

Syracuse University

**SURFACE**

---

Dissertations - ALL

SURFACE

---

May 2018

## MODELING AND RESOURCE ALLOCATION IN MOBILE WIRELESS NETWORKS

Jing Wang  
*Syracuse University*

Follow this and additional works at: <https://surface.syr.edu/etd>



Part of the [Engineering Commons](#)

---

### Recommended Citation

Wang, Jing, "MODELING AND RESOURCE ALLOCATION IN MOBILE WIRELESS NETWORKS" (2018).

*Dissertations - ALL*. 881.

<https://surface.syr.edu/etd/881>

This Dissertation is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact [surface@syr.edu](mailto:surface@syr.edu).

## ABSTRACT

We envision that in the near future, just as Infrastructure-as-a-Service (IaaS), radios and radio resources in a wireless network can also be provisioned as a service to Mobile Virtual Network Operators (MVNOs), which we refer to as Radio-as-a-Service (RaaS). In this thesis, we present a novel auction-based model to enable fair pricing and fair resource allocation according to real-time needs of MVNOs for RaaS. Based on the proposed model, we study the auction mechanism design with the objective of maximizing social welfare. We present an Integer Linear Programming (ILP) and Vickrey-Clarke-Groves (VCG) based auction mechanism for obtaining optimal social welfare. To reduce time complexity, we present a polynomial-time greedy mechanism for the RaaS auction. Both methods have been formally shown to be truthful and individually rational.

Meanwhile, wireless networks have become more and more advanced and complicated, which are generating a large amount of runtime system statistics. In this thesis, we also propose to leverage the emerging deep learning techniques for spatiotemporal modeling and prediction in cellular networks, based on big system data. We present a hybrid deep learning model for spatiotemporal prediction, which includes a novel autoencoder-based deep model for spatial modeling and Long Short-Term Memory units (LSTMs) for temporal modeling. The autoencoder-based model consists of a Global Stacked AutoEncoder (GSAE) and multiple Local SAEs (LSAEs), which can offer good representations for input data, reduced model size, and support for parallel and application-aware training.

Mobile wireless networks have become an essential part in wireless networking with the prevalence of mobile device usage. Most mobile devices have powerful sensing capabilities. We consider a general-purpose Mobile CrowdSensing(MCS) system, which is a multi-application multi-task system that supports a large variety of sensing applications.

In this thesis, we also study the quality of the recruited crowd for MCS, i.e., quality

of services/data each individual mobile user and the whole crowd are potentially capable of providing. Moreover, to improve flexibility and effectiveness, we consider fine-grained MCS, in which each sensing task is divided into multiple subtasks and a mobile user may make contributions to multiple subtasks. More specifically, we first introduce mathematical models for characterizing the quality of a recruited crowd for different sensing applications. Based on these models, we present a novel auction formulation for quality-aware and fine-grained MCS, which minimizes the expected expenditure subject to the quality requirement of each subtask. Then we discuss how to achieve the optimal expected expenditure, and present a practical incentive mechanism to solve the auction problem, which is shown to have the desirable properties of truthfulness, individual rationality and computational efficiency.

In a MCS system, a sensing task is dispatched to many smartphones for data collections; in the meanwhile, a smartphone undertakes many different sensing tasks that demand data from various sensors. In this thesis, we also consider the problem of scheduling different sensing tasks assigned to a smartphone with the objective of minimizing sensing energy consumption while ensuring Quality of SenSing (QoSS). First, we consider a simple case in which each sensing task only requests data from a single sensor. We formally define the corresponding problem as the Minimum Energy Single-sensor task Scheduling (MESS) problem and present a polynomial-time optimal algorithm to solve it. Furthermore, we address a more general case in which some sensing tasks request multiple sensors to report their measurements simultaneously. We present an Integer Linear Programming (ILP) formulation as well as two effective polynomial-time heuristic algorithms, for the corresponding Minimum Energy Multi-sensor task Scheduling (MEMS) problem.

Numerical results are presented to confirm the theoretical analysis of our schemes, and to show strong performances of our solutions, compared to several baseline methods.

MODELING AND RESOURCE ALLOCATION IN  
MOBILE WIRELESS NETWORKS

By

Jing Wang

B.Eng. Beihang University, 2011

M.S. Syracuse University, 2017

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of  
Doctor of Philosophy in Electrical and Computer Engineering

Syracuse University  
May 2018

Copyright © 2018 Jing Wang

All rights reserved

# ACKNOWLEDGMENTS

I would not be here to defend my dissertation without the guidance of my committee members, help from friends and support from my family.

My deepest gratitude goes to my advisor, Dr. Jian Tang. You have always been supportive for my research ideas. Your encouragements helped me greatly with my conference presentations and research since its infancy. You have set an example of excellence as a researcher, instructor and mentor. Dr. Tang, Dr. Dejun Yang from Colorado School of Mines and Dr. Guoliang Xue from Arizona State University have made my Ph.D research more productive and enjoyable with their continuous caring, professional guidance and profound knowledge. Thank you for all your mentorship.

I'd like to thank my thesis committee members for all of your guidance. Your discussions and feedbacks have been absolutely invaluable.

I also would like to thank all my dear friends from the same lab. I am very grateful for your company, encouragement and teamwork. My time at Syracuse was made enjoyable in large part because many friends became a part of my life.

Lastly, a very special thanks to my family. This journey would have been impossible without their immense support. I have relied, often without ever asking, on their help, love and encouragement. They supported me in all my pursuits. Without their patience and sacrifice, I could not have completed this thesis.

# TABLE OF CONTENTS

<b>Acknowledgments</b>	<b>v</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivations . . . . .	1
1.2 State of the Art and Literature Gap . . . . .	4
1.2.1 Radio-as-a-Service in Wireless Networks . . . . .	4
1.2.2 Modeling and Prediction in Wireless Networks . . . . .	6
1.2.3 Incentive Mechanisms in Mobile Crowdsensing Systems . . . . .	8
1.2.4 Task Scheduling in Mobile Crowdsensing Systems . . . . .	9
1.3 Contributions and Thesis Organization . . . . .	11
<b>2 Radio-as-a-Service in Wireless Networks: Modeling and Resource Allocation</b>	<b>13</b>
2.1 Overview . . . . .	13
2.2 Problem Formulation . . . . .	15
2.3 Auction Mechanism with Optimal Social Welfare . . . . .	19
2.3.1 Optimal RaaS Auction Design (Optimal-RaaS) . . . . .	19
2.3.2 Proof of Properties . . . . .	20
2.4 Greedy Auction Mechanism . . . . .	22
2.4.1 Greedy RaaS Auction Design (GRAD) . . . . .	22
2.4.2 Proof of Properties . . . . .	24
2.5 Auction Mechanisms with Reserve Prices . . . . .	25
2.5.1 RaaS Auction with Reserve Prices (RaaS-RP) . . . . .	26

2.5.2	Proof of Properties . . . . .	27
2.6	Performance Evaluation . . . . .	29
2.6.1	Performance Evaluation of Optimal-RaaS and GRAD . . . . .	29
2.6.2	Performance Evaluation of RaaS-RP . . . . .	31
2.7	Summary . . . . .	33
<b>3</b>	<b>Spatiotemporal Modeling and Prediction in Wireless Networks</b>	<b>34</b>
3.1	Overview . . . . .	34
3.2	Preliminary Data Analysis . . . . .	36
3.2.1	Dataset . . . . .	36
3.2.2	Data Analysis . . . . .	37
3.3	Spatiotemporal Modeling and Prediction . . . . .	40
3.3.1	Overview . . . . .	40
3.3.2	Spatial Modeling . . . . .	42
3.3.3	Temporal Modeling and Prediction . . . . .	47
3.4	Performance Evaluation . . . . .	49
3.4.1	Settings . . . . .	49
3.4.2	Prediction Results . . . . .	50
3.4.3	Spatial Modeling . . . . .	51
3.5	Summary . . . . .	54
<b>4</b>	<b>Quality-aware and Fine-grained Incentive Mechanisms in Mobile Crowdsensing Systems</b>	<b>55</b>
4.1	Overview . . . . .	55
4.2	Quality of Crowd (QoC) Models . . . . .	58
4.3	Auction Formulation . . . . .	61
4.3.1	Desirable Properties . . . . .	62
4.3.2	Virtual Cost . . . . .	64

4.4	Quality-aware Incentive Mechanisms (QIMs)	66
4.4.1	Optimal Solutions	67
4.4.2	Proof of QIM-Opt Properties	68
4.4.3	Computationally Efficient QIM	69
4.4.4	Proof of QIM-E Properties	71
4.5	Performance Evaluation	73
4.5.1	Baseline Methods	73
4.5.2	Simulation Settings	74
4.5.3	Simulation Results and Analysis	75
4.6	Summary	78
<b>5</b>	<b>Task Scheduling in Mobile Crowdsensing Systems: Models and Algorithms</b>	<b>79</b>
5.1	Overview	79
5.2	System Model	80
5.3	Single-Sensor Task Scheduling	85
5.4	Multi-Sensor Task Scheduling	89
5.4.1	ILP and LP Rounding based Algorithm	90
5.4.2	Greedy Algorithm	92
5.5	Performance Evaluation	95
5.6	Summary	100
<b>6</b>	<b>Conclusions</b>	<b>102</b>
	<b>References</b>	<b>104</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 Motivations

Virtualization, inspired by the success of application of Virtual Machines (VMs) in cloud computing, has been introduced to wireless networking recently [47], enabling support for multiple Mobile Virtual Network Operators (MVNOs) via isolated slices over a shared wireless substrate.

We envision that in the near future, just as Infrastructure-as-a-Service (IaaS), radios and radio resources in a wireless network can also be provisioned as a service to multiple MVNOs, which we refer to as *Radio-as-a-Service (RaaS)*. In an RaaS cloud, Base Stations (BSs) are operated by the cloud service provider, which can lease radio resources of BSs to MVNOs for profit. For an MVNO, similar to a tenant in an IaaS cloud, it pays the cloud service provider to use radio resources to serve its own users. Usually, multiple MVNOs share common radio resources in an RaaS cloud. For wide adoption of RaaS, on one hand, the cloud service provider needs to be able to collect a fair amount of payment from each MVNO for radio resources it leases; on the other hand, an MVNO needs to be able to obtain sufficient resources from the cloud service provider to well serve its users at a fair cost. In an IaaS cloud (such as Amazon EC2), resources are given to tenants in the format

of VM and storage space, which has guaranteed capabilities/capacities for computing and storage respectively. However, in an RaaS, bandwidth (i.e., transmission capability) of a wireless link is time-varying. An MVNO, which rents a certain amount of radio resources beforehand, may not have sufficient bandwidth for its users in certain periods of time. Hence, it is very important to study incentive mechanisms and fairness for allocating radio resources among MVNOs to support RaaS.

Meanwhile, wireless networks have become more and more advanced and complicated, which are generating a large amount of runtime system statistics (such as traffic load, resource usages, etc) every second. For example, In [20], Ding *et al.* showed the volume of spectrum state data could be in the order of zettabytes (ZBs,  $1 \text{ ZB} = 10^{21}$  Bytes) in a  $100 * 100 \text{ km}^2$  area, during one week, on a spectrum ranging from 0 to 5 GHz. We can call such data *big system data*.

Tremendous research efforts (e.g., [85, 86]) have been made to develop algorithms and protocols for wireless networks to utilize their resources efficiently and effectively. However, most of them aimed at optimizing resource allocation, assuming that some key factors (such as traffic load, spectrum usages, computing resource usages, etc) are given as input. Limited work has been done to model and predict the pattern of these key factors, which are highly time and location varying. Instead of treating big system data as an unwanted burden, we should leverage them as a great opportunity for better understanding user demands and system capabilities such that we can optimize resource allocation to better serve mobile users.

Mobile wireless networks have become an essential part in wireless networking with the prevalence of mobile device usage. However, some mobile users haven't realized that their smartphones have powerful sensing capabilities. Most smartphones are equipped with various embedded sensors, including microphone, camera, GPS, accelerometer, gyroscope, WiFi/3G/4G interfaces, etc. Moreover, booming wearable devices (such as Google Glass, Smart Watches, Fitbit, Sensordrone [75], etc.) can be connected to smartphones via net-

work interfaces, such as Bluetooth, to extend their sensing capabilities. Embedded sensors and wearable devices can enable applications and services in various domains, such as environmental monitoring, social networking, healthcare, transportation and safety.

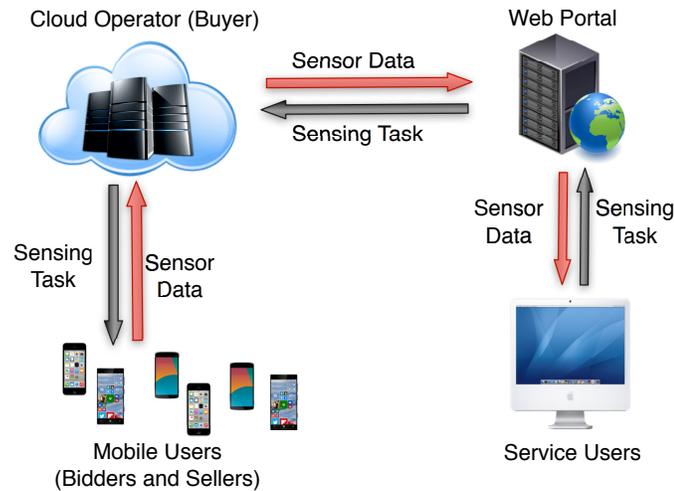


Fig. 1.1: An MCS System

Recently, *Mobile CrowdSensing*(MCS) have been gaining increasing popularity. As shown in Fig. 1.1, we consider a general-purpose MCS system [79], such as PRISM [19] and Medusa [68]. A service user can make a sensing service request via a web portal. The request is then analyzed by the cloud operator, which will use an incentive mechanism to recruit a sensing crowd (a set of mobile users) and distribute the request to them. Then their smartphones will perform the corresponding sensing activities and report sensor data to the cloud operator. The cloud operator will aggregate and analyze sensor data, and then send results back to the service user through the web portal.

The success of a crowdsourcing application highly depends on whether a quality crowd can be recruited to undertake the corresponding tasks. Recent research has been focused on incentive mechanisms [22, 97] for mobile crowdsourcing, which determine how to recruit a crowd mainly based on their prices/costs. However, limited research efforts have been made to quantify the quality of the recruited crowd, i.e., quality of services/data each individual mobile user and the whole crowd are potentially capable of providing, which is

a very meaningful topic to study.

Moreover, it can be energy-consuming to collect data from smartphone sensors. If sensor activities are not carefully managed in an energy-efficient manner, the energy-limited battery of a smartphone may be drained quickly. Specifically, it consumes energy to actively perform a scan to obtain a measurement for some sensors (such as WiFi interface); meanwhile, it also consumes energy to spawn a thread to obtain readings for some other sensors (such as accelerometer), even if they are always working. Moreover, it has been shown by [55, 91, 106] that some sensors, such as GPS, are power hungry.

To minimize energy consumption, several related works [55, 69, 91, 106] presented application-specific algorithms that determine how to control sampling and energy usages of particular sensors (such as GPS) for particular applications. In addition, a few recent works studied how to engage and select a proper subset of smartphones to participate in sensing activities [22, 97, 105], and how to generate and assign sensing tasks to participating smartphones [58, 78, 93] in a general MCS system. However, scant attention has been paid to the problem of scheduling sensing tasks on a smartphone with the objective of minimizing sensing energy consumption while guaranteeing Quality of SenSing (QoSS), which is an important topic to explore.

## **1.2 State of the Art and Literature Gap**

### **1.2.1 Radio-as-a-Service in Wireless Networks**

Cloud-based wireless networking and wireless virtualization have been studied recently. In [14], the framework CloudIQ was proposed to partition BSs into groups that are simultaneously processed on a shared homogeneous compute platform, and to schedule BSs to meet real-time processing requirements. A similar cloud-based wireless system, FluidNet, was introduced in [83]. In [33], Cudipati *et al.* introduced SoftRAN, a software defined centralized control plane for RANs that abstracts all BSs in a local geographical

area as a virtual big BS. In [47], Kokku *et al.* described the design and implementation of a Network Virtualization Substrate (NVS) for effective virtualization of wireless resources in cellular networks. In [62], the authors proposed AMPHIBIA, which enables end-to-end slicing over wired and wireless networks and exploits the advantages of virtualization and cognitive radio technology. In [111], Zhu *et al.* introduced the first TDD WiMAX-based SDR implementation on a commodity server, in conjunction with a novel design of a remote radio head. In [54], the authors presented a software defined cellular network architecture that supports flexible slicing of network resource. Unlike the cloud-based RANs introduced in [14, 83], SoftRAN [33] and the proposed CogCloud employs centralized control but still processes wireless signals at BSs (rather than in a data center) in a distributed manner. However, it does not support virtualization or multiple MVNOs, and moreover, the corresponding paper [33] did not present any resource allocation algorithms to enable the proposed architecture. More wireless virtualization works can be found in [13, 101, 94, 100, 41, 54, 111]. All these related works studied how to enable virtualization in specific wireless networks. We, however, present a general auction-based model and mechanisms for resource sharing among MVNOs in a wireless network, assuming virtualization is enabled on BSs. Most related works (except [59, 80]) on virtualization were focused on a single BS. We, however, aim to support RaaS for MVNOs over a network with multiple BSs.

The auction theory has been studied for decades. Vickery (1961) [89] proposed the notion of truthful bidding in a sealed-bid auction, and introduced the second-price auctions. Clarke and Groves extended his work, yielding the famous Vickery-Clarke-Groves (VCG) mechanism [65]. It has been proved in [65] that every VCG mechanism is truthful (incentive compatible). In the meanwhile, to mitigate high time complexity of VCG, some works were focused on proposing fast greedy heuristic algorithms without sacrificing truthfulness [8, 98].

Recently, efforts have been made to apply the auction theory to support network vir-

tualization. In [28], Gandhi *et al.* proposed a real-time spectrum auction framework to distribute spectrum among a large number wireless users under interference constraints. Their approach achieves conflict-free spectrum allocations that maximize auction revenue and spectrum utilization. Sengupta *et al.* [74] presented a winner determining sealed-bid knapsack auction mechanism that dynamically allocates spectrum to the wireless service providers based on their bids. The proposed dynamic pricing strategy is based on game theory to capture the conflict of interest between wireless service providers and end users, both of whom try to maximize their respective net utilities. In [110], based on a non-cooperative game model, Zhou *et al.* presented a bandwidth allocation scheme with Nash Equilibrium for a virtualized network environment. However, these works have not considered truthfulness, which is one of the major design goals of our work.

In [27], the interactions among Service Providers (SP) and Network Provider (NP) were modeled as a stochastic game; each stage of the game is played by SPs (on behalf of end users) and is regulated by the NP through a VCG mechanism. In [108], a truthful and computationally efficient spectrum auction named VERITAS was proposed to support eBay-like dynamic spectrum market with the objective of maximizing spectrum utilization. In [109], a general framework for truthful double spectrum auctions named TRUST is proposed. TRUST takes as input any reusability-driven spectrum allocation algorithm, and applies a winner determination and pricing mechanism to achieve truthfulness and other economic properties while improving spectrum utilization. Our work represents the first work to study the auction design for RaaS, which is mathematically different from the problems considered in these related works.

### **1.2.2 Modeling and Prediction in Wireless Networks**

Research efforts have been made for modeling and prediction in communication networks. Specifically, time series analysis methods have also been applied for predicting traffic load. In [81], Shu *et al.* showed that seasonal ARIMA models could be used to model and predict

wireless traffic. In [107], Zhou *et al.* proposed a network traffic prediction model, which is a combination of linear time series ARIMA model and non-linear GARCH model. Hong *et al.* applied SVR for short-term traffic load forecasting, and proposed a simulated annealing algorithm and a genetic algorithm to optimize the selection of SVR parameters in [40]. Spatial modeling and estimation methods have been proposed for traffic load in wireless networks [9, 88]. To predict the self-similar network traffic with high burstiness, the authors of [95] proposed a new hybrid method based on the combination of the covariance orthogonal prediction and the artificial neural network. A spatiotemporal compressive sensing framework was proposed for modeling Internet traffic matrices in [104]. Moreover, a very recent work [77] was focused on spatiotemporal analysis for application usages in wireless networks.

In addition, Akbar *et al.* proposed to model and predict the spectrum occupancy of licensed radio bands with Hidden Markov Models (HMMs) [7]. They introduced a Markov-based channel prediction algorithm for dynamic spectrum allocation in cognitive radio networks, when the channel state occupancy of primary are assumed to be Poisson distributed. In [87], Tumulus *et al.* designed two adaptive channel status predictor using a neural network based on multilayer perceptron and the hidden Markov model. *A priori* knowledge of the statistics of channel usage is not required in the prediction schemes. Chen *et al.* presented a detailed study [17] with first and second order statistics of collected data, including channel occupancy/vacancy, channel utilization and temporal, spectral and spatial correlation. A 2-dimensional frequent pattern mining algorithm was developed to predict channel availability based on past observations.

Unlike these works, we are the first to propose a deep learning model for spatiotemporal prediction in cellular networks.

### 1.2.3 Incentive Mechanisms in Mobile Crowdsensing Systems

There have been quite a few mobile phone/crowdsensing projects in different domains [51]. Recently, efforts have been made to develop general-purpose systems to support various mobile phone/crowdsensing applications, including PRISM [19], Medusa [68], the bubble sensing system [56], AnonySense [18] and Micro-blog [29].

Incentive mechanism design has been addressed in the context general MCS systems recently. Yang *et al.* introduced two models for MCS: platform-centric and user-centric; and designed an incentive mechanism using a Stackelberg game for the platform-centric model as well as an auction-based incentive mechanism for the user-centric model in [97]. Duan *et al.* proposed a reward-based collaboration mechanism in [22], in which collaborators share a total reward announced by the client. In addition, they investigated how the client can design an optimal contract by specifying different task-reward combinations for different user types. In [105], Zhao *et al.* considered the scenario where mobile users arrive one by one online in a random order. They presented two online incentive mechanisms, in which mobile users submit their private types to the crowdsourcer when arrive and the crowdsourcer aims to select a user subset for maximizing a utility function with a budget constraint. Feng *et al.* presented a reverse auction framework named TRAC in [24] to model location based auction interactions between a cloud and smartphones, which minimizes the social cost. In [25], the authors presented two truthful incentive mechanisms for both the offline and online cases, given dynamic smartphones, uncertain arrivals of tasks, strategic behaviors and private information of smartphones. In [103], the authors first designed an incentive mechanism, EFF, which eliminates dishonest behavior with the help from a trusted third party for arbitration. They then designed another mechanism DFF, which, without the help from any third party, discourages free-riding and false-reporting.

Recently, several research works have addressed incentive mechanism design with quality considerations. In [49], Koutsopoulos *et al.* sought a mechanism for user participation level determination and payment allocation which minimizes the total cost of compensat-

ing participants, while delivering a certain quality of experience to service requesters. They designed a mechanism that optimally solves this problem. In [36], the authors presented an approximation mechanism to find an efficient task allocation with quality of sensing requirements as well as a pricing mechanism based on bargaining theory. Luo *et al.* designed an incentive mechanism [57] based on all-pay auctions to attract contributions from mobile users. In [43], Jin *et al.* designed a truthful, individually rational and computationally efficient mechanism that approximately maximizes the social welfare for single-minded combinatorial models, which was shown to have an approximation ratio, assuming a linear quality model. Moreover, they designed an iterative descending mechanism with individual rationality for multi-minded combinatorial models.

We summarize the differences between our work and these related works in the following: 1) Unlike most related works, we consider fine-grained MCS, in which a sensing task consists of multiple subtasks and a mobile user may make contributions to multiple subtasks. 2) Many related works, such as [22, 24, 25, 97, 105, 103], have not offered careful consideration for QoC and quality requirements of subtasks, which, however, is one of the main topics in this thesis. 4) The auction formulation here (with the objective of minimizing the expected expenditure subject to quality requirements) is mathematically different from those in related works [36, 43, 49, 57]. 5) Unlike some previous works mainly focusing on a specific quality model [36, 43] (such as the linear model), we conduct a comprehensive study for QoC models.

#### **1.2.4 Task Scheduling in Mobile Crowdsensing Systems**

Sensing task scheduling and optimization have been addressed in the context of both general and application-specific mobile phone/crowdsensing. In [91], the authors presented an Energy Efficient Mobile Sensing System (EEMSS), which uses hierarchical sensor management strategy to recognize user states as well as to detect state transitions. They proposed to power only a minimum set of sensors and use appropriate sensor duty cycles

for energy savings. The paper [106] presented an adaptive location-sensing framework that significantly improves the energy efficiency of smartphones running location-based applications. The underlying design principles of the proposed framework involve substitution, suppression, piggybacking, and adaptation of application location-sensing requests to conserve energy. In [55], Lin *et al.* studied energy-accuracy trade-off for continuous mobile device location, and designed and prototyped an adaptive location service for mobile devices, a-Loc, which helps reduce this battery drain. In [69], Rachuri *et al.* proposed SociableSense, a smart phone based platform that captures user behaviors in office environments, while providing the users with a quantitative measure of their sociability and that of colleagues. The system provides an adaptive sampling mechanism as well as models to decide whether to perform computation of tasks, such as the execution of classification and inference algorithms, locally or remotely. In [58], the authors presented analytical results on the rate of information reporting by uncontrolled mobile sensors needed to cover a given geographical area. In [93], the authors introduced mechanisms for automated mapping of urban areas, which provide a virtual sensor abstraction to applications. They also proposed spatial and temporal coverage metrics for measuring the quality of acquired data. In [78], Sheng *et al.* presented algorithms for energy-efficient sensing scheduling and showed that significant power savings can be achieved by collaborative sensing via simulations.

We summarize the differences between our work and these related works in the following: 1) Unlike research works targeting at energy-efficient sensing scheduling or optimization for specific applications, such as [55, 69, 91, 106], we aim to address task scheduling in general MCS systems. 2) Most related works on general MCS, such as [58, 78, 93], essentially studied the problem of determining how to assign tasks to a group of participating smartphones. We, however, consider the problem of scheduling different sensing tasks assigned to a smartphone, which is mathematically different from their problems. 3) Task scheduling problems on a smartphone are also mathematically different from incentive mechanism design [22, 97, 105].

## 1.3 Contributions and Thesis Organization

In this thesis, we study the modeling and resource allocation in mobile wireless networks. Specifically, we make the following contributions.

- we present a novel auction-based model to enable fair pricing and fair resource allocation according to real-time needs of MVNOs for RaaS. Based on the proposed model, we study the auction mechanism design with the objective of maximizing social welfare. We present an Integer Linear Programming (ILP) and Vickrey-Clarke-Groves (VCG) based auction mechanism for obtaining optimal social welfare. To reduce time complexity, we present a polynomial-time greedy mechanism for the RaaS auction. Both methods have been formally shown to be truthful and individually rational. Two papers ([1] and [2]) have been published for this work.
- We propose to leverage the emerging deep learning techniques for spatiotemporal modeling and prediction in cellular networks, based on big system data. A hybrid deep learning model is presented for spatiotemporal prediction, which includes a novel autoencoder-based deep model for spatial modeling and Long Short-Term Memory units (LSTMs) for temporal modeling. The autoencoder-based model consists of a Global Stacked AutoEncoder (GSAE) and multiple Local SAEs (LSAEs), which can offer good representations for input data, reduced model size, and support for parallel and application-aware training. Moreover, we present a new algorithm for training the proposed spatial model. One paper ([3]) has been published for this work.
- We introduce mathematical models for characterizing the quality of a recruited crowd for different sensing applications. Based on these models, we present a novel auction formulation for quality-aware and fine-grained MCS, which minimizes the expected expenditure subject to the quality requirement of each subtask. Then we discuss how

to achieve the optimal expected expenditure, and present a practical incentive mechanism to solve the auction problem, which is shown to have the desirable properties of truthfulness, individual rationality and computational efficiency. One paper ([4]) has been published for this work.

- We consider the problem of scheduling different sensing tasks assigned to a smartphone with the objective of minimizing sensing energy consumption while ensuring Quality of SenSing (QoSS). First, we consider a simple case in which each sensing task only requests data from a single sensor. We formally define the corresponding problem as the Minimum Energy Single-sensor task Scheduling (MESS) problem and present a polynomial-time optimal algorithm to solve it. Furthermore, we address a more general case in which some sensing tasks request multiple sensors to report their measurements simultaneously. We present an Integer Linear Programming (ILP) formulation as well as two effective polynomial-time heuristic algorithms, for the corresponding Minimum Energy Multi-sensor task Scheduling (MEMS) problem. Two papers ([5] and [6]) have been published for this work.

The rest of the thesis is organized as follows. We study modeling and resource allocation of Radio-as-a-Service in Chapter 2. In Chapter 3, we investigate the emerging deep learning techniques for spatiotemporal modeling and prediction based on cellular network big system data. We design quality-aware and fine-grained incentive mechanisms for MCS in Chapter 4. In Chapter 5, we investigate task scheduling models and algorithms in MCS. The conclusions of this thesis is presented in Chapter 6.

# CHAPTER 2

## RADIO-AS-A-SERVICE IN WIRELESS NETWORKS: MODELING AND RESOURCE ALLOCATION

### 2.1 Overview

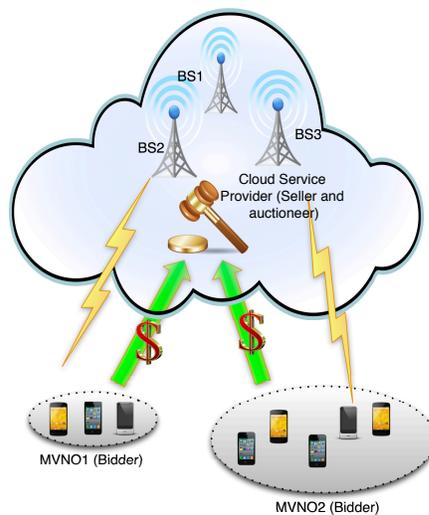


Fig. 2.1: The RaaS auction

In this chapter, we introduce a novel auction-based model to achieve the goal of en-

abling fair pricing and reasonable resource allocation for RaaS. In our model as illustrated in Fig. 2.1, cloud service provider (i.e., seller) sells its radio resources to MVNOs. MVNOs (i.e., bidders or buyers) participate in the auction, bid the resources according to their real-time needs and make payment to the cloud service provider. Moreover, the cloud service provider plays the role of auctioneer so that it will determine the winners among MVNOs and clear prices MVNOs should pay.

To support RaaS, we allow each MVNO to bid for a combination of demanded resources on each BS. This auction can be related to a *combinatorial auction* [65], with the difference lying in the fact that the demanded resources could be only a fraction of the available resources on each BS. So its resources are actually shared among multiple MVNOs. While in a conventional combinatorial auction, a bidder expresses its valuation of a combination of items, and the auction does not allow item sharing.

Auction mechanism design is crucial for supporting RaaS, because it directly determines the trading rules between the seller (cloud service provider) and bidders (MVNOs); furthermore, it implicitly defines the behaviors of bidders. Specifically, *truthfulness* (a.k.a. incentive capability or strategy-proofness) [65] and *individual rationality* [50] are highly desirable in RaaS auction mechanisms. An auction mechanism is truthful if a bidder will not increase its payoff by making any other bid instead of the true value. Revealing the true private value is every participating bidder's dominant strategy no matter what strategies other bidders are doing [96]. An auction lacking truthfulness could be vulnerable to market manipulation and produce very poor outcomes [46]. In addition, an auction mechanism is individually rational if the payoff of every bidder is non-negative.

To the best of our knowledge, we are the first to develop an auction-based model and auction mechanisms with provably-good properties for RaaS. We summarize our contributions of this chapter in the following:

- We formally define the RaaS auction mechanism design problem with the objective of maximizing social welfare.

- We present an Integer Linear Programming (ILP) and Vickrey-Clarke-Groves (VCG) based auction mechanism for obtaining optimal social welfare. To reduce time complexity, we present a polynomial-time greedy auction mechanism. Moreover, we show that the proposed mechanisms are both truthful and individually rational.
- To prevent winning bidders from making 0 payment, we introduce reserve prices and present auction mechanisms with reserve prices, which are shown to be both truthful and individually rational too.
- We present extensive simulation results to show the proposed greedy mechanism achieves significant running time savings and produces close-to-optimal solutions. Moreover, we justify effectiveness of the proposed auction mechanisms with reserve prices via simulation results.

## 2.2 Problem Formulation

First of all, we summarize major notations in Table 2.1

Table 2.1: Major Notations

Notation	Explanation
$i, N$ and $\mathbf{I}$	The index of BSs, the total number of BSs and the set of BSs
$j, M$ and $\mathbf{J}$	The index of MVNOs, the total number of MVNOs and the set of MVNOs
$r_i$ and $\mathbf{R}$	Available dynamic resources of BS $i$ and the corresponding vector
$v_j$ and $w_j$	True valuation and declared valuation of MVNO $j$
$\mathbf{Y}_j$ and $\mathbf{Z}_j$	True demanded dynamic resource vector and declared dynamic resource vector of MVNO $j$
$\mathbf{b}_j$ and $\mathbf{B}$	Bid of MVNO $j$ and the corresponding bid vector
$x_j$ and $\mathbf{x}$	Winner selection variable and the corresponding vector
$p_j$ and $\mathbf{p}$	Payment of MVNO $j$ and the corresponding vector
$u_i$ and $\mathbf{U}$	Price of BS $i$ and the corresponding vector
$f_j$ and $\mathbf{F}$	Reserve price of bid $b_j$ and the corresponding vector

We consider an RaaS cloud with  $N$  BSs and  $M$  MVNOs. We adopt the resource-based

provisioning model [47] for resource sharing among MVNOs: for a BS  $i$ , an MVNO  $j$  demands a slice (in terms of percentage) of its resources so that  $j$  can provide wireless service to its mobile users that are associated with BS  $i$ . In our model, resources of BSs are allocated to MVNOs in a hybrid way (both statically and dynamically). In a static manner, an MVNO  $j$  reserves certain percent of the total resources at each BS  $i$  (denoted by  $\overline{r}_{ij}$ ) for a long period of time (e.g., a month or a quarter) and makes the corresponding payment in advance according to a long-term forecasting for user traffic demands based on historical data. These resources are called static resources and are guaranteed to be available for MVNO  $j$ . However, since both user traffic demands and link data rates are time-varying, static resources may not be sufficient for an MVNO for a certain short period of time. So we need to provide a way for MVNOs to request more resources from BSs according to its real-time needs. The remaining resources of BS  $i$  can be given by  $r_i = 1 - \sum_{j=1}^M \overline{r}_{ij}$ , which are referred to as *available dynamic resources* of BS  $i$ .  $\mathbf{R} = (r_1, \dots, r_i, \dots, r_N)$  is a vector for available dynamic resources at each BS. Dynamic resource allocation is conducted periodically (e.g., once every 30min). Then the real-time demand of MVNO  $j$  at BS  $i$  can be given by  $y_{ij} = \max(d_{ij} - \overline{r}_{ij}, 0)$ , where  $d_{ij}$  is the fraction of resources needed by MVNO  $j$  at BS  $i$ , which can be estimated according to current link data rates and user traffic demands.  $\mathbf{Y}_j = (y_{1j}, \dots, y_{ij}, \dots, y_{Nj})$  denotes the *demanded dynamic resource vector* of MVNO  $j$ .

RaaS can be formulated as an auction mechanism design problem. In the RaaS auction, the seller (i.e., the cloud service provider) sells available dynamic resources to bidders or buyers (i.e., MVNOs) who bid for them. Each MVNO  $j$  is asked to declare a bid  $\mathbf{b}_j = (w_j, \mathbf{Z}_j)$ , where  $w_j$  is the valuation and  $\mathbf{Z}_j = (z_{1j}, \dots, z_{ij}, \dots, z_{Nj})$  is the declared dynamic resource vector. Note that the true valuation  $v_j$  and the true demanded dynamic resource vector  $\mathbf{Y}_j$  are private information only known to MVNO  $j$ . So  $w_j$  and  $\mathbf{Z}_j$  could be different from  $v_j$  and  $\mathbf{Y}_j$  respectively. Each MVNO  $j$  is a “single-minded bidder [65]” in the sense that valuation is  $v_j$  if it gets dynamic resource no less than  $\mathbf{Y}_j$  and 0 otherwise.  $\mathbf{B} =$

$(\mathbf{b}_1, \dots, \mathbf{b}_j, \dots, \mathbf{b}_M)$  is the bid vector. We use  $\mathbf{B}_{-j}$  to denote the bids of all bidders except  $j$ , so  $\mathbf{B} = (\mathbf{b}_j, \mathbf{B}_{-j})$ .

RaaS auction takes  $\mathbf{B}$  and  $\mathbf{R}$  as input, and the output includes a winner vector  $\mathbf{x}(\mathbf{B}, \mathbf{R}) = (x_1, \dots, x_j, \dots, x_M)$  and a payment vector  $\mathbf{p}(\mathbf{B}, \mathbf{R}) = (p_1, \dots, p_j, \dots, p_M)$ .  $x_j = 1$  if bidder  $j$  wins and is allocated the declared dynamic resources  $\mathbf{Z}_j$ ;  $x_j = 0$ , otherwise.  $p_j$  is the payment bidder  $j$  will make to the seller. The dynamic resource allocation must satisfy the following constraints:  $\sum_{j=1}^M z_{ij}x_j \leq r_i, \forall i \in \mathbf{I}$ . Based on the output of the auction, the *payoff* [65] of bidder  $j$  is defined as

$$u_j = \begin{cases} v_j - p_j, & x_j = 1; \\ 0, & x_j = 0. \end{cases} \quad (2.1)$$

The *social welfare* [65] is defined as the total valuation of all winning bidders, i.e.,  $\sum_{j=1}^M v_j x_j$ .

When designing an auction mechanism, it is desirable to have the following three properties [65]:

- **Individual Rationality:** an auction mechanism is *individually rational* if for any bidder  $j$ , the payoff is non-negative when bidder  $j$  bids its true value  $(v_j, \mathbf{Y}_j)$ .
- **Truthfulness:** an auction mechanism is *truthful* if and only if for every bidder  $j$  and  $\mathbf{B}_{-j}$ , bidder  $j$  will not increase its payoff by making any other bid  $(w_j, \mathbf{Z}_j)$  instead of its true value  $(v_j, \mathbf{Y}_j)$ ; i.e., bidder  $j$ 's payoff for bidding  $(v_j, \mathbf{Y}_j)$  is at least its payoff for bidding any other bid  $(w_j, \mathbf{Z}_j)$ .
- **Computational Efficiency:** an auction mechanism is *computationally efficient* if the outcome can be computed in polynomial time.

Among these three properties, truthfulness is the most challenging one to achieve. In order to design a truthful auction mechanism, we introduce the following definitions.

**Definition 2.1** (*w*-Monotonicity). *If bidder  $j$  wins by bidding  $(w_j^*, (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$ , then it also wins by bidding  $(w_j', (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$  with any  $w_j' \geq w_j^*$ .*

**Definition 2.2** (*z*-Monotonicity). *If bidder  $j$  wins by bidding  $(w_j^*, (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$ , then it also wins by bidding  $(w_j^*, (z'_{1j}, \dots, z'_{ij}, \dots, z'_{Nj}))$  with all  $z'_{ij} \leq z_{ij}^*$ .*

**Definition 2.3** (Critical Payment [65]). *The payment  $p_j$  for winning bidder  $j$  is set to the critical value  $c_j$  such that bidder  $j$  wins if  $w_j > c_j$ , and loses if  $w_j < c_j$ .*

**Lemma 2.1.** *In an RaaS auction mechanism, if *w*-Monotonicity, *z*-Monotonicity and Critical Payment are satisfied, a bidder will not increase its payoff by bidding  $(v_j, \mathbf{Z}_j) = (v_j, (z_{1j}, \dots, z_{ij}, \dots, z_{Nj}))$  instead of  $(v_j, \mathbf{Y}_j) = (v_j, (y_{1j}, \dots, y_{ij}, \dots, y_{Nj}))$ , where  $\mathbf{Y}_j \neq \mathbf{Z}_j$ .*

*Proof.* We examine two possible cases:

1)  $z_{ij} < y_{ij}$  for one or more  $i$ . In this case, by bidding  $(v_j, \mathbf{Z}_j)$ , the payoff is non-positive since the valuation is 0 when single-minded bidder  $j$ 's resource demand  $\mathbf{Y}_j$  cannot be met. However, the payoff of bid  $(v_j, \mathbf{Y}_j)$  is non-negative because if  $(v_j, \mathbf{Y}_j)$  is a losing bid, the payoff is 0; if  $(v_j, \mathbf{Y}_j)$  is a winning bid, the payoff will be non-negative.

2)  $z_{ij} \geq y_{ij}$  for every  $i$ . Denote the Critical Payment for bidding  $(v_j, \mathbf{Y}_j)$  by  $p$ , and denote the Critical Payment for bidding  $(v_j, \mathbf{Z}_j)$  by  $p^*$ . Based on *z*-Monotonicity, we know that if a bidder loses by bidding  $(v_j, \mathbf{Y}_j)$ , it will also lose by bidding  $(v_j, \mathbf{Z}_j)$ . Or equivalently, for any  $v_j < p$ , we have  $v_j < p^*$ . So  $p^* \geq p$ . We have two sub-cases: a)  $(v_j, \mathbf{Z}_j)$  is a losing bid. In this sub-case, the payoff of bid  $(v_j, \mathbf{Y}_j)$  is non-negative because if  $(v_j, \mathbf{Y}_j)$  is a losing bid, the payoff is 0; if  $(v_j, \mathbf{Y}_j)$  is a winning bid, the payoff will be non-negative. b)  $(v_j, \mathbf{Z}_j)$  is a winning bid. In this sub-case, a bidder with  $(v_j, \mathbf{Y}_j)$  will also win and the payment will not increase.  $\square$

**Theorem 2.1.** *An RaaS auction mechanism is truthful, if it satisfies *w*-Monotonicity, *z*-Monotonicity and Critical Payment.*

*Proof.* According to the above definition of truthfulness, we will show that a bidder will not increase its payoff by bidding any other bid  $(w_j, \mathbf{Z}_j) = (w_j, (z_{1j}, \dots, z_{ij}, \dots, z_{Nj}))$  instead of

$(v_j, \mathbf{Y}_j) = (v_j, (y_{1j}, \dots, y_{ij}, \dots, y_{Nj}))$ . We will first show that a bidder will not increase its payoff by bidding  $(w_j, \mathbf{Z}_j)$  instead of  $(v_j, \mathbf{Z}_j)$ , where  $v_j \neq w_j$ . Denote the Critical Payment for bidding  $(v_j, \mathbf{Z}_j)$  by  $p$ . We have two cases:

1)  $(v_j, \mathbf{Z}_j)$  is a losing bid. In this case,  $v_j < p$ . If a bidder with  $(w_j, \mathbf{Z}_j)$  loses, it would not be more beneficial than bidding  $(v_j, \mathbf{Z}_j)$ . If a bidder with  $(w_j, \mathbf{Z}_j)$  wins, it makes the same payment  $p$  because the Critical Payment is independent of  $w_j$ ; since  $p > v_j$ , the payoff of bidding  $(w_j, \mathbf{Z}_j)$  is negative.

2)  $(v_j, \mathbf{Z}_j)$  is a winning bid. If  $w_j > p$ , a bidder with  $(w_j, \mathbf{Z}_j)$  wins with the same payment  $p$ . If  $w_j < p$ , a bidder with  $(w_j, \mathbf{Z}_j)$  loses with 0 payoff.

The above two cases show that a bidder will not increase its payoff by bidding  $(w_j, \mathbf{Z}_j)$  instead of  $(v_j, \mathbf{Z}_j)$ . Furthermore, in Lemma 2.1, we have proved that a bidder will not increase its payoff by bidding  $(v_j, \mathbf{Z}_j)$  instead of  $(v_j, \mathbf{Y}_j)$ . Therefore, a bidder will not increase its payoff by bidding any other  $(w_j, \mathbf{Z}_j)$  instead of  $(v_j, \mathbf{Y}_j)$ . This completes the proof.  $\square$

## 2.3 Auction Mechanism with Optimal Social Welfare

In this section, we present a VCG-based (Vickery-Clarke-Groves[65]) auction mechanism that can achieve optimal social welfare.

### 2.3.1 Optimal RaaS Auction Design (Optimal-RaaS)

The RaaS auction design problem consists of two subproblems: Winner Selection and Price Determination. The Winner Selection problem can be formulated as the following Integer Linear Programming (ILP) problem:

ILP-Winner

$$\max \sum_{j \in \mathbf{J}} w_j x_j \quad (2.2)$$

Subject to:

$$\sum_{j \in \mathbf{J}} z_{ij} x_j \leq r_i, \quad \forall i \in \mathbf{I}; \quad (2.3)$$

$$x_j \in \{0, 1\}, \quad \forall j \in \mathbf{J}; \quad (2.4)$$

The objective is to maximize the social welfare. Constraints (2.3) ensure that for each BS, the sum of demanded dynamic resources does not exceed its available dynamic resources. Denote the optimal value of the ILP by  $\Psi(\mathbf{B})$ . Next, we present an auction mechanism that can achieve optimal social welfare, which is referred to as *Optimal-RaaS*.

(1) Winner Selection: select winners  $\mathbf{x}^*$  by solving ILP-Winner;

(2) Price Determination:  $p_j := \Psi(\mathbf{B}_{-j}) - (\Psi(\mathbf{B}) - w_j)$  if  $x_j^* = 1$  and  $p_j := 0$  otherwise.

$\Psi(\mathbf{B}_{-j})$  is the optimal value of ILP-Winner with bid  $\mathbf{b}_j$  removed.

### 2.3.2 Proof of Properties

Although Optimal-RaaS is VCG-based, the proofs of properties are non-trivial because the bids in RaaS model are multidimensional[50]. In order to prove the truthfulness of Optimal-RaaS, we show that the Winner Selection satisfies  $w$ -Monotonicity and  $z$ -Monotonicity. Furthermore, the Critical Payment condition is satisfied by the Price Determination.

**Lemma 2.2.**  *$w$ -Monotonicity is satisfied in the Winner Selection of Optimal-RaaS.*

*Proof.* Suppose that bidder  $j$  wins by bidding  $\mathbf{b}_j^* = (w_j^*, (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$ . Let  $\mathbf{x}$  be the winner vector. We will prove that it also wins by bidding  $\mathbf{b}_j' = (w_j', (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$  with any  $w_j' > w_j^*$  by contradiction. Suppose it will lose by bidding  $\mathbf{b}_j'$ . Then  $\Psi((\mathbf{b}_j', \mathbf{B}_{-j})) =$

$\Psi(\mathbf{B}_{-j})$ . Since bidder  $j$  wins by bidding  $\mathbf{b}_j^*$ ,  $\Psi(\mathbf{B}_{-j}) < \Psi((\mathbf{b}_j^*, \mathbf{B}_{-j}))$ . Therefore  $\Psi((\mathbf{b}'_j, \mathbf{B}_{-j})) < \Psi((\mathbf{b}_j^*, \mathbf{B}_{-j}))$ . Having the same winner vector  $\mathbf{x}$ , the social welfare with  $(\mathbf{b}'_j, \mathbf{B}_{-j})$  would be greater than the social welfare with  $(\mathbf{b}_j^*, \mathbf{B}_{-j})$ , because  $w'_j > w_j^*$ ; this contradicts the statement that  $\Psi((\mathbf{b}'_j, \mathbf{B}_{-j})) < \Psi((\mathbf{b}_j^*, \mathbf{B}_{-j}))$ . Hence the supposition is false, and bidder  $j$  will also win by bidding  $\mathbf{b}'_j$ . This completes the proof.  $\square$

**Lemma 2.3.**  *$z$ -Monotonicity is satisfied in the Winner Selection of Optimal-RaaS.*

*Proof.* Suppose that bidder  $j$  wins by bidding  $\mathbf{b}_j^* = (w_j^*, (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$ . Let  $\mathbf{x}$  be the winner vector. We will prove that it also wins by bidding  $\mathbf{b}'_j = (w_j^*, (z_{1j}^*, \dots, z'_{ij}, \dots, z_{Nj}^*))$  with any  $z'_{ij} < z_j^*$  by contradiction. Suppose it will lose by bidding  $\mathbf{b}'_j$ . Then  $\Psi((\mathbf{b}'_j, \mathbf{B}_{-j})) = \Psi(\mathbf{B}_{-j})$ . Because bidder  $j$  wins by bidding  $\mathbf{b}_j^*$ ,  $\Psi(\mathbf{B}_{-j}) < \Psi((\mathbf{b}_j^*, \mathbf{B}_{-j}))$ . Therefore  $\Psi((\mathbf{b}'_j, \mathbf{B}_{-j})) < \Psi((\mathbf{b}_j^*, \mathbf{B}_{-j}))$ . Having the same winner vector  $\mathbf{x}$ , the social welfare with  $(\mathbf{b}'_j, \mathbf{B}_{-j})$  is equal to  $(\mathbf{b}_j^*, \mathbf{B}_{-j})$ ; this contradicts the statement that  $\Psi((\mathbf{b}'_j, \mathbf{B}_{-j})) < \Psi((\mathbf{b}_j^*, \mathbf{B}_{-j}))$ . Hence the supposition is false, and bidder  $j$  will also win by bidding  $\mathbf{b}'_j$ . This completes the proof.  $\square$

**Lemma 2.4.**  *$p_j = \Psi(\mathbf{B}_{-j}) - (\Psi(\mathbf{B}) - w_j)$  is a critical value for each winning bidder  $j$  in Optimal-RaaS.*

*Proof.* In Optimal-RaaS, the payment of each winning bidder is calculated based on the opportunity cost [65], which is introduced to all the other bidders by the presence of the winning bidder. Therefore, if the bidder bids less than this price, it will not be selected as a winner, which leads to higher social welfare [97].  $\square$

**Theorem 2.2.** *Optimal-RaaS is truthful.*

*Proof.* According to Lemmas 2.2, 2.3, 2.4 and Theorem 2.1, Optimal-RaaS is truthful.  $\square$

**Theorem 2.3.** *Optimal-RaaS is individually rational.*

*Proof.* For any bidder  $j$  bidding its true value  $(v_j, \mathbf{Y}_j)$ , we consider two possible cases:  
1) Bidder  $j$  is a winner. Its payoff is  $u_j = v_j - p_j = v_j - (\Psi(\mathbf{B}_{-j}) - (\Psi(\mathbf{B}) - v_j)) =$

$\Psi(\mathbf{B}) - \Psi(\mathbf{B}_{-j}) \geq 0$ , where the last inequality follows from the optimality of  $\Psi(\mathbf{B})$ . 2) Bidder  $j$  is not a winner. Its payoff is 0. This completes the proof.  $\square$

## 2.4 Greedy Auction Mechanism

Although Optimal-RaaS is both individually rational and truthful, it is not computationally efficient since solving ILP-Winner may take exponential time. In this section, we present an auction mechanism, called Greedy RaaS Auction Design (GRAD), which has all the three desirable properties.

### 2.4.1 Greedy RaaS Auction Design (GRAD)

GRAD consists of two phases too: *Winner Selection* and *Price Determination*. In the *Winner Selection* (Algorithm 2.1), the basic idea is to keep adding the bidder with the largest weight to the solution. We adopt the following weight  $\alpha_j$  as the metric for sorting bidders and selecting winners:

$$\alpha_j = \frac{w_j}{\sum_{i \in \mathbf{I}} \frac{z_{ij}}{r_i}}. \quad (2.5)$$

In each iteration, the bidder with the maximum weight  $\alpha_j$  is selected as the winner. Then we update  $\mathbf{R}$  by subtracting the corresponding demanded dynamic resource vector  $\mathbf{Z}_j$  of the selected winner from it. All the bidders who demand more dynamic resources than the available resources in the updated  $\mathbf{R}$  will be eliminated from the auction. This process iterates until the bidder list is empty.

In the *Price Determination* (Algorithm 2.2), to find the payment for a winning bidder  $j$ , we remove  $j$  from the bidder list, do the Winner Selection as above with the rest bidders until a winning bidder  $k$  is found such that its selection can disqualify  $j$  from winning the auction and determine the price accordingly (lines 9–10 in Algorithm 2.2).

---

**Algorithm 2.1:** Winner Selection of GRAD
 

---

**Input :** Bid vector  $\mathbf{B}$  and Available dynamic resource vector  $\mathbf{R}$   
**Output:** Winner vector  $\mathbf{x}$

- 1  $x_j := 0, \forall j \in \mathbf{J};$
- 2  $\alpha_j := \frac{w_j}{\sum_{i \in \mathbf{I}} \frac{z_{ij}}{r_i}}, \forall j \in \mathbf{J};$
- 3 Sort the bidders in the non-increasing order of  $\alpha_j$  and store the sorted list of their indices into  $\mathbf{L};$
- 4 **while**  $\mathbf{L} \neq \emptyset$  **do**
- 5 Let  $j$  be the next bidder in  $\mathbf{L};$
- 6  $x_j := 1;$
- 7  $\mathbf{L} := \mathbf{L} \setminus \{j\};$
- 8  $\mathbf{R} := \mathbf{R} - \mathbf{Z}_j;$
- 9 **forall**  $m \in \mathbf{L}$  **do**
- 10 **if**  $\exists i \in \mathbf{I}$  s.t.  $z_{im} > r_i$  **then**
- 11  $\mathbf{L} := \mathbf{L} \setminus \{m\};$
- 12 **return**  $\mathbf{x};$

---



---

**Algorithm 2.2:** Price Determination of GRAD
 

---

**Input :** Bid vector  $\mathbf{B}$ , Available dynamic resource vector  $\mathbf{R}$ , Winner vector  $\mathbf{x}$ ,  
 Sorted bidder list  $\mathbf{L}$  and weight  $\alpha_j, \forall j \in \mathbf{J}$   
**Output:** Payment vector  $\mathbf{p}$

- 1 **forall**  $j \in \mathbf{L}$  **do**
- 2  $p_j := 0;$
- 3 **if**  $x_j = 1$  **then**
- 4  $\mathbf{L}' := \mathbf{L} \setminus \{j\}; \mathbf{R}' := \mathbf{R};$
- 5 **while**  $\mathbf{L}' \neq \emptyset$  **do**
- 6 Let  $k$  be the next bidder in  $\mathbf{L}';$
- 7  $\mathbf{L}' := \mathbf{L}' \setminus \{k\};$
- 8  $\mathbf{R}' := \mathbf{R}' - \mathbf{Z}_k;$
- 9 **if**  $\exists i \in \mathbf{I}$  s.t.  $z_{ij} > r'_i$  **then**
- 10  $p_j := (\alpha_k \sum_i \frac{z_{ij}}{r_i});$  **break;**
- 11 **forall**  $m \in \mathbf{L}'$  **do**
- 12 **if**  $\exists i \in \mathbf{I}$  s.t.  $z_{im} > r'_i$  **then**
- 13  $\mathbf{L}' := \mathbf{L}' \setminus \{m\};$
- 14 **return**  $\mathbf{p};$

---

## 2.4.2 Proof of Properties

**Lemma 2.5.** *w-Monotonicity is satisfied in the Winner Selection of GRAD.*

*Proof.* Suppose that bidder  $j$  wins by bidding  $(w_j^*, \mathbf{Z}_j^*) = (w_j^*, (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$ . We prove that it will also win by bidding  $(w_j', \mathbf{Z}_j^*) = (w_j', (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$  with any  $w_j' > w_j^*$ . Let  $\mathbf{L}^*$  and  $\mathbf{L}'$  denote the sorted lists when  $j$  bids  $(w_j^*, \mathbf{Z}_j^*)$  and  $(w_j', \mathbf{Z}_j^*)$  respectively. The positions of  $j$  in  $\mathbf{L}^*$  and  $\mathbf{L}'$  are denoted by  $q^*$  and  $q'$  respectively. Since  $\alpha_j' = \frac{w_j'}{\sum_{i=1}^N \frac{z_{ij}}{r_i}} > \alpha_j^* = \frac{w_j^*}{\sum_{i=1}^N \frac{z_{ij}}{r_i}}$ , it is clear that  $q' \leq q^*$ . Furthermore, at lines 9–11 in Algorithm 2.1, since  $j$  has not been eliminated at  $q^*$ , it will not be eliminated at  $q'$  neither. Therefore,  $j$  will still win with bid  $(w_j', \mathbf{Z}_j^*)$ .  $\square$

**Lemma 2.6.** *z-Monotonicity is satisfied in the Winner Selection of GRAD.*

*Proof.* Suppose that bidder  $j$  wins by bidding  $(w_j^*, \mathbf{Z}_j^*) = (w_j^*, (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$ . We prove that it will also win by bidding  $(w_j^*, \mathbf{Z}_j') = (w_j^*, (z_{1j}^*, \dots, z_{ij}', \dots, z_{Nj}^*))$  with any  $z_{ij}' < z_{ij}^*$ . Let  $\mathbf{L}^*$  and  $\mathbf{L}'$  denote the sorted lists when  $j$  bids  $(w_j^*, \mathbf{Z}_j^*)$  and  $(w_j^*, \mathbf{Z}_j')$  respectively; the positions of  $j$  in  $\mathbf{L}^*$  and  $\mathbf{L}'$  are denoted by  $q^*$  and  $q'$  respectively. For the sake of presentation, denote  $\sum_{i=1}^N \frac{z_{ij}}{r_i}$  by  $s(\mathbf{Z}_j)$ . With  $z_{ij}' < z_{ij}^*$ , we have  $s(\mathbf{Z}_j') < s(\mathbf{Z}_j^*)$ . So  $\alpha_j' = \frac{w_j^*}{s(\mathbf{Z}_j')} > \alpha_j^* = \frac{w_j^*}{s(\mathbf{Z}_j^*)}$ ; thus  $q' \leq q^*$ . Furthermore, at lines 9–11 in Algorithm 2.1, since  $j$  has not been eliminated at  $q^*$ , it will not be eliminated at  $q'$  neither. Therefore,  $j$  will still win with bid  $(w_j^*, \mathbf{Z}_j')$ .  $\square$

**Lemma 2.7.** *The payment  $p_j$  is set to a critical value for each winning bidder  $j$  in GRAD.*

*Proof.* Let  $k$  be the first bidder in the list, whose selection can disqualify  $j$ . Let  $c_j = \alpha_k \sum_{i=1}^N \frac{z_{ij}}{r_i}$ . If bidder  $j$  bids  $w_j < c_j$ , then  $\alpha_j < \alpha_k$ , meaning  $j$  will be placed behind  $k$  in the sorted list and thus will be eliminated from the auction. If bidder  $j$  bids  $w_j > c_j$ , then  $\alpha_j > \alpha_k$ , meaning  $j$  will be placed ahead of  $k$ .  $j$  is ahead of any bidder that can disqualify  $j$ , because  $k$  is the first of such bidders. Therefore  $j$  will be selected as a winner and  $c_j$  is the critical value for winning bidder  $j$ . Since the payment  $p_j$  is set to  $c_j$  in the algorithm, we prove the lemma.  $\square$

**Theorem 2.4.** *GRAD is truthful.*

*Proof.* According to Lemmas 2.5, 2.6, 2.7 and Theorem 2.1, GRAD is truthful.  $\square$

**Theorem 2.5.** *GRAD is individually rational.*

*Proof.* We consider two possible cases: 1) Bidder  $j$  is not a winner. From Algorithm 2.2,  $j$  pays 0. Therefore its payoff is 0. 2) Bidder  $j$  is a winner. Since GRAD satisfies the Critical Payment property as shown in Lemma 2.7, we have  $w_j > c_j = p_j$ . In a truthful mechanism,  $w_j = v_j$ . Hence we have  $v_j - p_j > 0$ . Therefore the payoff is always non-negative. This completes the proof.  $\square$

**Theorem 2.6.** *GRAD is computationally efficient.*

*Proof.* In Algorithm 2.1, calculating  $\alpha$  (line 2) takes  $O(MN)$  time. Furthermore, the while-loop (lines 4–11) takes  $O(M^2N)$  time. Hence time complexity of Algorithm 2.1 is  $O(M^2N)$ . In Algorithm 2.2, the for-loop takes  $O(M^3N)$  time, since the for-loop (lines 1–13) runs  $M$  iterations, and in each iteration, while-loop (lines 5–13) takes  $O(M^2N)$  time. So the time complexity of Algorithm 2.2 is  $O(M^3N)$ . Therefore, the overall time complexity of GRAD is  $O(M^3N)$ . This completes the proof.  $\square$

## 2.5 Auction Mechanisms with Reserve Prices

Optimal-RaaS and GRAD are both individually rational and truthful. Moreover, Optimal-RaaS can achieve optimal social welfare and GRAD is computationally efficient with sub-optimal social welfare. However, there is an issue that the Price Determination of Optimal-RaaS and GRAD might end up with 0 payment for some winners. To be more specific, in Optimal-RaaS with more available dynamic resources, for some winning bidders, it turns out that  $\Psi(\mathbf{B}_{-j}) = (\Psi(\mathbf{B}) - w_j)$ , yielding 0 payment. In GRAD, with more available dynamic resources, there is a higher chance that for some winning bidders, the *if* condition of line 9 will not be satisfied; therefore line 10 will not be executed, resulting in 0 payments.

In order to mitigate this problem and grant more right to the seller to determine the payment, we introduce the following definition [50]:

**Definition 2.4** (Reserve Price). *The seller reserves the right not to sell the declared dynamic resource of  $\mathbf{Z}_j$  if the payment determined is lower than some threshold price. Such a threshold price is called a reserve price, denoted by  $f_j$ .*

It is desirable to have reserve prices, whose values are closely related to declared dynamic resources of each BS. There may be multiple options to calculate such reserve prices. We choose to use the following formulation:

$$f_j = \sum_{i \in \mathbf{I}} u_i z_{ij}, \forall j \in \mathbf{J}; \quad (2.6)$$

where  $u_i$  is the price of BS  $i$ , i.e., the price of allocating 100% of the resources of BS  $i$ . Next, we present the auction mechanism with reserve prices.

### 2.5.1 RaaS Auction with Reserve Prices (RaaS-RP)

The RaaS auction mechanism design problem with Reserve Prices (RaaS-RP) consists of three subproblems: Bidder Screening, Winner Selection and Price Determination.

- (1) Bidder Screening: use Algorithm 2.3;
- (2) Winner Selection: solve the Winner Selection problem of Optimal-RaaS or GRAD with  $\mathbf{J}'$  (instead of  $\mathbf{J}$ );
- (3) Price Determination:  $p_j := \max\{f_j, p_j^*\}$ , if  $x_j^* = 1$  and  $p_j := 0$  otherwise.  $p_j^*$  is the payment determined by Optimal-RaaS or GRAD.

Note that we have integrated the reserve prices  $f_j$  in the Bidder Screening and Price Determination. In the Bidder Screening, those bidders with declared valuation  $w_j < f_j$ , will be screened out of the auction. In the Price Determination,  $p_j = \max\{f_j, p_j^*\}$ , ensuring

---

**Algorithm 2.3:** Bidder Screening of RaaS-RP
 

---

**Input** : Bid vector  $\mathbf{B}$ , Bidder set  $\mathbf{J}$  and BS price vector  $\mathbf{U}$   
**Output:** Remaining bidder set  $\mathbf{J}'$  and Reserve price vector  $\mathbf{F}$

- 1  $f_j := \sum_{i \in \mathbf{I}} u_i z_{ij}, \forall j \in \mathbf{J};$
- 2 **forall**  $j \in \mathbf{J}$  **do**
- 3     **if**  $f_j > w_j$  **then**
- 4          $x_j := 0;$
- 5          $\mathbf{J} := \mathbf{J} \setminus j;$
- 6  $\mathbf{J}' := \mathbf{J};$
- 7 **return**  $\mathbf{J}', \mathbf{F};$

---

the payment  $p_j$  is no less than the reserve price  $f_j$ . Note that RaaS-RP is designed on the basis of Optimal-RaaS or GRAD, so RaaS-RP could be either Optimal-RaaS with reserve prices (Optimal-RaaS-RP) or GRAD with reserve prices (GRAD-RP).

## 2.5.2 Proof of Properties

In order to prove the truthfulness of RaaS-RP, we will prove that  $w$ -Monotonicity,  $z$ -Monotonicity and Critical Payment condition will all be satisfied.

**Lemma 2.8.**  *$w$ -Monotonicity is satisfied in RaaS-RP.*

*Proof.* Suppose that bidder  $j$  wins by bidding  $\mathbf{b}_j^* = (w_j^*, (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$ . We will prove that it also wins by bidding  $\mathbf{b}_j' = (w_j', (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$  with any  $w_j' > w_j^*$ . Let  $f_j^*$  be the reserve price of  $\mathbf{b}_j^*$  and  $f_j'$  be the reserve price of  $\mathbf{b}_j'$ . In the Bidder Screening, since  $j$  wins by bidding  $\mathbf{b}_j^*$ , it is clear that  $f_j^* \leq w_j^*$ . By bidding  $\mathbf{b}_j'$ , the reserve price  $f_j' = f_j^*$ ; therefore  $w_j' > f_j'$ . Bidder  $j$  will not be screened out by bidding  $\mathbf{b}_j'$ . In the Winner Selection, from Lemma 2.2 and Lemma 2.5, if bidder  $j$  wins by bidding  $\mathbf{b}_j^*$ , it also wins by bidding  $\mathbf{b}_j'$  in Optimal-RaaS or GRAD. This completes the proof.  $\square$

**Lemma 2.9.**  *$z$ -Monotonicity is satisfied in RaaS-RP.*

*Proof.* Suppose that bidder  $j$  wins by bidding  $(w_j^*, \mathbf{Z}_j^*) = (w_j^*, (z_{1j}^*, \dots, z_{ij}^*, \dots, z_{Nj}^*))$ . We prove that it will also win by bidding  $(w_j^*, \mathbf{Z}_j') = (w_j^*, (z_{1j}^*, \dots, z_{ij}', \dots, z_{Nj}^*))$  with any  $z_{ij}' <$

$z_{ij}^*$ . Let  $f_j^*$  be the reserve price of  $(w_j^*, \mathbf{Z}_j^*)$  and  $f_j'$  be the reserve price of  $(w_j^*, \mathbf{Z}_j')$ . In the Bidder Screening, since  $j$  wins by bidding  $(w_j^*, \mathbf{Z}_j^*)$ , it is clear that  $f_j^* \leq w_j^*$ . By bidding  $b_j'$ , the reserve price  $f_j' < f_j^*$ ; therefore  $w_j^* > f_j'$ . Bidder  $j$  will not be screened out by bidding  $b_j'$ . In the Winner Selection, from Lemma 2.3 and Lemma 2.6, if bidder  $j$  wins by bidding  $(w_j^*, \mathbf{Z}_j^*)$ , it also wins by bidding  $(w_j^*, \mathbf{Z}_j')$  in Optimal-RaaS or GRAD. This completes the proof.  $\square$

**Lemma 2.10.**  $p_j = \max\{f_j, p_j^*\}$  is a critical value for each winning bidder  $j$  in RaaS-RP.

*Proof.* Let  $c_j = \max\{f_j, p_j^*\}$ . We now examine the following two cases:

1) If bidder  $j$  bids  $w_j < c_j$ , then we have either  $w_j < f_j$  or  $w_j < p_j^*$ . We now discuss these two cases: 1)  $w_j < f_j$ . In this case, bidder  $j$  will be screened out in the Bidder Selection of RaaS-RP. 2)  $w_j < p_j^*$ . In this case, since  $p_j^*$  is the critical value for winning bidder  $j$  in Optimal-RaaS or GRAD,  $j$  will not be selected as a winner according to Lemma 2.4 and Lemma 2.7. Therefore in either of the cases, bidder  $j$  will not win if  $w_j < c_j$ .

2) If bidder  $j$  bids  $w_j > c_j$ , it is clear that  $w_j > f_j$  and  $w_j > p_j^*$ . Hence  $j$  will not be screened out of the auction in the Bidder Screening, and furthermore, it will be selected as a winner according to Lemma 2.4 and Lemma 2.7.

Therefore,  $c_j$  is the critical value for winning bidder  $j$ . Since the payment  $p_j$  of RaaS is set to  $c_j$ , we prove the lemma.  $\square$

**Theorem 2.7.** RaaS-RP is truthful.

*Proof.* According to Lemmas 2.8, 2.9, 2.10 and Theorem 2.1, RaaS-RP is truthful.  $\square$

**Theorem 2.8.** RaaS-RP is individually rational.

*Proof.* For any bidder  $j$  bidding its true value  $(v_j, \mathbf{Y}_j)$ , we consider two possible cases:

1) Bidder  $j$  is a winner. Its payoff is  $u_j = v_j - p_j = v_j - \max\{f_j, p_j^*\}$ . For a winning bidder,  $w_j > f_j$ ; in a truthful mechanism,  $w_j = v_j$ . Hence we have  $v_j > f_j$ . Meanwhile,

in Theorem 2.3 and Theorem 2.5, we have proved  $v_j > p_j$  for a winning bidder. Therefore  $u_j = v_j - \max\{f_j, p_j^*\} > 0$ . 2) Bidder  $j$  is not a winner. Its payoff is 0. This completes the proof.  $\square$

## 2.6 Performance Evaluation

The simulation runs were conducted on a computer with a 2.5GHz Intel i5 CPU and 4GB memory. The social welfare is given in terms of credits, whose monetary worth can be determined by the cloud service provider (seller). In the simulation, there were 40 BSs in total.

### 2.6.1 Performance Evaluation of Optimal-RaaS and GRAD

We evaluated the performance of Optimal-RaaS and GRAD in terms of running time and social welfare by varying the number of MVNOs (bidders), the demanded dynamic resources and the available dynamic resources. Specifically, we came up with the following 3 scenarios for our simulation. All the numbers presented in the figures are averages over 20 runs.

1) In Scenario 1, the demanded dynamic resources were uniformly distributed in  $[0\%, 5\%]$ ; the available dynamic resources followed a uniform distribution in  $[50\%, 70\%]$ . The number of MVNOs was increased from 10 to 90 with a step size of 20. The corresponding results are presented in Figs. 2.2 (running time) and 2.3(a) (social welfare).

2) In Scenario 2, the number of MVNOs was fixed to 50; the available dynamic resources and demanded dynamic resources were uniformly distributed in  $[50\%, 70\%]$  and  $[0\%, u_1]$  respectively, where  $u_1$  was increased from 3% to 7% with a step size of 1%. The corresponding results are presented in Fig. 2.3(b).

3) In Scenario 3, the number of MVNOs was fixed to 50; the demanded dynamic resources and available dynamic resources were uniformly distributed in  $[0\%, 5\%]$  and

$[50\%, u_2]$  respectively, where  $u_2$  was increased from 50% to 90% with a step size of 10%.

The corresponding results are presented in Fig. 2.3(c).

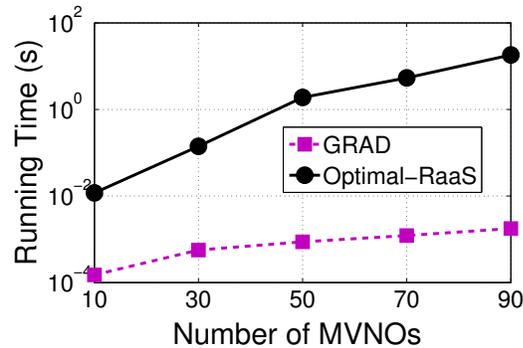


Fig. 2.2: Running time

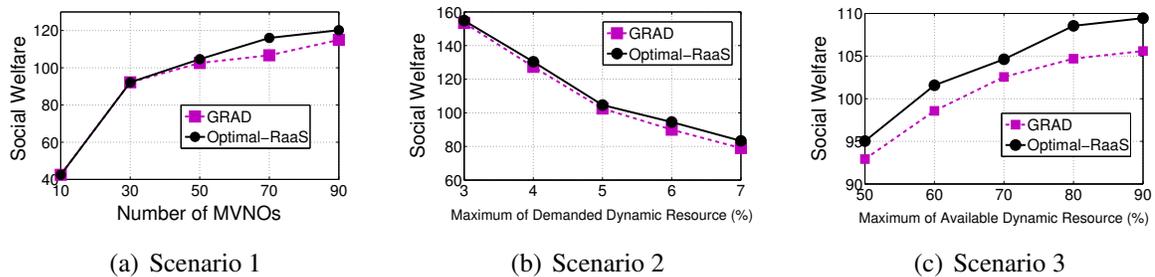


Fig. 2.3: Social welfare

We can make the following observations from these results:

1) Fig. 2.2 shows the running times of the proposed mechanisms with various numbers of MVNOs. The running time of GRAD is  $\frac{1}{78}$  of that of Optimal-RaaS on small cases with only 10 MVNOs. Running time savings become more and more significant when the number of MVNOs becomes larger and larger. Specifically, when it turns to 90, the running time of GRAD is about  $\frac{1}{10,000}$  of that of Optimal-RaaS. This leads us to believe that substantial running time savings can be achieved by using GRAD.

2) Fig. 2.3 shows the performance of the the proposed methods with regard to social welfare. Social welfare values given by GRAD are always lower than (as expected), but close to the optimal ones. On average, by varying the number of MVNOs, the maximum

demanded dynamic resource and maximum available dynamic resource, GRAD achieves 97.1%, 97.0% and 97.2% of optimal social welfare, respectively.

3) Monotonicity can be observed in Figure 2.3. Specifically, in Scenario 1, with more MVNOs to choose from, both methods lead to higher social welfare. In Scenario 2, more demanded dynamic resources result in fewer winning bidders (MVNOs), yielding lower social welfare. In Scenario 3, with larger available dynamic resource, more bidders are selected as winners, resulting in higher social welfare.

## 2.6.2 Performance Evaluation of RaaS-RP

With respect to RaaS-RP, we came up with 2 scenarios for our simulation. Scenario 4 is to reveal the fact that there may exist some winners with 0 payment in Optimal-RaaS and GRAD. Scenario 5 was developed to evaluate the performance of RaaS-RP in terms of running time and social welfare by varying BS prices. All the numbers presented in the figures are averages over 20 runs.

1) In Scenario 4, the number of MVNOs was fixed to 30; the demanded dynamic resources were uniformly distributed in  $[0\%, 5\%]$ ; and the available dynamic resources were increased from 50% to 90% with a step size of 10%. The corresponding results are presented in Fig. 2.4.

2) In Scenario 5, the number of MVNOs was fixed to 50; the demanded dynamic resources and available dynamic resources were uniformly distributed in  $[0\%, 5\%]$  and  $[50\%, 90\%]$  respectively; the prices of BSs were uniformly distributed in  $[0, u_3]$ , where  $u_3$  was increased from 0 to 12 with a step size of 3. The corresponding results are presented in Figs. 2.5 and 2.6.

We can make the following observations from these results:

1) Fig. 2.4 shows that for Optimal-RaaS and GRAD, in which there are no reserve prices, monotonicity can be observed between the number of winners with 0 payment and available dynamic resources. Specifically, with more available dynamic resources, both

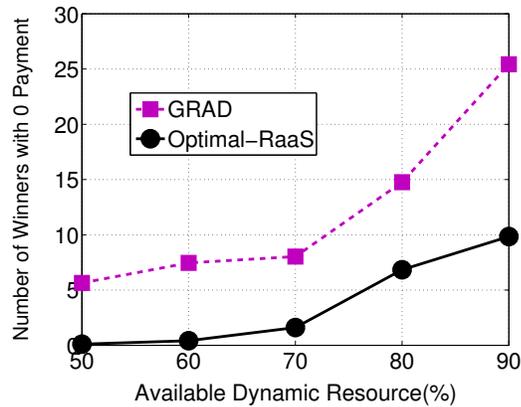


Fig. 2.4: Number of Winners with 0 Payment

methods lead to more winners with 0 payment.

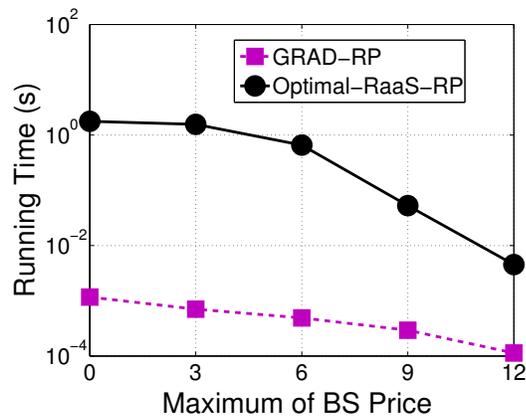


Fig. 2.5: Running Time of RaaS-RP

2) Fig. 2.5 shows that the running times of Optimal-RaaS-RP and GRAD-RP decrease with the increment of BS prices. The reason is, with higher BS prices, more bidders will be screened out of the auction in Bidder Screening of RaaS-RP because their declared valuation  $w_j$  is lower than the reserve price  $f_j$ . Moreover, the comparison of Optimal-RaaS-RP and GRAD-RP shows that running time savings are more significant when the BS price becomes lower. Specifically the running time of GRAD-RP is  $\frac{1}{1123}$  of that of Optimal-RaaS-RP in the case of 0 price and  $\frac{1}{76}$  with the maximum BS price of 12.

3) Fig. 2.6 shows the performance of Optimal-RaaS-RP and GRAD-RP with regards

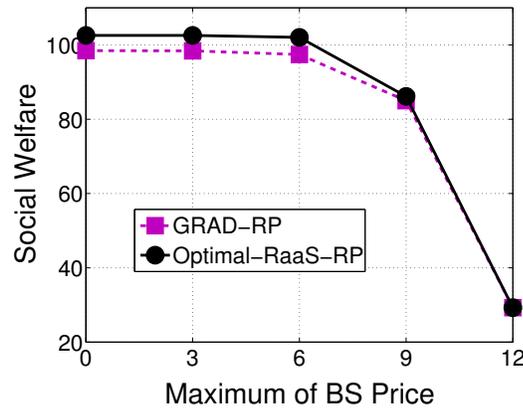


Fig. 2.6: Social Welfare of RaaS-RP

to social welfare. With the increment of BS prices, more bidders will be screened out of the auction, yielding lower social welfare. Furthermore, as expected, social welfare values given by GRAD-RP are always lower than but close to the Optimal-RaaS-RP; on average, GRAD-RP achieves 98.6% of optimal social welfare.

## 2.7 Summary

In this chapter, we proposed a novel auction-based model for RaaS. Based on the proposed model, we studied the auction mechanism design with the objective of maximizing social welfare. First, we proposed Optimal-RaaS, which is an ILP and VCG based auction mechanism that can achieve optimal social welfare. To reduce time complexity, we proposed GRAD, which is a polynomial-time greedy mechanism for the RaaS auction. Both methods have been formally shown to be truthful and individually rational. Extensive simulation results show that GRAD can quickly produce close-to-optimal solutions. Furthermore, to prevent winning bidders from making 0 payment, we introduced reserve prices and presented Optimal-RaaS-RP and GRAD-RP, which were designed based on Optimal-RaaS and GRAD respectively. We have showed that both mechanisms are truthful and individually rational too.

# CHAPTER 3

## SPATIOTEMPORAL MODELING AND PREDICTION IN WIRELESS NETWORKS

### 3.1 Overview

Quite a few models and methods [16] have been proposed for time series analysis with wireless system data. AutoRegression Integrated Moving Average (ARIMA) and Support Vector Regression (SVR) are two most widely used methods, which have been applied to wireless networks. For example, ARIMA has been used in [81, 107] to predict the future traffic load. However, the limitation of ARIMA lies in their natural tendency to concentrate on the mean values of the past series data, which makes it unable to capture the rapid variational process underlying traffic load [40]. SVR model is also limited for the reason that the users need to determine some key parameters for the model, and it lacks a structured way for determining best values for these parameters [40]. More importantly, these methods use only historical data of the target for prediction without taking into account spatial dependency (i.e., neighboring BSs), which, however, is very important in a wireless network.

In this chapter, we propose a novel deep learning approach for spatiotemporal model-

ing and prediction in cellular networks, using big system data. Deep learning is a multi-layer representation learning method [53], which aims to automatically discover a simple but proper representation for the given raw data. Each layer is a non-linear module that transforms the representation of the previous layer into a more compact representation. Deep learning has been shown to dramatically improve the state-of-art on many application domains, including image/video processing, natural language processing, etc [53]. It is particularly suitable to infer information from large datasets and requires very little domain knowledge and engineering by hand. This work aims to show how deep learning can be utilized to model time series data collected from a cellular network and make accurate prediction.

First, we perform a preliminary analysis for a big dataset from the largest wireless carrier in China, China Mobile, and use traffic load as an example to show non-zero temporal autocorrelation and non-zero spatial correlation among neighboring Base Stations (BSs), which motivate us to discover both temporal and spatial dependencies in our study. We then present a hybrid deep learning model for time series prediction, which includes a novel autoencoder-based deep model for spatial modeling and Long Short-Term Memory units (LSTMs) for temporal modeling. The autoencoder-based model consists of a Global Stacked AutoEncoder (GSAE) and multiple Local SAEs (LSAEs), which can offer good representations for input data, reduced model size, and support for parallel and application-aware training. Moreover, we present a new algorithm for training this autoencoder-based spatial model. In addition, we conducted extensive experiments to evaluate the performance of the proposed model using the China Mobile dataset. The results show that our model significantly improves prediction accuracy compared to two commonly used baseline methods, ARIMA and SVR. We also show some results to justify effectiveness of the autoencoder-based spatial model. To the best of our knowledge, we are the first to leverage the emerging deep learning techniques for spatiotemporal modeling and prediction in wireless networks by developing a new hybrid deep model, and showing its effectiveness

and superiority with real data from a major wireless carrier.

## 3.2 Preliminary Data Analysis

In this section, we first describe the dataset used for analysis and evaluation, and then we perform a preliminary analysis for the data, which motivates our design.

### 3.2.1 Dataset

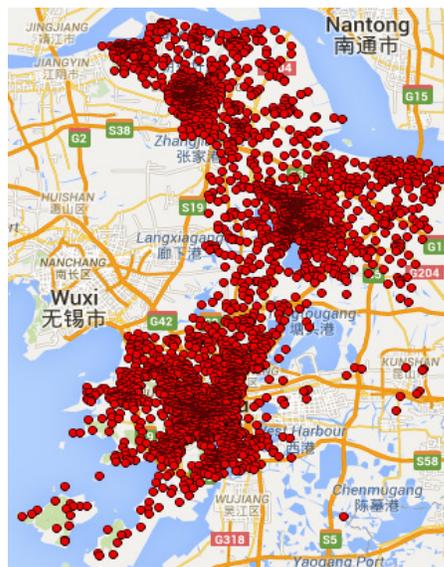


Fig. 3.1: Locations of BSs in our dataset

The dataset consists of data collected from a large LTE network of China Mobile at Suzhou, a major city located in the southeastern part of China. The data was collected from 2,844 BSs, roughly covering an area of 6,500 km<sup>2</sup>. Locations of all the BSs are shown in the map given by Fig. 3.1. Here, our analysis is performed based on the downlink and uplink traffic load. However, the proposed model (Section 3.3) can be applied to other features. The dataset includes average traffic load of each BS in every hour during the

period from 00:00 05/01/2015 to 23:00 09/30/2015. To facilitate data analysis, we divide the target area into a grid, with each cell covering a square of  $500 \times 500 \text{ m}^2$ . Then every BS can be mapped into a cell in the grid. If a cell includes more than one BS, then its traffic load is the aggregated load. Note that unlike traditional cellular networks, current dense small cell networks do not have a hexagon-based layout. A tuple  $(m, n)$  is used to uniquely identify each cell. We denote  $\mathbf{D} = \{d_{m,n,t}\}, \forall m, n, t$ , which is the downlink/uplink traffic load of cell  $(m, n)$  at timeslot  $t$ . Since uplink and downlink can be considered separately, without abusing the notation, we use this to denote both of them. In addition, we denote  $\mathbf{d}_{m,n} = \{d_{m,n,t}\}, \forall t$ .

For each cell  $(m, n)$ , we normalize the data into the range  $[0, 1]$ . We adopt the *tanh estimator* method, a robust and efficient method for normalizing time series data [35], which calculates the normalized values as follows:

$$\hat{\mathbf{d}}_{m,n} = 0.5(\tanh(\frac{0.01(\mathbf{d}_{m,n} - \bar{d}_{m,n})}{\sigma_{\mathbf{d}_{m,n}}}) + 1), \quad (3.1)$$

where  $\bar{d}_{m,n}$  and  $\sigma_{\mathbf{d}_{m,n}}$  are the average and standard deviation of  $\mathbf{d}_{m,n}$  respectively.

### 3.2.2 Data Analysis

In our preliminary analysis, we try to explore data dependency in both the temporal and spatial domains.  $\mathbf{d}_{m,n}$  can be treated as a collection of a random process samples at cell  $(m, n)$ . So we can examine data dependency in terms of temporal autocorrelation and spatial correlation in the temporal and spatial domains respectively. We summarize our main findings in the following:

*Observation 1: Dataset  $\mathbf{D}$  exhibits non-zero autocorrelation in the temporal domain.*

The *sample AutoCorrelation Function* (sample ACF) [16] is a widely used method for discovering data dependency in the temporal domain, which describes the dependency between the values of a sample process as a function of time lag  $h$ . The definition of the

sample ACF at cell  $(m, n)$  can be given as follows (for the sake of readability, we omit the notations of  $m, n$  in this definition):

$$\rho(h) = \frac{\sum_{t=1}^{T-|h|} (d_{t+|h|} - \bar{d})(d_t - \bar{d})}{\sum_{t=1}^T (d_t - \bar{d})^2}, \quad -T < h < T; \quad (3.2)$$

where  $T$  and  $\bar{d}$  are the total count and mean value of data in the temporal dimension, respectively. The autocorrelation value lies in the range  $[-1, 1]$ .  $\rho(h) = 1$  indicates total positive autocorrelation between data with a time lag of  $h$ ; while  $\rho(h) = -1$  means total negative autocorrelation. Note that  $\rho(h) = 0$  denotes no autocorrelation.

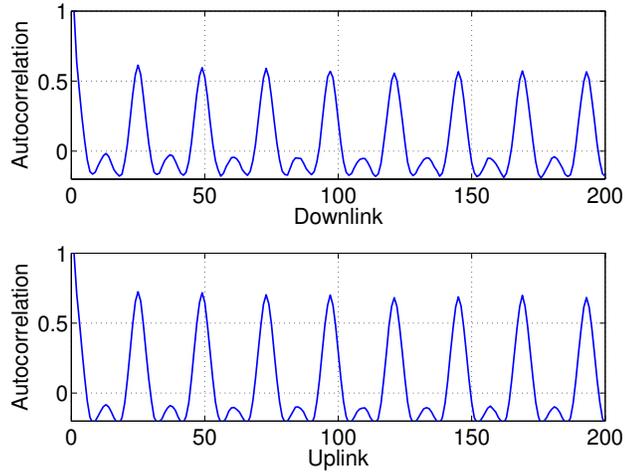


Fig. 3.2: Temporal autocorrelation

Fig. 3.2 shows a sample ACF at time lag  $h = 0, 1, \dots, 200$  for both downlink and uplink data. We can see that when the time lag equals one or multiple of 24 (hours), the autocorrelation is relatively high. This shows that the traffic load at a cell follows a clear daily pattern. For example, the traffic load peak and off-peak hours are similar on each day. Therefore, dataset **D** exhibits non-zero autocorrelation in the temporal domain.

*Observation 2: Dataset **D** reveals non-zero correlation in the spatial domain.*

We examine the data correlation in the spatial domain by calculating a widely used metric [17] for a pair of cells  $(m, n)$  and  $(m', n')$ :

$$\rho = \frac{cov(\mathbf{d}_{m,n}, \mathbf{d}_{m',n'})}{\sigma_{\mathbf{d}_{m,n}} \sigma_{\mathbf{d}_{m',n'}}}, \quad (3.3)$$

where  $cov(\cdot)$  is the covariance operator, and  $\sigma$  is the standard deviation. Similarly, this correlation coefficient ranges in  $[-1, 1]$  as well.

Table 3.1: Spatial Correlation

	Cell 1	Cell 2	Cell 3	Cell 4	Cell 5	Cell 6	Cell 7
Cell 1	1.000	0.167	0.435	0.130	0.040	0.341	0.307
Cell 2	0.396	1.000	0.338	0.129	0.084	0.310	0.222
Cell 3	0.345	0.541	1.000	0.159	0.162	0.697	0.536
Cell 4	0.437	0.439	0.458	1.000	0.104	0.131	0.114
Cell 5	0.360	0.471	0.492	0.508	1.000	0.163	0.080
Cell 6	0.286	0.491	0.550	0.432	0.535	1.000	0.603
Cell 7	0.284	0.506	0.526	0.459	0.535	0.577	1.000

We examine the correlation among cells for both downlink and uplink data, and present the results among 7 closely located cells in Table 3.1. Each cell is subsequently located on the east side of the previous one. Note that the upper triangular part of Table 3.1 shows the correlation for uplink data, while the lower triangular part is for downlink data. We can clearly observe non-zero correlation among these cells from the table. Actually, more than 50% of the correlation values are greater than 0.300. In addition, we can see that the correlation values among cells vary a lot. For instance, downlink data in Cell 1 and Cell 2 have a correlation value of 0.396; while Cell 5 and Cell 6 have a correlation value of 0.535, even though Cell 2 and Cell 6 are of the same spatial relationship to Cell 1 and Cell 5, respectively. This property indicates that the spatial correlation is highly location-dependent.

## 3.3 Spatiotemporal Modeling and Prediction

### 3.3.1 Overview

As mentioned above, simple temporal modeling that uses only historical data of the target may not work well here due to strong spatial correlation observed from the data. Motivated by the observations described above, we design a novel hybrid deep learning model to perform spatiotemporal modeling and prediction for each cell  $(m, n)$ , which leverages historical data collected from both the target cell and its neighboring cells surrounding it. The proposed model consists of three major components: Local Stacked AutoEncoders (LSAEs), a Global Stacked AutoEncoder (GSAE) and Long Short-Term Memory units (LSTMs). As illustrated by Fig. 3.3, the proposed model works as follows:

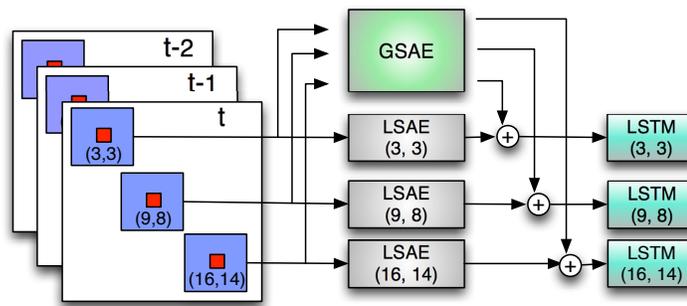


Fig. 3.3: The proposed deep learning model

1) Data of the cell of interest (marked red) and its neighboring cells form a data patch (marked blue), which can include values of one or multiple features of interest (such as downlink/uplink traffic load). The GSAE takes such a data patch as input, producing an encoded representation (called global representation). Note that there is only one GSAE, which is applied to all patches.

2) After being encoded by the GSAE, each patch will be fed to the corresponding LSAE to generate another representation (called local representation). The global representation and local representation will then be concatenated ( $\oplus$ ) to represent each patch.

3) The concatenated representations will then be passed to LSTMs for prediction.

For spatial modeling, we choose autoencoder [11] as a starting point in our design because it has been shown to be a simple and effective model for providing a good representation of input data with much smaller size. We come up with a new hybrid structure based on autoencoders by introducing GSAE and LSAE, whose benefits are explained in Section 3.3.2. However, existing training methods do not work for the proposed hybrid model. Hence, we also present a new training algorithm in Section 3.3.2. Note that one way to select neighboring cells for a target cell is to choose all those surrounding it and falling into a square box as shown in Fig. 3.3. However, the proposed model is not restricted to this method. This can be determined according to actual networks and applications.

In addition, we choose an RNN, particularly LSTM, for temporal modeling and prediction because gated RNNs (such as LSTM), use gates to control how to update hidden states and specify how much past information should be let through, which have been shown to be effective on modeling long-term dependencies [30].

We summarize major notations in Table 3.2

Table 3.2: Major Notations

Notation	Description
$(m, n)$	Index of cell and the corresponding data patch
$t$ and $T$	Index and total number of data points in the temporal domain
$i$ and $I$	Index and total number of GSAE layers
$j$ and $J$	Index and total number of LSAE layers
$\mathbf{W}_{g_i}$ and $\mathbf{W}'_{g_i}$	Weights of encoder and decoder in layer $i$ of GSAE
$\mathbf{W}_{l_j}$ and $\mathbf{W}'_{l_j}$	Weights of encoder and decoder in layer $j$ of LSAE
$\mathbf{b}_{g_i}$ and $\mathbf{b}'_{g_i}$	Biases of encoder and decoder in layer $i$ of GSAE
$\mathbf{b}_{l_j}$ and $\mathbf{b}'_{l_j}$	Biases of encoder and decoder in layer $j$ of LSAE
$\mathbf{h}_{g_i}$ and $\mathbf{h}_{l_j}$	Hidden units in GSAE and LSAE

### 3.3.2 Spatial Modeling

Here, we describe the proposed model for spatial modeling, which is a combination of a GSAE and multiple LSAEs.

An autoencoder is a model (usually a one-hidden-layer neural network) trained to reconstruct its input, which can be used to obtain a different representation (i.e., hidden layer) of the input with a much smaller size [12, 52]. The process to obtain a different representation is referred to as *encoding*, while the process to reconstruct its input is referred to as *decoding*. In our implementation, we adopt the denoising autoencoder, which is an extension of a classic autoencoder [90]. It was designed to make the learned representation robust by reconstructing partially corrupted input. Autoencoders can be stacked to form a deep network [90]. Stacked autoencoders have been shown to be able to effectively extract further non-linear representation [11, 90].

A global representation (i.e., hidden layer of an autoencoder) can be obtained, given the data patch of a cell and a trained GSAE. However, as discussed above, there exists location-dependent spatial correlation for a data patch. Therefore, it is necessary to obtain a better representation with less reconstruction loss. To achieve this goal, we propose to use an LSAE together with the trained GSAE to capture the local location-dependent spatial correlation and yield a better representation.

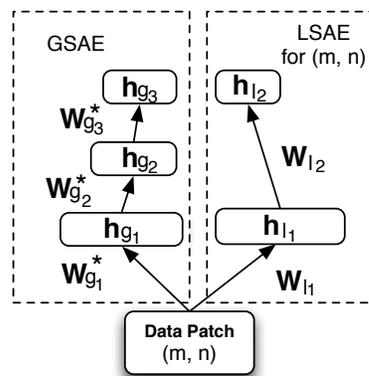


Fig. 3.4: The proposed autoencoder-based spatial model

An example of the LSAE for cell  $(m, n)$  with the GSAE is given in Fig. 3.4. Note that superscript  $(*)$  indicates they are trained variables. So  $\mathbf{W}_{g_i}^*$  are trained weights of layer  $i$  of GSAE. For the sake of readability, notations for bias variables  $\mathbf{b}_{l_j}$ ,  $\mathbf{b}'_{l_j}$ ,  $\mathbf{b}_{g_i}^*$  and  $\mathbf{b}'_{g_i}$  are omitted in both figures. Note that for either GSAE or LSAEs, the layer number can be 1, resulting in a single-layered autoencoder.

Given a trained GSAE, we use an LSAE to further reduce the reconstruction loss of a data patch. The layer 1 weights of the LSAE can be trained to reduce reconstruction loss of layer 1 in the GSAE. Then higher layers of the LSAE are trained to learn a different representation of the lower layers. Finally, the highest representation of the GSAE concatenated by the highest representation of the LSAE generates a better representation of a local data patch.

The proposed hybrid (global + local) model leads to the following benefits:

- *Better Representation*: Different cells share some common characteristics, which are captured by the GSAE. However, as discussed above, each cell also has its specific location-dependent characteristics, which are captured by the corresponding LSAE. Hence, compared to the GSAE-only model, the proposed hybrid model can provide a better presentation for the given data, which has been validated by results presented later.
- *Reduced Model Size*: An SAE with  $H_i$  hidden units in layer  $i$  has  $\sum_{i=1}^I H_{i-1} * H_i$  weight variables (where  $H_0$  is the input dataset size), and  $(H_0 + \sum_{i=1}^{I-1} 2H_i + H_I)$  bias variables. The number of variables will get very large, when the dataset size is big. A large model is usually difficult to train. With the proposed hybrid structure, we have one global, and multiple local SAEs, which both have moderate sizes. Training such models is much easier and faster.
- *Support for Parallel Training*: Given a trained GSAE, training LSAEs is independent of each other. Therefore, they can be trained in parallel.

- *Support for Application-aware Training*: LSAEs can be trained according to the needs of applications. For some applications, we may not be interested in all the cells in the cellular network. If so, we can only train those LSAEs corresponding to cells of interest.

Training the hybrid model in Fig. 3.4 is not straightforward. A well-known work [90] introduced a greedy layer-wise algorithm for effectively training SAE. GSAE can be trained using this algorithm. However, the next step is to train an LSAE with a trained GSAE, for which the existing algorithm [90] cannot be directly applied.

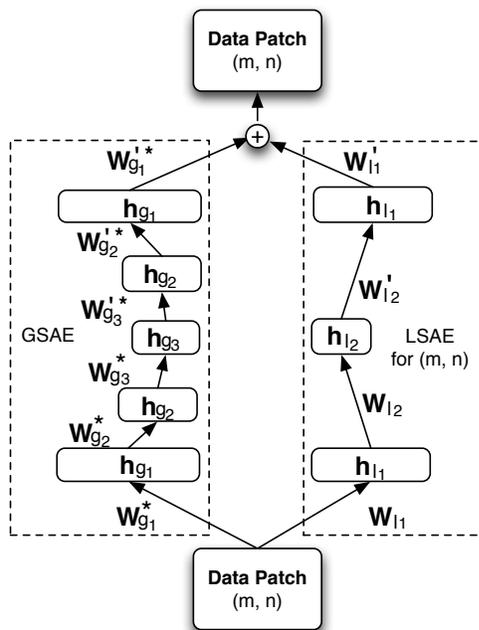


Fig. 3.5: The unrolled GSAE and LSAE

For training and fine-tuning, we need to unroll the GSAE and LSAE, which are shown in Fig. 3.5. Next, we formally define the encoding function  $q_{l_j}(\cdot)$  and decoding function  $q_{l_j}(\cdot)$  for each layer  $i$  of an LSAE.

$$\begin{aligned}
p_{l_j}(\mathbf{X}_{l_j}) &= \mathbf{Y}_{l_j} = \delta(\mathbf{W}_{l_j}\mathbf{X}_{l_j} + \mathbf{b}_{l_j}) \\
q_{l_j}(\mathbf{Y}_{l_j}) &= \begin{cases} \delta(\mathbf{W}'_{g_1}(\mathbf{W}_{g_1}\mathbf{X}_{l_1} + \mathbf{b}_{g_1}) + \mathbf{b}'_{g_1} \\ \quad \quad \quad + \mathbf{W}'_{l_1}\mathbf{Y}_{l_1} + \mathbf{b}'_{l_1}), & j = 1 \\ \delta(\mathbf{W}'_{l_j}\mathbf{Y}_{l_j} + \mathbf{b}'_{l_j}), & \text{otherwise.} \end{cases}
\end{aligned}$$

Here,  $\delta(\cdot)$  is the activation function (we used the sigmoid function in our implementation).  $\mathbf{X}_{l_j}$  is the input of layer  $j$ , which will be encoded.  $\mathbf{Y}_{l_j}$  is the encoded result of layer  $j$ , which can be decoded for reconstruction. However,  $\mathbf{Y}_{l_j}$  can also be encoded by upper layer to obtain a more abstract representation. That is to say,  $\mathbf{X}_{l_{j+1}} = \mathbf{Y}_{l_j}$ .  $\mathbf{W}_{l_j}$ ,  $\mathbf{W}'_{l_j}$ ,  $\mathbf{b}_{l_j}$  and  $\mathbf{b}'_{l_j}$  are the weights for encoding, weights for decoding, bias for encoding and bias for decoding, respectively, in the LSAE.  $\mathbf{W}_{g_1}$ ,  $\mathbf{W}'_{g_1}$ ,  $\mathbf{b}_{g_1}$  and  $\mathbf{b}'_{g_1}$  are the trained weights for encoding, trained weights for decoding, trained bias for encoding and trained bias for decoding, respectively, in the GSAE. Note that  $q_{l_1}(\cdot)$  establishes the connection between the GSAE and LSAE.

We use tied weights [90] for the GSAE and LSAE: the weight matrix in a decoding function is the transpose of the weight matrix in the encoding function, i.e.,  $\mathbf{W}'_{g_i} = \mathbf{W}_{g_i}^T$ ,  $\mathbf{W}'_{l_j} = \mathbf{W}_{l_j}^T$ . Note that if  $\mathbf{W}_{l_1} = \mathbf{0}$  and  $\mathbf{b}'_{l_1} = \mathbf{0}$ , the proposed model degenerates into a GSAE, because in this case,  $q_{l_1}(\mathbf{Y}_{l_1}) = \delta(\mathbf{W}'_{g_1}(\mathbf{W}_{g_1}\mathbf{X}_{l_1} + \mathbf{b}_{g_1}) + \mathbf{b}'_{g_1})$  is actually the reconstructed result of the GSAE. We can initialize  $\mathbf{W}_{l_j} = \mathbf{0}$  and  $\mathbf{b}'_{l_1} = \mathbf{0}$  as the starting point for training an LSAE. We formally present the LSAE training algorithm as Algorithm 3.1, which consists of two phases: pre-training and fine-tuning.

In this algorithm,  $patch(d_{m,n,t})$  gives input data corresponding to Cell  $(m, n)$  and its neighboring cells (surrounding Cell  $(m, n)$ ) at timeslot  $t$ .  $\tilde{\mathbf{X}}_i$  is the corrupted version of  $\mathbf{X}_i$ . Lines 1–3 generate the input data for the first layer in the LSAE. As discussed above, Line 4 initializes  $\mathbf{b}'_{l_1} = \mathbf{0}$  and line 5 initializes the weights  $\mathbf{W}_{l_j} = \mathbf{0}$ . Line 6 pre-trains the

---

**Algorithm 3.1:** Training the LSAE with  $J$  layers for cell  $(m, n)$ 


---

**Input** : Dataset  $\mathbf{D}$ , the trained GSAE with weights  $\mathbf{W}_{g_i}^*$ , bias  $\mathbf{b}_{g_i}^*$ ,  $\mathbf{b}'_{g_i}$   
**Output:**  $\mathbf{W}_{l_j}^*$ ,  $\mathbf{b}_{l_j}^*$ ,  $\mathbf{b}'_{l_j}$

- 1  $\mathbf{X}_{l_1} := \emptyset$ ;
- 2 **forall**  $t$  **do**
- 3    $\mathbf{X}_{l_1} := \mathbf{X}_{l_1} \cup \text{patch}(d_{m,n,t})$ ;
- 4  $\mathbf{b}'_{l_1} = \mathbf{0}$ ;
- 5  $\mathbf{W}_{l_j} := \mathbf{0}, \forall j$ ;
- 6  $\mathbf{W}_{l_1}, \mathbf{b}_{l_1}, \mathbf{b}'_{l_1} := \arg \min_{\mathbf{W}_{l_1}, \mathbf{b}_{l_1}, \mathbf{b}'_{l_1}} L(\mathbf{X}_{l_1}, q_{l_1}(p_{l_1}(\tilde{\mathbf{X}}_{l_1})))$ ;
- 7  $\mathbf{X}_{l_2} := p_{l_1}(\mathbf{X}_{l_1}), j := 2$ ;
- 8 **while** layer  $j \leq J$  **do**
- 9    $\mathbf{W}_{l_j}, \mathbf{b}_{l_j}, \mathbf{b}'_{l_j} := \arg \min_{\mathbf{W}_{l_j}, \mathbf{b}_{l_j}, \mathbf{b}'_{l_j}} L(\mathbf{X}_{l_j}, q_{l_j}(p_{l_j}(\tilde{\mathbf{X}}_{l_j})))$ ;
- 10    $\mathbf{X}_{l_{j+1}} := p_{l_j}(\mathbf{X}_{l_j})$ ;
- 11    $j := j + 1$ ;
- 12 unroll the GSAE and LSAE as in Fig. 3.5;
- 13  $\mathbf{W}_{l_j}^*, \mathbf{b}_{l_j}^*, \mathbf{b}'_{l_j}^* := \arg \min_{\mathbf{W}_{l_j}, \mathbf{b}_{l_j}, \mathbf{b}'_{l_j}} L(\tilde{\mathbf{X}}_{l_1}, \mathbf{X}'), \forall j \in \{1, 2, \dots, J\}$ ;
- 14 **return**  $\mathbf{W}_{l_j}^*, \mathbf{b}_{l_j}^*, \mathbf{b}'_{l_j}^*$ ;

---

first layer with a partially corrupted input. The reconstruction loss is defined to be the cross entropy as in [11, 52]:

$$L(\mathbf{X}, \mathbf{Z}) = \sum x \log(z) + (1 - x) \log(1 - z). \quad (3.4)$$

In our implementation, Stochastic Gradient Descent (SGD) [45] algorithm is applied to minimize the reconstruction loss. Other methods, such as RMSProp and AdaGrad [45], can also be applied here to train the model. Line 7 generates the input  $\mathbf{X}_{l_2}$  for the second layer with uncorrupted  $\mathbf{X}_{l_1}$ . Lines 8–11 show the pre-training process for layer 2 up to layer  $J$ . After layer  $j$  is pre-trained, the input  $\mathbf{X}_{l_{j+1}}$  for  $(j + 1)$ -th layer can be obtained from  $\mathbf{X}_{l_{j+1}} = p_{l_j}(\mathbf{X}_{l_j})$ . Note that the uncorrupted input  $\mathbf{X}_{l_j}$  is fed to the encoder. After all the layers have been pre-trained, we unroll the trained GSAE and LSAE as shown in Fig. 3.5 for fine-tuning (Line 13), where all weight matrices and bias variables are updated.  $\mathbf{X}'$  is the reconstructed input.

An LSAE cannot be trained without the GSAE because the decoding function of LSAE

relies on the trained GSAE. The first layer of the LSAE is pre-trained and fine-tuned differently from other (upper) layers, which takes the trained layer 1 of GSAE as input. Given the pre-trained and fine-tuned layer 1, layers 2 to  $J$  of the LSAE are pre-trained and fine-tuned independently from the GSAE. Note that, given a trained GSAE, all LSAEs can be trained in parallel. In addition, it is not required to have the same structure for the GSAE and LSAE: GSAE and LSAEs can have different numbers of layers; and the number of hidden units in each layer can also be set differently. Moreover, the structures of LSAEs do not have to be the same.

### 3.3.3 Temporal Modeling and Prediction

As mentioned above, we propose to use an RNN for temporal modeling and prediction, which takes the representations learned from the hybrid spatial model as input.

An RNN is a generalization of the feed forward neural network for modeling sequence (time series) data [71]. However, a well-known problem with standard RNNs is that it can be difficult to model long-term dependencies [38]. Long Short-Term Memory (LSTM) was proposed in [37], which is known to be able to capture long-term temporal dependencies [30, 31]. LSTM incorporates gates, which allow the model to learn how to forget previous hidden states and how to update the current states. A diagram of the LSTM unit from [102] is shown in Fig. 3.6, which is a slight simplification of [32].

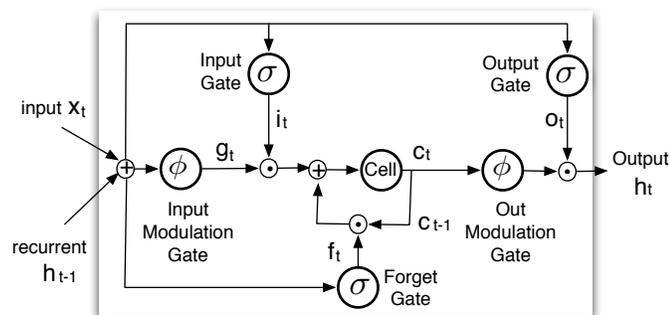


Fig. 3.6: LSTM unit [102]

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o) \\
\mathbf{g}_t &= \phi(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \phi(\mathbf{c}_t)
\end{aligned}$$

The LSTM unit consists of a single memory cell  $\mathbf{c}_t$ , an input and output modulation gate ( $\mathbf{g}_t$  and  $\mathbf{h}_t$ ) and three gates (input  $\mathbf{i}_t$ , output  $\mathbf{o}_t$  and forget  $\mathbf{f}_t$ ).  $\sigma(\cdot)$  is the sigmoid function; and  $\phi(x)$  is the hyperbolic tangent function  $\phi(x) = 2\sigma(2x) - 1$ .  $\odot$  and  $\oplus$  denote the dot product and sum of two vectors, respectively. The  $\mathbf{W}$  terms denote the weight matrices. For example,  $\mathbf{W}_{hf}$  is the hidden-forget weight matrix; while the  $\mathbf{b}$  terms are the biases.

The memory cell combines the previous cell states, current input and previous output, to update hidden states. The forget gate determines if the information should be forgotten or remembered. The output gate learns how the memory cell should affect the hidden states.

To predict the future value  $d_{m,n,t'+1}$  for a cell (m,n), the data patches corresponding to the past  $T$  timeslots are taken as the input. They will be encoded by the GSAE and the LSAE. For each timeslot  $t \leq t'$ , the following three values will be concatenated as a vector:  $d_{m,n,t}$ , and GSAE and LSAE representations of  $patch(d_{m,n,t})$ . In this way, we obtain a temporal sequence of vectors, as shown in Fig. 3.7. Then the LSTM unit processes this sequence as described above and predicts  $d_{m,n,t'+1}$ .

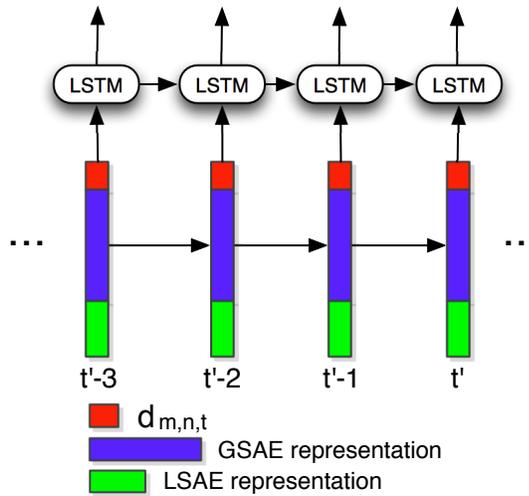


Fig. 3.7: Temporal modeling and prediction

## 3.4 Performance Evaluation

### 3.4.1 Settings

We compared our approach with two widely used methods for time series analysis. The first approach is ARIMA [39], which is one of the most popular linear models for time series forecasting and has been applied to wireless networks [92, 107]. The second baseline approach is SVR, which is a variant of Support Vector Machine (SVM) proposed for regression [21, 84]. It has been also applied for time series analysis in many applications [60, 72]. In the experiments, we used the implementation of ARIMA and SVR in two libraries [45] and [73], respectively. These two baselines were compared with the proposed model in terms of three commonly used performance metrics [26]: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Log Loss (also known as binary cross-entropy).

For neighboring cell selection in spatial modeling, we chose to use data from all cells located within a  $11 \times 11$  square box that is centered at the location of a target cell. That is to say, we considered data from 120 neighboring cells for modeling Cell  $(m, n)$ .

We chose the commonly used sigmoid function as the activation function in each layer

of both the GSAE and LSAEs. Regarding the corruption process in autoencoders, we adopted a stochastic method proposed in [90]. In our implementation, the corruption level was set to 0.1. The GSAE has two layers (unrolled), and the lower layer has 20 hidden units, and second layer has 2 hidden units. All the LSAEs has a single layer with 2 hidden units.

We randomly chose 15 cells for testing. For each cell  $(m, n)$ , we split the data into training set and test set. We presented the corresponding results in the following.

### 3.4.2 Prediction Results

First, we present the experimental results to show the overall prediction performance of the proposed model.

Fig. 3.8 shows a comparison between prediction results and the actual values (from the dataset) for both downlink and uplink traffic load at a randomly chosen cell. We can see that the prediction results well match the trend of actual values. Specifically, the MSE, MAE and Log Loss are 0.042, 0.165 and 0.583, respectively for downlink traffic load; while, they become 0.031, 0.137, 0.556 respectively for uplink. Moreover, prediction results are very close to the actual values around the major transition points, when the traffic load falls below or rises above 0.4.

Fig. 3.9 shows a comparison among ARIMA, SVR and the proposed model for both downlink and uplink traffic load in terms of MSE, MAE and Log Loss for one of the chosen locations, while Fig. 3.10 presents the average errors over all the chosen locations. From these two figures, we can see that the proposed model consistently outperforms ARIMA and SVR in terms of all the metrics. Specifically, in Fig. 3.10, the proposed model offers about 30.8%, 20.5%, 33.1% less error than SVR on average in terms of MSE, MAE and Log Loss, respectively. Moreover, it leads to around 40.4%, 28.4%, 18.5% less error than ARIMA on average in terms of MSE, MAE and Log Loss, respectively. These results well justify effectiveness of the emerging deep learning models on cellular network data

analysis and more importantly, the superiority of our design that takes into account data dependencies in both the temporal and spatial domains.

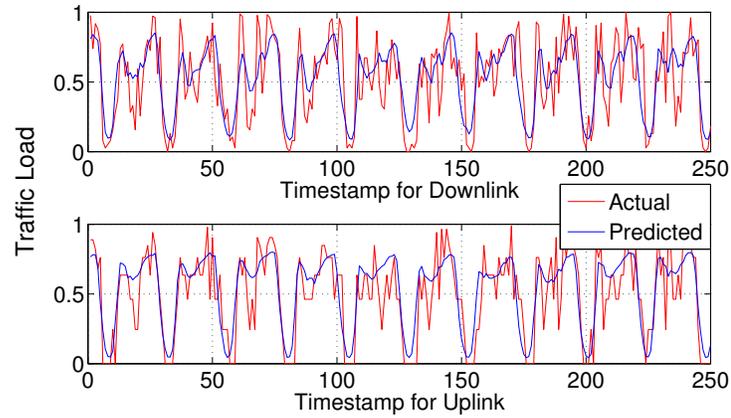


Fig. 3.8: Prediction results VS. actual values

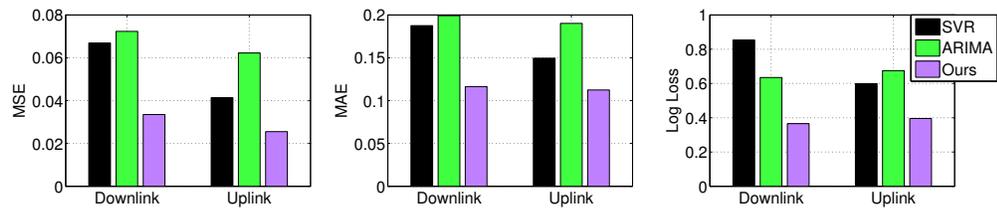


Fig. 3.9: Prediction errors for a randomly chosen cell

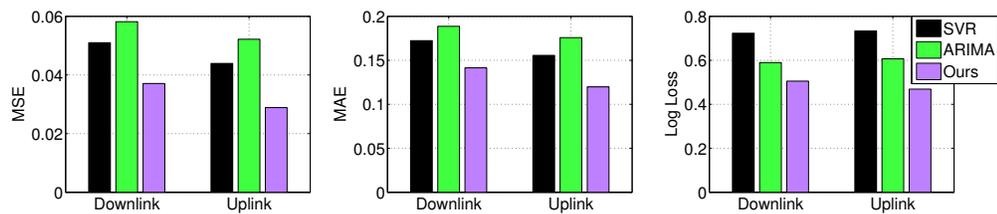


Fig. 3.10: Average prediction errors

### 3.4.3 Spatial Modeling

In this subsection, we present the results to justify the effectiveness of the proposed hybrid model for spatial modeling approach.

Data patches are encoded by both the GSAE and LSAEs. The decoders can reconstruct them so that we can take an in-depth look to make sure the encoded results are indeed good representations of the original data. Fig. 3.11 (downlink) and Fig. 3.12 (uplink) show the reconstructed results of data patches corresponding to 9 cells. Each image corresponds to a data patch; and each tiny block (i.e., pixel) in an image corresponds to a cell. Hence, each image has  $11 * 11$  blocks. Brightness of a pixel indicates how heavy the traffic load of the corresponding cell is (the brighter, the heavier).

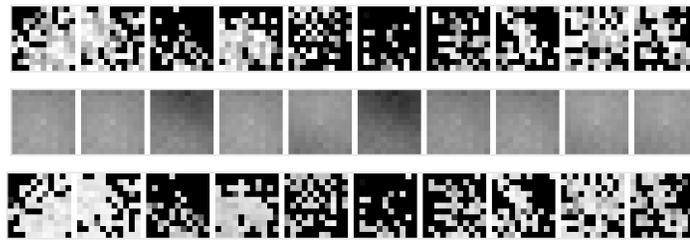


Fig. 3.11: Reconstructed downlink traffic load

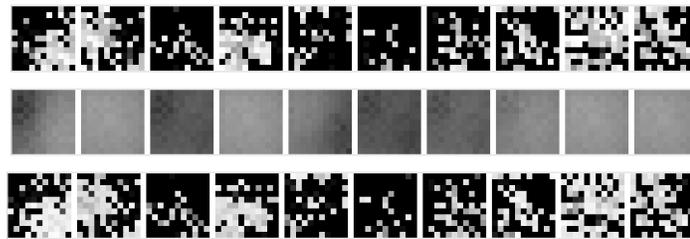


Fig. 3.12: Reconstructed uplink traffic load

In both figures, the first rows are the original data patches of the 9 randomly chosen cells; The second rows are the corresponding patches reconstructed by the GSAE. The last rows are data patches reconstructed by The proposed hybrid model (GSAE+LSAE). Note that multiple LSAEs were trained since the original data patches came from different cells.

From these two figures, we can see that the reconstructed results given by the GSAE is relatively “blurry” but somehow still captures the patterns of the original data; while the final reconstructed results given by the proposed hybrid model are very close to the original data. These results confirm that the proposed hybrid model does offer good representations for the original data.

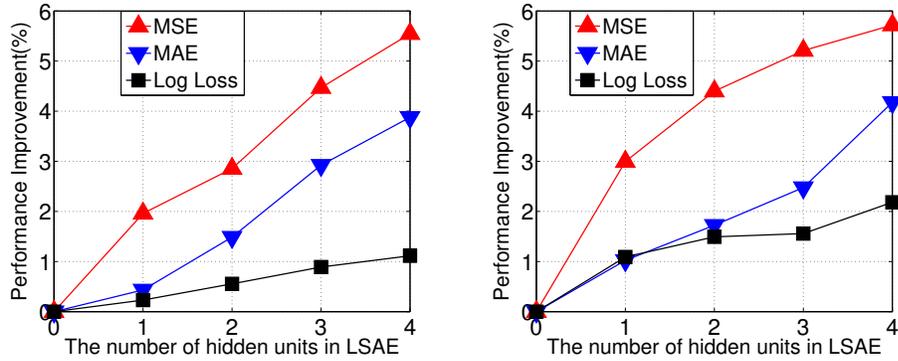


Fig. 3.13: Prediction performance improvement (Left: Downlink, Right: Uplink)

Now we show how LSAEs can help improve the prediction performance. In this experiment, the number of hidden units of our single-layered LSAEs,  $k$ , was changed from 0 to 4, with 0 corresponding to the case without LSAEs. The second layer of GSAE was then set to have  $(4 - k)$  hidden units. All other settings remain the same. In this way, even though the value of  $k$  is changed, the GSAE and a LSAE together had a representation with a constant length of 4, which ensures a fair comparison. Fig. 3.13 shows how the performance improvement ratio changes with  $k$ , which is defined as follows:

$$\frac{M(0) - M(k)}{M(0)} * 100\%,$$

where  $M(k)$  denotes the prediction error (MSE, MAE or Log Loss) corresponding to  $k$  ( $M(0)$  then corresponds to the case without LSAEs).

From Fig. 3.13, we can see that the prediction performance improvement rises monotonically with  $k$ . Specifically, in terms of MSE for downlink, the improvement ratio goes up from 1.96% to 5.54%, when  $n$  increases from 1 to 4. This observation validates our claim that learning local characteristics is essential and learning more helps improve prediction performance. However, the tradeoff is that more complicated local models may lead to much longer training time. Determining the best configurations for the GSAE and LSAEs is task dependent. It depends on the nature of input data, available computing resources and the number of cells of interest.

### 3.5 Summary

In this chapter, we first performed a preliminary analysis for a real dataset from China Mobile to show temporal and spatial dependencies. Then we presented a hybrid deep learning model for spatiotemporal prediction, which includes a novel autoencoder-based deep model for spatial modeling and LSTMs for temporal modeling. The autoencoder-based model consists of a GSAE and multiple LSAEs, which can offer better representations for input data (compared to the GSAE-only model), reduced model size, and support for parallel and application-aware training. Moreover, we presented a new algorithm for training the proposed spatial model. The experimental results show that, compared to ARIMA and SVR, the proposed deep model significantly improves prediction accuracy; and the autoencoder-based spatial model is effective and efficient.

# CHAPTER 4

## QUALITY-AWARE AND FINE-GRAINED INCENTIVE MECHANISMS IN MOBILE CROWDSENSING SYSTEMS

### 4.1 Overview

Mobile wireless networks have become an essential part in wireless networking with the prevalence of mobile device usage. However, some mobile users haven't realized that their smartphones have powerful sensing capabilities. Most smartphones are equipped with various embedded sensors, including microphone, camera, GPS, accelerometer, gyroscope, WiFi/3G/4G interfaces, etc. Moreover, booming wearable devices (such as Google Glass, Smart Watches, Fitbit, Sensordrone [75], etc.) can be connected to smartphones via network interfaces, such as Bluetooth, to extend their sensing capabilities. Embedded sensors and wearable devices can enable applications and services in various domains, such as environmental monitoring, social networking, healthcare, transportation and safety.

Recently, Mobile CrowdSensing (MCS) have been gaining increasing popularity. We consider a general-purpose MCS system [79], as shown in Fig. 1.1. While participating in

MCS, there is usually a *cost* occurring to a mobile user. For example, performing sensing activities consumes energy from a smartphone. So the mobile user may want to earn certain credits (e.g., money) to compensate for his/her energy loss. Most sensing tasks are location-dependent, which may require mobile users to travel to or around certain areas, leading to certain costs such as transportation. Furthermore, mobile users usually won't be willing to share their privacy while undertaking sensing tasks if there are no satisfactory rewards.

Therefore, we are motivated to consider a *reverse auction* based incentive mechanism to enable fair pricing between the cloud operator and mobile users in MCS. As illustrated in Fig. 1.1, after receiving a sensing task from a service user, the cloud operator (the buyer of sensor data) announces it to mobile users. Mobile users (bidders, sellers and service providers) offer their bids for undertaking the task and selling their sensor data. Based on the bids, the cloud operator will selectively determine winners and after collecting sensor data from winners, it will make payments to them. Auction mechanism design is crucial for supporting MCS, because the trading rules between the buyer (the cloud operator) and the sellers (mobile users) heavily depend on it. Specifically, among all the behavior characteristics of bidders, *truthfulness* [65] and *individual rationality* [50] are of special interest and most desirable in MCS. An auction mechanism is truthful if a bidder will not increase its payoff by submitting any other bids instead of his/her true values. An auction without truthfulness will be vulnerable to market manipulation and produce very poor outcomes [44]. An auction mechanism is individually rational if the payoff of every bidder is not negative by bidding his/her true values.

We aim to develop mathematical models to characterize the quality of a recruited crowd (a set of mobile users). We believe the models for Quality of Crowd (QoC) should be application-dependent and we introduce several such models to serve various applications. Furthermore, Unlike [22, 97], in our auction formulation, the bids are two-dimensional, which means the proof of mechanism properties in [22, 97] cannot be directly applied here; and we follow the Bayesian setting [65] (See Section 4.3), which is a more realistic

model.

We consider *fine-grained* MCS, in which each sensing task consists of multiple subtasks and a mobile user may make contributions to multiple subtasks. For example, if the goal of a sensing task is to cover a target area, then each subtask may correspond to a sub-area. In this way, the recruited crowd may provide a better coverage for the target area. In addition, a sensing task may even include a set of heterogeneous subtasks. For example, subtask 1 may request the sensing crowd to collect WiFi signal strengths, while subtask 2 may request for signal strengths of cellular networks. Fine-grained MCS can lead to a better quality of service and allow a service user to specify a sensing task more flexibly. However, it also introduces additional complexity for crowd selection because a mobile user may be a good candidate for multiple subtasks, but may contribute differently to different subtasks. Existing incentive mechanisms [49] select the crowd for a single task, ignoring benefits that can be brought by sharing service/data with other tasks/subtasks. However, we aim to select a crowd to undertake a sensing task, while meeting a certain quality requirement (explained in Section 4.2) for each of its subtasks. We summarize our contributions in the following:

- We introduce mathematical models for characterizing QoC for different sensing applications.
- Based on these models, we present a novel auction formulation for quality-aware and fine-grained MCS, which minimizes the expected expenditure subject to the quality requirement of each subtask.
- We discuss how to achieve the optimal expected expenditure, and present a practical incentive mechanism to solve the auction problem, which is shown to be truthful, individually rational and computationally efficient.
- We conducted trace-driven simulation using the mobility dataset of San Francisco taxis [67] and compared the proposed incentive mechanism with two well-designed

baseline methods (rather than trivial random solutions). Extensive simulation results show the proposed mechanism achieves noticeable expenditure savings compared to the baselines; moreover, it produces close-to-optimal solutions.

## 4.2 Quality of Crowd (QoC) Models

We summarize major notations in Table 4.1.

Table 4.1: Major Notations

Notation	Explanation
$i$ and $M$	Index of mobile users and the total number of mobile users
$c_i$ and $w_i$	True and declared costs of mobile user $i$ respectively
$\mathbf{Y}_i$ and $\mathbf{Z}_i$	True and declared quality score vectors of mobile user $i$ respectively
$\mathbf{b}_i$ and $\mathbf{B}$	Bid of mobile user $i$ and the corresponding vector
$x_i$ and $\mathbf{x}$	Winner selection variable of mobile user $i$ and the corresponding vector
$p_i$ and $\mathbf{p}$	Payment to mobile user $i$ and the corresponding vector
$j$ and $N$	Index of subtasks and the total number of subtasks
$r_j$ and $\mathbf{R}$	Quality requirement of subtask $j$ and the corresponding vector
$g_j(\cdot)$	QoC model of subtask $j$

We focus on a general-purpose MCS system with a sensing crowd of  $M$  mobile users. A subset of mobile users will be recruited to undertake a sensing task including  $N$  subtasks. For each selected mobile user, there is a cost of  $c_i$  as explained above. A *quality score* is given for mobile user  $i$  participating in subtask  $j$  (denoted as  $y_{ij}$ ), which quantifies the quality of services/data the mobile user is potentially capable of providing to that subtask. It is application-dependent and can be assigned according to various factors such as availability, accuracy of sensor data, reputation, etc. The cloud operator can calculate quality scores for mobile users and let them know their own quality scores. We use  $\mathbf{Y}_i = [y_{i1}, \dots, y_{ij}, \dots, y_{iN}]$  to denote the quality score vector of mobile user  $i$  for all sensing subtasks.

The Quality of Crowd (QoC) for subtask  $j$ ,  $q_j$ , quantifies the quality of services/data the sensing crowd is potentially capable of providing, which could be given by a function  $g_j(\cdot)$  that can satisfy the following properties: 1)  $g_j(\cdot)$  is a monotonically non-decreasing function of  $\langle y_{ij} \rangle$  and  $\langle x_i \rangle$ , where  $x_i$  is a binary value indicating whether mobile user  $i$  is recruited or not; and 2)  $g_j(\cdot)$  returns a value in  $[0, 1]$ . The first property reflects the nature that with a larger population of the recruited crowd and/or higher individual quality scores, the QoC for the corresponding subtask should not become worse. In order to make it easier for comparisons and understanding, the value of QoC should be scaled into  $[0, 1]$ , with 1 indicating the corresponding subtask can be perfectly completed by the recruited crowd. Note that the auction-based incentive mechanisms presented later are not restricted to any particular QoC model (function). In the following, we introduce several QoC models that can cover a large variety of sensing applications.

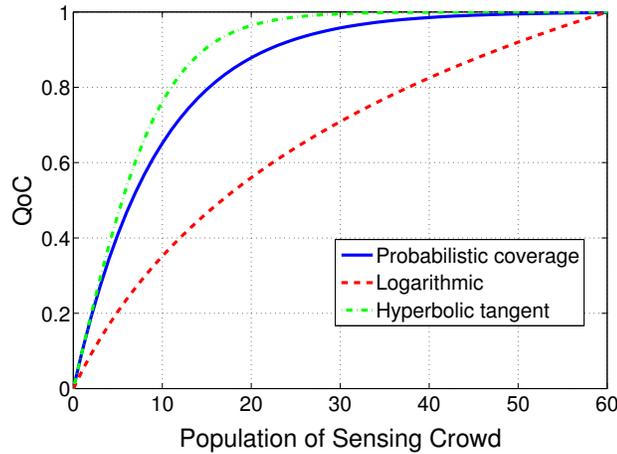


Fig. 4.1: QoC models

1) Linear model:

$$q_j = \frac{\min(\sum_i^M y_{ij} x_i, q_{\max})}{q_{\max}}. \quad (4.1)$$

This model simply sums up quality scores of all mobile users as the QoC if a goal  $q_{\max}$  has not yet achieved; otherwise, the QoC remains at  $q_{\max}$ . This model is suitable for applications with a goal/constraint of achieving a certain sensing duration or collecting a certain

number of samples. Here,  $y_{ij}$  can be the sensing duration or the number of sensing samples that mobile user  $i$  can potentially provide for subtask  $j$ . Linear models have been used in [36, 43].

2) Probabilistic coverage model:

$$q_j = 1 - \prod_i^M (1 - y_{ij}x_i). \quad (4.2)$$

If  $y_{ij}$  gives the probability that the target of subtask  $j$  (e.g., an area or a set of points of interest) can be covered by recruiting mobile user  $i$ , then  $q_j$  is the probability that the target can be covered by the recruited crowd. This model is suitable for sensing applications with a goal/constraint of covering a target area or a set of target points.

3) Logarithmic model:

$$q_j = \frac{\log(1 + \sum_i^M \log(1 + y_{ij}x_i))}{\log(1 + \sum_i^M \log(1 + y_{ij}))}. \quad (4.3)$$

In the numerator, the inner  $\log$  term causes the return value to have a diminishing increment with the quality score, and the outer  $\log$  term leads to diminishing increment with the population of the recruited crowd.

4) Hyperbolic tangent model:

$$q_j = \tanh\left(\sum_i^M y_{ij}x_i\right). \quad (4.4)$$

Note that it has been shown by [97] that function (4.3) is submodular, i.e., the increase of the return value diminishes with the input set. We can easily show that function (4.4) is submodular too. These two models are suitable for most applications which extract meaningful information from sensor data, because usually given a larger data set, the additional information that can be obtained diminishes. Fig. 4.1 illustrates how QoC changes with the population of the crowd according to the three non-linear models. In this example, all

mobile users have a common quality score of 0.1.

As mentioned above, we consider fine-grained MCS, in which each sensing task consists of multiple subtasks. Each subtask needs to be completed with a minimum quality requirement,  $r_j$ . We use  $\mathbf{R} = [r_1, \dots, r_j, \dots, r_N]$  to denote a vector of quality requirements of all subtasks. The cloud operator recruits mobile users and makes sure  $q_j = g_j(\mathbf{Y}, \mathbf{X}) \geq r_j, \forall j \in \{1, \dots, N\}$ , for the given sensing task, where  $\mathbf{X} = [x_1, \dots, x_i, \dots, x_M]$ .

### 4.3 Auction Formulation

In MCS, incentive mechanism design can be formulated as a *reverse* or *procurement* auction mechanism design problem. In the auction, 1) the cloud operator (the buyer) announces a sensing task to mobile users (bidders and sellers); 2) each mobile user  $i$  submits a bid  $\mathbf{b}_i$  (defined below); 3) the cloud operator uses an incentive mechanism to select the winners and determine payments; 4) winners carry out the sensing task and submit results to the cloud operator; 5) the cloud operator checks the results and makes payments to winners. In the following, we use *mobile user* and *bidder* interchangeably.

Specifically,  $\mathbf{b}_i = (w_i, \mathbf{Z}_i)$ , in which  $w_i$  is mobile user  $i$ 's declared cost, and  $\mathbf{Z}_i$  is mobile user  $i$ 's declared quality vector. If mobile user  $i$  does not want to participate in certain subtasks, the corresponding declared quality scores can be set to 0. Because mobile user  $i$ 's true cost  $c_i$  and true quality vector  $\mathbf{Y}_i$  are private and only known to mobile user  $i$  himself/herself,  $w_i$  and  $\mathbf{Z}_i$  could be different from  $c_i$  and  $\mathbf{Y}_i$ , respectively. Different from [1], mobile users' private costs,  $\langle c \rangle$ , are assumed to follow a known distribution here. This assumption is known as *Bayesian setting* [65], and it is a realistic assumption because such a distribution can be obtained from historical data of previous auction transactions.  $f_i(c)$  denotes the probability density function; and  $F_i(c)$  denotes the corresponding cumulative distribution function. So  $f_i(c) = \frac{d}{dc} F_i(c)$ .  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_i, \dots, \mathbf{b}_M]$  is the bid vector of all mobile users.  $\mathbf{B}_{-i}$  denotes the bids of all mobile users except  $i$ , so  $\mathbf{B} = [\mathbf{b}_i, \mathbf{B}_{-i}]$ . In

addition, each mobile user  $i$  is a *single-minded bidder* [65], i.e., at a cost of  $c_i$ , he/she will participate in those subtasks, to which he/she has non-zero quality scores; or none at a cost of 0 otherwise.

The cloud operator must complete each subtask  $j$  to the required quality  $r_j$ ,  $\forall j \in \{1, \dots, N\}$ . Moreover, the cloud operator also wishes to conserve money and minimize its expected expenditure by selectively recruiting mobile users. Specifically, an auction mechanism takes the bid vector  $\mathbf{B}$  and the quality requirement vector  $\mathbf{R}$  as input and returns a winner vector  $\mathbf{x} = [x_1, \dots, x_i, \dots, x_M]$ , where  $x_i = 1$  if mobile user  $i$  wins, and  $x_i = 0$  otherwise; it also returns a payment vector  $\mathbf{P} = [p_1, \dots, p_i, \dots, p_M]$ , where  $p_i$  is the payment that the cloud operator will make to mobile user  $i$ . Based on the output of the auction, the *payoff* of mobile user  $i$  is defined as

$$u_i = \begin{cases} p_i - c_i, & x_i = 1; \\ 0, & x_i = 0. \end{cases} \quad (4.5)$$

The *expenditure* of a reverse auction is the sum of the payments  $\sum_i^M p_i$  to all mobile users (bidders).

### 4.3.1 Desirable Properties

In this section, we describe three desirable properties for an auction mechanism:

- **Individual Rationality:** an auction mechanism is *individually rational* if for any bidder  $i$ , the payoff is non-negative when bidder  $i$  bids his/her true value  $(c_i, \mathbf{Y}_i)$ .
- **Truthfulness:** an auction mechanism is *truthful* if and only if for every bidder  $i$  and  $\mathbf{B}_{-i}$ , bidder  $i$  will not increase his/her payoff by making a bid  $(w_i, \mathbf{Z}_i)$  that is different from his/her true value  $(c_i, \mathbf{Y}_i)$ ; i.e., bidder  $i$ 's payoff for bidding  $(c_i, \mathbf{Y}_i)$  is at least his/her payoff for bidding any other bid  $(w_i, \mathbf{Z}_i)$ .

- **Computational Efficiency:** an auction mechanism is *computationally efficient* if the outcome can be computed in polynomial time.

Of the three properties, truthfulness is the most difficult to achieve. The bid is two-dimensional because for bidder  $i$ , the bid  $\mathbf{b}_i$  contains two parts: bidder  $i$ 's declared cost  $w_i$  and bidder  $i$ 's declared quality vector  $\mathbf{Z}_i$ . As a result, Myerson's theorem [61] about the properties of one-parameter truthful mechanisms cannot be directly applied. To design a truthful auction mechanism with two dimensions, we introduce the following definitions:

**Definition 4.1 ( $w$ -Monotonicity).** *If bidder  $i$  wins by bidding  $(w_i^*, (z_{i1}^*, \dots, z_{ij}^*, \dots, z_{iN}^*))$ , then he/she also wins by bidding  $(w_i', (z_{i1}^*, \dots, z_{ij}^*, \dots, z_{iN}^*))$  with any  $w_i' \leq w_i^*$ .*

**Definition 4.2 ( $z$ -Monotonicity).** *If bidder  $i$  wins by bidding  $(w_i^*, (z_{i1}^*, \dots, z_{ij}^*, \dots, z_{iN}^*))$ , then he/she also wins by bidding  $(w_i^*, (z'_{i1}, \dots, z'_{ij}, \dots, z'_{iN}))$  with all  $z'_{ij} \geq z_{ij}^*$ .*

**Definition 4.3 (Critical Payment).** *The payment  $p_i$  for winning bidder  $i$  is set to the critical value  $d_i$  such that bidder  $i$  wins if  $w_i < d_i$ , and loses if  $w_i > d_i$ .*

**Lemma 4.1.** *In an MCS auction mechanism, if  $w$ -Monotonicity,  $z$ -Monotonicity, and critical payment are satisfied, bidder  $i$  will not increase his/her payoff by bidding  $(c_i, \mathbf{Z}_i) = (c_i, (z_{i1}, \dots, z_{ij}, \dots, z_{iN}))$  instead of  $(c_i, \mathbf{Y}_i) = (c_i, (y_{i1}, \dots, y_{ij}, \dots, y_{iN}))$ , when  $\mathbf{Y}_j \neq \mathbf{Z}_j$ .*

*Proof.* We examine two possible scenarios:

1)  $z_{ij} < y_{ij}$  for every  $j$ . Let  $d_y, d_z$  denote the critical payments for bidding  $(c_i, \mathbf{Y}_i)$  and  $(c_i, \mathbf{Z}_i)$  respectively. We consider two sub-cases: a) bidder  $i$  wins by bidding  $(c_i, \mathbf{Z}_i)$ . Based on  $z$ -Monotonicity, we know that he/she will also win by bidding  $(c_i, \mathbf{Y}_i)$ . In other words, for any  $c_i < d_z$ , we have  $c_i < d_y$ . Hence,  $d_y \geq d_z$ ; the payment of bidding  $(c_i, \mathbf{Y}_i)$  will not be decreased. b) bidder  $i$  loses by bidding  $(c_i, \mathbf{Z}_i)$ . In this sub-case, the payoff of bidding  $(c_i, \mathbf{Y}_i)$  is 0 if he/she loses and non-negative if he/she wins.

2)  $z_{ij} > y_{ij}$  for one or more  $j$ 's. Before actually making payments to bidder  $i$ , the cloud operator has a quality control that makes sure the actual quality score  $y_{ij}$  (derived from the

submitted sensor data) is equal to or greater than  $z_{ij}$ . If not, no payments will be made to bidder  $i$ , yielding negative payoff for him/her with bidding  $(c_i, \mathbf{Z}_i)$ .

The above two cases complete the proof.  $\square$

**Lemma 4.2.** *In an MCS auction mechanism, if  $w$ -Monotonicity,  $z$ -Monotonicity, and critical payment are satisfied, bidder  $i$  will not increase his/her payoff by bidding  $(w_i, \mathbf{Z}_i)$  instead of  $(c_i, \mathbf{Z}_i)$ , when  $c_i \neq w_i$ .*

*Proof.* Denote the Critical Payment for bidding  $(c_i, \mathbf{Z}_i)$  by  $d$ . We consider two cases:

1)  $(c_i, \mathbf{Z}_i)$  is a losing bid. In this case,  $c_i > d$ . We consider two sub-cases: a)  $(w_i, \mathbf{Z}_i)$  is a losing bid. Bidder  $i$  would have a 0 payoff, which is not better than bidding  $(c_i, \mathbf{Z}_i)$ . b)  $(w_i, \mathbf{Z}_i)$  is a winning bid. He/she receives the payment  $d$  because the critical payment is independent of  $w_i$ ; the payoff of bidding  $(w_i, \mathbf{Z}_i)$  would be negative, since  $d < c_i$ .

2)  $(c_i, \mathbf{Z}_i)$  is a winning bid. If  $(w_i, \mathbf{Z}_i)$  is a winning bid, bidder  $i$  receives the same payment  $p$  with  $(c_i, \mathbf{Z}_i)$ . If  $(w_i, \mathbf{Z}_i)$  is a losing bid, he/she receives a payment of 0.

The above two cases complete the proof.  $\square$

**Theorem 4.1.** *An auction mechanism for MCS is truthful if it satisfies  $w$ -Monotonicity,  $z$ -Monotonicity and critical payment.*

*Proof.* Based on the definition of truthfulness, it suffices to show that bidder  $i$  will not increase his/her payoff by bidding any other bid  $(w_i, \mathbf{Z}_i)$  instead of  $(c_i, \mathbf{Y}_i)$ . Lemma 4.2 has shown that bidder  $i$  will not increase his/her payoff by bidding  $(w_i, \mathbf{Z}_i)$  instead of  $(c_i, \mathbf{Z}_i)$ . In Lemma 4.1, we have proved that bidder  $i$  will not increase his/her payoff by bidding  $(c_i, \mathbf{Z}_i)$  instead of  $(c_i, \mathbf{Y}_i)$ . Therefore, bidder  $i$  will not increase his/her payoff by bidding any  $(w_i, \mathbf{Z}_i)$  instead of  $(c_i, \mathbf{Y}_i)$ . This completes the proof.  $\square$

### 4.3.2 Virtual Cost

Next, we introduce *virtual cost* for reverse auctions and show its relationship with the expected expenditure. The concept of *virtual valuation* has been introduced for *forward*

auctions in [61].

**Definition 4.4** (Virtual Valuation). *In a forward auction, the virtual valuation of bidder  $i$  with valuation  $v_i$  is*

$$\phi_i(v_i) = v_i - \frac{1 - F_i(v_i)}{f_i(v_i)}, \quad (4.6)$$

where the hazard rate  $\frac{f_i(v_i)}{1 - F_i(v_i)}$  is assumed to be monotonically non-decreasing (regularity assumption).

**Theorem 4.2** ([65]). *Consider any (forward) truthful mechanism and fix the bids  $\mathbf{b}_{-i}$  of all bidders except for bidder  $i$ . The expected payment of bidder  $i$  satisfies:*

$$E[p_i(v_i)] = E[\phi_i(v_i)x_i(v_i)]. \quad (4.7)$$

However, in a reverse auction, the valuation of a bidder can be treated as the negative of its cost, i.e.,  $v_i = -c_i$ . Therefore,

$$\phi_i(v_i) = -c_i - \frac{1 - F_i(v_i)}{f_i(v_i)}$$

Moreover, it can be easily derived that

$$F_i(v_i) = 1 - F_i(c_i), \quad f_i(v_i) = f_i(-c_i) = f_i(c_i)$$

Hence, we have

$$\phi_i(v_i) = -\left(c_i + \frac{F_i(c_i)}{f_i(c_i)}\right)$$

**Definition 4.5** (Virtual Cost). *In a reverse auction, the virtual cost of bidder  $i$  with cost  $c_i$*

is

$$\beta_i(c_i) = c_i + \frac{F_i(c_i)}{f_i(c_i)} \quad (4.8)$$

where the *regularity assumption* requires that  $\frac{f_i(c_i)}{F_i(c_i)}$  is monotonically non-increasing. It is clear that  $\beta_i(c_i) = -\phi_i(v_i)$ .

**Theorem 4.3.** *Consider any reverse truthful mechanism and fix the bids  $\mathbf{b}_{-i}$  of all bidders except for bidder  $i$ . The expected payment to bidder  $i$  satisfies:*

$$E[p_i(c_i)] = E[\beta_i(c_i)x_i(c_i)] \quad (4.9)$$

*Proof.* The payment from the buyer to a seller in a reverse auction can be viewed as the negative of the payment from a buyer to the seller in a forward auction. Therefore:

$$E[p_i(c_i)] = E[-p_i(v_i)] = E[-\phi_i(v_i)x_i(v_i)] = E[\beta_i(c_i)x_i(c_i)]$$

□

Because of Theorem 4.3 and the independence of all bidders' costs, it is fairly easy to show that the expected expenditure of a reverse truthful mechanism is equal to the total virtual cost. Therefore, to minimize the expected expenditure, it suffices to minimize the total virtual cost  $\sum_i^M \beta_i(c_i)x_i$ .

## 4.4 Quality-aware Incentive Mechanisms (QIMs)

In this section, we present quality-aware incentive mechanisms (QIMs). First, we discuss how to achieve the optimal expected expenditure. Then, we present a practical QIM that is computationally efficient.

### 4.4.1 Optimal Solutions

The QIM design problem consists of two subproblems: *Winner Selection* and *Payment Determination*. Winner selection problem can be formulated as the following Integer Programming (IP) problem:

IP-Winner:

$$\min_{\mathbf{X}} \sum_{i=1}^M \beta_i(w_i) x_i \quad (4.10)$$

Subject to:

$$q_j = g_j(\mathbf{Z}, \mathbf{X}) \geq r_j, \quad \forall j \in \{1, \dots, N\} \quad (4.11)$$

$$x_i \in \{0, 1\} \quad (4.12)$$

The objective (4.10) is to minimize total virtual cost, i.e., the expected expenditure of the cloud operator. Constraints (4.11) ensure that each subtask's quality requirement is met. Let  $\Psi(\mathbf{B})$  denote the optimal value of IP-Winner and  $\Psi(\mathbf{B}_{-i})$  denote the optimal value of IP-Winner with bid  $\mathbf{b}_i$  removed. We can achieve the optimal as follows:

1. Winner Selection: Select winners  $\mathbf{X}^*$  by solving IP-Winner;
2. Payment Determination:  $p_i := \beta_i^{-1}(\Psi(\mathbf{B}_{-i}) - (\Psi(\mathbf{B}) - \beta_i(w_i)))$  if  $x_i^* = 1$ ;  $p_i := 0$ , otherwise.

This incentive mechanism(referred to as QIM-Opt) is designed by following the VCG (Vickrey-Clarke-Groves [65]) auction mechanism. Note that both the Winner Selection and Payment Determination are different from those in [1].

## 4.4.2 Proof of QIM-Opt Properties

**Lemma 4.3.** *w-Monotonicity is satisfied by the Winner Selection of QIM-Opt.*

*Proof.* Suppose mobile user  $i$  wins by bidding  $\mathbf{b}_i^* = (w_i^*, (z_{i1}^*, \dots, z_{ij}^*, \dots, z_{iN}^*))$ , or equivalently  $\Psi(\mathbf{B}_{-i}) > \Psi((\mathbf{b}_i^*, \mathbf{B}_{-i}))$ . We will prove that mobile user  $i$  also wins by bidding  $\mathbf{b}'_i = (w'_i, (z_{i1}^*, \dots, z_{ij}^*, \dots, z_{iN}^*))$  with any  $w'_i < w_i^*$  through contradiction. Suppose mobile user  $i$  will lose by bidding  $\mathbf{b}'_i$ . Then  $\Psi((\mathbf{b}'_i, \mathbf{B}_{-i})) = \Psi(\mathbf{B}_{-i})$ . Therefore,  $\Psi((\mathbf{b}'_i, \mathbf{B}_{-i})) > \Psi((\mathbf{b}_i^*, \mathbf{B}_{-i}))$ . However, with the same winner vector  $\mathbf{x}$  of  $(\mathbf{b}_i^*, \mathbf{B}_{-i})$ , the total virtual cost of  $(\mathbf{b}_i^*, \mathbf{B}_{-i})$  would be greater than  $(\mathbf{b}'_i, \mathbf{B}_{-i})$  because  $w_i^* > w'_i$ ; this contradicts the statement that  $\Psi((\mathbf{b}'_i, \mathbf{B}_{-i})) > \Psi((\mathbf{b}_i^*, \mathbf{B}_{-i}))$ . Hence, the supposition is false and mobile user  $i$  will also win by bidding  $\mathbf{b}'_i$ . This completes the proof.  $\square$

**Lemma 4.4.** *z-Monotonicity is satisfied by the Winner Selection of QIM-Opt.*

*Proof.* Suppose that mobile user  $i$  wins by bidding  $\mathbf{b}_i^* = (w_i^*, (z_{i1}^*, \dots, z_{ij}^*, \dots, z_{iN}^*))$ , or equivalently  $\Psi(\mathbf{B}_{-i}) > \Psi((\mathbf{b}_i^*, \mathbf{B}_{-i}))$ . We will prove that mobile user  $i$  also wins by bidding  $\mathbf{b}'_i = (w_i^*, (z'_{i1}, \dots, z'_{ij}, \dots, z'_{iN}))$  with all  $z'_{ij} \geq z_{ij}^*$  through contradiction. Suppose mobile user  $i$  will lose by bidding  $\mathbf{b}'_i$ . Then  $\Psi((\mathbf{b}'_i, \mathbf{B}_{-i})) = \Psi(\mathbf{B}_{-i})$ . Therefore,  $\Psi((\mathbf{b}'_i, \mathbf{B}_{-i})) > \Psi((\mathbf{b}_i^*, \mathbf{B}_{-i}))$ . However, with the same winner vector  $\mathbf{x}$  of  $(\mathbf{b}_i^*, \mathbf{B}_{-i})$ , the total virtual cost of  $(\mathbf{b}'_i, \mathbf{B}_{-i})$  is equal to the total virtual cost of  $(\mathbf{b}_i^*, \mathbf{B}_{-i})$ ; this contradicts the statement that  $\Psi((\mathbf{b}'_i, \mathbf{B}_{-i})) > \Psi((\mathbf{b}_i^*, \mathbf{B}_{-i}))$ . Hence, the supposition is false and mobile user  $i$  will also win by bidding  $\mathbf{b}'_i$ . This completes the proof.  $\square$

**Lemma 4.5.**  $p_i = \beta_i^{-1}(\Psi(\mathbf{B}_{-i}) - (\Psi(\mathbf{B}) - \beta_i(w_i)))$  is a critical value for winning mobile user  $i$  in QIM-Opt.

*Proof.* In QIM-Opt for winning mobile user  $i$ ,  $\Psi(\mathbf{B}_{-i}) - (\Psi(\mathbf{B}) - \beta_i(w_i))$  is calculated based on the opportunity cost, which is the increment of total virtual cost of other mobile users caused by the absence of mobile user  $i$ . The opportunity cost in a reverse auction corresponds to the concept of opportunity cost in a forward auction introduced in [65].

Because of the regularity assumption,  $\beta_i(\cdot)$  is a monotonically increasing function. Therefore if  $w_i > \beta_i^{-1}(\Psi(\mathbf{B}_{-i}) - (\Psi(\mathbf{B}) - \beta_i(w_i)))$ , it will result in a virtual cost higher than the opportunity cost. So mobile user  $i$  will not be selected as a winner. Otherwise if  $w_i < \beta_i^{-1}(\Psi(\mathbf{B}_{-i}) - (\Psi(\mathbf{B}) - \beta_i(w_i)))$ , it will yield a virtual cost lower than the opportunity cost. So mobile user  $i$  will be selected as a winner. This completes the proof.  $\square$

**Theorem 4.4.** *QIM-Opt is truthful.*

*Proof.* According to Lemmas 4.3, 4.4, 4.5 and Theorem 4.1, QIM-Opt is truthful.  $\square$

**Theorem 4.5.** *QIM-Opt is individually rational.*

*Proof.* If mobile user  $i$  bids true value  $(c_i, \mathbf{Y}_i)$ , his/her payoff is  $u_i = p_i - c_i = \beta_i^{-1}(\Psi(\mathbf{B}_{-i}) - \Psi(\mathbf{B}) + \beta_i(c_i)) - c_i$ . The optimality of  $\Psi(\mathbf{B})$  causes  $\Psi(\mathbf{B}_{-i}) - \Psi(\mathbf{B}) \geq 0$ . Moreover, since  $\beta_i(\cdot)$  is monotonically increasing, we have  $u_i \geq 0$ . This completes the proof.  $\square$

However, solving IP-Winner may take exponentially long time for a large-sized problem instance. Even for the linear QoC model, IP-Winner is still an Integer Linear Problem (ILP), which is usually hard to solve. In our simulation, we used an optimization solver to provide optimal solutions for the linear QoC model. If we consider other QoC models, then IP-Winner becomes a non-linear integer programming problem, which is even much harder. In addition to time complexity, it has been shown that truthfulness cannot be preserved by a VCG-based auction mechanism with an approximation (instead of optimal) algorithm [65]. Therefore, we present a non-VCG-based QIM with computational efficiency.

### 4.4.3 Computationally Efficient QIM

Here, we present a QIM that is truthful, individually rational and computationally efficient, which we call *QIM-E*. Similar to the above method, QIM-E consists of two phases: *Winner Selection* and *Payment Determination*. *Winner Selection* (Algorithm 4.1) is a heuristic approach, which keeps selecting the mobile user (bidder) with the smallest weight as a

winner. We adopt the following metric  $\alpha_i$  as the weight to assist the selection:

$$\alpha_i = \frac{\beta(w_i)}{\sum_{j=1}^N \frac{v_{ij}}{r_j}}, \quad (4.13)$$

where  $v_{ij}$  is the *marginal* quality score mobile user  $i$  can contribute to subtask  $j$  (according to a QoC model), given a prior winner set (i.e., crowd)  $S'$ :

$$v_{ij} = \begin{cases} \min(g_j(S' \cup \{i\}), r_j) - g_j(S'), & g_j(S') < r_j; \\ 0, & \text{otherwise.} \end{cases}$$

Then the algorithm updates  $\langle v_{ij} \rangle$  and  $\langle \alpha_i \rangle$  because in each iteration,  $g_j(S')$  changes with the newly selected mobile user. For those remaining mobile users with a marginal quality score of 0 for all subtasks, they will be eliminated from the auction. The algorithm stops when there are no mobile users left.

---

**Algorithm 4.1:** Winner Selection of QIM-E

---

**Input :** Bid vector  $\mathbf{B}$ , mobile user (bidder) set  $\mathbf{S}$ , subtask QoC models  $\langle g_j(\cdot) \rangle$  and quality requirement vector  $\mathbf{R}$

**Output:** Winner vector  $\mathbf{X}$

- 1  $x_i := 0, \forall i \in \{1, \dots, M\};$
- 2  $\mathbf{S}' := \emptyset;$
- 3 **while**  $\mathbf{S} \neq \emptyset$  **do**
- 4      $\alpha_i := \frac{\beta(w_i)}{\sum_{j=1}^N \frac{v_{ij}}{r_j}}, \forall i \in \mathbf{S};$
- 5      $k := \arg \min_{i \in \mathbf{S}} (\alpha_i);$
- 6      $x_k := 1;$
- 7      $\mathbf{S} := \mathbf{S} \setminus \{k\};$
- 8      $\mathbf{S}' := \mathbf{S}' \cup \{k\};$
- 9     update  $\langle v_{ij} \rangle, \forall i \in \mathbf{S};$
- 10    **forall**  $m \in \mathbf{S}$  **do**
- 11        **if**  $v_{mj} = 0, \forall j \in \{1, \dots, N\}$  **then**
- 12             $\mathbf{S} := \mathbf{S} \setminus \{m\};$
- 13 **return**  $\mathbf{X} := [x_1, \dots, x_{i..}, x_M];$

---

Note that the weight of each remaining mobile user changes in each iteration; so instead

of maintaining a fixed sorted bidder list, the algorithm updates their weights and makes the selection based on the updated weights. In *Payment Determination* (Algorithm 4.2), to determine the payment for winning mobile user  $l$ , the algorithm repeats the above winner selection for the mobile user set with  $l$  excluded. When a winning mobile user  $k$  is found such that his/her selection can disqualify  $l$  from winning the auction, the payment is set to the highest cost that helps  $l$  disqualify  $k$  or any winning mobile user before  $k$ .

---

**Algorithm 4.2:** Price Determination of QIM-E

---

**Input :** Bid vector  $\mathbf{B}$ , mobile user (bidder) set  $\mathbf{S}$ , subtask QoC models  $\langle g_j(\cdot) \rangle$ , quality requirement vector  $\mathbf{R}$ , winner vector  $\mathbf{X}$

**Output:** Payment vector  $\mathbf{P}$

```

1 forall  $l \in \mathbf{S}$  do
2    $p_l := 0$ ;
3   if  $x_l = 1$  then
4      $\mathbf{S}^* := \mathbf{S} \setminus \{l\}$ ;
5      $\mathbf{S}' := \emptyset$ ;
6     calculate  $\langle v_{lj} \rangle$ ;
7     while  $\mathbf{S}^* \neq \emptyset$  do
8        $\alpha_i := \frac{\beta(w_i)}{\sum_{j=1}^N \frac{v_{ij}}{r_j}}, \forall i \in \mathbf{S}^*$ ;
9        $k := \arg \min_{i \in \mathbf{S}^*} (\alpha_i)$ ;
10       $\mathbf{S}^* := \mathbf{S}^* \setminus \{k\}$ ;
11       $\mathbf{S}' := \mathbf{S}' \cup \{k\}$ ;
12       $p_l := \max(p_l, \beta_l^{-1}(\alpha_k \sum_{j=1}^N \frac{v_{lj}}{r_j}))$ ;
13      update  $\langle v_{ij} \rangle, \forall i \in \{\mathbf{S}^*, l\}$ ;
14      if  $v_{lj} = 0, \forall j \in \{1, \dots, N\}$  then
15        break;
16      forall  $m \in \mathbf{S}^*$  do
17        if  $v_{mj} = 0, \forall j \in \{1, \dots, N\}$  then
18           $\mathbf{S}^* := \mathbf{S}^* \setminus \{m\}$ ;
19 return  $\mathbf{P} := [p_1, \dots, p_l, \dots, p_M]$ ;

```

---

#### 4.4.4 Proof of QIM-E Properties

Next, we show that QIM-E is truthful, individually rational and computationally efficient.

**Lemma 4.6.** *w-Monotonicity and z-Monotonicity are preserved in the Winner Selection of QIM-E.*

*Proof.* Suppose mobile user  $i$  wins by bidding  $(w_i^*, \mathbf{Z}_i^*) = (w_i^*, (z_{i1}^*, \dots, z_{ij}^*, \dots, z_{iN}^*))$ . We prove that: 1) he/she will also win by bidding  $(w_i', \mathbf{Z}_i^*) = (w_i', (z_{i1}^*, \dots, z_{ij}^*, \dots, z_{iN}^*))$  with any  $w_i' < w_i^*$ ; and 2) he/she will also win by bidding  $(w_i^*, \mathbf{Z}_i') = (w_i^*, (z_{i1}', \dots, z_{ij}', \dots, z_{iN}'))$  with all  $z_{ij}' > z_{ij}^*$ .

Let  $\alpha_i^*, \langle v_{ij}^* \rangle$  denote the weight and marginal quality scores respectively when mobile user  $i$  bids  $(w_i^*, \mathbf{Z}_i^*)$ . Let  $\alpha_i'$  and  $\langle v_{ij}' \rangle$  denote the weight and marginal quality scores respectively when he/she bids  $(w_i', \mathbf{Z}_i^*)$  or  $(w_i^*, \mathbf{Z}_i')$ . In either case of  $(w_i', \mathbf{Z}_i^*)$  or  $(w_i^*, \mathbf{Z}_i')$ , it is clear that  $v_{ij}' \geq v_{ij}^*$  and  $\alpha_i' < \alpha_i^*$ ; i.e., the weight becomes smaller in each iteration for him/her. Moreover, as illustrated in lines 10–12 in Algorithm 4.1, if he/she has not been eliminated by bidding  $(w_i^*, \mathbf{Z}_i^*)$ , he/she will not be eliminated by bidding  $(w_i', \mathbf{Z}_i^*)$  or  $(w_i^*, \mathbf{Z}_i')$  either. Therefore, he/she will still win with bid  $(w_i', \mathbf{Z}_i^*)$  or  $(w_i^*, \mathbf{Z}_i')$ . This completes the proof.  $\square$

**Lemma 4.7.** *The payment  $p_l$  is set to a critical value for each winning mobile user (bidder)  $l$  in QIM-E.*

*Proof.* Let  $k$  be the index of mobile user with the smallest weight in each iteration until his/her selection disqualifies mobile user  $l$ . Let  $d_l = \max_k(\beta_l^{-1}(\alpha_k \sum_{j=1}^N \frac{v_{lj}}{r_j}))$ . Note that the marginal quality scores  $\langle v_{lj} \rangle$  are updated in each iteration. If mobile user  $l$  bids  $w_l > d_l$ , then  $\alpha_l > \alpha_k, \forall k$ , meaning  $l$  does not have the smallest weight in any iteration before he/she is disqualified by  $k$  and thus will be eliminated from the auction. If mobile user  $l$  bids  $w_l < d_l$ , then  $\alpha_l < \alpha_k$  in one or more iterations, meaning  $l$  will be chosen as a winner the first time when  $\alpha_l < \alpha_k$  happens. Hence  $d_l$  is the critical value for winning mobile user  $l$ . In Algorithm 4.2, the payment  $p_l$  is set to  $d_l$ . This completes the proof.  $\square$

**Theorem 4.6.** *QIM-E is truthful.*

*Proof.* According to Lemmas 4.6, 4.7 and Theorem 4.1, QIM-E is truthful.  $\square$

**Theorem 4.7.** *QIM-E is individually rational.*

*Proof.* We examine two possible cases. First, it is clear that the payoff of mobile user  $l$  is 0 if mobile user  $l$  is not a winner according to Algorithm 4.2. Second, if mobile user  $l$  is a winner, let the critical value be  $d_l$  and mobile user  $l$ 's cost be  $c_l$ . Since QIM-E preserves the critical payment property as shown in Lemma 4.7, it is obvious that  $w_l < d_l$  and  $d_l = p_l$ . Since  $w_l = c_l$  in a truthful mechanism, it is clear that  $p_l - c_l > 0$ . Therefore, the payoff is always non-negative. This completes the proof.  $\square$

**Theorem 4.8.** *QIM-E is computationally efficient.*

*Proof.* In Algorithm 4.1, line 4 takes  $O(MN)$  time to calculate  $\langle \alpha_i \rangle$  and update  $\langle v_{ij} \rangle$ . Note that finding the mobile user with the minimum weight only takes  $O(M)$  in line 5. Since the while-loop runs  $M$  times, the time complexity of Algorithm 4.1 is  $O(M^2N)$ .

However, in Algorithm 4.2, the for-loop (lines 1–18) iterates  $M$  times, and the inner while-loop (lines 7–18) takes  $O(M^2N)$  time because it has the same complexity with Algorithm 4.1. So Algorithm 4.2 takes  $O(M^3N)$  time. Therefore, the overall time complexity of QIM-E is  $O(M^3N)$ , which completes the proof.  $\square$

## 4.5 Performance Evaluation

### 4.5.1 Baseline Methods

For fair comparisons, we chose two well-designed incentive mechanisms (one of them is truthful and individually rational) as the baselines, instead of trivial random solutions. The first baseline is a revised version of the greedy method with a fixed list of bidders (referred to as Fix-L) presented in [98]. Since we deal with a two-parameter auction (cost and quality score), we use  $\alpha_i = \frac{\beta(w_i)}{\sum_{j=1}^N z_{ij}}$  as the weight to sort the bidders in non-decreasing order to obtain the fixed list. Then we iterate through the fixed list and select winners until the quality requirements of all subtasks are met. The winners are paid based on the

corresponding critical values [98]. Similar as in [98], it can be shown that Fix-L preserves truthfulness and individual rationality.

In the second baseline approach, all bidders are sorted in the non-decreasing order based on their virtual cost (referred to as Low-C). The algorithm repeatedly selects the bidder with the lowest virtual cost among the remaining bidder set. This process stops when quality requirements of all subtasks are met and winners are paid with critical values. Note that even though this approach is not truthful, it is still a good baseline to compare with because the cloud operator tends to directly reduce the expenditure by selecting bidders with low costs.

## 4.5.2 Simulation Settings

We conducted trace-driven simulation for performance evaluation using the mobility dataset [67] of San Francisco taxis, which contains GPS coordinates of approximately 500 taxis collected over 30 days in the San Francisco Bay Area. For the distributions of mobile user (bidder) costs, we considered the uniform distribution  $f_i(c_i) = 0.25$  in the range of  $(0, 4]$ , the exponential distribution  $f_i(c_i) = 0.5e^{-0.5c_i}$  in the range of  $(0, +\infty)$  and  $\chi^2$ -distribution with freedom degree of 2. Note that these functions have the same mean value of 2 and the first two distributions were also used in [42] and [98]. For QoC models of subtasks, we have implemented the linear model, the probabilistic coverage model and the hyperbolic tangent model introduced in Section 4.2. In our simulation, each subtask corresponds a sub-area, each of which is a square-like region with a randomly chosen center, whose left/top and right/bottom boundaries differ by 0.0005 degrees in both longitude and latitude (about 160 feet). We derived the quality score of each taxi  $i$  for a subtask  $j$  by dividing the number of samples of  $i$  within sub-area  $j$  by the number of weeks  $i$  showed up in the dataset, which captures the availability of the mobile user. Due to non-uniform distribution of samples, to ensure the quality requirements are satisfied, we normalized them by a large number, 50, and curved them with an upper and lower bounds of 0.15 and 0.04 respectively.

We came up with the following scenarios for simulation. Simulation runs were conducted on a computer with a 2.2GHz Intel Core i7 CPU and 16GB memory. When the linear QoC model was used, the optimal expected expenditures were obtained using the method presented in Section 4.4.1 (labeled as *QIM-Opt*), in which Gurobi Optimizer [34] was employed to solve the corresponding ILP problems. Each number presented here is an average over 30 runs.

1) In scenarios 1 and 2, the number of subtasks was fixed to 15; quality requirements of subtasks were set to be uniformly distributed in  $[0.7, 0.8]$ . In scenario 1, the linear model was applied for QoC; the number of mobile users was varied for all the cost distributions described above. In scenario 2, the above exponential distribution was applied for costs; the number of mobile users was varied for the three QoC models mentioned above. The results of scenario 1 are presented in Fig. 4.2 and results of scenario 2 are shown in Fig. 4.2(a) and Fig. 4.3.

2) In scenario 3, the linear model was applied for QoC; costs of mobile users were generated by following the above exponential distribution; the number of mobile users was set to 350. The number of subtasks was increased from 5 to 30 with a step size of 5. The corresponding results are presented in Fig. 4.4.

3) In scenario 4, we evaluated the running time of proposed mechanisms. The number of subtasks was fixed to 15; the linear model was applied for QoC; costs of mobile users were generated according to the above exponential distribution; quality requirements of subtasks were uniformly distributed in  $[0.7, 0.8]$ . The number of mobile users was increased from 250 to 500 with a step size of 50. The corresponding results are presented in Fig. 4.5.

### 4.5.3 Simulation Results and Analysis

We can make the following observations from the results.

1) In Fig. 4.2, we show the expected expenditures under different cost distributions, when the linear model was applied for QoC. In Fig. 4.4, we show how the expected ex-

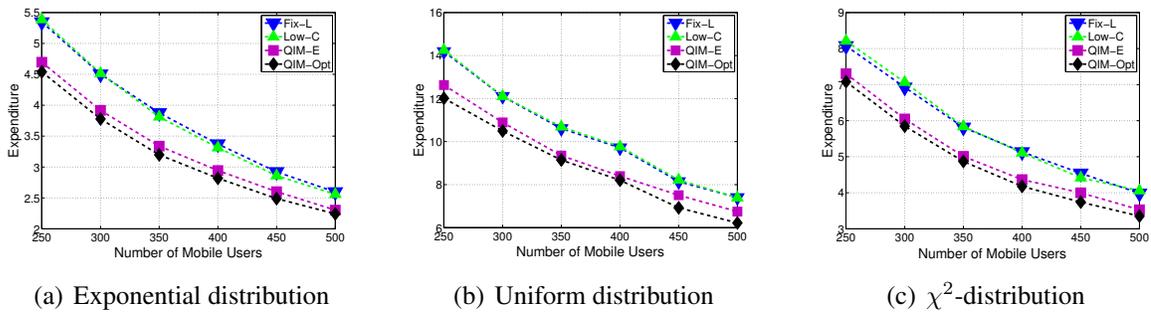


Fig. 4.2: Performance with the linear QoC model and different cost distributions (Scenario 1)

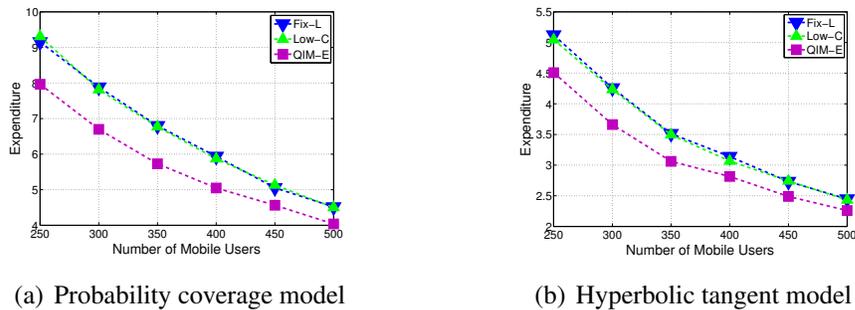


Fig. 4.3: Performance with different QoC models and the exponential cost distributions (Scenario 2)

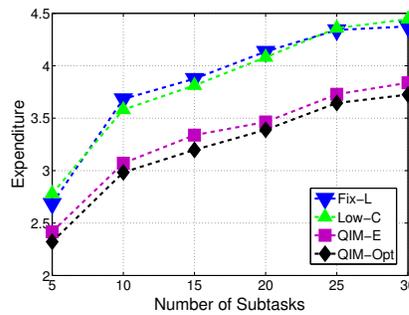


Fig. 4.4: Performance with different numbers of subtasks (Scenario 3)

penditure changes with the number of subtasks. From these figures, we can see that the expected expenditures given by QIM-E are consistently close to the optimal values. Specifically, in Fig. 4.2, QIM-E produces only 3.9%, 5.1% and 4.4% more expenditures than the optimal for the exponential, uniform and  $\chi^2$ -distributions of costs on average respectively. Moreover, in Fig. 4.4, QIM-E gives only 3.2% more expenditures than the optimal on average.

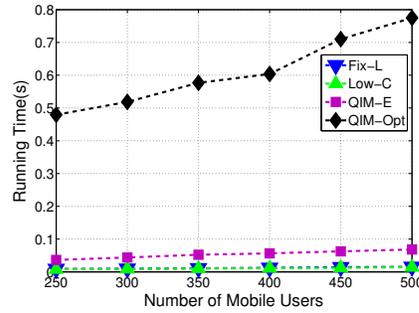


Fig. 4.5: Running time (Scenario 4)

2) From Figs. 4.2–4.4, we can see that QIM-E consistently outperforms Fix-L and Low-C. The reason is that when selecting winners, Low-C does not carefully consider the quality scores of the mobile users. Even though Fix-L considers the individual quality scores, it doesn't carefully take QoC into consideration. On the contrary, QIM-E favors those mobile users who contribute the most marginal QoC. Specifically, in Fig. 4.2, QIM-E produces about 11.9%, 10.7%, 12.6% less expenditures than Fix-L for the exponential, uniform and  $\chi^2$ -distributions of costs on average respectively. Moreover, in Figs. 4.2(a) and 4.3, QIM-E produces about 11.9%, 13.2%, 10.6% less expenditures than Fix-L for the linear, probabilistic coverage and hyperbolic tangent model of QoC respectively. Similar observations can be made from Fig. 4.4. Note that the performance of Low-C is very close to Fix-L in all the scenarios.

3) Monotonicity can be observed in Figs. 4.2–4.4. As expected, in Figs. 4.2 and 4.3, with more mobile users to choose from, all mechanisms yield lower expenditures. On the contrary, we can see that more subtasks lead to higher expenditures no matter which mechanism is used according to Fig. 4.4.

4) Fig. 4.5 shows the running time of different mechanisms with various numbers of mobile users. The running time of QIM-E is only 8.8% of that of QIM-Opt on average, which shows QIM-E is scalable. The running times of QIM-E, Fix-L and Low-C are fairly close to each other, which matches the theoretical analyses that suggest they all have a time complexity of  $O(M^3N)$ .

## 4.6 Summary

In this chapter, we have studied incentive mechanism design for quality-aware and fine-grained MCS. First, we have introduced several models to characterize QoC for different sensing applications. Based on these models, we have presented a novel auction formulation for quality-aware and fine-grained MCS, which minimizes the expected expenditure subject to the quality requirement of each subtask. We have discussed how to achieve the optimal expected expenditure, and presented a practical incentive mechanism to solve the auction problem, which has been shown to be truthful, individual rational and computational efficient. We have conducted trace-driven simulation using the mobility dataset of San Francisco taxis. Extensive simulation results have shown the proposed incentive mechanism achieves noticeable expenditure savings compared to two well-designed baseline methods, and moreover, it produces close-to-optimal solutions.

# CHAPTER 5

## TASK SCHEDULING IN MOBILE CROWDSENSING SYSTEMS: MODELS AND ALGORITHMS

### 5.1 Overview

To minimize energy consumption on smartphones in mobile crowdsensing systems, on one hand, we can strategically schedule sensor data collection activities for a given set of tasks without violating their QoS constraints. On the other hand, since multiple sensing tasks may request data from common sensors at the same or similar time instants, we can share sensor data among them to avoid redundant efforts.

In this chapter, we first consider a simple case called *single-sensor tasks*. In this case, each sensing task only requests data from a single sensor. In a MCS system, single-sensor tasks are quite common; for example, many location-dependent applications may just request smartphones to report their locations. Furthermore, we address a more general case called *multi-sensor tasks*. In this case, some sensing tasks request multiple sensors to report their measurements simultaneously. For example, some applications may request smart-

phones to report both locations and some other sensor readings (such as microphone and wireless signals) to create certain maps, such as urban noise map [70] and wireless signal strength map [76]. Moreover, some sophisticated applications may apply machine learning techniques to different sensor readings to infer non-trivial information from mobile users or their environment, such as the mobile social networking application Sociablesense [69]. The main contributions of our work are summarized in the following:

- We formally define the problem of scheduling a set of single-sensor tasks as the Minimum Energy Single-sensor task Scheduling (MESS) problem and present a polynomial-time optimal algorithm for this problem.
- We present an Integer Linear Programming (ILP) formulation for the Minimum Energy Multi-sensor task Scheduling (MEMS) problem, and present two effective heuristic algorithms to solve it in polynomial time.
- We present extensive simulation results based on real data on sensor energy usages to show the proposed algorithms achieve significant energy savings, compared to a widely-used baseline approach, and moreover, the proposed heuristic algorithms produce close-to-optimal solutions.

To the best of our knowledge, we are the first to conduct a comprehensive study for sensing task scheduling on a smartphone in the context of a general MCS system and present provably-good and practically efficient solutions.

## 5.2 System Model

In this section, we describe our system model, discuss the two cases and formally formulate the corresponding task scheduling problems. First of all, we summarize major notations in Table 5.1.

Table 5.1: Major Notations

Notation	Explanation
$i$ and $N$	The index of time instants and $N =  \Psi $
$j$ and $M$	The index of sensors and the total number of sensors (on a smartphone)
$\mathbf{J}_k$	The set of indices of sensors in multi-sensor task $k$
$k$ and $K$	The index of sensing tasks and the total number of tasks
$p_A(\cdot)$	The sensing accuracy
$q_k$	The QoS requirement of task $k$
$\mathbf{S}_k$	Sensing task $k$
$w_j$	Energy usage of collecting a reading from sensor $j$
$x_{ij}$	The scheduling variable
$\Psi$	The sequence of time instants, at which sensor data can be collected.
$\Omega_k$	The sensing time sequence of task $k$

As mentioned above, we consider a general multi-application multi-task MCS system and focus on a participating smartphone that is requested to undertake a set of sensing tasks involving various sensors. First, we consider a simple case in which each sensing task only involves a single sensor (i.e., single-sensor tasks). A single-sensor task is given by a 4-tuple  $\mathbf{S}_k = (k, j_k, \Omega_k, q_k)$ , where  $k$  is the task ID (that may include information about the user initiating the request);  $j_k$  is the index of the sensor (that is requested to take measurements);  $\Omega_k = \{t_1, \dots, t_{N_k}\}$  is a sequence of time instants at which the sensor readings are requested to be collected, which we call *sensing time sequence*; and  $q_k$  is the Quality of SenSing (QoS) requirement (which will be explained later). Note that many applications may simply request a smartphone to collect sensor readings periodically, i.e., time instants in  $\Omega_k$  are evenly distributed in the time domain. However, our sensing model and algorithms are not restricted to this case, i.e.,  $t \in \Omega_k$  could be any arbitrary time.

For the sake of energy saving, we argue that a sensor measurement may not need to be taken exactly at the requested time instant because readings of some sensors (such as light, temperature, etc) may change slowly over time, i.e., they can be collected at time instants that are slightly different from the requested ones. However, we need to make sure that

QoSS is maintained at an acceptable level. If a smartphone is requested to collect a reading from a sensor at time instant  $t$ , but it does so at  $t'$  instead, then we say the accuracy of this sensing action is  $p_A(t, t') \in [0, 1]$ . Certainly, the closer  $t'$  is to  $t$ , the larger the value of  $p_A(t, t')$ , i.e., the more accurate. Here, we aim to propose a general model for QoSS so any method can be applied to estimate the sensing accuracy (as long as it has the property just mentioned above and it reflects reality). Similar as in [55], a possible solution is to use a bell-shaped function within a value range between 0 and 1:

$$p_A(t, t') = e^{-\frac{(t-t')^2}{2\sigma^2}}, \quad (5.1)$$

where different  $\sigma$ 's can be used to model different sensor readings. Here, smaller  $\sigma$  corresponds to sensor readings that change quickly over time (such as GPS), while larger  $\sigma$  corresponds to those that change slowly over time (such as temperature and light). In our simulation, we used this function to model the accuracy. The model can even be extended by considering the impact of other factors, such as the current motion state (walking, running, driving, etc) of the mobile user, on the value of  $\sigma$ .

Suppose that a sensing task  $\mathbf{S}_k = (k, j_k, \Omega_k, q_k)$  requests a smartphone to collect sensor readings according to a time sequence  $\Omega_k = \{t_1, \dots, t_{N_k}\}$ . The phone does so according to a sensing schedule  $\Gamma = \{t'_1, \dots, t'_{N_k}\}$ . If  $\forall t \in \Omega_k, p_A(t, t') \geq q_k$  (where  $t'$  is the time instant in  $\Gamma$  that is closest to  $t$ , and  $q_k$  is the QoSS requirement of the sensing task), we say the sensing schedule  $\Gamma$  meets the *QoSS requirement of the sensing task*  $\mathbf{S}_k$  and denote it by  $p_A(\Omega_k, \Gamma) \geq q_k$ . Note that since different applications may demand different sensing accuracies,  $q_k$  is defined to an application-specific parameter that varies with sensing tasks.

With a higher QoSS requirement  $q_k$ , the actual sensing time instant  $t'$  needs to be closer to  $t$ . A special case is that when  $q_k = 1$ , the actual sensing time instant  $t'$  need to be exactly the same as  $t$ . However, with a lower QoSS requirement, a relatively larger difference between  $t'$  and  $t$  is allowed. In other words, it will be more flexible to schedule a time

instant  $t'$  to collect a sensor reading. For a smartphone in a mobile crowd sensing system, multiple sensing tasks may request data from common sensors at the same or similar time instants. As a result, with a lower QoS requirement, there is a better chance that sensor data can be shared among multiple sensing tasks to reduce energy consumption.

Without loss of generality, we discretize the time domain by evenly dividing a given sensing scheduling period into intervals (with equal durations) with a sequence of time instants  $\Psi = \{t_1, \dots, t_i, \dots, t_N\}$  and assume that sensor readings can only be taken at those instants. The finer the granularity, the better the QoS (i.e., accuracy), but the higher the computational complexity. The scheduling problem then becomes the problem of finding the "best" subset of such time instants. Now we are ready to define the scheduling problem for the single-sensor task case, which is referred to as the *Minimum Energy Single-sensor task Scheduling (MESS)* problem and is formally presented below:

Unknown decision variables:

- Scheduling variable  $x_{ij} \in \{0, 1\}$ :  $x_{ij} = 1$  if it is scheduled to collect a reading from sensor  $j$  at  $t_i$ ;  $x_{ij} = 0$ , otherwise.

MESS:

$$\min_{\mathbf{X}=\langle x_{ij} \rangle} \sum_{j=1}^M w_j \left( \sum_{i=1}^N x_{ij} \right) \quad (5.2)$$

Subject to:

$$p_A(\Omega_k, \mathbf{X}) \geq q_k, \quad \forall k \in \{1, \dots, K\}. \quad (5.3)$$

In this formulation, a set of  $K$  sensing tasks  $\mathbf{S}_k$  are given as input, the output is the

scheduling matrix  $\mathbf{X} = \langle x_{ij} \rangle$ , and  $w_j$  is the energy usage for taking a reading from sensor  $j$ . The objective (5.2) is to minimize the total sensing energy consumption. Without abusing notations, constraints (5.3) ensure that the sensing schedule  $\mathbf{X} = \langle x_{ij} \rangle$  meets the QoSS requirement of each sensing task.

Next, we address a more general case in which some sensing tasks request a smartphone to collect readings from multiple sensors simultaneously, i.e., multi-sensor tasks. In this case, we still use a 4-tuple  $\mathbf{S}_k = (k, \mathbf{J}_k, \boldsymbol{\Omega}_k, q_k)$  to denote a sensing task, where  $\mathbf{J}_k$  is the set of indices of sensors from which the task requests data. Note that a unique constraint here is that if a multi-sensor task requests data from multiple sensors then these sensor readings must be collected at exactly the same time such that they can be used to generate some meaningful results. Similar to its counterpart in a single-sensor task, the sensing time sequence  $\boldsymbol{\Omega}_k$  of a multi-sensor task, is a sequence of time instants at which sensor readings are requested to be collected.  $q_k$  is again the QoSS requirement of a multi-sensor task. Similarly, any functions or methods can be used in this case to model QoSS as long as they have the properties mentioned above. A feasible solution could still be a bell-shaped function (5.1), whose  $\sigma$ , however, needs to be properly chosen with consideration for multiple sensors. A conservative approach is to take the minimum one.

Note that data collected from a sensor  $j$  in a multi-sensor task may be used by a single-sensor task  $\mathbf{S}_k = (k, j_k, \boldsymbol{\Omega}_k, q_k)$  to fulfill its QoSS requirement if  $j = j_k$ . However, usually data from a single-sensor task are not sufficient to fulfill the QoSS requirement of a multi-sensor task unless multiple single-sensor tasks (with the same set of sensors as that of the multi-sensor task) are scheduled to collect data simultaneously.

Suppose that we are given a multi-sensor task  $\mathbf{S}_k$  and a *feasible* scheduling matrix  $\mathbf{X}$  (to be feasible, it has to meet the unique multi-sensor scheduling constraint mentioned above). If we can still use  $p_A(\boldsymbol{\Omega}_k, \mathbf{X}) \geq q_k$  to denote that a sensing schedule satisfies the QoSS requirement of task  $k$  (which could be a single-sensor task or a multi-sensor task) without abusing notations, then the *Minimum Energy Multi-sensor task Scheduling (MEMS)* prob-

lem can be formally defined in the same way as the MESS problem. We omit the formal definition due to similarity and space limitation. Even though they can be presented in the same way, the MESS problem is a special case of the MEMS problem, and the MEMS problem is much harder since the constraints  $p_A(\Omega_k, \mathbf{X}) \geq q_k, \forall k \in \{1, \dots, K\}$  imply that if  $k$  is a multi-sensor task, then readings from multiple requested sensors must be collected at the same time, and sensor data can be shared among multi-sensor tasks and single-sensor tasks.

### 5.3 Single-Sensor Task Scheduling

In this section, we present a polynomial-time optimal algorithm for the MESS problem defined above.

For the MESS problem, if two tasks request data from two different sensors, they obviously don't interfere with each other; therefore, they can be scheduled independently. The trouble makers are those tasks that request data from a common sensor, which need to be scheduled jointly. So we can apply a divide-and-conquer technique here by dividing given tasks into a collection of non-overlapping subsets of tasks on a sensor-by-sensor basis and solve a simpler problem of scheduling a subset of sensing tasks that request data from a common sensor (which we call *Simplified MESS (SMESS)* problem). Next, we show that we can pre-process a set of sensing tasks according to their QoSS requirements such that the SMESS problem can be formulated to an ILP problem with a nice property.

As mentioned above, the scheduling period is discretized to a sequence of time instants  $\Psi = \{t_1, \dots, t_i, \dots, t_N\}$ . According to the QoSS model, for a given sensing task  $\mathbf{S}_k = (k, j_k, \Omega_k, q_k)$  and a time instant  $t \in \Omega_k$ , we can identify an interval within the scheduling period (i.e., a subset of continuous time instants in  $\Psi$ ) such that at least one sensor reading needs to be collected within the interval to meet the QoSS requirement. Specifically, a time instant  $t_i \in \Psi$  is in the interval as long as  $p_A(t, t_i) \geq q_k$ . We use  $l(t, q_k)$  and  $u(t, q_k)$  to

denote the indices of starting and ending time instants of the interval respectively. Note that usually  $t_{l(t,q_k)} \in \Psi$  and  $t_{u(t,q_k)} \in \Psi$  are distributed on the two sides of  $t$ . Then we can formulate the SMESS problem to an ILP problem, which is formally presented in the following.

Unknown decision variables:

- Scheduling variable  $x_i \in \{0, 1\}$ :  $x_i = 1$  if it is scheduled to collect a reading at  $t_i$ ;  $x_i = 0$ , otherwise.

ILP-SMESS( $\Phi_j$ ):

$$\min \sum_{i=1}^N x_i \quad (5.4)$$

Subject to:

$$\sum_{i=l(t,q_k)}^{u(t,q_k)} x_i \geq 1, \quad \forall S_k \in \Phi_j, \forall t \in \Omega_k. \quad (5.5)$$

In this formulation, the set  $\Phi_j$  of sensing tasks requesting data from sensor  $j$  is given as input and the output is the schedule given by  $\langle x_i \rangle$ . The objective (5.4) is to minimize the total energy consumption. Constraints (5.5) ensure that for each time instant requested by a task,  $t$ , the measurement is taken at least once during the period  $[t_{l(t,q_k)}, t_{u(t,q_k)}]$ , i.e., the QoSS requirement of each task is guaranteed to be satisfied. Now we are ready to present the proposed algorithm for the MESS problem.

The algorithm first divides the given set of sensing tasks into a collection of non-overlapping subsets  $\Phi_j$  according to the sensors requested by them. Then these sensing task subsets will be fed to the ILP-SMESS as input to calculate the sensor-specific schedules ( $\langle x_i^* \rangle$ ), which are then combined to form the final solution, i.e., scheduling matrix

---

**Algorithm 5.1:** The Optimal Algorithm for MESS
 

---

**Input:**  $K$  sensing tasks  $\{\mathbf{S}_1, \dots, \mathbf{S}_K\}$

**Output:** The scheduling matrix  $\mathbf{X} = \langle x_{ij} \rangle$

---

- 1:  $\Phi_j := \emptyset, \forall j \in \{1, \dots, M\};$
  - 2:  $\Phi_{j_k} := \Phi_{j_k} + S_k; \forall k \in \{1, \dots, K\};$
  - 3:  $x_{ij} := 0; \forall i \in \{1, \dots, N\}, \forall j \in \{1, \dots, M\};$
  - 4: **for** ( $j := 1$  **to**  $M$ ) **do**
  - 5:   **if** ( $\Phi_j \neq \emptyset$ ) **then**
  - 6:     Solve the LP relaxation of ILP-SMSS( $\Phi_j$ ) to obtain the scheduling vector  $\langle x_i^* \rangle;$
  - 7:      $\langle x_{ij} \rangle := \langle x_i^* \rangle, \forall i \in \{1, \dots, N\};$
  - 8:   **end if**
  - 9: **end for**
  - 10: **return**  $\langle x_{ij} \rangle;$
- 

$\langle x_{ij} \rangle$ . Note that instead of solving ILP-SMESS, we solve its LP relaxation (denoted by LP-SMESS), which can be done in polynomial time. We will show that doing so always produces integer optimal solutions to ILP-SMESS and the proposed algorithm solves the MESS problem optimally in polynomial time. First, we show that a nice property of ILP-SMESS.

**Definition 5.1** (Totally Unimodular (TUM) [66]). *A square, integer matrix  $\mathbf{B}$  is called UniModular (UM) if its determinant  $\det(\mathbf{B}) = \pm 1$ . An integer matrix  $\mathbf{A}$  is called Totally UniModular (TUM) if every square, nonsingular submatrix of  $\mathbf{A}$  is UM.*

**Definition 5.2** (Consecutive-ones Property [63]). *if  $\mathbf{A}$  is (or can be permuted into) a  $0 - 1$  matrix in which for every row, the 1s appear consecutively, then  $\mathbf{A}$  is TUM.*

**Lemma 5.1.** *The constraint matrix of LP-SMESS in its standard form is TUM.*

*Proof.* Let  $\mathbf{A}_{\bar{N} \times N}$  denote the constraint matrix given by (5.5), which is in the canonical form.  $\bar{N}$  is the total number of constraints in (5.5). According to Definition 5.2,  $\mathbf{A}_{\bar{N} \times N}$  is TUM since in each row of  $\mathbf{A}_{\bar{N} \times N}$ , 1s appear consecutively. However, LP-SMESS has

additional constraints, which are given below:

$$x_i \leq 1, \forall i \in \{1, \dots, N\}. \quad (5.6)$$

By adding slack variables, we can transform the constraint matrix of LP-SMESS given by both (5.5) and (5.6) from the canonical form into the standard form [66], which is given as follows:

$$\mathbf{G} = \left( \begin{array}{c|c|c} \mathbf{A}_{\bar{N} \times N} & -\mathbf{I}_{\bar{N} \times \bar{N}} & \mathbf{0} \\ \hline \mathbf{I}_{N \times N} & \mathbf{0} & \mathbf{I}_{N \times N} \end{array} \right),$$

where  $\mathbf{I}_{N \times N}$  and  $\mathbf{I}_{\bar{N} \times \bar{N}}$  are identity matrices. Next, we need to show that  $\mathbf{G}$  is TUM.

Adding to a TUM matrix with a row or column that is a unit vector will preserve the total unimodularity [10]. So, without losing the total unimodularity, we can add rows with unit vectors iteratively to  $\mathbf{A}_{\bar{N} \times N}$ , yielding a TUM matrix:

$$\left( \begin{array}{c} \mathbf{A}_{\bar{N} \times N} \\ \hline \mathbf{I}_{N \times N} \end{array} \right).$$

Furthermore, more columns with unit vectors will be added with the total modularity preserved, resulting in:

$$\left( \begin{array}{c|c|c} \mathbf{A}_{\bar{N} \times N} & \mathbf{I}_{\bar{N} \times \bar{N}} & \mathbf{0} \\ \hline \mathbf{I}_{N \times N} & \mathbf{0} & \mathbf{I}_{N \times N} \end{array} \right).$$

In addition, total unimodularity will be preserved by multiplying some columns or rows in a TUM matrix with  $-1$  [10]. Clearly, if we multiply the columns in the middle part of the above matrix with  $-1$ , we can obtain  $\mathbf{G}$ , which is still TUM. This completes the proof.  $\square$

**Theorem 5.1.** *Algorithm 5.1 is a polynomial-time optimal algorithm for the MESS problem.*

*Proof.* According to Lemma 5.1 and [66], solving the LP relaxation of ILP-SMESS always produces integer optimal solutions to ILP-SMESS. Moreover, because any two sensing

tasks requesting data from two different sensors can be scheduled independently, combining solutions to a series of ILP-SMESS( $\Phi_j$ ) can yield an optimal solution to the MESS problem.

In addition, the pre-processing (lines 1–3) takes  $O(MN + K) = O(N + K)$  time since  $M$  (i.e., the number of sensors on a smartphone) can be considered as a small constant. The LP solving (lines 4–9) takes  $O(MT(\text{LP-SMESS})) = O(T(\text{LP-SMESS}))$  time, where  $T(\text{LP-SMESS})$  is the time for solving the LP-SMESS (i.e., the LP relaxation of ILP-SMESS). Since LP-SMESS includes  $N$  variables and at most  $(K\hat{N}_{\max}) + N$  constraints (where  $\hat{N}_{\max} = \max_{1 \leq k \leq K} |\Omega_k|$ ). Hence, Algorithm 5.1 is a polynomial-time algorithm. This completes the proof.  $\square$

## 5.4 Multi-Sensor Task Scheduling

In this section, we first show the MEMS problem can be formulated to an ILP problem, which can be used to provide optimal solutions. Then we present two effective heuristic algorithms to solve it.

Multi-sensor task scheduling shares some similarities with single-sensor task scheduling: for a given multi-sensor task  $S_k = (k, \mathbf{J}_k, \Omega_k, q_k)$  and a time instant  $t \in \Omega_k$ , an interval within the scheduling period can also be identified such that the QoSS requirement can be met if readings are collected from all the sensors in  $\mathbf{J}_k$  at least once within this interval. Again,  $l(t, q_k)$  and  $u(t, q_k)$  denote the indices of starting and ending time instants of the interval respectively. However, multi-sensor task scheduling differs from single-sensor task scheduling in the sense that for a multi-sensor task, readings must be collected from all the sensors in  $\mathbf{J}_k$  at exactly the same time. Moreover, for multi-sensor task scheduling, we cannot simply divide all sensing tasks into a collection of non-overlapping subsets of tasks according to their sensors since it is possible (not necessarily always) for two sensor tasks, say  $S_k$  and  $S_{k'}$ , to share data if  $\mathbf{J}_k \cap \mathbf{J}_{k'} \neq \emptyset$ . We can easily come up with a Non-linear

Integer Programming (NIP) formulation for the MEMS problem.

NIP-MEMS:

$$\min_{\mathbf{x}=\langle x_{ij} \rangle} \sum_{j=1}^M w_j \left( \sum_{i=1}^N x_{ij} \right)$$

Subject to:

$$\sum_{i=l(t,q_k)}^{u(t,q_k)} \left( \prod_{j \in \mathbf{J}_k} x_{ij} \right) \geq 1, \quad \forall k \in \{1, \dots, K\}, \forall t \in \Omega_k. \quad (5.7)$$

The objective is again to minimize the total energy consumption. In constraints (5.7), the non-linear term  $\prod_{j \in \mathbf{J}_k} x_{ij}$  takes a value of 1 if and only if  $x_{ij}$  is 1,  $\forall j \in \mathbf{J}_k$  (i.e., all the sensors in  $\mathbf{J}_k$  take measurements simultaneously). Therefore constraints (5.7) ensure that for each requested time instant  $t \in \Omega_k$  of a task  $k$ , readings are collected from all the requested sensors together at least once during the interval  $[t_{l(t,q_k)}, t_{u(t,q_k)}]$ , in other words, both the unique multi-sensor task scheduling constraints and the QoS requirements are met. Even though we have a mathematical programming formulation for the MEMS problem, an NIP problem is notoriously hard to solve. Next, we show we can transform NIP-MEMS to an equivalent ILP problem.

#### 5.4.1 ILP and LP Rounding based Algorithm

The transformation is not trivial. By introducing  $(u(t, q_k) - l(t, q_k) + 1)$  new binary variables for each time instant  $t \in \Omega_k$ ,  $y_i^{kt} = \prod_{j \in \mathbf{J}_k} x_{ij} \forall i \in \{l(t, q_k), \dots, u(t, q_k)\}$ , the non-linear terms in constraints (5.7) can be replaced by newly introduced variables. Furthermore, since  $x_{ij}$  takes binary values, we can establish the connections between the new variables and the scheduling variables by  $y_i^{kt} = \min_{j \in \mathbf{J}_k} x_{ij}$ ; i.e.,  $y_i^{kt} \leq x_{ij}, \forall j \in \mathbf{J}_k$ . By doing so, we can ensure if  $y_i^{kt} = 1$ , then  $x_{ij} = 1, \forall j \in \mathbf{J}_k$ . In this way, we linearize the non-

linear constraints (5.7). Then we can transform NIP-MEMS to an equivalent ILP problem, ILP-MEMS, which is presented in the following:

ILP-MEMS:

$$\min_{\mathbf{x}=\langle x_{ij} \rangle} \sum_{j=1}^M w_j \left( \sum_{i=1}^N x_{ij} \right)$$

Subject to:

$$\sum_{i=l(t,q_k)}^{u(t,q_k)} y_i^{kt} \geq 1, \quad \forall k \in \{1, \dots, K\}, \forall t \in \Omega_k; \quad (5.8)$$

$$y_i^{kt} \leq x_{ij}, \quad \forall k \in \{1, \dots, K\}, \forall t \in \Omega_k,$$

$$\forall j \in \mathbf{J}_k, \forall i \in \{l(t, q_k), \dots, u(t, q_k)\}. \quad (5.9)$$

Even though it is easier to solve ILP-MEMS than NIP-MEMS, it may still take exponentially long time for a large-size problem instance. Since we aim to solve the MEMS problem in an online manner, we need to design fast polynomial-time algorithms. First, we come up with a heuristic algorithm based on ILP-MEMS. The basic idea is to solve the LP relaxation of ILP-MEMS (instead of solving ILP-MEMS directly) and then round non-integer values to integers. Then the problem boils down to how to round. After an extensive empirical study and theoretical analysis, we found that the constraint matrix of the LP relaxation of ILP-MEMS (denoted by LP-MEMS) is unfortunately not (but seems close to be) TUM. However, an interesting finding is for most problem instances, most of scheduling variables take integer values if solving LP-MEMS. Therefore, we come up with a simple LP rounding based algorithm, which is formally presented as Algorithm 5.2.

In this algorithm, we first solve the LP relaxation of ILP-MEMS, which can be done in polynomial time. However, we may end up with values that are not 0 or 1, but fractional between 0 and 1. Those values will be simply rounded to 1. In this way, the QoSS require-

---

**Algorithm 5.2:** LP Rounding based Algorithm for MEMS
 

---

**Input:**  $K$  sensing tasks  $\{\mathbf{S}_1, \dots, \mathbf{S}_K\}$

**Output:** The scheduling matrix  $\mathbf{X} = \langle x_{ij} \rangle$

---

```

1: Solve the LP relaxation of ILP-MEMS to obtain the scheduling matrix  $\langle x_{ij}^* \rangle$ ;
2: for ( $i := 1$  to  $N$ ) do
3:   for ( $j := 1$  to  $M$ ) do
4:     if ( $x_{ij}^* \neq 0$  and  $x_{ij}^* \neq 1$ ) then
5:        $x_{ij} := 1$ ;
6:     else
7:        $x_{ij} := x_{ij}^*$ ;
8:     end if
9:   end for
10: end for
11: return  $\langle x_{ij} \rangle$ ;

```

---

ment of each task is guaranteed to be satisfied. So the solution will certainly be feasible. This simple rounding algorithm works very well on average cases (since as mentioned above, most scheduling variables take integer values after solving LP-MEMS), which will be shown by simulation results. The algorithm is obviously a polynomial-time algorithm.

### 5.4.2 Greedy Algorithm

Even though Algorithm 5.2 is a polynomial-time algorithm, solving an LP problem may take a long time in the worst case. So we propose a faster greedy heuristic algorithm that does not involve LP solving.

The scheduling problem is essentially to determine whether we should take a reading from sensor  $j$  at each time instant  $t_i \in \Psi$ . The basic idea of the proposed greedy algorithm is to keep adding the “best” (explained later) time-sensor pair  $(i, j)$  into the solution (i.e., scheduling sensor  $j$  to collect a reading at  $t_i$ , in other words, set  $x_{ij} := 1$ ) until all the QoS requirements are met. The proposed greedy algorithm is formally presented as Algorithm 5.3.

In the algorithm,  $\Lambda$  is the set of all time-sensor pairs and  $MAX$  is a large positive num-

---

**Algorithm 5.3:** Greedy Algorithm for MEMS
 

---

**Input:**  $K$  sensing tasks  $\{\mathbf{S}_1, \dots, \mathbf{S}_K\}$ ,  $\Lambda$

**Output:** The scheduling matrix  $\mathbf{X} = \langle x_{ij} \rangle$

---

- 1:  $x_{ij} := 0, \forall i \in \{1, \dots, N\}, \forall j \in \{1, \dots, M\}$ ;
  - 2:  $f_{\max} := MAX$ ;
  - 3: **while**  $f_{\max} > 0$  **do**
  - 4:    $(i^*, j^*) := \operatorname{argmax}_{(i,j) \in \Lambda} f(i, j)$ ;
  - 5:    $x_{i^*j^*} := 1$ ;
  - 6:    $\Lambda := \Lambda - \{(i^*, j^*)\}$ ;
  - 7:    $f_{\max} = \max_{(i,j) \in \Lambda} f(i, j)$ ;
  - 8: **end while**
  - 9: **return**  $\langle x_{ij} \rangle$ ;
- 

ber. The key issue is to determine which time-sensor pair  $(i, j)$  to select in each iteration. Here, we need to consider two factors: the energy usage associated with scheduling sensor  $j$  at  $t_i$  (i.e., cost), and the contribution of scheduling sensor  $j$  at  $t_i$  to fulfilling the corresponding QoSS requirements (i.e., profit). Usually it may not result in a good solution if adding the pair leading to the minimum cost or the maximum profit. We, instead, select the time-sensor pair that can lead to the maximum profit-to-cost ratio in each iteration. Specifically, we use the following metric for the time-sensor pair selection:

$$f(i, j) = \sum_k \sum_{t \in \Omega_k \setminus \hat{\Omega}_k} \frac{g(i, j, k, t)}{w_j}, \quad (5.10)$$

where  $\hat{\Omega}_k$  is the set of time instants in  $\Omega_k$  whose QoSS requirements have been fully satisfied, which needs to be updated accordingly;  $w_j$  is the energy usage (cost) for collecting a reading from sensor  $j$ ;  $g(i, j, k, t)$  is the contribution (profit) that can be made by selecting  $(i, j)$  (scheduling sensor  $j$  to collect a reading at  $t_i$ ) to fulfilling the QoSS requirement of time instant  $t \in \Omega_k \setminus \hat{\Omega}_k$  in task  $k$ . Basically,  $f(i, j)$  returns the ratio between the total profit that can be made by selecting  $(i, j)$  and its cost, and the algorithm greedily adds the time-sensor pair that can lead to the maximum profit-to-cost ratio in each iteration.

It is desirable to have a profit function,  $g(\cdot)$ , whose value range is  $[0, 1]$ , with 1 corresponds to the case the corresponding QoS requirement is fully satisfied. There may be multiple options for such a function. We choose to use the following function:

$$g(i, j, k, t) = \begin{cases} \frac{w_j + \sum_{j \in \mathbf{J}_k} x_{ij} w_j}{\sum_{j \in \mathbf{J}_k} w_j}, & l(t, q_k) \leq i \leq u(t, q_k); \\ 0, & \text{otherwise;} \end{cases} \quad (5.11)$$

where  $\sum_{j \in \mathbf{J}_k} x_{ij} w_j$  gives the total energy usage of the sensors that have been scheduled to collect readings at  $t_i$ ; and the denominator is the total energy needed to fully satisfied the QoS requirement at  $t_i$ . For a single-sensor task, it returns either 0 or 1. For a multi-sensor task  $k$ , in an iteration, the QoS requirement of a time instant  $t \in \Omega_k$  may be *partially* satisfied because only part (not all) of requested sensors have been scheduled to work at  $t_i$  ( $l(t, q_k) \leq i \leq u(t, q_k)$ ). We use energy usages  $w_j$  as weights for making  $(i, j)$  selections, which will hopefully lead to less energy consumption for future selections. This can be shown by a simple example in Fig. 5.1. In this example, there are 3 sensors, their energy usages  $w_1 < w_2 < w_3$ , and the scheduling period is discretized to 9 time instants. Suppose that a multi-sensor task  $k$  requests data collections from all three sensors at  $t_5$ , and according to its QoS requirement, candidate time instants are identified and circled by a square. The number "1" indicates the corresponding time-sensor pair has been selected. Consider two time-sensor pairs (that have not yet been selected),  $(5, 2)$  and  $(6, 2)$  (marked by two circles), which have the same cost  $w_2$ . According to our profit function,  $g(5, 2, k, t_5) < g(6, 2, k, t_5)$ , which means selecting  $(6, 2)$  is more favorable because after this selection, selecting  $(6, 1)$  with a cost of  $w_1$  can fulfill the request, otherwise, we have to select  $(5, 3)$  with a cost of  $w_3 > w_1$ . However, if the following unweighted profit function  $g_1(\cdot)$  is used,  $g_1(5, 2, k, t_5) = g_1(6, 2, k, t_5)$ , which is not desirable.

$$g_1(i, j, k, t) = \begin{cases} \frac{1 + \sum_{j \in \mathbf{J}_k} x_{ij}^*}{|\mathbf{J}_k|}, & l(t, q_k) \leq i \leq u(t, q_k); \\ 0, & \text{otherwise.} \end{cases} \quad (5.12)$$

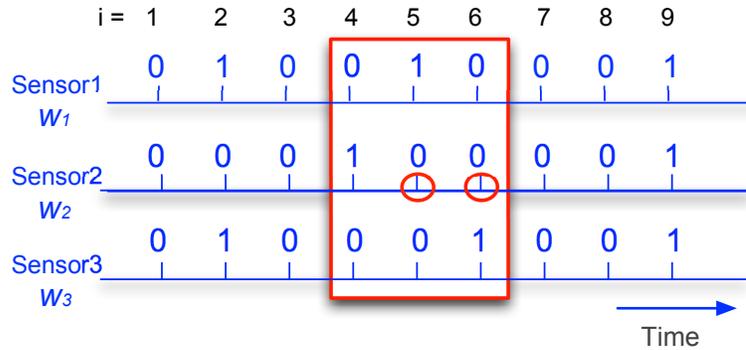


Fig. 5.1: An example for justifying the profit function  $g(\cdot)$

For each  $(i, j)$  pair, evaluating  $f(i, j)$  takes  $O(K\hat{N}_{\max})$  time. The total number of  $(i, j)$  pairs is  $(MN)$ , so we can find a  $(i, j)$  pair with the maximum profit-to-cost ratio in  $O(MNK\hat{N}_{\max})$  time. This process will be repeated at most  $(MN)$  times. Hence the time complexity of this greedy algorithm is  $O((NM)^2K\hat{N}_{\max}) = O(N^2K\hat{N}_{\max})$  since  $M$  (i.e., the number of sensors on a smartphone) can be considered as a small constant.

## 5.5 Performance Evaluation

In this section, we present and discuss simulation results to justify the effectiveness of the proposed algorithms.

We implemented a widely used baseline approach (labelled as “Baseline”) for performance comparisons. The baseline approach schedules sensors to collect readings exactly at the requested time instants given by  $\Omega_k$ . For fair comparisons, data sharing is allowed for the baseline too, i.e., if a common sensor is requested to collect its reading at some time by multiple tasks, the baseline does it only once for all the tasks. For the MESS problem, we compared our optimal algorithm (labelled as “Opt-MESS”) with the baseline method. For the MEMS problem, the proposed LP Rounding based algorithm (labelled as “LP-based”) and greedy algorithm (labelled as “Greedy”) were compared against the baseline approach and the optimal solutions provided by solving ILP-MEMS (“Opt-MEMS”).

In the simulation, energy consumption was used as the primary metric for performance

evaluation. We considered 6 frequently used embedded sensors, including GPS, light sensor, accelerometer, gyroscope, WiFi and 3G. We used real data on power usages of these sensors obtained from the power profile of a Google Nexus 4 [64] smartphone and multiplied them by estimated durations to obtain energy usages, which are summarized in the following table. As mentioned above, we used a bell-shaped function to model QoSS. The values of  $\sigma$  were set to different values, ranging from 6 minutes to 16 minutes.

Table 5.2: Sensor Energy Usages

Sensor	Energy (mAs)
Accelerometer	5
GPS	400
Gyroscope	7
Light sensor	2
WiFi	100
3G	240

The duration of the sensing scheduling period was set to 12 hours (say from 8AM to 8PM), which is evenly divided into 2 minute intervals. As described in Section 5.2, we obtained a sequence  $\Psi$  of evenly-spaced time instants (within the scheduling period), at which sensor readings can be collected. The sensing tasks were randomly generated. Specifically,  $\Omega_k$  in each task  $k$  was set to a sequence of evenly-spaced time instants. For a single-sensor task, the sensor was selected randomly from 6 available sensors mentioned above. While for a multi-sensor task, the set of sensors were randomly selected from the following combinations {GPS, WiFi}, {GPS, 3G}, {GPS, light}, {GPS, WiFi, 3G} and {GPS, accelerometer, gyroscope}. The input of a MEMS problem instance included both multi-sensor tasks and single-sensor tasks.

We evaluated the performance of the proposed algorithms extensively by varying the number of tasks, the duration of tasks, and the QoSS requirement of tasks in different simulation scenarios. Specifically, we came up with 8 scenarios in our simulation: the first four for MESS and the other four for MEMS. In scenario 1, the duration of each task was

set to 2 hours with starting time randomly chosen from  $[1, 10]$  such that their ending times do not exceed 12 (the length of the scheduling period). The QoS requirement was fixed to 0.8. we changed the number of tasks from 5 to 30 with a step size of 5. In scenario 2, the number of tasks and the QoS requirement were fixed to 15 and 0.8 respectively. We changed the duration of each task from 1 hour to 7 hours with a step size of 1 hour. Similar to scenario 1, the starting times were randomly generated in certain ranges accordingly such that their ending times do not exceed 12. In scenario 3, the starting times and durations of tasks were generated in the same way as scenario 1. The number of tasks was set to 15. We varied the QoS requirement of each task from 0.5 to all the way to 1 with a step size of 0.1. In scenario 4, instead of having a fixed value of 12 hours, the sensing periods varied from 8 hours to 28 hours with a step size of 4 hours. The starting times were randomly generated in certain ranges accordingly such that their ending times do not exceed the corresponding sensing periods. The number of tasks, the duration of tasks and the QoS requirement were fixed to 15, 2 hours and 0.8 respectively. The settings of scenarios 5, 6, 7 and 8 were the same as those in scenarios 1, 2, 3 and 4 respectively. But in these scenarios, we tested the algorithms for the MEMS problem instead of the MESS problem.

The simulation results for MESS and MEMS are presented in Fig. 5.2 and Fig. 5.3 respectively. Note that every number in these figures is the average over 50 runs. From these simulation results, we can make the following observations:

1) From Fig. 5.2, we can see that the optimal MESS algorithm reduces the sensing energy consumption significantly by 76.6% on average, compared to the baseline approach. Furthermore, energy savings become more and more significant when the input size (the number of tasks or the duration of tasks) become larger and larger. This leads us to believe that in a MCS system, significant energy savings can be achieved without sacrificing QoS too much by strategically scheduling sensor data collections according to the requirements of tasks.

2) Very similar observations can be made for the multi-sensor task case. Specifically,

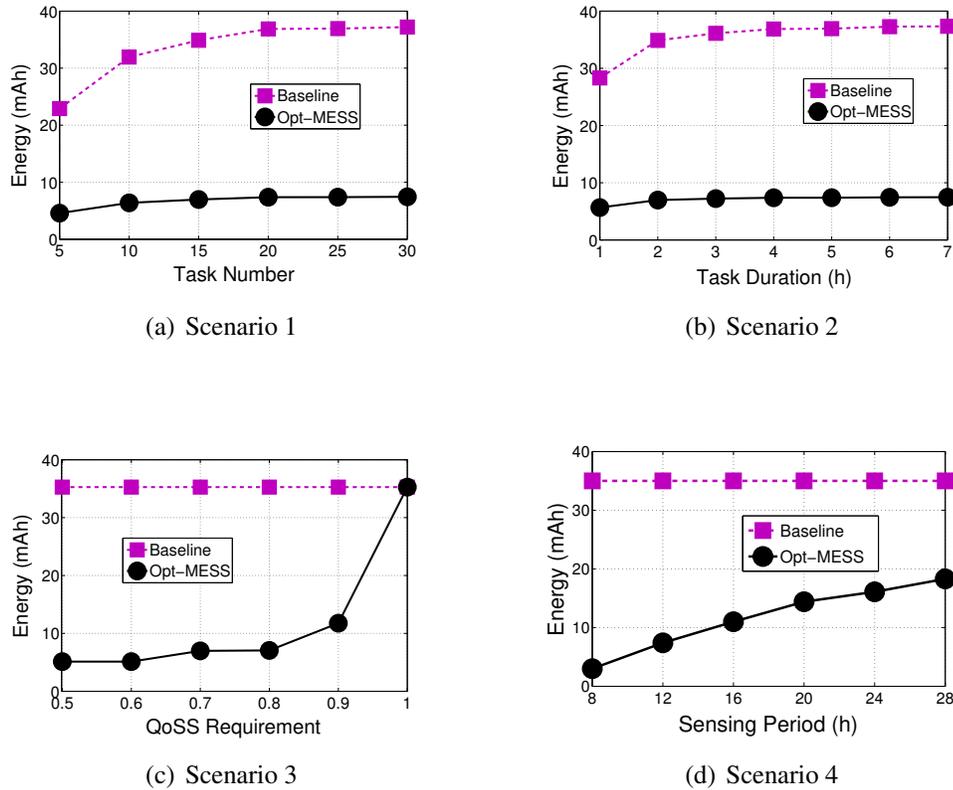


Fig. 5.2: Performance of the MESS algorithms

from Fig. 5.3, we can see that both the LP rounding based heuristic algorithm and the greedy algorithm achieve substantial energy savings of 77.6% and 75.5% respectively on average, compared to the baseline. Moreover, the LP rounding algorithm almost always gives optimal solutions as expected. The greedy algorithm also offers close-to-optimal solutions. The average difference from optimal is only 13.2%.

3) An interesting observation can be made from Figs. 5.2(c) and 5.3(c) about the trade-off between energy consumption and QoS: no matter which algorithm is used (Opt-MESS, LP-based or Greedy), the energy consumption grows with the QoS requirement monotonically but very slowly. Increasing the QoS requirement certainly increases energy consumption since more readings need to be collected at more time instants to fulfill the requirements. However, it turns out it is good to set QoS requirements to relatively large values, say 0.8 or 0.9, since doing so does not lead to substantial increase on energy consumption. This finding provides a valuable insight about how to operate a MCS in practice.

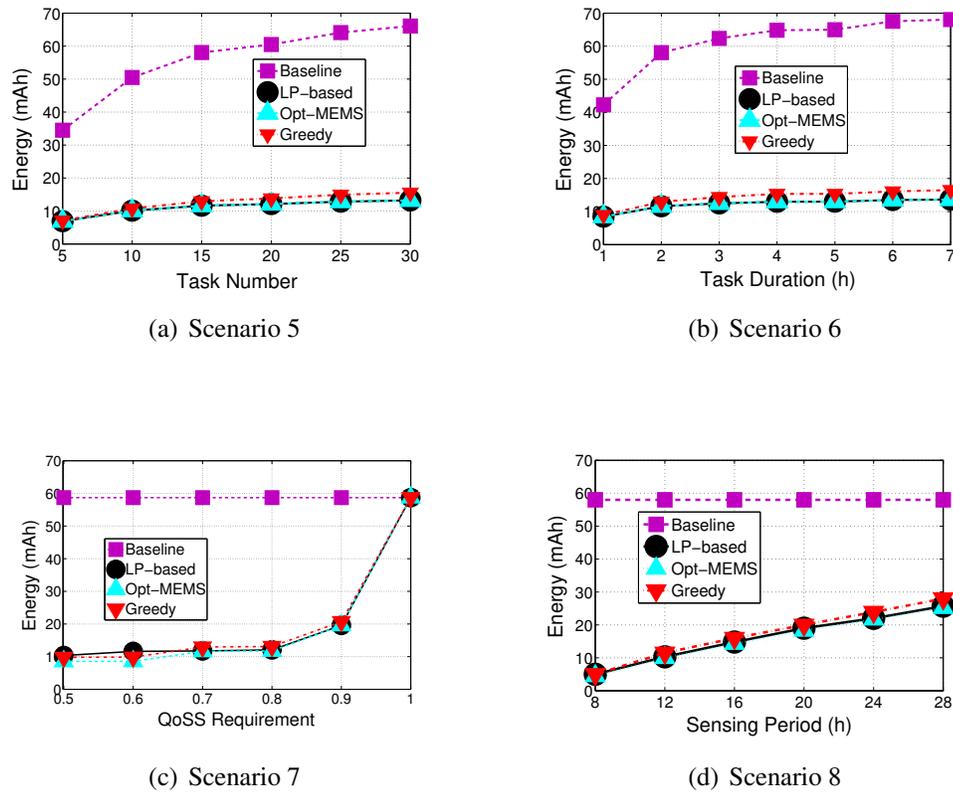


Fig. 5.3: Performance of the MEMS algorithms

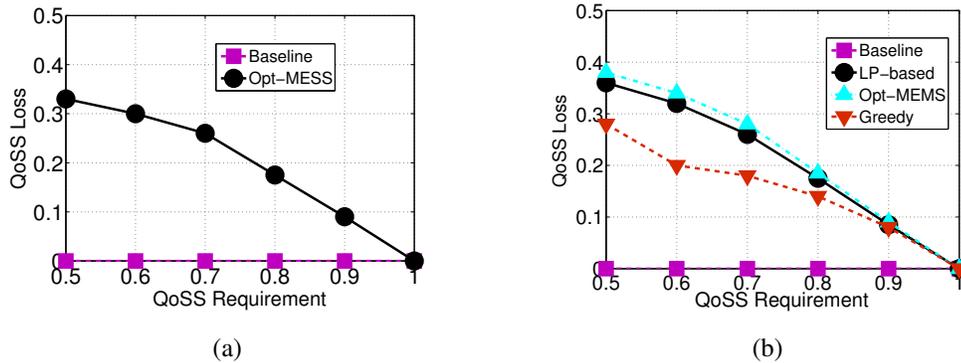


Fig. 5.4: QoS Loss

When the QoS requirement becomes 1 for all tasks, our algorithms perform exactly the same as the baseline approach.

4) From Figs. 5.2(a) and 5.3(a), we can see that no matter which algorithm is used, the energy consumption grows with the number of tasks monotonically. This is because more

sensing tasks result in more requested sensing time instants, which certainly lead to higher energy consumption no matter how sensing activities are optimized. A similar observation can be made from Figs. 5.2(b) and 5.3(b) about how the duration of sensing tasks affects the energy consumption.

5) We can see from Fig. 5.2(d) that, compared to the baseline, the optimal MESS algorithm reduces the sensing energy consumption by 66.6% on average; and from Fig. 5.3(d) that the LP rounding based heuristic algorithm and the greedy algorithm save energy by 72.2% and 70.0% respectively on average. Furthermore, energy savings are more significant when the sensing period is shorter. This is because with a shorter sensing period, it becomes more likely for sensing tasks to request data from common sensors at the same or similar time instants. Therefore, our algorithms can achieve better energy savings.

6) Since QoSS is an important performance metric, we define QoSS loss as  $(1 - p_A(t, t'))$  for a sensing action. In Figs. 5.4(a) and 5.4(b), we show the average QoS loss over all sensing actions in all sensing tasks for MESS and MEMS. The simulation settings were the same as those in scenarios 4 and 8, respectively. From Fig. 5.4, we can observe that the QoSS loss is always 0 for the baseline approach because the baseline approach collects sensor readings exactly at the requested time instants. As expected, all the proposed algorithms introduce certain QoSS losses, which decrease with the QoSS requirement. Moreover, the sum of QoS loss and QoSS requirement should be no greater than 1, which is shown by the results in Fig. 5.4. In addition, we can see from Fig. 5.4(b) that the greedy algorithm leads to smaller QoSS losses compared to the other two algorithms; while it offers quite similar performance in terms of energy consumption.

## 5.6 Summary

In this chapter, we considered the problem of scheduling sensing tasks assigned to a smartphone with the objective of minimizing sensing energy consumption while ensuring QoSS.

First, we considered a simple case in which each sensing task only requests data from a single sensor and formulated the MESS problem correspondingly. We presented an polynomial-time optimal algorithm for this problem. Furthermore, we addressed a more general case in which some sensing tasks request multiple sensors to report their measurements simultaneously, and formulated the MEMS problem correspondingly. We presented an ILP formulation and two effective heuristic algorithms to solve it. Extensive simulation results showed that 1) the proposed algorithms achieve over 75% energy savings on average, compared to a widely-used baseline approach, and 2) the proposed heuristic algorithms for the MEMS problem produce close-to-optimal solutions.

## CHAPTER 6

# CONCLUSIONS

The purpose of this study is to investigate modeling and resource allocation in mobile wireless networks. We envision that in the near future, radios and radio resources in a wireless network can be provisioned as a service to multiple MVNOs, which we refer to as Radio-as-a-Service (RaaS). So we introduced a novel auction-based model to achieve the goal of enabling fair pricing and reasonable resource allocation for RaaS. The RaaS auction mechanisms we proposed has some desirable properties, including truthfulness and individual rationality. Moreover, wireless networks have become more and more advanced and complicated, which are generating a large amount of runtime system statistics. Therefore we proposed to leverage the emerging deep learning techniques for spatiotemporal modeling and prediction in cellular networks, based on big system data. We presented a hybrid deep learning model, which includes a novel autoencoder-based deep model for spatial modeling and LSTMs for temporal modeling. Meanwhile, mobile wireless networks have become an essential part in wireless networking with the prevalence of mobile device usage. The success of a crowdsourcing application highly depends on whether a quality crowd can be recruited to undertake the corresponding tasks. Hence, we studied incentive mechanism design for quality-aware and fine-grained MCS. The incentive mechanisms also have the desirable properties of truthfulness and individual rationality. In a MCS system, a smart-

phone may undertake many different sensing tasks that demand data from various sensors. It can be energy-consuming to collect data from smartphone sensors. Thus, we considered the problem of scheduling different sensing tasks assigned to a smartphone with the objective of minimizing sensing energy consumption while ensuring quality of sensing.

In this thesis, we have explored how to design incentive mechanisms in some systems, where multiple participants are involved. For example, the participants in RaaS include a cloud service provider and multiple MVNOs. In MCS, the cloud operator is buying sensor data from mobile users. Incentive mechanisms are crucial in such systems because it directly determines the trading rules and implicitly defines the behaviors among the participants. Inspired by this idea, my work can be extended by studying incentive mechanism design for other systems in order to motivate participants and determine their behaviors.

In this thesis, we have also investigated a deep learning approach for spatiotemporal modeling and prediction in wireless networks. It shows deep learning can be utilized to model cellular network system data and make accurate prediction. Moreover, deep learning has been shown to dramatically improve the state-of-art on many application domains, including image/video processing, natural language processing, etc. It is particularly suitable to infer information from large datasets and requires very little domain knowledge and engineering by hand. Inspired by the results of this thesis, it could be an important research direction to further explore deep learning in the context of mobile wireless networks.

## REFERENCES

- [1] J. Wang, D. Yang, J. Tang and M. C. Gursoy, Radio-as-a-Service: auction-based model and mechanisms, *IEEE ICC'2015*, pp. 3567–3572.
- [2] J. Wang, J. Tang, D. Yang and M. C. Gursoy, Enabling radio-as-a-service with truthful auction mechanisms, *IEEE Transactions on Wireless Communications*, Vol. 16, No. 4, 2017, pp. 2340–2349.
- [3] J. Wang , J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang and D. Yang, Spatiotemporal modeling and prediction in cellular networks: a big data enabled deep learning approach, *IEEE Infocom'2017*, pp. 1323–1331.
- [4] J. Wang, J. Tang, D. Yang, E. Wang and G. Xue, Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing, *IEEE ICDCS'2016*, pp. 354–363.
- [5] J. Wang, J. Tang, X. Sheng , G. Xue and D. Yang, Enabling green mobile crowd sensing via optimized task scheduling on smartphones, *IEEE Globecom'2015*, pp. 1-7.
- [6] J. Wang, J. Tang, G. Xue and D. Yang, Towards energy-efficient task scheduling on smartphones in mobile crowd sensing systems, *Computer Networks Journal*, Vol. 115, 2017, pp. 100–109.
- [7] I. A. Akbar and W. H. Tranter, Dynamic spectrum allocation in cognitive radio using hidden markov models: poisson distributed case, *Proceedings of IEEE Southeast-Con'2007*, pp. 196–201.

- [8] A. Archer, C. Papadimitriou, K. Talwar and E. Tardos, An approximate truthful mechanism for combinatorial auctions with single parameter agents, *Internet Mathematics*, 2004, pp. 129–150.
- [9] F. Ashtiani, J. Salehi, and M. Aref, Mobility modeling and analytical solution for spatial traffic distribution in wireless multimedia networks, *IEEE Journal on Selected Areas on Communications*, Vol. 21, No. 10, 2003, pp. 1699–1709.
- [10] M. Bazaraa, J. Jarvis and H. Sherali, Linear programming and network flows, *John Wiley & Sons*, 2011.
- [11] Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, Greedy layer-wise training of deep networks, *Proceedings of NIPS'2006*, pp. 153–160.
- [12] Y. Bengio, Learning deep architectures for AI, *Foundations and Trends in Machine Learning*, Vol. 2, No. 1, 2009, pp. 1–127.
- [13] G. Bhanage, I. Seskar, R. Mahindra and D. Raychaudhuri, Virtual basestation: architecture for an open shared WiMAX framework, *Proceedings of ACM VISA'2010*, pp. 1–8.
- [14] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. K. A. Muralidhar, P. Polakos, V. Srinivasan and T. Woo, CloudIQ: a framework for processing base stations in a data center, *Proceedings of ACM MobiCom'2012*, pp. 125–136.
- [15] S. Bi, R. Zhang, Z. Ding and S. Cui, Wireless communications in the era of big data, *IEEE Communications Magazine*, Vol. 53, No. 10, 2015, pp. 190–199.
- [16] P. J. Brockwell and A. D. Richard, Introduction to time series and forecasting, *Springer Science & Business Media*, 2006.
- [17] D. Chen, S. Yin, Q. Zhang, M. Liu and S. Li, Mining spectrum usage data: a large-scale spectrum measurement study, *Proceedings of ACM Mobicom'2009*, pp. 13–24.

- [18] C. Cornelius, *et al.*, Anonymsense: privacy-aware people-centric sensing, *ACM MobiSys'2008*, pp. 211–224.
- [19] T. Das, *et al.*, PRISM: platform for remote sensing using smartphones, *ACM MobiSys'2010*, pp. 63–76.
- [20] G. Ding, Q. Wu, J. Wang and Y. D. Yao, Big spectrum data: the new resource for cognitive wireless networking, *arXiv preprint arXiv'2014*, Vol. 1404.6508.
- [21] H. Drucker, C. Burges, L. Kaufman, A. Smola and V. Vapnik, Support vector regression machines, *Proceedings of NIPS'1996*, pp. 155–161.
- [22] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa and J. Walrand, Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing, *IEEE Infocom'2012*, pp. 1701–1709.
- [23] K. El-Arini, G. Veda, D. Shahaf, and C. Guestrin, Turning down the noise in the blogosphere, *ACM KDD'2009*, pp. 289-298.
- [24] Z. Feng, Y. Zhu, Q. Zhang, L. Ni and A. Vasilakos, TRAC: truthful auction for location-aware collaborative sensing in mobile crowdsourcing, *IEEE Infocom'2014*, pp. 1231-1239.
- [25] Z. Feng, Y. Zhu, Q. Zhang, H. Zhu and J. Yu, Towards truthful mechanisms for mobile crowdsourcing with dynamic smartphones, *IEEE ICDCS'2014*, pp. 11-20.
- [26] C. Ferri, J. Hernandez-Orallo and R. Modroiu, An experimental comparison of performance measures for classification, *Pattern Recognition Letters*, Vol. 30, No. 1, 2009, pp. 27–38.
- [27] F. Fu and U. Kozat, Wireless network virtualization as a sequential auction game, *Proceedings of IEEE Infocom'2010*.

- [28] S. Gandhi, C. Buragohain, L. Cao, H. Zheng and S. Suri, A general framework for wireless spectrum auctions, *New Frontiers in Dynamic Spectrum Access Networks'2007*, pp. 22-33.
- [29] S. Gaonkar, *et al.*, Micro-blog: sharing and querying content through mobile phones and social participation, *ACM MobiSys'2008*, pp. 174–186.
- [30] F. A. Gers, J. Schmidhuber and F. Cummins, Learning to forget: continual prediction with LSTM, *Neural Computation*, Vol. 12, No. 10, 2010, pp. 2451–2471.
- [31] A. Graves, Supervised sequence labelling with recurrent neural networks, *Springer*, 2012.
- [32] A. Graves and N. Jaitly, Towards end-to-end speech recognition with recurrent neural networks, *Proceedings of ICML'2014*, pp. 1764–1772.
- [33] A. Gudipati, D. Perry, L. Li and S. Katti, SoftRAN: software defined radio access network, *Proceedings of ACM HotSDN'2013*, pp. 25–30.
- [34] Gurobi Optimizer, <http://www.gurobi.com/>
- [35] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw and W. A. Stahel, Robust statistics: the approach based on influence functions, *John Wiley & Sons*, 2011.
- [36] S. He, D. Shin, J. Zhang and J. Chen, Toward optimal allocation of location dependent tasks in crowdsensing, *IEEE Infocom'2014*, pp. 745-753.
- [37] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780.
- [38] S. Hochreiter, Y. Bengio, P. Frasconi and J. Schmidhuber, Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, *A Field Guide to Dynamical Recurrent Neural Networks*, IEEE Press, 2001.

- [39] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, *OTexts*, 2014.
- [40] W. C. Hong, Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting, *Neural Computing and Applications*, Vol. 21, No. 3, 2012, pp. 583–593.
- [41] M. M. Islam, M. M. Hassan, G-W. Lee and E-N Huh, A survey on virtualization of wireless sensor networks, *IEEE Sensors Journal*, Vol. 12, 2012, pp. 2175–2207.
- [42] J. Jia, Q. Zhang, Q. Zhang, and M. Liu, Revenue generation for truthful spectrum auction in dynamic spectrum access, *ACM MobiHoc'2009*, pp. 3-12.
- [43] H. Jin, L. Su, D. Chen, K. Nahrstedt and J. Xu, Quality of information aware incentive mechanisms for mobile crowd sensing systems, *ACM MobiHoc'2015*.
- [44] P. Klemperer, What really matters in auction design, *Journal of Economic Perspectives*, Vol. 16, No. 1, 2002, pp. 169-189.
- [45] <http://keras.io/>
- [46] P. Klemperer, What really matters in auction design, *Journal of Economic Perspectives*, 2002, pp. 169–189.
- [47] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, NVS: a substrate for virtualizing wireless resources in cellular networks, *IEEE/ACM Transactions on Networking*, Vol. 20, No. 5, 2012, pp. 1333–1346.
- [48] R. Kokku, R. Mahindra, H. Zhang and S. Rangarajan, CellSlice: cellular wireless resource slicing for active RAN sharing, *Proceedings of IEEE COMSNETS'2013*, pp. 1–10.
- [49] I. Koutsopoulos, Optimal incentive-driven design of participatory sensing systems, *IEEE Infocom'2013*, pp. 1402-1410.

- [50] V. Krishnan, Auction theory, *Academic Press*, 2009.
- [51] N. D. Lane, E. Miluzzo, H. Lu and D. Peebles, A survey of mobile phone sensing, *IEEE Communications Magazine*, Vol. 48, No. 9, 2010, pp. 140–150.
- [52] H. Larochelle, D. Erhan, A. Courville, J. Bergstra and Y. Bengio, An empirical evaluation of deep architectures on problems with many factors of variation, *Proceedings of ICML*, 2007, pp. 473–480.
- [53] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature*, Vol. 521, No. 7553, 2015, pp. 436–444.
- [54] L. Li, Z. M. Mao and J. Rexford, CellSDN: software-defined cellular networks, *Technical Report*, 2012.
- [55] K. Lin, A. Kansal, D. LyMBERopoulos and F. Zhao, Energy-accuracy trade-off for continuous mobile device location, *ACM MobiSys'2010*, pp. 285–297.
- [56] H. Lu, N. D. Lane, S. B. Eisenman and A. T. Campbell, Bubble-sensing: binding sensing tasks to the physical world, *Journal of Pervasive and Mobile Computing*, Vol. 6, No. 1, 2010, pp. 58–71.
- [57] T. Luo, H. Tan, and L. Xia, Profit-maximizing incentive for participatory sensing, *IEEE Infocom'2014*, pp. 127-135.
- [58] S. Madhani, M. Tauil and T. Zhang, Collaborative sensing using uncontrolled mobile devices, *International Conference on Collaborative Computing, Networking, Applications and Worksharing*, 2005.
- [59] R. Mahindra, M. A. Khojastepour, H. Zhang, and S. Rangarajan, Radio access network sharing in cellular networks, *Proceedings of IEEE ICNP'2013*, pp. 1–10.

- [60] K. R. Muller, A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen and V. Vapnik, Predicting time series with support vector machines, *Proceedings of ICANN*, 1997, pp. 999–1004.
- [61] R. Myerson, Optimal auction design, *Math of Operations Research*, Vol. 6, No. 1, 1981, pp. 58-73.
- [62] K. Nakauchi, K. Ishizu, H. Murakami, A. Nakao and H. Harada, AMPHIBIA: a cognitive virtualization platform for end-to-end slicing, *Proceedings of IEEE ICC'2011*.
- [63] G. Nemhauser and L. Wolsey, Integer and combinatorial optimization, *John Wiley & Sons*, 1988.
- [64] Google Nexus 4, <http://www.google.com/nexus/4>
- [65] N. Nisan, T. Roughgarden, E. Tardos and V. V. Vazirani, Algorithmic game theory, *Cambridge University Press*, 2007.
- [66] C. Papadimitriou and K. Steiglitz, Combinatorial optimization: algorithms and complexity, *Courier Dover Publications*, 1998.
- [67] M. Piorkowski, N. Sarafijanovic-Djukic and M. Grossglauser, A parsimonious model of mobile partitioned networks with clustering, *IEEE COMSNETS'2009*, pp. 1-10.
- [68] M-R Ra, B. Liu, T. L. Porta and R. Govindan, Medusa: a programming framework for crowd-sensing applications, *ACM MobiSys'2012*, pp. 337–350.
- [69] K. Rachuri *et al.*, SociableSense: exploring the trade-offs of adaptive sampling and computation offloading for social sensing, *ACM MobiCom'2011*, pp. 73–84.
- [70] R. Rana, C. Chou, S. Kanhere, N. Bulusu and W. Hu, Ear-phone: an end-to-end participatory urban noise mapping system, *ACM/IEEE IPSN'10*, pp. 105–116.

- [71] D. E. Rumelhart, G. E. Hinton and R. J. Williams, Learning internal representations by error propagation, *Parallel Distributed Processing*, MIT Press, pp. 318–362.
- [72] N. I. Sapankevych and R. Sankar, Time series prediction using support vector machines: a survey, *IEEE Computational Intelligence Magazine*, Vol. 4, No. 2, 2009, pp. 24–38.
- [73] <http://scikit-learn.org/stable/>
- [74] S. Sengupta, M. Chatterjee and S. Ganguly, An economic framework for spectrum allocation and service pricing with competitive wireless service providers, *New Frontiers in Dynamic Spectrum Access Networks'2007*, pp. 89-98.
- [75] Sensordrone, <http://sensorcon.com/sensordrone/>
- [76] Sensorly, <http://www.sensorly.com>
- [77] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang and J. Wang, Geospatial and temporal dynamics of application usage in cellular data networks, *IEEE Transactions Mobile Computing*, Vol. 14, No. 7, 2015, pp. 1369–1381.
- [78] X. Sheng, J. Tang and W. Zhang, Energy-efficient collaborative sensing with mobile phones, *IEEE Infocom'2012*.
- [79] X. Sheng, J. Tang, X. Xiao and G. Xue, Sensing as a service: challenges, solutions and future directions, *IEEE Sensor Journal*, Vol. 13, No. 10, 2013, pp. 3733-3741.
- [80] X. Sheng, J. Tang, C. Gao, W. Zhang and C. Wang, Leveraging load migration and basestation consolidation for green communications in virtualized cognitive radio networks, *Proceedings of IEEE Infocom'2013*, pp. 1291–1299.
- [81] Y. Shu, M. Yu, J. Liu and O. Yang, Wireless traffic modeling and prediction using seasonal ARIMA models, *Proceedings of IEEE ICC'2003*, pp. 1675–1679.

- [82] Y. Singer, Budget feasible mechanisms, *IEEE FOCS'2010*, pp. 765-774.
- [83] K. Sundaresan, M. Y. Arslan, S. Singh, S. Rangarajan and S. V. Krishnamurth, Fluid-Net: a flexible cloud-based radio access network for small cells, *Proceedings of ACM MobiCom'2013*, pp. 2816–2820.
- [84] J. A. K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, *Neural Processing Letters*, Vol. 9, No. 3, 1999, pp. 293–300.
- [85] J. Tang, G. Xue and C. Chandler, Interference-aware routing and bandwidth allocation for QoS provisioning in multihop wireless networks, *Wireless Communications and Mobile Computing (WCMC)*, Vol. 5, No. 8, 2005, pp. 933–944.
- [86] J. Tang, G. Xue and W. Zhang, Cross-layer optimization for end-to-end rate allocation in multi-radio wireless mesh networks *Wireless Networks Journal*, Vol. 15, No. 1, 2009, pp. 53–64.
- [87] V. K. Tumuluru, P. Wang and D. Niyato, Channel status prediction for cognitive radio networks, *Wireless Communications and Mobile Computing*, Vol. 12, No. 10, 2012, pp. 862–874.
- [88] K. Tutschku and P. Tran-Gia, Spatial traffic estimation and characterization for mobile communication network design, *IEEE Journal on Selected Areas on Communications*, Vol. 16, No. 5, 1998, pp. 804–811.
- [89] W. Vickrey, Counterspeculation, auctions, and competitive sealed tenders, *Journal of Finance*, 1961, pp. 8-37.
- [90] P. Vincent, H. Larochelle, Y. Bengio and P.A. Manzagol, Extracting and composing robust features with denoising autoencoders, *Proceedings of ICML*, 2008, pp. 1096–1103.

- [91] Y. Wang *et al.*, A framework of energy efficient mobile sensing for automatic user state recognition, *ACM MobiSys'2009*, pp. 179–192.
- [92] Z. Wang and S. Salous, Spectrum occupancy statistics and time series models for cognitive radio, *Journal of Signal Processing Systems*, Vol. 62, No. 2, 2011, pp. 145–155.
- [93] H. Weinschrott, F. Durr and K. Rothermel, StreamShaper: coordination algorithms for participatory mobile urban sensing, *IEEE MASS'2010*, pp. 195–204.
- [94] L. Xia, S. Kumar, X. Yang, P. Gopalakrishnan and Y. Liu, Virtual WiFi: bring virtualization from wired to wireless, *Proceedings of ACM VEE'2011*, pp. 181–192.
- [95] L. Xiang, X. Ge, C. Liu, L. Shu, and C. Wang, A new hybrid network traffic prediction method, *Proceedings of IEEE Globecom'2010*.
- [96] D. Yang, X. Fang and G. Xue, Truthful auction for cooperative communications, *Proceedings of ACM MobiHoc'2011*.
- [97] D. Yang, G. Xue, X. Fang and J. Tang, Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing, *ACM MobiCom'2012*, pp. 173–184.
- [98] D. Yang, X. Fang and G. Xue, Truthful auction for cooperative communications with revenue maximization, *Proceedings of IEEE ICC'2012*, pp. 4888–4892.
- [99] D. Yang, G. Xue, X. Fang, and J. Tang, Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing, *ACM MobiCom'2012*, pp. 173–184.
- [100] D. Yun and Y. Yi, Virtual network embedding in wireless multihop networks, *Proceedings of CFI'2011*.
- [101] Y. Zaki, L. Zhao, C. Goerg and A. Timm-Giel, LTE mobile network virtualization: exploiting multiplexing and multi-user diversity gain, *Mobile Network Applications*, 2011, Vol. 16, pp. 424–432.

- [102] W. Zaremba and I. Sutskever, Learning to execute, *arXiv preprint arXiv'2014*, Vol. 1410.4615.
- [103] X. Zhang, G. Xue, R. Zhou, D. Yang and J. Tang, You better be honest: discouraging free-riding and false-reporting in mobile crowdsourcing, *IEEE Globecom'2014*, pp. 4971–4976.
- [104] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, Spatio-temporal compressive sensing and internet traffic matrices, *Proceedings of ACM Sigcomm'2009*, pp. 267–278.
- [105] D. Zhao, X. Li and H. Ma, How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint, *IEEE Infocom'2014*.
- [106] Z. Zhuang, K. Kim and J. Singh, Improving energy efficiency of location sensing on smartphones, *ACM MobiSys'2010*, pp. 315–330.
- [107] B. Zhou, D. He, Z. Sun and W. H. Ng, Traffic modeling and prediction using ARIMA/GARCH model, *Proceedings of HET-NETs*, 2005.
- [108] X. Zhou, S. Gandhi, S. Suri and H. Zheng, eBay in the sky: strategy-proof wireless spectrum auctions, *Proceedings of ACM MobiCom'2008*, pp. 2–13.
- [109] X. Zhou and H. Zheng, TRUST: a general framework for truthful double spectrum auctions, *Proceedings of IEEE Infocom'2013*, pp. 999–1007.
- [110] Y. Zhou, Y. Li, G. Sun, D. Jin, L. Su and L. Zeng, Game theory based bandwidth allocation scheme for network virtualization, *Proceesings of IEEE Globecom'2010*, pp 1–5.
- [111] Z. Zhu, P. Gupta, Q. Wang, S. Kalyanaraman, Y. Lin, H. Franke and S. Sarangi, Virtual base station pool: towards a wireless network cloud for radio access networks, *Proceedings of ACM CF'10*.

# Jing Wang

## Qualifications

- Best Paper Award of 2015 IEEE GLOBECOM;
- 3 years experience in Machine Learning and Deep Learning applications;
- 3 years Java and Android development on mobile phones;
- 2 years Python and Django framework for server implementation;
- Solid mathematical and engineering background with strong problem solving skills.

## Education

- 08/2011 - **Ph.D of Electrical & Computer Engineering**, *Syracuse University, NY*,  
Present GPA: 3.98/4.0.  
Research interests: Machine learning, Deep learning, Mobile sensing, Crowdsourcing.
- 09/2007 - **B.Eng in Electronic Information Engineering**, *Beihang University, China*,  
07/2011 GPA: 3.8/4.0.

## Experience

- 05/2017 - **Google Intern**, Research and Machine Intelligence.  
08/2017 Design and implement machine learning methods to model and predict mobile user behaviors.
- 05/2013 - **Research Assistant**, NetLab, Syracuse University.  
05/2017 Design and implement projects to enhance mobile phone' sensing ability and make mobile phones as service providers.

## Projects

### A Deep Learning Approach for Spatiotemporal Modeling and Prediction

Introduction: We propose a hybrid deep learning approach to model data with both spatial and temporal dimensions.

- Contributions:
- Incorporate a global stacked autoencoder and multiple local stacked autoencoders to obtain a representation of the local data for spatial modeling;
  - Present the algorithm of pre-training process and fine-tuning process for spatial modeling;
  - Model the temporal characteristics with Long Short-Term Memory;
  - Apply real wireless traffic data to justify the effectiveness of our approach.

### Lifestyle Learning via Phone Sensing

Introduction: We utilize the unique sensing capabilities of a smartphone to learn its user's lifestyle.

- Contributions:
- Implement Android mobile app and remote Django server;
  - Apply machine learning algorithms in server to discover user's points of interest;
  - Improve energy efficiency with adaptive sampling mechanism;
  - Predict user's near future behaviors based on lifestyle pattern and current context.

### An Objective Ranking System Based on Mobile Phone Sensing

Introduction: Unlike Yelp and TripAdvisor, which rate and rank venues based on subjective ratings provided by users, we design and implement an objective ranking system based on data collected via mobile phone sensing.

- Contributions:
- Implement Android mobile app to manage sensing process, collect sensing data and communicate with remote server;
  - Integrate external sensors to mobile app via bluetooth interfaces;
  - Design Django server to manage sensing tasks and process sensing data;
  - Apply Protocol Buffers, PostgreSQL, BeanShell, etc. for better performance.

### Enabling Green Mobile Crowd Sensing via Optimized Task Scheduling on Smartphones

Introduction: We investigate how to schedule different sensing tasks assigned to a smartphone with the objective of minimizing energy consumption while ensuring quality of sensing.

- Contributions:
- Study the problem of single-sensor task scheduling with minimum energy consumption and present a polynomial-time optimal algorithm;
  - Present the problem of multi-sensor task scheduling and design an effective heuristic algorithm to solve it in polynomial time to reduce energy consumption;
  - Present extensive simulations on real data to justify our proposed algorithms.

### Quality-aware Incentive Mechanisms for Mobile Crowdsensing

Introduction: We focus on incentive mechanisms in mobile crowdsensing with the consideration of quality of services/data.

- Contributions:
- Introduce mathematical quality models for various sensing applications;
  - Present a novel auction formulation for crowdsensing to minimize the expected expenditure with quality requirements;
  - Formulate a truthful, individually rational and computationally efficient incentive mechanism to solve the auction problem;
  - Apply the mobility dataset of San Francisco taxis to justify our mechanism.

---

## Publications

1. J. Wang, D. Yang, J. Tang and M. C. Gursoy,  
Radio-as-a-Service: auction-based model and mechanisms,  
IEEE International Conference on Communications (ICC), 2015.
2. J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang and D. Yang,  
Spatiotemporal Modeling and Prediction in Cellular Networks: A Big Data Enabled Deep Learning Approach,  
IEEE Conference on Computer Communications (INFOCOM), 2017.
3. J. Wang, J. Tang, D. Yang, E. Wang and G. Xue,  
Quality-aware and Fine-grained Incentive Mechanisms for Mobile Crowdsensing,  
IEEE International Conference on Distributed Computing Systems (ICDCS), 2016.
4. J. Wang, J. Tang, X. Sheng, G. Xue and D. Yang,  
Enabling Green Mobile Crowd Sensing via Optimized Task Scheduling on Smartphones,  
IEEE Global Communications Conference (GLOBECOM), 2015, Best Paper Award.
5. J. Wang, D. Yang, J. Tang and M. C. Gursoy,  
Enabling Radio-as-a-Service with Truthful Auction Mechanisms,  
IEEE Transactions on Wireless Communications, 2017.
6. J. Wang, J. Tang, G. Xue and D. Yang,  
Towards Energy-efficient Task Scheduling on Smartphones in Mobile Crowd Sensing Systems,  
Journal of Computer Networks, 2016.
7. Z. Xu, Y. Wang, J. Tang, J. Wang and M. C. Gursoy,  
A Deep Reinforcement Learning based Framework for Power-Efficient Resource Allocation in Cloud RANs,  
IEEE International Conference on Communications (ICC), 2017.
8. X. Sheng, J. Tang, J. Wang, T. Li and G. Xue,  
LIPS: Lifestyle Learning via Mobile Phone Sensing,  
IEEE Global Communications Conference (GLOBECOM), 2016.
9. X. Sheng, J. Tang, J. Wang, C. Gao and G. Xue,  
SOR: An Objective Ranking System Based on Mobile Phone Sensing,  
IEEE International Conference on Distributed Computing Systems (ICDCS), 2014.

---

## Resources

Linkedin <https://www.linkedin.com/in/wang27jing>

My Site <https://wang27jing.github.io/>