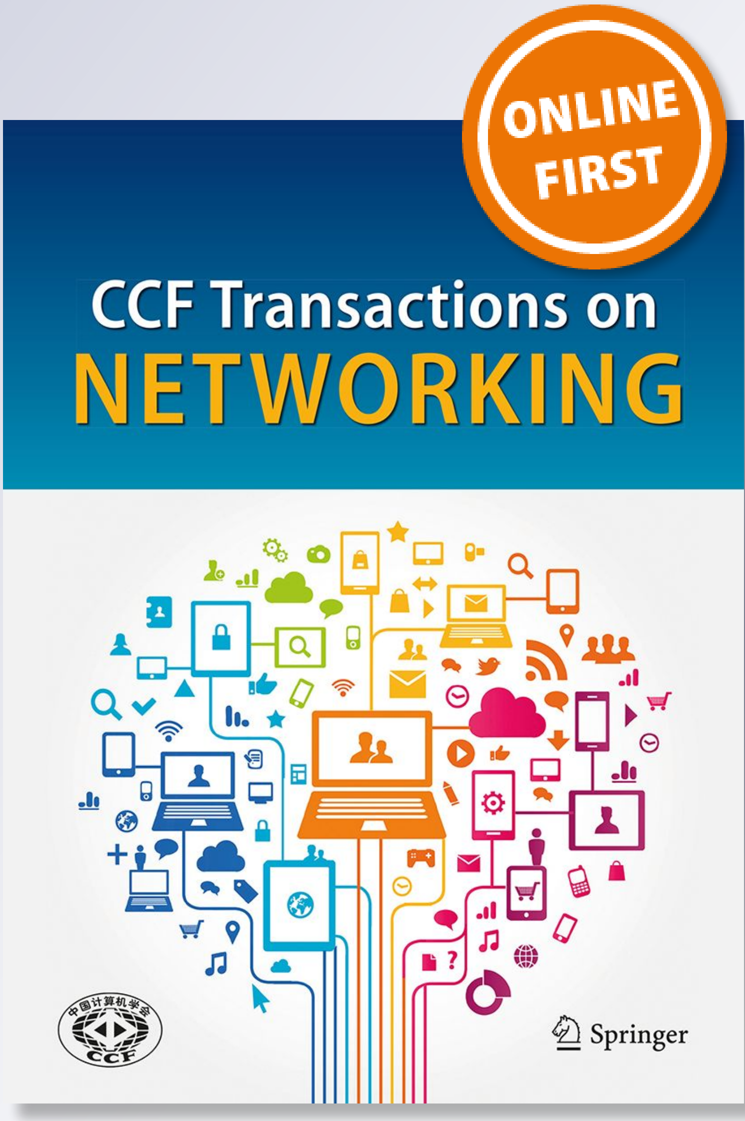
Online task dispatching and pricing for quality-of-service-aware sensing data collection for mobile edge clouds

Tong Liu, Yanmin Zhu, Yuanyuan Yang, Fan Ye & Jiadi Yu

CCF Transactions on Networking

ISSN 2520-8462

CCF Trans. Netw.
DOI 10.1007/s42045-018-0008-8



Springer

CCF Transactions on Networking
https://doi.org/10.1007/s42045-018-0008-8

REGULAR PAPER

Online task dispatching and pricing for quality-of-service-aware sensing data collection for mobile edge clouds

Tong Liu[1] · Yanmin Zhu[2] · Yuanyuan Yang[3] · Fan Ye[3] · Jiadi Yu[2]

Received: 29 July 2018 / Accepted: 8 November 2018

Abstract

The proliferation of mobile devices equipped with rich sensing and computing resources has pushed the emergence of a new cloud paradigm, mobile edge clouds, where tasks are dispatched from the centralized cloud to the network edge. By taking the advantage of widely-distributed mobile devices, urban monitoring-oriented crowdsourcing services can be provided by a mobile edge cloud, where fine-grained monitoring data over time are crowdsourced by mobile devices and then useful information is extracted. However, as considerable costs are incurred on mobile devices, there exists a major problem that a high financial budget is required to guarantee the quality of service. Fortunately, we observe that real-world sensing data exhibit strong spatial and temporal correlations, and advanced inference methods can be employed to efficiently recover missing data. Motivated by the observation, we provide a near-optimal online task dispatching approach for crowdsourcing services provided by a mobile edge cloud, aiming to minimize the total cost incurred on devices while guarantee the quality of service in the meantime. Besides, considering strategic device users with private cost information, we also propose a truthful pricing policy. Extensive simulations based on real datasets show that our approach outperforms other competing schemes, producing a high quality of service with a much lower budget.

Keywords Edge computing · Task dispatching · Quality of service and pricing

1 Introduction

Recent years have witnessed the wide penetration of smartphones in our daily life. It is reported that the number of smartphone users in the world is as high as 2.1 billions (World-wide smartphone users 2018). Thanks to the proliferation of smartphones equipped with various sensors and computing resources, *mobile edge clouds* (also known as edge computing and fog computing) (Shi et al. 2016; Luan et al. 2016) has become a promising paradigm for collecting and processing sensing data, where smartphones are seen as small servers at the network edge. Taking the advantage of widely-distributed smarthphones, urban monitoring services can be provided by a mobile edge cloud, in which fine-grained sensing data over time can be crowdsourced by smartphone users, such as traffic event reporting (Shi et al. 2016), noise mapping (Rana et al. 2010) and air pollution monitoring (Mendez et al. 2011).

In this work, a general mobile edge cloud is considered, comprised of base stations and mobile smartphones with cloud-like computing capacity and storage, as shown in Fig. 1. When a crowdsourcing service request arrives to the

✉ Yanmin Zhu
yzhu@sjtu.edu.cn

Tong Liu
tong_liu@shu.edu.cn

Yuanyuan Yang
yuanyuan.yang@stonybrook.edu

Fan Ye
fan.ye@stonybrook.edu

Jiadi Yu
jiadiyu@sjtu.edu.cn

1 School of Computer Engineering and Science, Shanghai University, 99 Shangda Road, Shanghai 200444, China

2 Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, China

3 Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA

Published online: 30 November 2018

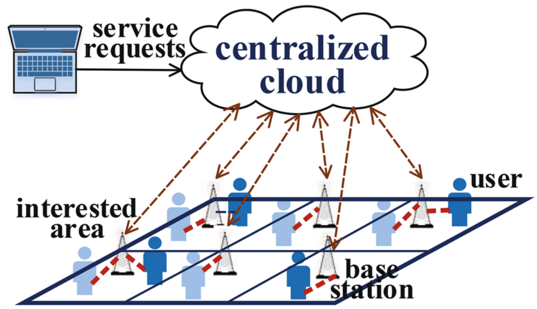


Fig. 1 An illustration of a mobile edge cloud system, where base stations and mobile smartphones are seen as small servers at the network edge. For an urban monitoring service, fine-grained sensing data need to be collected and processed by these small servers

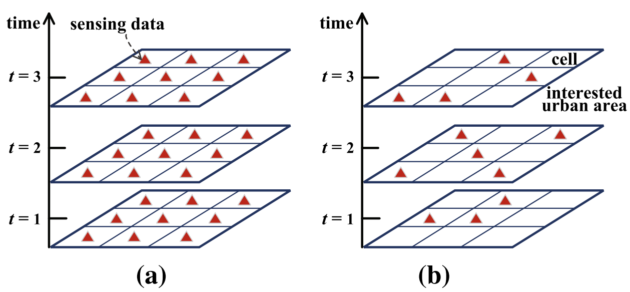


Fig. 2 **a** The straightforward approach to satisfy an urban monitoring service by collecting the complete sensing data. It incurs a high budget to the service requester. **b** The proposed approach leverages the spatial–temporal correlations and collects a subset of sensing data, which significantly reduces the budget

system, the centralized cloud first divides it into individual sensing and computing tasks, and then dispatches them to mobile smartphones through base stations. Such services need to collect *spatial–temporal sensing data* in order to monitor a given urban area over time. For simplicity, the urban area is divided into grids of the same size and the time slotted. The straightforward method is to collect the complete sensing data of all grids in all time slots, as illustrated in Fig. 2a. However, many smartphones need be employed, which incurs a high financial budget to the service requester. As we know, doing sensing and computing tasks incurs a certain cost on each resource-limited smartphone, such as battery consumption and bandwidth usage. Thus, monetary incentives should be provided to smartphone users for performing crowdsourcing tasks.

Fortunately, we observe that in a real-world urban monitoring application, the sensing data exhibits strong spatial and temporal correlations. For example, we analyzed *PM2.5 Air Quality* (Zheng et al. 2013) data of Beijing, China, by spliting the monitored area into 1 km × 1 km grids and time into 1 h time slots. We found that the spatial correlation between two adjacent grids is as high as 0.9518, and the temporal correlation between two contiguous time slots is as high as 0.9311. The existence of strong spatial and temporal correlations suggests that it is unnecessary to collect all sensing data of all grids in all time slots. Instead, only a subset of sensing data is collected (as shown in Fig. 2b) and data recovery methods can be exploited, such as matrix factorization (Dhillon and Sra 2006), to infer the rest missing data.

Motivated by the observation, we propose a new sensing data collection and pricing scheme for a mobile edge cloud. The main idea is collecting only a small subset of sensing data, and then applying data recovery methods via leveraging the spatial–temporal correlations. Thus, the total budget is dramatically reduced since the total amount of collected sensing data is much smaller. To implement this idea, however, there are two coupled problems to address. The *first problem* is to determine the cells and the corresponding smartphone users to do the sensing tasks, with the objective to lower the total cost while retain a high quality of service. The *second problem* is to determine the pricing policy provided to strategic smartphone users. The payment should be made in such a way that all strategic users would truthfully report their sensing and computing costs of doing tasks. We stress that the two problems are *coupled*, since the amount of payment to each recruited user is typically dependent on the set of specific users who are recruited.

Solving the two coupled problems faces several major challenges. *First*, mobile samrtphones are dynamic who may join and leave at any time. And their costs may change over time. This suggests that greedily selecting users with the lowest costs may not be optimal, since better users may become available in future. *Second*, the relationship between the collected sensing data and the accuracy of recovered data, i.e., the quality of service, is not clear. A formalized model is required to characterize the quality of service based on collected sensing data. *Finally*, smartphone users are typically strategic and their cost information is private. A strategic user may misreport his or her cost information to the platform in order to gain a higher payment.

A few existing works (Xu et al. 2015a, b; Wang et al. 2015) have noticed the correlations in sensing data and proposed methods to lower the cost of data collection. However, they simply assume smartphone users are voluntary crowdsourcing workers. Recently, some task dispatching methods (Liu et al. 2016; Xu et al. 2017; Sun et al. 2017; Mao et al. 2016; Tan et al. 2017) have proposed for mobile edge computing. Most of them focus on determining whether a task should be executed on the edge severs or the remote cloud, with the objective of minimizing the total delay. Different from these previous works, we consider the quality of sensing data collection service provided by a mobile edge cloud, and leverage the correlations in sensing data.

In response to the aforementioned challenges, we first introduce the concept of *spatial–temporal coverage* which bridges the quality of service and the set of collected sensing data. Then, our approach consists of two main building blocks. *The first building block* determines in which cells sensing data are collected and which smartphone is selected to do the sensing task. Note that even the offline version of this problem is an integer linear program (ILP), which is a well-known NP-Complete problem. An online near-optimal algorithm is proposed in this paper, which minimizes the time-averaged total cost while meeting the requirement of quality of service. *The second building block* determines the payment to each smartphone users. We first introduce a reverse auction model with two levels of competition. A pricing policy is designed, which guarantees the truthfulness of strategic users. To evaluate the performance of our approach, we have conducted extensive simulations based on real datasets. Comparative results show that our online approach outperforms other competing schemes, producing a high quality of service with a much lower budget.

The major contributions of this work are summarized as follows.

- This work is the first attempt, to the best of our knowledge, to consider the spatial and temporal correlations among sensing data in task dispatching for mobile edge clouds with rational and selfish edge-cloud servers. We analyze the spatial and temporal correlations based on real datasets and show that a subset of sensing data collected can recover the uncollected data.
- By defining the quality of service as the spatial and temporal coverage of collected data, we formulate the task dispatching problem as an ILP problem, assuming all true costs of smartphone users over time are known. An online algorithm is proposed, which is proved to achieve the near-optimal time-averaged total cost.
- Based on the algorithm of task dispatching, we propose a pricing policy by modeling the interactions between the platform and users as a reverse auction with two-level competition. The payment policy is proved to guarantee the truthfulness of rational users.
- We perform comprehensive simulations based on real datasets. The results show that our online algorithm achieves lower total incentives compared with baseline algorithms under different settings. Moreover, the recover error of uncollected data is reduced as well.

The remainder of the paper proceeds as follows. The next section presents the motivation, the system model, the problem formulation, and the overview of our proposed approach. In Sect. 3, the online task dispatching algorithm for mobile edge clouds is described. Section 4 describes the pricing policy. The performance is evaluated in Sect. 5.

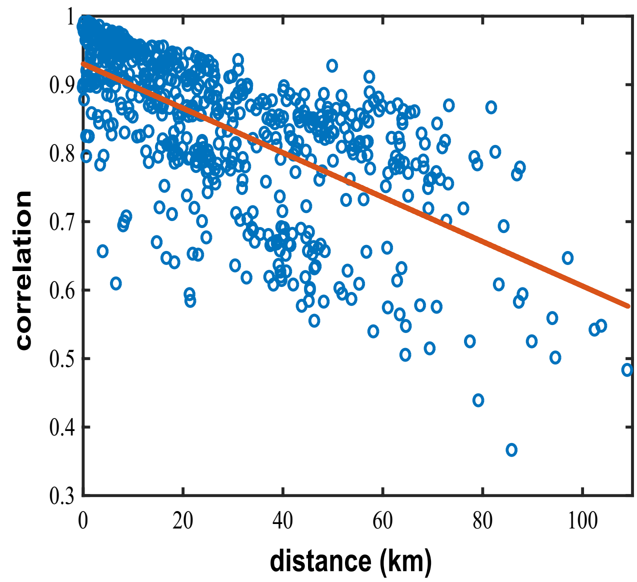


Fig. 3 Correlation vs. distance. The red line linearly fits all blue points

Related work is discussed in Sect. 6. We conclude the paper in Sect. 7.

2 Motivation and overview

2.1 Motivation

In this subsection, we first employ a real dataset, *PM2.5 Air Quality* (Zheng et al. 2013), to analyze the existence of spatial and temporal correlations in sensing data. Then, we present how spatial and temporal correlations can help reduce the amount of collected sensing data. The dataset is collected from 36 air pollution monitoring stations in Beijing, where each station records a measurement of the local PM2.5 concentration per hour. Like (Zheng et al. 2013), we split the whole area into 1 km × 1 km grids (only grids with stations are used) and divide time into 1 h time slots (during 2/2/2014~2/8/2014). Thus, there are 36 (grids) × 168 (time slots) measurements in total.

Figures 3 and 4 show the correlations among the measurements collected in different locations and time slots, respectively. In Fig. 3, each blue point is plotted by calculating the Pearson correlation between the measurements collected by two stations over all time slots and the distance between the two stations. The red line linearly fits all the points plotted by all pairs of the stations. Similarly, the vertical axis of Fig. 4 presents the Pearson correlation between the measurements collected in two time slots by all stations, while the horizontal axis presents the time difference

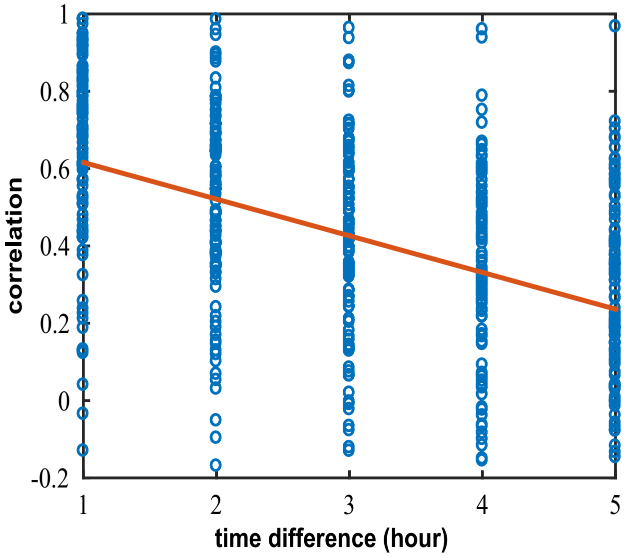


Fig. 4 Correlation vs. time slot. The red line linearly fits all blue points (color figure online)

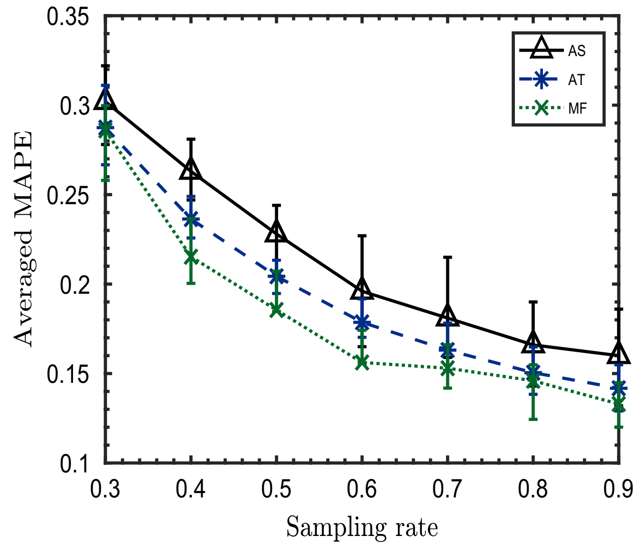


Fig. 5 Recovery error vs. sampling rate

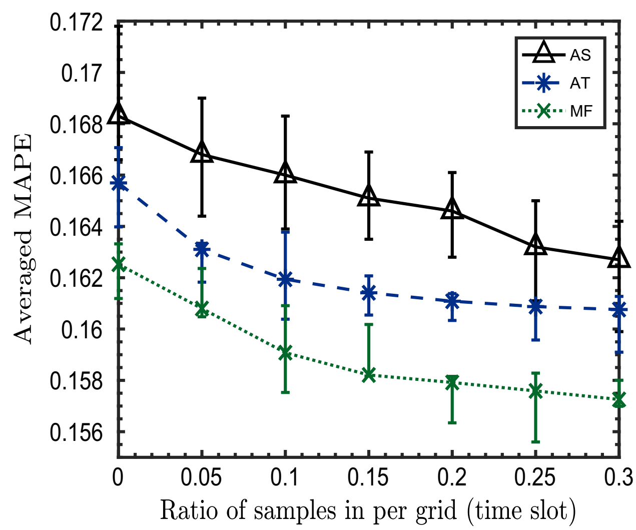


Fig. 6 Recovery error vs. ratio of measurements per grid (slot)

between the two time slots. Any pair of time slots whose time difference is between 1 and 5 h are plotted, and the red line linearly fits all the points. We can clearly see that sensing data in nearby locations and time slots have strong correlations, while the correlation declines as the distance or time difference increases.

Inspired by the existence of spatial and temporal correlations, a required service can be completed by collecting a subset of sensing data and recovering the uncollected data by employing interpolation methods. To verify this intuition, we do experiments on the PM2.5 dataset. We randomly select a proportion of the measurements as collected data, and others are viewed as missing data. The ratio of selected measurements is noted as *sampling rate*. To infer the missing data, we employ three interpolation methods: (1) *AS* fills a missing data with the average of the measurements collected by the nearest three stations belonging to the time slot; (2) *AT* fills a missing data with the average of the measurements collected in the nearest three time slots by the same station; (3) *MF* (Dhillon and Sra 2006) interpolates a missing data by searching the optimal factorization of the measurement matrix belonging to $\mathbb{R}^{36\times168}$. By comparing the recovered results and the ground truth of the missing data, averaged Mean Absolute Percentage Error (MAPE) (2018) is calculated to quantify the performance of recovery accuracy achieved.

Figure 5 shows the accuracy achieved by the three methods when the sampling rate increases from 0.3 to 0.9. In accord with our intuition, more samples, better recovery accuracy. However, we can find that MF can achieve less than 20% error with even only 50% samples. Moreover, the increase of recovery accuracy becomes smaller and smaller when the sampling rate tends to 1. Figure 5 demonstrates that collecting a small subset of sensing data is sufficient to obtain a high recovery accuracy, due to the existence of spatial and temporal correlations in sensing data.

Besides the sampling rate, the distribution in space and time dimensions of collected sensing data can also impact the performance of recovery. To study it, we limit the minimum number of measurements collected in each grid over time (as well as in per time slot of all grids) instead of completely random selection. Note that a higher ratio indicates that collected sensing data are more uniformly distributed in space and time. In Fig. 6, we fix the sampling rate as 0.7, and vary

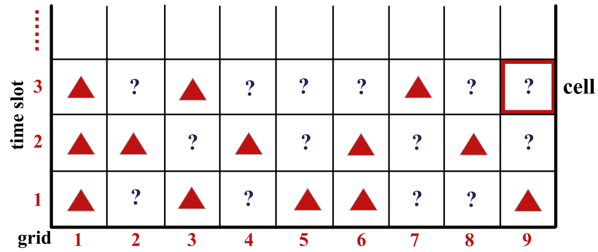


Fig. 7 An example of matrix $\mathbf{M}$, where missing data are marked as "?"

the ratio of measurements per-grid (slot) from 0 to 0.3 (i.e., at least $11 \approx 0.3 \times 36$ measurements per grid and $50 \approx 0.3 \times 168$ measurements per slot). From Fig. 6, we can see that the more uniform distribution, the better performance achieved by interpolation methods. However, the performance improvement is very limited when the ratio exceeds a threshold (e.g., 0.2 in Fig. 6). Thus, we can conclude that the spatial–temporal distribution of collected data is unnecessary to be completely uniform, which brings more choices for data collection. As shown in Figs. 5 and 6, MF performs better than AS and AT, as it takes advantage of both spatial and temporal correlations.

2.2 System and service model

In this work, we consider a typical edge-cloud system for urban monitoring applications, which consists of a centralized cloud, base stations and mobile smartphones at the network edge. The goal of a specific service (e.g., noise mappling) is to continuously collect fine-grained sensing data by recruiting mobile smartphone users. We divide the area into grids of the same size, and divide time into slots of equal interval. The sets of grids and time slots are denoted as $\mathcal{N} = \{1, 2, \dots, N\}$ and $\mathcal{T} = \{1, 2, \dots, T\}$, respectively. Then, the complete set of sensing data can be represented by a two-dimensional matrix $\mathbf{M} \in \mathbb{R}^{N \times T}$ as shown in Fig. 7, where each data entry $\mathbf{M}(n, t)$ is called a *cell*. For simplicity, we assume that the sensing data collected in a cell have the same value, while different cells may have different values. Thus, sensing data collected from an arbitrary user in grid n and time slot t can fulfill cell $\mathbf{M}(n, t)$.

We denote the set of smartphones which can participate in collecting sensing data for cell $\mathbf{M}(n, t)$ as $\mathcal{S}_n[t]$. Note that $\mathcal{S}_n[t]$ varies over time, because edge-cloud smartphones are mobile who may join and leave at any time. Besides, the union of all smartphones in the edge-cloud system, $\bigcup\{\mathcal{S}_n[t]\}$, is dynamic as well. A considerable cost, such as energy consumption and bandwidth usage, is incurred to an arbitrary smartphone s for collecting sensing data, which is denoted as $c_s \geq 0$. Note that the costs of different smartphones are heterogenous and unknown to the centralized cloud. We assume there exists a base station in each grid, which acts as a edge server in our edge-cloud system model. The set of base stations are denoted by $\mathcal{B} = \{B_1, B_2, \dots, B_N\}$.

In a real-world edge-cloud system, monetary incentives need to be provided to the smartphone users who are dispatched tasks. To reduce the total budget of a service, the system tends to choose the smartphone with the lowest cost for collecting the data in a specific cell. Thus, with true cost information of all smartphones, the cost spent for collecting sensing data in cell $\mathbf{M}(n, t)$ can be derived as

$$c_n[t] = c_{s_{n,t}} = \min\{c_s, \forall s \in \mathcal{S}_n[t]\}, \tag{1}$$

where $s_{n,t}$ denotes the smartphone with the lowest cost in grid n and time slot t. The payment given to $s_{n,t}$ is denoted by $p_n[t]$. However, strategic smartphone users may misreport their costs to obtain a higher payment. To differentiate from true cost c_s of smartphone s, we denote the claimed cost of s as c'_s.

According to the observations presented in Sect. 2.1, only a small subset of the complete sensing data in all cells of $\mathbf{M}$ is needed to collect. Thus, the budget of a service can be lowered by controlling which cells are selected to collect sensing data. We define an index vector $\mathbf{I} = \{I_n[t], \forall n \in \mathcal{N}, \forall t \in \mathcal{T}\}$ to indicate whether a cell is selected, where

$$I_n[t] = \begin{cases} 1, & \text{if } \mathbf{M}(n, t) \text{ is selected to collect data,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Accordingly, smartphone $s_{n,t}$ is recruited to contribute sensing data for cell $\mathbf{M}(n, t)$ when $I_n[t] = 1$.

Finally, we characterize the quality of service by defining the concept of *spatial and temporal coverage*, based on the amount and spatial–temporal distribution of the cells with collected sensing data.

Definition 1 (*Spatial and temporal coverage*) Given a matrix $\mathbf{M} \in \mathbb{R}^{N \times T}$ with fulfilled cells indexed by vector $\mathbf{I}$, its spatial and temporal coverage is defined as three metrics:

- Spatial–temporal coverage, i.e., $\frac{1}{TN}\sum_{t=1}^{T}\sum_{n=1}^{N} I_n[t]$;
- Temporal coverage per grid, i.e., $\frac{1}{T}\sum_{t=1}^{T} I_n[t], \forall n \in \mathcal{N}$;
- Spatial coverage per slot, i.e., $\frac{1}{N}\sum_{n=1}^{N} I_n[t], \forall t \in \mathcal{T}$.

Remark We would like to emphasize that one of the key contributions of this work is the novel concept definition of quality of service, which is inspired by our observations based on real datasets. As shown in the problem formulation, this concept bridges the quality of service and the amount and distribution of collected sensing data.

2.3 Problem formulation

In this paper, two coupled problems should be addressed. The *first problem* is to determine the cells and the corresponding smartphones selected to collect sensing data, with the goal of minimizing the total cost spent and guarantee the quality of service required by a certain request. For clarity, we only consider a service request in our work. As shown in Sect. 2.1, a high recovery accuracy can be achieved by data recovery methods if the spatial and temporal coverage of the cells satisfies certain requirements. Thus, we mathematically formulate the problem of task dispatching in real time as Definition 2.

Definition 2 (*Task Dispatching Problem*) Based on true cost c_s of each smartphone, the centralized cloud selects a subset of cells (indexed by $\mathbf{I}$) to collect sensing data by solving the following problem with constraints on the spatial and temporal coverage of Matrix $\mathbf{M}$. Smartphone $s_{n,t}$ is recruited if cell $\mathbf{M}(n,t)$ is selected.

Problem 1 (ILP):

$$\min \frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N} c_n[t]\cdot I_n[t] \tag{3}$$

$$\text{s.t.} \frac{1}{TN}\sum_{t=1}^{T}\sum_{n=1}^{N} I_n[t] \ge \delta_1, \tag{4}$$

$$\frac{1}{T}\sum_{t=1}^{T} I_n[t] \ge \delta_2, \quad \forall n \in \mathcal{N}, \tag{5}$$

$$\frac{1}{N}\sum_{n=1}^{N} I_n[t] \ge \delta_3, \quad \forall t \in \mathcal{T}, \tag{6}$$

$$I_n[t] \in \{0,1\}, \forall n \in \mathcal{N}, \quad \forall t \in \mathcal{T}. \tag{7}$$

Constraints (4), (5) and (6) are used to guarantee the quality of service, in which parameters δ_1, δ_2 and δ_3 are the requirements of a service request.[1] For completeness, we provide a straightforward approach for determining the parameter values, with the assumption that the correlations among sensing data of a specific application keep unchanged over time. The approach is executed in a real-world crowdsourcing application as follows. When the platform receives a new request from an urban monitoring application, it continuously collects sensing data in all grids for a short period. With the complete sensing data, the application can obtain the relationship between the data recovery accuracy and the values of $\delta_1, \delta_2, \delta_3$ as shown in Figs. 5 and 6, by randomly removing some data. Given a specific requirement, the platform can decide the values of $\delta_1, \delta_2, \delta_3$ according to their relationship.

For convenience, we use $C(\mathbf{I})$ to denote the total cost incurred on smartphones, i.e., $C(\mathbf{I}) = \sum_{t=1}^{T}\sum_{n=1}^{N} c_n[t]\cdot I_n[t]$. Due to constraint (7), Problem 1 is an ILP problem, which is a well-known NP-Complete problem. We denote the optimal solution of Problem 1 as $\mathbf{I}^* = \{I_n^*[t] \in \{0,1\}, \forall n \in \mathcal{N}, \forall t \in \mathcal{T}\}$.

The *second problem* is to determine the amount of payment given to each recruited smartphone. As we consider strategic smartphone users who may misreport their cost information, a pricing policy should be provided to smartphone users which can enforce them being truthful. Consequently, Problem 1 can be solved based on the true costs reported by smartphone users. To achieve this goal, the pricing policy should satisfy the following three properties.

Definition 3 (*Individual Rationality*) The payoff (i.e., $p_n[t] - c_n[t]$) of each recruited smartphone user is nonnegative.

A rational smartphone user will not participate in sensing if his/her cost is not covered by the payment.

Definition 4 (*Computational Efficiency*) An algorithm has the property of computational efficiency if it terminates in polynomial time.

As the process of recruiting users is repeated over time, the pricing policy cannot have high computational complexity. Otherwise, it is useless for an online system.

Definition 5 (*Truthfulness*) An incentive mechanism is truthful if and only if each smartphone user s, cannot raise its payoff by reporting a false cost, i.e., $\varrho_s(c_s, c_{-s}) \ge \varrho_s(c_s', c_{-s})$, where $\varrho_s(\cdot)$ denotes the payoff obtained by s given all claimed costs, and c_{-s} denotes the set of costs reported by other users except s.

This property guarantees that strategic smartphone users have no motivation to misreport their costs.

2.4 Overview

To reduce the budget of an urban monitoring service in a edge cloud system by leveraging the spatial and temporal correlations of sensing data, we propose an online approach containing two main building blocks, which are explained in details in the following two sections, respectively. To better understand our approach, we first give an overview in this subsection.

[1] Note that there should always exist $\delta_1 > \max\{\delta_2, \delta_3\}$, otherwise (4) can be satisfied directly when (5) or (6) is satisfied.

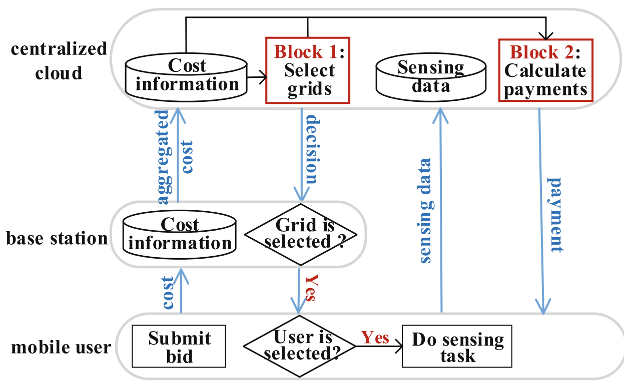


Fig. 8 The interactions between the centralized cloud, base stations and smartphones in our proposed approach in each time slot. Two main building blocks are designed to determine the grids selected for collecting sensing data and determine the payments given to the recruited smartphones, which are enclosed in red rectangles

The interactions between the centralized cloud, base stations and each smartphone in each time slot are shown in Fig. 8. At the beginning of a time slot, each smartphone who can do sensing tasks reports its cost to the base station in its grid firstly. Then, each base station selects the smartphone with the lowest cost in the grid and report the cost information to the centralized cloud. Then, the centralized cloud selects a subset of grids where sensing data are collected, based on the cost information in all grids. Thus, the smartphone with the lowest cost in the selected grids are dispatched tasks and collect sensing data. Finally, the payment given to each recruited smartphone is calculated according to the pricing policy. To implement the approach, two main building blocks are designed, i.e., determining which grids selected for collecting sensing data and determining how many payments given to the recruited smartphones, respectively.

For the *first building block*, an online near-optimal algorithm is proposed. In each time slot, with true costs of smartphone users, the algorithm selects a subset of grids to collect sensing data. The time-averaged total cost achieved by this algorithm is close to the offline optimum. For the *second building block*, we introduce the reverse auction model with two-level competition and design a truthful pricing policy for strategic smartphone users. Thus, true cost information can be obtained. Based on the pricing policy and the result of the first block, the payment given to each recruited smartphone can be calculated.

3 Online task dispatching algorithm

In this section, we design an online algorithm for the first building block in Fig. 8, which determines the grids selected to collect sensing data in the current time slot. Here, we assume true costs are reported by smartphones in the beginning of the current time slot, given the pricing policy introduced in the next section.

3.1 Problem decomposition

We first decompose Problem 1 defined over time into a series of subproblems, each of which can be solved in a time slot.

3.1.1 Virtual queues

According to the Lyapunov optimization (Neely 2010), *virtual queues* can be introduced to guarantee time-averaged constraints [such as (4) and (5)] to be satisfied. Here, we define a virtual queue P for constraint (4) and a virtual queue Q_n for each constraint in (5), respectively. These queues are called "virtual" as they do not exist physically. Note that only their backlogs need to be kept by the centralized cloud. We denote the backlog of P in time slot t as $P[t]$, which can be updated as

$$P[t+1] = \max\{P[t] - \frac{1}{N}\sum_{n=1}^{N} I_n[t], 0\} + \delta_1, \tag{8}$$

according to (4). The value of $P[t]$ can be seen as the averaged number of grids that should be selected in time slot t to satisfy the requirement on the spatial–temporal coverage. The arrival rate of P is equal to δ_1 while the departure in t is $\frac{1}{N}\sum_{n=1}^{N} I_n[t]$.

Similarly, the backlog $Q_n[t]$ of Q_n represents how many grids should be selected in t to satisfy the requirement on the temporal coverage of grid n. The updating rule of Q_n can be derived as

$$Q_n[t+1] = \max\{Q_n[t] - I_n[t], 0\} + \delta_2, \quad \forall n \in \mathcal{N}, \tag{9}$$

where δ_2 is the arrival rate of cells required to fulfill, and $I_n[t]$ denotes the number of cells departing from the queue in time slot t. Note that, satisfying constraints (4) and (5) can be equivalently converted into maintaining the stability of virtual queues P and $\{Q_n, \forall n\}$.

3.1.2 Queue stability

To maintain the stability of P and $\{Q_n, \forall n\}$, we define the *Lyapunov function*, i.e.,

$$L[t] \triangleq \frac{1}{2}(P[t]^2 + \sum_{n\in\mathcal{N}} Q_n[t]^2), \tag{10}$$

to measure the level of queue congestion. The smaller value of $L[t]$, the smaller queue backlogs of P and $\{Q_n, \forall n\}$. Then, the *Lyapunov drift* can be defined as

$$\Delta[t] \triangleq L[t+1] - L[t], \tag{11}$$

which represents the shift of Lyapunov function between two consecutive time slots. A smaller value of $\Delta[t]$ indicates that the backlogs of all virtual queues are pushed towards to a lower level. The smaller value of $\Delta[t]$, the stronger stability is achieved by the virtual queues.

Corollary 1 *An upper bound of the Lyapunov drift can be derived as*

$$\Delta[t] \le B + P[t]\left(\delta_1 - \frac{1}{N}\sum_{n\in\mathcal{N}} I_n[t]\right) + \sum_{n\in\mathcal{N}} Q_n[t](\delta_2 - I_n[t])$$

where $B \triangleq \frac{1}{2}[(1+\delta_1^2) + N\cdot(1+\delta_2^2)]$ *is a constant.*

Proof Based on the fact that $(\max\{a-b,0\}+c)^2 \le a^2+b^2+c^2+2a(c-b)$, we can deduce that

$$\begin{aligned} P[t+1]^2 &\le P[t]^2 + \left(\frac{1}{N}\sum_{n\in\mathcal{N}} I_n[t]\right)^2 + \delta_1^2 + 2P[t](\delta_1 - I_n[t]) \\ &\le P[t]^2 + (1+\delta_1^2) + 2P[t]\left(\delta_1 - \frac{1}{N}\sum_{n\in\mathcal{N}} I_n[t]\right), \end{aligned}$$

$$\begin{aligned} Q_n[t+1]^2 &\le Q_n[t]^2 + I_n[t]^2 + \delta_2^2 + 2Q_n[t](\delta_2 - I_n[t]) \\ &\le Q_n[t]^2 + (1+\delta_2^2) + 2Q_n[t](\delta_2 - I_n[t]). \end{aligned}$$

We can obtain that

$$\begin{aligned} \Delta[t] &= \frac{1}{2}[(P[t+1]^2 - P[t]^2) + \sum_{n\in\mathcal{N}}(Q[t+1]^2 - Q[t]^2)] \\ &\le \frac{1}{2}[(1+\delta_1^2) + 2P[t]\left(\delta_1 - \frac{1}{N}\sum_{n\in\mathcal{N}} I_n[t]\right) \\ &\quad + N(1+\delta_2^2) + 2\sum_{n\in\mathcal{N}} Q_n[t](\delta_2 - I_n[t])]. \end{aligned}$$

Therefore, the upper bound of $\Delta[t]$ is proved.

3.1.3 Problem reformulation

To minimize the total cost and maintain the queue stability at the same time, we define a new objective in each time slot by combining the two aspects, named *drift-plus-penalty*, i.e.,

$$\Delta'[t] = \Delta[t] + V\cdot\sum_{n\in\mathcal{N}} c_n[t] I_n[t], \tag{12}$$

where V is a tunable nonnegative parameter, denoting the weight of minimizing total cost, compared with keeping queue stability. Note that minimizing the drift-plus-penalty function directly is impossible due to the definition of $\Delta[t]$. Thus, we turn to minimizing its upper bound instead. The upper bound of $\Delta'[t]$ can be easily obtained based on Corollary 1 as

$$\begin{aligned} \Delta'[t] \le & B + \delta_1 P[t] + \delta_2 \sum_{n\in\mathcal{N}} Q_n[t] \\ & + \sum_{n\in\mathcal{N}}(Vc_n[t] - \frac{P[t]}{N} - Q_n[t])I_n[t]. \end{aligned} \tag{13}$$

3.2 Online algorithm

In each time slot, given the backlogs of virtual queues, the centralized cloud can decide which grids should be selected by minimizing the last term in the upper bound of $\Delta'[t]$ (as the other terms are constants). So far, we have decomposed Problem 1 into a series of subproblems, each of which can be solved in a time slot given the current true cost information. The subproblem solved in time slot t is formulated as follows.

Problem 2 (in time slot t**):**

$$\min \sum_{n=1}^{N}(c_n[t] - \frac{P[t]+NQ_n[t]}{VN})I_n[t] \tag{14}$$

$$\begin{aligned} \text{s.t.,} & \sum_{n=1}^{N} I_n[t] \ge \delta_3\cdot N, \\ & I_n[t]\in\{0,1\}, \quad \forall n\in\mathcal{N}. \end{aligned} \tag{15}$$

The time-averaged constraints in Problem 1 have been converted to keeping stability of virtual queues and combined into optimization objective (14). According to (14), we define *regulated cost* as

$$\widetilde{c}_n[t] \triangleq c_n[t] - \gamma_n[t], \tag{16}$$

where $\gamma_n[t] = \frac{P[t]+NQ_n[t]}{VN}$. The optimal solution of Problem 2 in time slot t is denoted as $\mathbf{I}^{\ddagger}[t] = \{I_n^{\ddagger}[t]\in\{0,1\}, \forall n\in\mathcal{N}\}$. The overall solution, comprised of the solutions of Problem 2 in all time slots, is denoted as $\mathbf{I}^{\ddagger} = \{\mathbf{I}^{\ddagger}[t], \forall t\in\mathcal{T}\}$.

Algorithm 1 Grid selection in time slot t

Input: True costs $c_n[t]$ and queue backlogs $P[t], Q_n[t]$
Output: Solution $\mathbf{I}^{\ddagger}[t]$ and queue backlogs $P[t+1], Q_n[t+1]$ prepared for time slot $(t+1)$

// Each base station B_n
Calculate $\widetilde{c}_n[t]$ according to (16) and reports to the centralized cloud;
// The centralized cloud
Initialize $I_n^{\ddagger}[t] = \begin{cases} 1, & \text{if } \widetilde{c}_n[t] \le 0 \\ 0, & \text{otherwise} \end{cases}, \forall n\in\mathcal{N}$;
while $\sum_{n=1}^{N} I_n^{\ddagger}[t] < \delta_3\cdot N$ **do**
 $n^* = \arg\min_n\{\widetilde{c}_n[t] | I_n^{\ddagger}[t] = 0, \forall n\}$;
 $I_{n^*}^{\ddagger}[t] = 1$;
end while
Use $\mathbf{I}^{\ddagger}[t]$ to update the backlogs of virtual queues according to (8) and (9);
return $\mathbf{I}^{\ddagger}[t]$, $P[t+1]$ and $Q_n[t+1]$.

As Problem 1 has been decomposed into Problem 2 to be solved in each time slot, we propose a *greedy-search-based* algorithm to determine which grids should be selected in time slot t to collect sensing data. The details are given in Algorithm 1. The regulated cost $\widetilde{c}_n[t]$ is calculated by base station B_n based on the true costs of all smartphones in the grid. All grids with non-positive values are selected. If constraint (15) is not satisfied, the centralized cloud sorts other grids in an increasing order of their regulated costs, and select grids one by one until (15) is satisfied. The smartphones with the lowest costs in the selected grids are dispatched sensing tasks. The computational complexity of Algorithm 1 is $O(N\log(N))$. Note that $\mathbf{I}^{\ddagger}[t]$ output by Algorithm 1 is the optimal solution of Problem 2. The online algorithm for task dispatching is to repeatedly execute Algorithm 1 in each time slot.

Remark Note that our key contribution in this part is not on the utilization of Lyapunov optimization theory, but rather how we explicitly define the virtual queues, derive the upper bound, and propose the optimal algorithm.

3.3 Theoretical analysis

Theorem 1 (Optimality) *The time-averaged total cost obtained by our proposed online algorithm is near-optimal when the number of time slots tends to infinity. Especially, the gap between it and the offline optimum is within a constant* $\frac{B}{V}$, *i.e.,*

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t\in\mathcal{T}}C(\mathbf{I}^{\ddagger})\le\lim_{T\to\infty}\frac{1}{T}C(\mathbf{I}^{*})+\frac{B}{V}.$$

Proof We assume there is an arbitrary online policy θ that chooses a feasible solution $\mathbf{I}^{\theta}$ of Problem 1. As our algorithm minimizes the upper bound of $\Delta^{'}[t]$, there is

$$\begin{aligned}\Delta^{'}[t]\le{}& B+\delta_1P[t]+\delta_2\sum_{n\in\mathcal{N}}Q_n[t]\\&+\sum_{n\in\mathcal{N}}(Vc_n[t]-\frac{P[t]}{N}-Q_n[t])I_n^{\theta}[t]\\\le{}& B+P[t](\delta_1-\frac{1}{N}\sum_{n\in\mathcal{N}}I_n^{\theta}[t])\\&+\sum_{n\in\mathcal{N}}Q_n[t](\delta_2-I_n^{\theta}[t])+V\sum_{n\in\mathcal{N}}c_n[t]I_n^{\theta}[t]\end{aligned}$$

Let us sum both sides from $t=1$ to $t=T$ and divide them by T simultaneously. We can obtain that

$$\begin{aligned}\frac{L[T+1]-L[1]}{T}&+\frac{V}{T}\sum_{t\in\mathcal{T}}C(\mathbf{I}^{\ddagger})\le B+\frac{V}{T}\sum_{t\in\mathcal{T}}C(\mathbf{I}^{\theta})\\&+\frac{1}{T}\sum_{t\in\mathcal{T}}P[t](\delta_1-\frac{1}{N}\sum_{n\in\mathcal{N}}I_n^{\theta}[t])\\&+\frac{1}{T}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}Q_n[t](\delta_2-I_n^{\theta}[t])\end{aligned}$$

Taking the limit $T\to\infty$ of both sides after diving them by V, we obtain that

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t\in\mathcal{T}}C(\mathbf{I}^{\ddagger})\le\lim_{T\to\infty}\frac{1}{T}C(\mathbf{I}^{*})+\frac{B}{V},$$

since $\lim_{T\to\infty}\frac{L[T+1]-L[1]}{T}=0$, and

$$\begin{aligned}&\lim_{T\to\infty}\frac{1}{T}\sum_{t\in\mathcal{T}}P[t](\delta_1-\frac{1}{N}\sum_{n\in\mathcal{N}}I_n^{\theta}[t])\\&\quad\le P^{\max}\cdot(\delta_1-\lim_{T\to\infty}\frac{1}{NT}\sum_{t\in\mathcal{T}}\sum_{n\in\mathcal{N}}I_n^{\theta}[t])=0,\\&\lim_{T\to\infty}\frac{1}{T}\sum_{t\in\mathcal{T}}Q_n[t](\delta_2-I_n^{\theta}[t])\\&\quad\le Q_n^{\max}\cdot(\delta_2-\lim_{T\to\infty}\frac{1}{T}\sum_{t\in\mathcal{T}}I_n^{\theta}[t])=0,\end{aligned}$$

where $P^{\max}=\max\{P[t],\forall t\}$ and $Q_n^{\max}=\max\{Q_n[t],\forall t\}$.

From Theorem 1, it can be easily seen that a larger value of V implies better performance in minimizing the total cost. However, it may also cause time-averaged constraints not satisfied, due to the less stability of virtual queues.

4 Online pricing policy

In this section, we design an online pricing policy for the second building block in Fig. 8, which determines the payments given to the smartphones dispatched tasks in real time. The pricing policy is proved to guarantee the truthfulness of strategic smartphone users, which makes the assumption of our online task dispatching algorithm is satisfied.

4.1 Reverse auction modeling

We first introduce *reverse auction* (Nisan et al. 2007) to model the interactions between the centralized cloud, base stations and smartphone users, as shown in Fig. 8. In this auction, smartphone users trading in sensing data are sellers while the centralized cloud is the unique buyer. The auction is called "reverse" because sellers compete to obtain business from the buyer. Different from a standard reverse auction, there exist *two levels of competition* in our problem, which increases the difficulty of designing a truthful pricing policy.

- **Level 1**: The competition among smartphone users in the same grid, e.g., $\{\forall s \in \mathcal{S}_n[t]\}$;
- **Level 2**: The competition among the smartphone users winning in their own grids, i.e., $\{s_{n,t}, \forall n \in \mathcal{N}\}$.

The final winners are the smartphones selected to do sensing tasks by Algorithm 1, i.e., $\{s_{n,t} | I_n^{\ddagger}[t] = 1, \forall n \in \mathcal{N}\}$.

4.2 Payment design

To begin with, we define some notations for the convenience of description. Let $\{I_k^{\Pi}[t], \forall k \in \Pi\}$ be the solution of Problem 2 when $\mathcal{N}$ is replaced by a subset of grids $\Pi \subseteq \mathcal{N}$. The total regulated cost optimized by Algorithm 1 in time slot t is

$$C_t(\Pi) \triangleq \sum_{k \in \Pi} \widetilde{c}_k[t] I_k^{\Pi}[t], \tag{17}$$

which is a function of Π. We have the following corollary, which can be easily proved by the reduction to absurdity.

Corollary 2 *$C_t(\Pi)$ is monotonously decreasing, i.e., for $\forall \Gamma \subsetneq \Pi$, there exists $C_t(\Gamma) \geq C_t(\Pi)$.*

For the first level (l_1) of competition, according to the second price theory (Nisan et al. 2007), smartphones will keep truthful if the payment given to winner $s_{n,t}$ is equal to the second minimum cost of all smartphones in the same grid, i.e.,

$$p_n^{l_1}[t] \triangleq \min\{c_s, \forall s \in \mathcal{S}_n[t] \backslash \{s_{n,t}\}\}. \tag{18}$$

For the second level (l_2) of competition, the idea of Vickrey-Clarke-Groves (VCG) mechanism (Nisan et al. 2007) is employed. Besides cost $c_{s_{n,t}}$, an extra payment should be given to $s_{n,t}$ if it is selected, which is equal to the marginal cost caused due to the absence of $s_{n,t}$. Thus, we define

$$p_n^{l_2}[t] \triangleq C_t(\mathcal{N} \backslash \{n\}) - \sum_{k \in \mathcal{N} \backslash \{n\}} \widetilde{c}_k[t] I_k^{\mathcal{N}}[t] + \gamma_n[t], \tag{19}$$

where the first two terms are derived from the standard VCG mechanism, and the third term is for compensating the cost regulation.

Taking the two payments into consideration, we design a pricing policy for determining the payments given to selected smartphones as follows.

Algorithm 2 Online payment determination

Input: Parameter V

Output: Cell selection $\mathbf{I}^{\ddagger}$ and payments $p_n[t]$

```
1: Initialize P[1] = 0 and Q_n[1] = 0
2: for time slot t = 1, 2, ..., T do
3:     Collect cost information from each smartphone s ∈ {S_n[t], ∀n ∈ N};
4:     Select grids I‡[t] = arg min_{I_n[t]} Σ_{n∈N} c̃_n[t] I_n[t];
5:     Determine payments p_n[t] = I_n‡[t] · min{p_n^{l1}[t], p_n^{l2}[t]};
6:     Update virtual queues P[t+1] and Q_n[t+1];
7: end for
```

Pricing Policy: If grid n is selected to collect sensing data in time slot t according to the solution of Problem 2, i.e., $I_n^{\ddagger}[t] = 1$, the payment given to winner $s_{n,t}$ is

$$p_n[t] = \min\{p_n^{l_1}[t], p_n^{l_2}[t]\}. \tag{20}$$

Based on the above payment policy and the result of task dispatching, the payments given to the selected smartphones can be calculated in each time slot, which is detailed described in Algorithm 2. At the beginning of each time slot t, the centralized cloud calculates the result of grid selection $\mathbf{I}^{\ddagger}[t]$ based on the true cost information reported by smartphones, and then determines the payments given to the selected smartphones with according to the pricing policy.

Remark We would like to emphasize that our key contributions in this part are two-fold. First, we develop the standard reverse auction model into a novel one with two-level competition among smartphones, to accord with the result of task dispatching. Second, we propose a novel pricing policy, which sophisticatedly combines the payments of the two levels of competition.

4.3 Theoretical analysis

Theorem 2 (Truthfulness) *The proposed pricing policy is truthful, i.e., each smartphone submits its true cost no matter what others submit.*

Proof Firstly, we show that a smartphone has no motivation to claim a higher cost. If smartphone s claims a higher cost, i.e., $c_s^{'} > c_s$, the only possible situation is that it wins in the two-level competition (e.g., $s = s_{n,t}$ and $I_n^{\ddagger}[t] = 1$) regardless of bidding c_s or $c_s^{'}$. However, smartphone s obtains the same payment under both strategies. It is because that $p_n^{l_1}[t]$ and $p_n^{l_2}[t]$ have no relation to the value of $c_{s_{n,t}}$ according to (18) and (19), as long as the bids submitted by others are fixed.

Secondly, we show that a smartphone has no motivation to claim a lower cost. If smartphone s claims a lower cost, i.e., $c_s^{'} < c_s$, the only possible situation is that it loses (wins) if bidding truthfully (non-truthfully). We illustrate that its payoff (e.g., $p_n[t] - c_s$) is negative under the two cases. Case 1: smartphone s loses (wins) in the first-level competition if bidding truthfully (non-truthfully), which means $c_s^{'} \leq \min\{c_r, \forall r \in \mathcal{S}_n[t] \backslash \{s\}\} < c_s$. Thus, we have

$c_s > p_n^{l_1}[t] \geq p_n[t]$. Case 2: smartphone s wins in the first-level competition but loses (wins) in the second-level competition if bidding truthfully (non-truthfully), i.e., $I_n^{\ddagger}[t] = 0(1)$ when $\widetilde{c}_n[t] = c_s(c_s^{'}) - \gamma_n[t]$. As $I_n^{\ddagger}[t]$ is set to one by the increasing order of $\widetilde{c}_n[t]$ in Algorithm 1, there exists $\max\{\widetilde{c}_k[t] | I_k^{\ddagger}[t] = 1\} \leq \min\{\widetilde{c}_k[t] | I_k^{\ddagger}[t] = 0\}$. We can derive that

$$C_t(\mathcal{N}\backslash\{n\}) - \sum_{k\in\mathcal{N}\backslash\{n\}} \widetilde{c}_k[t] I_k^{\mathcal{N}}[t] < \min\{\widetilde{c}_k[t] | I_k^{\mathcal{N}\backslash\{n\}}[t] = 0\}.$$

Thus, we have $c_s > p_n^{l_2}[t] \geq p_n[t]$.

Theorem 3 (Individual Rationality) *Our proposed pricing policy achieves individual rationality, i.e., $p_n[t] \geq c_n[t]$ if $I_n^{\ddagger}[t] = 1$.*

Proof As smartphones are truthful, there exists $c_n[t] = c_{s_{n,t}} = \min\{c_s, \forall s \in \mathcal{S}_n[t]\}$. Clearly, $p_n^{l_1}[t] \geq c_n[t]$ according to (18). Based on (19), we can deduce that

$$\begin{aligned} p_n^{l_2}[t] &= C_t(\mathcal{N}\backslash\{n\}) - C_t(\mathcal{N}) + \widetilde{c}_n[t] I_n^{\mathcal{N}}[t] + \gamma_n[t] \\ &= C_t(\mathcal{N}\backslash\{n\}) - C_t(\mathcal{N}) + c_n[t]. \end{aligned}$$

Since $C_t(\mathcal{N}\backslash\{n\}) - C_t(\mathcal{N}) \geq 0$ according to Corollary 2, we have $p_n^{l_2}[t] \geq c_n[t]$. Therefore, we prove that $p_n[t] \geq c_n[t]$ no matter it equals $p_n^{l_1}[t]$ or $p_n^{l_2}[t]$. □

Theorem 4 (Computational Efficiency) *Algorithm 2 proposed for the payment determination problem has polynomial time complexity.*

Proof In each time slot, the complexity of selecting grids (line 4), and updating queues (line 6) is $O(N)$ while the complexity of determining payments (line 5) is $O(N^2)$. Thus, the aggregate complexity of Algorithm 2 is $O(N^2)$.

5 Performance evaluation

5.1 Simulation setup

We have conducted extensive simulations to evaluate the performance of our online algorithm for grid selection, by comparing with two baseline algorithms described in the following.

- *Random algorithm*: In each time slot, $\delta_1 N$ grids are randomly selected to collect data. Note that our pricing policy does not work in this grid selection algorithm.[2]
- *Greedy algorithm*: In each time slot, $\delta_1 N$ grids with the lowest costs are selected to collet data. Our pricing policy can be applied to this algorithm by replacing the regulated cost with the true cost of each smartphone.[3]

In our simulations, we consider an interested area with size 8 km × 8 km in New York City, which is divided into 1600 grids (each grid is a square of 200 m × 200 m). The total time span is ten days, divided into 240 time slots with each time slot equal to 1 h. The number of complaints about noise in 311 data (2018) is counted in each cell to represent the ground truth of its noise level. The missing data are recovered based on matrix factorization method. To simulate the population densities in different grids and time slots, we employ the check-in dataset crawled from Gowalla, a location-based social network. There are 127,558 check-ins from April 24, 2009 to October 13, 2013. Since not all smartphones would like to participate in sensing, we set a default percentage of willing smartphones, e.g., 50%. Two different cost distributions of each smartphone, i.e., uniform distribution (e.g., $c_s \sim \mathcal{U}(1, 10)$) and normal distribution (e.g., $c_s \sim \mathcal{N}(5, 1)$) are studied respectively. The default values of spatial and temporal coverage thresholds are $\delta_1 = 0.3$, $\delta_2 = 0.1$ and $\delta_3 = 0.1$.

5.2 Impact of *V*

The impact of parameter V on the performance of Algorithm 1 is studied first, as shown in Figs. 9, 10 and 11. Three metrics are compared by varying the value of V from 0.02 to 0.12, which are 1) total cost $C(\mathbf{I}^{\ddagger})$, 2) averaged backlog $\frac{1}{N+1}(P[T] + \sum_{n=1}^{N} Q_n[T])$, and 3) gap between δ_1 and the achieved spatial–temporal coverage. Figure 9 confirms our analysis in Sect. 3.1 that a larger V can achieve better performance in minimizing the total cost. Figure 10 shows that the averaged backlog of virtual queues increases with V. Accordingly, Fig. 11 reveals that V cannot be infinitely enlarged to achieve better performance, since the time-averaged constraints cannot be satisfied. For fairness, we take the value of V satisfying all constraints in the following performance comparison.

5.3 Performance comparison

5.3.1 Total cost

We compare the total costs achieved by our online algorithm and the two baseline algorithms. From Figs. 12 and 15, we can see that the total cost rises with the increase of

[2] Because VCG only works when the optimal selection is achieved. Obviously, random selection cannot achieve the minimum total cost.

[3] Note that these two baseline algorithms cannot guarantee the spatial coverage constraint and the temporal coverage constraint are satisfied.

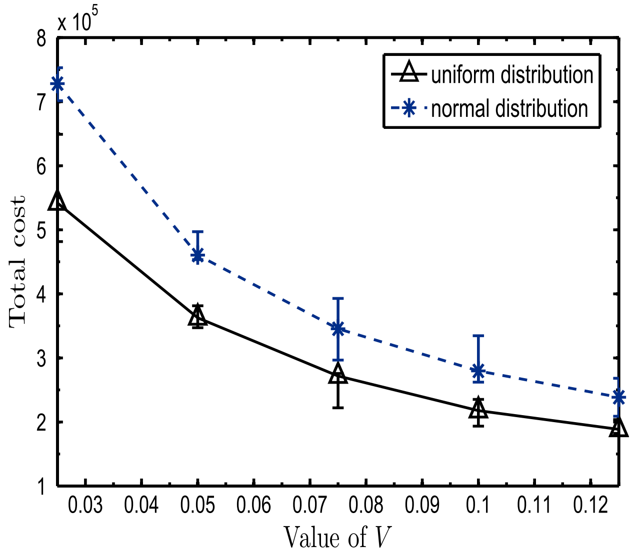


Fig. 9 Total cost of online algorithm vs. V

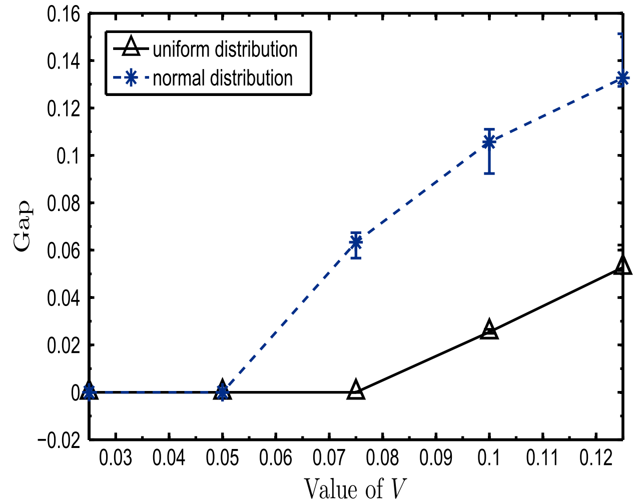


Fig. 11 Gap between coverage and δ_1 vs. V

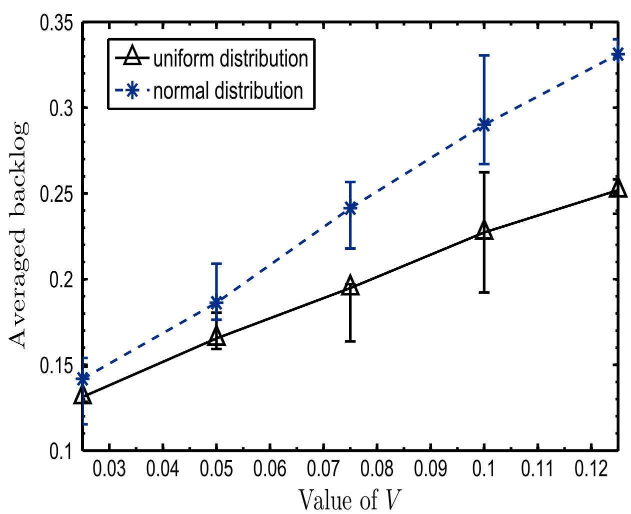


Fig. 10 Averaged backlog of virtual queues vs. V

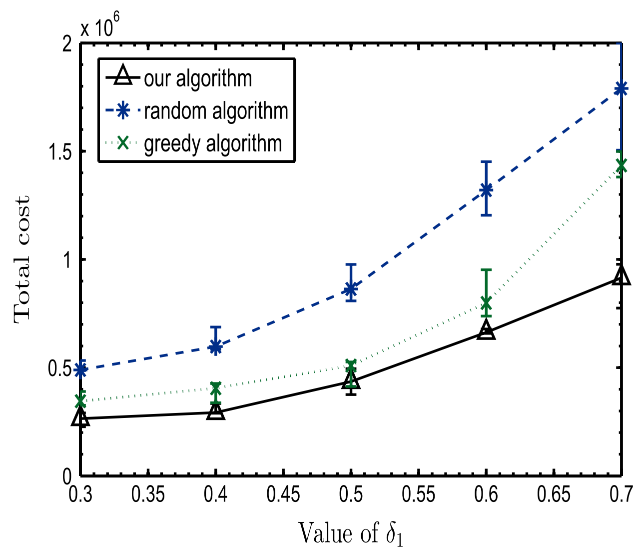


Fig. 12 Total cost vs. δ_1 under uniform distribution

δ_1, as more sensing data are collected. The results show that our online algorithm achieves the lowest total cost. When $\delta_1 = 0.7$, 91.9% and 44.8% costs are saved by our algorithm under the normal cost distribution, compared with random algorithm and greedy algorithm respectively.

5.3.2 Total payment

We evaluate the total payment achieved by our pricing policy integrating with Algorithm 1 and greedy algorithm. The total payment has the same variation trend with the total cost, when varying the value of δ_1. From Figs. 13 and 16, it is easy to find that the total payment achieved by our algorithm is lower than greedy algorithm, no matter the value of δ_1 and the cost distribution. When $\delta_1 = 0.6$, 51.9% and 47.7% payments are saved by our algorithm under the uniform and normal cost distributions, respectively.

5.3.3 Recovery accuracy

Averaged MAPE is computed to evaluate the accuracy of recovering missing data based on the collected data given by different grid selection algorithms. Figures 14 and 17 show our algorithm and random algorithm outperform greedy algorithm. This is because the grids with the lowest costs may concentrate in a small area, resulting in high recovery errors in the area without collected data. By integrating the evaluation results of total payment and recovery

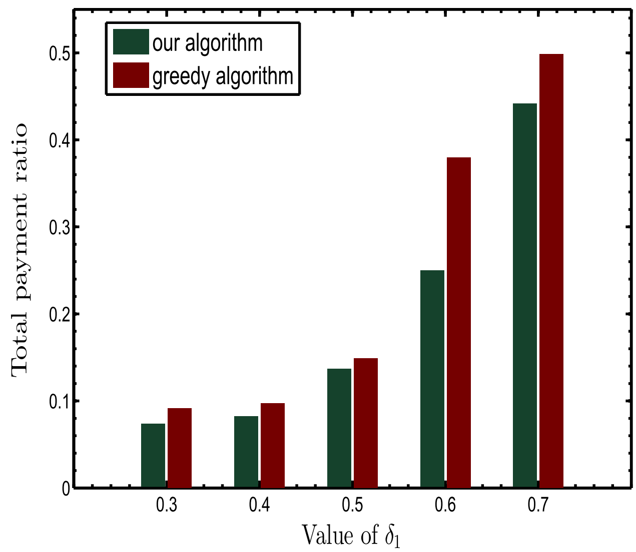


Fig. 13 Total payment vs. δ_1 under uniform distribution

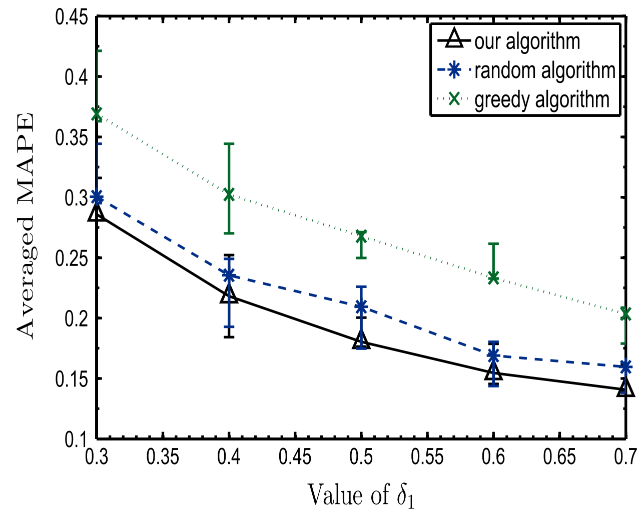


Fig. 14 Recovery error vs. δ_1 under uniform distribution

accuracy, we can conclude that our proposed approach achieves a high data recovery accuracy with a lower budget (Fig. 15, 16, 17).

6 Related work

6.1 Edge computing

Cloud computing (Fernand et al. 2013; Buyya et al. 2009) enables heavy computation jobs offloaded to remote cloud data centers. However, it suffers long latency for data transmission, processing and storage, as data are increasingly generated at the edge of the network. With the development of Internet of Things, edge computing (Shi et al. 2016; Luan et al. 2016) is becoming a novel computing paradigm, which allow computation performed at the edge of the network. Here, any computing and network resources along the path of data transmission between edge devices and the centralized cloud can be referred as "edge", such as smartphones and gateways. Previous work have proposed micro data centers (Greenberg et al. 2008; Cuervo et al. 2010), cloudlet (Satyanarayanan et al. 2009), and fog computing (Bonomi et al. 2012) which belong to the community of edge computing.

6.2 Task dispatching

Task dispatching problem is a major open issue in mobile edge computing, which has attracted a lot of research interests. The challenging of solving this problem is to decide whether a task is offloaded to the edge devices or the

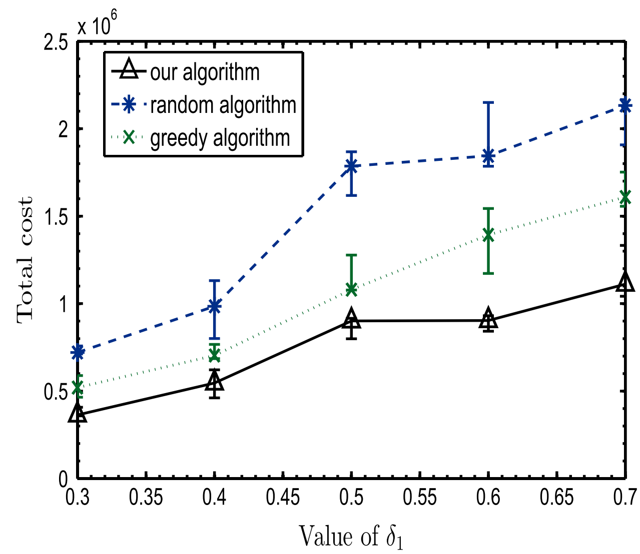


Fig. 15 Total cost vs. δ_1 under normal distribution

centralized cloud, by taking both energy consumption and network transmission into consideration. In Liu et al. (2016) and Tan et al. (2017), the stochastic nature of task arrivals are considered. A Markov decision precess approach is proposed for task scheduling to minimize the averaged delay of tasks under power constraints in Liu et al. (2016). A competitive algorithm is proposed to minimize the total weighted response delays by Tan et al. in Tan et al. (2017). Mao et al. consider energy harvesting technologies to power mobile edge devices in mode edge computing (Mao et al. 2016). A Lyapunov optimization-based dynamic computation offloading algorithm is proposed, which jointly determines the offloading decision, CPU-cycle frequencies and transmission power. Similarly, renewable energy harvesting mobile edge

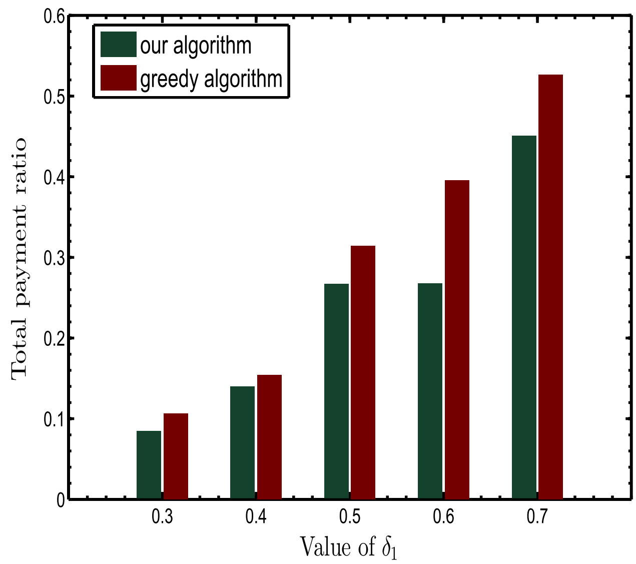


Fig. 16 Total payment vs. δ_1 under normal distribution

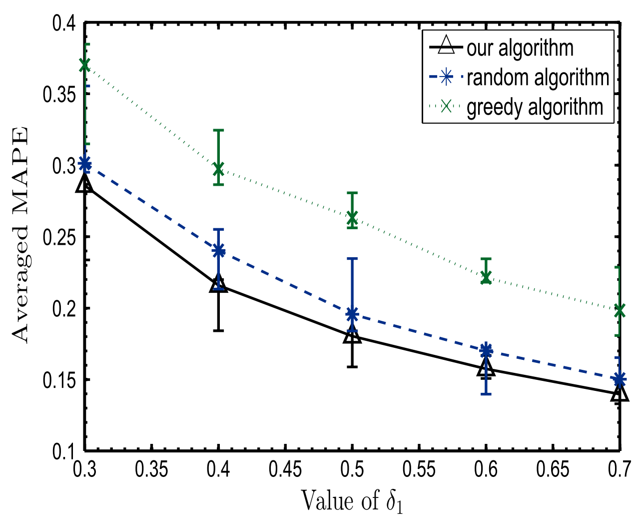


Fig. 17 Recovery error vs. δ_1 under normal distribution

computing is considered in Xu et al. (2017). In this work, a reinforcement learning-based resource management algorithm with considering the dynamics of renewable energy is provided, to minimize the long-term system cost. In Sun et al. (2017), propose an online algorithm for mobile users to decide which base station their tasks are offloaded to, with considering base stations may switch on or off randomly. Different from these works, we consider how to guarantee the quality of urban monitoring services provided by mobile edge computing, and propose an online task dispatching algorithm by taking the advantage of the existence of spatial and temporal correlations in sensing data.

6.3 Spatial–temporal data recovery

A few works (Xu et al. 2015a, b; Wang et al. 2015) have studied the spatial–temporal correlations among sensing data in urban monitoring applications. In Xu et al. (2015a, b), compressive sensing was applied, which first converts spatial–temporal sensing data into a sparse structure, then does sampling under the sparse structure and finally reconstructs the original data based on samples. Both Xu et al. (2015a) , (b) focus on solving the technical issues of applying compressive sensing. In Wang et al. (2015), a novel framework was proposed for compressive crowdsensing, where a minimum number of grids are selected for task allocation in each time slot to satisfy the quality requirement. Different from the previous works, we consider strategic smartphone users, who may misreport their cost information to the centralized cloud. Thus, a truthful pricing policy should be provided.

7 Conclusions

In this work, we have proposed a new approach for providing urban monitoring crowdsourcing services by mobile edge clouds, which leverages the spatial and temporal correlations existing in sensing data. This approach handles two coupled problem, i.e., determining the smartphones selected to collect sensing data and determining the amount of payments given to them. An online near-optimal algorithm and a truthful pricing policy are designed to solve the problems, respectively. Rigorous theoretical analysis demonstrates that our pricing policy guarantees the truthfulness of strategic smartphone users. Extensive simulation results show that our algorithm outperforms the baseline algorithms, achieving a higher quality of service with a lower budget.

Acknowledgements This research is supported in part by 973 Program (No. 2014CB340303), Shanghai Sailing Program 18YF1408200, and NSFC (Nos. 61772341, 61472254, 61572324, 61170238, and 61802245). This work is also supported by the Program for the Program for Changjiang Young Scholars in University of China, the Program for China Top Young Talents, and the Program for Shanghai Top Young Talents.

References

311 data. [Online]. http://nycopendata.socrata.com/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9

Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things, Edition of the Mcc Workshop on Mobile Cloud Computing, pp. 13–16 (2012)

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. Future Gen. Comput. Syst. **25**(6), 599–616 (2009)

Cuervo, E., et al., MAUI:making smartphones last longer with code offload. In: International Conference on Mobile Systems, Applications, and Services, pp. 49–62 (2010)

Dhillon I. S., Sra, S.: Generalized nonnegative matrix approximations with bregman divergences, Neural Information Proc Systems, pp. 283–290 (2006)

Fernand, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: a survey. Future Gen. Comput. Syst. **29**(1), 84–106 (2013)

Greenberg, A., Hamilton, J., Maltz, D.A., Patel, P.: The cost of a cloud:research problems in data center networks. Acm Sigcomm Comput. Commun. Rev. **39**(1), 68–73 (2008)

Liu, J., Mao, Y., Zhang, J., Letaief, K.B.: Delay-optimal computation task scheduling for mobile-edge computing systems. In: IEEE International Symposium on Information Theory, pp. 1451–1455 (2016)

Luan, T.H., Gao, L., Li, Z., Xiang, Y., Wei, G., Sun, L.: Fog computing: focusing on mobile users at the edge. Comput. Sci. (2016)

Mao, Y., Zhang, J., Letaief, K.B.: Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J. Select. Areas Commun. **34**(12), 3590–3605 (2016)

Mean absolute percentage error. [Online]. https://en.wikipedia.org/wiki/Mean_absolute_percentage_error

Mendez, D., Perez, A. J., Labrador, M. A., Marron, J. J.: P-sense: A participatory sensing system for air pollution monitoring and control. In: Proceedings of IEEE Percom Workshops'11, pp. 344–347 (2011)

Neely, M.J.: Stochastic network optimization with application to communication and queueing systems. Synth. Lect. Commun. Netw. **3**(1), 1–211 (2010)

Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory, vol. 1. Cambridge University Press, Cambridge (2007)

Rana, R.K., Chou, C.T., Kanhere, S.S., Bulusu, N., Hu, W.: Ear-phone: an end-to-end participatory urban noise mapping system. In: Proceedings of IEEE IPSN'10, pp. 105–116 (2010)

Satyanarayanan, M., Bahl, P., Davies, N.: The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009)

Shi, W., Cao, J., Zhang, Q., Zhang, Q., Li, Y., Xu, L.: Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)

Sun, Y., Zhou, S., Xu, J.: EMM: Energy-aware mobility management for mobile edge computing in ultra dense networks. IEEE J. Select. Areas Commun. PP(99), 1–1 (2017)

Tan, H., Han, Z., Li, X., Lau, F.C.M.: Online job dispatching and scheduling in edge-clouds. In: IEEE Conference on Computer Communications (INFOCOM), pp. 1–9, (2017)

Wang, L., Zhang, D., Pathak, A., Chen, C., Xiong, H., Yang, D., Wang, Y.: Ccs-ta: Quality-guaranteed online task allocation in compressive crowdsensing. In: Proceedings of ACM Ubicomp'15, pp. 683–694 (2015)

Waze app. https://www.waze.com/

World-wide smartphone users. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

Xu, J., Chen, L., Ren, S.: Online learning for offloading and autoscaling in energy harvesting mobile edge computing. IEEE Trans. Cognit. Commun. Netw. **PP**(99) (2017)

Xu, L., Hao, X., Lane, N. D., Liu, X., Moscibroda, T.: Cost-aware compressive sensing for networked sensing systems. In: Proceedings of IEEE IPSN'15, pp. 130–141 (2015a)

Xu, L., Hao, X., Lane, N. D., Liu, X., Moscibroda, T.: More with less: lowering user burden in mobile crowdsourcing through compressive sensing. In: Proceedings of ACM Ubicomp'15, pp. 659–670 (2015b)

Zheng ,Y., Liu, F., Hsieh, H.-P.: U-air: when urban air quality inference meets big data. In: Proceedings of ACM KDD'13, pp. 1436–1444 (2013)