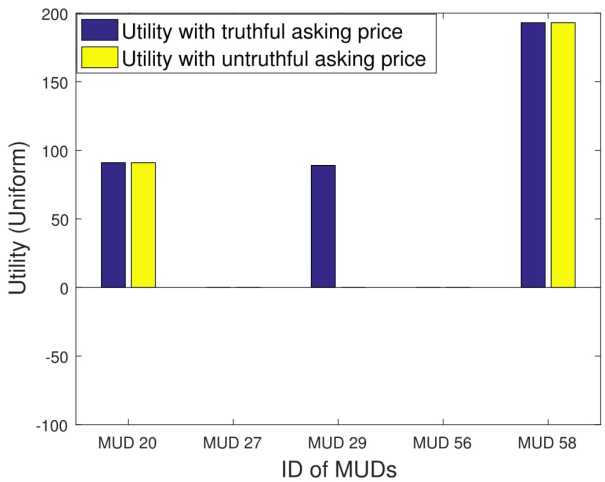


Figure 3.8. MUDs' truthfulness in TPAS under uniform distribution.

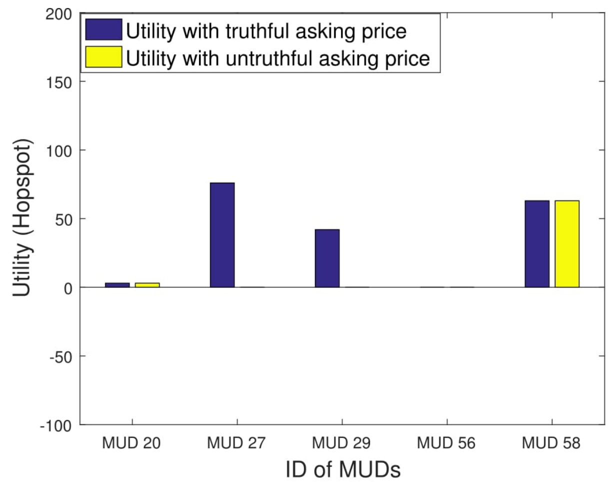


Figure 3.9. MUDs' truthfulness in TPAS under hotspot distribution.

the 99th MUD. More specifically, the situations of the five MUDs are illustrated in the following: i) the 3rd MUD is a winner when truthful bidding but a loser when cheating; ii) the 25th MUD wins the auction when bidding truthfully and untruthfully, but its utility is reduced when bidding untruthfully; iii) the 36th MUD receives the same utility when bidding truthfully and untruthfully; iv) the 89th MUD obtains a zero utility with truthful prices but a negative utility with fake prices; and v) the 99th MUD's utility is reduced to a negative value when cheating.

3.6 Summary

To motivate mobile users to join sensing tasks in MCSs, we propose a reverse auction model and two novel distributed auction schemes, CPAS and TPAS, for task assignment and scheduling. Specifically speaking, the novelty of the proposed auction model and auction schemes lies in the following aspects: i) the auction model is practical taking into account partial fulfillment, attribute diversity, and price diversity; ii) the two auction schemes can be implemented within a well-designed distributed auction framework; iii) both two auction schemes are proved to be computationally efficient, individually rational, budget balanced, and truthful.

Chapter 4

TRUTHFUL INCENTIVE MECHANISMS FOR SOCIAL COST MINIMIZATION IN MOBILE CROWDSOURCING SYSTEMS

4.1 Introduction

Nowadays, new emerging embedded technology drives the rapid growth of mobile devices. With powerful processors, mobile devices such as smartphones, tablets, and watches can be used as portable computers to undertake heavy computational tasks. With the help of embedded sensors like Global Position System (GPS), accelerometers, and cameras, mobile devices can be used to sense and deliver information. On the other hand, the utilization of mobile devices is ubiquitous. Some works [55, 56] show that almost 64% of adults own a smartphone and 42% of adults own a tablet in America as of October 2014. All of the above conditions, along with the mobility of users who carry these mobile devices and the convenient communication infrastructures enable mobile devices to connect to the Internet, stimulating the development of Mobile Crowdsourcing Systems (MCSs). The MCS is a new system model used to outsource tasks. Generally speaking, two types of participants exist in a MCS. One is the crowdsourcing platform, which acts as the server to publish tasks, determine the set of mobile devices to work on the tasks, and collect the final results. The other participants consist of users with mobile devices. They can participate to finish the tasks published by the crowdsourcing platform and get payments as rewards. Lots of tasks can be done by a MCS, such as information collection, environmental monitoring, or customized survey. These tasks used to be performed by a specialist or an expert, but now through the MCS can be done by a group of undefined users with mobile devices.

A variety of MCS applications can be found in our daily life, among which applications focusing on the environment, infrastructure, and social activities are the three most popular categories. In the environmental MCS applications, such as Common Sense and

Creek watch (introduced in [1, 57], respectively), mobile devices can be used to monitor the environmental pollution levels. For example, microphones on mobile devices can monitor the noise information of a place and pictures can be taken by cameras to show the amount of trash in a park in Common Sense. Existing applications of the Infrastructure interested in the detection of traffic congestion, parking availability, and outages of public works. For example, mobile devices with CarTel [58] installed on cars can detect the speed and location of cars, and send the detected information to a data center. ParkNet [59] can help cars to find available parking places. Applications regarding social activities take advantage of users' willingness to share sensed information with each other [3, 24]. All MCS applications require the participation of hundreds or thousands of mobile devices without deploying any static sensors or machines.

The enormous utilization potential of MCSs attracts lots of attention from researchers [34, 38]. One of the most popular topics in MCSs is how to determine the best set of mobile devices to allocate the tasks (such as computational or sensing tasks) published by a MCS platform so that a predefined objective can be achieved. A commonly used objective is to optimize the social efficiency, such as maximizing social welfare or minimizing social cost. The fundamental of a MCS is to have enough participants. However, working for MCS platforms will consume users' resources, including execution capacity and battery. Joining a MCS will also put a threat on users' privacy. For example, the results submitted to MCS platform may expose users' locations. Considering the above-mentioned facts, some users may refuse to participate in the MCS. If the number of users with mobile devices is insufficient, the objective is impossible to be achieved. Thus, a MCS platform should provide enough reward for participants for incentive purposes.

In this chapter, we focus on the design of an incentive mechanism for a MCS to minimize the social cost. The social cost represents the total cost of mobile devices when all tasks published by the MCS are finished. To achieve the objective in the MCS, we confront several challenges:

- *True cost revelation.* The cost of each mobile device for finishing a task is private. It is difficult to encourage all participants to report their real costs.

- *Minimal cost optimization.* Assume all users report their true costs to the MCS. Since mobile devices may vary in capacities and costs, it is hard to select the optimal set of users.

- *Incentive mechanism.* As discussed, the MCS platform should reward each user who works for it as incentives. Within the budget, the reward should be greater than the cost of the user. Deciding a proper reward for each participant is still challenging.

These challenges lead us to investigate an auction mechanism that concentrates on the trade between the MCS platform and mobile users. This paper begins with the assumption that the MCS platform publishes only one task in one round and the task consists of pieces of sub-tasks. Each user with a mobile device can work for one or more sub-tasks. The auction mechanism in our paper aims to minimize the social cost of mobile users while guaranteeing the truthful cost of bidding from each participating user.

Depending on the requirements of a MCS platform, there are two different working patterns. The first one is the *continuous* working pattern, which requires each participant to work on a set of continuous sub-tasks. We call another working pattern the *discontinuous* working pattern, where a participant can work for any set of sub-tasks. The detailed definition of the two working patterns are discussed in Section 4.3.

The main contributions of this chapter are as follows [60]:

1. The social cost minimization problem in a MCS has been discussed. We first present the working process of a MCS, and then build an auction market for the MCS where the MCS platform acts as an auctioneer and users with mobile devices act as bidders.

2. Depending on the different requirements of the MCS platform, we design a Vickrey-Clarke-Groves (VCG)-based auction mechanism for the continuous working pattern and a suboptimal auction mechanism for the discontinuous working pattern. Both of

them can ensure that the bidding of users are processed in a truthful way and the utilities of users are maximized.

3. Experiments are conducted to verify performances of the proposed mechanisms. Results suggest that the two auction mechanisms achieve truthfulness and utility maximization. In addition, the VCG-based mechanism could guarantee the minimum social cost and the suboptimal mechanism is more computationally efficient.

The remainder of this chapter is organized as follows: In Section 4.2, we present the MCS model and analyze its working process. The auction problem is defined in Section 4.3, followed by two truthful auction mechanisms presented in Section 4.4. The experimental results and discussions are provided in Section 4.5. Conclusions and future works are shown in the last section.

4.2 System Model Overview

Figure 5.1 demonstrates an example of the Mobile Crowdsourcing System (MCS). The model includes two types of participants: a Crowdsourcing Platform (CP) and lots of Mobile Users with Devices (MUDs). The CP consists of several servers, which are deployed in the cloud and provide services for clients. A smartphone or a tablet carried by a user is regarded as a MUD. The CP communicates with MUDs via cellular networks or WiFi. The CP publishes a computational or a sensing task which contains a series of sub-tasks. Each subtask only occupies a time slot. Any two time slots have no intersection. Each MUD is allowed to work on one or more time slots and provide computational or sensing services to the CP during these time slots. The concepts sub-task and time slot are used interchangeably in this chapter. Working for computational or sensing tasks will bring battery consumption, computing capacity consumption, and privacy threats to MUDs. Thus, in order to stimulate MUDs' participation, the CP rewards these users who have been selected to provide serves. For simplicity, we assume that the CP publishes one task in each round.

In general, as shown in Figure 4.2, the interactive process between the CP and MUDs have three stages in each round, including the publishing stage, auction stage, and working stage:

Publishing stage. In this stage, the CP decides the task that it plans to finish in this round. It generates the description of the task according to predefined functions and then publishes it among all participated MUDs.

Auction stage. After receiving the task description and requirements, each MUD decides its working plan. That is the subset of sub-tasks of task k it can work for. If a MUD can work for task k, it will continue to evaluate its cost. The MUD calculates its base price and submits a bid to the CP. The bid of a MUD consists of its working plan and the base price. After receiving the bids from MUDs, the CP will choose the winner set of MUDs, make the work schedule, determine their rewards and then announce the auction results to all participated MUDs.

Working stage. For each task k, its working stage starts at the start time a_k and ends at the end time d_k. During this stage, the CP will activate the MUDs in the winner set one by one based on its working schedule. Once activated, a MUD begins to work according to the requirements and then submits the result to the CP. The reward is given to the MUD once it finishes its claimed sub-tasks.

Different from the auction stage, the other three stages in the MCS are beyond the scope of this chapter. We focus on designing efficient and effective auction mechanisms. Table 4.1 lists frequently used notations.

4.3 Problem Formulation in Auction Stage

The working process of a MCS can be divided into infinite rounds with time. For any two rounds, their tasks are independent and the available MUDs are independent as well. Thus, we focus our investigation on the discussion of one round in detail. Assume there is a dense set of MUDs, represented as $V_k = \{v_1,\ v_2,\ \dots,v_i,\ \dots,\ v_N\}$, where $|V_k| = N$. At the beginning of round k (k is an integer denoting the identifier of one round), the CP publishes the task description R_k, defined as:

$$R_k = \{a_k, d_k, \mathrm{T}_k, \Omega_k,\ \Pi_k\},$$

where a_k and d_k are the start and end time of task k, respectively. $T_k = \{\tau_{k1},\ \tau_{k2},\ ...,\ \tau_{kj},\ ...,\ \tau_{kM}\}$ represents the set of sub-tasks of task k. Each time slot represents a sub-task of task k. M is the size of T_k (the number of sub-tasks in task k). The CP requires MUDs to work for task T_k. As shown in Figure 4.3, on the one hand, the durations of any two time slots $\tau_{ki} \in T_k$ and $\tau_{kj} \in T_k$, where $i \neq j$, may vary from each other. On the other hand, all these time slots are distributed over the time line. The interval between two adjacent time slots could be larger than or equal to 0, but not smaller than 0, which means that no overlapping time interval exists between any two adjacent time slots. Ω_k indicates the hardware requirements of task k on MUDs. Hardware requirements contain minimum computation speed, free storage capacities, and sensor types. It requires that only the MUDs satisfying the requirements can bid for the sub-tasks at this round. Π_k indicates the requirement of task k regarding MUDs' working patterns. There are two kinds of working patterns: the continuous pattern ($\Pi_k = C$) and the discontinuous pattern ($\Pi_k = \bar{C}$).

Continuous case (C): $\forall\ v_i \in V_k$, this working pattern requires the sub-tasks set S_i claimed by v_i are continuous (S_i represents the set of sub-tasks v_i can work for). That is, v_i is able to work continuously from the earliest sub-task to the last sub-task in S_i. For example, suppose five sub-tasks are included in task k, as shown in Figure 4.3. $S_i = \{\tau_{k2},\ \tau_{k3},\ \tau_{k4}\}$ is an example that v_i works in a continuous working pattern, while $S_i = \{\tau_{k2},\ \tau_{k3},\ \tau_{k5}\}$ is not.

Discontinuous case ($\bar{C}$): $\forall\ v_i \in V_k$, in this case, v_i can work for any subset of T_k. For example, both $S_i = \{\tau_{k2},\ \tau_{k3},\ \tau_{k4}\}$ and $S_i = \{\tau_{k2},\ \tau_{k3},\ \tau_{k5}\}$ can be regarded as examples of discontinuous working patterns.

$\forall\ v_i \in V_k$, after receiving a task description from the CP, if its hardware qualifies, v_i will decide the subset of sub-tasks, denoted as $S_i \subseteq T_k$ according to the working patterns requirements of task k. Then, $\forall\ v_i \in V_k$, the bid of v_i can be represented as,

$$b_i = \{\Omega_i,\ \Pi_i,\ S_i,\ A_i\ \},$$

where A_i is v_i's asking price when it works on the sub-tasks in S_i for the CP, which is also the base price. Because auction mechanisms are expected to be truthful, so $A_i = c_i$. In practice, the value of c_i can be estimated by v_i. A_i and c_i are used interchangeably in this chapter. For convenience, all MUDs' costs are restricted to follow simple cost functions which makes all MUDs *single-minded MUDs.*

Definition 4.3.1. *A cost function $\mathbb{C}$ is called single-minded if there exists a set of sub-tasks $S \subseteq T_k$ and a cost c, $\mathbb{C}(S^*) = c$ for all allocations $S^* \subseteq S$ and $\mathbb{C}(S^*) = \infty$ for all other S^*. A MUD bids with S and c is single-minded.*

Definition 1 shows that the base price of each v_i will be same even though the set of sub-tasks S_i^* allocated to v_i is a subset of S_i in its bid after the auction stage. One step further, the MUD v_i wouldn't accept any allocation S^*, where $\exists\ \tau \in T_k,\ \tau \in S^*$ *but* $\tau \notin S$.

In the auction stage, the CP can be regarded as an auctioneer who makes the decision for the sub-tasks allocation and payment. MUDs act as bidders to make bids. We are interested in minimizing the social cost from a macroscopic and social perspective in this chapter. The social cost is defined as the cost brought by the trading within the MCS. Formally, the objective can be written as,

$$\begin{aligned} &Minimize \quad \sum_{v_i \in W_k} (P_i + (c_i - p_i)) \\ &s.t. \quad W_k \subseteq V_k,\ \ T_k = \bigcup_{v_i \in W_k} S_i \end{aligned} \tag{4.1}$$

where W_k is the set of winner MUDs in this round. P_i is the price paid by the CP for using the computational or sensing services provided by v_i, which can be regarded as the cost of the CP. p_i is the payment winner v_i received from the CP when the assigned sub-tasks are done, and c_i is the cost of v_i. So, the social cost of winner v_i is $c_i - p_i$. For effectiveness, each sub-task τ in T_k needs at least one MUD to work.

Based on the predefined model, we have $P_i = p_i$ for each $v_i \in W_k$. Hence, the objective above can be rewritten as

$$\begin{aligned} &Minimize \quad \sum_{v_i \in W_k} c_i, \\ &s.t. \quad W_k \subseteq V_k, \;\; T_k = \bigcup_{v_i \in W_k} S_i. \end{aligned} \tag{4.2}$$

Within the auction stage, the auctioneer CP should design a proper auction mechanism with efficient sub-tasks allocation and rewards determination to minimize the social cost. Under the auction mechanism chosen by the CP, each participated MUD bids with a strategy which maximizes its utility. The utility of v_i (denoted by U_i) in one round can be defined as:

$$U_i = \begin{cases} p_i - c_i & v_i \; wins, \\ \\ 0 & otherwise. \end{cases} \tag{4.3}$$

In order to achieve the objective in an efficient and effective way, the auction mechanisms used by the CP should have the following properties:

Individual Rationality. All MUDs are self-interested to benefit themselves. Thus, the utility of any MUD in each round should be non-negative: $U_i \geq 0$.

Truthfulness. The mechanisms are considered truthful when the four values $(\Omega_i,\ \Pi_i,\ S_i,\ A_i)$ in the bid of each MUD are truthful. The utility of v_i will be maximized when it bids truthfully and v_i cannot improve its utility through any misreport,

$$U_i(b_i, \mathbf{b}_{-i}) \geq U_i(\hat{b}_i, \mathbf{b}_{-i}), \tag{4.4}$$

where $\mathbf{b}_{-i} = \{b_1,\ \ldots\ ,\ b_{i-1},\ b_{i+1},\ \ldots\ ,\ b_n\ \}$ represents the set of truthful bids of all MUDs excluding v_i. b_i is the truthful bid of v_i, and $\hat{b}_i \neq b_i$. If an auction mechanism satisfies this property, *Nash Equilibrium* exists [61]. Misreports of the first two values (hardware parameters and working pattern of a MUD) in a bid can be easily detected by the CP through the submitted results from MUDs. Thus, the truthfulness of the first two values is guaranteed. We focus on the truthfulness of the last two values in a bid: claimed set of sub-tasks and asking price.

Computational Efficiency. An auction mechanism is considered computationally efficient if the task allocation and payment decision can be made in polynomial time.

Only when the above three properties are satisfied at the same time can an auction mechanism be regarded as useful. Without individual rationality, a MUD may receive negative utility and refuse to participate in the MCS. Then, because the c_i in bid b_i is private to v_i, the CP wouldn't know it. If an auction mechanism is truthful, all MUDs only need to bid with their true costs: $A_i = c_i$, which not only simplifies the strategies, but also avoids manipulation. Finally, computational efficiency will guarantee that the auction mechanism can be practically implemented.

4.4 Design of Incentive Auction Mechanisms

Formally, an auction mechanism contains two phases: winner MUDs set the selection and payment decision. Specifically, the most challenging and important part of the auction mechanism design is truthfulness. According to the characterization of truthful auction mechanism concluded in [61], we have:

Theorem 9. *For any fixed bids $\boldsymbol{b}_{-i}$, an auction mechanism is truthful to MUD v_i if and only if the winner MUDs set selection algorithm is monotone and the payment for each winner MUD is critical.*

If the MUD v_i is selected as a winner when it bids with S_i^* and A_i^*, and v_i will still be selected for any (S_i', A_i'), where $S_i' \supseteq S_i^*$ and any bid with $A_i' \leq A_i^*$. The process in the selection of winner MUDs set is monotone.

There exists a critical payment ${c^c}_i$ *for each winner MUD* v_i*, which is independent to the asking price* a_i *in its bid.* v_i *will win when it bids with any* $(S_i,\ A'_i)$*, where* $A'_i \leq {c^c}_i$*. Otherwise,* v_i *loses.*

4.4.1 Mechanism with Optimal Social Cost

Based on Section 4.3, we can tell that different tasks may have different working pattern requirements. If a task k requires MUDs to work in the continuous pattern ($\Pi_k = C$), the optimal solution to the minimization Problem (4.2) can be achieved through dynamic programming. Thus, in this case, a well known VCG-based auction mechanism is a good choice. The detailed design is as follows:

Step 1, Given the bids of all candidate MUDs, use the dynamic programming method, shown in Algorithm 1, to compute the optimal winner set W. In Algorithm 1, Line (8) represents the optimal substructure, recording the minimal cost for sub-task τ in T if adding the v_n to the winner set W. Notation $\tau.MUD$ is used to indicate that the winner MUD will work for the sub-task τ. This algorithm returns three results: the winner set W, the work schedule, and minimal social cost of MUDs in W.

The complexity of Algorithm 1 is $O(mn)$, which indicates that the VCG-based auction is practical.

Step 2, Calculate the payment p_i of each v_i in the winner set W. The payment for each winner v_i is defined as the increase in the total social cost brought by its contribution, as

$$p_i = \sum_{v_j \in W_{-i}} c_j \ - \sum_{v_j \in W,\ v_j \neq v_i} c_j, \tag{4.5}$$

where W_i is the obtained winner set without v_i's participation.

The truthfulness and individual rationality of VCG-based mechanisms have been proven in [61].

4.4.2 Mechanism with Suboptimal Social Cost

If the working pattern is discontinuous, the winner set determination Problem (4.2) can be regarded as a classical set cover problem. Because the set cover problem has been proven to be NP-hard, it is impossible to obtain the optimal solution in a MCS with large scale. Without optimal winner selection, truthfulness cannot be guaranteed by VCG-based auction mechanism. Therefore, we propose another mechanism with suboptimal social cost, acceptable computational complexity, and truthfulness.

Algorithm 5: Optimal Winner MUDs Set Selection

input : T, $|T| = M$, (S_1, c_1), ..., (S_i, c_i), ..., $(S_N, c_N$

output: miniCost, Winner set W, $\tau.MUD, \forall \tau \in T$.

```
for each τ in T do
    τ_cost = MAX.VALUE;
for each τ in T do
    for each v_i in V do
        if τ is in S_i then
            currentMini = (τ_0 == τ_i = S_i.first)? c_i : (c_i + τ_{i−1}.cost);
        if currentMini < τ.cost then
            τ.cost = currentMini;
            τ.MUD = v_i;
            for each τ' in S_n do
                τ'.cost = currentMini;
                τ'.MUD = v_i;
for each τ in T do
    Put τ.MUD into W;
miniCost = T.last.cost;
```

The detailed winner set determination algorithm is shown in Algorithm 2. It consists of two steps. First, sort all MUDs in ascending order, according to their average cost $\frac{c_i}{|S_i|}$ (as shown in line (2)). Then, MUDs will be added to winner set W one by one, according to the ascending order derived in the last step, until all sub-tasks in T are covered, as shown in lines (4–13). The payment of each winner v_i is determined based on Algorithm 3. Specifically, it first reorders all MUDs, excluding v_i, in ascending order based on their average cost $\frac{c_j}{|S_j|}$ (as shown in line (2)). Then find the least position j in the order that: v_i may lose in the case of $\frac{c_i}{|S_i|} > \frac{c_j}{|S_j|}$.

Lemma 7. *The winner MUDs set selection provided in Algorithm 2 is monotonic.*

Proof: Suppose v_i is selected as a winner by bidding with c_i and S_i, its average cost is $\alpha_i = \frac{c_i}{|S_i|}$. Let $c'_i \leq c_i$, and S_i remain unchanged, we should prove that v_i would still win when bidding with c'_i and S_i. The new average cost is $\alpha'_i = \frac{c'_i}{|S_i|}$. Since $\alpha'_i \leq \alpha_i$, v_i will be selected earlier by Algorithm 2. It is easy to know that v_i will continue to win with bidding with c_i and S'_i, where $S'_i \supseteq S_i$. For the reason that the new average cost $\alpha'_i = \frac{c_i}{|S'_i|}$ is smaller than $\alpha_i = \frac{c_i}{|S_i|}$.

Lemma 8. *The payment p_i to each $v_i \in W$ is equal to its critical cost c_i^c.*

Algorithm 6: Suboptimal Winner Set Selection

---

input : T, (S_1, c_1), ..., (S_i, c_i), ..., (S_N, c_N)
output: $MiniCost$, W, $\tau.MUD, \forall \tau \in T$

$\forall\, v_i \in V$, sort ascendingly:

$$\frac{c_1}{|S_1|} \leq \frac{c_2}{|S_2|} \leq ... \leq \frac{c_N}{|S_N|},$$

set $U = \emptyset$, j=1;
while $U \neq T$ **do**
  if $U \cap S_j != S_j$ **then**
    put v_j into W, $U = U \bigcup S_j$,
    $miniCost = miniCost + c_j$;
  for *Each* τ *in* S_j **do**
    $\tau.MUD = v_j$;
  j++;

Proof: Assume that critical cost of each MUD is equal to its payment, that is, $\forall\, v_i \in W$, $c_i^c = p_i$. Based on Algorithm 3, $c_i^c = p_i = \frac{c_j}{|S_j|}|S_i|$, where $S_i \subset \bigcup\limits_{i'=1,\ldots,j} S_{i'}$. If v_i bids with $c'_i > c_i^c$, then $\frac{c'_i}{|S_i|} > \frac{c_j}{|S_j|}$, indicating that the average cost of v_i is larger than the average cost of v_j, the position of v_j in this new order is before v_i. In this case, once the v_j is in W, v_i

Algorithm 7: Price Determination

input : T, (S_1, c_1), ..., (S_i, c_i), ..., (S_N, c_N), W;
output: $p_i, \forall v_i \in W$

for *each* v_i *in* W **do**

$\forall\ v_j \in V \backslash v_i$, sort ascendingly:

$$\frac{c_1}{|S_1|} \leq \frac{c_2}{|S_2|} \leq ... \leq \frac{c_{N-1}}{|S_{N-1}|};$$

Find the least index j that $S_i \subset \bigcup\limits_{i'=1,...,j} S_{i'}$;

The payment for MUD v_i is:

$$p_i = \frac{c_j}{|S_j|}\,|S_i|\,;$$

will have no chance to be selected as a winner. On the other hand, if v_i bids with $c_i' \leq c_i^c$, it will still be a winner according to Lemma 7. Thus the assumption is verified.

Theorem 10. *The suboptimal cost mechanism is truthful.*

Proof: With Lemmas 7 and 8, this theorem can be proven based on Theorem 9.

Theorem 11. *The suboptimal cost mechanism is individual rational.*

Proof: For $\forall\ v_i \in V_k$, its payment p_i is equal to its critical cost c_i^c. If v_i wins, there must be $c_i \leq c_i^c$. Hence, $U_i = p_i - c_i \geq 0$. Or v_i loses, its utility is 0. Individual rationality is guaranteed.

Theorem 12. *The suboptimal cost mechanism is computationally efficient.*

Proof: The complexity of Algorithm 2 is $O(n^2)$. The complexity of Algorithm 3 is $O(n^2)$.

4.5 Performance Evaluation

To evaluate the performance of the incentive auction mechanisms proposed in this chapter, experiments are conducted.

4.5.1 Continuous Working Pattern

We set the number of sub-tasks in one round to vary from 50 to 100. The number of MUDs is fixed as 30 ($F = 30$). Each selects a subset of continuous sub-tasks and the size of the subset is randomly chosen from [3, 15]. The cost of each MUD is distributed over [10, 15] uniformly. The VCG-based auction mechanism is designed for the continuous working pattern. Because the continuous case can be regarded as a special case of discontinuous working pattern, the suboptimal auction mechanism can also be used to solve the problem here. Both the VCG-based auction mechanism and suboptimal auction mechanism are evaluated in this section.

In this experiment, the number of sub-tasks changes from 50 to 100, the remaining parameters are set as above. We first compare the performances of the VCG-based auction mechanism and the suboptimal auction mechanism regarding the social cost and running time. The results of the social costs of the two auction mechanisms are shown in Figure 4.4a. The social cost of the VCG-based mechanism is smaller than the suboptimal auction mechanism because the dynamic winner set selection algorithm used in the VCG-based mechanism can find the subset of MUDs which is able to minimize the social cost. However, when considering the running times, as shown in Figure 4.4b, the suboptimal auction mechanism outperforms the VCG-based auction mechanism. We can also observe that with the increase of the number of total sub-tasks, both the cost and running time increase. When the number of sub-tasks increases, the running time of the VCG-based auction mechanism and the suboptimal auction mechanism increase by 700% and 200%, respectively. However, the difference of social costs between two mechanisms keeps steady at the same time. Thus, it is better to use the VCG-based auction mechanism in the continuous working pattern on the condition that a task has fewer sub-tasks.

Then we try to observe the utilities of winner MUDs by the VCG-based auction mechanism. For simplicity, we choose two MUDs randomly, denoted as MUD 2 and MUD 3. We allow the two MUDs to ask different prices and show the truthfulness in Figure 4.5a. It is shown that both the MUD 2 and MUD 3 will reach their maximal utilities when they ask the prices truthfully: $A_2 = c_2 = 10$ and $A_3 = c_3 = 9$. Figure 4.5b presents the utilities of MUD 9 and MUD 17 in the suboptimal auction mechanism. Note that MUD 9 and MUD 17 are chosen randomly too. Similarly, they will also achieve their maximal utilities when acting truthfully: $A_9 = c_9 = 9$ and $A_{17} = c_{17} = 5$.

4.5.2 Discontinuous Working Pattern

The experimental setting of the discontinuous working pattern is similar to the continuous working pattern. Besides, each MUD could select the subset of sub-tasks randomly. Figure 4.6a,b show the results of suboptimal auction mechanism of the social cost and running time when the number of MUDs is 30 and 50 ($F = 25$ and $F = 50$), respectively. Because the more sub-tasks that are contained within a task, the more works need to be done. We can see that when the number of sub-tasks of a task changes from 50 to 100, both the social cost and the running time increase (see Figure 4.6a,b). So, it is a trade-off between objective and efficiency in the suboptimal auction mechanism.

Then we try to verify the truthfulness of the suboptimal auction mechanism in the discontinuous working pattern. MUD 16 and MUD 19 are picked randomly, where $c_{16} = 14$ and $c_{19} = 10$. Let the two MUDs ask different prices from their true costs, their utilities are shown in Figure 4.7. We can see that both MUDs will get their maximal utilities when asking prices truthfully.

4.6 Summary

In this chapter, we investigate the incentive auction mechanisms for mobile crowdsourcing systems. We consider two working patterns in works allocation: the continuous working pattern and the discontinuous working pattern. The objective of the MCS platform is to minimize the social cost in both cases. To achieve the truthfulness, individual rationality, and computational efficiency, we design a VCG-based auction mechanism for the continuous case and a suboptimal auction mechanism for the discontinuous case. We have proven that the two mechanisms can implement the three properties simultaneously. In the future, we plan to design an online incentive mechanism to minimize the social cost and try to maximize the utility of each participated MUD.

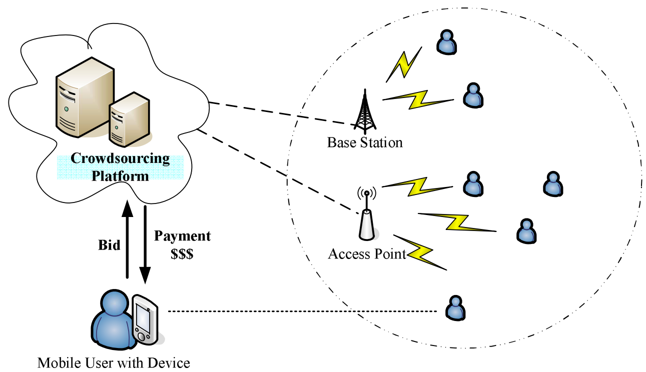


Figure 4.1. An overview of a Mobile Crowdsourcing System (MCS).

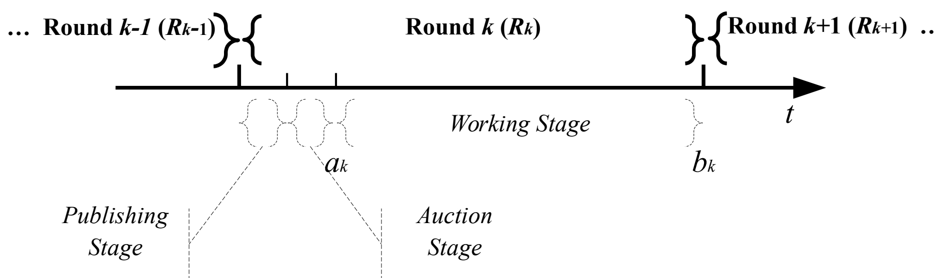


Figure 4.2. The interactive process between the crowdsourcing platform (CP) and mobile users with devices (MUDs).

Table 4.1. Table of Notations.

Notation	Description
CP	Crowdsourcing Platform
MUD	Mobile User Device
V, v_i	set of MUDs and MUD
k	round and task identifier
R_k	description of task k
a_k, d_k	the start time and end time of task k
T_k,	set of sub-tasks in task k
τ_{ki}, τ	sub-task in task k, sub-task
Ω_k, Π_k	hardware parameters and working patterens
b_i	bid of MUD v_i
$\mathbf{b}_{-i}$	bids of all MUDs except v_i
c_i, p_i, U_i	cost, payment and utility of MUD v_i
A_i	asking price of MUD v_i
S_i	subset of sub-tasks MUD v_i can work for
W_k	set of winner MUDs
F	number of MUDs

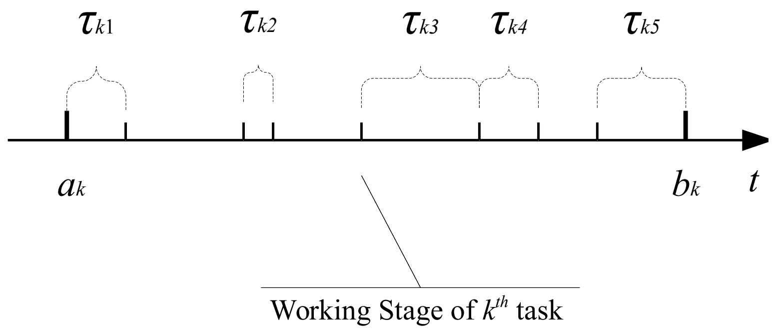
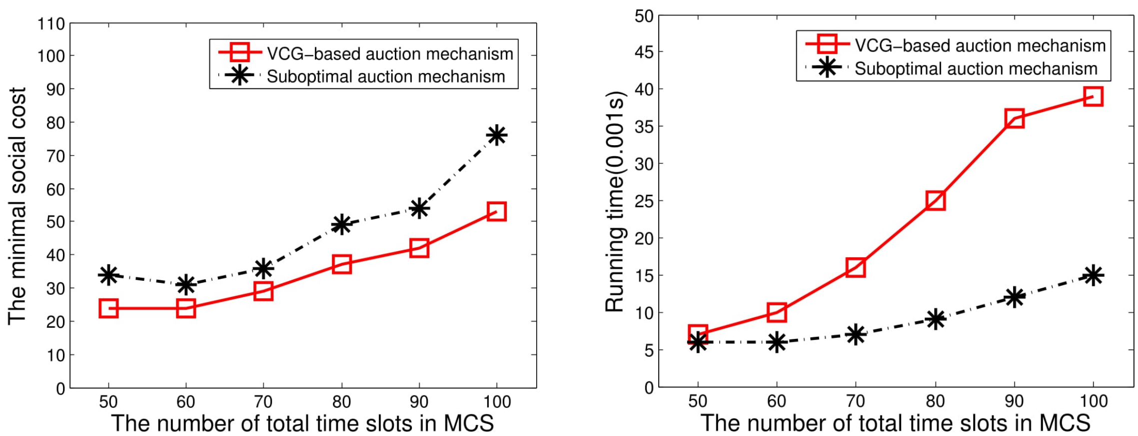
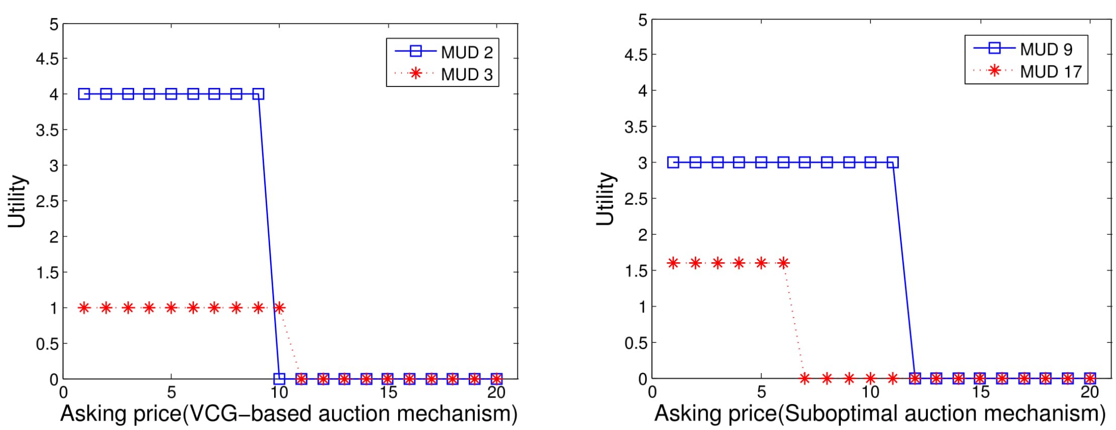


Figure 4.3. An example of sub-tasks in one round.



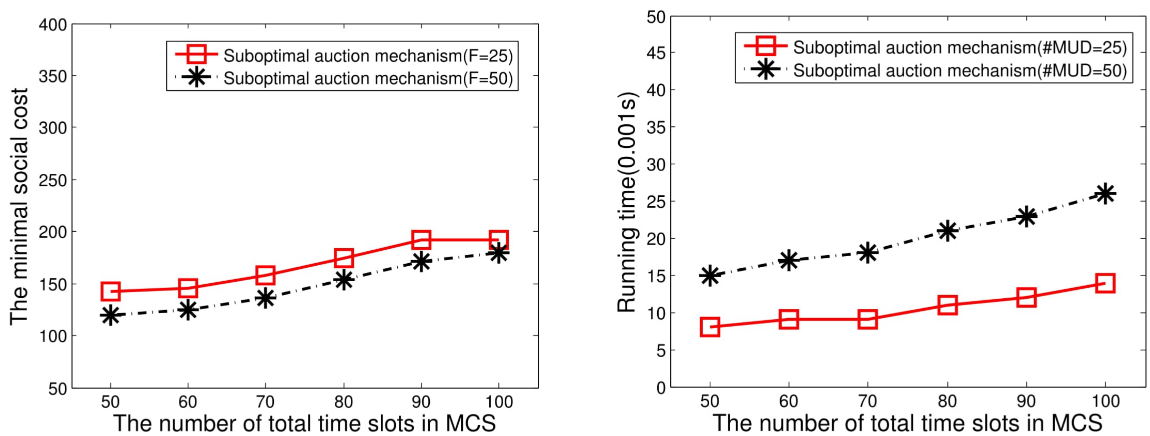
(a) (b)

Figure 4.4. (**a**) The social cost of two auction mechanisms in continuous working pattern; (**b**) The running time of two auction mechanisms in continuous working pattern. VCG: Vickrey–Clarke–Groves.



(a) (b)

Figure 4.5. (**a**) The utility of MUD 2 and MUD 3 by VCG-based auction mechanism in continuous working case; (**b**) The utility of MUD 9 and MUD 17 by suboptimal auction mechanism in continuous working case.



(a) (b)

Figure 4.6. (**a**) The social cost of suboptimal auction mechanism in discontinuous working pattern; (**b**) The running time of suboptimal auction mechanism in discontinuous working pattern.

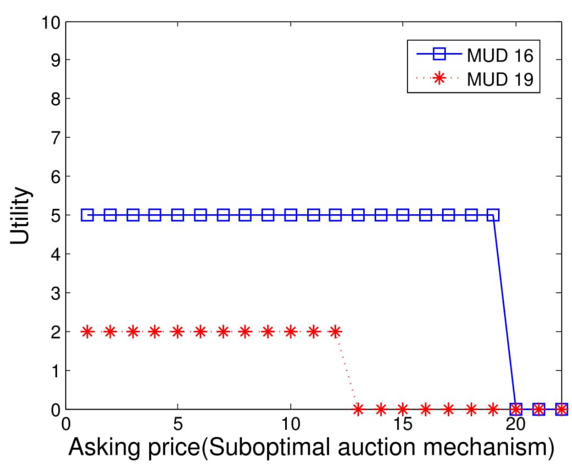


Figure 4.7. The utility of MUD 16 and MUD 19 by suboptimal auction mechanism in discontinuous case.

Chapter 5

PRACTICAL INCENTIVE MECHANISMS FOR IOT-BASED MOBILE CROWDSENSING SYSTEMS

5.1 Introduction

The highly distributed paradigm Internet of Thing (IoT) extends ubiquity of the Internet through integrating every terminal for interaction via embedded systems, in which all the physical terminals can collect and exchange data. IoT will be the fast-growing, largest market potential and the most attractive emerging economy according to the *Top 10 Predictions of 2014* by Gartner. In IoT, the new emerging techniques integrate multiple types of sensors and high-performance processors into physical terminals, *e.g.*, smartphones (iPhones, Sumsung Galaxy, etc.), tablets (iPad, etc.), and vehicle-embedded sensing terminals (GPS). These mobile terminals can be used to sense and collect data, so that become data sources. All above mentioned properties make IoT a perfect choice for the Mobile Crowdsening System (MCS). In an MCS, a complicated sensing job is divided into several simpler tasks. Each participated mobile physical terminal can undertake one or more simpler sensing tasks. The most attractive properties of MCSs is that it aims at letting the regular mobile physical terminals work for the complicated job, while keeping the users of these mobile physical terminals unconscious. In tradition, however, the job must be done by professional experts and the sensors have to be deployed in advance.

The MCSs have already been applied to our daily life. It can be used to collect information around the city and then contributes to the intelligent operation of public services. In detail, it tracks public vehicles and map bumps on the road for the urban transportation systems in a city. The Microblogs provide a mechanism where mobile physical terminals can share their information (like travel, restaurants, and news) through a universal platform. Then, the center server in the platform processes and analyzes the shared data and provides an alternative solution for problems or helps to make decisions. MCSs can also be used in surveillance applications, such as monitoring pollution levels or traffic, measuring water levels, and collecting wildlife habitats. Practical surveillance applications include Common Sense and CreekWatch [62–72].

However, users of these mobile physical terminals participating in an MCS will suffer from extra resource consumption (battery and computing capacities) and the risk of privacy exposure (location exposure). So effective and efficient incentive mechanisms are needed in MCSs to attract enough mobile physical terminals' participation. A common strategy designed in MCSs is to give rewards to participated users as compensation and stimulation. Lots of works can be found on incentive mechanisms and most of them are based on game theory. We classify the existing works into two categories: the offline incentive mechanisms and the online incentive mechanisms. The former will collect the information of all participants before making the decision, while the latter decides whether to accept a new arriving participant sequentially without the information of next following participants. After analyzing these existing works, we find they are not appropriate to surveillance applications for the following reasons: i) the tasks allocation algorithms are unfair over the time dimension. Most of the surveillance applications require continuous sensing information for a period of time. Taking the noise level monitoring application as an example, if a cloud center wants to surveil the noise level of a place, it expects to get noise data of the place for a period of time. Generally speaking, more than one mobile physical terminal will participate in the sensing task. It is better to evenly schedule sensing tasks among a set of mobile physical terminals over the particular period of time. However, mechanisms proposed in existing works [1] may lead to the situation that several mobile physical objects are assigned to sense the noise at the same time incidentally. ii) existing mechanisms require deep interaction between participated users and their mobile physical terminals. That is, sometimes participated users are required to pay lots of attention to their devices or forced to change their own schedule when working for the MCS. For example, the working schedules of users are decided by the MCS. The significant advantage of MCSs over Wireless Senor Networks (WSNs) is that we don't need to deploy the sensors or workers in advance. However, the required deep interaction will interfere participants' original plan which should be avoided. The incentive mechanisms investigated in this work try to overcome the two weaknesses.

We consider an MCS on surveillance applications from both the time and space dimensions. Each sensing task published by the MCS is tagged with a location requirement and a period of time requirement. Once being published, the sensing task is required to be done multiple times over the time period at the specific location. For fairness over the time dimension, a time period is divided into smaller time slots and the sensing task will be processed periodically over these time slots. The participants of the MCS are the mobile vehicles with sensors installed and are able to work for sensing tasks. The device on the vehicle will communicate with cloud servers by 3G or LTE techniques. The drivers of these vehicles are general office workers commuting between home and office. Their routes are relatively stable and they will let the servers know their routes in advance. When a vehicle passes through a location where a sensing task is required, the sensors can work for the sensing task automatically. The objective of the MCS is to select a set of qualified vehicles with devices so that as many tasks can be done evenly over time.

In this paper, we first design an offline incentive mechanism where the *proportional share allocation rule* is applied. Then we consider the realistic situations and propose online auction mechanisms where each winner vehicle will be decided relying on the information of the vehicle itself and the vehicles arriving before it. The contributions of this paper are as follows [73]:

- We first discuss and investigate the importance of unconsciousness in MCSs and get the conclusion that the frequencies of interaction between participants and cloud center should be minimized.

- We introduce the MCS model on surveillance applications. After that, the design of incentive mechanisms under the offline and online cases are designed. The task allocation algorithms are implemented fairly considering practical property requirements.

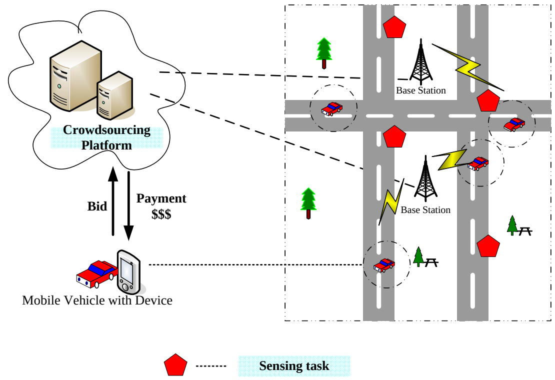


Figure 5.1. An overview of a Mobile Crowdsensing System (MCS).

- In order to improve the performance of the online incentive mechanism, we relax the truthfulness and unconsciousness requirements and propose a (ε, μ)-unconsciousness online incentive mechanism.

In the rest of the paper, we present and discuss previous works in section **??**. Then the MCS system model and problem are formulated in section 5.2. Incentive mechanisms for the offline and online cases are introduced in section 5.3 and section 5.4, respectively. We evaluate the performance of these proposed incentive mechanisms in section 5.5 and conclude the paper in section 5.6.

5.2 System model and problem formulation

5.2.1 Problem Formulation

Considering a Mobile Crowdsensing System (MCS) (as shown in Fig. 5.1) which is able to undertake sensing tasks such like traffic surveillance and environmental pollution monitoring. In the MCS, a Crowdsensing Platform (CP) publishes a set of sensing tasks $\Gamma = \{\tau_1, \tau_2, ..., \tau_m\}$ ($|\Gamma| = m$). Each sensing task $\tau \in \Gamma$ is defined by a collection of features:

$$\tau = (l_\tau, r_\tau),$$

where $l_\tau \in L$ specifies the location of the sensing task and L represents the set of locations which are along routes. r_τ represents the reward that the CP would like to pay if the sensing task τ is done. Each sensing task in Γ is required to be sensed during time T. For simplicity, T is divided into multiple time slots $T = \{t_1, t_2, ..., t_{|T|}\}$. Let $V = \{v_1, v_2, ..., v_n\}$ be the set of Mobile Vehicles with Devices (MVDs). These mobile vehicles move on routes and will pass through one or more locations of sensing tasks. Corresponding sensing tasks will be performed by the devices installed on MVDs automatically. Given an MVD $v \in V$, its features can be denoted as

$$v = \{\rho_v, c_v, \gamma_v\},$$

where ρ_v is the route of v and is defined as discrete location-time points information $\rho_v = \{(l_v^a, t_v^a), (l_v^1, t_v^1), ..., (l_v^i, t_v^i), ..., (l_v^d, t_v^d)\}$. Each element (l_v^i, t_v^i) in ρ_v indicates v will pass through location l_v at time slot t_v^i by estimation. (l_v^a, t_v^a) and (l_v^d, t_v^d) are used to represent v's starting and destination location-time points, respectively. Assume a vehicle would not visit a location more than once in T. An MVD is able to finish any sensing task if the MVD passes through the location of the sensing task. Let c_v be the cost of v if v works for all the sensing tasks located in its route. γ_v is v's driving speed which determines how many time slots are required for v to move between any two different sensing tasks.

All these surveillance sensing tasks (traffic surveillance or environmental pollution monitoring) require to be sensed multiple times in T. However, it is difficult to persuade an MVD to stay at a location without influencing its original routine. Alternatively, a sensing task could be sensed multiple times by different MVDs over different time slots. We call the number of times a sensing task required to be sensed as its space-time coverage requirement. To be fair, for each sensing task, its space-time coverage requirement is distributed over the time slots in T evenly. Matrix $F = [f_{\tau,t}] \in (0,1)^{\Gamma \times T}$ is used to represent the space-time coverage requirements of all sensing tasks over the time dimension. For example, $f_{\tau,t} = 1$ represents that the sensing task τ needs to be sensed once in time slot t. Otherwise, $f_{\tau,t} = 0$. The objective of the CP is to choose winner MVDs, set W that can reach the best coverage requirements over all sensing tasks. The problem can be defined as,

$$\begin{aligned}
Maximize \quad & \sum_{v \in W} \sum_{\tau \in \Gamma} \sum_{t \in T} f_{\tau,t} x_{v,\tau,t} \\
s.t. \quad & x_{v,\tau,t} \in \{0,1\} \\
& \sum_{v \in V} x_{v,\tau,t} \leq f_{\tau,t}, \quad \forall\, \tau \in \Gamma,\ t \in T \\
& W \subseteq V
\end{aligned} \tag{5.1}$$

where W is the winner MVDs. Matrix $x_v = [x_{v,\tau,t}] \in (0,1)^{\Gamma \times T}$ represents the allocated working schedule for v. $x_{v,\tau,t} = 1$ indicates v is allocated to work for sensing task τ in time slot t, otherwise $x_{v,\tau,t} = 0$. The second constraint specifies that a sensing task should be allocated to no more than one MVD in a specific time slot.

5.2.2 Reverse Auction Model Design

Working for sensing tasks brings extra battery consumption, hardware loss and privacy threats to MVDs. Therefore, the winner MVDs expect to receive monetary rewards from the CP as stimulation and compensation. We apply reverse auction model to the interaction between the CP and MVDs, where the CP acts as the buyer and auctioneer at the same time. The roles of MVDs in the model are sellers.

After the CP publishes the sensing tasks, each $v \in V$ submits its bid, which can be denoted as,

$$b_v = \{\hat{\rho}_v,\ A_v\}$$

where $\hat{\rho}_v$ is the set of location-time points that v will pass through. A_v is the asking price when v is selected as a winner to work for these sensing tasks on its route. If the reverse auction mechanism is truthful, $\hat{\rho}_v = \rho_v$ and $A_v = c_v$. That is all MVDs will submit their real route and take the asking price as their base price. Assume, all MVDs are *single-minded* (Definition 1) so that they have simple cost functions.

Definition 1. *A cost function $c(\cdot)$ is called* single-minded *if there exists a sensing tasks' allocation $S^* \subseteq \Gamma$ and a cost c^* such that $c(S) = c^*$ for any $S \subseteq S^* \subseteq \Gamma$ and $c(S) = \infty$ for all other S.*

For each MVD $v \in V$ in our model, $S^* = \{(\tau, t) | (\tau, t) \in \rho_v\}$ is the set of sensing tasks can be done by v and S denotes the sensing tasks allocated to v by the CP when v wins in the auction. Therefore, once a winner v is allocated any set of sensing tasks which v is able to sense, its cost is a consent value. If the allocated sensing task set includes one or more sensing tasks which v can not sense, v will reject the allocation and the cost of v is set as infinity for clarity. Each v sets its bid according to the strategy aiming to maximize its own utility. U_v is used to denote the utility of v and defined as:

$$U_v = \begin{cases} p_v - c_v \quad v_v \ wins, \\ \\ 0 \quad otherwise. \end{cases} \tag{5.2}$$

Generally speaking, the incentive mechanism should satisfy several properties to guarantee its efficiency and effectiveness.

Individual Rationality. Because all MVDs are self-interest to benefit themselves, the utility of any $v \in V$ should be non-negative: $U_v \geq 0$.

Truthfulness. An auction mechanism is called truthful if all MVDs bid with their true value (real cost). The utility of v_j will be maximized when it reports true values in its bid and v_j cannot improve its utility through any misreport:

$$U_{v_j}(b_{v_j}, \mathbf{b}_{v_{-j}}) \geq U_{v_j}(\hat{b}_{v_j}, \mathbf{b}_{v_{-j}}), \tag{5.3}$$

where, $\mathbf{b}_{v_{-j}} = \{b_{v_1}, \ \ldots \ , \ b_{v_{j-1}}, \ b_{v_{j+1}}, \ \ldots \ , \ b_{v_n}\}$ represents the set of truthful bids of all MVDs excluding v_j. b_{v_j} is the truthful bid of v_{v_j}, and $\hat{b}_{v_j} \neq b_{v_j}$. If an auction mechanism satisfies this property, *Nash Equilibrium* exists.The misreports of first value (route) in a bid can be easily detected by the CP through the submitted results of their works. Thus the truthfulness of the first value is guaranteed. We focus on the truthfulness of the second value in a bid: asking price.

Budget Balance. The upper bound of the total payments for all the MVD winners is $B = \sum_{\tau \in \Gamma} r_\tau$, and we call B as the budget constraint of the CP. In other word, the auction mechanism should be budget balance: $B \geq \sum_{v \in W} p_v$.

Unconsciousness. Participation for the MCS are subordinate to MVDs' original target. In detail, the route of each MVD has been scheduled before the CP publishes the sensing tasks. An MVD will not change its route for the reward. On the other hand, when an MVD passes through the location of a sensing task, the sensors installed on the MVD should work automatically without requiring operation from the driver. We call this kind of participation as unconsciousness.

Computational Efficiency. An auction mechanism is considered computationally efficient if the task allocation and payment decision can be implemented in polynomial time.

When the above properties are all satisfied, an auction mechanism can be considered as useful. Without individual rationality, an MVD may receive negative utility, and refuses to participate in the MCS. Then, because the c_v in bid b_v is private to v, the CP wouldn't know it. If an auction mechanism is truthful, all MVDs just need to bid with their true costs: $A_i = c_i$, which not only simplify the strategies, but also avoid possible manipulation from some MVDs. Budget balance make all winner MVDs get their deserved payments. Unconsciousness attracts more MVDs to participate in the MCS. Finally, computational efficiency will guarantee that the auction mechanism can be practically implemented.

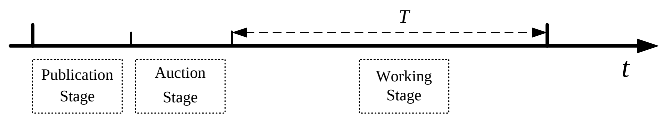


Figure 5.2. Offline interaction process between the CP and MVDs.

5.3 Offline auction mechanism

5.3.1 Offline Working Process of MCSs

In this section, we first focus on the design of offline incentive mechanisms. The working process of an offline MCS can be divided into three stages: publishing stage, auction stage and working stage, as shown in Fig. 5.2.

Publishing stage. In this stage, the CP decides the sensing tasks that it plans to finish within T. Then it publishes the description of these sensing tasks among the MVDs.

Auction stage. After receiving requirements of sensing tasks and their description, each MVD generates location-time points sequence according to its original scheduled route. The sequence of location-time points implies the set of sensing tasks an MVD can take. If an MVD is able to work for a set of sensing task τ, it will further evaluate the cost caused by them. An MVD calculates its cost as the base price and submits a bid to the CP. The bid submitted by an MVD consists of its location-time points sequence and the base price. After receiving bids from all participating MVDs, the CP will choose a set of winner, make the work schedule, determine each winner's reward and then announce the auction result to all participated MVDs.

Working stage. According to the working schedules, each MVD winner will be activated by the CP while passing through a specific location at a specific time. The reward is given to an MVD once it finishes all allocated sensing tasks.

In this work, our focus is the design of efficient and effective incentive mechanisms during the auction stage. The other two stages are omitted.

5.3.2 Modified Proportional Share Auction Mechanism

The design of an offline incentive mechanism for problem (5.1) is more complex than our past work because the consideration of the budget balance property. We rewrite the problem function in (1) as a new form $g(W) = |\underset{v\in W}{\cup} S_v^*|$, where $S_v^* = \{(\tau, t)|(\tau, t \in \rho_v,\ f_{\tau,t} = 1)\}$ and find an interesting point: it is a nondecreasing submodular function.

Definition 2. *A function $h(\cdot)$ is submodular if:*

$$h(\omega \cup \{v\}) - h(\omega) \geq h(X \cup \{v\}) - h(X),$$

where Λ is a finite set, $\omega \subseteq X \subseteq \Lambda$ and $v \in \Lambda \backslash X$, and $h(\cdot):\ 2^V \rightarrow R^+$.

Theorem 13. *The objective function $g(W)$ is a nondecreasing submodular function.*

Proof: For any $W \subseteq X \subseteq V$ and $v \in V\backslash X$, there have $\underset{v'\in W}{\cup} S_{v'}^* \subseteq \underset{v'\in X}{\cup} S_{v'}^*$ and $S_v^* \cap (\underset{v'\in W}{\cup} S_{v'}^*) \subseteq S_v^* \cap (\underset{v'\in X}{\cup} S_{v'}^*)$, so we can get

$$\begin{aligned} g(W \cup v) - g(W) &= |S_v^*| - |S_v^* \cap (\underset{v'\in W}{\cup} S_{v'}^*)| \\ &\geq |S_v^*| - |S_v^* \cap (\underset{v'\in X}{\cup} S_{v'}^*)| \\ &= g(X \cup v) - g(X). \end{aligned}$$

Then,it is easy to obtain a conclusion

$$g(X) - g(W) = |(\underset{v'\in X\backslash W}{\cup} S_{v'}^*) \cap (\underset{v\in W}{\cup} S_v^*)| \geq 0,$$

so $g(W)$ is nondecreasing.

Based on the above analysis, we apply the modified proportional share auction mechanism proposed in [74], which is based on the *proportional share allocation rule.* The auction mechanism has two stages: winner set determination and payment decision.

Algorithm 8: Winner set determination

input : $\{B;\ F;\ (b_v,\ v \in V)\}$
output: $\{W;\ p_v$ and $x_v,\ v \in W\}$

Initialization:
$W = \emptyset$, $v \leftarrow \arg\ \max_{v' \in V}(g_{v'}(W)/A_{v'})$.
$x_{v,\tau,t} = 0,\ \forall v \in V, \tau \in \Gamma, t \in T$;
$y_{\tau,t} = 0,\ \forall \tau \in \Gamma, t \in T$.
while $A_v \leq \frac{g_v(W)B}{g(W \cup v)}$ **do**
    $W = W \cup v$;
    $v \leftarrow \arg\ \max_{v' \in V \setminus W}(g_{v'}(W)/A_{v'})$. **for** *each* $(l^v,\ t^v) \in \rho_v$, *each* $\tau \in \Gamma$ **do**
        if l_τ *is same to* l^v *and* $y_{\tau,t^v} == 0$ **then**
            $x_{v,\tau,t^v} = 1,\ y_{\tau,t^v} = 1$.

The winner set determination process is shown in algorithm 1, where $g_v(W)$ denotes the marginal contribution of v to the coverage requirements, and is calculated as:

$$g_v(W) = g(W \cup v) - g(W).$$

The winner set determination algorithm iteratively selects the MVD who has the largest marginal contribution to the coverage requirement until condition $A_v \leq \frac{g_v(W)B}{g(W \cup v)}$ becomes false.

Once the winner set is identified, payment of each winner v in W will be calculated as follows. Firstly, sort all $v_j \in W \backslash v$ in the non-increasing sorting as,

$$\frac{g_{v_1}^{-v}(O_0)}{A_{v_1}} \geq \frac{g_{v_2}^{-v}(O_1)}{A_{v_2}} \geq \dots \frac{g_{v_j}^{-v}(O_{j-1})}{A_{v_j}} \geq \dots \frac{g_{v_{n-1}}^{-v}(O_{n-2})}{A_{v_{n-1}}}, \tag{5.4}$$

where O_j represents the set of first j MVDs in the sorting result ($O_0 = \emptyset$) and $g_{v_j}^{-v}(O_{j-1})$ is the marginal contribution of v_j when v is removed. Then find the MVD $v' \in W \backslash v$ in the position z of the sorting result which satisfies $A_{v_z} \leq \frac{g_{v_z}^{-v}(O_{z-1})B}{g(O_z)}$. The payment of v will be

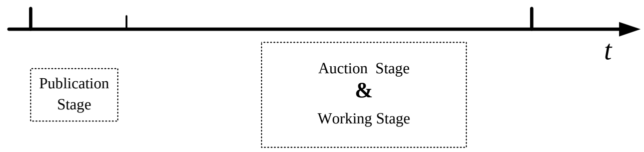


Figure 5.3. Online interaction process between the CP and MVDs.

determined by,

$$p_v = \max_{j\in[1,2,\ldots,z+1]} \{\min\{\frac{g^{-v}_{v(j)}(O_{j-1})A_{v_j}}{g_{v_j}(O_{j-1})}, \frac{g^{-v}_{v(j)}B}{g(O_{j-1}\cup\{v\})}\}\}, \tag{5.5}$$

where $g^{-v}_{v(j)}(O_{j-1}) = g(O_{j-1}\cup\{v\}) - g(O_{j-1})$ represents the marginal contribution of v at position j in the sorting result.

Theorem 14. *The modified proportional share auction mechanism satisfies: individual rationality, truthfulness, Budget Balance, and computational efficiency [74].*

Theorem 15. *Participation in the MCS are unconscious to all MVDs.*

Proof: The working scheduling for each winner MVD is on its predefined route and the MVD will be triggered automatically, so theorem 15 is true.

5.4 Online reverse auction mechanism

5.4.1 Online Working Process of MCSs

In this section, we try to solve the problem formulated in section 5.2 online. Compared with the offline interactions process in MCSs, the online interaction between the CP and MVDs are more flexible. The auction stage and working stage are mixed, as shown in Fig. 5.3. The CP will publish the sensing tasks in advance. Then for any MVD, it can participate in the MCS and submit its bid at anytime within T. Once the CP receives the bid, it will immediately determine whether the MVD wins or not. If the MVD wins, the CP will make the working schedule and determine the payment for this MVD. Then the MVD will work for the sensing tasks according to the received working schedule. After all scheduled sensing tasks are done, the CP will make payment to the MVD.

5.4.2 Simple Online Incentive Mechanism

When design online incentive mechanisms, one precondition should be kept in mind: the CP has no knowledge about the upcoming MVDs and isn't able to make predictions about that. In order to satisfy these property requirements discussed in section 5.2, we first propose a simple online incentive mechanism, which is also based on the *proportional share allocation rule* as shown in algorithm 2. For each new coming v, we first calculates a temporary payment p'_v for v which is proportional to the marginal contribution of v over all coverage requirements (line (1)). If p'_v isn't smaller than the asking price A_v, v will win. Its payment $p_v = p'_v$ and its working schedule will be set (line (5-6)).

Algorithm 9: Simple Online Incentive Mechanism ($Simple - OIM$)

input : $\{B; F; W; b_v; Y\}$
output: $\{W; p_v; x_v; Y; B\}$

1 $p'_v = \frac{g_v(W)}{g(\mathrm{F})} B;\ p_v = 0;$
2 **if** $p'_v \geq A_v$ **then**
3     $W = W \cup v,\ p_v = p'_v;$
4   **for** *each* $(l^v,\ t^v) \in \rho_v,$ *each* $\tau \in \Gamma$ **do**
5   **if** l_τ *is same to* l^v *and* $y_{\tau,t^v} == 0$ **then**
6             $x_{v,\tau,t^v} = 1,\ y_{\tau,t^v} = 1,\ B = B - p_v;$

As shown in theorem 4, the simple auction mechanism presented in algorithm 2 satisfies all the desired auction mechanism properties proposed in section 5.2.

Theorem 16. Simple-OIM *satisfies the desired individual rationality, truthfulness, budget balance, unconsciousness and computational efficiency.*

Proof: Individual rationality: an MVD becomes a winner only under the condition that $p'_v \geq A_v$, then there always has $U_v = p_v - A_v \geq 0$ because $p_v = p'_v$. So individual rationality property is guaranteed;

Truthfulness: an MVD v wins in the case that $p'_v = \frac{g_v(W)}{g(\mathrm{F})}B \geq A_v$. If v bids with $A'_v \leq A_v$, it will still win. Thus we can say the incentive mechanism in algorithm 2 is monotone. On the other hand, if $A_v \leq p'_v$, v will win with payment $p_v = p'_v$. v will lose otherwise. So p'_v can be regarded as the critical value for v. Therefore, according the theorem 5, algorithm 2 is truthfulness;

Budget balance: for each winner $v \in W$, its payment is calculated based on line (1) of algorithm 2 and it is easy to get $\sum\limits_{v \in W} p_v = \sum\limits_{v \in W} \frac{g_v(W)}{g(F)}B \leq B$;

Unconsciousness: the route of each MVD doesn't change due to its participation in the MCS and the mobile physical objects of winner MVDs will be triggered automatically according to their working schedules;

Computational efficiency: the number of location-time points in ρ_v is bounded by m. So the time complexity of algorithm 2 is $O(m * m)$.

Theorem 17. *An incentive mechanism is truthful if and only if it is monotone and the payment for each winner is a critical value [61].*

5.4.3 (ε, μ)-unconsciousness Online Incentive Mechanism

Simple-OIM is simple and able to determine the winner MVD set and make payment decision. In order to further improve the performance, a new online incentive mechanism is proposed which targets at covering more sensing tasks over time with relaxed truthfulness and unconsciousness requirements. Our new online incentive mechanism is motivated by the following two facts. First, most of the incentive mechanisms achieve truthfulness at the expense of effectiveness. Our objective is to get as many sensing tasks covered over time as possible within a limited budget. Based on this concern, the real cost of each MVD is not crucial to the CP. Second, the most sensitive information of a route are the source location, destination location, and the total time duration. Taking a commuter as an example. Most of the time, a commuter will drive from home in the morning. He or she should arrive at office within a specific time duration. With a reasonable reward, the commuter probably accepts to take a new route from home to the office if the commuting time isn't extended strongly. Thus, the idea of the new online incentive mechanism is to recommend another alternative candidate route for losing MVDs based on *Simple-OIM*. The candidate route should maintain the lowest influence on the participating MVDs' unconsciousness. One step further, the candidate route should be as close to the original route of the commuter as possible. In this way, more sensing tasks are expected to be covered by MVDs over time and the utility of the losing MVD can also be increased if it accepts the recommended candidate route. So the decision of candidate route is a trade-off between utility and effectiveness. For simplicity, two new definitions are introduced here.

Definition 3. (ε, μ)-Potential Route ($\rho^{(\varepsilon,\mu)}$). *The $\rho^{(\varepsilon,\mu)}$ of route ρ=$\{(l_0,t_0),\ (l_1,t_1),...,\ (l_{|\rho|},t_{|\rho|})\}$ should satisfy:*

i) ρ and $\rho^{(\varepsilon,\mu)}$ should start at the same location-time points and end at the same destinations.

ii) the similarly degree between ρ and $\rho^{(\varepsilon,\mu)}$ should be larger than ε ($\varepsilon \in [0,1]$). The similarity degree is calculated as,

$$\frac{|\{l = l' | l \in (l,t) \ and \ (l,t) \in \rho, \ l' \in (l',t') \ and \ (l',t') \in \rho^{(\varepsilon,\mu)}\}|}{|\{l | l \in (l,t) \ and \ (l,t) \in \rho\}|}, \tag{5.6}$$

where $l = l'$ means that l and l' are the same location.

iii) the total travel time of $\rho^{(\varepsilon,\mu)}$ is no more than the total travel time of ρ plus a delay tolerance threshold μ ($\mu \geq 0$),

$$\Bigg(\sum_{\substack{n=1,\\(l_n,t_n)\in\rho^{(\varepsilon,\mu)}}}^{|\rho^{(\varepsilon,\mu)}|} (t_n - t_{n-1})\Bigg) - \sum_{\substack{n=1,\\(l_n,t_n)\in\rho}}^{|\rho|} (t_n - t_{n-1}) \leq \mu. \tag{5.7}$$

Definition 4. Candidate Route (CR). $\Delta_\rho^{(\varepsilon,\rho)}$ *represents the set of potential routes for ρ. Candidate route ρ^{CR} in $\Delta_\rho^{(\varepsilon,\rho)}$ is the (ε, μ)-Potential Route of ρ with largest marginal contributions, that is $\rho^{CR} \leftarrow \arg\ \max_{\rho^{(\varepsilon,\mu)} \in \Delta_\rho^{(\varepsilon,\rho)}} (g_{\rho^{(\varepsilon,\mu)}}(W))$.*

Based on the above discussion, we propose an online incentive mechanism: *(ε, μ)-OIM*. For each v, let a symmetric matrix $D_v = [d^v_{l,l'}] \in [Z^+]^{L\times L}$ represent the number of time slots needed by v to travel between any two locations in L. Specifically, values of D_v are based on the speed of v, denoted by γ_v. In the online incentive mechanism, each v submits its bid in a new form:

$$b_v = \{\rho_v,\ A_v,\ D_v\}.$$

The detailed design of *(ε ,μ)-OIM* is shown in algorithm 3. It first applies the *Simple-OIM* for each new coming v (line (1)). If v wins, v's payment and work schedule will be decided by the *Simple-OIM*. Otherwise, a candidate route will be found by tweaked *Depth First Search (DFS)* and recommended to v (line (3)). Here the tweaked DFS is a traversal algorithm which can find all routes between two specific locations. If the candidate route exists, the payment and work schedule of v is decided as shown in line (6-12).

Algorithm 10: (ε, μ)-unconsciousness Online Incentive Mechanism ($(\varepsilon, \mu) - OIM$)

input : $\{B;\ F;\ W;\ b_v;\ Y;\ \varepsilon;\ \mu\}$
output: $\{W;\ p_v;\ x_v;\ Y;\ B\}$

1 $(B,\ W,\ p_v,\ Y,\ x_v) = Simple - OIM(B,\ F,\ W,\ b_v,\ Y)$;
2 **if** $v \notin W$ **then**
3   Adopt the tweaked $Depth\ First\ Search$ to find the candidate route ρ_v^{CR} for v;
4   **if** $\frac{g_{\rho(\varepsilon,\mu)}(W)}{g(F)}B \geq A_v$ **then**
5     $p_v = \frac{g_{\rho(\varepsilon,\mu)}(W)}{g(F)}B$
6     $W = W \cup v$;
7   **for** *each* $(l^v,\ t^v) \in \rho_v^{CR}$*, each* $\tau \in \Gamma$ **do**
8   **if** l_τ *is same to* l^v *and* $y_{\tau,t^v} == 0$ **then**
9         $x_{v,\tau,t^v} = 1,\ y_{\tau,t^v} = 1,\ B = B - p_v$;

(ε, μ)-OIM gives MVDs who lose in the *Simple-OIM* another opportunity to win. This leads to more MVDs working for the CP and more tasks can be done.

Theorem 18. (ε, μ)-OIM *satisfies the individual rationality, budget balance, and* (ε, μ)-*unconsciousness property requirements.*

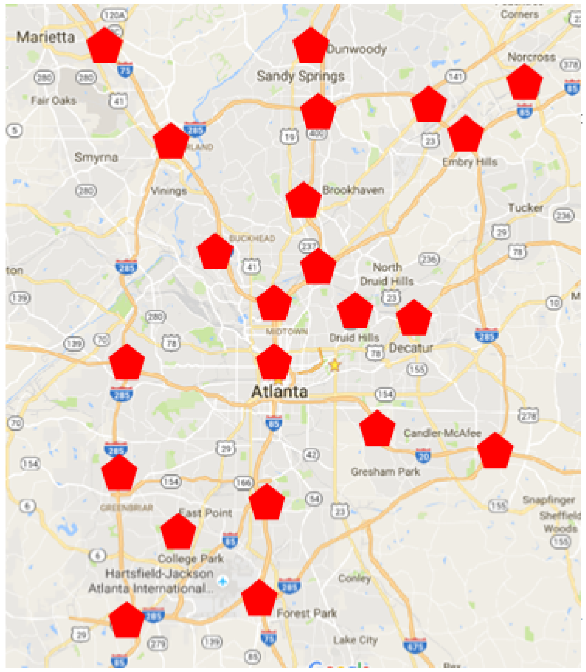


Figure 5.4. The locations distribution of tasks over Atlanta metropolitan area ($30km$*$40km$).

Proof: The proof is similar to *theorem 4*.

5.5 Performance evaluation

5.5.1 Evaluation of the offline case

The sensing region is $30km$*$40km$ and located in the Atlanta metropolitan area. We mark 22 popular locations within the region in Google map as sensing task locations (shown in Fig. 5.4). Then the budget B varies from 2000 to 14000. The number of MVDs varies from 50 to 250. For each MVD, its speed and cost are randomly generated from [$25km/h$, $60km/h$] and [10, 30], respectively. The total time ($T = 2.5h$) is divided into 150 time slots. The route of each MVD is a sequence of locations on the map and the time at when the MVD will pass through them is obtained based on its speed. One step further, the starting time of the route is distributed over T.

The total number of tasks covered by winner MVDs in the offline incentive mechanism is shown in Fig. 5.5. We observe that more tasks can be done with the increase of either

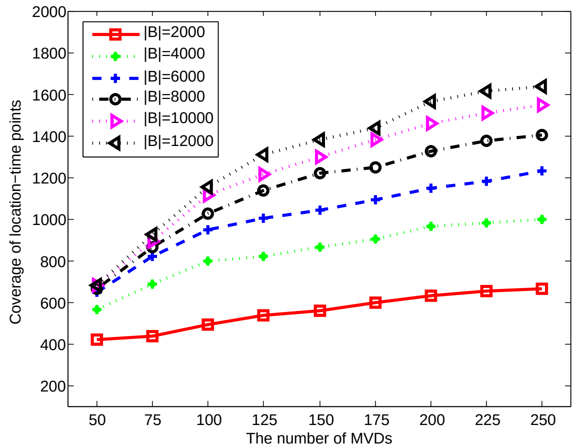
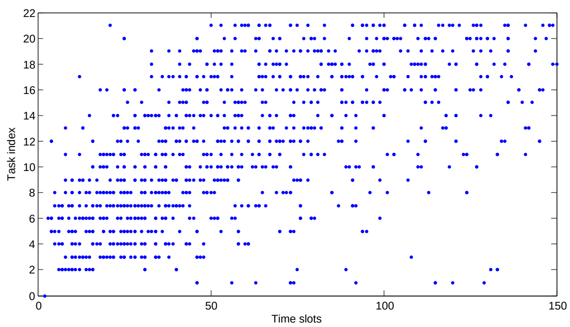
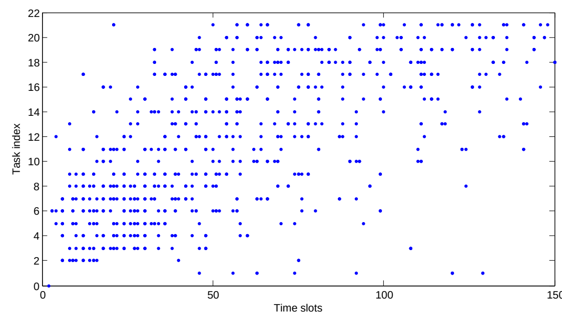


Figure 5.5. The total number of tasks covered by the offline incentive mechanism.



(a)



(b)

Figure 5.6. (a)The location-time points coverage by offline incentive mechanism. (b)The location-time points coverage.

the number of MVDs or the budget, respectively. Secondly, we compare the coverage of sensing tasks in our proposed offline incentive mechanism with the mechanism which hasn't considered the time dimension. The results are shown in Fig. 5.6(a) and 5.6(b), separately. We can see the covered location-time points in Fig. 5.6(a) is denser than that in Fig. 5.6(b) by about 20 percent. The result shows that the offline incentive mechanism in our paper gets more sensing tasks done over the space and time dimensions. The reason for the sparse coverage in Fig. 5.6(b) is the overlapping: a sensing task may be covered by more than one MVD at a specific time slot.

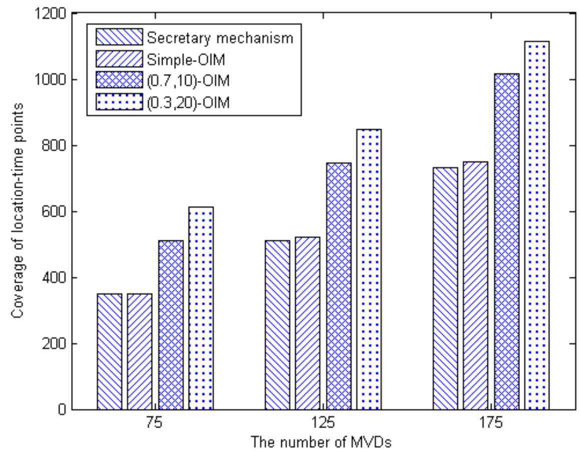


Figure 5.7. The coverage of online incentive mechanisms

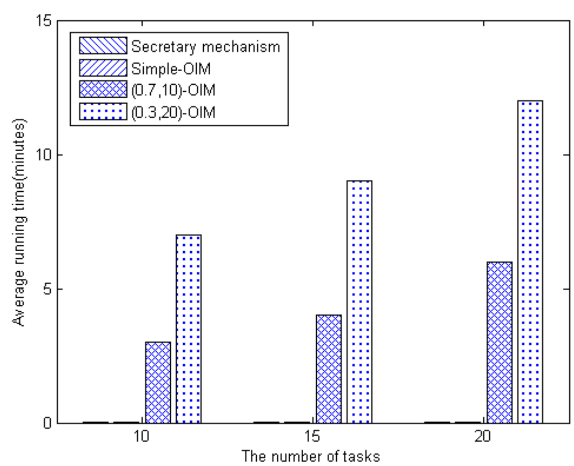


Figure 5.8. The average running time of online incentive mechanisms.

5.5.2 Evaluation of the online case

To evaluate the performance of online incentive mechanisms designed in this paper, we take the *secretary mechanism* as a benchmark which is based on the classical *secretary algorithm.* That can be summarized as:

Secretary mechanism. Let the first k arrived MVDs as set K, reject the MVDs in K, and calculate $\delta = \max\limits_{v \in K}\{\frac{g_v(K)}{c_v}\}$ as threshold. Then for each new coming MVD denoted as v' which satisfies $\frac{g_{v'}(w)}{c_{v'}} \geq \delta$, calculate its temporary payment $p'_{v'} = \frac{g_{v'}(w)}{\delta}$. v' will be selected as a winner ($W = W \cup v'$) with payment $p_{v'} = p'_{v'}$ if $p_{v'} + \sum\limits_{v \in W} p_v \leq B$.

The experimental setup of the online case is similar to the offline case. The MVDs are required to submit bids at their starting time. We first compare the number of sensing tasks covered by winner MVDs over time obtained from the two online incentive mechanisms proposed in this paper with the secretary mechanism. The result is shown in Fig. 5.7. We can observe that for each online incentive mechanism, its coverage increases with the increase of participating MVDs. Then, the results of the secretary mechanism and the *Simple-OIM* are almost same. However, first k MVDs is rejected in secretary mechanism which can not guarantee sovereignty because these MVDs in K are excluded arbitrarily. The MCS should make each MVD have the same opportunity to win. From this aspect, *Simple-OIM* is better than the secretary mechanism. *(ε, μ)-OIM* outperforms the other two mechanisms because it gives the losing MVDs one more chance to win. We know that ε and μ are used to constrain the potential routes. Strict constraints (larger of ε and smaller of μ) will limit number of potential routes and decrease running time. But loose constraints will lead to more tasks covered over time. Thus, *(0.3, 20)-OIM* performs better than *(0.7, 10)-OIM.*

Then we test the performance of the four online incentive mechanisms under different number of sensing tasks. The results of average running time for each MVD under different setups are shown in Fig. 5.8. The average running time of each MVD in secretary mechanism and the *Simple-OIM* turns out to be negligible with the increase of the number of sensing tasks. Because the tweaked DFS is adopted to calculate the potential routes for each losing MVD, the time complexity is high in *(ε, μ)-OIM.* So *(0.3, 20)-OIM* needs more time than

(0.7, 10)-OIM. In order to guarantee the computational efficiency, the *(ε, μ)-OIM* should choose larger ε and smaller μ even though the part of effectiveness will be sacrificed.

5.6 Summary

In this paper, we focus on incentive mechanisms design in IoT-based MCSs for surveillance applications. We investigate practical requirements and the importance of fairness and unconsciousness in winner MVDs selection. Two kinds of incentive mechanisms are proposed which can be applied in realistic applications. Extensive simulations are conducted to validate the performance of them.

Chapter 6

FUTURE RESEARCH DIRECTIONS

6.1 Future Research Directions

In this section, we are discussing more future problems, challenges, and research directions.

Internet of Things (IoT) grows explosively, in which a number of devices are connected. The connection magnifies individual data privacy threats, exposing the personal information of millions devices.

The data collected in IoT can be published, enabling researchers and goverments to analyze the data and learn important information which can benefit the society as a result. Examples inculde inducement of a certain desease, effectiveness public policy formation, guidance of drug research and development. That is to say, society can gain utility through the published data from IoT. On the other hand, these data from IoT contains specific information of devices or users of device. In other words, publishing data from Iot would bring privacy loss for users whose devices are connected into the IoT. Consequently, the privacy-preserving data publishing in IoT becomes a foundamental problem focusing on how to make the proper trade-off between privacy and utility.

Game theory has been widely used to design data privacy to analyze users behaviors in IoT since game theory can model situations of conflict and predict the behavior of users.

CONCLUSION

This dissertation conducts the problem of incentive mechanism design in mobile crowdsensing system with consideration of economic properties. We study different models and algorithms to provide explorations and resolutions.

To motivate mobile users to join sensing tasks in MCSs, we propose a reverse auction model and two novel distributed auction schemes, CPAS and TPAS, for task assignment and scheduling. Specifically speaking, the novelty of the proposed auction model and auction schemes lies in the following aspects: i) the auction model is practical taking into account partial fulfillment, attribute diversity, and price diversity; ii) the two auction schemes can be implemented within a well-designed distributed auction framework; iii) both two auction schemes are proved to be computationally efficient, individually rational, budget balanced, and truthful.

Second, we investigate the incentive auction mechanisms for mobile crowdsourcing systems. We consider two working patterns in works allocation: the continuous working pattern and the discontinuous working pattern. The objective of the MCS platform is to minimize the social cost in both cases. To achieve the truthfulness, individual rationality, and computational efficiency, we design a VCG-based auction mechanism for the continuous case and a suboptimal auction mechanism for the discontinuous case. We have proven that the two mechanisms can implement the three properties simultaneously.

Thirdly, we focus on incentive mechanisms design in IoT-based MCSs for surveillance applications. We investigate practical requirements and the importance of fairness and unconsciousness in winner MVDs selection. Two kinds of incentive mechanisms are proposed which can be applied in realistic applications. Extensive simulations are conducted to validate the performance of them.

Besides, we also propose several very important and challenging potential further work directions.

REFERENCES

[1] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff, "Common sense: participatory urban sensing using a network of handheld air quality monitors," in *Proceedings of the 7th ACM conference on embedded networked sensor systems*. ACM, 2009, pp. 349–350.

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

[3] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 1, p. 6, 2009.

[4] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *ACM MobiHoc*, Hangzhou, China, June 2015, pp. 167–176.

[5] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen, "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4203–4214, 2015.

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

[7] Z. Zheng, Y. Peng, F. Wu, S. Tang, and G. Chen, "Trading data in the crowd: Profit-driven data acquisition for mobile crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 486–501, 2017.

[8] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *IEEE ICDCS*, Atlanta, Georgia, USA, June 2017.

[9] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *IEEE INFOCOM*, Toronto, Canada, April 2014, pp. 1231–1239.

[10] J. Xu, J. Xiang, and D. Yang, "Incentive mechanisms for time window dependent tasks in mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6353–6364, 2015.

[11] J. Li, Y. Zhu, Y. Hua, and J. Yu, "Crowdsourcing sensing to smartphones: A randomized auction approach," *IEEE Transactions on Mobile Computing*, no. 99, p. 1, 2017.

[12] R. Ben Messaoud, Y. Ghamri-Doudane, and D. Botvich, "Preference and mobility-aware task assignment in participatory sensing," in *ACM MSWiM*, Malta, Malta, November 2016, pp. 93–101.

[13] J. Xu, H. Li, Y. Li, D. Yang, and T. Li, "Incentivizing the biased requesters: Truthful task assignment mechanisms in crowdsourcing," in *IEEE SECON*, San Diego, CA, USA, June 2017, pp. 1–9.

[14] Z. Song, C. H. Liu, J. Wu, J. Ma, and W. Wang, "Qoi-aware multitask-oriented dynamic participant selection with budget constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4618–4632, 2014.

[15] A. Biswas, S. Jain, D. Mandal, and Y. Narahari, "A truthful budget feasible multi-armed bandit mechanism for crowdsourcing time critical tasks," in *AAMAS*, Istanbul, May 2015.

[16] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *ACM Mobicom*, Istanbul, Turkey, August 2012, pp. 173–184.

[17] Y. Zhu, Q. Zhang, H. Zhu, J. Yu, J. Cao, and L. M. Ni, "Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones," in *IEEE ICDCS*, Madrid, Spain, July 2014, pp. 11–20.

[18] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Truthful incentive mechanisms for crowdsourcing," in *IEEE INFOCOM*, HONG KONG, China, April 2015, pp. 2830–2838.

[19] S. Ji and T. Chen, "Incentive mechanisms for discretized mobile crowdsensings," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 146–161, Janary 2016.

[20] M. H. Cheung, R. Southwell, F. Hou, and J. Huang, "Distributed time-sensitive task selection in mobile crowdsensing," in *ACM MobiHoc*, Hangzhou, China, June 2015, pp. 157–166.

[21] H. Shah-Mansouri and V. W. Wong, "Profit maximization in mobile crowdsourcing: A truthful auction mechanism," in *IEEE ICC*, London, UK, June 2015, pp. 3216–3221.

[22] "GigWalk," http://www.gigwalk.com/, 2015.

[23] "Field Agent," https://fieldagent.net/, 2016.

[24] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, "Image browsing, processing, and clustering for participatory sensing: lessons from a dietsense prototype," in *Proceedings of the 4th workshop on Embedded networked sensors*. ACM, 2007, pp. 13–17.

[25] 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 *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 1366–1374.

[26] C.-J. Ho and J. W. Vaughan, "Online task assignment in crowdsourcing markets." in *AAAI*, vol. 12, 2012, pp. 45–51.

[27] C.-J. Ho, S. Jabbari, and J. W. Vaughan, "Adaptive task assignment for crowdsourced classification," in *International Conference on Machine Learning*, 2013, pp. 534–542.

[28] V. Nunia, B. Kakadiya, C. Hota, and M. Rajarajan, "Adaptive task scheduling in service oriented crowd using slurm," in *International Conference on Distributed Computing and Internet Technology*. Springer, 2013, pp. 373–385.

[29] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *ACM MobiCom*, Istanbul, Turkey, August 2012, pp. 173–184.

[30] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 1701–1709.

[31] J.-S. Lee and B. Hoh, "Sell your experiences: a market mechanism based incentive for participatory sensing," in *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*. IEEE, 2010, pp. 60–68.

[32] A. Singla and A. Krause, "Truthful incentives in crowdsourcing tasks using regret minimization mechanisms," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 1167–1178.

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

[34] J. Li, Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networksto explore influence maximization," in *IEEE INFOCOM*, San Francisco, USA, April 2016, pp. 1–9.

[35] V. Krishna, *Auction Theory*, 2nd ed. Elsevier, 2010.

[36] Q. Hu, S. Wang, L. Ma, R. Bie, and X. Cheng, "Anti-malicious crowdsourcing using the zero-determinant strategy," in *IEEE ICDCS*, Atlanta, USA, June 2017.

[37] W. Li, J. Yu, X. Cheng, R. Bie, and F. Zhao, "An extensible and flexible truthful auction framework for heterogeneous spectrum markets," *to appear in IEEE Transactions on Cognitive Communications and Networking*, vol. 2, no. 4, pp. 427–441, 2016.

[38] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Computer Networks*, vol. 102, pp. 157–171, 2016.

[39] W. Li, S. Wang, X. Cheng, and R. Bie, "Truthful multi-attribute auction with discriminatory pricing in cognitive radio networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 18, no. 1, pp. 3–13, January 2014.

[40] W. Li, S. Wang, and X. Cheng, "Truthful multi-attribute auction with discriminatory pricing in cognitive radio networks," in *ACM MobiCom Workshop on CRAB*, Miami, USA, October 2013, pp. 21–30.

[41] Y. Zhang and M. van der Schaar, "Reputation-based incentive protocols in crowdsourcing applications," in *IEEE INFOCOM*, Orlando, Florida, USA, March 2012, pp. 2140–2148.

[42] H. Xie and J. Lui, "Modeling crowdsourcing systems: Design and analysis of incentive mechanism and rating system," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 2, pp. 52–54, September 2014.

[43] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices." *IEEE Network Magazine*, 2018.

[44] J. Li, S. Cheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Approximate holistic aggregation in wireless sensor networks." *ACM Transactions on Sensor Networks*, vol. 13, no. 2, pp. 1–11, 2017.

[45] X. Zheng, Z. Cai, J. Yu, C. Wang, and Y. Li, "Follow but no track: Privacy preserved profile publishing in cyber-physical social systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1868–1878, 2017.

[46] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart iot systems: A consideration from privacy perspective," *IEEE Communications Magazine*, 2018.

[47] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems," *Computer Networks*, vol. 135, pp. 32–43, 2018.

[48] J. Wang, Z. Cai, Y. Li, D. Yang, J. Li, and H. Gao, "Protecting query privacy with differentially private k-anonymity in location-based services," *Personal and Ubiquitous Computing*, pp. 1–17, 2018.

[49] J. Lu, Z. Cai, X. Wang, L. Zhang, P. Li, and Z. He, "User social activity-based routing for cognitive radio networks," *Personal and Ubiquitous Computing*, pp. 1–17, 2018.

[50] J. Li, Z. Cai, J. Wang, M. Han, and Y. Li, "Truthful incentive mechanisms for geographical position conflicting mobile crowdsensing systems," *IEEE Transactions on Computational Social Systems*, 2018.

[51] X. Zheng, G. Luo, and Z. Cai, "A fair mechanism for private data publication in online social networks," *IEEE Transactions on Network Science and Engineering*, 2018.

[52] Q. Chen, H. Gao, S. Cheng, X. Fang, Z. Cai, and J. Li, "Centralized and distributed delay-bounded scheduling algorithms for multicast in duty-cycled wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–14, 2017.

[53] Z. Duan, W. Li, and Z. Cai, "Distributed auctions for task assignment and scheduling in mobile crowdsensing systems," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 635–644.

[54] K. Aspelin, "Establishing pedestrian walking speeds," *Portland State University*, pp. 5–25, 2005.

[55] "Mobile Fact Sheet," http://www.pewinternet.org/fact-sheet/mobile/, 2015.

[56] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, vol. 16, no. 5, pp. 36–44, 2012.

[57] S. Kim, C. Robson, T. Zimmerman, J. Pierce, and E. M. Haber, "Creek watch: pairing usefulness and usability for successful citizen science," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 2125–2134.

[58] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "Cartel: a distributed mobile sensor computing system," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 125–138.

[59] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "Parknet: drive-by sensing of road-side parking statistics," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 123–136.

[60] Z. Duan, M. Yan, Z. Cai, X. Wang, M. Han, and Y. Li, "Truthful incentive mechanisms for social cost minimization in mobile crowdsourcing systems," *Sensors*, vol. 16, no. 4, p. 481, 2016.

[61] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*, 2007, vol. 1.

[62] Z. Cai, G. Lin, and G. Xue, "Improved approximation algorithms for the capacitated multicast routing problem," in *In Proceedings of the 11th International Computing and Combinatorics Conference (COCOON 2005)*, vol. 8881. Springer, 2005, pp. 136–145.

[63] G. Lin, Z. Cai, and D. Lin, "Vertex covering by paths on trees with its applications in machine translation," *Information processing letters*, vol. 97, no. 2, pp. 73–81, 2006.

[64] Z. Cai, Z.-Z. Chen, G. Lin, and L. Wang, "An improved approximation algorithm for the capacitated multicast tree routing problem," *The 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA2008)*, vol. 5165, pp. 286–295, 2008.

[65] C. Vu, Z. Cai, and Y. Li, "Distributed energy-efficient algorithms for coverage problem in adjustable sensing ranges wireless sensor networks," *Discrete Mathematics, Algorithms and Applications*, vol. 1, no. 03, pp. 299–317, 2009.

[66] Z. Cai, Z.-Z. Chen, and G. Lin, "A 3.4713-approximation algorithm for the capacitated multicast tree routing problem," *Theoretical Computer Science*, vol. 410, no. 52, pp. 5415–5424, 2009.

[67] C. Ai, L. Guo, Z. Cai, and Y. Li, "Processing area queries in wireless sensor networks," in *The Fifth International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2009)*. IEEE, 2009, pp. 1–8.

[68] L. Guo, C. Ai, X. Wang, Z. Cai, and Y. Li, "Real time clustering of sensory data in wireless sensor networks." in *The 28th IEEE International Performance Computing and Communications Conference (IPCCC 2009)*, 2009, pp. 33–40.

[69] Z. Cai, R. Goebel, and G. Lin, "Size-constrained tree partitioning: A story on approximation algorithm design for the multicast k-tree routing problem." in *The 3nd Annual International Conference on Combinatorial Optimization and Applications (COCOA2009)*. Springer, 2009, pp. 363–374.

[70] Y. Li, C. Ai, Z. Cai, and R. Beyah, "Sensor scheduling for p-percent coverage in wireless sensor networks," *Cluster Computing*, vol. 14, no. 1, pp. 27–40, 2011.

[71] Z. Cai, R. Goebel, and G. Lin, "Size-constrained tree partitioning: approximating the multicast k-tree routing problem," *Theoretical Computer Science*, vol. 412, no. 3, pp. 240–245, 2011.

[72] Y. Huang, X. Guan, Z. Cai, and T. Ohtsuki, "Multicast capacity analysis for social-proximity urban bus-assisted vanets," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6138–6142.

[73] Z. Duan, L. Tian, M. Yan, Z. Cai, Q. Han, and G. Yin, "Practical incentive mechanisms for iot-based mobile crowdsensing systems," *IEEE Access*, vol. 5, pp. 20 383–20 392, 2017.

[74] Y. Singer, "Budget feasible mechanisms," in *IEEE FOCS*, Las Vegas, Nevada, USA, October 2010.