8-7-2018

Data Collection and Aggregation in Mobile Sensing

Ji Li

DATA COLLECTION AND AGGREGATION IN MOBILE SENSING

by

JI LI

Under the Direction of Zhipeng Cai, PhD

ABSTRACT

Nowadays, smartphones have become ubiquitous and are playing a critical role in key aspects of people's daily life such as communication, entertainment and social activities. Most smartphones are equipped with multiple embedded sensors such as GPS (Global Positioning System), accelerometer, camera, etc, and have diverse sensing capacity. Moreover, the emergence of wearable devices also enhances the sensing capabilities of smartphones since most wearable devices can exchange sensory data with smartphones via network interfaces. Therefore, mobile sensing have led to numerous innovative applications in various fields including environmental monitoring, transportation, healthcare, safety and so on. While all these applications are based on two critical techniques in mobile sensing, which are data collection and data aggregation, respectively. Data collection is to collect all the sensory data in the network while data aggregation is any process in which information is gathered and expressed in a summary form such as SUM or AVERAGE. Obviously, the above two problems can be solved by simply collect all the sensory data in the whole network. But that will lead to huge communication cost.

This dissertation is to reduce the huge communication cost in data collection and data aggregation in mobile sensing where the following two technical routes are applied. The first technical route is to use sampling techniques such as uniform sampling or Bernoulli sampling. In this way, an aggregation result with acceptable error can be can be calculate while only a small part of mobile phones need to submit their sensory data. The second technical rout is location-based sensing in which every mobile phone submits its geographical position and the mobile sensing platform will use the submitted positions to filter useless sensory data. The experiment results indicate the proposed methods have high performance.

INDEX WORDS: Data aggregation, Data Collection, Sampling, Smartphone, Crowdsensing, Auction

DATA COLLECTION AND AGGREGATION IN MOBILE SENSING

by

JI LI

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

DATA COLLECTION AND AGGREGATION IN MOBILE SENSING

by

JI LI

DEDICATION

This dissertation is dedicated to my family.

ACKNOWLEDGEMENTS

I would never have been able to finish my dissertation without the guidance of my committee members, help from my group, and support from my family and my friends. Therefore, I would like to show my gratitude to them.

I would like to express my deepest gratitude to my advisor Dr. Zhipeng Cai for his excellent inspiration, guidance, patience, and supporting me for doing research. Dr. Cai always gave me the greatest support during my PHD studies, he patiently inspired me and financially supported my research.

I would like to thank my committee members, Dr. Anu Bourgeois, Dr. Wei Li and Dr. Kai Zhao for their great support for my PHD study. They also kindly provided me some teaching suggestions.

I also would like to thank all the professors and staffs in our department, especially Dr. Yingshu Li, Dr. Yi Pan and Ms. Tammie Dudley for their patient help makes my life easier.

I would like to thank my group members, such as Dr. Mingyuan Yan and Dr. Meng Han, who provided me help and happiness on my research.

Last but not least, I would like to thank all my family and friends for their support, understanding, and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- ADDF-Estimated Annual Average Daily Flows
- CPS-Cyber Physical System
- GPS-Global Positioning System
- IoT-Internet of Things
- MANET-Mobile Ad hoc Network
- NP-Non-deterministic Polynomial-time
- PTAS-Polynomial-Time Approximation Scheme
- VCG-VickreyCClarkeCGroves
- WSN-Wireless Sensor Network

Chapter 1

INTRODUCTION

1.1 Background and Motivations

Nowadays, smartphones have become ubiquitous and are playing a critical role in key aspects of people's daily life such as communication, entertainment and social activities. Most smartphones are equipped with multiple embedded sensors such as GPS (Global Positioning System), accelerometer, camera, etc, and have diverse sensing capacity. Moreover, the emergence of wearable devices also enhances the sensing capabilities of smartphones since most wearable devices can exchange sensory data with smartphones via network interfaces. For example, the sensors in a mobile phones in shown in Fig. 1.1.

Therefore, mobile sensing have led to numerous innovative applications in various fields including environmental monitoring, transportation, healthcare, safety and so on [1]. While all these applications are based on two critical techniques in mobile sensing, which are data collection and data aggregation, respectively. Data collection is to collect all the sensory data in the network while data aggregation is any process in which information is gathered and expressed in a summary form such as SUM or AVERAGE. For example, if the sensory dataset is $S = s_1, s_2, s_3$ and we have s_1=1, s_2=1, s_1=3. The data collection is to collect the whole sensory dataset from all the users. The final result is $S = s_1, s_2, s_3$ The data aggregation is to get a aggregation result. For final result for SUM operation is 5 while the final result for the AVERAGE operation is $\frac{5}{3}$.

Obviously, the above two problems can be solved by simply collect all the sensory data in the whole network. After this process, the data aggregation is finished and all kinds of data aggregation operations can be easily executed since all the sensory data are gathered. However, the above method has a very obvious drawback. In the above method, all the smartphones need to submit their sensory data and the size of users may be quite large in

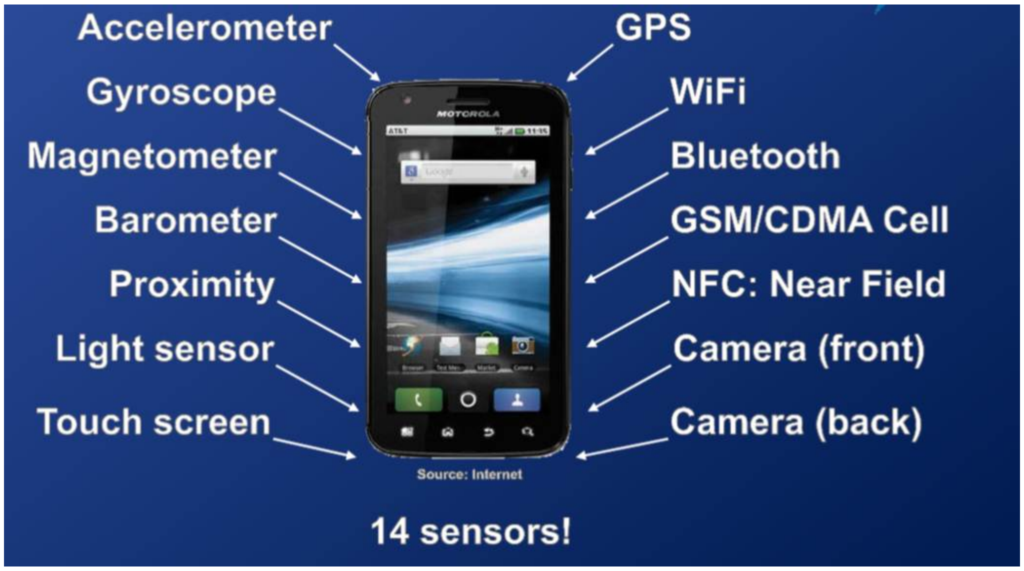


Figure 1.1. Sensors in smartphones

practice [2], [3]. While on the other hand, data transmission for mobile phones may cost a great deal of energy. For example, according to [4], the energy cost for transmitting one bit of data is enough for executing 1000 instructions. Therefore, it is a critical problem to reduce the energy cost in mobile sensing.

This dissertation is to reduce the huge communication cost in data collection and data aggregation in mobile sensing, where the following two technical routes are applied.

Technical Route 1: Sampling The first technical route is to use sampling techniques such as uniform sampling or Bernoulli sampling, whose process is as follows.

1. Determine a proper sampling size or a proper sampling probability.

2. Broadcast the sampling size or the sampling probability and sample the sensory data in the network.

3. Submit sensory data and do partial data aggregation.

In this way, an aggregation result with acceptable error can be can be calculated while only a small part of mobile phones need to submit their sensory data. Since only a small part of mobile phones needs to submit their sensory data, the total energy cost will be reduced.

Technical Route 2: Location-based Sensing The second technical route is location-based sensing whose basic idea is as follows.

1. The mobile sensing platform broadcasts the sensing task.
2. Each user submits his/her position to the mobile sensing platform.
3. The mobile crowdsensing platform decides the winner set according to the submitted positions to filter out useless sensory data.
4. All the selected smartphones submit their sensory data to the mobile sensing platform.

Since only the selected mobile phones need to submit their sensory data, the total energy cost will also be reduced.

Based on the above two technical routes, the following problems are proposed and solved, which are listed as follows.

1. Approximate holistic aggregation in mobile sensing.
2. Bernoulli sampling based approximate holistic aggregation in mobile sensing.
3. Data collection in geographical position conflicting mobile sensing.
4. Approximate holistic aggregation in small-scale networks.
5. Data collection in geographical position dependent mobile sensing.

The above problems are briefly introduced in the following three sections. For the detailed information, please refer to Chapter 3, 4, 5, 6 and 7.

1.2 Approximate Holistic Aggregation in Mobile Sensing

Holistic aggregations are popular queries for users to obtain detailed summary information in mobile sensing. An aggregation operation is holistic if there is no constant bound on the size of the storage needed to describe a sub-aggregation. Since holistic aggregation cannot be distributable, it requires that all the sensory data should be sent to the sink in order to obtain the exact holistic aggregation results, which costs lots of energy. However, in most applications, exact holistic aggregation results are not necessary, instead, approximate results are acceptable. To save energy as much as possible, we study the approximated holistic aggregation algorithms based on uniform sampling. In this problem, four holistic aggregation operations, frequency, distinct-count, rank and quantile, are investigated. The mathematical methods to construct their estimators and determine optional sample size are proposed, and the correctness of these methods are proved. Four corresponding distributed holistic algorithms to derive (ϵ, δ)-approximate aggregation results are given. The solid theoretical analysis and extensive simulation results show that all the proposed algorithms have high performance on the aspects of accuracy and energy consumption.

1.3 Bernoulli Sampling Based Approximate Holistic Aggregation in Mobile Sensing

A frequency query is to acquire the occurrence frequency of each value in a sensory data set, which is a popular operation in mobile sensing. However, exact frequency results are not easy to obtain due to the unique characteristics of mobile phone networks. Fortunately, approximate frequency results are acceptable and affordable in most mobile sensing applications. In this problem, we study how to process approximate ordinary frequency queries, approximate single value frequency queries and approximate range frequency queries, and propose Bernoulli sampling based method to estimate approximate frequencies. The distributed algorithms to calculate approximate frequency results are introduced. The simulation results show that on the aspects of both energy efficiency and accuracy, the proposed algorithms have high performance.

1.4 Data Collection in Geographical Position Conflicting Mobile Sensing

Sensor-embedded smartphones have become ubiquitous nowadays, further leveraging the popularity of mobile crowdsensing. A mobile crowdsensing platform gathers sensory data from smartphone users and makes payments to them in return. Due to the spatial correlation of sensory data in various applications, users close to each other in geographical positions usually provide similar sensory data, and it is quite an economic waste for a mobile sensing platform to buy duplicated sensory data with multiple payments to geographically close users. Unfortunately, the existing works do not take this matter into consideration. To prevent waste, the third problem considers geographical position conflicting mobile crowdsensing systems in which any two users within a limited geographical distance cannot obtain payments simultaneously while participating in crowdsensing tasks. Two algorithms are proposed to select appropriate mobile crowdsensing participants and calculate the payments to them. Solid theoretical proofs are presented to demonstrate the beneficial properties of our proposed algorithms. The extensive experiment results based on real-world datasets indicate that our proposed algorithms are efficient while providing beneficial properties.

1.5 Approximate Holistic Aggregation in Small Scale Networks

With the ever-increasing population, problems of everyday sustainability have become onerous. According to the survey by United Nations, 54% of the world's current population lives in urban areas. So, to collect such well spread data, they exploit various sensor equipped devices in the city to collect data and interpret information at the city level. In today's world, all the devices from smart home devices to intelligent transport systems are well connected with the internet. Such a network with well-connected devices is called Internet-Of-Things (IoT)

Sensors are the building blocks for many IoT devices. Utilizing sensors as the communication media helps us resolve many of the problems discussed earlier. The fourth problem is to propose two algorithms to process δ-approximate maximum queries and δ-distinct-set queries in small-scale networks. These two algorithms are based on uniform sampling and Bernoulli sampling, respectively. Proposed algorithms to return the exact query results with probability not less than 1-δ where the value of δ can be arbitrarily small.

1.6 Data Collection in Geographical Position Dependent Mobile Sensing

In real-life applications, the mobile crowdsensing platforms aim to maximize the quality of gathered sensory data during the above process because the mobile crowdsensing platform needs to pay significant economic costs to the mobile crowdsensing participants for their sensory data. However, in many applications, the sensory data exhibits strong spatial correlation. Directly selecting the winners with lower bids may select multiple disjoint users as the winners, which cannot fully describe sensory data's spatial correlation or show the whole circumstance.

The fifth problem is to study incentive mechanisms in the geographical position dependent mobile crowdsensing platforms. In this problem, the mobile crowdsensing platform cannot select multiple disjoint users as the winners. All the selected winners must form a connected dependent graph so that the sensory data can fully describe the whole circumstance of the sensory region.

1.7 Organization

The rest of this dissertation proposal is organized as follows: Chapter 2 summarized the related literature. Chapter 3 is about approximate holistic aggregation in mobile sensing. Chapter 4 studies Bernoulli sampling based approximate holistic aggregation in mobile sensing. Chapter 5 solves the problem of data collection in geographical position conflicting mobile sensing. Chapter 6 is about approximate holistic aggregation in small networks Chapter 7 solves the problem of data collection in geographical position dependent mobile sensing. Chapter 8 is the conclusion.

Chapter 2

RELATED WORK

2.1 Sampling-based Data Aggregation

Since the summary information contained by distributable, algebraic and holistic aggregations are quite important for analysis in networks, many related works have been proposed.

Considering that the sensory data are highly correlated in spatial and temporal dimensions, the work in [5] presents a distributed approximate aggregation algorithm, which brings considerable energy savings. By partitioning the precision constraint of data aggregation and allocating error bounds to individual sensor nodes, the work in [6] discusses the tradeoff between energy consumption and data quality in order to prolong network lifetime. The experimental results show that the proposed scheme significantly improves network lifetime. However, the algorithms in both [5] and [6] are only designed for continuous queries. Therefore, they cannot be used for snapshot queries discussed in our dissertation.

The work in [7] proposes an approximate algorithm for the quantile operation in mobile sensing. This algorithm reduces energy consumption based on the sampling technique. However, since the variance of the estimator in [7] is larger than 0, the probability of obtaining a result with the required approximation error is fixed according to Chebyshev's inequality, so that it cannot meet the arbitrary precision requirement given by users. The work in [8] proposes a data aggregation scheme that can be extended to process a class of queries, including the approximate quantile queries. Strict theoretical guarantees of approximation quality are provided in [8]. However, the error bound is related to the memory size m, which is limited in practice. Therefore, the error bound cannot be arbitrarily adjusted. The work in [9] proposes a protocol to compute approximate median and designs an algorithm to obtain the median in the presence of compromised nodes. The analysis and simulation results show that the proposed algorithm is scalable and efficient. However, it is only suitable to deal with median queries and cannot calculate any quantile required by users, and it also needs additional storage space to decrease the error bound.

Based on a binary trie based summary structure for representing transaction sets, the work in [10] proposes two algorithms to process frequency queries towards large-scale databases. The work in [11] proposes an algorithm for computing the approximate distinct-count. These algorithms are based on the sampling synopsis. However, they are centralized and cannot be used in large scale WSNs.

The work in [12] proposes an adaptive sampling solution named SILENCE in CPSs. SILENCE is efficient to reduce redundancy in raw data without compromising accuracy of reconstruction of the phenomenon at the sink. However, it is only suitable for the circumstance that sensory data have spatial and temporal correlations. The algorithm in [13] can adjust the sampling rate adaptively during the monitoring period. However, the theoretical bound of monitoring accuracy cannot be guaranteed.

The works in [14] and [15] propose two algorithms for frequency queries in large-scale database systems. However, these algorithms are centralized and cannot be employed in MANETs.

2.2 Mobile Crowdsensing

The work in [16] designs a reverse auction based dynamic price incentive mechanism where users can sell sensory data to the crowdsensing platform at users' claimed bid prices. However, the incentive mechanism in [16] does not satisfy the property of truthfulness.

The work in [17] studies the distributed time-sensitive and location-dependent task s-election problem in mobile crowdsensing. An asynchronous and distributed task selection algorithm is proposed. But [17] does not take the geographical position conflicting into consideration. The work in [18] is about photo crowdsensing in disruption tolerant net-works. However, this work focuses on users' graphical position-based cooperation instead of graphical position-based confliction. Therefore, it cannot solve our problem directly.

Massive works use auction-based mechanisms to study the problem of graphical position conflicting spectrum allocation [19], [20]. However, these works are for allocating fixed number of spectrums rather than maximizing the total social welfare in crowdsensing data gathering. Moreover, these works' definition on graphical position confliction also differs with that of our problem. Therefore, these works cannot be used for our problem directly.

2.3 Other Related Work

The random geometric graph introduced in [21] is an excellent theoretical model to be used in network connectivity analysis. But it is only suitable for static networks. The work in [22] proposes a derived version of the random geometric graph named dynamic random geometric graph which can be used to analyze MANET connectivity. But it only studies the period for a network to be connected or disconnected. The work in [23] studies the connectivity of MANETs by simulations. The simulation result is the probability of a network to be connected. But our work focuses on the number of the connected nodes. The work in [24] is like an evaluation of MANET connectivity. The simulation results for the size of the largest connected component are shown. But it does not specify how to calculate the size of the largest connected component. Furthermore, the simulation results shown in [24] are the average size. This kind of results cannot be applied to our work since it is possible that the actual size is smaller than the average size.

The spatial correlation of sensory data is well studied in [25] and [26]. The authors in [27] proposed a model about users' sensing area in photo crowdsensing. The factor of both sensing range and sensing angle are taken into consideration in this model. Massive spatial correlated sensory data can be found in the Greenorbs Project [28].

The work in [29] uses the Brightkite and Gowalla datasets to study friendships in online social networks and users' movements in the physical world. The works in [30] and [31] use subsets of the Brightkite and Gowalla datasets to study the influence maximization problem in location based social networks. The work in [32] uses the Swoopo dataset to study information asymmetries in pay-per-bid auctions, which indicates bid distribution closely maps with a widely adopted assumption in optimal auction design research [33].

Chapter 3

APPROXIMATE HOLISTIC AGGREGATION IN MOBILE SENSING

3.1 Introduction

Mobile sensing have been widely employed in many applications, including military defense, environment monitoring, traffic monitoring, health care, structural health monitoring and so on [34] [35] [36] [37] [38] [39] [40]. Since sensors are always redundantly deployed in a monitored region to improve network robustness, the sensory data generated are always very huge and redundant to users [41] [42] [43] [44] [45]. To deal with such a great amount of data, the summary information returned by aggregation operations is extremely important for users to make decisions.

As we know, the existing aggregation operations can be classified into three categories, distributable aggregations (such as SUM and COUNT), algebraic aggregations (such as AVERAGE), and the holistic aggregations. Among these aggregations, the holistic aggregations are the most complicated ones to process since there is no constant bound on the size of the storage needed to describe a sub-aggregation [46] and they cannot be processed in a distributed manner [47]. However, the summary information returned by many holistic aggregations, such as the quantile and frequency queries, are quite valuable in applications. For example, the frequency result of air pollution not only reflects the overall pollution situation in the monitored region, but also shows the detailed distribution of each pollution grade. Such information cannot be provided by the SUM or AVERAGE aggregation results. Therefore, it is important to deal with the holistic aggregation efficiently in mobile sensing.

Since a holistic aggregation cannot be distributable, it requires that all sensory data to be transmitted to the sink to derive the exact holistic aggregation results, which costs high energy consumption. In practice, exact holistic aggregation results are not necessary in most applications and approximate ones are also acceptable [48] [49] [50] [51]. Therefore, some approximate aggregation techniques were proposed, such as [52] and [53]. These methods save lots of energy comparing with the exact holistic aggregation algorithms. However, all of them have a fixed error bound and cannot satisfy the arbitrary precision requirement specified by users. For example, the work in [53] focuses on continuous holistic queries in wireless sensor networks. An approach to derive approximate results for rapid data changing circumstances is proposed, which significantly reduces the traffic cost. However, the error of the approximate result cannot be arbitrarily specified by users.

To overcome such problems, this chapter take *frequency*, *distinct-count*, *rank* and *quantile* as example aggregation operations to investigate the sampling based holistic aggregation algorithms in this chapter. Our aim is to return an (ϵ, δ)-approximate holistic aggregation result to users, which satisfies that the probability of the relative error bound of the results being larger than ϵ is smaller than δ, where ϵ and δ can be arbitrarily small.

Although the works in [54] and [55] propose (ϵ, δ)-approximate aggregation algorithms based on the uniform and Bernoulli sampling techniques, they can only deal with distributable and the algebraic aggregations, and are not suitable for our problems. In summary, the contributions of this chapter are as follows.

1. Four mathematical estimators for the *frequency*, *distinct-count*, *rank* and *quantile* aggregation operations are provided, and the unbiased properties of these estimators are proved.

2. The mathematical methods to determine the optional sample size for calculating (ϵ, δ)-frequency, (ϵ, δ)-distinct-count, (ϵ, δ)-rank and (ϵ, δ)-quantile results are designed, and their correctness are guaranteed.

3. The distributed algorithms for (ϵ, δ)-frequency, (ϵ, δ)-distinct-count, (ϵ, δ)-rank and (ϵ, δ)-quantile are provided based on the uniform sampling technique, and the computation complexities and energy costs of these algorithms are analyzed.

4. The extensive simulation results are presented to verify that all the proposed algorithms have high performance on the aspects of accuracy and energy consumption.

The rest of this chapter is organized as follows. Section 3.2 presents the problem definition. Section 3.3 provides the mathematical foundations of the (ϵ, δ)-approximate aggregation algorithms. Section 3.4 explains the (ϵ, δ)-approximate aggregation algorithms. Section 3.5 shows the simulation results. Section 3.6 concludes this part.

3.2 Problem Definition

Let n denote the size of network and i $(1 \leq i \leq n)$ be the ID of a sensor node. We divide a network into grids, and the sensor nodes in the same grid form a cluster, where the edge length of each grid is determined by the transmission radius of a sensor node so that it guarantees that the sensor nodes in the same cluster can communicate with each other by one-hop communication.

Let s_{ti} be the sensory value of node i at time t. Thus $S_t = \{s_{t1}, s_{t2}, \ldots, s_{tn}\}$ is the set of all the sensory data in the network at time t. In practice, there always exist some reduplicated values in S_t. We use $Dis(S_t)$ to denote the distinct set of S_t, which means that $Dis(S_t) \subseteq S_t$ and $Dis(S_t)$ only contains all the distinct values in S_t. We assume all the data distributes randomly in the network. The spatial and temporal correlation for the sensory data is ignored. Four aggregation operations on S_t are studied in this chapter, which are *frequency*, *rank*, *distinct count* and *quantile*. The definition of these four operations are as follows.

1. For any $x \in Dis(S_t)$, the exact frequency denoted by $F(S_t, x)$ satisfies $F(S_t, x) = \frac{|\{s_{tj}|s_{tj}=x \wedge 1 \leq j \leq n\}|}{n}$.

2. The exact rank denoted by $R(S_t, x)$ satisfies $R(S_t, x) = \frac{|\{s_{tj}|s_{tj} \leq x \wedge 1 \leq j \leq n\}|}{n}$.

3. The exact distinct-count of S_t denoted by $DC(S_t)$ satisfies that $DC(S_t) = |Dis(S_t)|$.

4. For any given r $(0 \leq r \leq 1)$, the exact quantile denoted by $Q(S_t, r)$ satisfies $Q(S_t, r) = argmin_{s_{ti}} \frac{|\{s_{tj}|s_{tj} \leq s_{ti} \wedge 1 \leq j \leq n\}|}{n} \geq r$.

All the above four holistic aggregations are quite useful for mobile sensing. For example, in the application of pollution monitoring, the results returned by frequent queries not only reflect the overall pollution situation in the monitored area, but also provide the detailed distribution of each pollution grade. Similarly, the results returned by the distinct-count queries indicate the number of pollution levels in the monitored area, which could help the users to determine the actions to be carried out. Furthermore, the results of the rank queries reflect the ratio the of over-polluted area, and the results of the quantile queries show the pollution degree under a given ratio. Such information is important for users to make prompt and proper decisions. Obviously, comparing with sum, max/min and average aggregations, the above four holistic aggregation queries present more detailed information, and provide more evidences for users to take actions. Therefore, it is quite important to design distributed and energy-efficient algorithms for these four holistic aggregation queries.

As mentioned in Section 3.1, it leads to a huge communication cost and computation cost for calculating an exact aggregation result. Therefore, we study how to obtain an (ϵ, δ)-approximate result for the above four aggregation operations. Suppose that I_t denotes an exact aggregation result of S_t at time t, and $\widehat{I_t}$ is an approximate result to estimate I_t. The relative error between I_t and $\widehat{I_t}$, denoted by $Error(I_t, \widehat{I_t})$, satisfies that $Error(I_t, \widehat{I_t}) = \frac{|I_t - \widehat{I_t}|}{I_t}$. The definition of the (ϵ, δ)-estimator is given as follows, which has been proposed in [54].

Definition 3.2.1 ((ϵ, δ)-estimator)**.** *For any ϵ $(\epsilon > 0)$ and δ $(0 \leq \delta \leq 1)$, $\widehat{I_t}$ is called the (ϵ, δ)-estimator of I_t if $Pr(Error(I_t, \widehat{I_t}) \geq \epsilon) \leq \delta$, where $Error(I_t, \widehat{I_t}) = \frac{|I_t - \widehat{I_t}|}{I_t}$.*

According to Definition 3.2.1, the problem of computing (ϵ, δ)-approximate frequency, (ϵ, δ)-approximate rank, (ϵ, δ)-approximate distinct-count and (ϵ, δ)-approximate quantile is defined as follows.

Input: (1) A network with n nodes; (2) The sensory data set S_t; (3) Aggregation operator $Agg \in \{Frequency, Rank, DistinctCount, Quantile\}$; (4) ϵ $(\epsilon > 0), \delta$ $(0 \leq \delta \leq 1)$, x (only for *rank*) and r (only for *quantile*).

Output: (ϵ, δ)-approximate aggregation result of Agg.

For the clarity, all the frequently used symbols are summarized in Table 3.1.

Name	Description
n	The size of a network
i $(1 \le i \le n)$	The ID of a sensor node
s_{ti}	The sensory value of node i at time t
$S_t = \{s_{t1}, s_{t2}, \ldots, s_{tn}\}$	The set of all the sensory data in the network at time t
$Dis(S_t)$	The distinct set of S_t
$F(S_t, x)$	The exact frequency of value x for dataset S_t
$R(S_t, x)$	The exact rank of value x for dataset S_t
$DC(S_t)$	The exact distinct-count of S_t
$Q(S_t, r)$	The exact quantile of any given r for dataset S_t
$\widehat{I_t}$	Approximate result to estimate I_t
$Error(I_t, \widehat{I_t})$	Relative error between I_t and $\widehat{I_t}$
$U(m) = \{X_1, X_2, ..., X_m\}$	A uniform sample of S_t with sample size m
n_{min}/n_{max}	The number of appearances for the least/most appearing data
f_i	The frequency of the i-th smallest sensory data

Table 3.1. Symbols

3.3 Preliminaries

According to [56], uniform sampling is the best choice for estimating aggregation results. Therefore, our (ϵ, δ)-approximate aggregation algorithm is based on the uniform sampling technique.

Let $X_1, X_2, ..., X_m$ denote m simple random samplings with replacement from S_t, that is, X_i and X_j are independent with each other for all $1 \le i \ne j \le m$, and $\Pr(X_i = s_{tj}) = \frac{1}{n}$ for any $1 \le i \le m$, $1 \le j \le n$, where $\Pr(X_i = s_{tj})$ denotes the probability of s_{tj} being sampled by the i-th simple random sampling. Thus, $U(m) = \{X_1, X_2, ..., X_m\}$ is called a uniform sample of S_t with size m. The preliminaries of computing (ϵ, δ)-approximate frequency, (ϵ, δ)-approximate rank, (ϵ, δ)-approximate distinct-count and (ϵ, δ)-approximate quantile are presented in the following subsections respectively.

3.3.1 (ϵ, δ)-Approximate Frequency

To obtain (ϵ, δ)-approximate frequency, the mathematical estimator of a frequency is needed firstly. Let $\widehat{F(S_t, x)}$ denote the estimator of exact value $F(S_t, x)$. Then $\widehat{F(S_t, x)}$

should satisfy that

$$\widehat{F(S_t, x)} = \frac{|\{X_i | X_i = x \wedge 1 \leq i \leq m\}|}{m}.$$

The following Theorem 3.3.1 indicates that $\widehat{F(S_t, x)}$ is an unbiased estimator of $F(S_t, x)$, and the variance of $\widehat{F(S_t, x)}$ is provided by Theorem 3.3.2.

Theorem 3.3.1. *$\widehat{F(S_t, x)}$ is an unbiased estimator of $F(S_t, x)$.*

Proof. Let $I(a, b)$ be a variable which satisfies

$$I(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases}$$

Then we have

$$E[\widehat{F(S_t, x)}] = \frac{\sum_{i=1}^{m} I(X_i, x)}{m} \tag{3.1}$$

Since $X_i \in U(m)$, $\Pr(X_i = s_{tj}) = \frac{1}{n}$ for any $1 \leq i \leq m$, and thus

$$E[I(X_1, x)] = E[I(X_2, x)] = \cdots = E[I(X_m, x)] = \sum_{i=1}^{n} \Pr(X_1 = s_{ti}) I(s_{ti}, x) \tag{3.2}$$

$$= \sum_{i=1}^{n} \frac{1}{n} I(s_{ti}, x) = \frac{1}{n} \sum_{i=1}^{n} I(s_{ti}, x) = F(S_t, x)$$

Based on the formulas (3.1) and (3.2), we have $E[\widehat{F(S_t, x)}] = F(S_t, x)$, so that $\widehat{F(S_t, x)}$ is an unbiased estimator of $F(S_t, x)$. □

Theorem 3.3.2. $Var[\widehat{F(S_t, x)}] = \frac{F(S_t, x)(1 - F(S_t, x))}{m}$.

Proof. According to the proof of Theorem 3.3.1, we have $\Pr(X_i = s_{tj}) = \frac{1}{n}$ for any $X_i \in U(m)$. Therefore, $I(X_i, x)$ can be regarded as a 0-1 random variable, where $\Pr\{I^2(X_i, x) = 1\} = \Pr\{I(X_i, x) = 1\} = F(S_t, x)$. Thus,

$$Var[I(X_i, x)] = E(I^2(X_i, x)) - (E(I(X_i, x)))^2 = F(S_t, x)(1 - F(S_t, x))$$

for any $1 \leq i \leq m$.

Thus, $Var[\widehat{F(S_t,x)}] = \frac{Var[\sum_{i=1}^{m} I(X_i,x)]}{m^2} = \frac{Var[I(X_1,x)]}{m} = \frac{F(S_t,x)(1-F(S_t,x))}{m}$. □

Since $\widehat{F(S_t,x)}$ is an unbiased estimator of an exact frequency and its variance is bounded, $Error(F(S_t,x), \widehat{F(S_t,x)})$ can be arbitrarily small.

Obviously, it is critical to determine a proper sample size for a sampling based algorithm. The following theorem shows that $\widehat{F(S_t,x)}$ is an (ϵ,δ)-estimator of $F(S_t,x)$ if the sample size m satisfies that $m \geq \frac{\phi_{\delta/2}^2}{\epsilon^2}(\frac{n}{n_{min}} - 1)$.

Theorem 3.3.3. *$\widehat{F(S_t,x)}$ is an (ϵ,δ)-estimator of $F(S_t,x)$ if $m \geq \frac{\phi_{\delta/2}^2}{\epsilon^2}(\frac{n}{n_{min}} - 1)$, where $\phi_{\delta/2}$ is the $\frac{\delta}{2}$ fractile of the standard normal distribution and n_{min} is the number of appearances for the least appearing data.*

Proof. According to the center limit theory [57], we have $\widehat{F(S_t,x)} \sim N(E[\widehat{F(S_t,x)}], Var[\widehat{F(S_t,x)}])$ if the sample size $m \geq 30$, *i.e.*, $\frac{\widehat{F(S_t,x)}-F(S_t,x)}{\sqrt{Var[\widehat{F(S_t,x)}]}} \sim N(0,1)$ when $m \geq 30$, where $N(E,V)$ denotes a normal distribution with expectation E and variance V for any $E \in (-\infty,+\infty)$ and $V \in [0,+\infty)$.

Since the required sample size is far more than 30 for most cases due to the large scale of a network, we can assume that $\frac{\widehat{F(S_t,x)}-F(S_t,x)}{\sqrt{Var[\widehat{F(S_t,x)}]}}$ follows the standard normal distribution in the rest of the chapter. Meanwhile, we also have $Pr(\left|\frac{\widehat{F(S_t,x)}-F(S_t,x)}{\sqrt{Var[\widehat{F(S_t,x)}]}}\right| \geq \phi_{\delta/2}) = \delta$ since $\phi_{\delta/2}$ denotes the $\frac{\delta}{2}$ fractile of the standard normal distribution, that is

$$Pr(|\widehat{F(S_t,x)} - F(S_t,x)| \geq \phi_{\delta/2}\sqrt{Var[\widehat{F(S_t,x)}]}) = \delta. \tag{3.3}$$

Based on the condition of Theorem 3.3.3, we have

$$m \geq \frac{\phi_{\delta/2}^2}{\epsilon^2}(\frac{n}{n_{min}} - 1) \geq \frac{\phi_{\delta/2}^2}{\epsilon^2}(\frac{1}{F(S_t,x)} - 1) \tag{3.4}$$

Furthermore,

$$Var[\widehat{F(S_t,x)}] = \frac{F(S_t,x)(1-F(S_t,x))}{m} \tag{3.5}$$

according to the conclusion of Theorem 3.3.2.

Thus, according to Formulas (3.3), (3.4) and (3.5), we have $Pr(|\widehat{F(S_t, x)} - F(S_t, x)| \geq \epsilon F(S_t, x)) \leq \delta$, *i.e.*, $\widehat{F(S_t, x)}$ is an (ϵ, δ)-estimator of $F(S_t, x)$ based on Definition 3.2.1. $\square$

3.3.2 (ϵ, δ)-Approximate Rank

The mathematical estimator of rank is defined as follows

$$\widehat{R(S_t, x)} = \frac{|\{X_i | X_i \leq x \wedge 1 \leq i \leq m\}|}{m}.$$

Using the similar proofs of Theorem 3.3.1 and Theorem 3.3.2, we have that $\widehat{R(S_t, x)}$ is an unbiased estimator of the exact value, and $Var[\widehat{R(S_t, x)}] = \frac{R(S_t,x)(1-R(S_t,x))}{m}$. Meanwhile, we also have the following theorem.

Theorem 3.3.4. *$\widehat{R(S_t, x)}$ is an (ϵ, δ)-estimator of $R(S_t, x)$ if $m \geq \frac{\phi^2_{\delta/2}}{\epsilon^2}(\frac{1}{\inf(R(S_t,x))} - 1)$, where $\inf(R(S_t, x))$ denotes the lower bound of $R(S_t, x)$ and $\phi_{\delta/2}$ is the $\frac{\delta}{2}$ fractile of the standard normal distribution.* $\square$

The proof of Theorem 3.3.4 is similar to that of Theorem 3.3.3.

3.3.3 (ϵ, δ)-Approximate Distinct Count

The mathematical estimator of distinct count is defined by

$$\widehat{DC(S_t)} = \sum_{s_{tv}^{(d)} \in U(m)} \frac{1}{Pr(s_{tv}^{(d)} \in U(m))}.$$

Let X_v $(1 \leq v \leq DC(S_t))$ denote a 0-1 random variable that satisfies

$$X_v = \begin{cases} 1 & \text{if } \widehat{F(S_t, s_{tv}^{(d)})} > 0 \\ 0 & \text{if } \widehat{F(S_t, s_{tv}^{(d)})} = 0 \end{cases}$$

We define Y_v as

$$Y_v = \frac{X_v - \Pr(X_v = 1)}{n \Pr(X_v = 1)}, 1 \leq v \leq DC(S_t).$$

The following theorem proves that $\widehat{DC(S_t)}$ is an unbiased estimator of $DC(S_t)$ and provides the variance of $\widehat{DC(S_t)}$.

Theorem 3.3.5. *$E(\widehat{DC(S_t)}) = DC(S_t)$ and the variance of $\widehat{DC(S_t)}$, $Var(\widehat{DC(S_t)}) = \sum_{s_{tv}^{(d)} \in Dis(S_t)} \frac{1-Pr(F(\widehat{S_t, s_{tv}^{(d)}})>0)}{Pr(F(\widehat{S_t, s_{tv}^{(d)}})>0)}$.*

Proof. For all the $s_{tv}^{(d)} \in Dis(S_t)$, $F(\widehat{S_t, s_{tv}^{(d)}}) > 0$ if and only if $s_{tv}^{(d)} \in U(m)$ according to the analysis in Section 3.3.1, thus $\widehat{DC(S_t)} = \sum_{s_{tv}^{(d)} \in U(m)} \frac{1}{Pr(F(\widehat{S_t, s_{tv}^{(d)}})>0)}$.

$E(X_v) = \Pr(F(\widehat{S_t, s_{tv}^{(d)}}) > 0)$, and thus

$$E(\widehat{DC(S_t)}) = E(\sum_{s_{tv}^{(d)} \in U(m)} \frac{1}{\Pr(F(\widehat{S_t, s_{tv}^{(d)}}) > 0)}) = E(\sum_{s_{tv}^{(d)} \in Dis(S_t)} \frac{X_v}{\Pr(F(\widehat{S_t, s_{tv}^{(d)}}) > 0)})$$

$$= \sum_{v=1}^{DC(S_t)} \frac{E(X_v)}{\Pr(F(\widehat{S_t, s_{tv}^{(d)}}) > 0)} = \sum_{v=1}^{DC(S_t)} 1 = DC(S_t)$$

Meanwhile

$$Var(\widehat{DC(S_t)}) = Var(\sum_{s_{tv}^{(d)} \in Dis(S_t)} \frac{X_v}{Pr(F(\widehat{S_t, s_{tv}^{(d)}}) > 0)})$$

$$= \sum_{s_{tv}^{(d)} \in Dis(S_t)} \frac{Var(X_v)}{Pr^2(F(\widehat{S_t, s_{tv}^{(d)}}) > 0)} = \sum_{s_{tv}^{(d)} \in Dis(S_t)} \frac{1 - Pr(F(\widehat{S_t, s_{tv}^{(d)}}) > 0)}{Pr(F(\widehat{S_t, s_{tv}^{(d)}}) > 0)}.$$

□

The following two lemmas and the theorem give a way to determine the sampling size m according to ϵ and δ. The detailed proofs of *Lemma 3.3.1* and Lemma 3.3.2 are as follows.

Lemma 3.3.1. $\Pr(|\sum_{v=1}^{DC(S_t)} Y_v| \geq \frac{DC(S_t)}{n}\epsilon) \leq 2e^{-\frac{\epsilon^2 DC(S_t)^2}{4Var(\widehat{DC(S_t)})}}$.

Proof. Let n_v be the number of appearance of $s_{tv}^{(d)}$ in S_t. According to the definition of Y_v,

$$E(Y_v) = \frac{E(X_v) - \Pr(X_v = 1)}{n \Pr(X_v = 1)} = 0$$

and

$$-\frac{1}{n} \leq Y_v \leq \frac{1-\Pr(X_v=1)}{n\Pr(X_v=1)} \tag{3.6}$$

Since $\Pr(X_v = 1) = 1 - (1 - \frac{1}{n_v})^m \geq \frac{1}{n_v} \geq \frac{1}{n}$, $|Y_v| \leq 1$ according to formula (3.6).

According to Chernoff bound [58], we have

$$\Pr(|\sum_{v=1}^{DC(S_t)} Y_v| \geq \frac{DC(S_t)}{n}\epsilon) \leq 2e^{-\frac{DC(S_t)^2\epsilon^2}{4n^2 Var(\sum_{v=1}^{DC(S_t)} Y_v)}} \tag{3.7}$$

Meanwhile, we also have

$$Var(\sum_{v=1}^{DC(S_t)} Y_v) = \frac{Var(\widehat{DC(S_t)})}{n^2} \tag{3.8}$$

According to Formulas (3.7) and (3.8), $\Pr(|\sum_{v=1}^{DC(S_t)} Y_v| \geq \frac{DC(S_t)}{n}\epsilon) \leq 2e^{-\frac{\epsilon^2 DC(S_t)^2}{4Var(\widehat{DC(S_t)})}}$. □

Lemma 3.3.2. $-\frac{\epsilon^2 DC(S_t)^2}{4Var(\widehat{DC(S_t)})} \leq ln(\delta/2)$ *if* $m \geq \frac{ln(n\epsilon^2)-ln(n\epsilon^2+4n_{max}ln(2/\delta))}{ln(1-n_{min}/n)}$, *where* n_{min} *and* n_{max} *are the numbers of the appearances for the least appearing data and most appearing data respectively.*

Proof. Let n_v be the number of appearance of $s_{tv}^{(d)}$ in S_t. First, we have

$$\frac{(1-\frac{n_v}{n})^m}{1-(1-\frac{n_v}{n})^m} \leq \frac{(1-\frac{n_{min}}{n})^m}{1-(1-\frac{n_{min}}{n})^m} \leq \frac{n\epsilon^2}{4n_{max}ln(2/\delta)} \tag{3.9}$$

since $m \geq \frac{ln(n\epsilon^2)-ln(n\epsilon^2+4n_{max}ln(2/\delta))}{ln(1-n_{min}/n)}$ and $n_{min} \leq n_v$, where n_v is the number of appearance of $s_{tv}^{(d)}$ in S_t. According to Theorem 3.3.5 and Formula (3.9), we have

$$Var(\widehat{DC(S_t)}) = \sum_{s_{tv}^{(d)} \in Dis(S_t)} \frac{1-Pr(\widehat{F(S_t, s_{tv}^{(d)})} > 0)}{Pr(\widehat{F(S_t, s_{tv}^{(d)})} > 0)} = \sum_{v=1}^{DC(S_t)} \frac{(1-\frac{n_v}{n})^m}{1-(1-\frac{n_v}{n})^m}$$

$$\leq \sum_{v=1}^{DC(S_t)} \frac{n\epsilon^2}{4n_{max}ln(2/\delta)} = \frac{DC(S_t)n\epsilon^2}{4n_{max}ln(2/\delta)}$$

$$= \frac{DC(S_t)^2 n\epsilon^2}{4ln(2/\delta)DC(S_t)n_{max}} \leq \frac{DC(S_t)^2 n\epsilon^2}{4ln(2/\delta)\sum_{v=1}^{DC(S_t)} n_v} = \frac{DC(S_t)^2\epsilon^2}{4ln(2/\delta)}$$

that is, $-\frac{\epsilon^2 DC(S_t)^2}{4Var(\widehat{DC(S_t)})} \leq ln(\delta/2)$. □

Theorem 3.3.6. *$\widehat{DC(S_t)}$ is an (ϵ, δ)-estimator of $DC(S_t)$ if the sample size $m \geq \frac{ln(n\epsilon^2) - ln(n\epsilon^2 + 4n_{max} ln(2/\delta))}{ln(1-n_{min}/n)}$, where n_{min} and n_{max} are the numbers of the appearances for the least appearing data and most appearing data respectively.*

Proof. According to the definition of X_v $(1 \leq v \leq DC(S_t))$ and Y_v $(1 \leq v \leq DC(S_t))$, we have

$$\widehat{DC(S_t)} = \sum_{s_{tv}^{(d)} \in U(m)} \frac{1}{Pr(s_{tv}^{(d)} \in U(m))} = \sum_{v=1}^{DC(S_t)} \frac{X_v}{Pr(X_v = 1)}$$

$$= \sum_{v=1}^{DC(S_t)} (nY_v + 1) = DC(S_t) + n \sum_{v=1}^{DC(S_t)} Y_v.$$

Thus,

$$\Pr(Error(DC(S_t), \widehat{DC(S_t)}) \geq \epsilon) = \Pr(\frac{n}{DC(S_t)} | \sum_{v=1}^{DC(S_t)} Y_v| \geq \epsilon) = \Pr(| \sum_{v=1}^{DC(S_t)} Y_v| \geq \frac{DC(S_t)}{n}\epsilon),$$

Based on the condition of the theorem, we have

$$m \geq \frac{ln(n\epsilon^2) - ln(n\epsilon^2 + 4n_{max} ln(2/\delta))}{ln(1 - n_{min}/n)}$$

Then we have $Pr(Error(DC(S_t), \widehat{DC(S_t)}) \geq \epsilon) \leq 2e^{ln(\delta/2)} = \delta$ based on Lemma 3.3.1 and Lemma 3.3.2. Thus, $\widehat{DC(S_t)}$ is an (ϵ, δ)-estimator of $DC(S_t)$. □

3.3.4 (ϵ, δ)-Approximate Quantile

For any given r $(0 \leq r \leq 1)$, the exact quantile, denoted by $Q(S_t, r)$, satisfies that

$$Q(S_t, r) = argmin_{s_{ti}} \{ \frac{|\{s_{tj} | s_{tj} \leq s_{ti} \wedge 1 \leq j \leq n\}|}{n} \geq r \}$$

according to the definition given in Section 3.2.

Let $f_j = F(S_t, s_{tj}^{(d)})$ denote the frequency of $s_{tj}^{(d)}$ for any $s_{tj}^{(d)} \in Dis(S_t)$, thus Formula

(3.10) can be reduced to

$$Q(S_t, r) = argmin_{s_{ti}^{(d)}}\{\sum_j f_j \geq r \wedge s_{tj}^{(d)} \leq s_{ti}^{(d)}\}. \tag{3.10}$$

Since f_j is estimated by $F(\widehat{S_t, s_{tj}^{(d)}})$ in Section 3.3.1, the estimator of quantile, denoted by $Q(\widehat{S_t, r})$, can be easily constructed. For simplicity, we use $\widehat{f_j}$ to denote $F(\widehat{S_t, s_{tj}^{(d)}})$. Therefore,

$$Q(\widehat{S_t, r}) = argmin_{s_{ti}^{(d)}}\{\sum_j \widehat{f_j} \geq r \wedge s_{tj}^{(d)} \leq s_{ti}^{(d)}\}. \tag{3.11}$$

Suppose that $Q(S_t, r)$ and $Q(\widehat{S_t, r})$ are the k-th and $\widehat{k}$-th smallest values in $Dis(S_t)$. Since the actual relative error between $Q(S_t, r)$ and $Q(\widehat{S_t, r})$ is hard to guarantee since it depends on sensory data distribution, we use $Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i)$ to evaluate the relative error between $Q(S_t, r)$ and $Q(\widehat{S_t, r})$, where $Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i) = \frac{|\sum_{i=1}^{k-1} f_i - \sum_{i=1}^{\widehat{k}-1} f_i|}{\sum_{i=1}^{k-1} f_i}$. In most cases, we have $\frac{n_{min}}{n}$ being less than the average of a group of estimated frequencies, and we assume $r > \frac{n_{max}}{n}$, then we have the following lemma. The detailed proof of the following lemma is as follows.

Lemma 3.3.3. $Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i) \leq \frac{|\sum_{i=1}^{k} \widehat{f_i} - \sum_{i=1}^{k} f_i| n_{max} n + n_{min} n_{max}}{n_{min}(nr - n_{max})}$.

Proof. According to the above analysis,

$$Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i) = \frac{|\sum_{i=1}^{k-1} f_i - \sum_{i=1}^{\widehat{k}-1} f_i|}{\sum_{i=1}^{k-1} f_i} = \frac{|\sum_{i=1}^{k-1} f_i - \sum_{i=1}^{\widehat{k}-1} f_i|}{\sum_{i=1}^{k} f_i - f_k}.$$

Since $Q(S_t, r) = argmin_{s_{ti}^{(d)}}\{\sum_j f_j \geq r \wedge s_{tj}^{(d)} \leq s_{ti}^{(d)}\}$ and $Q(S_t, r)$ is the k-th smallest value in $Dis(S_t)$, $\sum_{i=1}^{k} f_i \geq r$. Moreover, $f_k \leq \frac{n_{max}}{n}$ according to the conditions in the theorem. Thus,

$$Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i) \leq \frac{|\sum_{i=1}^{k-1} f_i - \sum_{i=1}^{\widehat{k}-1} f_i|}{r - \frac{n_{max}}{n}}.$$

If $\widehat{k} < k$, $|\sum_{i=1}^{k-1} f_i - \sum_{i=1}^{\widehat{k}-1} f_i| = |\sum_{i=\widehat{k}}^{k-1} f_i| \leq |\sum_{i=\widehat{k}}^{k-1} \frac{n_{max}}{n}| = (k - \widehat{k})\frac{n_{max}}{n}$. Similarly,

$|\sum_{i=1}^{k-1} f_i - \sum_{i=1}^{\widehat{k}-1} f_i| = |\sum_{i=k}^{\widehat{k}-1} f_i| \leq |\sum_{i=k}^{\widehat{k}-1} \frac{n_{max}}{n}| = (\widehat{k}-k)\frac{n_{max}}{n}$ if $\widehat{k} \geq k$. Thus, $|\sum_{i=1}^{k-1} f_i - \sum_{i=1}^{\widehat{k}-1} f_i| \leq |\widehat{k}-k|\frac{n_{max}}{n}$, so that

$$Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i) \leq \frac{|\widehat{k}-k|\frac{n_{max}}{n}}{r - \frac{n_{max}}{n}} = \frac{|\widehat{k}-k|n_{max}}{nr - n_{max}}.$$

If $\widehat{k} = k$, we have $\Pr(Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i) \geq \epsilon) \leq \delta$.

If $\widehat{k} > k$, we have $\sum_{i=1}^{\widehat{k}-1} \widehat{f_i} < r$, $\sum_{i=k+1}^{\widehat{k}-1} \widehat{f_i} < r - \sum_{i=1}^{k} \widehat{f_i}$,

$\widehat{k} - k < \frac{\sum_{i=1}^{k} f_i - \sum_{i=1}^{k} \widehat{f_i}}{\sum_{i=k+1}^{\widehat{k}-1} \widehat{f_i}/(\widehat{k}-k-1)} + 1$. And we suppose $\frac{n_{min}}{n} \leq \sum_{i=k+1}^{\widehat{k}-1} \widehat{f_i}/(\widehat{k}-k-1)$, thus $\widehat{k} - k < \frac{\sum_{i=1}^{k} f_i - \sum_{i=1}^{k} \widehat{f_i}}{n_{min}/n} + 1$. Then we have

$$Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i) \leq \frac{|\sum_{i=1}^{k} \widehat{f_i} - \sum_{i=1}^{k} f_i| n_{max} n + n_{min} n_{max}}{n_{min}(nr - n_{max})}.$$

□

Based on Lemma 3.3.3, we have the following theorem to determine the sample size.

Theorem 3.3.7. $\Pr(Error(\sum_{i=1}^{k-1} f_i, \sum_{i=1}^{\widehat{k}-1} f_i) \geq \epsilon) \leq \delta$ *if the sample size*

$$m \geq (\frac{\phi_{\delta/2} n_{max} n}{2\epsilon n_{min}(nr - n_{max}) - 2n_{min}n_{max}})^2(\frac{nr}{n_{min}} + 1)$$

, where $\phi_{\delta/2}$ is the $\frac{\delta}{2}$ fractile of the standard normal distribution, r is the input parameter, and n_{min} and n_{max} are the numbers of the appearances for the least appearing data and most appearing data respectively.

Proof. According to the property of normal distribution [59], we have $\sum_{i=1}^{k} \widehat{f_i} \sim N(\sum_{i=1}^{k} f_i, \sum_{i=1}^{k} Var(\widehat{f_i}))$, that is,

$$\Pr(|\sum_{i=1}^{k} \widehat{f_i} - \sum_{i=1}^{k} f_i| \geq \phi_{\delta/2}\sqrt{\sum_{i=1}^{k} Var(\widehat{f_i})}) = \delta. \tag{3.12}$$

Since $(k-1)\frac{n_{min}}{n} \leq \sum_{i=1}^{k-1} f_i < r$, $k < \frac{nr}{n_{min}} + 1$. Based on the condition of the theorem,

$m \geq (\frac{\phi_{\delta/2} n_{max} n}{2\epsilon n_{min}(nr-n_{max})-2n_{min}n_{max}})^2(\frac{nr}{n_{min}}+1)$, so that

$$m \geq (\frac{\phi_{\delta/2} n_{max} n}{2\epsilon n_{min}(nr-n_{max})-2n_{min}n_{max}})^2 k.$$

Since $0 \leq f_i \leq 1$, $f_i(1-f_i) \leq \frac{1}{4}$ for all $1 \leq i \leq DC(S_t)$. According to Formula (3.10), we have

$$m \geq (\frac{\phi_{\delta/2} n_{max} n}{\epsilon n_{min}(nr-n_{max})-n_{min}n_{max}})^2 \sum_{i=1}^{k} f_i(1-f_i)$$

$$= m(\frac{\phi_{\delta/2} n_{max} n}{\epsilon n_{min}(nr-n_{max})-n_{min}n_{max}})^2 \sum_{i=1}^{k} Var(\widehat{f_i})$$

based on Theorem 3.3.2. Therefore, $\phi_{\delta/2}\sqrt{\sum_{i=1}^{k} Var(\widehat{f_i})} \leq \frac{\epsilon n_{min}(nr-n_{max})-n_{min}n_{max}}{n_{max}n}$.

Then according to Formula (3.12), we have $\Pr(|\sum_{i=1}^{k}\widehat{f_i}-\sum_{i=1}^{k}f_i| \geq \frac{\epsilon n_{min}(nr-n_{max})-n_{min}n_{max}}{n_{max}n}) \leq \delta$, that is, $\Pr(\frac{|\sum_{i=1}^{k}\widehat{f_i}-\sum_{i=1}^{k}f_i|n_{max}n+n_{min}n_{max}}{n_{min}(nr-n_{max})} \geq \epsilon) \leq \delta$. Since $Error(\sum_{i=1}^{k-1}f_i, \sum_{i=1}^{\widehat{k}-1}f_i) \leq \frac{|\sum_{i=1}^{k}\widehat{f_i}-\sum_{i=1}^{k}f_i|n_{max}n+n_{min}n_{max}}{n_{min}(nr-n_{max})}$ according to Lemma 3.3.3, we have $\Pr(Error(\sum_{i=1}^{k-1}f_i, \sum_{i=1}^{\widehat{k}-1}f_i) \geq \epsilon) \leq \delta$.

If $\widehat{k} < k$, similarly, we have $k-\widehat{k} < \frac{|\sum_{i=1}^{\widehat{k}}\widehat{f_i}-\sum_{i=1}^{\widehat{k}}f_i|}{n_{min}/n}+1$, $Error(\sum_{i=1}^{k-1}f_i, \sum_{i=1}^{\widehat{k}-1}f_i) \leq \frac{|\sum_{i=1}^{\widehat{k}}\widehat{f_i}-\sum_{i=1}^{\widehat{k}}f_i|n_{max}n+n_{min}n_{max}}{n_{min}(nr-n_{max})}$. We have

$$m \geq (\frac{\phi_{\delta/2} n_{max} n}{2\epsilon n_{min}(nr-n_{max})-2n_{min}n_{max}})^2 k \geq (\frac{\phi_{\delta/2} n_{max} n}{2\epsilon n_{min}(nr-n_{max})-2n_{min}n_{max}})^2 \widehat{k}$$

$$\geq m(\frac{\phi_{\delta/2} n_{max} n}{\epsilon n_{min}(nr-n_{max})-n_{min}n_{max}})^2 \sum_{i=1}^{\widehat{k}} Var(\widehat{f_i}).$$

Therefore, we have

$$\phi_{\delta/2}\sqrt{\sum_{i=1}^{\widehat{k}} Var(\widehat{f_i})} \leq \frac{\epsilon n_{min}(nr-n_{max})-n_{min}n_{max}}{n_{max}n}$$

Then we have $\Pr(Error(\sum_{i=1}^{k-1}f_i, \sum_{i=1}^{\widehat{k}-1}f_i) \geq \epsilon) \leq \delta$. □

3.4 (ϵ, δ)-Approximate Aggregation Algorithms

To compute (ϵ, δ)-approximate frequency, (ϵ, δ)-approximate quantile, (ϵ, δ)-approximate distinct-count, and (ϵ, δ)-approximate rank, the following three steps are required.

1. A proper sample size m needs to be determined according to ϵ, δ and the given aggregation operation.

2. A distributed uniform sampling algorithm is needed to sample the sensory data from a network, which will be discussed in Section 3.4.1.

3. The (ϵ, δ)-approximate aggregation results are calculated using the sampled data, which will be introduced in Sections 3.4.2, 3.4.3, 3.4.4 and 3.4.5, respectively.

3.4.1 The Uniform Sampling Algorithm

When the sample size m is determined, the naive sampling algorithm consists of two steps. First, the sink generates m random numbers in the range of $\{1, 2, 3, \ldots, n\}$ and broadcasts in a network. Second, the sensor node whose id is one of the m numbers sends its data to the sink.

However, as the simulation results shown in Fig.3.7, the above algorithm costs too much energy since a large amount of raw data needs to be transmitted. Therefore, we divide the whole network into k disjoint clusters $C_1, C_2, \ldots, C_k$ and adopt the uniform sampling algorithm proposed by [54] to further reduce the communication cost during the sampling process. For clarity, the uniform sampling algorithm is introduced as follows.

The uniform sampling algorithm is called the Uniform Sampling algorithm based on Clusters (USC) which requires a network to be organized into clusters, and it has three steps.

1. Each cluster randomly elects a node in it as the cluster-head. All the cluster-heads are organized as a minimum hop-count spanning tree rooted at the sink using the

method in [60]. Because the cluster-heads are randomly selected and they have limited transmission range, it cannot guarantee that the cluster-heads from two adjacent clusters could communicate with one-hop communication. Therefore, more ordinary sensor nodes are selected in the spanning tree by a greedy routing method to keep it connected.

2. The sink determines the sample size of each cluster as follows. First, the sink generates a random number Y_i with $\Pr(Y_i = l) = \frac{n_l}{n}(1 \leq i \leq m)$, where n_l is the number of the nodes in cluster C_l. Second, the sample size of C_l, denoted by m_l, is determined by $m_l = |\{Y_i | Y_i = l\}|$ for each $1 \leq l \leq k$, where k is the number of the clusters in a network. Finally, the sink sends the sample size $\{m_l \mid 1 \leq l \leq k\}$ to the cluster heads along the spanning tree.

3. When each cluster head receives the sample size, it samples the sensed data in its own cluster by the aforementioned naive sampling algorithm. When a cluster head receives the sampled data, the cluster head calculates the partial aggregation result according to aggregation operation Agg. Finally, the obtained partial aggregation result is transmitted and aggregated along the spanning tree towards the sink.

3.4.2 The (ϵ, δ)-Approximate Frequency Algorithm

For (ϵ, δ)-approximate frequency query, the problems need to be addressed are as follows.

1. How to calculate the partial frequency result inside each cluster.

2. How to aggregate the partial frequency results during transmission.

3. How to return the (ϵ, δ)-frequency result when the sink receives the partial frequency results from the network.

There are three steps to solve the first problem. First, for each cluster head of the clusters C_l $(1 \leq l \leq k)$, it uniformly generates random numbers k_1, k_2, $\ldots, k_{m_l}$ when it receives m_l from the sink. Then, it broadcasts $k_1, k_2, \ldots, k_{m_l}$ inside the cluster. Second, the member node whose id is one of $\{k_1, k_2, \ldots, k_{ml}\}$ sends its sensory value to the cluster-head. The cluster-head can then receive the sample data $U(m_l) = \{s_{tk_1}, s_{tk_2}, \ldots, s_{tk_{m_l}}\}$ from its own cluster. Third, the cluster-head calculates the partial frequency result $Fre(U(m_l))$ according to sample data s_{tk1}, s_{tk1}, $\ldots$, $s_{tk_{ml}}$.

For the second problem, there are three steps.

1. For each node j in the spanning tree, it sets Fre_j to $\emptyset$.

2. For each node j in the spanning tree, send $Fre(U(m_j))$ to its parent node if j is a leaf node. Otherwise, j receives the partial frequency results from its children and merges them to have Fre_j.

3. If j is the sink, it returns Fre_j as the final result. If j is not the sink, it sends Fre_j to its parent.

Finally, it is easy to solve the third problem when the sink receives the partial frequency result Fre from the network. The whole algorithm is shown in Algorithm 1.

According to the analysis in Section 3.3.1, the sample size $m = \min(\lceil \frac{\phi^2_{\delta/2}}{\epsilon^2}(\frac{n}{n_{min}} - 1)\rceil, n)$, thus $m = O(\frac{\phi^2_{\delta/2}}{\epsilon^2}) = O(\frac{1}{\epsilon^2} ln(\frac{1}{\delta}))$. In practice, $|Fre|$ can be regarded as a constant. According to [54], the communication cost and the energy cost of the (ϵ, δ)-frequency algorithm is $O(|Fre|\frac{1}{\epsilon^2} ln\frac{1}{\delta}) = O(\frac{1}{\epsilon^2} ln\frac{1}{\delta})$.

3.4.3 (ϵ, δ)-Approximate Rank Algorithm

Since the rank of v is equal to the frequency of the sensory values being less than v, a rank query is a special case of the frequency query. Therefore, the algorithm to calculate (ϵ, δ)-approximate rank can be easily obtained by slightly modifying the (ϵ, δ)-frequency algorithm. The detailed (ϵ, δ)-approximate rank algorithm is omitted due to space limitation.

The communication and computation complexities of the (ϵ, δ)-approximate rank algorithm are the same as those of the (ϵ, δ)-approximate frequency algorithm, which are $O(\frac{1}{\epsilon^2} ln\frac{1}{\delta})$.

Algorithm 1: (ϵ, δ)-Approximate Frequency Algorithm

```
Input: $\epsilon, \delta$
Output: $(\epsilon, \delta)$-approximate frequency
1  $m = min(\lceil \frac{\phi^2_{\delta/2}}{\epsilon^2}(\frac{n}{n_{min}} - 1)\rceil, n)$;
2  generate $Y_i$ following $\Pr(Y_i = l) = \frac{n_l}{n}$, where $n_l$ is the number of the nodes in cluster
   $C_l$ $(1 \le i \le m, 1 \le l \le k)$;
3  $m_l = |\{Y_i \mid Y_i = l\}|$ $(1 \le i \le m, 1 \le l \le k)$, the sink sends $m_l$ to each cluster head
   by multi-hop communication;
4  for each cluster head of the clusters $C_l$ $(1 \le l \le k)$ do
5      generates random numbers $k_1$, $k_2$, $\ldots, k_{m_l}$, then broadcast inside the cluster;
6  end
7  for each cluster member of the clusters $C_l$ $(1 \le l \le k)$ do
8      send sensory value to cluster head if $id \in \{k_1, k_2, \ldots, k_{ml}\}$;
9  end
10 for each cluster head of the clusters $C_l$ $(1 \le l \le k)$ do
11     receive sample data $U(m_l)$ and calculate partial frequency result $Fre(U(m_l))$;
12 end
13 for each node $j$ in the spanning tree do
14     $Fre_j = \emptyset$;
15     send $Fre(U(m_j))$ to parent if $j$ is a leaf, otherwise get $Fre_j$ by merging
       children's results;
16     return $Fre_j$ if $j$ is the sink, otherwise send $Fre_j$ to parent;
17 end
```

3.4.4 (ϵ, δ)-Approximate Distinct-count Algorithm

Based on the analysis in Section 3.3, the mathematical estimator of distinct-count satisfies

$$\widehat{DC(S_t)} = \sum_{s^{(d)}_{tv} \in U(m)} \frac{1}{1 - (1 - F(S_t, s^{(d)}_{tv}))^m}. \tag{3.13}$$

However, it is impossible to obtain the accurate frequency in Formula (3.13). In practice, the approximate frequency computed in Section 3.4.2 can be used to calculate (ϵ, δ)-approximate distinct-count as follows

$$\widehat{DC(S_t)}' = \sum_{s^{(d)}_{tv} \in U(m)} \frac{1}{1 - (1 - \widehat{F(S_t, s^{(d)}_{tv})})^m}. \tag{3.14}$$

Obviously, the approximate frequency will introduce new error to the approximate distinct-count result. Therefore, the sample size must be further enlarged to guarantee that the returned distinct-count result is the (ϵ, δ)-estimator of the exact one.

Fortunately, the following theorem provides a method to determine the sample size m to derive (ϵ, δ)-approximate distinct-count. To prove Theorem 3.4.1, the following two lemmas needs to be guaranteed firstly. The detailed proofs of Lemma 3.4.1 and Lemma 3.4.2 are as follows.

Lemma 3.4.1. *Let p and $\widehat{p}$ be the frequency of an element in S_t, then $Error(\frac{1}{1-(1-p)^m}, \frac{1}{1-(1-\hat{p})^m}) \leq \frac{\epsilon(1-(1-\epsilon)n_{min}/n)^{\lfloor \frac{m}{2} \rfloor}}{1-\epsilon}$ if $Error(p, \hat{p}) \leq \epsilon$.*

Proof. If $\hat{p} > p$, we have $\hat{p} = (1+\delta)p$, $0 < \delta \leq \epsilon$

$$Error(\frac{1}{1-(1-p)^m}, \frac{1}{1-(1-\hat{p})^m}) = \frac{(1-p)^m - (1-\hat{p})^m}{1-(1-\hat{p})^m} \leq \frac{(1-p)^m - (1-\hat{p})^m}{1-(1-p)^m}$$

$$= \frac{((1-p)-(1-\widehat{p}))\sum_{i=0}^{m-1}(1-p)^{m-1-i}(1-\hat{p})^i}{p\sum_{i=0}^{m-1}(1-p)^i}$$

$$= \delta\frac{\sum_{i=0}^{m-1}(1-p)^{m-1-i}(1-\hat{p})^i}{\sum_{i=0}^{m-1}(1-p)^i} \leq \delta\frac{\sum_{i=0}^{m-1}(1-p)^{m-1-i}(1-p)^i}{\sum_{i=0}^{m-1}(1-p)^i} = \delta\frac{m(1-p)^{m-1}}{\sum_{i=0}^{m-1}(1-p)^i}$$

According to the property of concave function, we have $(1-p)^i + (1-p)^{m-i-1} \geq 2(1-p)^{\lceil \frac{m}{2} \rceil - 1}$, thus $\sum_{i=0}^{m-1}(1-p)^i \geq m(1-p)^{\lceil \frac{m}{2} \rceil - 1}$, in other words

$$Error(\frac{1}{1-(1-p)^m}, \frac{1}{1-(1-\hat{p})^m}) \leq \delta\frac{m(1-p)^{m-1}}{m(1-p)^{\lceil \frac{m}{2} \rceil - 1}} \leq \epsilon(1-\frac{n_{min}}{n})^{\lfloor \frac{m}{2} \rfloor}$$

If $\hat{p} \leq p$, we have $\hat{p} = (1-\delta)p$, $0 \leq \delta \leq \epsilon$

$$Error(\frac{1}{1-(1-p)^m}, \frac{1}{1-(1-\hat{p})^m}) = \frac{(1-\hat{p})^m - (1-p)^m}{1-(1-\hat{p})^m}$$

$$= \frac{((1-\hat{p})-(1-p))\sum_{i=0}^{m-1}(1-\hat{p})^{m-1-i}(1-p)^i}{\hat{p}\sum_{i=0}^{m-1}(1-\hat{p})^i}$$

$$\leq \frac{\delta}{1-\delta} \times \frac{\sum_{i=0}^{m-1}(1-\hat{p})^{m-1-i}(1-\hat{p})^i}{\sum_{i=0}^{m-1}(1-\hat{p})^i} = \frac{\delta}{1-\delta} \times \frac{m(1-\hat{p})^{m-1}}{\sum_{i=0}^{m-1}(1-\hat{p})^i}$$

$$\leq \frac{\delta}{1-\delta} \times \frac{m(1-\hat{p})^{m-1}}{m(1-\hat{p})^{\lceil \frac{m}{2} \rceil - 1}} = \frac{\delta}{1-\delta} \times (1-\hat{p})^{\lfloor \frac{m}{2} \rfloor} \leq \frac{\epsilon(1-(1-\epsilon)n_{min}/n)^{\lfloor \frac{m}{2} \rfloor}}{1-\epsilon}.$$

And it is easy to prove

$$\epsilon(1-\frac{n_{min}}{n})^{\lfloor \frac{m}{2} \rfloor} < \frac{\epsilon(1-(1-\epsilon)n_{min}/n)^{\lfloor \frac{m}{2} \rfloor}}{1-\epsilon}$$

Then this lemma is proved. □

Lemma 3.4.2. *If $Error(\frac{1}{1-(1-F(s_{tv}^{(d)}))^m}, \frac{1}{1-(1-\widehat{F(s_{tv}^{(d)})})^m})$ and the error of distinct count are both at most $\sqrt{1+\epsilon}-1$, the error of the final result is at most ϵ*

Proof. Let $Error(\frac{1}{1-(1-F(s_{tv}^{(d)}))^m}, \frac{1}{1-(1-\widehat{F(s_{tv}^{(d)})})^m}) = \epsilon'$, then we have

$$(1-\epsilon')\widehat{DC(S_t)} \leq \widehat{DC(S_t)}' \leq (1+\epsilon')\widehat{DC(S_t)}$$

Let ϵ'' be the error of distinct count based on accurate frequency, we have

$$(1-\epsilon'')DC(S_t) \leq \widehat{DC(S_t)} \leq (1+\epsilon'')DC(S_t)$$

$$(1-\epsilon')(1-\epsilon'')DC(S_t) \leq \widehat{DC(S_t)}' \leq (1+\epsilon')(1+\epsilon'')DC(S_t)$$

Let $\epsilon' = \epsilon''$, $(1-\epsilon')(1-\epsilon'') = 1-\epsilon$, then $\epsilon' = \epsilon'' = 1-\sqrt{1-\epsilon}$. Let $\epsilon' = \epsilon''$, $(1+\epsilon')(1+\epsilon'') = 1+\epsilon$, we have $\epsilon' = \epsilon'' = \sqrt{1+\epsilon}-1$. And it is easy to prove $1-\sqrt{1-\epsilon} > \sqrt{1+\epsilon}-1$, so this lemma is proved. □

Theorem 3.4.1. *$\widehat{DC(S_t)}'$ is the (ϵ,δ)-estimator of $DC(S_t)$ if the sampling size m satisfies $m \geq max(m_1, m_2)$. $m_1 = \frac{ln(n(\sqrt{\epsilon+1}-1)^2)-ln(n(\sqrt{\epsilon+1}-1)^2+4n_{max}ln(2/(1-\sqrt{1-\delta})))}{ln(1-n_{min}/n)}$, $m_2 = \frac{\phi_{(1-\sqrt{1-\delta})/2}}{\epsilon'^2}(\frac{n}{n_{min}}-1)$, where ϵ' is the solution of the equation $\frac{\epsilon'(1-(1-\epsilon')n_{min}/n)^{\lfloor \frac{m_1}{2} \rfloor}}{1-\epsilon'} = \sqrt{1+\epsilon}-1$.*□

Algorithm 2: (ϵ, δ)-Approximate Distinct Count Algorithm

```
Input: $\epsilon, \delta$
Output: $(\epsilon, \delta)$-approximate distinct count
1  $\epsilon_1 = \sqrt{1+\epsilon} - 1$;
2  $\delta_1 = 1 - \sqrt{1-\delta}$;
3  $m_1 = \lceil \frac{ln(n\epsilon_1^2) - ln(n\epsilon_1^2 + 4n_{max}ln(2/\delta_1))}{ln(1-n_{min}/n)} \rceil$;
4  Solve the equation $\frac{\epsilon_2(1-(1-\epsilon_2)\frac{n_{min}}{n})^{\lfloor m_1/2 \rfloor}}{1-\epsilon_2} = \epsilon_1$;
5  $m_2 = \lceil \frac{\phi^2_{\delta_1/2}}{\epsilon_2^2}(\frac{n}{n_{min}} - 1) \rceil$;
6  $m = \min(\max(m_1, m_2), n)$;
7  get the approximate frequency $Fre$ with sample size $m$;
8  $sum = 0$;
9  for $i = 1$ to $|Fre|$ do
10     $temp = 1 + (1 - Fre.Count[i])^m$;
11     $sum = sum + 1/temp$;
12 end
13 return $sum$;
```

Proof. According to Lemma 3.4.1 and the definitions of m_1 and m_2, we know that $Error(\frac{1}{1-(1-F(s_{tv}^{(d)}))^m}, \frac{1}{1-(1-F(\widehat{s_{tv}^{(d)}}))^m})$ and the error of distinct count based on accurate frequency are both at most $\sqrt{1+\epsilon} - 1$. According to Lemma 3.4.2, the error of the final result is at most ϵ.

Moreover, according to the definitions of m_1 and m_2, the probability of returning over-error results for frequency and distinct count are both at most $1 - \sqrt{1-\delta}$, so the probability of returning an over-error final result is at most δ. It satisfies the definition of (ϵ, δ)-estimator.

□

Then we can calculate (ϵ, δ)-approximate distinct-count based on the approximate frequency. The detailed algorithm is shown in Algorithm 2.

Since this algorithm is based on the approximate frequency algorithm, the communication cost and the energy cost of the (ϵ, δ)-approximate distinct-count algorithm is $O(m) = O(\frac{1}{\epsilon_2^2} ln \frac{1}{\delta_1}) = O(\frac{1}{\epsilon_2^2} ln \frac{1}{\delta})$.

Algorithm 3: (ϵ, δ)-Approximate Quantile Algorithm

```
Input: $\epsilon$, $\delta$, rank $r$
Output: $(\epsilon, \delta)$-approximate value
1  $m_1 = \lceil (\frac{\phi_{\delta/2} n_{max} n}{2\epsilon n_{min}(nr - n_{max}) - 2n_{min}n_{max}})^2 (\frac{nr}{n_{min}} + 1) \rceil$;
2  $m_2 = \lceil (\frac{\phi_{\delta/2} n_{max} n}{2\epsilon n_{min}(n(1-r) - n_{max}) - 2n_{min}n_{max}})^2 (\frac{n(1-r)}{n_{min}} + 1) \rceil$;
3  $m = \min(m_1, m_2, n)$;
4  get the approximate frequency $Fre$ with sample size $m$;
5  $sum = 0$;
6  if $m_1 < m_2$ then
7      for $i = 1$ to $|Fre|$ do
8          $sum = sum + Fre.Count[i]$;
9          if $sum \geq r$ then
10             return $Fre.Value[i]$;
11         end
12     end
13 else
14     for $i = |Fre|$ to $1$ do
15         $sum = sum + Fre.Count[i]$;
16         if $sum \geq 1 - r$ then
17             return $Fre.Value[i]$;
18         end
19     end
20 end
```

3.4.5 (ϵ, δ)-Approximate Quantile Algorithm

Suppose that $Dis(S_t)$ is in an ascending order. For any given r, a quantile query can be processed by adding the frequency of each item in $Dis(S_t)$ one by one until the sum exceeds r. Similarly, if $Dis(S_t)$ is in a descending order, we can return the quantile result by adding the frequency of each item one by one until the sum is greater than $1 - r$. Thus, the (ϵ, δ)-approximate quantile algorithm can also be obtained by revising the (ϵ, δ)-approximate frequency algorithm. The detailed algorithm is shown in Algorithm 3.

Similarly, the communication cost and the energy cost of the (ϵ, δ)-approximate quantile algorithm is $O(m) = O(\frac{\phi_{\delta/2}}{\epsilon^2}) = O(\frac{1}{\epsilon^2} ln \frac{1}{\delta})$.

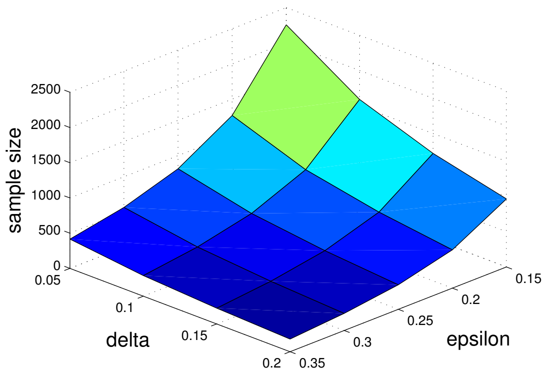
3.5 Simulation Results

To evaluate the proposed algorithms, we stimulated a network with 5000 nodes. The nodes are randomly distributed in a rectangular region with size $300m \times 300m$. The sink is placed in the center of the region. The region is divided into 10×10 grids and the nodes in the same gird are grouped into the same cluster. The cluster head is randomly chosen among all the nodes in the same grid. According to [4], for each node, the energy cost to send and receive one byte are set to be 0.0144mJ and 0.0057mJ, respectively. The results in [61] for the same type of sensor nodes indicate that the radio range of a sensor node could be 50m or even longer. Therefore, the radio range of each sensor node is set to be $30\sqrt{2}$m so that every sensor node can communicate with its cluster head by one-hop message.

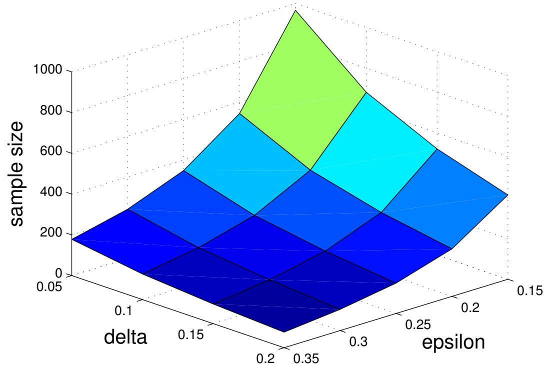
3.5.1 Evaluation of Sample Size and Relative Error

The first group of simulations is to investigate the relationship among ϵ, δ and the sample size. Since we consider four aggregation operations in this chapter, the required sample sizes for different operations are calculated respectively with different ϵ and δ. The results are presented in Fig.3.1(a), Fig.3.1(b), Fig.3.1(c) and Fig.3.1(d). All the four figures show that the sample size increases with the decline of ϵ and δ since more sample data are needed when the specified precision is high. Meanwhile, all the sample sizes are very small compared with the size of the network. For example, when $\epsilon = \delta = 0.2$, the required sample size is about 550 for deriving (ϵ, δ)-approximate frequency while the size of the network is 5000, which means that we only need to sample 11% sensory data from the network to guarantee that the probability of the relative error of approximate frequency being less than 0.2 is greater than 0.8. Therefore, our USC algorithm saves lots of energy since a little amount of sensory data is sampled and transmitted in the network.

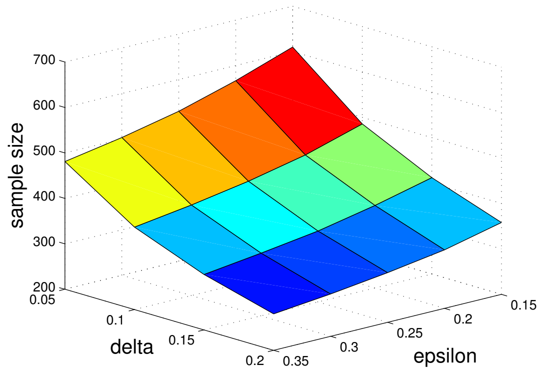
The second group of simulations is to investigate the relationship among the relative error of the final result while ϵ varies from 0.35 to 0.15 and δ varies from 0.05 to 0.2 for the (ϵ, δ)-approximate frequency algorithm and (ϵ, δ)-approximate rank algorithm. Fig.3.2(a)



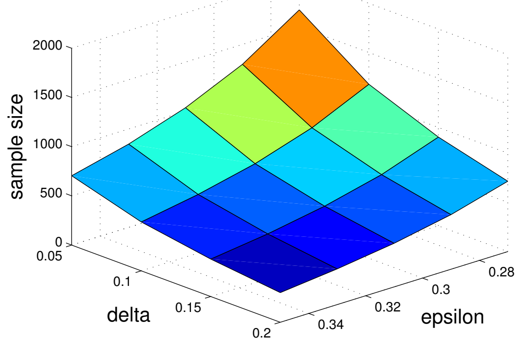
(a) Frequency



(b) Rank



(c) Distinct-count



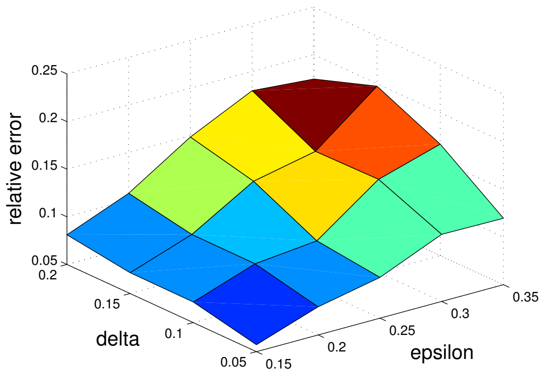
(d) Quantile

Figure 3.1. The relationship among ϵ, δ and the sample size.

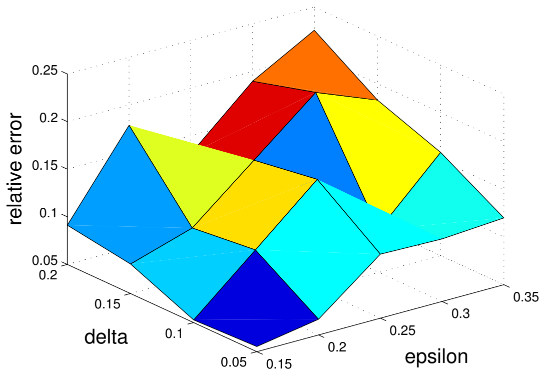
and Fig.3.2(b) show the results. We can see the approximate algorithms can achieve the specified precision. The results also show that our algorithms can obtain an arbitrary precision.

The third group of simulations is to investigate the relationship between the relative error of the final result and the sampling ratio while the sampling ratio varies from 0.05 to 0.4 for the (ϵ, δ)-approximate frequency algorithm and (ϵ, δ)-approximate rank algorithm. The results are shown in Fig.3.3(a) and Fig.3.3(b). The results show that in most cases, the error decreases with the increase of sample ratio since more sensory data are sampled.

The forth group of simulations is to investigate the relationship among the relative error of the final result, the sampling ratio and the network size for the (ϵ, δ)-approximate

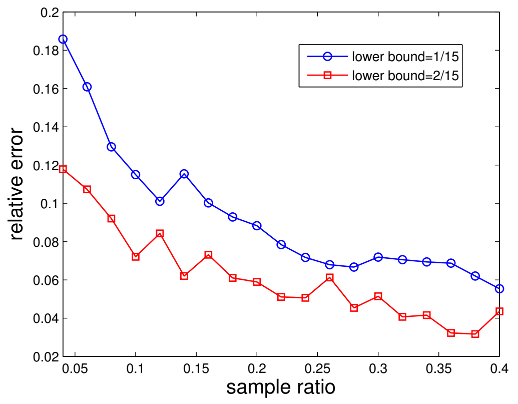


(a) Frequency

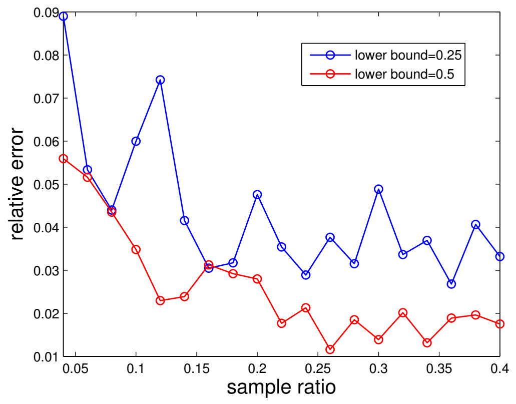


(b) Rank

Figure 3.2. The relationship among ϵ, δ and the relative error.



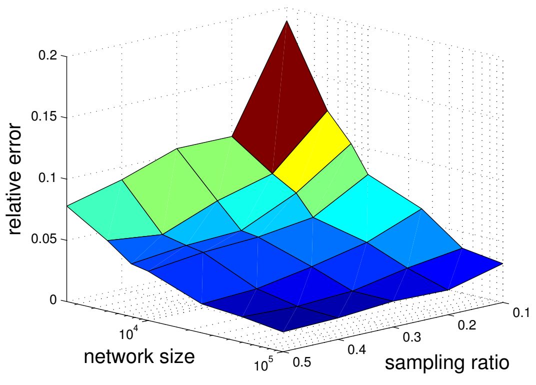
(a) Frequency



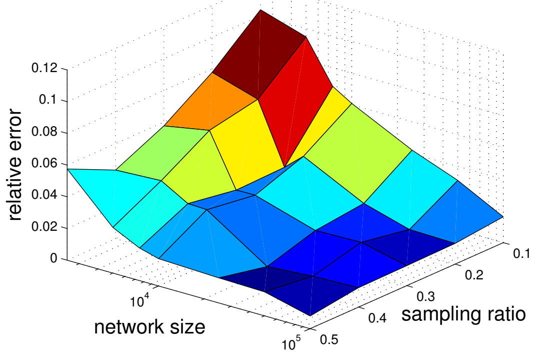
(b) Rank

Figure 3.3. The relationship between the sampling ratio and the relative error.

frequency algorithm and (ϵ, δ)-approximate rank algorithm. The sampling ratio varies from 0.1 to 0.5 and the network size varies from 2500 to 100000. The results are presented in Fig.3.4(a) and Fig.3.4(b). The results show that for different network sizes, in most cases, the error decreases with the increase of sampling ratio since more sensory data are sampled. It is also easy to find that for the same sampling ratio, the error of the result decrease with the increase of network size since for the same sampling ratio, the sample size increases with the increase of network size. It indicates that our algorithms are suitable for large-scale networks.

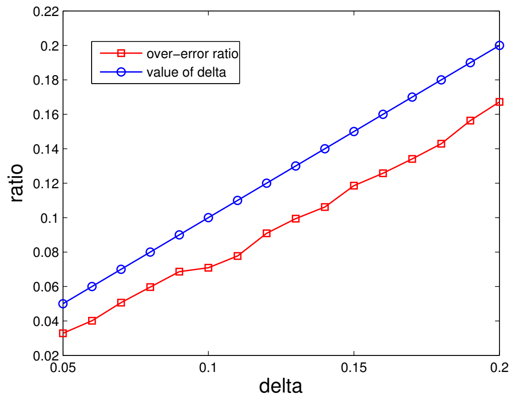


(a) Frequency

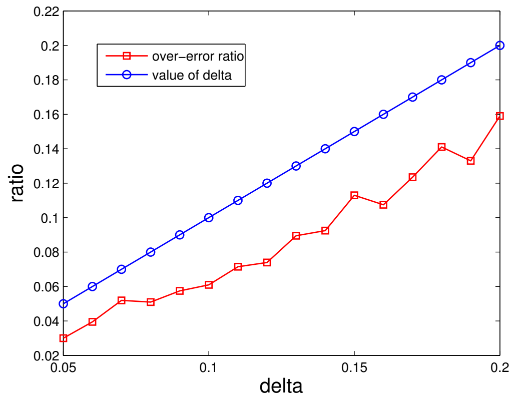


(b) Rank

Figure 3.4. The relationship among network size, sampling ratio and relative error.



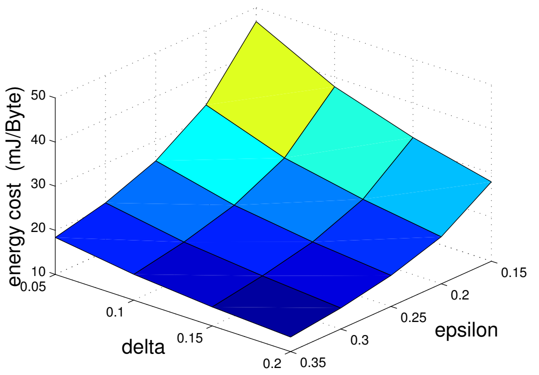
(a) Frequency



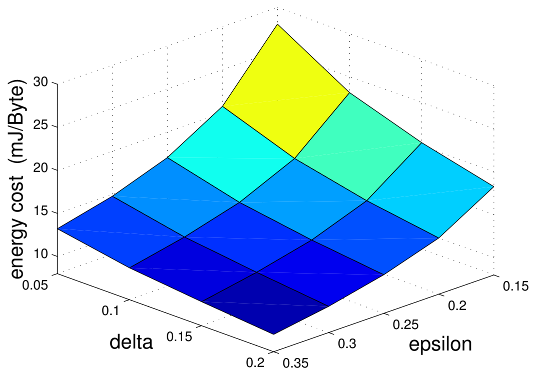
(b) Rank

Figure 3.5. The relationship between δ and the over-error ratio.

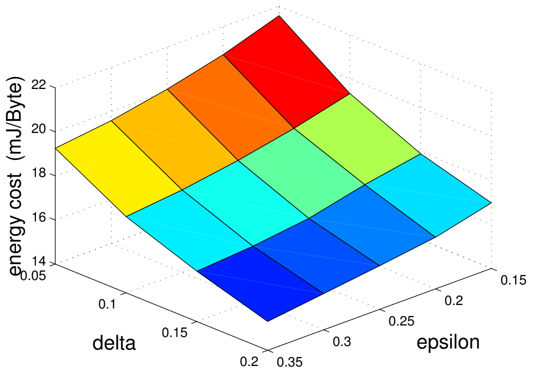
The fifth group of simulations is to investigate the relationship between the ratio of results whose error is larger than ϵ and δ for the (ϵ, δ)-approximate frequency algorithm and (ϵ, δ)-approximate rank algorithm. ϵ is set to 0.2 and δ varies from 0.05 to 0.2. The results are shown in Fig.3.5(a) and Fig.3.5(b). We can see that although this ratio increases with the increase of δ, it is lower than δ. That means these algorithms can return the approximate results which satisfy the required precision requirement with a probability greater than $1-\delta$ as expected.



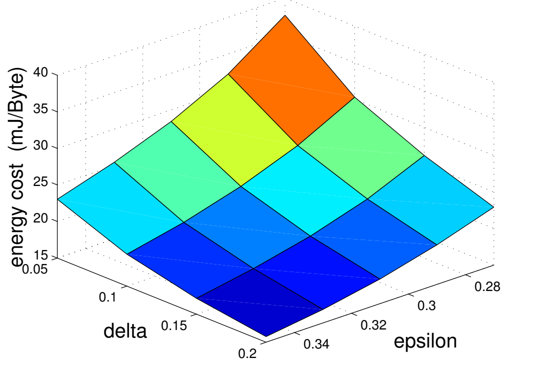
(a) Frequency



(b) Rank



(c) Distinct-count



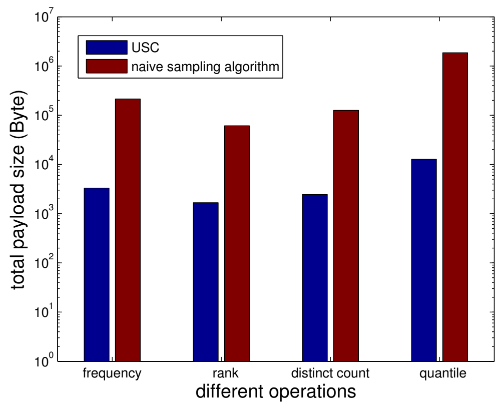
(d) Quantile

Figure 3.6. The relationship among ϵ, δ and the energy cost.

3.5.2 Evaluation of Energy Cost

The first group of simulations is to investigate the relationship among ϵ, δ and the energy cost. The energy costs for different operations are calculated respectively with different ϵ and δ. The results are shown in Fig.3.6(a), Fig.3.6(b), Fig.3.6(c) and Fig.3.6(d). All the four figures show that the energy cost increases with the decline of ϵ and δ since more sample data are needed when the specified precision is high.

The second group of simulations is to compare the total payload size of the transmitted packets between the naive sampling algorithm and the USC algorithm during the process of broadcasting the sampling information. The results are shown in Fig.3.7. According to the



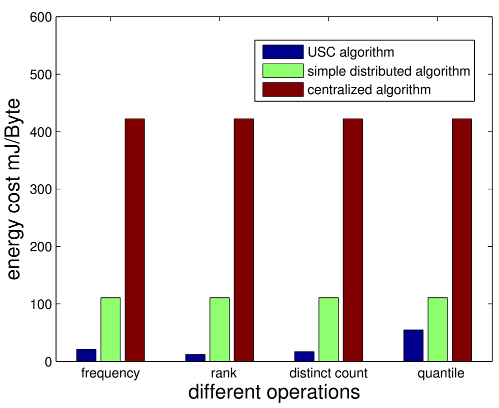


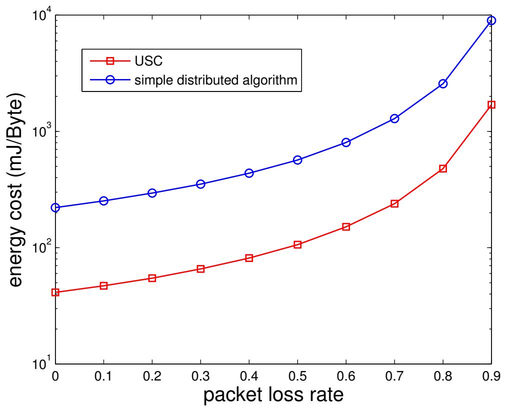
Figure 3.7. The data flow comparison

Figure 3.8. The energy cost comparison

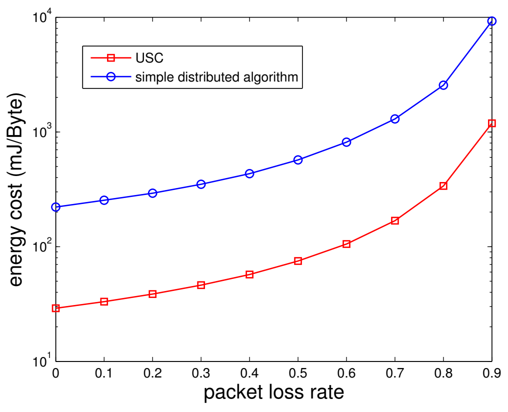
results, the USC algorithm has much smaller total payload size since the payload of packets transmitted in the USC algorithm only contains the sample size in each cluster rather than the IDs of all the sampled nodes. Therefore, the total energy cost of the USC algorithm is largely reduced comparing with the naive sampling algorithm.

The third group of simulations is to compare the energy cost between the USC algorithm, the simple distributed algorithm and the centralized algorithm. We set $\epsilon = \delta = 0.2$. The results are shown in Fig.3.8, which indicate the USC algorithm has the least energy cost among all the three aggregation algorithms. Furthermore, since data are aggregated during transmission, the energy cost of the simple distributed algorithm is much smaller than that of the centralized one.

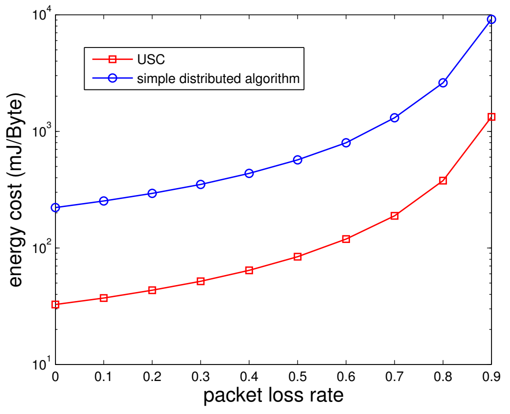
The forth group of simulations investigates the relationship between the packet loss rate and the energy cost. In this group of simulations, the packet loss rate of data transmission between two sensor nodes changes from 0 to 0.9 and we set $\epsilon = \delta = 0.2$. We use the stop-and-wait protocol [62] to do retransmission to make sure the sink node could still receive the sensory data. The results are shown in Fig.3.9. The results indicate that for both the USC algorithm and the simple distributed algorithm, their energy cost increases with the increase of packet loss rate due to additional packet retransmission. However, the energy cost for the USC algorithm is still much smaller than that of the simple distributed algorithm, which means the USC algorithm has high performance even when the pocket loss rate is



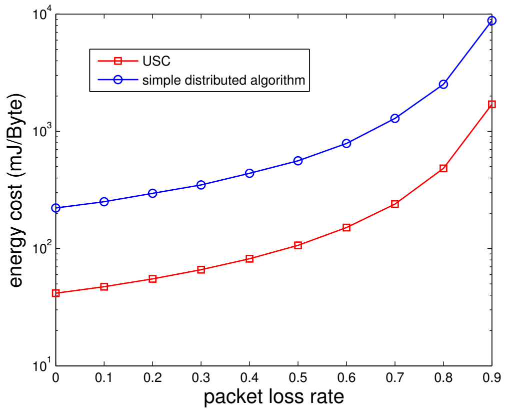
(a) Frequency



(b) Rank



(c) Distinct-count



(d) Quantile

Figure 3.9. The relationship between packet loss rate and the energy cost.

high, and the energy cost of the simple distributed algorithm is almost the same for the four operations. The reason is that the simple distributed algorithm needs to collect all the sensory data, while the USC algorithm only needs to transmit a sample in the network. Thus, the number of data packets transmitted by the USC algorithm is quite smaller, so that the retransmitting times of the USC algorithm is also much smaller compared with the simple distributed algorithm when a packet is missing. Finally, the energy consumed by the USC algorithm is much less than that of the simple distributed algorithm.

The fifth group of simulations is about the energy cost in small scale networks. The results are presented in Fig.3.10. The values of both ϵ and δ are set to 0.3 and the network

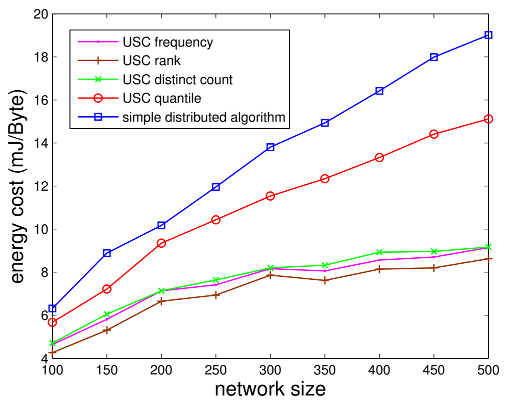


Figure 3.10. Energy cost comparison for small scale network

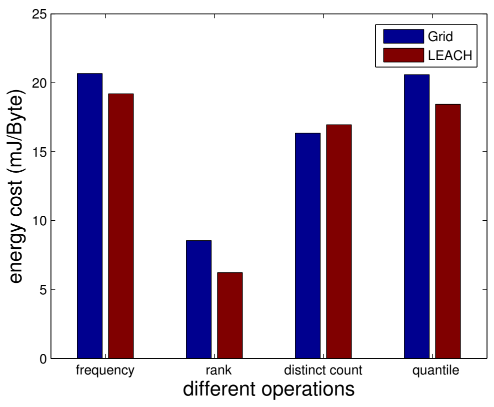


Figure 3.11. Energy cost comparison for different clustering methods

size varies from 100 to 500. Based on the results, the energy costs of the USC algorithm is still smaller than that of the simple distributed algorithm even in small-scale networks since it only uses the sample data instead of the raw sensory data to process queries. Furthermore, the energy cost of the USC algorithm increases slowly with the growth of network size, which also verifies that our USC algorithm is suitable for large scale networks.

In the sixth group of simulations, another famous clustering method, LEACH [63], is considered. The results are shown in Fig.3.11. We can see that the USC algorithm consumes low energy to deal with the four aggregation operations when different clustering methods are employed.

3.6 Conclusions

In this chapter, the (ϵ, δ)-approximate algorithms for the frequency, rank, distinct-count and quantile aggregation operations in networks are proposed. Furthermore, the sample size which can make the final result to satisfy the specified precision and failure probability requirements is derived. In addition, a cluster-based uniform sampling algorithm is provided. The simulation results show that the proposed algorithms have high performance on both energy cost and accuracy.

Chapter 4

BERNOULLI SAMPLING BASED APPROXIMATE HOLISTIC AGGREGATION IN MOBILE SENSING

4.1 Introduction

With the development of wireless communication techniques and embody systems, the Cyber Physical Systems (CPSs) [64] are being rapidly developed and widely employed. As an important CPS instance, Mobile Ad hoc Networks (MANETs) play an important role in many areas such as industrial control and intelligent traffic systems. MANETs are continuously self-configured and infrastructure-less networks consisting of mobile nodes. The mobile nodes in a MANET are able to communicate using wireless links. This kind of networks can be widely used in many applications such as pollution monitoring, animal surveillance, *etc.* [65]. Many works such as [66], [67] [68] study the problem of routing, fuzzy intrusion detection and performance evaluation in MANETs.

Frequency query is a popular operation in MANETs aiming at acquiring the frequency of some values in a sensory data set. For example, a frequency query can help with estimating the numbers of the injured persons and survivors in a disaster [69].

For many MANET applications, the monitoring period is quite long and the size of the network is very large, so that the amount of sensory data in a MANET could exceed the affordable processing cost [70] [71] [72]. Although an exact frequency result can be calculated by collecting all the sensory data and aggregating partial results during the transmission, it results in a huge amount of energy consumption. Moreover, because most nodes in a MANET are mobile and each node's communication radius is limited, the isolated nodes cannot send their data immediately. Then a severe delay may be incurred to wait for the isolated nodes to get connected with some other nodes in the network. The above example indicates that special techniques should be applied to deal with the big sensory data in MANETs [73].

Currently, many MANET applications are in favor of approximate results for the purpose of energy conservation and reduced delay [74] [75] [76] [77] [78] [79]. Although the algorithms in the previous chapter and the works in [54] and [55] propose approximate data aggregation algorithms in Wireless Sensor Networks (WSNs). However, they cannot be employed in MANETs because these algorithms are for stationary networks.

In this part, a Bernoulli sampling based approximate algorithm to process frequency queries in MANETs is proposed. This algorithm is able to return the query result which satisfies the user-specified precision and failure probability. The simulation results show that on the aspects of both energy efficiency and accuracy, the proposed algorithm has high performance.

4.2 Problem Definition

4.2.1 Network Model

Suppose there is a MANET with n mobile nodes. The set of all the mobile nodes in a MANET is denoted by $S = \{s_1, s_2, \dots, s_n\}$. All the mobile nodes are deployed in an $l \times l$ square uniformly, randomly and independently. All the mobile nodes are moving under the random waypoint model [80]. For each mobile node, the destination, speed and pause time during the movement are chosen in $[0, l)^2$, $[v_{min}, v_{max})$ and $[p_{min}, p_{max})$ uniformly, randomly and independently. At time t, a frequency query is proposed by query node q located at position $(\frac{l}{2}, \frac{l}{2})$. All the mobile nodes including the query node have communication radius r. For $\forall u, v \in S \cup \{q\}$, let $dis(u, v)$ be the Euclidean distance between u and v. Suppose u and v can communicate if $dis(u, v) \leq r$. Obviously, since all the nodes are moving and the communication radius r is not large, that is, not all the nodes can communicate with the query node in one or multiple hops. Then we use S' to denote the set of the nodes which can communicate with the query node in one or multiple hops. We assume the topology of a MANET changes between different queries while the network topology remains the same during one particular query even if it needs multiple hops for data transmission. The reason is that, compared with the time for transmitting limited bytes of data using wireless links, longer time is needed for the change of the the network topology.

4.2.2 Frequency Queries

Suppose a query is proposed at time t. For mobile node v, $d(v,t)$ is the sensory data collected by v at time t. For mobile node set V, $D(V,t)$ is the sensory data set containing all the data collected by all the nodes in V at time t. We assume all the possible values of the sensory data set distribute uniformly for all the nodes. Spatial and temporal correlations of sensory data are ignored. In this chapter, the ordinary frequency queries, *i.e.*, single value frequency queries and range frequency queries, are studied. An ordinary frequency query is to get the frequency of each distinct value in a sensory data set. For a given data set D, the exact ordinary frequency query result of D is denoted by

$$F(D,x) = \frac{|\{d \in D \mid d = x\}|}{|D|}$$

where x is any distinct value in D.

In some applications, we only need the frequency of some particular value rather than the frequency of all the distinct values in a data set. For example, we may only concern about the frequency of injured persons in a disaster [69]. For a given data set D and a particular value x', the exact single value frequency is defined as

$$SF(D,x') = \frac{|\{d \in D \mid d = x'\}|}{|D|}.$$

Some sensory data vary continuously in many applications, such as temperature data, humidity data and so on. For such sensory data, it is almost impossible and impractical to evaluate the frequency of a specific value. Therefore, a range frequency becomes more meaningful. For a given data set D and a user-specified range $[Min,\ Max]$, the exact range frequency is defined as

$$RF(D, Min, Max) = \frac{|\{d \in D \mid d \in [Min, Max]\}|}{|D|}.$$

If $Min = -\infty$, $RF(D, Min, Max)$ is the frequency of data being less than or equal to Max. If $Max = +\infty$, $RF(D, Min, Max)$ is the frequency of data being greater than or equal to Min.

In this chapter, we study how to obtain an (ϵ, δ)-approximate frequency query result for an ordinary frequency query, a single value frequency and a range frequency query, respectively. The problem of computing (ϵ, δ)-approximate frequency is then defined as
Input: A MANET with n nodes, the sensory data set D_t, ϵ $(\epsilon > 0)$ and δ $(0 \leq \delta \leq 1)$.
Output: (ϵ, δ)-approximate frequency result.

4.3 Preliminaries

In this section, detailed methodology regarding how to decide an optional sampling probability is introduced. We use $\phi_{\delta/2}$ to denote the $\delta/2$ fractile of the standard normal distribution and $\inf(x)$ to denote the lower bound of value x.

4.3.1 Basics of Sampling Probability for Ordinary Frequency Queries

Let $B(D, q) = \{b_1, b_2, \ldots, b_{|B(D,q)|}\}$ denote a Bernoulli sample of data set $D = \{d_1, d_2, \ldots, d_{|D|}\}$ with sample probability q. $Count(D) = |D|$ is the exact size of D. Then the estimator of $Count(D)$ is $\widehat{Count(D)} = \frac{|B(D,q)|}{q}$ [55]. Furthermore, we have the following theorem proved in [55].

Theorem 4.3.1. *$\widehat{Count(D)}$ is an unbiased estimator of $Count(D)$. $\widehat{Count(D)}$ is an (ϵ, δ)-estimator of $Count(D)$ if sampling probability q satisfies $q \geq \frac{\phi^2_{\delta/2}}{inf(|D|)\epsilon^2 + \phi^2_{\delta/2}}$.* $\square$

Let $I(a,b)$ be a variable such that

$$I(a,b) = \begin{cases} 1 & \text{if a=b} \\ 0 & \text{otherwise} \end{cases}$$

Let $Equal(D,x)$ be the number of elements equal to value x in data set D, then we have

$$Equal(D,x) = |\{d_i \in D \mid d_i = x\}| = \sum_{d_i \in D} I(d_i,x).$$

The estimator of $Equal(D,x)$ is defined as

$$\widehat{Equal(D,x)} = \frac{1}{q} \sum_{d_i \in B(D,q)} I(d_i,x).$$

The following theorem indicates that $\widehat{Equal(D,x)}$ is an unbiased estimator of $Equal(D,x)$ and provides the variance of $\widehat{Equal(D,x)}$.

Theorem 4.3.2. *$\widehat{Equal(D,x)}$ is an unbiased estimator of $Equal(D,x)$ and $Var(\widehat{Equal(D,x)}) \leq \frac{1-q}{q} Equal(D,x)$.* $\square$

Proof: For all $1 \leq i \leq |D|$, let random variable X_i be

$$X_i = \begin{cases} 1 & \text{if } d_i \in B(D,q) \\ 0 & \text{if } d_i \notin B(D,q) \end{cases}$$

We have $E(X_i) = q$ and $Var(X_i) = q(1-q)$, then

$$E(\widehat{Equal(D,x)}) = \frac{E(\sum_{d_i \in D} I(d_i,x)X_i)}{q} = \frac{\sum_{d_i \in D} I(d_i,x)E(X_i)}{q} = Equal(D,x)$$

$$Var(\widehat{Equal(S,x)}) = \frac{\sum_{d_i \in D} I^2(d_i,x)Var(X_i)}{q^2} \leq \frac{1-q}{q} Equal(S,x). \quad \square$$

Theorem 4.3.3. *$\widehat{Equal(D}, x)$ is an (ϵ, δ)-estimator of $Equal(D, x)$ if sampling probability q satisfies $q \geq \frac{\phi^2_{\delta/2}}{\epsilon^2 inf(|D|) inf(F(D,x)) + \phi^2_{\delta/2}}$.* □

Proof: According to the center limit theory [57], if sample size $m \geq 30$, then

$$\widehat{Equal(D}, x) \sim N(E[\widehat{Equal(D}, x)], Var[\widehat{Equal(D}, x)])$$

where $N(E, V)$ denotes a normal distribution with expectation E and variance V. Since the required sample size is far more than 30 for most cases due to the large scale of a MANET, we can assume

$$\frac{\widehat{Equal(D}, x) - Equal(D, x)}{\sqrt{Var[\widehat{Equal(D}, x)]}} \sim N(0, 1).$$

Thus,

$$Pr(|\widehat{Equal(D}, x) - Equal(D, x)| \geq \phi_{\delta/2}\sqrt{Var[\widehat{Equal(D}, x)]}) = \delta.$$

Based on the condition, we have

$$\phi_{\delta/2}\sqrt{Var(\widehat{Equal(D}, x))} \leq \epsilon Equal(D, x).$$

Therefore,

$$Pr(\frac{|\widehat{Equal(D}, x) - Equal(D, x)|}{Equal(D, x)} \geq \epsilon) \leq \delta. \quad \square$$

4.3.2 Sampling Probability for Ordinary Frequency Queries

The estimator of $F(D, x)$ is defined as

$$\widehat{F(D, x)} = \frac{\widehat{Equal(D}, x)}{\widehat{Count}(D)}$$

where x is any distinct value in D. Using the similar strategy in [55], it is easy to have the following corollary.

Corollary 4.3.1. *If sampling probability $q < 1$, $\widehat{F(D, x)}$ is a biased estimator of $F(D, x)$. If* *both $\widehat{Equal(D,x)}$ and $\widehat{Count(D)}$ are the $(\frac{\epsilon}{2+\epsilon}, \frac{\delta}{2})$-estimator of $Equal(D,x)$ and $Count(D)$, then $\widehat{F(D,x)}$ is an (ϵ,δ)-estimator of $F(D,x)$.* $\square$

However, not all the nodes can communicate with the query node in one or multiple hops. At time t, we can only do data aggregation on $D(S',t)$. So we can only get the (ϵ,δ)-approximate ordinary frequency result of $D(S',t)$. Fortunately, the following corollaries offer a solution to this problem.

Data set $D(S',t)$ can be regarded as a sample of $D(S,t)$ without replacement with sampling ratio $\frac{|D(S',t)|}{|D(S,t)|} = \frac{|S'|}{|S|}$. $F(D(S',t),x)$ can be regarded as an estimator of $F(D(S,t),x)$. According to [81], it is easy to have the following corollary using the similar strategy in the proof of Theorem 4.3.3.

Corollary 4.3.2. *$F(D(S',t),x)$ is an unbiased estimator of $F(D(S,t),x)$. If we have $F(D(S',t),x)$ as an (ϵ,δ)-estimator of $F(D(S,t),x)$, then ϵ and δ satisfy that $\phi^2_{\delta/2}(\frac{1}{\inf(F(D(S,t),x))} - 1)(\frac{1}{|S'|} - \frac{1}{|S|}) \leq \epsilon^2$.* $\square$

Suppose we have $F(\widehat{D(S',t)},x)$ as an (ϵ_1,δ_1)-estimator of $F(D(S',t),x)$ and $F(D(S',t),x)$ as an (ϵ_2,δ_2)-estimator of $F(D(S,t),x)$, respectively. According to the definition of (ϵ,δ)-estimator, we have the following corollary.

Corollary 4.3.3. *$F(\widehat{D(S',t)},x)$ is an (ϵ,δ)-estimator of $F(D(S,t),x)$ if $(1-\delta_1)(1-\delta_2) \geq 1-\delta$, $(1+\epsilon_1)(1+\epsilon_2) \leq 1+\epsilon$ and $(1-\epsilon_1)(1-\epsilon_2) \geq 1-\epsilon$.* $\square$

Finally, according to the above theorems and corollaries, we have the final methodology to calculate sampling probability q according to the specified ϵ and δ which make $F(\widehat{D(S',t)},x)$ to be an (ϵ,δ)-estimator of $F(D(S,t),x)$. The detailed steps are listed below.

First, find a possible pair of ϵ_1 and δ_1 satisfying the following condition according to the input ϵ and δ: $(1-\delta_1)(1-\delta_2) \geq 1-\delta$, $(1+\epsilon_1)(1+\epsilon_2) \leq 1+\epsilon$, $(1-\epsilon_1)(1-\epsilon_2) \geq 1-\epsilon$, $\phi^2_{\delta_2/2}(\frac{1}{\inf(F(D(S,t),x))} - 1)(\frac{1}{|S'|} - \frac{1}{|S|}) \leq \epsilon_2^2$. Second, calculate q_1 which makes $\widehat{Count(D(S',t))}$ to be an $(\frac{\epsilon_1}{2+\epsilon_1}, \frac{\delta_1}{2})$-estimator of $Count(D(S',t))$. Third, calculate q_2 which makes $\widehat{Equal(D(S',t),x)}$ to be an $(\frac{\epsilon_1}{2+\epsilon_1}, \frac{\delta_1}{2})$-estimator of $Equal(D(S',t),x)$. Finally, $q = \max(q_1,q_2)$ is the final sampling probability.

4.3.3 Sampling Probability for Single Value Frequency and Range Frequency Queries

For the single value and range frequency queries, the estimator of the results, denoted by $\widehat{SF(D,x')}$ and $RCount(\widehat{D, Min}, Max)$ respectively, can be calculated as

$$\widehat{SF(D,x')} = \frac{\widehat{Equal(D}, x')}{\widehat{Count}(D)}$$

and

$$RF(D, \widehat{Min}, Max) = \frac{RCount(\widehat{D, Min}, Max)}{\widehat{Count}(D)}$$

where x' and $[Min, Max]$ are the user-specified value and range, $RCount(\widehat{D, Min}, Max) = \frac{1}{q}|\{d_i \in B(D,q) \mid d_i \in [Min, Max]\}|$ is the estimator of $RCount(D, Min, Max)$, and $RCount(D, Min, Max) = |\{d_i \in D \mid d_i \in [Min, Max]\}|$ is the exact number of elements belonging to $[Min, Max]$ in D.

The methods of determining the sampling probabilities for single value and range frequency queries are almost the same as the one introduced in Section 4.3.2. The only differences are the use of $\inf(SF(D(S,t),x'))$ to replace $\inf(F(D(S,t),x))$ for single value frequency queries, and the use of $\inf(RF(D, Min, Max))$ to replace $\inf(F(D(S,t),x))$ for range frequency queries when determining the sampling frequency, where $\inf(SF(D(S,t),x'))$ and $\inf(RF(D, Min, Max))$ are the lower bound of $SF(D(S,t),x')$ and $RF(D, Min, Max)$, respectively.

In most cases, we have $\inf(SF(D(S,t),x')) > \inf(F(D(S,t),x))$ and $\inf(RF(D, Min, Max)) > \inf(F(D(S,t),x))$. It means that we can use the same sampling probability for ordinary, single value and range frequency queries if they appear together, and ϵ and δ are the same. Therefore, much energy can be saved.

4.4 Network Connectivity

The calculation of sampling probability depends on $|S|$ and $|S'|$. In most cases, it is reasonable to assume $|S|$ is known since the number of mobile nodes is known before the MANET is deployed [82], but it is almost impossible to know the value of $|S'|$.

This problem can be solved by doing an exact $COUNT$ query on S' before carrying out an approximate frequency query. However, it consumes a huge amount of extra energy. Furthermore, the network topology possibly changes during the gap between the exact $COUNT$ query and the approximate frequency query.

Another solution is to use a formula to calculate $|S'|$ according to the related variables n, r, v_{min}, v_{max}, p_{min}, p_{max} and t. However, we have to confine ourselves to simulation results due to the intractability of analysis with the existing mathematical methods.

Moreover, it is still a hard problem to find the relationship between the exact value of $|S'|$ and the related variables n, r, v_{min}, v_{max}, p_{min}, p_{max} and t by simulation. Therefore, we only present the simulation results for the lower 0.05 fractile of $|S'|$ which is denoted by $f_{0.05}(|S'|)$. However, $f_{0.05}(|S'|)$ is not the exact value of $|S'|$. It is necessary to discuss the result of replacing $|S'|$ with $f_{0.05}(|S'|)$.

Obviously, if we have $f_{0.05}(|S'|) < |S'|$, the sampling probability will be greater than the necessary sampling probability. The final frequency query result still satisfies the specified ϵ and δ. But if we have $f_{0.05}(|S'|) \geq |S'|$, the final frequency query result may not necessarily satisfy the specified ϵ and δ. Fortunately, according to the definition of (ϵ, δ)-estimator, this problem can be solved by changing the original δ to a new $\delta' = 1 - \frac{1-\delta}{0.95}$.

Due to space limitation, in this chapter, we only present the conclusions based on the simulation results. The detailed simulation results can be downloaded at [83].

The first group of simulations is to find the relationship between $f_{0.05}(|S'|)$ and the factors of n, r, v_{min}, v_{max}, p_{min}, p_{max}, and t when $f_{0.05}(|S'|)$ becomes stable. The results indicate that if v_{min}, v_{max}, p_{min}, p_{max} or t is in a reasonable range and $f_{0.05}(|S'|) \geq 0.75n$, $f_{0.05}(|S'|)$ tends to be stable. The results also show that if we have $f_{0.05}(|S'|) \geq 0.75n$, the value of $f_{0.05}(|S'|)$ has almost nothing to do with v_{min}, v_{max}, p_{min}, p_{max} or t, but has much to do with n and r. Then the remaining work is just to find the relationship between $f_{0.05}(|S'|)$ and the factors of n and r. The impact of v_{min}, v_{max}, p_{min}, p_{max} or t can be ignored. For these variables, we can simply use a random value as the input in the simulations.

The second group of simulations is to find the smallest r causing $f_{0.05}(|S'|) \geq 0.75n$. The results indicate that $f_{0.05}(|S'|) > 0.75n$ if $r > \frac{6l}{5\sqrt{n}}$.

The third group of simulations is to find the smallest r causing $f_{0.05}(|S'|) \geq 0.99n$. The results indicate that $f_{0.05}(|S'|) > 0.99n$ if $r > \frac{11l}{5\sqrt{n}}$.

The fourth group of simulations is to find the relationship between $f_{0.05}(|S'|)$ and the factors of n and r under the condition of $\frac{6l}{5\sqrt{n}} < r \leq \frac{11l}{5\sqrt{n}}$. The results indicate that $f_{0.05}(|S'|)$ is closely mapped to $\frac{1}{4}n\sqrt{1 - n(r - \frac{11}{5\sqrt{n}})^2} + 0.74n$.

Finally, we have the following conclusions. First, $f_{0.05}(|S'|)$ is not stable if $r \leq \frac{6l}{5\sqrt{n}}$. Second, $f_{0.05}(|S'|)$ is closely mapped to $\frac{1}{4}n\sqrt{1 - n(r - \frac{11}{5\sqrt{n}})^2} + 0.74n$ if $\frac{6l}{5\sqrt{n}} < r \leq \frac{11l}{5\sqrt{n}}$. Third, $f_{0.05}(|S'|)$ is greater than $0.99n$ if $r > \frac{11l}{5\sqrt{n}}$.

4.5 (ϵ, δ)-Approximate Frequency Query Algorithms

To calculate an approximate frequency, the following steps are required.

1. Determine a proper sampling probability q according to the specified ϵ and δ.

2. Use a distributed Bernoulli sampling algorithm to sample sensory data.

3. Based on the sampled data, calculate the (ϵ, δ)-approximate frequency.

This section introduces the Bernoulli sampling algorithm and the algorithms for ordinary frequency queries, single value frequency queries and range frequency queries. Due to space limitation and high similarity, the algorithms for single value frequency queries and range frequency queries are introduced together.

4.5.1 The Bernoulli Sampling Algorithm

Because all the nodes in a network are moving and the communication radius is limited, the network topology is changing all the time. However, it is still possible to organize the whole network as a spanning tree using the similar method in [84]. Once a proper sampling probability q is determined, we can design the Bernoulli sampling algorithm as follows. First, the query node broadcasts sampling probability q to every node. Second, each node produces a random number $rand$ in range $[0, 1]$. Third, for each node, if $rand \leq q$, it sends its data to its parent node.

4.5.2 The (ϵ, δ)-Approximate Algorithm for Ordinary Frequency Queries

The basic idea of calculating an approximate frequency is as follows.

1. Each node in the spanning tree maintains Fre and Sub_Count. Fre records the number of the occurrences of each value, and Sub_Count records the number of the sensory data items.

2. The query node calculates the frequency according to the final Fre and Sub_Count.

The methodology of aggregating the partial result during the transmission is as follows.

1. Set Fre_j to $\emptyset$ and Sub_Count_j to 0 for each node j in the spanning tree.

2. Each leaf node sends its data to its parent node with probability q.

3. Each non-leaf node receives and merges data from its children, and adds its own data with probability q, then sends the result to its parent node.

The detailed algorithms are shown in Algorithm 4 and Algorithm 5 respectively. $\inf(F(D(S, t))$ and $\inf(F(D(S', t))$ are the lower bounds for $F(D(S, t))$ and $F(D(S', t))$, respectively. $f_{0.05}(|S'|)$ is the lower 0.05 fractile of $|S'|$. c is the number of children for node j in the spanning tree.

Using the similar strategy in [55], we have the communication cost and energy cost of the proposed algorithm as $O(\frac{|Fre|}{\epsilon_1^2} \ln \frac{1}{\delta_1})$. In practice, $|Fre|$ can be regarded as a constant. Then the communication cost and energy cost of the proposed algorithm can be simplified as $O(\frac{1}{\epsilon_1^2} \ln \frac{1}{\delta_1})$.

Algorithm 4: (ϵ, δ)-Approximate Frequency Algorithm

```
Input: $\epsilon, \delta$
Output: $(\epsilon, \delta)$-approximate frequency
1: Find $\epsilon_1$ and $\delta_1$ satisfying $(1-\delta_1)(1-\delta_2) \geq \frac{1-\delta}{0.95}$, $(1+\epsilon_1)(1+\epsilon_2) \leq 1+\epsilon$,
   $(1-\epsilon_1)(1-\epsilon_2) \geq 1-\epsilon$ and $\phi^2_{\delta_2/2}(\frac{1}{\inf(F(D(S,t)))} - 1)(\frac{1}{f_{0.05}(|S'|)} - \frac{1}{|S|}) \leq \epsilon_2^2$
2: $\epsilon_3 = \frac{\epsilon_1}{2+\epsilon_1}$, $\delta_3 = \frac{\delta_1}{2}$
3: $q_1 = \frac{\phi^2_{\delta_3/2}}{f_{0.05}(|S'|)\epsilon_3^2 + \phi^2_{\delta_3/2}}$
4: $q_2 = \frac{\phi^2_{\delta_3/2}}{\epsilon_3^2 f_{0.05}(|S'|)\inf(F(D(S',t))) + \phi^2_{\delta_3/2}}$
5: Sink node broadcasts $q = \max(q_1, q_2)$
6: Call SubData1 for each node in the spanning tree
7: Query node maintains $Fre$ and $Sub\_Count$, and divides each value in $Fre$ by
   $Sub\_Count$
8: return $Fre$
```

Algorithm 5: Submitting-Data Algorithm (SubData1)

```
Output: $(Fre, Sub\_Count)$
1: $Fre_j = \emptyset$, $Sub\_Count_j = 0$ for all $j$ in the spanning tree
2: for each leaf node $j$ in the spanning tree do
3:    if $rand < q$ then
4:       $Fre_j[j.data] = 1$, $Sub\_Count_j = 1$
5:       Send $(Fre_j, Sub\_Count_j)$ to its parent node
6:    end if
7: end for
8: for each non-leaf node $j$ in the spanning tree do
9:    Receive $\{(Fre_{jv}, Sub\_Count_{jv}) \mid 1 \leq v \leq c\}$ from its children
10:   $Fre_j = \cup_{v=1}^{c} Fre_{jv}$
11:   $Sub\_Count_j = \sum_{v=1}^{c} Sub\_Count_{jv}$
12:   if $rand < q$ then
13:      $Fre_j[j.data]++$
14:      $Sub\_Count_j++$
15:   end if
16:   if $j$ is the query node then
17:      return $(Fre_j, Sub\_Count_j)$
18:   end if
19:   Send $(Fre_j, Sub\_Count_j)$ to its parent node
20: end for
```

4.5.3 The (ϵ, δ)-Approximate Algorithms for Single Value Frequency Queries and Range Frequency Queries

The basic algorithms for single value and range frequency queries are almost the same as that for ordinary frequency queries except two major differences. The first difference is that the algorithm does not need to use Fre to record the number of the occurrences of each value. It only uses an integer Sum to record the number of the data items which satisfy the condition. The second difference is that it is necessary to judge whether the sensory data satisfies the condition rather than submit the sensory values directly. The detailed algorithms are shown in Algorithm 6 and Algorithm 7. For single value frequency queries, $Q(D(S,t)$ and $Q(D(S',t))$ denote $SF(D(S,t),x')$ and $SF(D(S',t),x')$. For range frequency queries, $Q(D(S,t)$ and $Q(D(S',t))$ denote $RF(D(S,t),Min,Max)$ and $RF(D(S',t),Min,Max)$.

Algorithm 6: (ϵ, δ)-Approximate Frequency Algorithm

```
Input: $\epsilon, \delta, x'$ for a single value frequency query, $Min, Max$ for a range frequency query
Output: $(\epsilon, \delta)$-approximate frequency
1: Find $\epsilon_1$ and $\delta_1$ satisfying $(1-\delta_1)(1-\delta_2) \geq \frac{1-\delta}{0.95}$, $(1+\epsilon_1)(1+\epsilon_2) \leq 1+\epsilon$,
   $(1-\epsilon_1)(1-\epsilon_2) \geq 1-\epsilon$ and $\phi^2_{\delta_2/2}(\frac{1}{\inf(Q(D(S,t)))}-1)(\frac{1}{f_{0.05}(|S'|)}-\frac{1}{|S|}) \leq \epsilon_2^2$
2: $\epsilon_3 = \frac{\epsilon_1}{2+\epsilon_1}$, $\delta_3 = \frac{\delta_1}{2}$
3: $q_1 = \frac{\phi^2_{\delta_3/2}}{f_{0.05}(|S'|)\epsilon_3^2+\phi^2_{\delta_3/2}}$
4: $q_2 = \frac{\phi^2_{\delta_3/2}}{\epsilon_3^2 f_{0.05}(|S'|)\inf(Q(D(S',t)))+\phi^2_{\delta_3/2}}$
5: Sink node broadcasts $q = \max(q_1, q_2)$
6: Call SubData2 for each node in the spanning tree
7: Query node maintains $Sum$ and $Sub\_Count$
8: return $Sum/Sub\_Count$
```

Using the similar strategy in [55], we have the communication cost and energy cost of the proposed algorithm as $O(\frac{1}{\epsilon_1^2}\ln\frac{1}{\delta_1})$.

4.6 Experimental Results

To evaluate the proposed algorithm, we employ a MANET with 5000 mobile nodes in our simulations. These nodes are deployed in a 1000m $\times$ 1000m square uniformly, randomly

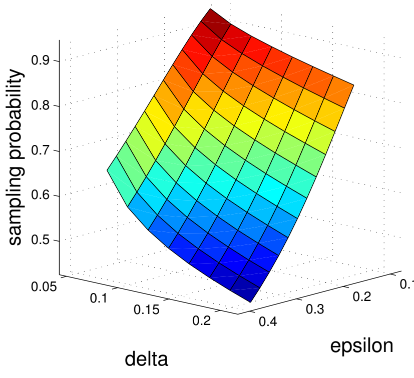
Algorithm 7: Submitting-Data Algorithm (SubData2)

```
Output: $(Sum, Sub\_Count)$
1: $Sum_j = Sub\_Count_j = 0$ for all $j$ in the spanning tree
2: for each leaf node $j$ in the spanning tree do
3:    if $rand < q$ and sensory data satisfies the condition then
4:       Send $(1, 1)$ to its parent node
5:    end if
6: end for
7: for each non-leaf node $j$ in the spanning tree do
8:    Receive $\{(Sum_{jv}, Sub\_Count_{jv}) \mid 1 \leq v \leq c\}$ from its children
9:    $Sum_j = \sum_{v=1}^{c} Sum_{jv}$
10:   $Sub\_Count_j = \sum_{v=1}^{c} Sub\_Count_{jv}$
11:   if $rand < q$ and sensory data satisfies the condition then
12:      $Sum_j ++$
13:      $Sub\_Count_j ++$
14:   end if
15:   if $j$ is the query node then
16:      return $(Sum_j, Sub\_Count_j)$
17:   end if
18:   Send $(Sum_j, Sub\_Count_j)$ to its parent node
19: end for
```

and independently. All the nodes are moving under the random waypoint model [80]. We set $v_{min} = 0$, $v_{max} = 500m/s$, $p_{min} = 0$, $p_{max} = 0.5s$, and $r = 20$m. The sensory data set is from the AADF (Estimated Annual average daily flows) Data Traffic Counts Metadata [85]. For each mobile node, the energy consumption to send and receive one byte of data is set to be 0.0144mJ and 0.0057mJ respectively according to [4]. The energy consumption for computation is omitted since it is much smaller than that of communication [86]. The simulation results for ordinary frequency queries, single value frequency queries and range frequency queries are presented in the following three subsections.

4.6.1 Sampling Probability and Energy Cost of Ordinary Frequency Query

The first group of simulations is to investigate the relationship among ϵ, δ and the sampling probability. The results are presented in Fig.4.1. The results show that the required sampling probability increases with the decrease of ϵ and δ. The sampling probability is



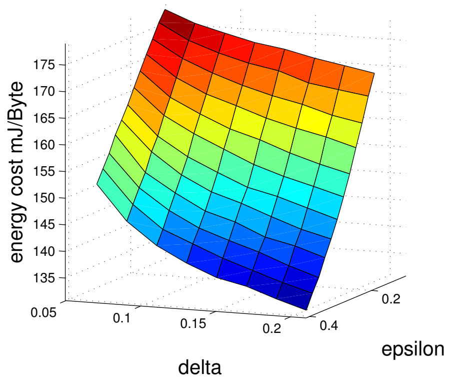


Figure 4.1. The relationship among ϵ, δ and sampling probability.

Figure 4.2. The relationship among ϵ, δ and energy cost.

relatively small. For example, when $\epsilon = 0.3$ and $\delta = 0.2$, the sampling probability is about 0.5. This means only about half of the mobile nodes in a MANET need to transfer data to get the required approximate result. Therefore, the proposed algorithm saves much energy since not all the sensory data are transmitted.

The second group of simulations is to investigate the relationship among ϵ, δ and the energy cost. The energy cost is calculated towards different values of ϵ and δ. The results are shown in Fig.4.2. It shows that the energy cost increases with the decrease of ϵ and δ since more data need to be sampled and transferred. For example, if $\epsilon = \delta = 0.2$, the energy consumption is 157mJ/Byte. The energy consumption is 173mJ/Byte if $\epsilon = \delta = 0.14$.

The third group of simulations is to investigate the relationship between sampling probability q and energy cost. The energy cost is calculated towards different sampling probabilities. The sampling probability goes from 0.04 to 0.4 at the step of 0.02. The results are shown in Fig.4.3. It can be seen that the energy cost increases with the increase of q. Furthermore, the increase of the energy cost is not proportional to the increase of q. The reason is that there is extra energy consumption for broadcasting the sampling probability. The results also show that the energy cost increases with the growth of network size when the sampling probability q keeps the same.

The fourth group of simulations is to investigate the relationship among sampling proba-

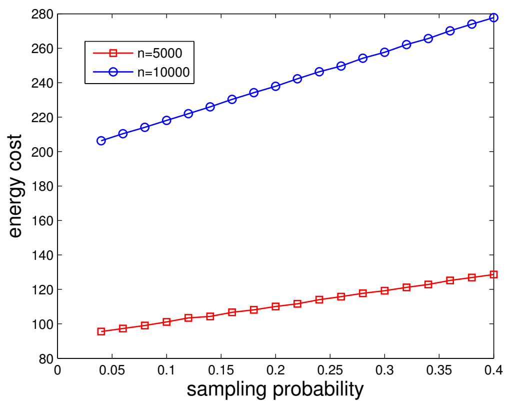


Figure 4.3. The relationship between sample probability and energy cost.

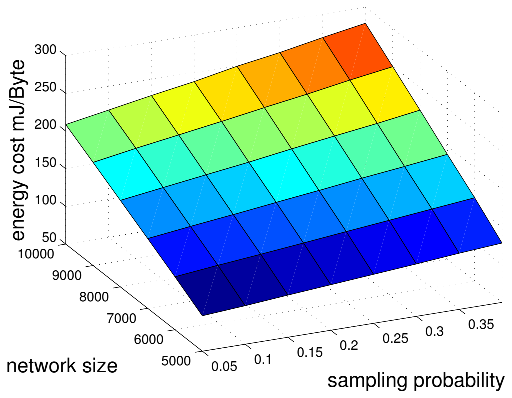


Figure 4.4. The relationship among sampling probability, network size and energy cost.

bility, network size and energy cost. The energy cost is calculated towards different sampling probabilities and network sizes. The results are shown in Fig.4.4. For the same network size, the energy cost increases with the increase of sampling probability. Similarly, the increase of energy cost is not proportional to the increase of sampling probability. For the same sampling probability, the energy cost increases with the increase of network size since more data are sampled and transmitted.

The fifth group of simulations is to compare the energy consumption of our algorithm with that of the naive algorithm. The naive algorithm is to collect all the sensory data in S' and aggregate the partial results during the transmission of the sensory data. The results are shown in Fig.4.5. It can be seen that our algorithm has smaller energy consumption since not all the mobile nodes in a network need to transfer their own sensory data. The energy consumption increases with the decrease of ϵ and δ. Furthermore, although the naive method collects all the data in S', it only returns the accurate frequency results of $D(S', t)$ rather than $D(S, t)$.

4.6.2 Relative Error of Ordinary Frequency Queries

The first group of simulations is to investigate the relationship among ϵ, δ and relative error of the final result. The results are shown in Fig.4.6. We can see that our algorithm can

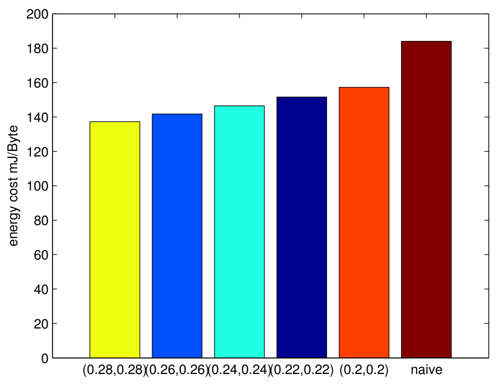


Figure 4.5. Energy cost comparison.

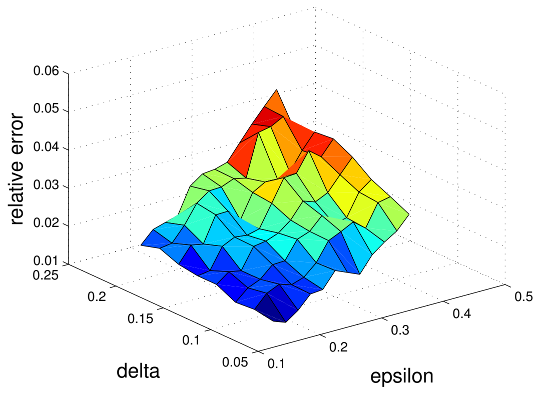


Figure 4.6. The relationship among ϵ, δ and relative error.

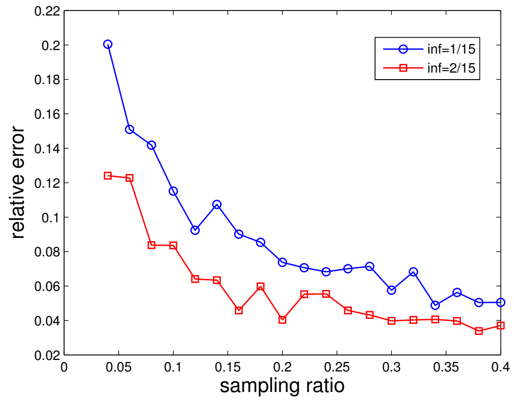


Figure 4.7. The relationship between sampling probability and relative error.

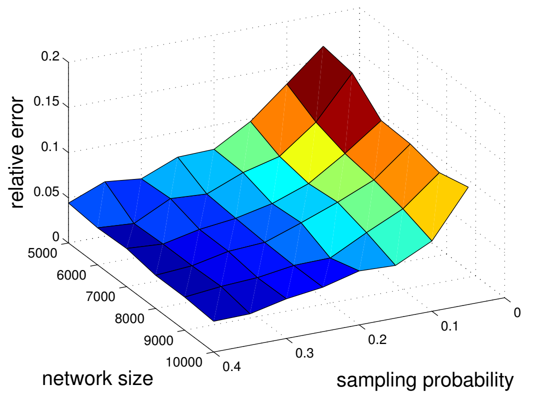


Figure 4.8. The relationship among sampling probability, network size and relative error.

achieve the specified precision. In most cases, the relative error increases with the increase of ϵ and δ since less sensory data are sampled in the network.

The second group of simulations is to investigate the relationship between sampling probability and relative error of the final result. The results are shown in Fig.4.7. We can see that in most cases, the relative error decreases with the increase of the sampling probability since more sensory data are sampled. We can also observe that the precision is still high even when the sampling probability is relatively low. The results also indicate the relative error decreases when frequency lower bound becomes larger.

The third group of simulations is to investigate the relationship among network size, sampling probability and relative error of the final result. The results are shown in Fig.4.8. Similar to the previous group of simulations, for the same network size, the relative error decreases with the increase of sampling probability. For the same sampling probability, the relative error decreases with the increase of network size since more data are sampled.

4.6.3 Simulation Results for Single Value Frequency Queries

The first group of simulations is to investigate the relationship among ϵ, δ and energy cost. The energy cost was calculated while ϵ varies from 0.14 to 0.36 and δ varies from 0.06 to 0.2. The results are shown in Fig.4.9. It shows that the energy cost increases with the decrease of ϵ and δ since more data need to be sampled and transmitted. Moreover, for the same ϵ and δ, the single value frequency queries have lower energy cost than ordinary frequency queries since only the sensory data equal to x' needed to be submitted.

The second group of simulations investigates the relationship between the network size and the energy cost where ϵ and δ were set to be 0.2. The results are shown in Fig.4.10. It shows that the energy cost increases with the growth of network size. It is strange that energy cost is even higher when frequency lower bound is greater which is because there will be more sensory data equal to x' even though lower sampling probability is needed. Other simulation results about single value frequency queries are similar to ordinary ones. So that these results are not presented in the chapter due to space limitation.

4.6.4 Simulation Results for Range Frequency Queries

The first group of simulations is to investigate the relationship among ϵ, δ and sampling probability. The results are shown in Fig.4.11. It shows that the required sampling probability increases with the decrease of ϵ and δ and the sampling probability is relatively small. Moreover, for the same ϵ and δ, the sampling probability is smaller than that of the ordinary frequency queries. The reason is that we have $\inf(RF(D, Min, Max)) > \inf(F(D(S,t), x))$ for range frequency queries in most cases.

The second group of simulations is to investigate the relationship among ϵ, δ and energy

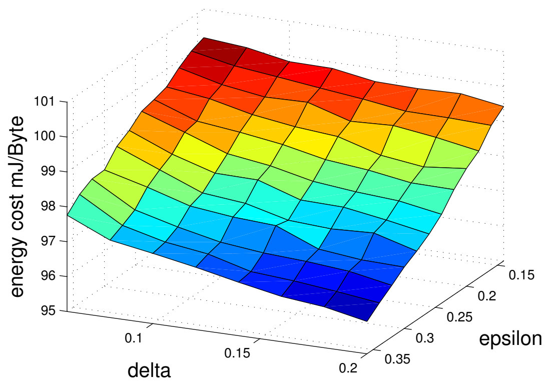


Figure 4.9. The relationship among ϵ, δ and energy cost.

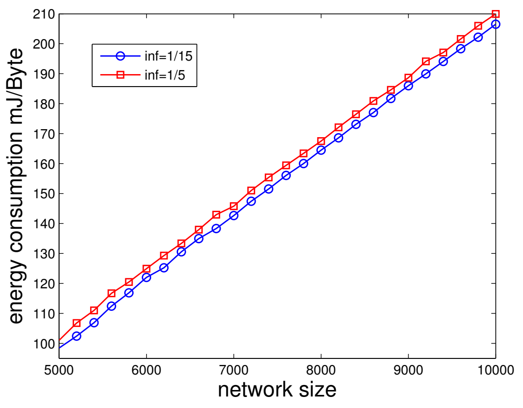


Figure 4.10. The relationship between network size and energy cost.

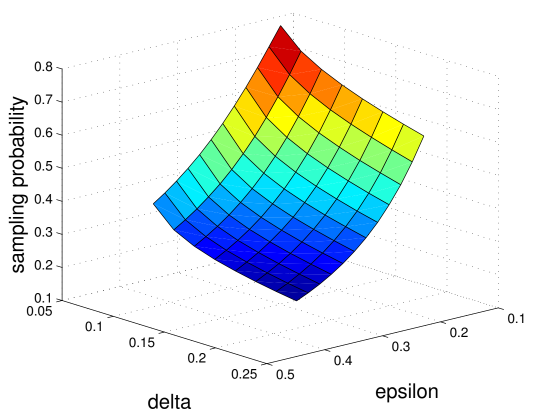


Figure 4.11. The relationship among ϵ, δ and sampling probability.

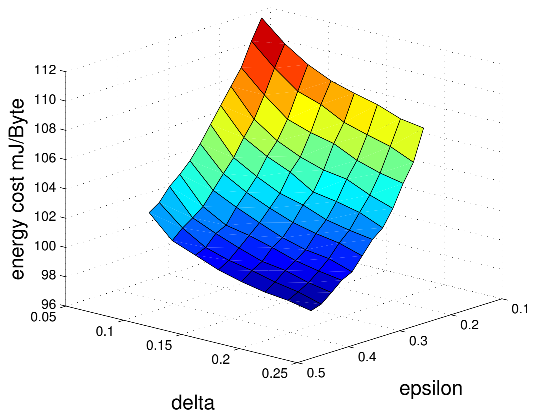


Figure 4.12. The relationship among ϵ, δ and energy cost.

cost. The results are shown in Fig.4.12. It shows that the energy cost increases with the decrease of ϵ and δ since more data need to be sampled and transferred. Furthermore, for the same ϵ and δ, the energy cost is smaller than that of the ordinary frequency queries. The first reason is that we have smaller sampling probability required for range frequency with the same ϵ and δ. The second reason is that we need to judge if the sensory data are in the user-specified range. The sensory data out of the range will not be transmitted.

The third group of simulations is to investigate the relationship between sampling probability and relative error of the final result. The results are shown in Fig.4.13. We can see that in most cases, the relative error decreases with the increase of sampling probability since

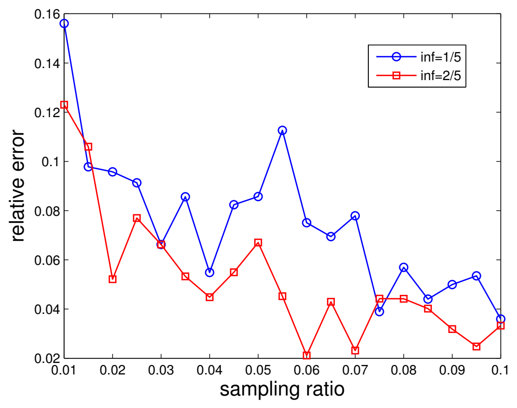


Figure 4.13. The relationship between sampling probability and relative error.

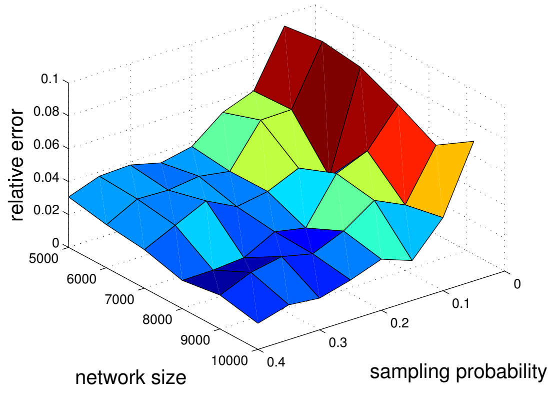


Figure 4.14. The relationship among sampling probability, network size and relative error.

more sensory data are sampled. Moreover, for the same sampling probability, the relative error is lower than that of the ordinary frequency queries. The results also indicate the relative error decreases for greater frequency lower bound.

The fourth group of simulations is to investigate the relationship among network size, sampling probability and relative error of the final result. The results are shown in Fig.4.14. Similar to the previous group of simulations, the relative error decreases with the increase of sampling probability for a same network size. The relative error decreases with the increase of network size for a same sampling probability. Furthermore, for a same sampling probability and network size, the relative error is lower than that of ordinary frequency queries.

4.7 Conclusion

In this chapter, an (ϵ, δ)-approximate algorithm to process a frequency query is proposed. This algorithm is based on Bernoulli sampling, so only some nodes in a network need to transfer sensory data. Furthermore, the methodology to calculate the sampling probability which can make the final query result to satisfy the specified precision and failure probability is derived. The simulation results for MANET connectivity are presented. These simulation results can be used in the calculation of sampling probability. Finally, a Bernoulli sampling algorithm is provided and analyzed. The simulation results show that on the aspects of both energy efficiency and accuracy, the proposed algorithm has high performance.

Chapter 5

DATA COLLECTION IN GEOGRAPHICAL POSITION CONFLICTING MOBILE SENSING

5.1 Introduction

Nowadays, smartphones have become ubiquitous and are playing a critical role in key aspects of people's daily life such as communication, entertainment and social activities [87] [88] [89] [90] [91] [92] [93] [94]. Most smartphones are equipped with multiple embedded sensors such as GPS (Global Positioning System), accelerometer, camera, etc [95], and have diverse sensing capacity. Moreover, the emergence of wearable devices also enhances the sensing capabilities of smartphones since most wearable devices can exchange sensory data with smartphones via network interfaces. Therefore, mobile sensing have led to numerous innovative applications in various fields including environmental monitoring, transportation, healthcare, safety and so on [1] [96] [97] [98] [99] [100].

Mobile crowdsensing which makes use of the sensing capabilities of smartphone has been gaining increasing popularity in recent years. Multiple works such as [101] and [102] have studied the problem of incentive mechanism design in mobile crowdsensing platforms. Unlike traditional applications proposed in the previous chapters, mobile crowdsensing platforms are not based on voluntary participation. Mobile crowdsensing task participants get paid for their contribution in return for their own resource consumption including battery, computing power and data transition fee when participating in a sensing task. For instance in multiple general-purpose mobile crowdsensing platforms such as Medusa [103], the following process is used to gather sensory data.

1. The mobile sensing platform broadcasts the description of a mobile sensing task to smartphone users.

2. The smartphone users who are willing to participate in the mobile sensing task transmit feedback to the mobile crowdsensing platform. A feedback can be a detailed sensing plan or a bid which is the expected price at which the user wants to sell the sensory data.

3. The mobile sensing platform selects some smartphone users as winners for participating in the mobile sensing task and make payments to them.

4. The winners transmit their senory data to the mobile sensing platform.

In real-life applications, during the above process, the mobile crowdsensing platforms aim to maximize the quality of gathered sensory data, since significant economic costs are paid to the mobile crowdsensing winners for their sensory data. However, users close to each other geographically will provide similar sensory data due to the spatial correlation [25], [26] of the sensing targets. The crowdsensing platform may get enormous amount of similar, or even duplicated sensory data if it simply gathers all the sensory data from the crowdsensing participants. Such situation not only leads to unnecessary data collection but also significant economic waste. Therefore, it is a critical problem to design incentive mechanisms which can avoid the above situation. Unfortunately, this problem has not been taken into consideration by existing works.

Our chapter focuses on incentive mechanisms in the geographical position conflicting mobile crowdsensing platforms, in which two users are defined as conflicting users if the geographical distance between them is within a predefined threshold. The mobile crowdsensing platform will not make payments to conflicting users simultaneously. In the design and implementation of such mobile crowdsensing platform naturally arise two problems. First, how to select the winner set for a mobile crowdsensing task to maximize the mobile crowdsensing platform's profit while avoiding simultaneous participation of conflicting users? Second, how to calculate appropriate payments to the winners satisfying beneficial properties including truthfulness, individual rationality and profitability? This chapter proposes algorithms to solve the above two problems with different computation efficiency. In summary, the main contributions of this chapter are as follows.

1. A geographical position conflict based winner selection problem is proposed and formulated as an integer linear programming problem.

2. Two algorithms are proposed to select winners and calculate their payments. The first algorithm is an A* algorithm based on a polynomial-time approximation scheme, and it maximizes the total social welfare. The second algorithm achieves sub-optimal total social welfare and has polynomial time complexity.

3. Solid theoretical proofs are listed which indicate the proposed algorithms have beneficial properties such as truthfulness, individual rationality and profitability.

4. Actual datasets based extensive experimental results are presented, which demonstrate the proposed algorithms have high performance and confirm the beneficial properties of these algorithms.

The rest of this chapter is organized as follows. Section 5.2 present the problem definition. The optimal and non-optimal winner selection algorithms are introduced in Section 5.3 and Section 5.4, respectively. Section 5.5 provides the experimental results. Finally, Section 5.6 concludes this chapter.

5.2 Problem Definition

5.2.1 Mobile Crowdsensing Platform

Suppose there are n users in a region $[0,1)^2$ and $U = \{u_1, u_2, \ldots, u_n\}$ is used to denote the set of users. Each user is carrying a smart phone equipped with positioning device to acquire user's position. Here $u.x$ and $u.y$ are used to denote the x-coordinate and y-coordinate for user u. We denote the circular region around position (x, y) with radius r by $R(x, y, r)$ which is defined as

$$R(x, y, r) = \{(x', y') \in [0,1)^2 | (x' - x)^2 + (y' - y)^2 \leq r^2\}.$$

Suppose a snapshot sensing task T is proposed by the mobile crowdsensing platform which invite users to submit sensory data in a sensory data gathering region $R(T.x, T.y, T.r)$. $T.x$ and $T.y$ are the x-coordinate and y-coordinate of the center of $R(T.x, T.y, T.r)$ and $T.r$ is the radius of $R(T.x, T.y, T.r)$. We use

$$U_T = \{u \in U | (u.x, u.y) \in R(T.x, T.y, T.r)\}$$

to denote all the users in the sensory data gathering region $R(T.x, T.y, T.r)$. Once the crowdsensing platform broadcasts the crowdsensing task T, all the users in U will receive the crowdsensing task T. However, only users within U_T will participate in the sensing task T due to the time-sensitiveness of the snapshot sensing task and the massive moving cost for users in U/U_T to enter the data gathering region [17]. The mobile crowdsensing platform will collect sensory data using the following process.

1. The crowdsensing platform broadcasts the crowdsensing task T.
2. Each user $u \in U_T$ submits his/her position $(u.x, u.y)$ and bid $u.b$ to the mobile crowdsensing platform. The bid $u.b$ is the expected price user u wants to sell the sensory data for.
3. The mobile crowdsensing platform decides the winner set W, calculates the payment to each winner $w \in W$, then sends the payments to the winners.
4. All the winners submit their sensory data to the mobile crowdsensing platform.

The above process is illustrated in Fig.5.1.

Let P denote the profit of the crowdsensing platform generated from a single user's sensory data. Then the total social welfare of the mobile crowdsensing platform is

$$P|W| - \sum_{w \in W} w.b$$

where $|W|$ is the size of set W.

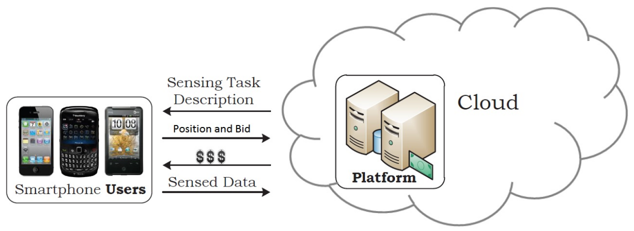


Figure 5.1. Mobile crowdsensing platform

5.2.2 Optimal Winner Selection Problem

According to the basic setting of the mobile crowdsensing platform introduced in Section 5.2.1, the mobile crowdsensing platform should simply select the all the users whose bids are lower than P as the winners to maximize the total social welfare. However, in many applications, the sensory data exhibits strong spatial correlation [25], [26], which means users who are very close to each other will submit similar or even duplicate sensory data. To save economic cost of the mobile crowdsensing platform, it is not necessary to make payments to users simultaneous if their positions are too close in the physical world. Therefore, we have the following definition of conflicting users.

Definition 5.2.1 (conflicting users)**.** User u and user v are conflicting users if and only if $dis(u, v) \leq T.d$ where is the Euclidean distance between user u and user v and $T.d$ is a predefined minimum graphical distance for the winners in crowdsensing task T.

Definition 5.2.2 (conflict-free winner set)**.** A winner set W is conflict-free if and only if $\nexists u, v \in W$ where u and v are conflicting users.

During the process of deciding the winners, the mobile crowdsensing platform should not declare both user u and user v as winners if they are conflicting users. Based on this winner determination setting, we address the problem of optimal winner selection as follows.

Input: user set U where each user $u \in U$ has attributes $u.x$, $u.y$ and $u.b$.

Output: 1) Conflict free winner set W which maximizes $P|W| - \sum_{w \in W} w.b$. 2) The payment for each winner.

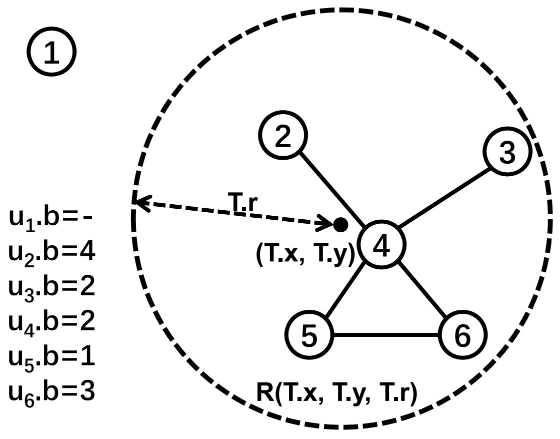


Figure 5.2. Example of the winner selection problem

Fig.5.2 shows an example of our problem. Suppose we have user set $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$. The circle is used to denote the sensory data gathering region. Suppose u_4 conflicts with u_2, u_3, u_5, u_6 and u_5 conflicts with u_6. According to the sensory data gathering region, we have $U_T = \{u_2, u_3, u_4, u_5, u_6\}$, and u_1 cannot be a winner because $u_1 \notin U_T$. Moreover, u_2 and u_4, u_3 and u_4 cannot be the winners simultaneously since u_2 and u_4, u_3 and u_4 are conflicting users. Similarly, at most one user among u_4, u_5 and u_6 can be a winner. If we have $P = 3$, $u_1.b = 1$, $u_2.b = 4$, $u_3.b = 2$, $u_4.b = 2$, $u_5.b = 1$, $u_6.b = 3$, then the optimal winner set is $\{u_3, u_5\}$ with social welfare as 3.

Our objective is to design incentive mechanisms to select the winners and calculate winners' payments which satisfy the following properties:

1. Computational efficiency: In practice, the number of crowdsensing participants may be quite large [2], [3]. For a mobile crowdsensing platform to carry out winner selection and payment determination in a short percoid of time, the incentive mechanism needs to be computationally efficient. A mechanism is computationally efficient if the outcome can be computed in polynomial time.

2. Truthfulness: During the whole process of crowdsensing, it is likely that multiple users submit bids which are different from their true valuations to improve their utility, which may greatly affect the fairness of the crowdsensing. Therefore, the incentive

mechanism needs to be truthful. A mechanism is truthful if no bidder can improve its utility by submitting a bid different from its true valuation, no matter what others submit.

3. Individual rationality: During the whole process of crowdsensing, the participants incur additional costs for battery, computing power, data transition fee, etc. In order to encourage user participation, the incentive mechanism needs to be individually rational which means each participant should have a non-negative utility.

4. Profitability: The crowdsensing platform should not incur a deficit. In other words, the value brought by the winners should be no less than the total payment paid to the winners.

5.2.3 Problem Formulation

We use boolean variable x_i to denote if user u_i is the winner or not.

$$x_i = \begin{cases} 1 & \text{if } \ u_i \in W \\ 0 & \text{if } \ u_i \notin W \end{cases}$$

Then the optimal winner selection problem can be formalized as the following integer programming problem.

$$\max \quad P\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i b_i$$

$$\text{s.t.} \quad x_i \in \{0,1\}, 1 \leq i \leq |U|$$

$$x_i + x_j \leq 1 \text{ if } dis(u_i, u_j) \leq T.d \text{ for } 1 \leq i, j \leq |U|.$$

In order to increase the readability, we summarized frequently used symbols in Table 5.1.

Symbol	Description
$u.x$	x-coordinate of user u's position
$u.y$	y-coordinate of user u's position
$u.b$	bid of user u
$u.w$	weight of user u defined as $u.w = P - u.b$
$u.p$	payment of user u, $u.p = 0$ if u is not a winner
$T.x$	x-coordinate of sensory data gathering region center
$T.y$	y-coordinate of sensory data gathering region center
$T.r$	radius of sensory data gathering region
$T.d$	minimum distance between any two winners
$dis(u,v)$	Euclidean distance between user u and user v
$R(x,y,r)$	$\{(x',y') \in [0,1)^2 \mid (x'-x)^2 + (y'-y)^2 \leq r^2\}$
U	the set of users
U_T	the set of users in the data gathering region of task T
W	the set of winners
P	the profit brought to the platform by one user's data

Table 5.1. Symbols

5.3 Optimal Winner Selection Algorithm

5.3.1 Problem Representation

According to the problem definition in Section 5.2, the conflict relationships among users can be represented by a unit disk graph $G = (U_T, E)$ [104]. For any $u, v \in U_T$, we have $(u,v) \in E$ if $dis(u,v) \leq T.d$. We also assign a weight $u_i.w = P - u_i.b$ for each user $u_i \in U_T$. Then the optimal winner selection problem can be represented as a maximum weighted independent set problem on a unit disk graph $G = (U_T, E)$ [105]. For example, the winner selection problem shown in Fig.5.2 can be presented as a maximum weighted independent set problem of the graph shown in Fig.5.3, where each number near a node represents the weight of that node.

Obviously, there is a brute-force algorithm for the optimal winner selection problem, and the process is as follows.

1. Enumerate all the subsets of U_T.

2. Remove the subsets with conflicts, *i.e.* the subset with user u, user v and $(u,v) \in E$.

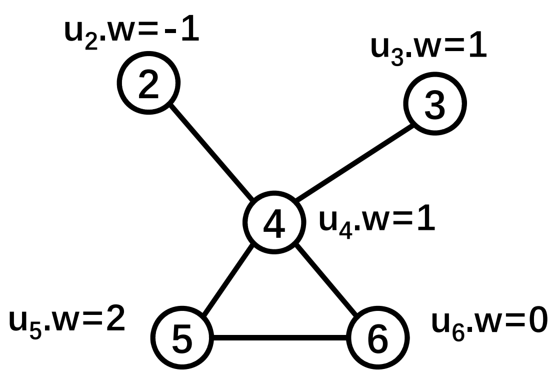


Figure 5.3. Example of maximum weighted independent set problem

3. Find the subset which maximizes the social welfare.

Moreover, we can also directly solve the integer programming problem shown in Section 5.2.3. However, all these algorithms have huge computation cost since they need to examine all the possible winner combinations. Therefore, we propose the following A* algorithm to select winners with maximal social welfare efficiently.

5.3.2 A* Algorithm for Optimal Winner Selection

Our A* algorithm is based on a robust Polynomial-Time Approximation Scheme (PTAS) proposed in [106]. The ratio bound and computation complexity of the PTAS in [106] are $1+\epsilon$ and $O(n^{1/\epsilon^2 \log 1/\epsilon})$, where ϵ is a real number which can be arbitrarily small. According to the definition of ratio bound [107], we have the following theorem.

Theorem 5.3.1. *For any graph $G = (V, E)$, suppose V_{opt} is the maximum weighted independent set of G and V_{app} is the solution of the PTAS on graph G, then we have*

1. *$\sum_{v \in V_{app}} v.w$ is the lower bound of $\sum_{v \in V_{opt}} v.w$*

2. *$(1+\epsilon)\sum_{v \in V_{app}} v.w$ is the upper bound of $\sum_{v \in V_{opt}} v.w$*

where $v.w$ is the weight of node v.

According to the above theorem, the PTAS in [106] plays the following roles in our optimal winner selection algorithm:

1. The PTAS is used to calculate the lower bound of maximum independent set's total weight at the beginning of our algorithm. Then our algorithm can make use of the lower bound for pruning.

2. The PTAS is used to calculate the upper bound of the total weight for an induced subgraph's maximum independent set, and this establishes the essential process for the heuristic function of the A* algorithm.

Since the ratio bound of the PTAS is $1 + \epsilon$ where ϵ is a real number which can be arbitrarily small, using the PTAS can make accurate estimations for both the lower bound and the upper bound of maximum independent set's total weight. Therefore, the time complexity of the A* algorithm can be further reduced. The above conclusion has been verified by the experiment result shown in Fig. 5.10.

Our algorithm uses a 0-1 bit string to denote a conflict-free winner set. For any given 0-1 bit string str and node set U where all the nodes are sorted in ascending order of their IDs, $str[i] = 1$ if $U[i]$ is the winner. For the example shown in Fig.5.3, 0-1 bit string "01010" is used to denote the optimal winner set $\{u_3, u_5\}$ (note that u_1 is ignored in the 0-1 bit string). Then the main job of our algorithm is to find a 0-1 bit string to denote the optimal winner set.

In our algorithm, Q is used to denote the priority queue in the A* algorithm which contains multiple 0-1 bit strings. All the elements in Q are sorted according to the heuristic function introduced in Algorithm 9. Q.top is used to denote the element in Q with the highest priority. U'_T is used to denote the node set which contains all the nodes in U_T with positive weights. Moreover, for any node set $V \subseteq U'_T$, $ISG(V) = G(V, \{(u, v)|u \in V \wedge v \in V \wedge dis(u, v) \leq T.d\})$ is used to denote the induced subgraph of node set V. Finally, we have our A* algorithm to compute the optimal winner selection with basic idea as follows.

1. Calculate the lower bound of maximum independent set's total weight. Then push an empty 0-1 bit string into the priority queue Q.

2. While Q is not empty and $|Q.\text{top}| < |U'_T|$, pop Q.top as q, then examine $q+$"0" and

$q+$"1". For each of them, compute its score using the heuristic function, if the score is no less than the lower bound computed in step 1 and the examined string is conflict-free, push the examined string into Q.

3. Compute the final winner set according to Q.top when the above step is finished.

The detailed process is shown in Algorithm 8 where LB denotes the lower bound of maximum independent set's total weight. HF in Algorithm 8 denotes the heuristic function of the A* algorithm whose input is a 0-1 bit string str. Here str stands for a set of possible winner sets.

According to the definition of A* algorithm, the HF function needs to calculate upper bound of maximum independent set's total weight. A straightforward approach to solve this problem is to calculate the sum of all the nodes' weights without considering the conflict between them. But the above method leads to inaccurate estimation of the upper bound. Therefore, in our algorithm, we use a PTAS-based heuristic function whose basic procedure is as follows.

1. Calculate the total weight for the users who have already been chosen as the winner.

2. For the undecided users, the heuristic function uses the PTAS to calculate the upper bound of total weight for the maximum weighted independent set.

The detailed process of the heuristic function is shown in Algorithm 9. In our A* algorithm, the HF function is used in the following two scenarios. First, it is used to calculate the priorities of the elements in the priority queue Q in order to sort those elements. Second, the HF function is used for pruning of the A* algorithm.

Fig.5.4 shows the execution process of Algorithm 8 which is used to solve the maximum weighted independent set problem in Fig. 5.3 where $\epsilon = 0.1$. Initially, we have $U'_T = \{u_3, u_4, u_5\}, LB = 3$. Node 1 is the root node of the search tree. Algorithm 1 will extend node 1 and get node 2, node 3. Node 2 will not be further extended since the value of heuristic function is less than LB. By extending node 3, we get node 4 and node 5. Then

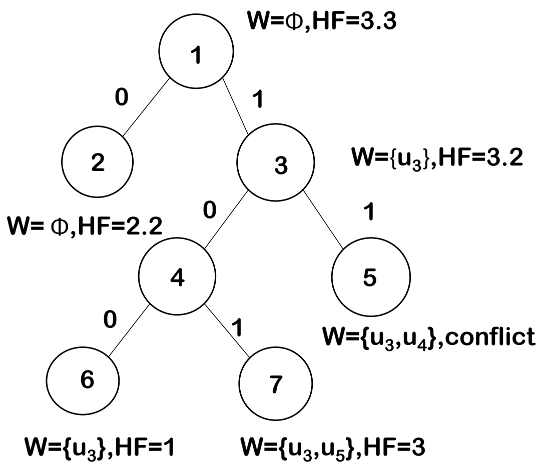


Figure 5.4. Searching tree

node 5 will not be further extended due to conflict. By extending node 3, we have node 6 and node 7. Finally, we get the optimal solution $\{u_3, u_5\}$ at node 7.

For the optimal winner selection algorithm, we can see that in the 14th step, it checks if the winner set represented by q_1 is conflict free. This step ensures that the final winner set does not contain conflict users.

5.3.3 Payment Determination

Once the optimal winners are selected, we will use the VCG (VickreyCClarkeCGroves) auction to calculate the payment for each winner [108]. The detailed process is shown in Algorithm 10.

According to the process of Algorithm 8, we can easily find that the winner set W calculated by Algorithm 8 is conflict free and it maximizes the total social welfare. Moreover, according to [108], we can conclude that the payment strategy is truthful, individual rational and profitable.

Algorithm 8: Optimal Winner Selection Algorithm Opt(U_T)

```
Input: User set $U_T$
Output: Winner set $W$
1: $U'_T = \{u \in U_T | u.w > 0\}$
2: $V_{app}$ =PTAS($ISG(U'_T)$)
3: $LB = \sum_{v \in V_{app}} v.w$
4: $Q$.push(“”)
5: while $Q \neq \emptyset$ do
6:    if $|Q.\text{top}| = |U'_T|$ then
7:       break
8:    end if
9:    $q = Q.\text{top}$, $Q$.pop()
10:   $q_0 = q +$“0”, $q_1 = q +$“1”
11:   if HF($q_0, U'_T$) $\geq LB$ then
12:      $Q$.push($q_0$)
13:   end if
14:   if $q_1$ is conflict-free and HF($q_1, U'_T$) $\geq LB$ then
15:      $Q$.push($q_1$)
16:   end if
17: end while
18: $W = \emptyset$
19: for $i$=1 to $|U'_T|$ do
20:    if $Q$.top[i]=1 then
21:       $W = W \cup \{U'_T[i]\}$
22:    end if
23: end for
24: return $W$
```

Algorithm 9: Heuristic Function HF

```
Input: 0-1 bit string $str$ and user set $U'_T$
Output: Upper bound of maximum independent set's total weight if $str$ is further
    extended
1: $V_1 = \{U'_T[i] | 1 \leq i \leq |str| \wedge str[i] = 1\}$
2: $V_2 = \{U'_T[i] | |str| + 1 \leq i \leq |U'_T| \wedge dis(U'_T[i], v) > T.d$ for any $v \in V_1\}$
3: $ub_1 = \sum_{v \in V_2} v.w$
4: $V_3$=PTAS($ISG(V_2)$)
5: $ub_2 = (1+\epsilon) \sum_{v \in V_3} v.w$
6: return min($ub_1, ub_2$) $+ \sum_{v \in V_1} v.w$
```

Algorithm 10: Payment Determination Algorithm

```
Input: Winner set $W$ and $U'_T$
Output: Payment for each winner $u \in W$
1: $maxw = \sum_{u \in W} u.w$
2: for $u \in W$ do
3:     $W$'=Opt($U'_T - \{u\}$)
4:     $maxw' = \sum_{v \in W'} v.w$
5:     $u.p = maxw - maxw'$
6: end for
7: return $W$
```

5.4 Non-optimal Winner Selection Algorithm

5.4.1 Algorithm Design

First, we introduce the following theorem proposed in [104].

Theorem 5.4.1. *The maximum weighted independent set problem in unit disk graphs is NP-hard.*

Since the A* algorithm proposed in the previous section can maximize the total social welfare, we have the following corollary.

Corollary 5.4.1. *The optimal winner selection algorithm has exponential time complexity.*

The above corollary indicates that the optimal winner selection algorithm still has high computational complexity although it uses PTAS in [106] to further reduce the computation cost. However, the number of users can be quite large for multiple representative samples of crowdsensing applications [2], [3]. Therefore, it is critical to propose a computationally efficient winner selection algorithm even if the algorithm may not maximize the total social welfare.

A straightforward approach to solve the above mentioned problem is to use the PTAS in [106] to calculate an approximate solution for the optimal winner selection problem, and then use VCG auction to calculate the payment. However, [109] indicates VCG auction loses its truthful property when applied along with an approximation algorithm to compute resource allocation. Therefore, we propose another truthful winner selection methodology based on a theorem in [110], which is listed as follows.

Theorem 5.4.2. *An auction mechanism is truthful if and only if:*

1. *The selection rule is monotone: if user i wins the auction by bidding b_i, it also wins by bidding $b'_i \leq b_i$.*

2. *Each winner is paid the critical value: user i would not win the auction if it bids higher than this value.*

Based on the above theorem, we propose an algorithm to calculate the winner set W and the payment for each winner according to user set U and mobile crowdsensing task T. The basic idea of our proposed algorithm is explained as below.

Obviously, it is not necessary to consider the users whose bid is greater or equal to P. Therefore, first the crowdsensing platform calculates user set U'_T which contains all the users in the sensory data gathering region whose bid is less than P according to U and T. Second, the crowdsensing platform set $W = \emptyset$. Then the crowdsensing platform sorts all the users in U'_T in ascending order of their priorities. The priority is defined as follows.

1. If two users have different bids, the user with a lower bid has a higher priority.

2. If two users have the same bid, the user with a lower ID has a higher priority.

Finally, the crowdsensing platform checks each user $u \in U'_T$ using the following strategy where $U'_T[i]$ is used to denote the user with the i-th lowest bid in U'_T.

1. Add $U'_T[i]$ into winner set W if no user in W conflicts with $U'_T[i]$.

2. Set $U'_T[i]$'s payment $U'_T[i].p$ to $\min\{u.b | u \in U'_T \wedge u.b > w.b \wedge dis(u, w) < T.d\}$ if $\{u.b | u \in U'_T \wedge u.b > w.b \wedge dis(u, w) < T.d\} \neq \emptyset$. Otherwise, set $U'_T[i]$ to P.

3. Update the users in U'_T by removing all the users who conflict with $U'_T[i]$ and bid greater than $U'_T[i]$'s bid.

Algorithm 11: Non-optimal Winner Selection Algorithm

```
Input: User set $U_T$ and mobile crowdsensing task $T$
Output: Winner set $W$ and payment to each winner
 1: $U'_T = \{u \in U_T | u.b < P\}$
 2: Sort the users in $U'_T$ in ascending order of their priorities.
 3: $W = \emptyset$
 4: for $i$=1 to $|U'_T|$ do
 5:    for $j$=1 to $|W|$ do
 6:       if $dis(U'_T[i], W[j]) < T.d$ then
 7:          break
 8:       end if
 9:    end for
10:    if $j > |W|$ then
11:       if $\{u.b | u \in U'_T \wedge u.b > U'_T[i].b \wedge dis(u, U'_T[i]) < T.d\} \neq \emptyset$ then
12:          $U'_T[i].p = \min\{u.b | u \in U'_T \wedge u.b > U'_T[i].b \wedge dis(u, U'_T[i]) < T.d\}$
13:       else
14:          $U'_T[i].p = P$
15:       end if
16:       $W = W \cup \{U'_T[i]\}$
17:       $U'_T = U'_T - \{u \in U'_T | u.b > U'_T[i].b \wedge dis(u, U'_T[i]) < T.d\}$
18:    end if
19: end for
20: return $W$
```

The detailed process of the above algorithm is shown in Algorithm 11.

For the non-optimal winner selection algorithm, we can see that in the 10th step, it checks if $j > |W|$. In other words, it checks if there exists a user in W which conflicts with $U'_T[i]$. Therefore, the final winner set does not contain conflict users.

For the example shown in Fig.5.2, the execution process of the non-optimal winner selection algorithm is as follows. First, we have $U_T = \{u_2, u_3, u_4, u_5, u_6\}$ and $U'_T = \{u_5, u_3, u_4\}$. In the first iteration, u_5 is chosen as the winner. In the second iteration, u_3 is chosen as the winner. Finally, we have $W = \{u_3, u_5\}$ and the payment for both u_3 and u_5 is 3.

5.4.2 Beneficial Properties of the Algorithm

According to Algorithm 11, we can observe that the winner set W calculated using this algorithm is conflict free. In addition, we also have the following theorems which indicate the non-optimal winner selection algorithm is computationally efficient, truthful, individual rational and profitable.

Theorem 5.4.3. *The time complexity of Algorithm 11 is* $O(|U_T|^2)$

Proof: The time complexity of calculating U'_T and sorting the users in U'_T are $O(|U_T|)$ and $O(|U'_T| \log |U'_T|)$, respectively. The time complexity of calculating winner set W and payment is $O(|U'_T||W|)$. Finally, the total time complexity is $O(|U_T|^2)$ since $W \subseteq U'_T \subseteq U_T$. □

Theorem 5.4.4. *The non-optimal winner selection mechanism is truthful.*

Proof: Let W_i be the winner set W in the ith iteration of Algorithm 11. If user u is chosen as a winner in the ith iteration, then there is no user in W_i conflicting with u. Suppose u proposes another bid $u.b' \leq u.b$ and u is checked in the jth iteration, then we have $j \leq i$ since all the users in U'_T are sorted in the ascending order of their bids. Then there is no user in W_j conflicting with u since $W_j \subseteq W_i$. Therefore, u will also be chosen as a winner. Then we can conclude the selection rule in Algorithm 11 is monotone.

According to the process of Algorithm 11, we have $w.p = P$ if $\{u.b|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\} = \emptyset$. In this case, if we have $w.b > w.p$, then we have $w \notin U'_T$ which lead to $w \notin W$. Similarly, if $\{u.b|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\} \neq \emptyset$ and $w.b > w.p$, another user bidding at $w.p$ in $\{u|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\}$ will become the winner and w cannot be the winner in the following iterations due to conflict. Then we can conclude that each winner in Algorithm 11 is paid the critical value.

Finally, we conclude that the non-optimal winner selection mechanism is truthful according to Theorem 5.4.2. □

Theorem 5.4.5. *The non-optimal winner selection algorithm is individual rational.*

Proof: According to the process of Algorithm 11, for each winner $w \in W$, we have $w.p = P \geq w.b$ if $\{u.b|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\} = \emptyset$ according to the definition of U'_T. Similarly, if $\{u.b|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\} \neq \emptyset$, then we have $w.p = \min\{u.b|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\} \geq w.b$. What's more, according to Theorem 5.4.4, the value of $w.b$ could be regarded as the true valuation of winner w. Then this theorem is proved. □

Theorem 5.4.6. *The non-optimal winner selection algorithm is profitable.*

Proof: According to the process of Algorithm 11, for each winner $w \in W$, we have $w.p \leq P$ if $\{u.b|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\} = \emptyset$. Similarly, if $\{u.b|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\} \neq \emptyset$, then we have $w.p = \min\{u.b|u \in U'_T \wedge u.b > w.b \wedge dis(u,w) < T.d\} \leq P$ according to the definition of U'_T. Therefore, we have $w.p \leq P$ for each winner $w \in W$, which leads to $P|W| - \sum_{w \in W} w.p$ to be non-negative. Therefore, this theorem is proved. □

5.5 Experiment

5.5.1 Experiment Settings

We investigated two actual datasets from the Stanford Large Network Dataset Collection named Brightkite and Gowalla as user distribution in the physical world [111]. Both

Brightkite and Gowalla are location-based social network service providers where users can share their locations. In the original Brightkite and Gowalla datasets, all the users distribute worldwide. However, in actual crowdsensing applications, the range of sensing region is limited [18]. Therefore, we employ part of the original Brightkite and Gowalla datasets. In the employed two subsets, the users are distributed in $400km \times 400km$ rectangle regions. These regions include New York, Washington and Philadelphia where users are densely distributed. Random way point model is used to estimate users' positions [112]. The detailed position estimation method is omitted due to page limitation. Two groups of measurement results on Brightkite and Gowalla datasets are presented in Fig.5.5 and Fig.5.6. Fig.5.5 shows the number of users in U_T and Fig.5.6 depicts the average of degree of conflict graph $G(U_T, E)$, respectively. In the conflict graph, a node denotes a user. There is an undirected edge between two nodes if two users are conflict users. Therefore, the average degree of the conflict graph is used to denote the average number of conflict users.

We use an actual bidding dataset named Swoopo to generate users' bids [113]. The preprocessed the raw data as following.

1. A subset of the original dataset which only contains the bids of a single good is extracted.

2. For simplicity, all the original bids are normalized by dividing the bid average.

The preprocessed bid dataset contains 5372 bids. The distribution of users' normalized bids is shown in Fig.5.7. The parameters and their default values for the experiments are listed in Table 5.2. The experiment results for the optimal winner selection algorithm and non-optimal winner selection problem are shown in Section 5.5.2 and Section 5.5.3.

5.5.2 Optimal Winner Selection Algorithm

We compared the execution speed of our algorithm with the Debbassac algorithm proposed by [114]. The Debbassac algorithm explores the space of feasible winner selections

Parameter	Brightkite Dataset	Gowalla Dataset
$T.x$	0.627164	0.837312
$T.y$	0.485815	0.681470
$T.r$	0.0045	0.003
$T.d$	0.001	0.001
P	1	1
ϵ	0.1	0.1

Table 5.2. Default Parameters

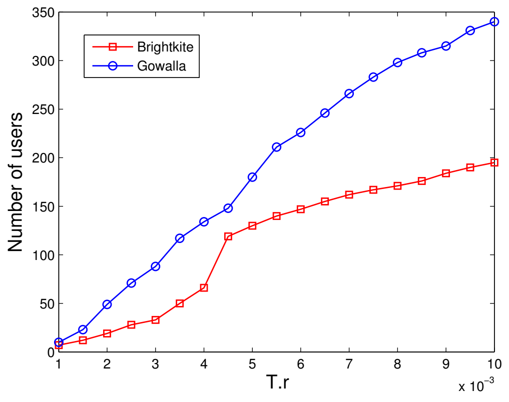


Figure 5.5. Number of users in U_T

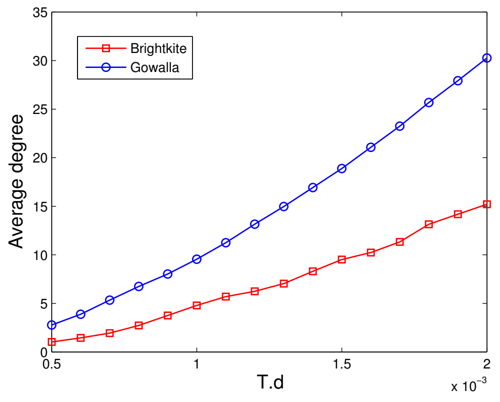


Figure 5.6. Average degree of conflict graph

in a depth-first-search fashion and uses a branch-and-bound approach to prune the search space.

The first group of experiments is to compare the execution time with different radiuses of sensing data gathering regions denoted by $T.r$. The results for both the Brightkite and Gowalla datasets are shown in Fig.5.8. We can see that the execution time for both the A* algorithm and the Debbassac algorithm increases with $T.r$ due to the increase of the total number of crowdsensing participants. Also, the execution time for the A* algorithm is much lower than that of the Debbassac algorithm for the same $T.r$, which demonstrates the efficiency of the algorithm we proposed. Moreover, we can see that for the Brightkite dataset, the execution time for the A* algorithm does not increase when $T.r > 4 \times 10^{-3}$. The reasons are as follows.

1. The number of users almost remains the same when $T.r > 4 \times 10^{-3}$ due to user's

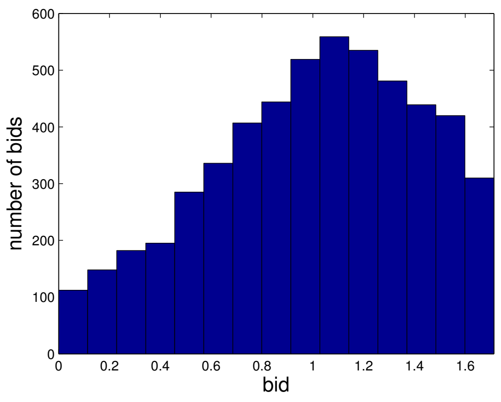
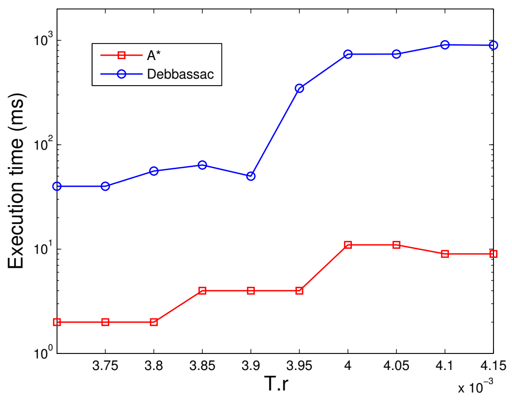
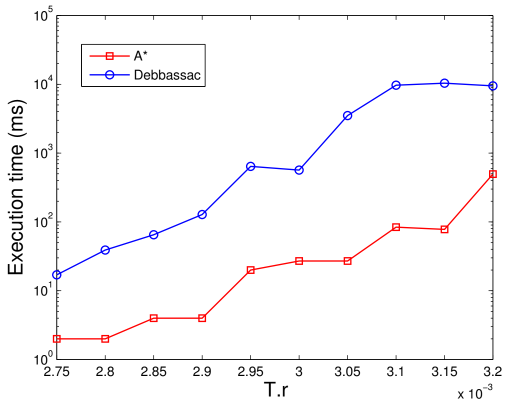


Figure 5.7. Bid distribution



(a) Brightkite



(b) Gowalla

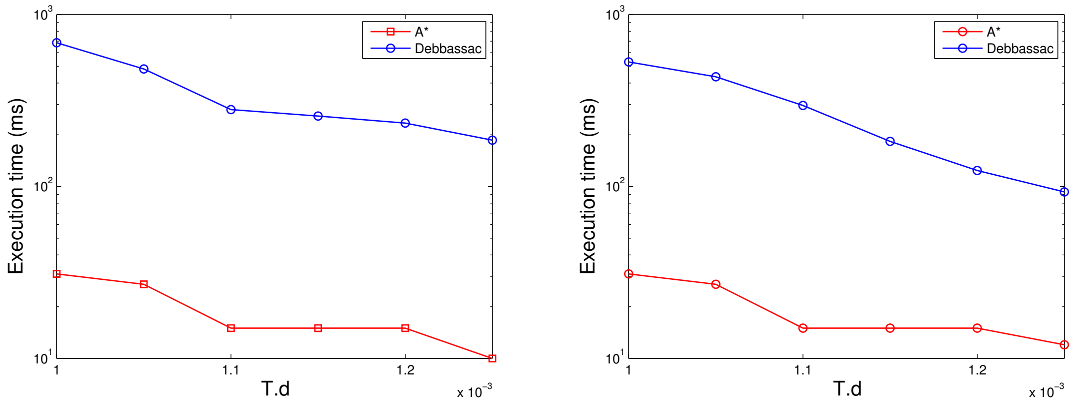
Figure 5.8. Execution time comparison for different $T.r$.

geographical distribution.

2. Newly induced users have high bids, which greatly helps the pruning of the A* algorithm.

The second group of experiments is to compare the execution time with different minimum distances for the winners which is denoted by $T.d$. We set the value of $T.r$ to 0.003 for both the Brightkite and Gowalla datasets. The results are shown in Fig.5.9. We can observe that the A* algorithm executes much faster than the Debbassac algorithm, which indicates that the proposed A* algorithm is efficient.

The Debbassac algorithm proposed in [114] can also solve the optimal winner selec-



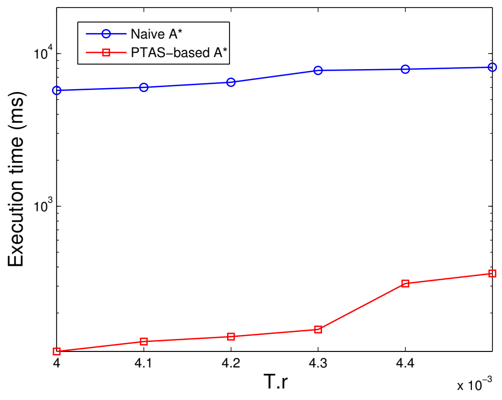
(a) Brightkite (b) Gowalla

Figure 5.9. Execution time comparison for different $T.d$.

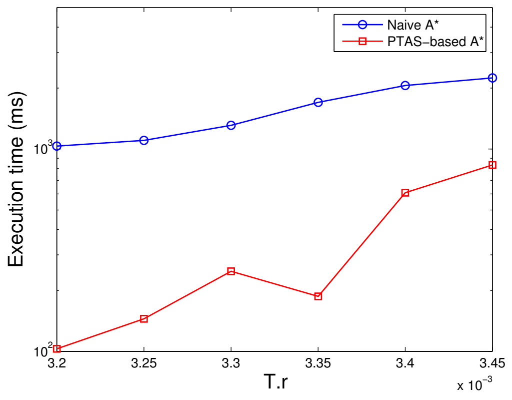
tion problem. However, according to [114], the Debbassac algorithm also has exponential time complexity. Although both these two algorithms have exponential time complexity, the experiment results show that our algorithm is much more efficient than the Debbassac algorithm. The reason is that PTAS [106] can accurately estimate the maximized total weight of a given graph, which greatly helps the pruning of the A* algorithm.

The third group of experiments is to compare the execution time between the PTAS-based A* algorithm and the naive A* algorithm. The only difference between the naive A* algorithm and the A* algorithm is that the naive A* algorithm simply sets the lower bound to 0 and the upper bound to $sum + ub_1$ for calculating the heuristic function. The results are shown in Fig.5.10. The results indicate that the proposed A* algorithm runs faster than the naive A* algorithm on both Brightkite and Gowalla datasets although additional overhead is investigated in the PTAS-based calculation of lower bound and upper bound.

The fourth group of experiment is to compare the execution time of the PTAS-based A* algorithm for different choices of ϵ. The results are shown in Fig.5.11. We can see that the execution time for the A* algorithm increases with the increase of ϵ for both the Brightkite and Gowalla datasets. The results indicate that an accurate estimation of lower bound and upper bound is helpful for the reduction of total execution time, even though using lower ϵ value may cause additional calculation overhead.



(a) Brightkite



(b) Gowalla

Figure 5.10. Execution time comparison between different A* algorithms

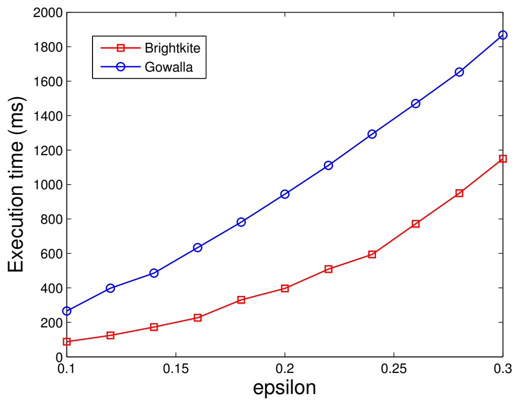


Figure 5.11. Execution time for different ϵ

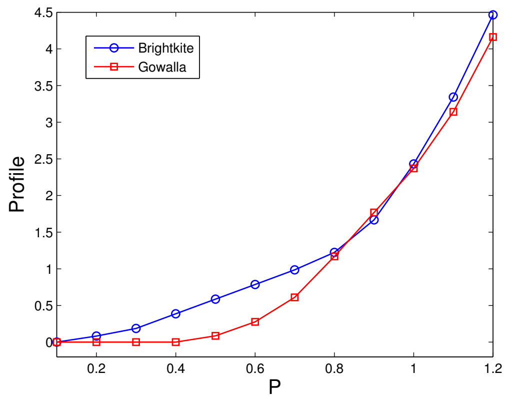


Figure 5.12. Crowdsensing platform's profile

The fifth group of experiments is to show the benefits of inducing geographical position conflict. All the users are assigned with spatial correlated sensory data. The results are shown in Table 5.3. The naive method simply collects the sensory data for all the users whose bids are greater than P. We can see for both Brightkite and Gowalla, the naive method collects a large amount of sensory data while the number of the distinct readings is limited. However, for the optimal winner selection method, the number of the distinct readings remains the same while the amount of sensory data is much less than that of the naive method. The above results indicate that the optimal method can greatly reduce redundant sensory data.

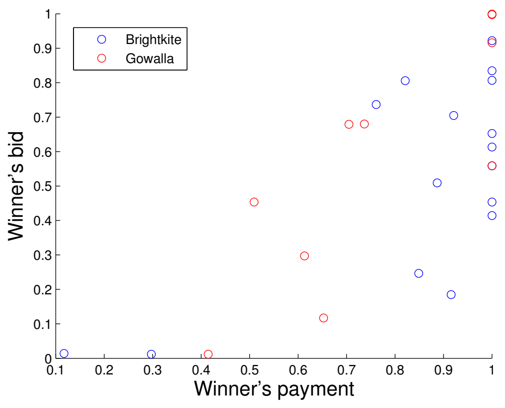


Figure 5.13. Winners' payments and bids

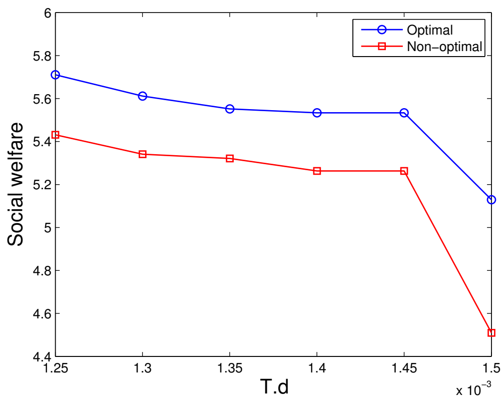
Dataset	Sensory data count	Distinct values
Naive method on Brightkite	23	3
Naive method on Gowalla	24	3
Optimal method on Brightkite	6	3
Optimal method on Gowalla	5	3

Table 5.3. Amount of sensory data comparison

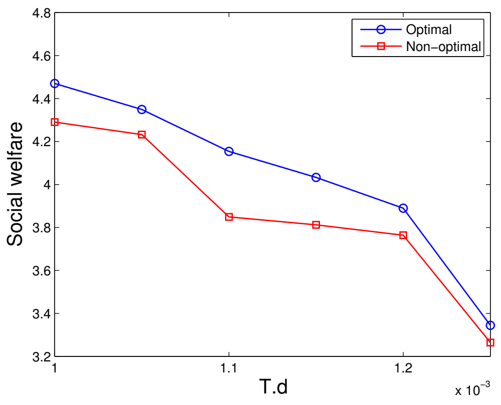
5.5.3 Non-optimal Winner Selection Algorithm

The first group of experiments is to compare the total social welfare between the optimal and non-optimal algorithm. The results are shown in Fig.5.14. The results for the Brightkite dataset and the Gowalla dataset are shown separately for comparison. For both the optimal and non-optimal winner selection algorithm, we can see that the total social welfare decreases with the increase of $T.d$ due to the increase of conflicts. Moreover, for the same $T.d$, the total social welfare of the non-optimal algorithm is smaller than that of the optimal algorithm, which illustrates our tradeoff between the total social welfare and algorithm performance.

The second group of experiments is to compare the profile of crowdsensing platform with different P. The results are shown in Fig.5.12. We can see the total social welfare of the crowdsensing platform increases with the increase of P for both the Brightkite and the Gowalla datasets. Moreover, the results also indicate the crowdsensing platform has non-negative social welfare even if P is relatively low, which verifies the correctness of Theorem 5.4.6.



(a) Brightkite



(b) Gowalla

Figure 5.14. Social welfare comparison

The third group of experiments is to compare winners' payments and bids. The results are shown in Fig.5.5.2 where the dots are used to denote the bids and payments of the winners. The results indicate each winner's payment is greater than its bid for both the Brightkite and the Gowalla datasets, which verifies the correctness of Theorem 5.4.5.

5.6 Conclusion

In this chapter, we propose a geographical position conflicting based winner selection problem in mobile crowdsensing. An optimal winner selection algorithm is proposed to maximize the total social welfare. Moreover, we also propose a computationally efficient non-optimal winner selection algorithm. Solid theoretical proofs indicate both these algorithms are truthful, individual rational and profitable. The experimental results which are based on actual datasets indicate the proposed algorithms are efficient. An interesting aspect of research in the future is to make use of the sensory data's spatial correlation to study users' cooperation in geographical position based crowdsensing tasks.

Chapter 6

APPROXIMATE HOLISTIC AGGREGATION IN SMALL SCALE NETWORKS

6.1 Introduction

With the ever-increasing population, problems of everyday sustainability have become onerous. According to the survey by United Nations, 54% of the world's current population lives in urban areas. It is also expected that the percentage increases to 66 by 2050. With this escalation in urban population, new challenges have emerged. These challenges include the constant power supply, public safety, disaster prediction, and traffic maintenance. Smart City has become inevitable to address these challenges [115] [116] [117] [118] [119] [120]. Many cities like New York, Detroit, Singapore, and London are working towards smart city development. These cities have adopted for various technologies like smart parking services, intelligent street light systems, sensors to redirect traffic, and water conservation. Although these applications are designed for urban living, they should also be incorporated in rural areas so that more resources could be preserved for future generations. SC networks should collect data from all over the city to provide better information. So, to collect such well spread data, they exploit various sensor equipped devices in the city to collect data and interpret information at the city level.

In today's world, all the devices from smart home devices to intelligent transport systems are well connected with the internet. Such a network with well connected devices is called Internet-Of-Things (IoT) [121] [122] [123] [124] [125] [126]. As the network grows, our need for intelligent devices develop and so does the necessity to sense various activities for the convenient living of people in the cities. some of the applications like transportation, healthcare, and seamless internet connection widely use IoT technologies. The main aim in IoT is to reduce cost and provide faster access to the data. But the primary challenge is that the deployment of IoT network is expensive as it requires a large number of sensing devices. Additionally, it is also important to collect data autonomously and provide intelligent methods that address the issues of dynamic traffic, accommodating new services, channel conditions, and ever-increasing user requirements.

Sensors are the building blocks for many IoT devices. Utilizing sensors as the communication media helps us resolve many of the problems discussed earlier. IoT devices with self-configurable sensors do not require external infrastructure for communication [127] [128] [129] [130] [131] [132]. Previous researchers have studied the issues of routing, topology control and time synchronization in networks that include sensors for communication [66], [133], [134]. Although using sensors reduces the communication cost, it raises the issue of processing cost. Sensors collect data for a more extended period over vast networks. Therefore, we end up with massive data being processed for information retrieval at a single sensor node and thus increasing the processing cost. To address this issue, we need data aggregation at the sensor level. When data is aggregated, we do not transmit the whole sensing data but instead, send the aggregated partial data into the network. This further raises the energy consumption issue as the aggregation costs much energy and the sensor are not equipped with huge amounts of power supply. According to [86], cost of transmitting one bit of data using wireless link is equivalent to the cost of executing 1000 instructions. So, reducing the data transmission is one of the major ways to decrease the energy consumption in IoTs [135] [136] [137] [138] [139] [140]. Hence, it is critical to design energy efficient data aggregation models for sensor equipped IoT networks.

There are two kinds of aggregation queries: maximum query and distinct set query. The maximum query is to calculate the maximum of all the readings in the sensory data while the distinct set query is to calculate the unique values in the sensory data. Both the queries are essential for a given network. For example, while monitoring pollution, maximum query results in the most polluted area along with its values. Similarly, distinct set query shows the pollution levels in all the regions. Hence, the energy efficient data aggregation model should accommodate both queries in its development.

In practice, exact query results are not always necessary while approximate query results may also be acceptable for the purpose of energy conservation [141] [142]. Therefore, in this chapter, two algorithms to process δ-approximate maximum queries and δ-distinct-set queries are proposed. These two algorithms are based on uniform sampling and Bernoulli sampling, respectively. Unlike the algorithms proposed in Chapter 3 and Chapter 4, which are for large scale networks, the algorithms proposed in this chapter are for small scale networks. The proposed algorithms are able to return the exact query results with probability not less than 1-δ where the value of δ can be arbitrarily small. The main contributions of this chapter can be summarized are as follows.

1. Mathematical estimators for the maximum value and distinct-set aggregation operations are provided.

2. The mathematical methods to determine the required sample size and sample probability for calculating δ-approximate maximum value and δ-approximate distinct-set are designed.

3. Distributed algorithms for δ-approximate maximum value and δ-approximate distinct-set are provided. Additionally, the energy costs of these two algorithms are analyzed.

4. Extensive simulation results are presented which indicates that both δ-approximate maximum value and δ-approximate distinct set algorithms perform significantly better than a simple distributed algorithm in terms of energy consumption.

The rest of the chapter is organized as follows. Section 6.2 defines the problem. Section 6.3 provides the mathematical proof for the δ-approximate aggregation algorithms. Section 6.4 explains the proposed δ-approximate aggregation algorithms. Section 6.5 shows the simulation results. Section 6.6 concludes this chapter.

6.2 Problem Definition

Let us assume that we have a sensor equipped IoT network with n sensor nodes and s_{ti} is the sensory value of node i at time t. $S_t = \{s_{t1}, s_{t2}, \ldots, s_{tn}\}$ is used to denote the set of all the sensory data in the network at time t. We use $Dis(S_t) = \{s^d_{t1}, s^d_{t2}, \ldots, s^d_{t|Dis(S_t)|}\}$ to denote the distinct set of S_t, which contains the distinct values in S_t. For example, if we have $S_t = \{s_{t1}, s_{t2}, s_{t3}, s_{t4}, s_{t5}\}$ and $s_{t1} = 1, s_{t2} = 1, s_{t3} = 2, s_{t4} = 3, s_{t5} = 3$; then the $Dis(S_t) = \{1, 2, 3\}$. In this chapter, we assume that the data is distributed randomly in the network while the spatial and temporal correlation for the sensory data is ignored.

In this chapter, we focus on two aggregation operations on S_t, which are *max* and *distinct set*. The definition of the maximum value and distinct-set are as follows:

1. The exact maximum value denoted by $Max(S_t)$ satisfies $Max(S_t) = \max\{s_{ti} \in S_t | 1 \leq i \leq n\}$.

2. The exact distinct-set of S_t denoted by $Dis(S_t)$ satisfies that $\forall s \in S_t, \exists s^d \in Dis(S_t), s = s^d$ and $\forall s^d_x,\ s^d_y \in Dis(S_t), x \neq y \Rightarrow s^d_x \neq s^d_y$.

A naive method that solves the max and distinct set aggregation problems has three main steps.

1. Organize all the nodes in the network into an aggregation tree. The sink node broadcasts the aggregation operation in the network.

2. All the nodes in the network submits its sensory data to the sink node along the aggregation tree.

3. The intermediate nodes in the aggregation tree aggregates the partial results during the data transmission.

However, the above method will lead to an immense communication cost and computation cost for calculating exact aggregation result. Therefore, we propose a δ-approximate result for the above two aggregation operations. Let I_t and $\widehat{I_t}$ are the exact aggregation result and

Name	Description
n	The size of network
s_{ti}	The sensory value of node i at time t
s_{ti}^d	The i-th distinct sensory value at time t
$S_t = \{s_{t1}, s_{t2}, \ldots, s_{tn}\}$	The sensory data in the network at time t
$Max(S_t)$	The maximum value of S_t
$Dis(S_t)$	The distinct set of S_t
$U(m)$	A uniform sample of S_t with size m
$B(q)$	A Bernoulli sample of S_t with sampling probability q
n_x	Number of appearance of value x in S_t
n_{min}	Number of appearances for least appearing data
k	Number of clusters
C_i	The i-th cluster

Table 6.1. Symbols

approximate aggregation result of S_t at time t respectively. The definition of the δ-estimator is as follows.

Definition 6.2.1 (δ-estimator)**.** *For any δ $(0 \leq \delta \leq 1)$, $\widehat{I_t}$ is called the δ-estimator of I_t if* $\Pr(\widehat{I_t} \neq I_t) \leq \delta$,

According to Definition 6.2.1, the problem of computing δ-approximate maximum value and δ-approximate distinct-set is defined as follows.

Input: (1) A sensor equipped IoT network with n nodes; (2) The sensory data set S_t; (3) Aggregation operator $Agg \in \{Max,\ DistinctSet\}$ and δ $(0 \leq \delta \leq 1)$.

Output: δ-approximate aggregation result of *Agg*.

Summary of all the frequently used symbols is shown in Table 6.1.

6.3 Preliminaries

6.3.1 Uniform Sampling Based Approximate Aggregation

Let $u_1, u_2, ..., u_m$ denote m simple random samplings with replacement from S_t, $U(m) = \{u_1, u_2, ..., u_m\}$ is used to denote a uniform sample of S_t with sample size m, then we have the following conclusions.

1. u_i and u_j are independent with each other for all $1 \leq i \neq j \leq m$.

2. $\Pr(u_i = s_{tj}) = \frac{1}{n}$ for any $1 \leq i \leq m$, $1 \leq j \leq n$.

Based on the above conclusions, we have the following theorem.

Lemma 6.3.1. *For any given value $x \in Dis(S_t)$, we have*

$$\Pr(x \notin U(m)) = (1 - \frac{n_x}{n})^m$$

where n_x is the number of appearance of value x in S_t.

Proof: $\Pr(x \notin U(m)) = \Pr(u_1 \neq x \wedge u_2 \neq x \wedge \ldots u_m \neq x)$. Since all the samples $u_1, u_2, \ldots u_m$ are independent with each other, we have

$$\Pr(x \notin U(m)) = \prod_{i=1}^{m} \Pr(u_i \neq x) = (\Pr(u_1 \neq x))^m.$$

Moreover, we also have

$$\Pr(u_1 \neq x) = 1 - \Pr(u_1 = x) = 1 - \frac{n_x}{n}.$$

Then this lemma is proved. □

To obtain δ-approximate maximum value, the mathematical estimator is needed firstly. Let $\widehat{Max(S_t)}_u$ denote the uniform sampling based estimator of exact value $Max(S_t)$. Then $\widehat{Max(S_t)}_u$ is defined as

$$\widehat{Max(S_t)}_u = Max(U(m)) = \max\{u_i \in U(m) | 1 \leq i \leq m\}.$$

Based on Lemma 6.3.1, we have the following theorem.

Theorem 6.3.1. *$\widehat{Max(S_t)}_u$ is a δ-estimator of $Max(S_t)$ if*

$$m \geq \frac{\ln \delta}{\ln(1 - \frac{n_{min}}{n})}$$

where n_{min} is the number of appearances for the least appearing data.

Proof: Based on the condition, we have

$$m \ln(1 - \frac{n_{min}}{n}) \leq \ln \delta$$

$$(1 - \frac{n_{min}}{n})^m \leq \delta.$$

According to Lemma 6.3.1, we have

$$\Pr(Max(S_t) \notin U(m)) = (1 - \frac{n_{Max(S_t)}}{n})^m$$

where $n_{Max(S_t)}$ is the number of appearance for the maximum value in S_t. Since $n_{Max(S_t)} \geq n_{min}$, we have

$$\Pr(Max(S_t) \notin U(m)) \leq (1 - \frac{n_{min}}{n})^m \leq \delta.$$

Then this theorem is proved. $\square$

Let $\widehat{Dis(S_t)}_u$ denote the uniform sampling based estimator of exact result $Dis(S_t)$. Then $\widehat{Dis(S_t)}_u$ is defined as

$$\widehat{Dis(S_t)}_u = Dis(U(m)).$$

Based on Lemma 6.3.1, we have the following theorem.

Theorem 6.3.2. *$\widehat{Dis(S_t)}_u$ is a δ-estimator of $Dis(S_t)$ if*

$$m \geq \frac{\ln(1 - (1 - \delta)^{n_{min}/n})}{\ln(1 - \frac{n_{min}}{n})}$$

where n_{min} is the number of appearances for the least appearing data.

Proof: Based on the condition, we have

$$(1-\frac{n_{min}}{n})^m \leq 1-(1-\delta)^{n_{min}/n}$$

$$(1-(1-\frac{n_{min}}{n})^m)^{n/n_{min}} \geq 1-\delta$$

$$1-\prod_{i=1}^{|Dis(S_t)|}(1-(1-\frac{n_{min}}{n})^m) \leq \delta$$

Let $n_{s^d_{ti}}$ to denote the number of appearance for $s^d_{t_i}$, then we have

$$1-\prod_{i=1}^{|Dis(S_t)|}(1-(1-\frac{n_{s^d_{ti}}}{n})^m) \leq \delta$$

since $n_{min} \leq n_{s^d_{ti}}$. Moreover, according to Lemma 6.3.1, we have

$$1-\prod_{i=1}^{|Dis(S_t)|}(1-\Pr(s^d_{ti} \notin U(m))) \leq \delta$$

$$1-\prod_{i=1}^{|Dis(S_t)|}\Pr(s^d_{ti} \in U(m)) \leq \delta$$

$$1-\Pr(\widehat{Dis(S_t)}_u = Dis(S_t)) \leq \delta$$

$$\Pr(\widehat{Dis(S_t)}_u \neq Dis(S_t)) \leq \delta$$

Then this theorem is proved. □

6.3.2 Bernoulli Sampling Based Approximate Aggregation

Let $B(q) = \{b_1, b_2, \ldots, b_{|B(q)|}\}$ denote a Bernoulli sample of data set S_t with sample probability q. Then we have the following lemma.

Lemma 6.3.2. *For any given value $x \in Dis(S_t)$, we have*

$$\Pr(x \notin B(q)) = (1-q)^{n_x}$$

where n_x is the number of appearance of value x in S_t.

Proof: Without loss of generality, we assume $s_{t1} = s_{t2} = \cdots = s_{tn_x} = x$, then we have $\Pr(x \notin B(q)) = \Pr(s_{t1} \notin B(q) \wedge s_{t2} \notin B(q) \wedge \cdots \wedge s_{tn_x} \notin B(q))$. Therefore, we have

$$\Pr(x \notin B(q)) = \prod_{i=1}^{n_x} \Pr(s_{ti} \notin B(q)) = (\Pr(s_{t1} \notin B(q)))^{n_x}.$$

According to the definition of Bernoulli sampling, we have

$$\Pr(s_{t1} \notin B(q)) = 1 - \Pr(s_{t1} \in B(q)) = 1 - q.$$

Then this lemma is proved. □

Let $\widehat{Max(S_t)}_b$ denote the Bernoulli sampling based estimator of exact value $Max(S_t)$. $\widehat{Max(S_t)}_b$ is defined as

$$\widehat{Max(S_t)}_b = Max(B(q)) = \max\{b_i \in B(q) | 1 \leq i \leq |B(q)|\}.$$

Based on Lemma 6.3.2, we have the following theorem.

Theorem 6.3.3. *$\widehat{Max(S_t)}_b$ is a δ-estimator of $Max(S_t)$ if*

$$q \geq 1 - (\delta)^{1/n_{min}}$$

where n_{min} is the number of appearances for the least appearing data.

Proof: Based on the condition, we have

$$(1 - q)^{n_{min}} \leq \delta.$$

According to Lemma 6.3.2, we have

$$\Pr(Max(S_t) \notin B(q)) = (1 - q)^{n_{Max(S_t)}}$$

where $n_{Max(S_t)}$ is the number of appearance for the maximum value in S_t. Since $n_{Max(S_t)} \geq n_{min}$, we have

$$\Pr(Max(S_t) \notin B(q)) \leq (1-q)^{n_{min}} \leq \delta.$$

Then this theorem is proved. □

Let $\widehat{Dis(S_t)}_b$ denote the Bernoulli sampling based estimator of exact result $Dis(S_t)$. Then $\widehat{Dis(S_t)}_b$ is defined as

$$\widehat{Dis(S_t)}_b = Dis(B(q)).$$

Based on Lemma 6.3.1, we have the following theorem.

Theorem 6.3.4. *$\widehat{Dis(S_t)}_b$ is a δ-estimator of $Dis(S_t)$ if*

$$q \geq 1 - (1-(1-\delta)^{n_{min}/n})^{1/n_{min}}$$

where n_{min} is the number of appearances for the least appearing data.

Proof: According to the condition, we have

$$(1-q)^{n_{min}} \leq 1-(1-\delta)^{n_{min}/n}$$

$$(1-(1-q)^{n_{min}})^{n/n_{min}} \geq 1-\delta$$

$$1 - \prod_{i=1}^{|Dis(S_t)|} (1-(1-q)^{n_{min}}) \leq \delta.$$

Let $n_{s_{ti}^d}$ to denote the number of appearance for $s_{t_i}^d$, since $n_{min} \leq n_{s_{ti}^d}$, we have

$$1 - \prod_{i=1}^{|Dis(S_t)|} (1-(1-q)^{n_{s_{ti}^d}}) \leq \delta.$$

Moreover, according to Lemma 6.3.2, we have

$$1 - \prod_{i=1}^{|Dis(S_t)|} (1-\Pr(s_{ti}^d \notin B(q))) \leq \delta$$

$$1 - \prod_{i=1}^{|Dis(S_t)|} \Pr(s_{ti}^d \in B(q)) \leq \delta$$

$$1 - \Pr(\widehat{Dis(S_t)}_b = Dis(S_t)) \leq \delta$$

$$\Pr(\widehat{Dis(S_t)}_b \neq Dis(S_t)) \leq \delta.$$

Then this theorem is proved. □

6.4 δ-Approximate Aggregation Algorithms

The theorems in Section 6.3 show how to calculate the required sampling size and sampling probability according to given δ. However, we still have the following problems to be solved.

1. How does the sink node broadcasts the sampling information in the whole network.
2. How to sample the sensory data from the whole network.
3. How to transmit and aggregate the partial aggregation results.

The methodology to solve the above problems will be introduced in the following two subsections.

6.4.1 Uniform Sampling Based Aggregation Algorithm

When the sample size m is calculated using the theorems in Section 6.3.1, there is a simple method to sample the sensory data.

1. The sink generates m random numbers in $\{1, 2, 3, . . . , n\}$ and broadcasts them in the whole network.
2. The sensor node whose id belongs to the m numbers sends its sensory data to the sink node.

However, the above algorithm has a huge energy cost during the first step since a significant amount of sampling information needs to be transmitted. To further reduce the energy cost, we divide the whole network into k disjoint clusters C_1, C_2, ... , C_k. Each cluster randomly selects one of its node as the cluster head. By using the method, proposed in [60], all the cluster heads in the network are organized as a minimum hop-count spanning tree rooted at the sink node. We then adopt the uniform sampling algorithm proposed by [54], described as follows.

1. The sink generates a series of random numbers Y_i with the probability $\Pr(Y_i = l) = \frac{|C_l|}{n}(1 \leq i \leq m)$,
2. Let m_l be the sample size of C_l. Then m_l is calculated by $m_l = |\{Y_i | Y_i = l\}|$.
3. The sink node sends the sample size $\{m_l \mid 1 \leq l \leq k\}$ to each cluster head. Each cluster head samples the sensory data in the cluster using the above naive sampling algorithm.

When the cluster head of the l-th cluster receives all the sampled sensory data, $U(ml)$, it calculates the partial aggregation result $R(U(m_l))$ according to aggregation operation Agg by using the following method.

$$R(U(m_l)) = \begin{cases} Max(U(m_l)) & \text{if } Agg = Max \\ Dis(U(m_l)) & \text{elsewhere} \end{cases}$$

The partial aggregation result $R(U(m_l))$ is transmitted along the spanning tree to the sink node. To further reduce the transmission cost, the intermediate nodes in the spanning tree aggregate the received partial result while transmitting the data. Above process is explained in Algorithm 12.

Algorithm 12: Uniform Sampling Based Aggregation Algorithm

```
Input: $\delta$, aggregation operator $Agg \in \{Max,\ DistinctSet\}$
Output: $\delta$-approximate aggregation results
1: if $Agg = Max$ then
2:     $m = \lceil \frac{\ln \delta}{\ln(1-\frac{n_{min}}{n})} \rceil$
3: else
4:     $m = \lceil \frac{\ln(1-(1-\delta)^{n_{min}/n})}{\ln(1-\frac{n_{min}}{n})} \rceil$
5: end if
6: generate $Y_i$ following $\Pr(Y_i = l) = \frac{|C_l|}{n}$,
7: $m_l = |\{Y_i \mid Y_i = l\}|$ $(1 \le i \le m, 1 \le l \le k)$, the sink sends $m_l$ to each cluster head by
   multi-hop communication
8: for each cluster head of the clusters $C_l$ $(1 \le l \le k)$ do
9:     generates random numbers $k_1$, $k_2$, $\ldots$, $k_{m_l}$ then broadcast inside the cluster
10: end for
11: for each cluster member of $C_l$ $(1 \le l \le k)$ do
12:     send sensory value to cluster head if its $id \in \{k_1, k_2, \ldots, k_{ml}\}$;
13: end for
14: for each cluster head of the clusters $C_l$ $(1 \le l \le k)$ do
15:     receive sample data $U(m_l)$ and calculate partial result $R(U(m_l))$;
16: end for
17: for each node $j$ in the spanning tree do
18:     if $j$ is the leaf node then
19:         Send $R_j$ to its parent node
20:     else
21:         Receive partial results $R_{j1}, R_{j2}, \ldots, R_{jc}$ from its children
22:         if $Agg = Max$ then
23:             $R_j = \max(R_{j1}, R_{j2}, \ldots, R_{jc})$
24:         else
25:             $R_j = \bigcup_{i=1}^{c} R_{ji}$
26:         end if
27:         if $j$ is the sink node then
28:             return $R_j$
29:         else
30:             Send $R_j$ to its parent node
31:         end if
32:     end if
33: end for
```

According to the analysis in Section 6.3.1, for the sample size m, we have

$$m = \begin{cases} \lceil \frac{\ln\delta}{\ln(1-\frac{n_{min}}{n})} \rceil & \text{if } Agg = Max \\ \lceil \frac{\ln(1-(1-\delta)^{n_{min}/n})}{\ln(1-\frac{n_{min}}{n})} \rceil & \text{if } Agg = Dis \end{cases}$$

Therefore, we have

$$m = \begin{cases} O(\ln\frac{1}{\delta}) & \text{if } Agg = Max \\ O(\ln(\frac{1}{1-(1-\delta)^{n_{min}/n}})) & \text{if } Agg = Dis \end{cases}$$

In practice, $|R_j|$ can be regarded as a constant. According to [54], the communication cost and the energy cost of the uniform sampling based δ-approximate aggregation algorithm is $O(\ln\frac{1}{\delta})$ if $Agg = Max$, while the cost is $O(\ln(\frac{1}{1-(1-\delta)^{n_{min}/n}}))$ if $Agg = Dis$.

6.4.2 Bernoulli Sampling Based Aggregation Algorithm

Unlike the uniform sampling based aggregation algorithm, the sampling information of Bernoulli sampling based aggregation algorithm utilizes only the sampling probability q. Additionally, Bernoulli based method provides a mechanism for each node in the network to do the sampling independently. Therefore, the following steps are used in the Bernoulli sampling based aggregation algorithm to perform sampling and the network need not be divided into clusters.

1. Sink node broadcasts the sampling probability q in the network.

2. Each node generates a random number $rand$ in the range of [0,1], submit its sensory data to the parent node if $rand < q$.

When the intermediate nodes in the spanning tree receive the submitted sensory data, they will calculate the partial aggregation results using the similar method introduced in Section 6.4.1. These nodes then transmit the partial results along the spanning tree. Similarly, during the process of transmitting partial aggregation results to the sink node along the spanning tree, the intermediate nodes in the spanning tree aggregate the received partial results. The process mentioned above is explained in detail in the Algorithm 13.

According to the analysis in Section 6.3.2, for the sample probability q, we have

$$q = \begin{cases} 1 - (\delta)^{1/n_{min}} & \text{if } Agg = Max \\ 1 - (1 - (1-\delta)^{n_{min}/n})^{1/n_{min}} & \text{if } Agg = Dis \end{cases}$$

Similarly, we have the communication cost and the energy cost of the Bernoulli sampling based δ-approximate aggregation algorithm is $O(n - n(\delta)^{1/n_{min}})$ if $Agg = Max$, while the cost is $O(n - n(1 - (1-\delta)^{n_{min}/n})^{1/n_{min}})$ if $Agg = Dis$.

6.5 Simulation Results

To evaluate the proposed algorithms, we have simulated a network with 1000 nodes. All the nodes are randomly distributed in a rectangular region of size $300m \times 300m$ and the sink is in the center of the region. For the uniform sampling based aggregation algorithm, the following strategy is used to define the clusters.

1. Divide the whole region into 10×10 grids.

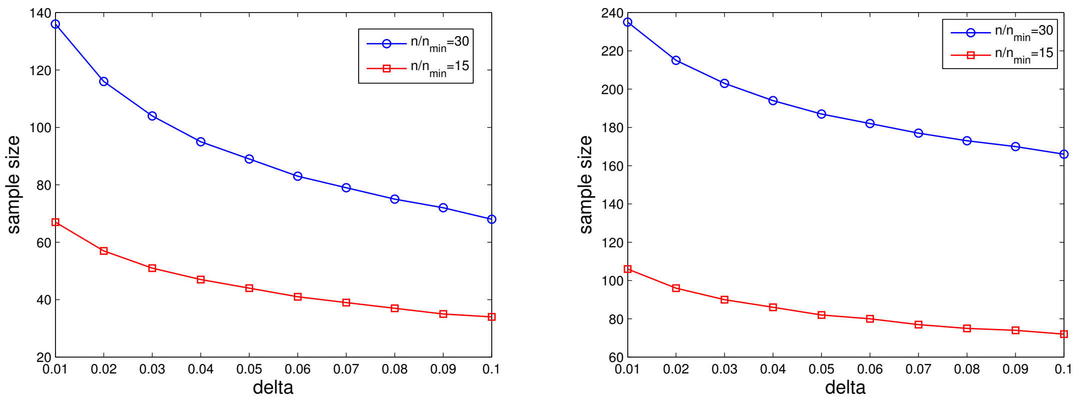
2. Group the nodes in the same gird into the same cluster.

3. Randomly chose the cluster head among the nodes of the same grid.

For each node, the energy cost to send and receive one byte is set as 0.0144mJ and 0.0057mJ according to [4]. According to the experiment results in [61] about the same type of sensor node, the communication range of each sensor node is set to be $30\sqrt{2}$m in our simulation. This kind of simulation setting can make every sensor node communicate with its cluster head by a one-hop message.

Algorithm 13: Bernoulli Sampling Based Aggregation Algorithm

```
Input: $\delta$, aggregation operator $Agg \in \{Max, DistinctSet\}$
Output: $\delta$-approximate aggregation results
 1: if $Agg = Max$ then
 2:     $q = 1 - (\delta)^{1/n_{min}}$
 3: else
 4:     $q = 1 - (1 - (1 - \delta)^{n_{min}/n})^{1/n_{min}}$
 5: end if
 6: Sink node broadcasts $q$ in the network
 7: for each leaf node $j$ in the spanning tree do
 8:     if $rand < q$ then
 9:         Send its own sensory data to its parent node;
10:     end if
11: end for
12: for each non-leaf node $j$ in the spanning tree do
13:     Receive partial results $R_{j1}, R_{j2}, \ldots, R_{jc}$ from its children
14:     if $Agg = Max$ then
15:         $R_j = \max(R_{j1}, R_{j2}, \ldots, R_{jc})$
16:     else
17:         $R_j = \bigcup_{i=1}^{c} R_{ji}$
18:     end if
19:     if $rand < q$ then
20:         if $Agg = Max$ then
21:             $R_j = max(R_j, j.data)$
22:         else
23:             $R_j = R_j \cup \{j.data\}$
24:         end if
25:     end if
26:     if $j$ is the sink node then
27:         return $R_j$
28:     else
29:         Send $R_j$ to its parent node
30:     end if
31: end for
```



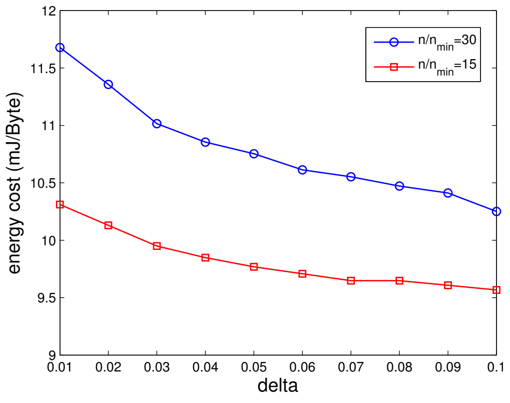
(a) Maximum Value (b) Distinct-Set

Figure 6.1. The relationship between δ and the sample size.

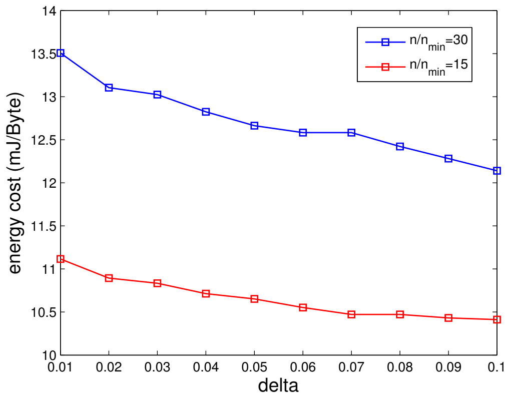
6.5.1 Uniform Sampling Based Aggregation Algorithm

The first group of simulations is about the relationship between δ and the sample size. The results are presented in Fig.6.1. The results for both the maximum value aggregation and the distinct-set aggregation are listed. Additionally, two groups of results with different values of $\frac{n}{n_{min}}$ are listed for comparison. These results show that the sample size increases with the decline of δ. Moreover, the sample sizes are much smaller than the size of the network. For example, when $\delta = 0.01$, the sample size is about 67 for deriving δ-approximate maximum value. If $\frac{n}{n_{min}} = 15$, which indicates that we just need to sample 6.7% sensory data from the network to guarantee that the estimated maximum value being equal to the actual maximum value with the probability greater than 99%. Therefore, our uniform sampling based algorithm saves a tremendous amount of energy as it only needs a little amount of sensory data to be sampled and transmitted in the network. Moreover, we can see that in the same condition, the required sample size for the distinct-set aggregation is greater than that of the maximum value aggregation since distinct-set aggregation needs to make sure all the distinct values being sampled.

The second group of simulations is about the relationship between δ and the energy cost. The results are shown in Fig.6.2. These results indicate that the energy cost increases with



(a) Maximum Value

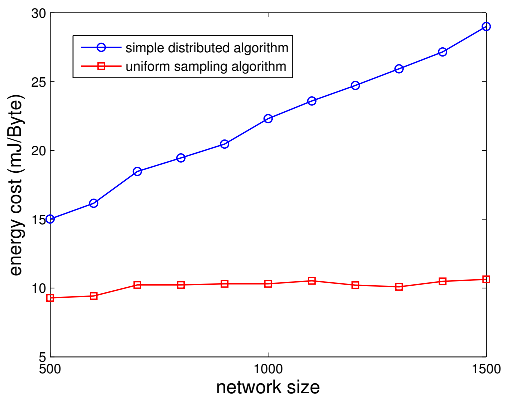


(b) Distinct-Set

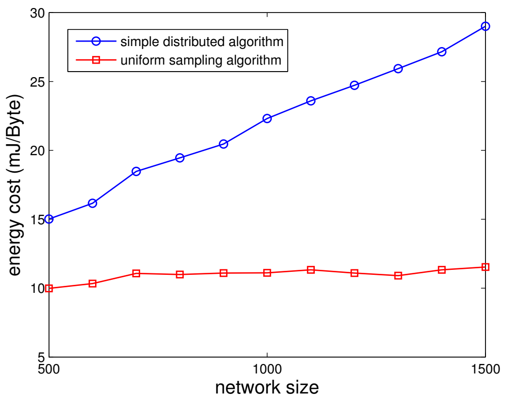
Figure 6.2. The relationship between δ and the energy cost for the uniform sampling based aggregation algorithm.

the decline of δ. We can also see that in the same condition, the energy cost for the distinct-set aggregation is higher than that of the maximum value aggregation as the distinct-set aggregation has a greater sample size.

The third group of simulation is to compare the energy cost between the uniform sampling based aggregation algorithm and the simple distributed algorithm. The simple distributed algorithm is to collect all the raw sensory data and aggregate the partial results during the transmission, which can always return accurate aggregation results. For the uniform sampling based aggregation algorithm, we set $\delta = 0.1$ and $\frac{n}{n_{min}} = 15$ and the network size varies from 500 to 1500. The results are listed in Fig.6.3. We can see that for all the proposed algorithms, the energy cost increases with the increase of the network size. Moreover, for the same network size, the energy cost of the uniform sampling based aggregation algorithm is much lower than that of the naive distributed algorithm as there are only a small number of nodes need to submit their sensory data. These results indicate that the uniform sampling based aggregation algorithm performs better in terms of energy consumption although it returns wrong aggregation results with the probability less than δ. It is also to be observed that with an increase in the network size, the energy cost of the naive distributed algorithm proliferates, while the energy cost of the uniform sampling



(a) Maximum Value



(b) Distinct-Set

Figure 6.3. Energy cost comparison between the uniform sampling based aggregation algorithm and the simple distributed algorithm.

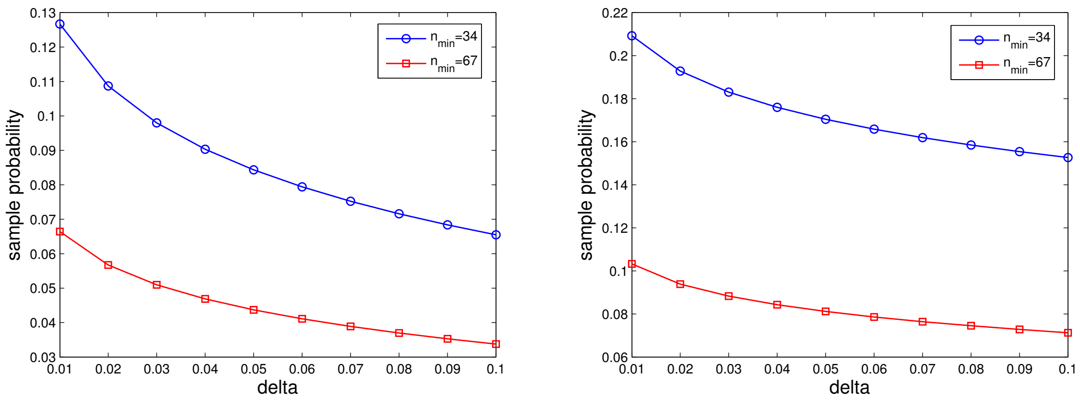
based aggregation algorithm almost remains the same. The above phenomenon indicates that the uniform sampling based aggregation algorithm has even better performance when the network size is large.

6.5.2 Bernoulli Sampling Based Aggregation Algorithm

The first group of simulations is about the relationship between δ and the sample probability. These results are presented in Fig.6.4. The results show that the sample probability increases with the decline of δ. Moreover, the sample probabilities are much smaller than 1. For example, when $\delta = 0.01$, the sample probability is about 0.066 for deriving δ-approximate maximum value. Therefore, the proposed Bernoulli sampling based algorithm saves a huge amount of energy. Similarly, the required sample size for the distinct-set aggregation is greater than that of the maximum value aggregation in the same condition.

The second group of simulations is about the relationship between δ and the energy cost. The results are shown in Fig.6.5. We can observe from these results that the energy cost increases with the decline of δ and the energy cost for the distinct-set aggregation is higher than that of the maximum value aggregation.

The third group of simulation is to compare the energy cost between the Bernoulli sam-

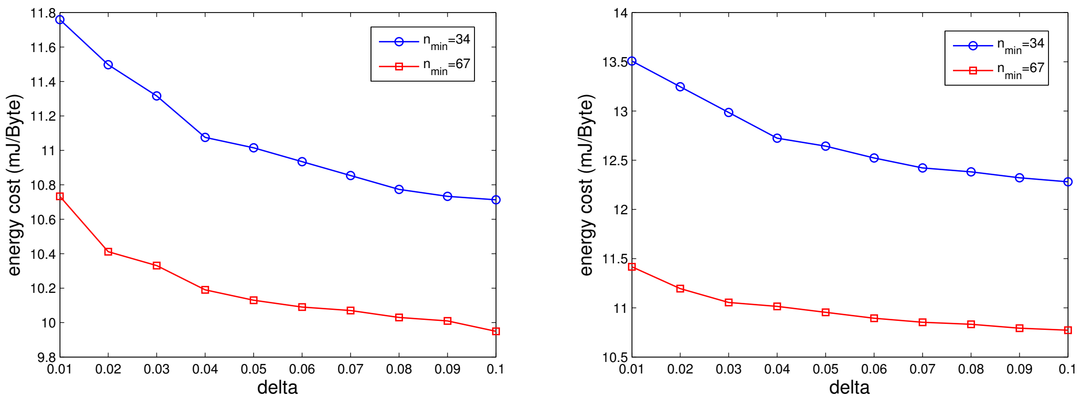


(a) Maximum Value (b) Distinct-Set

Figure 6.4. The relationship between δ and the sample probability.

pling based aggregation algorithm and the simple distributed algorithm. For the Bernoulli sampling based aggregation algorithm, we set $\delta = 0.1$ and $n_{min} = 67$. The network size varies from 500 to 1500 nodes. The results are listed in Fig.6.6. Similar results can be observed for the same network size, the energy cost of the Bernoulli sampling based aggregation algorithm is much lower than that of the simple distributed algorithm. These results indicate that the Bernoulli sampling based aggregation algorithm has high performance on energy consumption. Moreover, we can also see that the Bernoulli sampling based aggregation algorithm has even better performance on large-scale networks.

The fourth group of simulation is to compare the energy cost between the Bernoulli sampling based aggregation algorithm and the uniform sampling based aggregation algorithm. We set $\delta = 0.1$ and $\frac{n}{n_{min}} = 15$. To ensure the network connectivity when the network size is small, we set node's communication to 60m for this group of simulation. The results are shown in Fig.6.7. From these results, we observe that for both the uniform sampling based aggregation algorithm and the Bernoulli sampling based aggregation algorithm, the energy cost increases with the increase in network size. Additionally, the Bernoulli sampling based aggregation algorithm has lower energy cost when the network size is small, while the uniform sampling based aggregation algorithm has lower energy cost when the network size is large. From the above results, we can see that the Bernoulli sampling based aggregation



(a) Maximum Value (b) Distinct-Set

Figure 6.5. The relationship between δ and the energy cost for the Bernoulli sampling based aggregation algorithm.

algorithm has the following advantages.

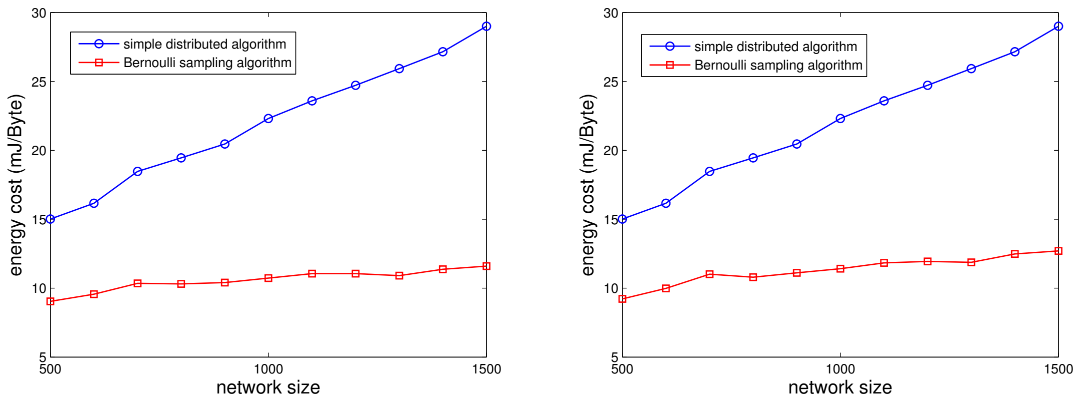
1. The Bernoulli sampling based aggregation algorithm can be used in unclustered networks.

2. The Bernoulli sampling based aggregation algorithm has lower energy cost in small-scale networks.

While on the other hand, the uniform sampling algorithm is appropriate for large-scale clustered networks.

6.6 Conclusions

In this chapter, the δ-approximate algorithms for the maximum value and distinct-set aggregation operations in sensor equipped IoT networks are proposed. These algorithms are based on the uniform sampling and Bernoulli sampling respectively. Mathematical proofs have been made for better understanding of these algorithms. Additionally, we have also proposed mathematical estimators for the two algorithms. Moreover, we have derived the values for the sample size and the sample probability which satisfies the specified failure

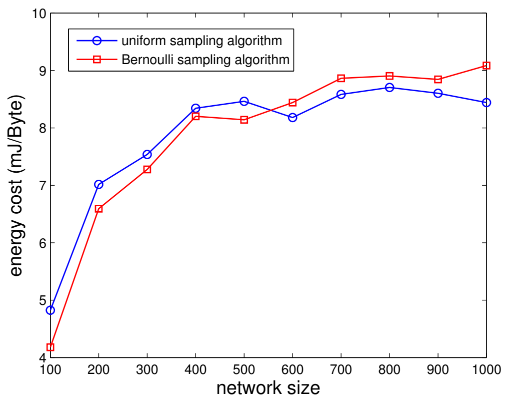


(a) Maximum Value (b) Distinct-Set

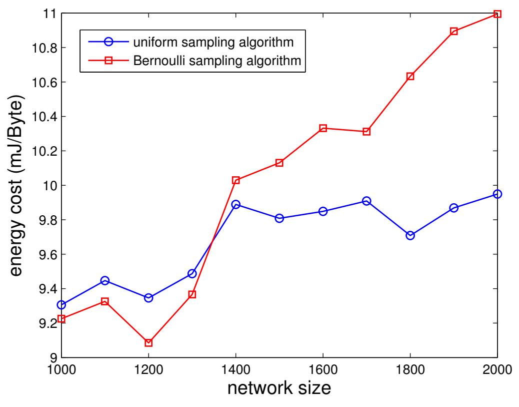
Figure 6.6. Energy cost comparison between the Bernoulli sampling based aggregation algorithm and the simple distributed algorithm.

probability requirements of the final result. Finally, a uniform sampling based algorithm and a Bernoulli sampling based algorithm are provided.

Experiments are conducted for various delta values and the network sizes. The results are then compared between the naive method and the proposed algorithms. The simulation results indicate that the proposed algorithms have high performance with respect to the energy cost.



(a) Maximum Value



(b) Distinct-Set

Figure 6.7. Energy cost comparison between the uniform sampling based aggregation algorithm and Bernoulli sampling based aggregation algorithm.

Chapter 7

DATA COLLECTION IN GEOGRAPHICAL POSITION DEPENDENT MOBILE SENSING

7.1 Introduction

Nowadays, smartphones have gained popularity and are playing a key role in multiple aspects of people's daily life such as entertainment, communication and social activities. Most smartphones are equipped with embedded sensors such as camera, GPS (Global Positioning System), accelerometer, etc [95] [143] [144] [145] [146] [147]. Moreover, the popularity of wearable devices also enhances the smartphones' sensing capabilities by exchanging sensory data with smartphones using network interfaces. Therefore, in various fields, mobile sensing has numerous innovative applications including environmental transportation, monitoring, safety, healthcare and so on [1] [148] [149] [150] [151] [152].

Mobile crowdsensing is to make use of the sensing capabilities of smartphones to get sensory data. This problem has been gaining significant popularity in recent years. Multiple works have studied the problem of incentive mechanism design in mobile crowdsensing platforms [101] [102]. In traditional applications and systems of mobile sensing such as [153] and [154], the mobile sensing platform is based on voluntary participation. While in mobile crowdsensing, task participants get paid for their contribution [155] [156] [157]. The reason is the participants have their own resource consumption such as battery and data transition fee while participating in a sensing task. For example, the following process is used to gather sensory data for multiple general-purpose mobile crowdsensing platforms [103],

1. The mobile crowdsensing platform broadcasts a mobile sensing task to the users.

2. The users transmit feedback to the mobile crowdsensing platform. A feedback can be a detailed sensing plan or a bid which is the expected price at which the user wants to

sell the sensory data.

3. The mobile crowdsensing platform selects some smartphone users as winners for participating in the mobile sensing task.

4. The mobile crowdsensing platform makes payments to the winners. The winners submit their sensory data to the mobile crowdsensing platform.

In real-life applications, the mobile crowdsensing platforms aim to maximize the quality of gathered sensory data during the above process because the mobile crowdsensing platform needs to pay significant economic costs to the mobile crowdsensing participants for their sensory data. However, in many applications, the sensory data exhibits strong spatial correlation [25], [26]. Directly selecting the winners with lower bids may select multiple disjoint users as the winners, which cannot fully describe sensory data's spatial correlation or show the whole circumstance. This phenomenon is shown in the example in Section 7.2.2. Therefore, it is necessary to design incentive mechanisms to solve the above problem.

This chapter is to study incentive mechanisms in the geographical position dependent mobile crowdsensing platforms. Unlike the previous chapter, this chapter is to select users who could form a connected graph. In this problem, the mobile crowdsensing platform cannot select multiple disjoint user as the winners. All the selected winners must form a connected dependent graph so that the sensory data can fully describe the whole circumstance of the sensory region. Then we have the following two problems to be solved.

1. How to select the winner set to maximize the social welfare for a mobile crowdsensing task while the selected winners form a connected dependent graph.

2. How to calculate appropriate payments to the winners?

This chapter proposes algorithms to solve the above two problems. Moreover, we also propose and solve two extended winner selection problems, which are the min-K and budget-bounded winner selection problem, respectively. In summary, the main contributions of our chapter are as follows.

1. A geographical position dependent winner selection problem is proposed and formulated as an optimization problem.

2. Proposes two algorithms to select winners and calculate winners' payments. The first algorithm is an A* algorithm which can maximize the total social welfare. The second algorithm has polynomial time complexity while achieves sub-optimal total social welfare only.

3. Proposes two extended winner selection problems. Another two algorithms are proposed to solve the extended problems.

4. Offers solid theoretical proofs which indicate the proposed algorithms have beneficial properties such as truthfulness, individual rationality and profitability.

5. Real datasets based experimental results are presented, which indicate the proposed algorithms have high performance.

The rest of this chapter is organized as follows. Section 7.2 shows the problem definition. Section 7.3 and Section 7.4 are about the optimal and non-optimal winner selection algorithms. Section 7.5 is about the algorithms for two extended problems. Experimental results are provided in Section 7.6. Finally, Section 7.7 concludes this chapter.

7.2 Problem Definition

7.2.1 Crowdsensing Platform

Suppose there are n users deployed in a region and the set of users is denoted by $U = \{u_1, u_2, \ldots, u_n\}$. Each user in U carries a smartphone equipped with positioning device. $u.x$ and $u.y$ are used to denote user u's x-coordinate and y-coordinate. $R(x, y, r)$ is used to denote the circular region around position (x, y) with radius r which is defined as

$$R(x, y, r) = \{(x', y') | (x' - x)^2 + (y' - y)^2 \leq r^2\}.$$

Suppose a snapshot crowdsensensing task T is proposed by the mobile crowdsensing platform. Task T invites users to submit sensory data in a sensory data gathering region $R(T.x, T.y, T.r)$. $T.x$ and $T.y$ are the x-coordinate and y-coordinate of the center of $R(T.x, T.y, T.r)$ and $T.r$ is task T's sensing radius.

$$U_T = \{u \in U | (u.x, u.y) \in R(T.x, T.y, T.r)\}$$

is used to denote the users in the sensory data gathering region $R(T.x, T.y, T.r)$. After the crowdsensing platform broadcasts the crowdsensing task T, all the users in U will receive it. But only the users in U_T will participate in the sensing task T due to the following reason.

1. Snapshot sensing task has high time-sensitiveness
2. There is massive moving cost for the users in $U - U_T$ to enter the region $R(T.x, T.y, T.r)$ [17].

The mobile crowdsensing platform uses the following process to collect sensory data.

1. The crowdsensing task T is sent to all the users by the crowdsensing platform.
2. Each user $u \in U_T$ submits its bid $u.b$ and position $(u.x, u.y)$ to the mobile crowdsensing platform. The bid $u.b$ is user u's expected price to sell the sensory data for. The users in $U - U_T$ will not further paticipate the crowdsensing task.
3. The mobile crowdsensing platform calculates the winner set W and the payment $w.p$ for each winner $w \in W$, then sends the payments to each winner.
4. The winners send their sensory data to the crowdsensing platform.

The above process is shown in Fig.7.1.

We use the total social welfare of the mobile crowdsensing platform to evaluate the winner set W. Suppose P is the profit the crowdsensing platform acquires from a single user's sensory data. Since the mobile sensing platform does not know the exact payment to

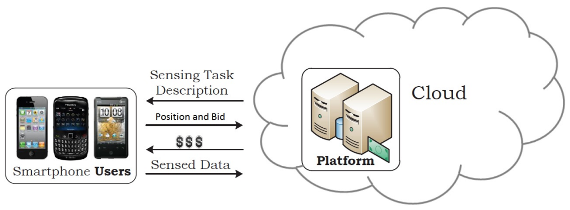


Figure 7.1. Mobile sensing platform

each user during the process of winner selection, then the total social welfare is defined as its total profile minus the sum of winners' bids. In other words, the total social welfare equals to

$$P|W| - \sum_{w \in W} w.b$$

where $|W|$ is the size of winner set W.

In practice, users may submit bids differ from their true valuations to improve their utility. However, the theoretical proofs in Section 7.3, Section 7.4 and Section 7.5 indicates the proposed algorithms are truthful. In other words, one user's profit is maximized if its bid is equal to its true cost. Therefore, $u.b$ can be regarded as user u's true cost.

7.2.2 Optimal Winner Selection Problem

According to the basic setting introduced in Section 7.2.1, there is a straightforward method for the mobile crowdsensing platform to maximize the total social welfare, which is to select the all the users whose bids are lower than P as the winners. However, in actual applications, the sensory data may have strong spatial correlation [25], [26]. Directly using the above method will have the following drawback.

Since each user decides its own bid independently and some users may have high bids in order to get high payments, it is likely that users whose bids are lower than P sparsely distribute in the sensing region. In other words, the users whose bids are lower than P may be very far away from each other. In this way, directly using the above method may select multiple disjoint users as the winners. Therefore, the positions for the gathered sensory data will also be far away from each other, which cannot fully describe sensory data's spatial correlation. The above problem can be verified in the example shown in Fig. 7.2. In order to solve the above problem, we have the following definitions of dependent users, dependent graph and connected winner set.

Definition 7.2.1 (dependent users). *User u and user v are dependent users if and only if $dis(u, v) \le T.d$ where*

$$dis(u, v) = \sqrt{(u.x - v.x)^2 + (u.y - v.y)^2}$$

is the Euclidean distance between user u and user v and $T.d$ is a predefined distance.

Definition 7.2.2 (dependent graph). *Let $G = (V, E)$ to denote a graph with node set V and edge set E. For a winner set W, the dependent graph of W is defined as*

$$DG(W) = G(W, E(W))$$

where

$$E(W) = \{(u, v) | u, v \in W \wedge u, v \text{ are dependent users}\}.$$

Definition 7.2.3 (connected winner set). *A winner set W is a connected winner set if and only if the dependent graph $DG(W)$ is a connected graph.*

Based on the above winner determination setting, the problem of optimal winner selection is defined as follows.

Input: user set U and mobile sensing task T.

Output: 1) connected winner set W which maximizes $P|W| - \sum_{w \in W} w.b$. 2) The payment for each winner $w \in W$.

Fig.7.2 shows an example of our optimal winner selection problem. Suppose the user set is $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$. The blue-shadowed circle is the sensory data gathering region. The circles are used to denote the users and the number in a circle is user's ID. The number beside a circle is the corresponding user's weight which is defined as the value of P

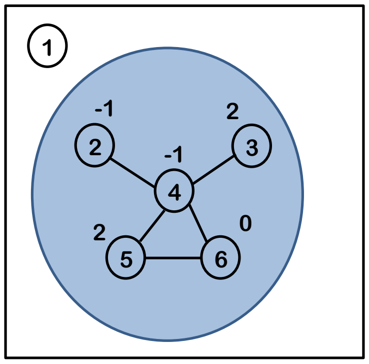


Figure 7.2. Example of the winner selection problem

minus corresponding user's bid. Two dependent users are connected by an edge. According to the sensory data gathering region, we have $U_T = \{u_2, u_3, u_4, u_5, u_6\}$, and u_1 cannot be a winner because $u_1 \notin U_T$. If we have $P = 3$, $u_2.b = 4$, $u_3.b = 1$, $u_4.b = 4$, $u_5.b = 1$, $u_6.b = 3$, then the optimal winner set is $\{u_3, u_4, u_5\}$ with social welfare as 3. Although using $\{u_3, u_5\}$ as the optimal winner set can further increase the total social welfare, but $\{u_3, u_5\}$ is not a connected winner set, which does not satisfy the conditions of our problem definition.

This chapter is to design incentive mechanisms which can select the winners and determine winners' payments. Moreover, the proposed incentive mechanisms should satisfy the following properties:

1. Computational efficiency: In practice, the number of users who participants in mobile crowdsensing may be very large [2], [3]. Therefore, the incentive mechanism should be computational efficient. A mechanism is computationally efficient if the outcome can be calculated in polynomial time.

2. Truthfulness: In mobile crowdsensing, users may submit bids differ from their true valuations to improve their utility. This may significantly affect the fairness of mobile crowdsensing. Therefore, the incentive mechanism needs to be truthful which means no bidder can improve its utility by submitting a bid different from its true valuation, no matter what others submit.

3. Individual rationality: Each participant has a non-negative utility. In other words, one

participant's payment must be greater or equal to its actual cost in order to encourage user's participation.

4. Profitability: The platform cannnot incur a deficit. The value brought by the winners should be greater than or equal to the payment to the winners.

7.2.3 Problem Formulation

Boolean variable x_i is used to denote if user u_i is the winner or not, in other words,

$$x_i = \begin{cases} 1 & \text{if} \;\; u_i \in W \\ 0 & \text{if} \;\; u_i \notin W \end{cases}$$

$ISG(V)$ is used to denote the induced subgraph of node set V, which is defined as follows

$$ISG(V) = G(V, \{(u,v)|u \in V \wedge v \in V \wedge dis(u,v) \leq T.d\}).$$

Moreover, $\mathit{connected}(G)$ is used to denote whether graph G is connected or not, in other words,

$$\mathit{connected}(G) = \begin{cases} 1 & \text{if} \;\; G \text{ is connected} \\ 0 & \text{if} \;\; G \text{ is not connected} \end{cases}$$

Then the optimal winner selection problem can be formalized as the following optimization problem.

$$\max \quad P\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i b_i$$

$$\text{s.t.} \quad x_i \in \{0,1\}, 1 \leq i \leq |U_T|$$

$$\mathit{connected}(ISG(\{U[i]|x_i = 1\})) = 1.$$

In order to increase the readability of this chapter, we summarized frequently used symbols in Table 7.1.

Symbol	Description
$u.x/u.y$	x/y-coordinate of user u's position
$u.b/u.w/u.p$	user u's bid/weight/payment
$T.x/T.y$	x/y-coordinate of sensory data gathering region center
$T.r$	sensory data gathering region's radius
$T.d$	dependent distance between two users
$dis(u, v)$	Euclidean distance between user u and user v
$R(x, y, r)$	$\{(x', y') \in [0, 1)^2 \| (x' - x)^2 + (y' - y)^2 \leq r^2\}$
U	the set of users
U_T	Users in the data gathering region of task T
W	winner set
P	the profit of one user's sensory data

Table 7.1. Symbols

7.3 Optimal Winner Selection Algorithm

According to Section 7.2, our optimal winner selection problem can be formulated as the maximum weight connected subgraph problem where each user is a node in the graph and the weight of node u is defined as $P - u.b$ [158]. Obviously, the following brute-force algorithm can be used to solve the optimal winner selection problem.

1. Enumerate all subsets of U_T.
2. Remove the subsets which are not connected winner sets.
3. Find the subset with the maximum social welfare.

However, this algorithm has huge computation cost since it needs to examine all the possible subsets of U_T. Therefore, the following A* algorithm is proposed to efficiently select winner set which can maximal the social welfare.

7.3.1 Algorithm to Select the Winners

In our algorithm, a winner set is denoted by a 0-1 bit string. For a 0-1 bit string *str* and node set U where all the nodes are sorted in ascending order of their IDs, we have

$$str[i] = \begin{cases} 1 & \text{if } U[i] \text{ is the winner} \\ 0 & \text{elsewhere} \end{cases}$$

For the example shown in Fig.7.2, 0-1 bit string "01110" is used to denote the optimal winner set $\{u_3, u_4, u_5\}$ Then the our algorithm is to find a 0-1 bit string to denote the optimal winner set.

We use Q to denote the priority queue of the A* algorithm. Multiple 0-1 bit strings which are used to denote all the possible winner combinations are stored in Q. Heuristic function HF is used to calculate the priority of the elements in Q. The detailed steps of HF are introduced in the remaining part of this section. Q.top is the element in Q which has the highest priority. $Q.pop()$ is used to indicate the operation of removing the element with the highest priority. $Q.push()$ is used to denote the operation of inserting a new element into priority queue Q. Finally, the basic process of our A* algorithm is as follows.

1. Calculate the lower bound of maximum connected subgraph's total weight, push 0-1 bit string "" into Q.

2. While $Q \neq \emptyset$ and $|Q.\text{top}| < |U_T|$, pop Q.top as q, then examine $q+$"0" and $q+$"1". For each of them, use the heuristic function HF to compute its score, if the score is no less than the lower bound computed in step 1, push the examined string into Q.

3. Calculate the final winner set according to Q.top.

The detailed process is shown in Algorithm 14. LB denotes the lower bound of maximum connected subgraph's total weight and HF denotes the heuristic function of the A* algorithm.

According to the process of A* algorithm, the HF function should calculate the upper bound of connected winner set's total weight. The basic procedure of our heuristic function is as follows.

1. If $|Q.\text{top}| = |U_T|$, the heuristic function returns 0 if $ISG(\{U_T[i]|\text{str[i]=1}\})$ is connected. otherwise, the heuristic function returns $-\infty$.
2. If $|Q.\text{top}| < |U_T|$, the heuristic function returns the total weight for the undecided users with positive weights.

The detailed process of the heuristic function HF is shown in Algorithm 15. The symbol $u.w$ is used to denote user u's weight which is defined as

$$u.w = P - u.b.$$

Algorithm 14: Optimal Winner Selection Algorithm Opt(U_T)

```
Input: User set $U_T$
Output: Winner set $W$
1: $LB = max(\max_{u \in U_T} u.w, 0)$
2: $Q$.push(“”)
3: while $Q \neq \emptyset$ do
4:    if $|Q.\text{top}| = |U_T|$ then
5:       break
6:    end if
7:    $q = Q.\text{top}$, $Q$.pop()
8:    $q_0 = q+$“0”, $q_1 = q+$“1”
9:    if HF($q_0$) $\geq LB$ then
10:      $Q$.push($q_0$)
11:   end if
12:   if HF($q_1$) $\geq LB$ then
13:      $Q$.push($q_1$)
14:   end if
15: end while
16: return $\{U_T[i]|Q.top[i] = 1\}$
```

We can see that the heuristic function will return $-\infty$ if the induced subgraph is not connected. This will make sure that the selected winners will form a connected geographical dependent graph.

Algorithm 15: Heuristic Function HF

```
Input: 0-1 bit string $str$
Output: Upper bound of maximum independent set's total weight if $str$ is further extended
1: if $|str| = |U_T|$ then
2:     if Connected($ISG(\{U_T[i]|\text{str[i]=1}\})$)=false then
3:         return $-\infty$
4:     else
5:         return 0
6:     end if
7: end if
8: sum=0
9: for i=$|str|+1$ to $U_T$ do
10:    if $U_T[i].w > 0$ then
11:        sum=sum+$U_T[i].w$
12:    end if
13: end for
14: return sum
```

7.3.2 Payment Determination

After the optimal winners are selected, we will use the Vickrey-Clarke-Groves (VCG) auction to calculate each winner's payment [108]. The detailed process for the payment determination is shown in Algorithm 16.

Algorithm 16: Payment Determination Algorithm

```
Input: Winner set $W$ and $U'_T$
Output: Payment for each winner $u \in W$
1: $maxw = \sum_{u \in W} u.w$
2: for $u \in W$ do
3:     $W$'=Opt($U'_T - \{u\}$)
4:     $maxw' = \sum_{v \in W'} v.w$
5:     $u.p = maxw - maxw'$
6: end for
7: return $W$
```

According to Algorithm 14, we can easily find that the winner set W calculated by Algorithm 14 is a connected winner set and the total social welfare is maximized. Moreover, we can conclude that the payment strategy is truthful, individual rational and profitable according to [108].

7.4 Non-optimal Winner Selection Algorithm

Although Section 7.3 introduces an A* algorithm which can maximize the total social welfare and it uses the heuristic function to reduce the computation cost. However, according to the following theorem [159], this algorithm still has high computational complexity.

Theorem 7.4.1. *Maximum weighted connected subgraph problem is NP-complete.*

While on the other hand, the number of participants can be quite large in many crowd-sensing applications [2], [3]. Therefore, it is necessary to propose a computationally efficient winner selection algorithm even if the total social welfare may not be maximized.

Obviously, there is straightforward approach to solve the above problem, whose process is as follows.

1. Decide a winner set using an existing approximate algorithm with polynomial complicity.
2. Decide the payment for each winner using the VCG auction.

However, [109] indicates VCG auction loses its truthful property if it the total social welfare is not maximized. Therefore, we propose another winner selection algorithm based on a theorem in [110], which is as follows.

Theorem 7.4.2. *An auction mechanism is truthful if and only if:*

1. *The selection rule is monotone: if user i wins the auction by bidding b_i, it also wins by bidding $b'_i \leq b_i$.*
2. *Each winner is paid the critical value: user i would not win the auction if it bids higher than this value.*

Moreover, this winner selection algorithm is based on the following definition.

Definition 7.4.1 (maximum positive connected component). *Graph G's connected component C is a maximum positive connected component if and only if:*

1. *Every node in C has positive weight.*

2. *C cannot be further expended by adding another node.*

For the example shown in Fig. 7.2, we can see that $\{u_3\}$ and $\{u_5\}$ are maximum positive connected components while $\{u_3, u_4, u_5\}$ is not a maximum positive connected component although $\{u_3, u_4, u_5\}$ has greater total weight.

7.4.1 Algorithm Design

Based on *Theorem 7.4.2* and *Definition 7.4.1*, we have the following algorithm to select the winners and determine winners' payments where C_i is used to denote the i-th maximum positive connected component and m is used to denote the number of maximum positive connected components.

1. Set $W = \emptyset$, check each maximum positive connected component $C_1, C_2, \ldots, C_m$ of graph G

2. Set $W = C_i$ if $\sum_{u \in W} u.w < \sum_{v \in C_i} v.w$

3. All winners' payments are set to P

The above process is shown in Algorithm 17.

According to Algorithm 17's process, we can see that it will select one maximum positive connected component in $C_1, C_2, \ldots, C_m$. This will make sure the selected winners form a connected geographical dependent graph.

7.4.2 Algorithm's Properties

According to the process of Algorithm 17, we also have the following theorems which indicate the proposed algorithm is computational efficient, truthful, individual rational and profitable.

Algorithm 17: Non-optimal Winner Selection Algorithm

```
Input: User set $U_T$ and mobile crowdsensing task $T$
Output: Winner set $W$
1: $W = \emptyset$
2: for i=1 to $m$ do
3:     if $\sum_{u \in W} u.w < \sum_{v \in C_i} v.w$ then
4:         $W = C_i$
5:     end if
6: end for
7: for i=1 to $|W|$ do
8:     $W[i].p = P$
9: end for
10: return $W$
```

Theorem 7.4.3. *The time complexity of Algorithm 17 is $O(U_T^2)$*

Proof: All the maximum positive connected components can be calculated using breadth-first search whose time complexity is $O(U_T^2)$. Moreover, the time complexity of checking all the maximum positive connected components and calculating the payments are $O(U_T^2)$ and $O(U_T)$ since we have $m \leq |U_T|$ and $C_i \leq |U_T|$ $1 \leq i \leq m$. Finally, the total time complexity is $O(U_T^2)$. □

Theorem 7.4.4. *The suboptimal auction mechanism is truthful.*

Proof: According to the process of Algorithm 17, Algorithm 17 is to choose the maximum positive connected component with the highest total weight. So if user $u \in C_i$ proposes another bid $u'_b < u_b$, the total weight of C_i will become even larger. Then user u will still be chosen as the winner. Therefore, we conclude that Algorithm 17 is monotone.

According to the process of Algorithm 17, each winner's payment is P. If user u's bid is greater than P, u will not belong to any maximum positive connected component, which makes u cannot be the winner. Therefore, we conclude that each winner in Algorithm 17 is paid the critical value.

Since Algorithm 17 is monotone and each winner is paid the critical value, we can conclude that Algorithm 17 is truthful according to *Theorem 7.4.2.* □

Theorem 7.4.5. *The non-optimal auction is individual rational.*

Proof: According to the process of Algorithm 7.4.2, each winner's payment is P. Then this theorem can be easily proved according to the definition of maximum positive connected component. $\square$

Theorem 7.4.6. *The non-optimal auction is profitable.*

Proof: According to the process of Algorithm , the payment to each winner is P. In other words, $|W|P - \sum_{w \in W} w.p$ will always be 0. Therefore, the value brought by the winners is no less than the total payment paid to the winners $\square$

7.5 Problem Extension

Although the above two sections proposed two algorithms to select the winners, there are still some troublesome problems to be solved, which could be regarded as the extension of the winner selection problem defined in Section 7.2.

7.5.1 Min-K Winner Selection Problem

In practice, the mobile crowdsensing platform may need enough sensory data to do the analysis. Therefore, it is possible to select a connected winner set while letting the number of winners to be greater or equal to a given value K? The formal problem definition is as follows.

Input: user set U where each user $u \in U$ has attributes $u.x$, $u.y$ and $u.b$.

Output: 1) connected winner set W which maximizes $P|W| - \sum_{w \in W} w.b$ and $|W| \geq K$. 2) The payment for each winner.

For the example shown in Fig.7.2, if we have $K = 5$, then the optimal winner set is $\{u_2, u_3, u_4, u_5, u_6\}$ where the total social welfare is 2. Comparing with the optimal winner set $\{u_3, u_4, u_5\}$ for the original winner selection problem, we can see that in this problem, the mobile crowdsensing platform may select some users whose bid is equal or even greater than P in order to let the number of winners to be greater or equal to K. In this way, the total social welfare may be less than that of the original winner selection problem.

This problem can be solved by the following methodology where the total social welfare can be maximized.

1. The mobile crowdsensing platform still uses Algorithm 14 to select the winners and Algorithm 16 to calculate the payments.
2. Revise the original heuristic function. If $|str| = |U_T|$ and the number of winners is less than K, the heuristic function returns $-\infty$.

The revised heuristic function is shown in Algorithm 18. According to [108], we can conclude that the payment strategy for the min-K winner selection problem is truthful, individual rational and profitable.

Algorithm 18: Heuristic Function HF

```
Input: 0-1 bit string $str$, $k$
Output: Upper bound of maximum independent set's total weight if $str$ is further
    extended
if $|str| = |U_T|$ then
    if Connected(ISG($\{U_T[i]|$str[i]=1$\}$))=false then
        return $-\infty$
    else if $|\{1 \leq i \leq |U_T||str[i] = 1\}| < k$ then
        return $-\infty$
    else
        return 0
    end if
end if
sum=0
for i=$|str| + 1$ to $U_T$ do
    if $U_T[i].w > 0$ then
        sum=sum+$U_T[i].w$
    end if
end for
return sum
```

7.5.2 Budget-bounded Winner Selection Problem

Since the mobile crowdsensing platform need to pay enormous payments to the winners, then how does the crowdsensing platform select the winners while the total payment to the winners is less than or equal to a given budget B? The formal problem definition is as follows.

Input: user set U where each user $u \in U$ has attributes $u.x$, $u.y$ and $u.b$.

Output: 1) connected winner set W which maximizes $P|W| - \sum_{w \in W} w.b$ and $|W| \geq K$. 2) The payment for each winner where the total payment is less or equal to budget B.

This problem can be solved using the following methodology which has polynomial execution time.

1. Calculate the initial winner set W using Algorithm 17. Return W as the winner set if the total payment is less than or equal to B.
2. If the total payment is greater than B, use the following steps to remove some winners from W.
    (a) Check the winners in W in decreasing order of bid.
    (b) Remove winner w from winner set W if $Connected(ISG(W - \{w\})) = \textbf{true}$.
    (c) The above process keep on until the total payment is less than or equal to B.
3. Return the winner set W.

The above process is shown in Algorithm 19.

Using the similar methodology shown in Section 7.4.2, it is easy to have the following theorem.

Theorem 7.5.1. *Algorithm 19 is computational efficient, truthful, individual rational and profitable.*

For the example shown in Fig.7.2, if we have $B = 2.4$, then output winner set of Algorithm 19 is $\emptyset$ where the total social welfare is 0. Comparing with the output winner set

Algorithm 19: Non-optimal Winner Selection Algorithm

```
Input: User set $U_T$, crowdsensing task $T$ and budget $B$
Output: Winner set $W$
1: $W$=Non-optimal($U_T, T$)
2: Sort the winners in $W$ in increasing order of bid
3: while $P|W| > B$ do
4:    for i=1 to $|W|$ do
5:       if $Connected(ISG(W - \{W[i]\})) =$ true then
6:          $W = W - \{W[i]\}$
7:       end if
8:    end for
9: end while
10: return $W$
```

$\{u_3\}$ of Algorithm 17, we can see that in Algorithm 19, the mobile crowdsensing platform may select fewer winners in order to let the total payment to be less or equal to budget B. Similarly, the total social welfare of this problem may be less than that of the original winner selection problem.

7.6 Experiment

7.6.1 Experiment Settings

We investigated one real dataset from trip record data from New York City Taxi & Limousine Commission to generate users' positions [160]. The preprocessed data contains the positions of 413 users and these users are distributed in a $4.68km \times 13.20km$ rectangle region in downtown Manhattan.

The Swoopo bidding dataset named is used to generate users' bid [113]. The following steps are used to preprocess the raw data.

1. A subset of the original dataset which only contains the bids of one single good is extracted.

2. For simplicity, all the original bids are normalized by dividing the bid average.

Parameter	Value	Parameter	Value
$T.x$	2km	$T.y$	4km
$T.r$	600m	$T.d$	300m
P	1	K	5

Table 7.2. Default Parameters

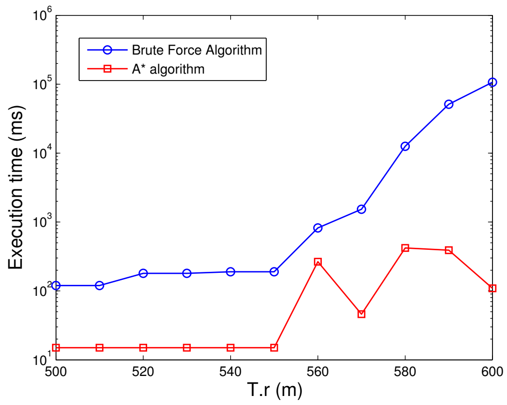


Figure 7.3. Execution time comparison for different $T.r$

There are 5372 bids in the preprocessed bid set. The bid of each individual user is determined by randomly selecting one bid from this bid set. The parameters and their default values for the experiments are listed in Table 7.2. The setting for the value $T.r$ is relatively low since the baseline algorithm has exponential time complexity. The experiment results for the optimal winner selection algorithm, non-optimal winner selection problem and the extended winner selection problems are shown in Section 7.6.2, Section 7.6.3 and Section 7.6.4, respectively.

7.6.2 Optimal Winner Selection Algorithm

The first group of experiments is to compare the execution time between the A* algorithm and the brute force algorithm with different $T.r$. The results are shown in Fig. 7.3 where $T.r$ varies from 500m to 600m. We can see the execution time of the brute force algorithm increases sharply with the increase of $T.r$ due to the increase of $|U_T|$ while the execution time of the A* algorithm varies with the increase of $T.r$. Moreover, the execution time of the A* algorithm is lower than that of the brute force algorithm, which demonstrates the efficiency of the algorithm we proposed.

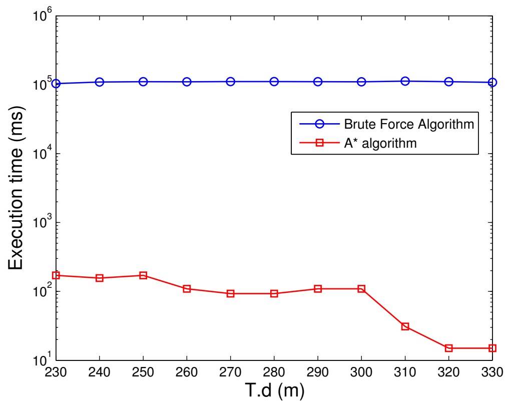


Figure 7.4. Execution time comparison for different $T.d$

The second group of experiments compares the execution time between the A* algorithm and the brute force algorithm with different $T.d$. The value of $T.d$ changes from 230m to 330m. The results are listed in Fig. 7.4. We can see the execution time of the A* algorithm is much lower than the brute force algorithm with different $T.d$.

7.6.3 Non-optimal Winner Selection Algorithm

The first group of experiments is to compare the social welfare, *i.e.* $|W|P - \sum_{u \in W} u.b$ of the optimal winner selection algorithm and non-optimal winner selection algorithm with different $T.r$. The results are shown in Fig. 7.5. We can see that for the same $T.d$, the non-optimal algorithm's total social welfare is smaller than the optimal algorithm's welfare. This kind of result illustrates the tradeoff between the total social welfare and the algorithm performance.

The second group of experiments is to compare the social welfare of the optimal winner selection algorithm and non-optimal winner selection algorithm with different $T.d$. The results are shown in Fig. 7.6. Similarly, we can see that for different $T.d$, the non-optimal algorithm's social welfare is smaller than the optimal algorithm's social welfare although the non-optimal algorithm has polynomial execution time.

The third group of experiments shows the bid of the winners. We set $T.r = 2km$ for this group of experiments. The results are shown in Fig. 7.7. We can see that winners' bids are

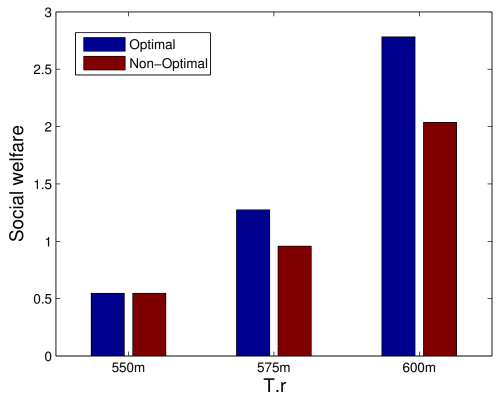


Figure 7.5. Social welfare comparison for different $T.r$

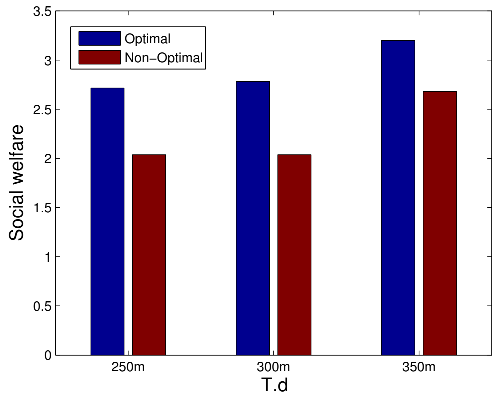


Figure 7.6. Social welfare comparison for different $T.d$

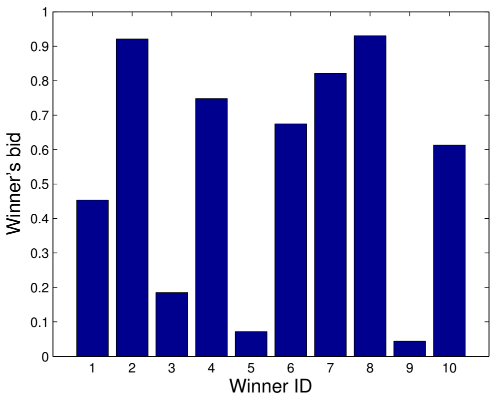


Figure 7.7. Bids of the winners

lower than the payments they get, which is equal to $P = 1$. This kind of results verifies the correctness of Theorem 7.4.5, in other words, the non-optimal winner selection algorithm is individual rational.

7.6.4 Extended Problems

The first group of experiments is to compare the execution time between the A* algorithm and the brute force algorithm with different $T.r$ for the min-K winner selection problem. The results are shown in Fig. 7.8. Similarly, we can see the execution time of the brute force algorithm increases sharply with the increase of $T.r$ while the execution time of the A* algorithm varies with the increase of $T.r$. Moreover, the execution time of the A* algorithm is lower than that of the brute force algorithm.

The second group of experiments compares the execution time between the A* algorithm and the brute force algorithm with different $T.d$ for the min-K winner selection problem. The results are listed in Fig. 7.9. Similarly, we can see the execution time of the A* algorithm is much lower than the brute force algorithm with different $T.d$.

The third group of experiments is to compare the total payment to the winners and the budget for the budget-bounded winner selection problem. The results are listed in Fig. 7.10. The results show the total payment calculate by Algorithm 7.4.2 is always less than the budget B as expected.

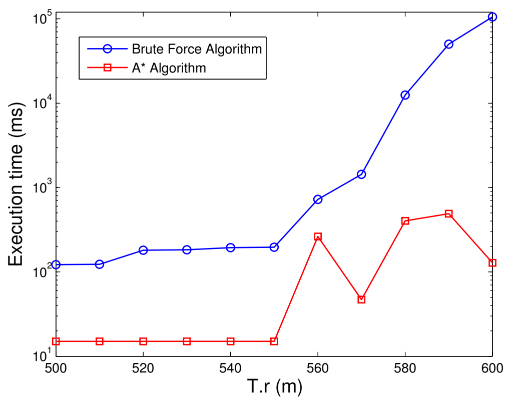


Figure 7.8. Execution time comparison for different $T.r$

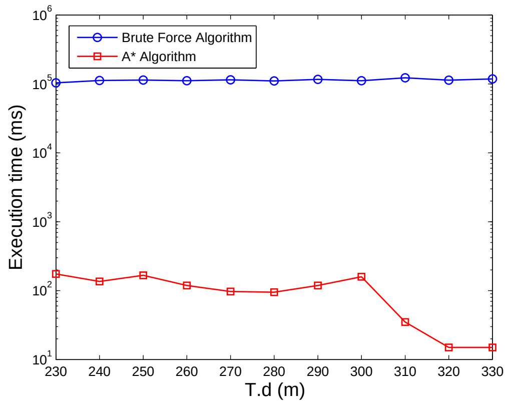


Figure 7.9. Execution time comparison for different $T.d$

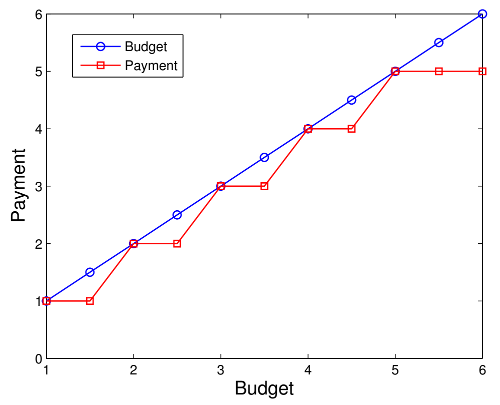


Figure 7.10. Total payments under different budgets

7.7 Conclusion

We propose a geographical position dependent winner selection problem in mobile crowdsensing in this chapter. An optimal winner selection algorithm which can maximize the total social welfare is proposed. We also designed a non-optimal winner selection algorithm which is computationally efficient. Moreover, we also proposed and solved two extended problems. Solid theoretical proofs indicate these algorithms are truthful, individual rational and profitable. The real datasets based experimental results indicate the proposed algorithms have high performance.

Chapter 8

CONCLUSION

Data collection and aggregation has been considered as essential techniques in mobile sensing. Using the naive methodology to do data collection and data aggregation in mobile sensing will cause huge data transmission cost. Therefore, to reduce the cost, this dissertation follows the following technical routes. The first route is to use sampling technique which only let a small part of nodes to submit their sensory data. The second route is to make use of users' positions to reduce useless sensory data submission. Based on the above two technical routes the following problems are solved.

First, the (ϵ, δ)-approximate algorithms for the frequency, rank, distinct-count and quantile aggregation operations in networks are proposed. Furthermore, the sample size which can make the final result to satisfy the specified precision and failure probability requirements are derived. In addition, a cluster-based uniform sampling algorithm is provided. The simulation results show that the proposed algorithms have high performance on both energy cost and accuracy.

Second, an (ϵ, δ)-approximate algorithm to process a frequency query is proposed. This algorithm is based on Bernoulli sampling, so only some nodes in a network need to transfer sensory data. Furthermore, the methodology to calculate the sampling probability which can make the final query result to satisfy the specified precision and failure probability is derived. The simulation results for MANET connectivity are presented. These simulation results can be used in the calculation of sampling probability. Finally, a Bernoulli sampling algorithm is provided and analyzed. The simulation results show that on the aspects of both energy efficiency and accuracy, the proposed algorithm has high performance.

Third, we propose a geographical position conflicting based winner selection problem in mobile crowdsensing. An optimal winner selection algorithm is proposed to maximize the total social welfare. Moreover, we also propose a computationally efficient non-optimal winner selection algorithm. Solid theoretical proofs indicate both these algorithms are truthful, individual rational and profitable. The experimental results which are based on actual datasets indicate the proposed algorithms are efficient. An interesting aspect of research in the future is to make use of the sensory data's spatial correlation to study users' cooperation in geographical position based crowdsensing tasks.

Fourth, the δ-approximate algorithms for the maximum value and distinct-set aggregation operations in sensor-equipped IoT networks are proposed. These algorithms are based on the uniform sampling and Bernoulli sampling respectively. Mathematical proofs have been made for better understanding of these algorithms. Additionally, we have also proposed mathematical estimators for the two algorithms. Moreover, we have derived the values for the sample size and the sample probability which satisfies the specified failure probability requirements of the final result. Finally, a uniform sampling based algorithm and a Bernoulli sampling based algorithm are provided. Experiments are conducted for various delta values and the network sizes. The results are then compared between the naive method and the proposed algorithms. The simulation results indicate that the proposed algorithms have high performance with respect to the energy cost.

Finally, we propose a geographical position dependent winner selection problem in mobile crowdsensing in this paper. An optimal winner selection algorithm which can maximize the total social welfare is proposed. We also designed a non-optimal winner selection algorithm which is computationally efficient. Moreover, we also proposed and solved two extended problems. Solid theoretical proofs indicate these algorithms are truthful, individual rational and profitable. The real datasets based experimental results indicate the proposed algorithms have high performance.

REFERENCES

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[2] R. K. Balan, K. X. Nguyen, and L. Jiang, "Real-time trip information service for a large taxi fleet," in *Proceedings of the 9th international conference on Mobile systems, applications, and services.* ACM, 2011, pp. 99–112.

[3] S. Wakamiya, R. Lee, and K. Sumiya, "Crowd-sourced urban life monitoring: urban area characterization based crowd behavioral patterns from twitter," in *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication.* ACM, 2012, p. 26.

[4] *MPR-Mote Processor Radio Board User's Manual*, Crossbrow Inc.

[5] A. Boulis, S. Ganeriwal, and M. B. Srivastava, "Aggregation in sensor networks: an energy–accuracy trade-off," *Ad hoc networks*, vol. 1, no. 2, pp. 317–331, 2003.

[6] X. Tang and J. Xu, "Extending network lifetime for precision-constrained data aggregation in wireless sensor networks," in *Proceedings IEEE INFOCOM 2006. 25th IEEE International Conference on Computer Communications*, April 2006, pp. 1–12.

[7] Z. Huang, L. Wang, K. Yi, and Y. Liu, "Sampling based algorithms for quantile computation in sensor networks," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data.* ACM, 2011, pp. 745–756.

[8] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, "Medians and beyond: new aggregation techniques for sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems.* ACM, 2004, pp. 239–249.

[9] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Securely computing an approximate median in wireless sensor networks," in *Proceedings of the 4th international conference on Security and privacy in communication netowrks.* ACM, 2008, p. 6.

[10] D.-Y. Yang, A. Johar, A. Grama, and W. Szpankowski, "Summary structures for frequency queries on large transaction sets," in *Data Compression Conference, 2000. Proceedings. DCC 2000.* IEEE, 2000, pp. 420–429.

[11] K. Beyer, P. J. Haas, B. Reinwald, Y. Sismanis, and R. Gemulla, "On synopses for distinct-value estimation under multiset operations," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data.* ACM, 2007, pp. 199–210.

[12] E. K. Lee, H. Viswanathan, and D. Pompili, "Distributed data-centric adaptive sampling for cyber-physical systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 9, no. 4, p. 21, 2015.

[13] J. Bai, E. P. Eyisi, F. Qiu, Y. Xue, and X. D. Koutsoukos, "Optimal cross-layer design of sampling rate adaptation and network scheduling for wireless networked control systems," in *Proceedings of the IEEE/ACM Third International Conference on Cyber-Physical Systems.* IEEE Computer Society, 2012, pp. 107–116.

[14] H. H. Malik and J. R. Kender, "Optimizing frequency queries for data mining applications," in *Data Mining, ICDM. Seventh IEEE International Conference on.* IEEE, 2007, pp. 595–600.

[15] D.-Y. Yang, A. Johar, A. Grama, and W. Szpankowski, "Summary structures for frequency queries on large transaction sets," in *Data Compression Conference. Proceedings. DCC.* IEEE, 2000, pp. 420–429.

[16] 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.

[17] M. H. Cheung, R. Southwell, F. Hou, and J. Huang, "Distributed time-sensitive task selection in mobile crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing.* ACM, 2015, pp. 157–166.

[18] Y. Wu, Y. Wang, W. Hu, X. Zhang, and G. Cao, "Resource-aware photo crowdsourcing through disruption tolerant networks," in *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on.* IEEE, 2016, pp. 374–383.

[19] J. Jia, Q. Zhang, Q. Zhang, and M. Liu, "Revenue generation for truthful spectrum auction in dynamic spectrum access," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing.* ACM, 2009, pp. 3–12.

[20] W. Wang, B. Liang, and B. Li, "Designing truthful spectrum double auctions with local markets," *IEEE Transactions on Mobile Computing*, vol. 13, no. 1, pp. 75–88, 2014.

[21] M. Penrose, *Random geometric graphs.* Great Clarendon Street, Oxford OX2 6DP, United Kingdom: Oxford University Press Oxford, 2003, vol. 5.

[22] J. Díaz, D. Mitsche, and X. Pérez-Giménez, "Large connectivity for dynamic random geometric graphs," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 6, pp. 821–835, 2009.

[23] G. T. Pitsiladis, A. Krokos, A. D. Panagopoulos, and P. Constantinou, "Connectivity calculation in mobile ad hoc networks: Realistic performance simulation," in *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 3rd International Congress on.* IEEE, 2011, pp. 1–5.

[24] P. Santi and D. M. Blough, "An evaluation of connectivity in mobile wireless ad hoc networks," in *Dependable Systems and Networks, DSN, Proceedings. International Conference on.* IEEE, 2002, pp. 89–98.

[25] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004.

[26] I. F. Akyildiz, M. C. Vuran, and O. B. Akan, "On exploiting spatial and temporal correlation in wireless sensor networks," in *Proceedings of WiOpt*, vol. 4, 2004, pp. 71–80.

[27] Y. Wu, Y. Wang, and G. Cao, "Photo crowdsourcing for area coverage in resource constrained environments," in *IEEE INFOCOM*, 2017.

[28] Greenorbs. [Online]. Available: http://www.greenorbs.org/

[29] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *ACM SIGKDD*, 2011, pp. 1082–1090.

[30] J. Li, Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*, 2016.

[31] M. Han, J. Li, Z. Cai, and Q. Han, "Privacy reserved influence maximization in gps-enabled cyber-physical and online social networks," in *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on.* IEEE, 2016, pp. 284–292.

[32] J. W. Byers, M. Mitzenmacher, and G. Zervas, "Information asymmetries in pay-per-bid auctions," in *Proceedings of the 11th ACM conference on Electronic commerce.* ACM, 2010, pp. 1–12.

[33] R. B. Myerson, "Optimal auction design," *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] X. Zheng, Z. Cai, J. Li, and H. Gao, "A study on application-aware scheduling in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1787–1801, July 2017.

[40] X. Wang, L. Guo, C. Ai, J. Li, and Z. Cai, "An urban area-oriented traffic information query strategy in vanets," in *The 8th International Conference on Wireless Algorithms, Systems and Applications (WASA2013)*. Springer, 2013, pp. 313–324.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh, "Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals," *Data mining and knowledge discovery*, vol. 1, no. 1, pp. 29–53, 1997.

[47] K. Zhang, H. Gao, X. Han, Z. Cai, and J. Li, "Probabilistic skyline on incomplete data," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 427–436.

[48] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles in wireless sensor networks," in *The 8th Annual International Conference on Combinatorial Optimization and Applications (COCOA2014).* Springer, 2014, pp. 161–172.

[49] J. Li, S. Cheng, H. Gao, and Z. Cai, "Approximate physical world reconstruction algorithms in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3099–3110, 2014.

[50] S. Cheng, J. Li, and Z. Cai, "O (ε)-approximation to physical world by sensor networks," in *The 32rd Annual IEEE International Conference on Computer Communications (IEEE INFOCOM 2013).* IEEE, 2013, pp. 3084–3092.

[51] Z. He, Z. Cai, S. Cheng, and X. Wang, "Approximate aggregation for tracking quantiles and range countings in wireless sensor networks," *Theoretical Computer Science*, vol. 607, pp. 381–390, 2015.

[52] G. Cormode, M. Garofalakis, S. Muthukrishnan, and R. Rastogi, "Holistic aggregates in a networked world: Distributed tracking of approximate quantiles," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data.* ACM, 2005, pp. 25–36.

[53] K. Liu, L. Chen, M. Li, and Y. Liu, "Continuous answering holistic queries over sensor networks," in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on.* IEEE, 2008, pp. 1–11.

[54] S. Cheng and J. Li, "Sampling based (epsilon, delta)-approximate aggregation algorithm in sensor networks," in *Distributed Computing Systems, 2009. ICDCS'09. 29th IEEE International Conference on.* IEEE, 2009, pp. 273–280.

[55] S. Cheng, J. Li, Q. Ren, and L. Yu, "Bernoulli sampling based (ε, δ)-approximate aggregation in large-scale sensor networks," in *Proceedings of the 29th conference on Information communications.* IEEE Press, 2010, pp. 1181–1189.

[56] Z. Bar-Yossef, R. Kumar, and D. Sivakumar, "Sampling algorithms: lower bounds and applications," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing.* ACM, 2001, pp. 266–275.

[57] J. Rice, *Mathematical statistics and data analysis.* Nelson Education, 2006.

[58] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomized algorithms and probabilistic analysis.* Cambridge University Press, 2005.

[59] M. H. M. H. DeGroot *et al.*, *Probability and statistics*, 1986, no. 04; QA273, D4 1986.

[60] R. Lachowski, M. E. Pellenz, M. C. Penna, E. Jamhour, and R. D. Souza, "An efficient distributed algorithm for constructing spanning trees in wireless sensor networks," *Sensors*, vol. 15, no. 1, pp. 1518–1536, 2015.

[61] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance measurements of motes sensor networks," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems.* ACM, 2004, pp. 174–181.

[62] A. S. Tanenbaum, *Computer networks.* Prentice Hall New Jersey, 1996, vol. 4.

[63] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on.* IEEE, 2000, pp. 10–pp.

[64] E. Lee *et al.*, "Cyber physical systems: Design challenges," in *Object Oriented Real-Time Distributed Computing (ISORC), 11th IEEE International Symposium on.* IEEE, 2008, pp. 363–369.

[65] R. S. Mohan, R. Sachin, and U. Sakthivel, "Vehicular ad hoc network based pollution monitoring in urban areas," in *Computational Intelligence and Communication Networks (CICN), Fourth International Conference on.* IEEE, 2012, pp. 214–217.

[66] S. Fujiwara, T. Ohta, and Y. Kakuda, "An inter-domain routing for heterogeneous mobile ad hoc networks using packet conversion and address sharing," in *Distributed Computing Systems Workshops (ICDCSW), 32nd International Conference on.* IEEE, 2012, pp. 349–355.

[67] A. Chaudhary, V. N. Tiwari, and A. Kumar, "Design an anomaly based fuzzy intrusion

detection system for packet dropping attack in mobile ad hoc networks," in *Advance Computing Conference (IACC), IEEE International.* IEEE, 2014, pp. 256–261.

[68] T. Afroze, "Performance evaluation of the hostile environment in mobile ad-hoc network," in *Telecommunication Networks and Applications Conference (ATNAC), Australasian.* IEEE, 2012, pp. 1–8.

[69] J. T. B. Fajardo, K. Yasumoto, N. Shibata, W. Sun, and M. Ito, "Dtn-based data aggregation for timely information collection in disaster areas," in *Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE 8th International Conference on.* IEEE, 2012, pp. 333–340.

[70] 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.

[71] M. Ren, J. Li, L. Guo, and Z. Cai, "Data collection with probabilistic guarantees in opportunistic wireless networks," *International Journal of Sensor Networks*, vol. 24, no. 2, pp. 125–137, 2017.

[72] K. Zhang, Q. Han, Z. Cai, G. Yin, and J. Lin, "Doami: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks," *Theoretical Computer Science*, 2016.

[73] D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "Toward energy efficient big data gathering in densely distributed sensor networks," *Emerging Topics in Computing, IEEE Transactions on*, vol. 2, no. 3, pp. 388–397, 2014.

[74] Q. Chen, H. Gao, S. Cheng, J. Li, and Z. Cai, "Distributed non-structure based data aggregation for duty-cycle wireless sensor networks," in *The 36th Annual IEEE International Conference on Computer Communications (INFOCOM 2017)*, May 2017, pp. 1–9.

[75] Q. Chen, H. Gao, Z. Cai, L. Cheng, and J. Li, "Energy-collision aware data aggregation scheduling for energy harvesting sensor networks," in *The 37th Annual IEEE International Conference on Computer Communications (INFOCOM 2018)*, 2018.

[76] T. Shi, J. Li, H. Gao, and Z. Cai, "Coverage in battery-free wireless sensor networks," in *The 37th Annual IEEE International Conference on Computer Communications (INFOCOM 2018)*, 2018.

[77] 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.

[78] 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.

[79] 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.

[80] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking.* ACM, 1998, pp. 85–97.

[81] J. Devore, *Probability and Statistics for Engineering and the Sciences.* 20 Channel Center Street Boston, MA 02210, USA: Cengage Learning, 2015.

[82] M. Pascoe, J. Gomez, V. Rangel, and M. Lopez-Guerrero, "An upper bound on network size in mobile ad-hoc networks," in *Global Telecommunications Conference, IEEE GLOBECOM. IEEE.* IEEE, 2008, pp. 1–6.

[83] http://grid.cs.gsu.edu/$\sim$jli30/simulation/.

[84] G. Huang, X. Li, and J. He, "Dynamic minimal spanning tree routing protocol for large wireless sensor networks," in *Industrial Electronics and Applications, 2006 1ST IEEE Conference on.* IEEE, 2006, pp. 1–5.

[85] 2013. [Online]. Available: https://www.gov.uk/government/collections/road-traffic-statistics

[86] J. Li and J. Li, "Data sampling control, compression and query in sensor networks," *International Journal of Sensor Networks*, vol. 2, no. 1-2, pp. 53–61, 2007.

[87] 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.

[88] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[89] T. Shi, S. Cheng, Z. Cai, Y. Li, and J. Li, "Retrieving the maximal time-bounded positive influence set from social networks," *Personal and Ubiquitous Computing*, vol. 20, no. 5, pp. 717–730, 2016.

[90] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and Q. Han, "A game theory-based trust measurement model for social networks," *Computational Social Networks*, vol. 3, no. 1, p. 2, 2016.

[91] Z. He, Z. Cai, and J. Yu, "Latent-data privacy preserving with customized data utility for social network data," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 665–673, 2018.

[92] Y. Huang, Z. Cai, and A. G. Bourgeois, "Location privacy protection with accurate service," *Journal of Network and Computer Applications*, vol. 103, p. 146C156, 2018.

[93] Z. He, Z. Cai, J. Yu, X. Wang, Y. Sun, and Y. Li, "Cost-efficient strategies for restraining rumor spreading in mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2789–2800, March 2017.

[94] X. Wang, Y. Lin, Y. Zhao, L. Zhang, J. Liang, and Z. Cai, "A novel approach for inhibiting misinformation propagation in human mobile opportunistic networks," *Peer-to-Peer Networking and Applications*, vol. 10, no. 2, pp. 377–394, 2017.

[95] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 1, pp. 402–427, 2013.

[96] Z. He, Z. Cai, Y. Sun, Y. Li, and X. Cheng, "Customized privacy preserving for inherent data and latent data," *Personal and Ubiquitous Computing*, vol. 21, no. 1, pp. 43–54, 2017.

[97] X. Wang, Y. Lin, S. Zhang, and Z. Cai, "A social activity and physical contact-based routing algorithm in mobile opportunistic networks for emergency response to sudden disasters," *Enterprise Information Systems*, vol. 11, no. 5, pp. 597–626, 2017.

[98] M. Han, M. Yan, Z. Cai, Y. Li, X. Cai, and J. Yu, "Influence maximization by probing partial communities in dynamic online social networks," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, 2017.

[99] Z. He, Z. Cai, Q. Han, W. Tong, L. Sun, and Y. Li, "An energy efficient privacy-preserving content sharing scheme in mobile social networks," *Personal and Ubiquitous Computing*, vol. 20, no. 5, pp. 833–846, 2016.

[100] Z. He, Z. Cai, and X. Wang, "Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks," in *The 35th IEEE International Conference on Distributed Computing Systems (ICDCS 2015)*, June 2015, pp. 205–214.

[101] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th annual international conference on Mobile computing and networking.* ACM, 2012, pp. 173–184.

[102] J. Wang, J. Tang, D. Yang, E. Wang, and G. Xue, "Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing," in *Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on.* IEEE, 2016, pp. 354–363.

[103] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proceedings of the 10th international conference on Mobile systems, applications, and services.* ACM, 2012, pp. 337–350.

[104] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.

[105] J. M. Robson, "Algorithms for maximum independent sets," *Journal of Algorithms*, vol. 7, no. 3, pp. 425–440, 1986.

[106] T. Nieberg, J. Hurink, and W. Kern, "A robust ptas for maximum weight independent sets in unit disk graphs," in *Graph-theoretic concepts in computer science.* Springer, 2004, pp. 214–221.

[107] T. H. Cormen, *Introduction to algorithms.* MIT press, 2009.

[108] T. Sandholm, "Algorithm for optimal winner determination in combinatorial auctions," *Artificial intelligence*, vol. 135, no. 1, pp. 1–54, 2002.

[109] N. Nisan and A. Ronen, "Computationally feasible vcg mechanisms." *J. Artif. Intell. Res.(JAIR)*, vol. 29, pp. 19–47, 2007.

[110] Y. Singer, "Budget feasible mechanisms," in *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on.* IEEE, 2010, pp. 765–774.

[111] J. Leskovec and A. Krevl, “Snap datasets: Stanford large network dataset collection,” http://snap.stanford.edu/data, Jun. 2014.

[112] D. B. Johnson and D. A. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *Mobile computing*. Springer, 1996, pp. 153–181.

[113] [Online]. Available: http://people.bu.edu/zg/swoopo.html

[114] P. Papadimitriou and H. Garcia-Molina, “Sponsored search auctions with conflict constraints,” in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 283–292.

[115] Q. Chen, S. Cheng, H. Gao, J. Li, and Z. Cai, “Energy-efficient algorithm for multicasting in duty-cycled sensor networks,” *Sensors*, vol. 15, no. 12, pp. 31 224–31 243, 2015.

[116] L. Zhang, Z. Cai, J. Lu, and X. Wang, “Mobility-aware routing in delay tolerant networks,” *Personal and Ubiquitous Computing*, vol. 19, no. 7, pp. 1111–1123, 2015.

[117] X. Guan, A. Li, Z. Cai, and T. Ohtsuki, “Coalition graph game for robust routing in cooperative cognitive radio networks,” *Mobile Networks and Applications*, vol. 20, no. 2, pp. 147–156, 2015.

[118] S. Cheng, Z. Cai, and J. Li, “Curve query processing in wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5198–5209, 2015.

[119] Q. Chen, H. Gao, S. Cheng, and Z. Cai, “Approximate scheduling and constructing algorithms for minimum-energy multicasting in duty-cycled sensor networks,” in *International Conference on Identification, Information and Knowledge in the Internet of Things 2015 (IIKI 2015)*. IEEE, 2015, pp. 163–168.

[120] T. Shi, J. Wan, S. Cheng, Z. Cai, Y. Li, and J. Li, “Time-bounded positive influence in social networks,” in *International Conference on Identification, Information and Knowledge in the Internet of Things 2015 (IIKI 2015)*. IEEE, 2015, pp. 134–139.

[121] K. Zhang, Q. Han, Z. Cai, G. Yin, and J. Lin, "Metric and distributed on-line algorithm for minimizing routing interference in wireless sensor networks," in *The 9th Annual International Conference on Combinatorial Optimization and Applications (COCOA 2015).* Springer, 2015, pp. 279–292.

[122] T. Zhu, X. Wang, S. Cheng, Z. Cai, and J. Li, "Critical point aware data acquisition algorithm in sensor networks," in *The 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2015).* Springer, 2015, pp. 798–808.

[123] J. Li, S. Cheng, Z. Cai, Q. Han, and H. Gao, "Bernoulli sampling based (epsilon, delta)-approximate frequency query in mobile ad hoc networks," in *The 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2015).* Springer, 2015, pp. 315–324.

[124] S. Cheng, Z. Cai, J. Li, and X. Fang, "Drawing dominant dataset from big sensory data in wireless sensor networks," in *The 34th Annual IEEE International Conference on Computer Communications (INFOCOM 2015)*, April 2015, pp. 531–539.

[125] J. Gao, J. Li, Z. Cai, and H. Gao, "Composite event coverage in wireless sensor networks with heterogeneous sensors," in *The 34th Annual IEEE International Conference on Computer Communications (INFOCOM 2015)*, April 2015, pp. 217–225.

[126] Y. Huang, M. Chen, Z. Cai, X. Guan, T. Ohtsuki, and Y. Zhang, "Graph theory based capacity analysis for vehicular ad hoc networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–5.

[127] X. Guan, Y. Huang, Z. Cai, and T. Ohtsuki, "Intersection-based forwarding protocol for vehicular ad hoc networks," *Telecommunication Systems*, vol. 62, no. 1, pp. 67–76, 2016.

[128] T. Zhu, S. Cheng, Z. Cai, and J. Li, "Critical data points retrieving method for big sensory data in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 18, 2016.

[129] L. Guo, Y. Li, and Z. Cai, "Minimum-latency aggregation scheduling in wireless sensor network," *Journal of Combinatorial Optimization*, vol. 31, no. 1, pp. 279–310, 2016.

[130] J. Lu, Z. Cai, X. Wang, L. Zhang, P. Li, and Z. He, "Primary and secondary social activity aware routing for cognitive radio networks." in *The International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI 2016)*, 2016.

[131] M. Yan, M. Han, C. Ai, Z. Cai, and Y. Li, "Data aggregation scheduling in probabilistic wireless networks with cognitive radio capability (globecom 2016)," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[132] L. Zhang, Z. Cai, P. Li, and X. Wang, "Exploiting spectrum availability and quality in routing for multi-hop cognitive radio networks," in *International Conference on Wireless Algorithms, Systems, and Applications (WASA 2016)*. Springer, 2016, pp. 283–294.

[133] Y. Wang, "Topology control for wireless sensor networks," in *Wireless sensor networks and applications*. Springer, 2008, pp. 113–147.

[134] J. Elson and D. Estrin, *Time synchronization for wireless sensor networks*. IEEE, 2001.

[135] T. Shi, S. Cheng, Z. Cai, and J. Li, "Adaptive connected dominating set discovering algorithm in energy-harvest sensor networks," in *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*, April 2016, pp. 1–9.

[136] L. Zhang, Z. Cai, P. Li, L. Wang, and X. Wang, "Spectrum-availability based routing for cognitive sensor networks," *IEEE Access*, vol. 5, pp. 4448–4457, 2017.

[137] K. Zhang, Q. Han, Z. Cai, and G. Yin, "Rippas: a ring-based privacy-preserving aggregation scheme in wireless sensor networks," *Sensors*, vol. 17, no. 2, p. 300, 2017.

[138] L. Zhang, X. Wang, J. Lu, M. Ren, Z. Duan, and Z. Cai, "A novel contact prediction-based routing scheme for dtns," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 1, 2017.

[139] S. Cheng, Z. Cai, J. Li, and H. Gao, "Extracting kernel dataset from big sensory data in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 813–827, April 2017.

[140] T. Shi, S. Cheng, J. Li, and Z. Cai, "Constructing connected dominating sets in battery-free networks," in *The 36th Annual IEEE International Conference on Computer Communications (INFOCOM 2017)*, May 2017, pp. 1–9.

[141] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases," pp. 449–460, 2004.

[142] G. Hartl and B. Li, "infer: A bayesian inference approach towards energy efficient data collection in dense sensor networks," pp. 371–380, 2005.

[143] J. Wang, Z. Cai, C. Ai, D. Yang, H. Gao, , and X. Cheng, "Differentially private k-anonymity: achieving query privacy in location-based services," in *The International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI 2016)*, 2016.

[144] J. Li, Z. Cai, M. Yan, and Y. Li, "Using crowdsourced data in location-based social networks to explore influence maximization," in *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*, April 2016, pp. 1–9.

[145] X. Zheng, Z. Cai, J. Li, and H. Gao, "Location-privacy-aware review publication mechanism for local business service systems," in *The 36th Annual IEEE International Conference on Computer Communications (INFOCOM 2017)*, May 2017, pp. 1–9.

[146] 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)*, June 2017, pp. 635–644.

[147] L. Zhang, X. Wang, J. Lu, P. Li, and Z. Cai, "An efficient privacy preserving data aggregation approach for mobile sensing," *Security and Communication Networks*, vol. 9, no. 16, pp. 3844–3853, 2016.

[148] Y. Liang, Z. Cai, Q. Han, and Y. Li, "Location privacy leakage through sensory data," *Security and Communication Networks*, vol. 2017, 2017.

[149] 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.

[150] 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.

[151] 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.

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

[153] 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.

[154] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th international conference on Mobile systems, applications, and services.* ACM, 2009, pp. 55–68.

[155] 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.

[156] 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.

[157] L. Zhang, Z. Cai, and X. Wang, "Fakemask: a novel privacy preserving approach for smartphones," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 335–348, 2016.

[158] E. Álvarez-Miranda, I. Ljubić, and P. Mutzel, "The maximum weight connected subgraph problem," in *Facets of Combinatorial Optimization*. Springer, 2013, pp. 245–270.

[159] D. S. Johnson, "The np-completeness column: an ongoing guide," *Journal of Algorithms*, vol. 6, no. 3, pp. 434–451, 1985.

[160] [Online]. Available: http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml