Optimal Selection of Crowdsourcing Workers Balancing Their Utilities and Platform Profit

Sujan Sarker

Exam Roll: 1314

Registration No: Ha - 771

Session: 2015 - 2016

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science and Engineering

Department of Computer Science And Engineering

University of Dhaka

Dhaka-1000, Bangladesh

November, 2017

Declaration of Authorship

We declare that this thesis titled, "Optimal Selection of Crowdsourcing Workers Balancing Their Utilities and Platform Profit" and the works presented in it are our own. We confirm that:

- The full part of the work was done while in candidature for an MS research degree in this University.
- Any part of this thesis has not previously been submitted for a degree or any other qualification in this University or any other institution.
- We have consulted the published works of others with appropriate references.
- This thesis work is done entirely by us and our contributions and enhancements from other works are clearly stated.

Signed:

Candidate

Countersigned:

Supervisor

Abstract

A Mobile Crowd Sourcing (MCS) platform outsources sensing tasks to numerous mobile worker devices. The collected data is analyzed and processed information is shared among many other interested users. The platform pays to the workers for sensing data and earns money from the users receiving processed information services. Distributing the sensing workloads among the potential workers so as to maintain required data quality and to make a reasonable amount of profit is a challenging problem for an MCS platform.

In this thesis, we develop a workload allocation policy, namely PQ-Trade system, that analyzes the boundary performances and makes a reasonable trade-off between worker utility and platform profit. The PQ-Trade system quantifies the utility (i.e., quality of the sensed data) of a worker as a function of worker mobility, current location and past sensing records. The workload allocation problem is formulated as a multi-objective non-linear programming (MONLP) problem which aims to make desired trade-off between the worker utility and platform profit. The allocation problem is shown to be NP-hard and thus we develop two greedy algorithms with relaxed constraints to achieve near-optimal solutions.

Performance of the proposed workload allocation policy is evaluated in a distributed computation environment using MATLAB. The results show the effectiveness of the proposed system compared to state-of-the-art-works in terms of platform profit, utility of the workers and request service satisfaction.

Acknowledgment

As a matter of first importance, I am thankful to almighty creator for giving me the ca-pability, ability, chance, collaborating hands and stamina to stroll on this path. I also want to express my sincere gratitude to my research supervisor, Md. Abdur Razzaque. Without his support and committed inclusion in every venture all through the methodology, this thesis would have never been fulfilled. His ability to indulge me in my work in order to clarify my own inquiry effectively is something significant for me which I owed to him profoundly. I would want to thank him kindly for his aspiring guidance, invaluably constructive criticism and friendly advice during the thesis work.

The individuals from the Green Networking and Research(GNR) group have contributed tremendously to my scholarly life at University of Dhaka. I would like to thank the members of GNR group with whom I had discussed the early versions of the thesis. I should likewise be thankful to my teachers who helped me in various ways by giving different resources and moral support. At long last, my novel thanks must go to my mother for teaching me to strive for the challenging choices and my friends for their consistent support and inspiration.

I must also be thankful to the ICT department of MoPTIT, GoB, for the fellowship support which has helped me keep full concentration on my thesis work.

Sujan Sarker

November 2017

Table of Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Introduction

The rapid growth of mobile computing devices, such as smartphones and tablet computers equiped with embedded sensors (e.g., Global Positioning System (GPS), Camera, etc.) [1] and seamless Internet connectivity through Wi-Fi/3G/4G/LTE interfaces, has facilitated development of Mobile Crowdsourcing Systems (MCSs). An MCS is a distributed computing system that outsources sensing tasks to numerous mobile devices for collecting ubiquitous data to a central system and for sharing analyzed and processed information among many other potential users. Emerging applications of MCS includes environmental monitoring, social networking, health care and transportation safety, etc. [1]). For example, a groups of researchers from University of California at Los Angeles (UCLA) developed PIER (Personal Environmental Impact Report) system [2], which uses location data sampled from mobile phones everyday to calculate personalized estimates of environmental impact and exposure; Nericell [3] and VTrack [4] provide real-time traffic information; and, www.sensory.com offers free access to 100 community-powered coverage maps for various wireless networks (3G/4G/WiFi). In these applications, data are collected from devices using a free application; more mobile phone sensing applications can be found in [1], [5], [6], [7].

As shown in Fig. 1.1, an MCS system typically consists of a cloud platform (P), data requesting (R) devices and worker (W) devices [8], [9]. The platform P is a cloud service

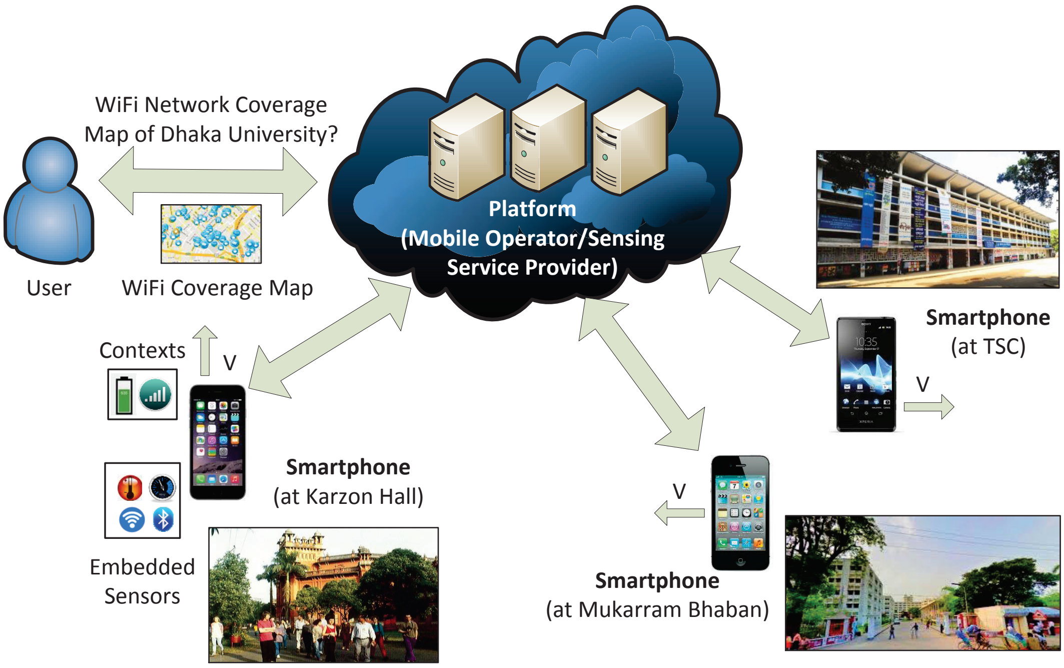


Figure 1.1: A mobile crowdsourcing system

provider that recruits worker devices (Ws) to sense and send data; the collected data are then processed at the platform and summary information on designated sensing and monitoring tasks are then delivered to Rs. The platform charges money from Rs for the delivered services and pays to Ws for performing sensing tasks.

The success of such a system greatly relies on the quality of data sensed by worker devices. Ensuring quality of sensed data in practical MCS systems is critical since it requires selection of worker devices offering higher utilities. The selection problem is further intensified due to mobility of worker devices and their service qualities in previous stages, especially for location-aware MCS applications. However, when a worker device performs sensing tasks, it incurs some costs such as power consumption for diving sensors, CPU utilization, etc[10]. Workers with higher utilities might claim more costs for rendering quality services. As a result, the profit of the platform may be reduced for ensuring better services. Thus, a balance between the quality of sensed data and the platform profit is required to maintain MCS ecosystem.

1.2 Motivation

Nowadays the proliferation of embedded technology results in the rapid growth of mobile devices such as smartphones, tablets, smart watches, etc. For mobility support these devices can be used as a portable computer for undertaking heavy computational tasks. Moreover, these devices come with a set of embedded sensors like Global Position System (GPS), accelerometers, cameras, etc., which enables sensing in diverse domains. On the other hand, the utilization of mobile devices is ubiquitous. According to studies [11], [12], almost 77 % of adults own a smartphone and 51 % of adults own a tablet in America as of November 2016. All the above conditions along with the advancement of mobile Internet and social networking technologies have facilitated the scope of crowd-problem solving using mobile devices and the traditional Internet Crowdsourcing has evolved into a new paradigm, i.e., Mobile Crowdsourcing (MCS). MCS involves the increasing number of mobile devices with high computational and sensing resources to perform crowdsourcing tasks. Different from the Internet crowdsourcing, mobile crowdsourcing leverages both sensory data from mobile devices (offline community) and user-contributed data from mobile social networking services (online community). Moreover, MCS extends user participation in crowdsourcing tasks from explicit participation to implicit participation. As a result, a number of crowdsourcing tasks that were previously difficult to complete in Internet crowdsourcing have now become feasible, e.g., monitoring pollution level or noise level at the city-scale, predicting the arrival time of vehicles, collecting the truth happenings after a disaster, etc. Meanwhile, mobile crowdsourcing also brings a number of challenges such as designing efficient framework to support MCS, allocating sensing task to workers considering their spatial and temporal availability, ensuring the quality and accuracy (e.g., sensing quality, coverage quality, etc.), designing incentive mechanism to motivate MCS participants, evaluating and recording participants past sensing records, considering the profit of the provider, trade-off issues (e.g., trade-off between profit and quality). As a result, MCS has drawn the attention of the researchers as an emerging field. Moreover, challenges available in MCS have also fascinated the application developers to concentrate on developing new crowdsourcing applications.

1.3 Crowdsourcing

As illustrated in Fig. 1.2, crowdsourcing is the practice of engaging a large group or crowd especially from an online community for obtaining services or contents - often innovation, problem solving, or efficiency. In [13], crowdsourcing is defined as a distributed problem solving model which engages a undefined size of crowd through a open call. The term crowdsourcing was coined by [14] as a form of ”peer production” that outsources works to a large group of people. Crowdsourcing provides a way to solve the problems which human can solve easily but difficult to computer. As a result, soliciting the solution of various tasks using online labor markets has increasing become popular in recent years.

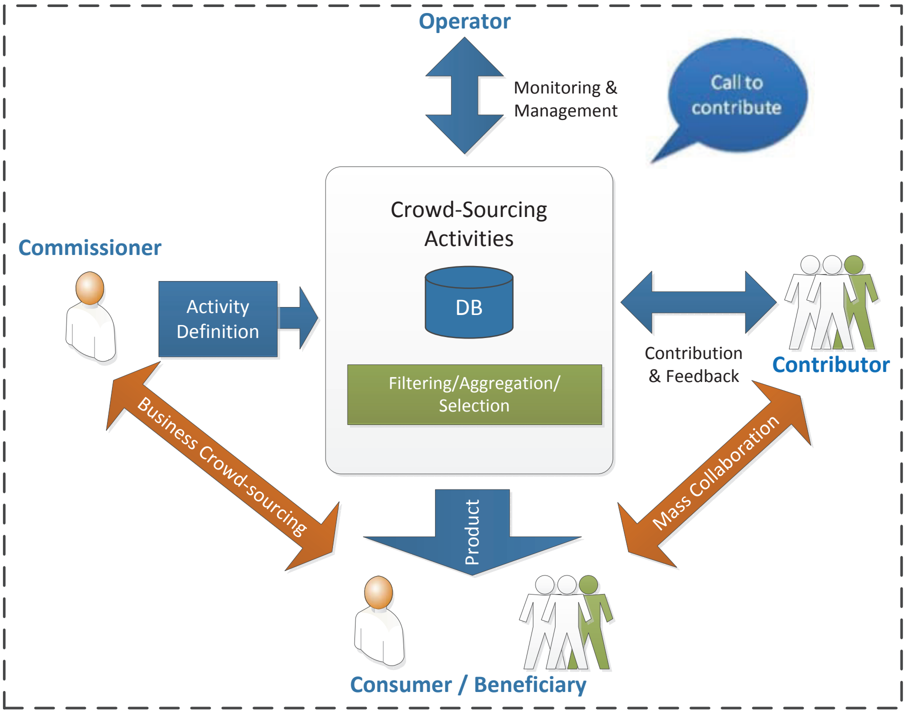


Figure 1.2: Main participants and abstract data flow in a crowdsoucing system

1.4 Mobile Crowdsourcing

Crowdsourcing has still not fully penetrated in mobile workspace; However Nowadays, smartphones are ubiquitous and widely used around the world with embedded sensors (e.g., GPS, accelerometer, camera etc.) and has seamless Internet connectivity (e.g., Wi-Fi, cellular, etc.). This trend enables individuals to sense, collect, process and distribute data around people at any time and place. Naturally, The mixing smartphone based mobile technologies and crowdsourcing offers vast resources of computation, and leads to a new paradigm called Mobile Crowdsourcing (MCS). Smartphones offer a great platform for extending existing Web-based crowdsourcing applications to a larger contributing crowd, making contribution easier and omnipresent. Furthermore, smartphones multisensing capabilities including geolocation, light, movement, and audio and visual sensors offer a variety of new, efficient ways to opportunistically collect data, enabling new crowdsourcing applications. A general architecture of MCS system is shown in Fig. 1.3.

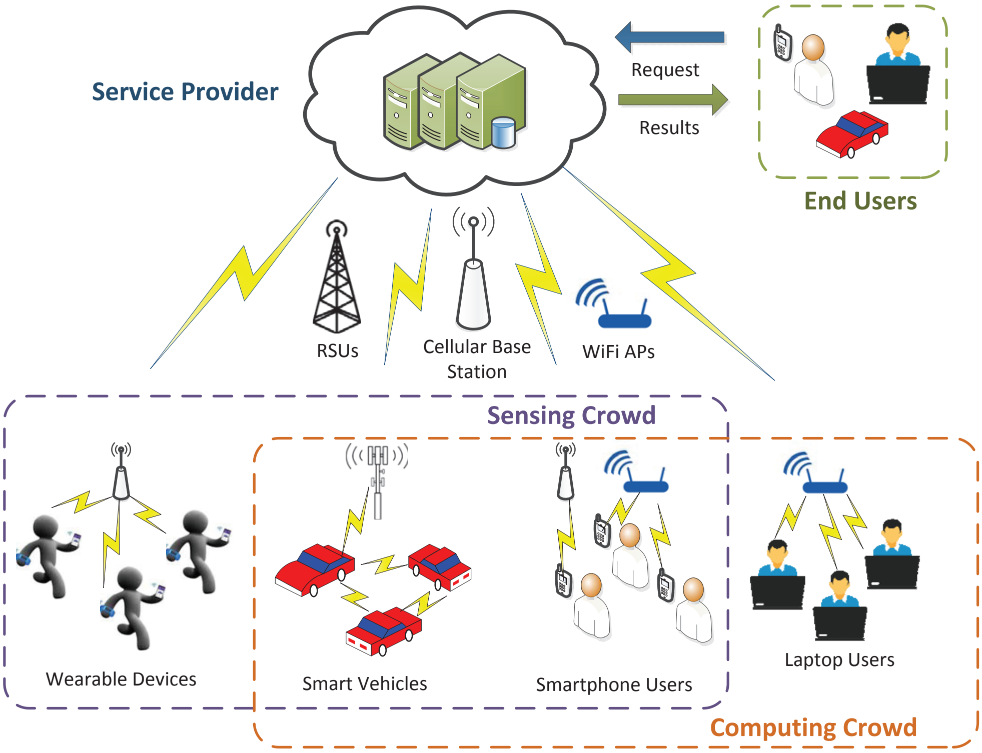


Figure 1.3: A General Architecture of MCS system

1.4.1 Entities in a MCS system

Generally, a MCS system includes three basic entities: service provider, end users and the crowd (sensing crowd and computing crowd) [15].

Service provider, also known as crowdsoucer, crowdsoucing platform or cloud platform provides crowdsoucing services to both end users and public cloud. Service provider receives service requests from the end users and partitions these task into small subtasks which can be crowdsourced. Tasks are then crowdsourced among the crowd, results are collected and processed information is provided to the end users. In some cases, service provider is only responsible for publishing tasks to the crowd while other processing are done by the requesting end users. Fig. 1.4 illustrates deferent functional components of the service provider. The requested task is first decomposed by the task decomposition component and then distributed among the crowd by the task distributor. After collecting the results final processing on the results is done by the task recomposition component. Task recommendation components takes care of users preferences and recommend tasks accordingly.

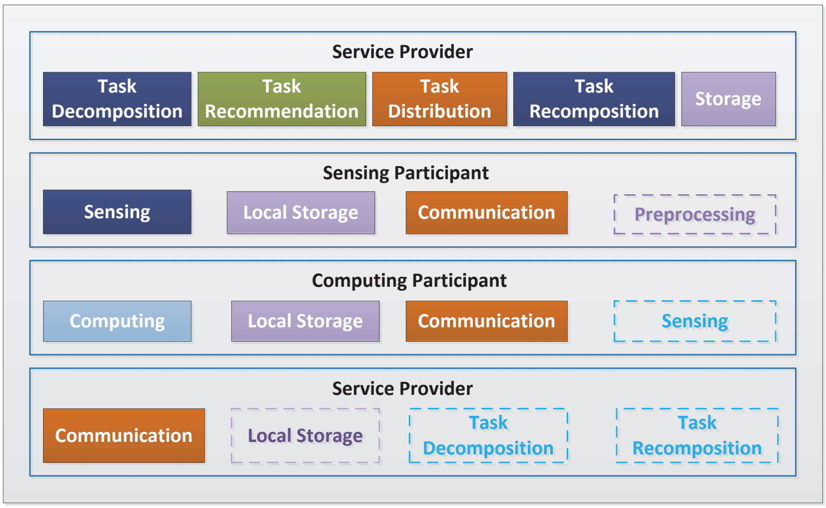


Figure 1.4: Component of each entity of a MCS system

End users are the customers who purchase crowdsourcing services at certain cost. They submit service requests to the service provider and get the results from it. In some cases, the end users may decompose and recompose the tasks by themselves and rely only on service provider for crowdsoucing their taks.

The crowd is a group of mobile users who accepts and participate crowdsourced tasks. The crowd can be broadly divided into two major types according to the type of tasks performed by them - the sensing crowd and the computing crowd. The sensing crowd includes a group of mobile users who performs crowsourced sensing tasks. On the other hand, the computing crowd involves the group of users in computing task. Fig. 1.4 illustrate components of both the sensing crowd and the computing crowd. Both of them include communication component which facilitates communication with the provider using cellular networks, WiFi, Bluetooth, NFC, and others. The sensing crowd optionally have the preprocessing component but exclude computing component. Similarly, the computing crowd optionally includes sensing component.

1.4.2 MCS Framework

A generic mobile crowdsourcing framework is proposed in [13]. As shown in Fig. 1.5, this framework consists of multiple functional modules that are independent of specific applications and can accommodate multi-modal data sources. We explain each module in detail as follows:

- *Task management* module characterizes the sensing specifications and use cases, including the types of participants, the required sampling rate for each type of sensors, the requirements of data visualization and representation, etc.
- *Mobile crowdsourcing frontend* module provides crowd (participants) with a cross-platform user interface for reporting crowdsourced data in participatory and/or opportunistic ways.
- *Crowdsourcer* module publishes the appropriate tasks to platform through interacting with task management module; provide quality feedback about the sensed contents offered by crowd; and pay the crowd for participating in crowdsourcing tasks.

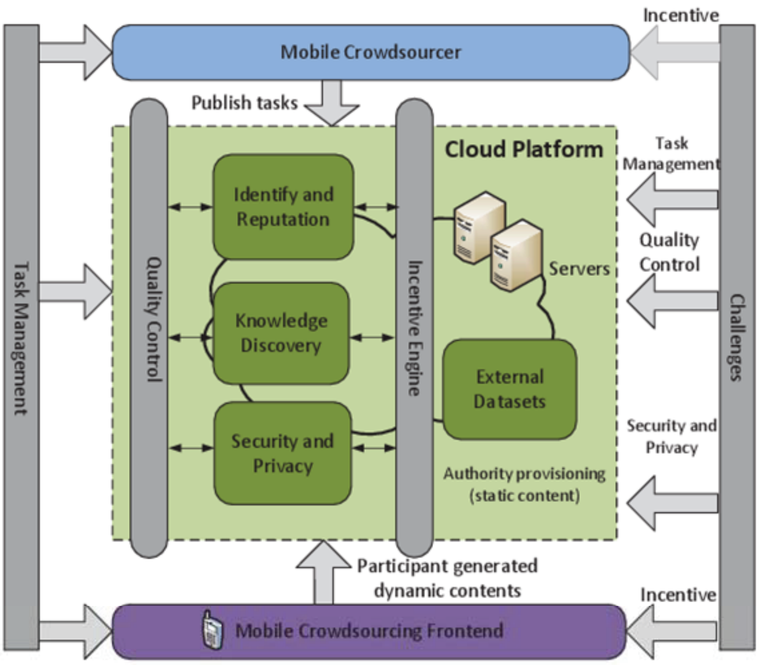


Figure 1.5: A generalize framework of MCS system

- *Incentive engine* generally stimulates crowd to actively participate and contribute high quality data and encourage crowdsourcers to provide truthful feedback about the quality of crowdsourcing data. It may reward/punish crowd and crowdsourcers with monetary, ethical, entertainment, and priority, etc., based on their behaviors.

- *Identity and reputation management* module manages the identities of participants and crowdsourcers, build reputation for them based on their past behaviors so as to enhance the quality of sensed data provided by crowd and trustworthiness of feedbacks by crowdsourcers.

- *Knowledge discovery module* provides intelligent data processing capability to extract and re-construct useful information from the raw sensing data submitted by participants.

- *Security and privacy module* aims to protect user's privacy, increase privacy awareness of users and make application validated.
- *Quality control module* analyzes the feedback of quality information and adjusts the parameters of the module other modules, to achieve a higher quality service.
- *External datasets* module allows sensing data from external datasets to be incorporated into the system to enrich the functionalities and services.

1.4.3 MCS Applications

Recent years have witnessed the proliferation of smarthphone based crowsourcing applications. According to [13], MCS applications can be divided into two categories: human intelligence and human sensor. The category of human intelligence utilizes human wisdom (wisdom of crowd) to perform tasks that are easy for humans but remain difficult for computers. Especially, mobile phones can reveal crowdsourcing's full potential and enable users to transparently contribute to complex and novel problem solving, such as knowledge sharing, natural language processing, etc. On the other hand, the category of human sensor incorporates the concept of "human-as-a-sensor" into the mobile crowdsourcing system to collect human observations, in addition to sensor measurements from mobile devices owned by the public crowd, for various services such as smart transportation, environment monitoring, etc.

A number of smartphone applications have been developed which use embedded sensors of the smartphone and exploit advantages of either opportunistic sensing or collaborative sensing or both. Here we discuss some area of MCS application along with the existing smartphone applications in these areas.

1.4.3.1 Environment monitoring

For monitoring environment condition such as detecting noise level in a city, measuring air pollution level, etc., mobile crowdsourcing can be used. Rajib Kumar Rana et al. [5] presents the design, implementation and performance evaluation of an end-to-end participatory urban noise mapping system called *Ear-Phone*. The key idea is to crowdsource the collection of environmental data in urban spaces to people, who carry smart phones equipped with sensors and location-providing Global Positioning System (GPS) receivers. Similarly *NoiseTube* [16] constructions fine-grained noise maps using uploaded data captured by users smartphone microphones. *PIER* (Personal Environment Impact Report) [2], developed by a groups of researchers from University of California at Los Angeles (UCLA), uses location data sampled from mobile phones everyday to calculate personalized estimates of environmental impact and exposure.

1.4.3.2 Traffic monitoring and smart navigation

MCS applications can be used to update the traffic at the required location, provide real time traffic information by displaying user-generated reports on traffic, construction, and speed traps. Moreover, MCS applications facilitate planning route according to weather conditions, accidents, and traffic jams. For example, [17] presents *TrafficInfo*, a smart phone prototype application implementing a participatory sensing based live public transport information service, which exploits the power of crowd to gather the required data, share information and send feedback. *TrafficInfo* also visualizes the actual position of public transport vehicles with live updates on a map, and gives support to crowd sourced data collection and passenger feedback. *Nericell* [3] and *VTrack* [4] provide real time traffic information such as road-traffic delay estimation. *PotHole* [18] helps users to identify holes of the streets by sharing their location and vibration data captured by the smartphone. *NeviTweets* [19] allows user to generate and share geo-tagged image traffic report. Based on these reports, Traffic Digest are delivered to driver to provide reliable information supporting the route choice. *TeleEye* [20] helps user to inquire information regarding location. User will get the answer through other user at the inquired location.

1.4.3.3 Smart parking and smart traffic light

MCS applications enable monitoring of parking space availability in the city and recommend with charges. For example, *Advanced services smart parking* [21] helps users to search available parking space closest to their intended destination. Real time traffic load and emerging events can be monitored using crowdsourcing data and these information can be used to control traffic lights. *SignalGuru* [22] provide advisory services for collaborative traffic light scheduling.

1.4.3.4 Health monitoring and disease diagnosis

Health status information such as heart rate, electrocardiography, blood pressure, etc, are collected using MCS application. These personal health parameters are then used to diagnosis different diseases. MCS application also facilitates food recommendation according to health conditions of a user. MCS data can also be used to monitor the water quality and study its eligibility for drinking.

1.4.3.5 Weather monitoring and hazard management

FloodPatrol [23] aims at contributing to flood monitoring and public awareness by allowing the crowd (user) to report the flood levels in various locations. *Hazard Reporting* [24] allows users to record hazard whenever they found one and uploading them to a server which enable municipalities could collaborate with utility companies in fixing such hazards in timely manners.

1.4.3.6 Disaster reporting

Project Jagriti [25] helps people make a report of child abuse. The reports are forwarded to the Child Welfare Committee. *iShake* [26] enables users to report ground motion measurements from a mobile device to a centralized data-collection-system when the mobile devices are triggered by a shaking event. *CrowdHelp* [27] helps user by displaying on the map with suggestion on type of help that is most needed. User within the radius of a natural disaster are able to send text, pictures, videos, locations, and descriptions of what they see.

1.4.3.7 Social networking

crowdSMILE [28] is a Crowdsourcing-based Social and Mobile integrated system for learning by exploration. Users can access location-based learning content anytime and anywhere. One of the most fundamental parts of the linguistic pipeline is part-of-speech tagging (POS), a basic form of syntactic analysis which has many applications in natural language processing. A special mobile crowdsourcing system [29] is proposed to address the problem of POS for English data from the popular micro bogging service Twitter.

1.4.3.8 Making 3D models

Mobile 3D Modeler [30] allows users to create, submit and vote 3D models of building components. Helps user to build a 3D model of the internal structure of a building. *IndorCrowd* [31] helps indoor 3D maps reconstructions. Users can help by capturing their preference indoor environment and uploading key frame to the cloud, along with real time sensory data and labeling information.

1.4.3.9 Other MCS applications

Jeffrey P. Bigham et al. [32] presents VizWiz, a project aimed at enabling blind people to recruit remote sighted workers to help them with visual problems in nearly real-time.

Blind people can use VizWiz on their existing camera phones: take a picture with their phone, speak a question, and then receive multiple spoken answers. *Librorium* [33] is a fully informational dictionary with accurate definitions, ratings, and example for every Filipino and English words. Allows user and expert can interact with each other. *UbiAsk* [34] provides translation services. User can upload an image containing different language. The other users that know the language will response to the image. *AirPlace* [35] provides real-time fine grained indoor localization services that exploit the radio signal strength of Wi-Fi access points.

1.5 MCS Challenges and the Scope of the Work

Most of the existing MCS applications rely on user's voluntary contribution. However, while participating in crowdsourcing sensing tasks a mobile user consumes it's own resources (e.g., battery, cellular data, memory). Thus incentive mechanisms are required to motivate users to participate in MCS system. However, it is extremely challenging to select high utility workers from the participants as they may be dishonest, selfish, erroneous or even malicious. The selection problem becomes further intensive due to worker mobility and location information. The platform should also make enough profit to get incentive to run MCS application in the long run. These issues make the MCS as an promising research area. Some of the challenges of MCS is listed below:

- *Task management:* Task design is the model under which the requester prescribes his or her task. It is an important requirement as tasks are heterogeneous in terms of required sensing service, task location, etc,. Factors that contribute to quality of task includes task definition, user interface, and granularity, etc,.
- *Provide incentive for the participants:* An incentive is a kind of stimulus or encouragement to stimulate one to take action, and work harder, etc. This issue is even more critical when the devices (e.g., mobile phones, wearable sensors) have

very limited resources (e.g., energy and storage capacity). Thus lucrative incentive mechanism is required to attract the users to participate in crowsourcing tasks. For example, the MCS system can reward the participants for contributing with high quality data.

- *Sensing quality control:* MCS system should be able to assess the quality of data sensed by the workers. As the quality sensing service depends on high quality data send by the workers, the MCS system should select hight utility workers to render quality services. Sensing quality varies depending on application requirements including quality of the sensor, coverage quality, etc. MCS should also be able to record workers past sensing reputations and treat the workers accordingly.

- *User's spacial and temporal availability:* Mobile users accept crowdsourced tasks based on their interests, locations, or device conditions (residual battery, available sensors, etc.). The network topology also changes over time due to human mobility and dynamic user join/leave. Thus workers spacial and temporal availability should be incorporated in the mechanism design.

- *Profit-aware mechanism:* MCS system should be profitable for both the participants and the provider. Without enough profit they will lose their motivation to put enough contribution in MCS system.

1.6 Problem Statement

Most of the research works on MCS system focus on designing incentive mechanisms to ensure participation of the users in crowdsoucing system. Some works incorporated worker utility (i.e., sensing quality) and tried to maximize the total utilities of the selected workers. SACRM [36] quantified worker utility as a function of required task completion delay, past sensing reputation and worker preferences. ABSee [37] measures sensing quality as the deviation of data qualities sensed previously by the selected workers. However, the main challenge in MCS is to recruit workers for distributing sensing workloads among them, since the system is highly dynamic due to the mobility of the workers and completion of sensing tasks are not guaranteed for the temporal and spatial availability of the worker and tasks have explicit delay deadline before which requesters must be served. Task heterogeneity in terms of task's Area of Interest (AOI), required sensing resources, etc., makes the problem further intensive. On the other hand, crowd platform aims to maximize its profit from the crowdsourcing system. A few of literature works concentrate on maximizing the profit of the platform while developing worker selection strategies [9], [38], [39]. However, profit of the platform depends on the claimed cost of workers for performing sensing task and market value of the corresponding sensing service. The claimed cost of a worker can't be known in advance and market value estimation requires extensive market analysis and learning. Thus incorporating profit in MCS mechanism design introduces another challenge.

None of the existing workers consider the platform profit and the sensing quality jointly while developing MCS system. As these two metrics are crucial for proving longterm high quality sensing services it requires making a balance between these two. Trade-off mechanism should also be adaptive with changing environment while ensuring the marginal requirements of each of the two metrics. In this thesis, we try to answer the following three questions.

- How to maximize quality of sensed data while fixing a profit margin of the platform?
- How to maximize profit of a platform while keeping the required quality of sensed data for MCS applications?
- How to make a reasonable trade-off in between the above two performance metrics?

1.7 Solution Methodology

In Fig. 1.6 we present the abstract view of our proposed solution methodology. In this work, we have proposed a framework for the cloud platform which is applicable for any kind of MCS applications. we design different modules responsible for performing different functionalities and define the interactions among them.

Our proposed model takes heterogeneous task requests from the service requesters and define workload of tasks. Then tasks are advertised to the crowd. After receiving bids from the workers, our model calculates profit of the provider and expected utility of the worker's bid for the tasks. Then our model selects the most suitable workers according to the strategies of the platform. After that workers contribution is evaluated and reward are given accordingly.

1.8 Thesis Contributions

In this work we investigate the problem of fundamental tradeoff between the quality of sensed data and the platform profit, named as PQ-Trade for location aware MCS applications. We quantify quality of sensed data using a worker's utility, which is expressed as a function of it's mobility, distance from the centre location of the application task and quality of past sensing responsibilities. On the other side, profit of a platform for each sensing task is the difference of the amount it receives from the data requesters and the amount it pays to the sensing worker devices. The proposed PQ-Trade system optimaly selects workers to make a good balance between worker utility and platform profit. The main contributions of this paper are itemized as follows:

- We develop a utility function for a worker based on its mobility, current location and past reputation.
- Multi- Objective Nonliner Programming (MONLP) objective function (with nec-

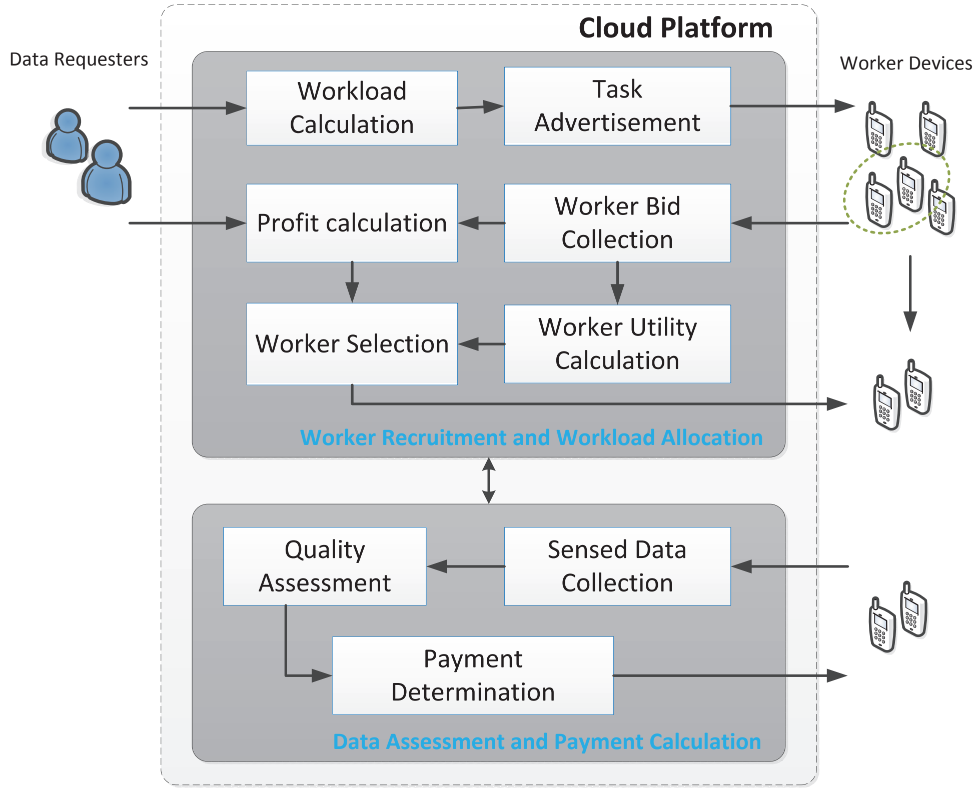


Figure 1.6: Architecture of proposed system

essary constraints) has been formulated for selecting worker devices to make a reasonable balance between platform profit and their utilities.

- Due to NP-hardness of the MONLP-based optimal solution, we then develop first-fit greedy algorithms either to maximize utility or profit while keeping the other one at a desired level.
- A payment policy for the selected worker devices has been developed following the quality of sensing data.
- We implement the proposed system using MATLAB and compare with the state-

of-the-art works.

1.9 Organization of the Thesis

The rest of the report is organized as follows. The state-of-the-art works on worker selection in MCS have been described in details in chapter 2. The limitations of the existing works are also specified here. In chapter 3, the assumptions, notations and the proposed PQ-Trade system has been demonstrated.The performance evaluation of our proposed task scheduling model compared to the sate-of-the-art works has been demonstrated in chapter 4. In chapter 5, conclusion along with the directions for future research has been outlined.

Chapter 2

State-of-the-art Works

In this chapter, we overview the necessary backgrounds on MCS system and their design principle. Some state-of-the-art research works are described in detail. Furthermore, we focus on the motivations that lead us to develop a novel architecture model and worker recruitment and sensing workload allocation framework for MCS system.

2.1 Introduction

Recently, as an emerging field of information collection mechanism MCS sensing crowdsourcing in mobile has been studied extensively by the research communities. Today smartphones are the core communication decives in people's everyday life. Issues relating both MCS systems and MCS applications design have drawn equal attention in the literature. A number of the state-of-the-art works focused on designing practical MCS systems. For the selecting suitable workers for crowdsourcing tasks among them. Most of related works focuses on designing incentive mechanism to stimulate smartphone users to participate in crowdsourcing tasks. In a practical crowdsourcing system, thousands workers may coexist. They are also heterogeneous in terms of their social attributes (i.e., location, mobility, etc.), sensing and computing resources, etc. On the hand, service requesters also have heterogeneous task demands in terms of required sensing or computational services, service location, deadline, budget, etc. Thus selecting a suitable set of workers and allocating sensing workloads is a critical one in such a dynamic environment introduced a crucial problem. The selection problem is further intensified due to users spacial and temporal availability, sensing quality, reputation of users past sensing responsibilities, platform's profit from the crowdsourcing system,etc. A good number of literature works can be found which consider either quality of sensed data or profit of the crowdsoucing platform. However, none of the existing works focuses on the trade-off issues of these two parameters. Moreover, evaluating sensing reports, managing users reputation accordingly and making adaptive payment policy have got less importance in the literature.

In this chapter, we first discuss existing research works related to MCS system design and application developments. Then we extend our study on the existing works which incorporate platform profit, sensing quality and adaptive payment policy into their design. Finally we conclude this chapter discussing the uniqueness of our work.

2.2 MCS System and Application Design

A crowdsourcing system mainly consists of a crowd platform, service requester and the crowd. On arrival of requests from service requesters or customers, a MCS platform outsources the sensing tasks among the crowd (a group of users who participate in crowdsourcing tasks) by an open call. The crowd performs assigned tasks and sensing or computational results are sent to the platform. The platform then provide the customers required services. To manage the interaction among the these entities and enhancing the quality MCS services, works have been done proposing sustainable architecture MCS design. In [15], a generalized architecture of a MCS system. The functional components of each entity along with functionalities have been presented in detail. Authors in [13] proposed a generic framework for MCS system and application design along with the challenges. An extensive study on the current state of crowdsoucing applications that exist today, in done. According to the study MCS has gain popularity both in research community and industry.

A number of MCS applications in specific fields can be found in the literature including traffic and navigation [17], [4], [3], [19], [20], environment monitoring [2], weather monitoring [23], disaster management [25], [26], [27], social networking [28], etc. More crowdsoucing applications in different fields can be found in [22], [24], [29], [30], [31], [32], [33], [35]. However, most of them considered the sensing tasks as a volunteer service. User participates in crowdsourcing tasks without any payment or reward which is not viable in a practical crowsoucing application.

2.3 Incentive Mechanisms Design for MCS System

While participating in crowdsourcing tasks smartphone users incur some cost such as power consumption for driving sensors and CPU utilization, bandwidth cost for communicating with platform and submitting sensing task. As a result, MCS system should ensure enough compensation in monetary or other form to stimulate workers. On the other hand, a adaptive rewarding polity is required to encourage workers with good contribution and discourage dishonest workers with penalty. Thus designing incentive mechanism with lucrative payment policy is one of important consideration while designing MCS systems and applications.

Several incentive mechanisms have been widely studied in the literature [8], [9], [39], [40], [41], [42], [43], [44], [45]. Most of them exploit dynamic pricing to give incentive to the participants. Auction theory (mainly reverse auction) is used to achieve the properties of truthfulness, individual rationality and computational efficiency in selecting crowdsourcing workers [8], [42], [44], [46], [47]. The auction theory ensures truthfulness and individual rationality of the workers. Truthfulness is required to encourage user biding with their real cost. Individual rationality is required to give incentive to the worker for good contribution.

Yang et. al. [9] proposed two incentive mechanisms called MSensing for a user-centric and a platform centric model, respectively. For platform centric model, a Stackleberg game based incentive mechanism has been designed. In MSensing, each user decides the amount of time it would like to spend in sensing tasks and gets its proportionate amount from the declared total reward. On the other hand, for user centric model, an auction based truthful mechanism has been proposed which assigns sensing tasks to the users using a greedy-based heuristic approach. However, in MSensing, a user can only submit a single bid, thus it fails to make the best use of multi-sensing capabilities of users.

TRAC [8] is a truthful auction mechanism for location-aware sensing in MCS. In TRAC, the platform assigns sensing tasks with a goal to minimize social cost of the selected users. A user bids for a subset of sensing tasks if it is within the Area of Interest (AOI) of those tasks. TRAC platform then sorts tasks based on minimum cost per unit task and greedily assigns tasks to the workers. A critical payment based payment mechanism has also been proposed to pay the workers according to their marginal contributions. However, TRAC does not consider the profit of the platform which is a must for sustainability of a crowdsourcing platform. TRAC also assumes that, there are sufficient worker for alternate bidding combinations thus all the crowdsourced tasks can be surely assigned which is unrealistic in a practical application scenario.

A truthful incentive mechanism has also been proposed in [46] to minimize the social cost of the crowdsourcing workers. Two different working patterns of a task have been considered: continuous and discontinuous. In continuous working pattern, workers can only bids for subtasks in continuous time slots. For this kind of patterns, a Vickrey-Clarke-Groves (VCG)-based auction mechanism is proposed and worker selection and task allocation problem is solved using a dynamic programming solution. A suboptimal auction mechanism is introduced for the discontinuous patterns which selects workers according to minimum bidding cost using a greedy sorting based mechanism with a aim to minimize the social cost. The Platform also pays the selected workers according to the critical payment mechanism of auction theory. However, the worker selection and task allocation mechanism is run only for one task in each round thus it fails to serve heterogeneous task requests from the users. Moreover, platform overlooks workers spatial and temporal availability as well as the quality of the data sensed by them.

2.4 Profit Aware MCS Systems

Although several auction mechanisms have been proposed for MCS system, most of them do not consider the profit of the mobile crowdsourcing system (e.g., [8], [46], [44], [45]). However, profit is crucial especially the profit of the MCS platform for designing a sustainable mobile crowdsourcing system. It gives incentive to the platform for running MCS application. In literature, there are few works which try to incorporate profit of the crowdsourcing system into their design [9], [39], [38].

In Msensing [9], tasks are assigned gradually to smartphone users by a greedy-based heuristic approach with a aim to maximize the profit of the platform. However, profit is sacrificed for the sake of computational efficiency of the system. In [38], Koutsopoulos designs an optimal incentive mechanism to maximize the profit of the platform. However, the winner determination problem is NP-hard and cannot allocate the tasks in a computationally efficient manner.

PROMOT [39] is a reverse auction based incentive mechanism that outperforms the profit of MSensing. In PROMOT, each task has some positive value (i.e., the market price of the sensing services) and platform profit is calculated as the difference between this value and the bidding cost of the workers. PROMOT winner determination problem is formulated as an Integer Linear Programming (ILP) problem and due to NP-Hardness of ILP problem, it is converted to a Linear Programming (LP) problem. Based on the solution obtained from LP problem, users are selected greedily to maximize the profit of the platform. However, PROMOT platform does not consider the utility of the recruited workers (i.e., sensing quality of the service). PROMOT platform also overlooks crucial quality parameters of a worker such as worker's current location, mobility, past sensing reputation, etc. In some works mechanism is proposed for mobile crowdsourcing while a limited budget is assigned for sensing tasks and platform performs a subset of tasks according to its budget constraint [48], [48], [49].

2.5 Quality of Sensing Aware MCS Systems

Quality of sensing is also linked to a few existing incentive mechanisms [38], [50], [49], [51], [52], [37], [36]. Koutsopoulos [38] proposed an incentive mechanism which considers the quality of sensing. However, the platform does not have a budget constraint and only has one task. In [50], an auction mechanism is designed to maximize the platform's valuation where quality of sensing of each user is assumed to be known by the platform. A sequential Bayesian approach is used in [49] to determine the quality of sensing of users, but they assume that quality of sensing is previously known and tasks have only binary values. In [51], a quality-aware algorithm is proposed only considers the coverage quality. A quality-based incentive mechanism is designed in [52]. However, it does not consider the strategic behavior of users.

ABSee [37] is a quality of sensing aware budget feasible mechanism which introduced quality indicator to estimate the sensing quality of the users under a budget constraint. In ABSee, truth value of the sensed data is estimated using truth discovery methods proposed in [53] and the quality indicator is calculated from the deviation of sensed data from the truth value. Then previously stored quality indicator is updated using EWMA and new value is stored. This value is used as an quality indicator in the next worker selection round. However, ABsee only considers the quality of sensing based on workers past sensing accuracy while critical quality parameters such as workers location and mobility information are not considered. ABSee also runs worker selection algorithm within limited budget. In [54], a recruitment framework is developed to enable the data requester to identify well-suited participants based on geographic and temporal availability which approximately maximizes the coverage over a specific area and time period under a limited campaign budget with a greedy algorithm. A recruitment framework is also propose in [55] to select suitable participants in the friend circle by the multi-hop friendship relations. However, none of them consider the social preferences of mobile users and adaptive rewards allocation.

SACRM [36] addressed the quality of sensed data by measuring utility of a worker which is a function of task completion delay, reputation of the past sensing responsibilities and overlap between social attributes of the workers and the tasks. SACRM allows the requesters to select a subset of users to maximize the total utility of selected users under a budget constraint. It also evaluates the quality of sensed data and provide payment to the selected workers accordingly. Worker's performance is recorded as the reputation value in a reputation database which is incorporated in the worker selection decision in the next round. Though SACRM considers some quality parameters of the workers, it overlooks worker mobility and current location information which is important for designing a location aware crowdsourcing application. In SACRM, platform only stores worker reputation information and facilitated interactions among the requester and the workers where task allocation and payment decision is entirely taken by the requesters. Another limitations of SACRM is that, the requester can run task allocation algorithm for only one task at a time which results in under utilization of worker's sensing resources.

2.6 Data Assessment and Reputation Management

Another challenge in MCS system design is sensed data assessment and worker's reputation management as well as evaluating the trustworthiness of the sensing data and mobile users. Some literature works introduce these issues in designing MCS systems

[36], [37], [56], [57], [58]. In [56], a robust trajectory estimation strategy named TrMCD is developed to alleviate the negative influence of abnormal crowdsourced users. A reputation framework for social participatory sensing is proposed in [59]. Huang et al. [60] employ the Gompertz function [61] to compute the device reputation score and evaluate the trustworthiness of the contributed data. SACRM [36] and ABSee [37] use existing truth discovery methods [53] taken from data mining. Privacy issues are also highlighted in the reputation system design of mobile sensing in some of the existing works [57], [58], [59].

2.7 Discussion

Most of the existing works focus on developing incentive mechanisms to stimulate smartphone user to participate in crowdsourcing tasks. However, many real time MCS applications demand high quality and accurate data from the MCS system. To sustain in the market a MCS platform should render high quality services which relies on selecting high utility workers. However, high utility workers demand high monetary reward for engaging their resources in sensing tasks. As a result, the platform losses its incentive due to poor profit made from the crowdsourcing system. Though some literature works considered profit of the MCS system, profit is sacrificed for the shake of computational efficiency. Moreover, data quality is overlooked for higher profit. A number of workers linked sensing quality to their design. However, most of them don't consider the spacial and temporal availability or past sensing reputations of the workers which are crucial for providing high quality services to the customers.

None of the existing works jointly consider the platform profit and worker utility while designing MCS systems. Therefore, we have design a worker recruitment and task allocation framework for MCS namely PQ-Trade which considers worker spacial and temporal availability to quantify the utility of the worker. Then PQ-trade formulates a Multi-objective Non-linear Programming (MONLP) problem to make a reasonable balance between the profit of the platform and utilities of the selected worker devices. Finally, PQ-Trade develop two computationally efficient greedy solutions which aims at maximizing either the profit of the platform or the quality of sensing while keeps the other one in a marginal level.

To best of our knowledge, proposed PQ-Trade system is the first one in the area of MCS that has considered the trade-off between the platform profit and the worker utility (i.e., quality of data sensed by them). Moreover, we have carried out the boundary performance analysis for sensing quality and platform profit.

A comparative study among different worker selection mechanisms in MCS discussed earlier is presented in Table 2.1.

Table 2.1: Comparison Among MCS Systems

MCS system	Platform profit	Sensing quality	Location awareness	Worker mobility	Worker reputation	Profit quality trade-off
M-Sense [9]	Yes	No	No	No	No	No
TRAC [8]	No	No	Yes	No	No	No
PROMOT[39]	Yes	No	No	No	No	No
MSC [46]	No	No	No	No	No	No
SACRM [36]	No	Yes	No	No	Yes	No
PQ-Trade	Yes	Yes	Yes	Yes	Yes	Yes

2.8 Summary

In this chapter, we have discussed some of the recent noteworthy contributions in MCS system focusing worker recruitment, workload allocation and sensing report assessment. We studied their working principles and tried to alleviate their shortcomings using our idea. As one of the major limitations of the existing mechanisms is ignoring trade-off issues of platform profit and sensing quality, our proposed PQ-Trade mechanism emphasizes on these fact during designing worker recruitment and workload allocation mechanism. Finally, we have discussed the uniqueness and importance of our work in the context of worker utility, platform profit and trade-off between these two. In the next chapter we discuss our proposed PQ-Trade system in detail.

Chapter 3

Proposed PQ-Trade System Design

In this chapter, we present our proposed workload allocation policy namely PQ-Trade system,in detail. We first discuss the system model and assumptions that have been considered in our system. Then we go through the design components of the proposed system and describe the functionalities of each components in detail.

3.1 Introduction

In the previous chapter we have explored the existing worker selection and workload allocation policies in MCS system. Due to the limitations of existing works and the challenges to establish better worker recruitment and workload allocation policy with necessary trade-off, it creates ample scope to do research on it. We have also seen that, most of the existing workload allocation policies do not consider worker mobility information or past sensing reputation. Thus we came up with an efficient workload allocation policy namely PQ-Trade to make a reasonable trade-off between two aforementioned parameters.

In this chapter, we first discuss the system model and assumptions that have been considered in our work. After that, we unfold different design components of our proposed system and finally we describe the work flow of our proposed workload allocation policy.

3.2 System Model and Assumptions

In this section, we describe different entities in a typical MCS system and illustrate the interactions among them.

3.2.1 Entities in Mobile Crowdsourcing System

We assume a crowdsourcing system consisting of three entities: a cloud Platform (P), data Requesters (Rs) and Worker devices (Ws). The platform initiates sensing tasks on reception of requests from customers (Rs) and then the tasks are being distributed among a number of worker devices (Ws) for execution.

Let $\mathcal{T} = \{t_1, t_2, ..., t_N\}$ denotes the set of N sensing tasks submmited by Rs. Each sensing task represents a specific service which has some value, $V_t > 0$ to the platform [39]. Each sensing task $t \in \mathcal{T}$ is characterized by a five parameter tuple $< \mathcal{A}_t, l_t, r_t, D_t, \mathscr{W}_t >$, where, $\mathcal{A}_t$ specifies the desired sensing service, l_t is the corresponding location from where the sensing data should be collected, r_t is the radius of tasks' Area of Interest (AOI), $\mathcal{D}_t$ is the time deadline and $\mathscr{W}_t$ is the workload of task t [62]. In PQ-Trade, platform divides a task into a number of subtasks of uniform size. The total number of subtasks for a particular task t defines workload of that task. For example, a task (e.g., environment monitoring application) of taking a total of 10 pictures at a particular Point of Interest (POI) and if uniform subtask is defined as taking 2 pictures at a time, then workload of this task is 5. We also assume that AOI of a task is given by a circle with radius r_t.

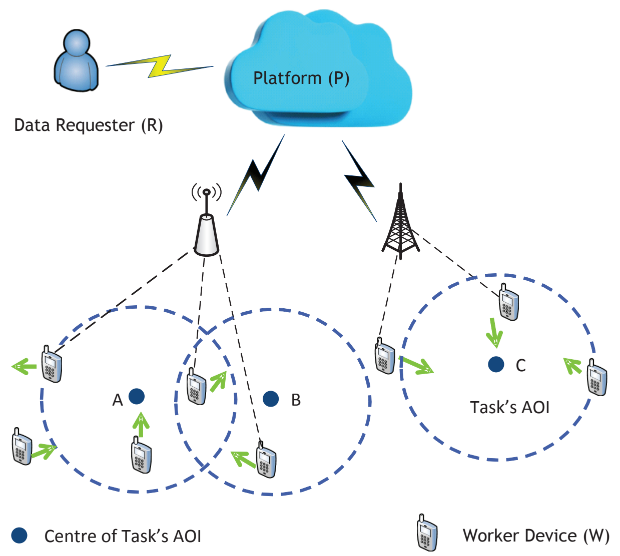


Figure 3.1: System model of a typical MCS application

We assume a set $\mathcal{M} = \{m_1, m_2, ..., m_M\}$ of worker devices that are interested to perform sensing tasks assigned by the platform P. A worker $m \in \mathcal{M}$ can perform a sensing task t if it is located within the radios of task's circle. The location of the worker can be extracted using GPS [63]. The lifetime of a worker in the AOI of a task depends on its velocity and movement direction.

3.2.2 Interactions Among Entities

Figure 3.2 shows the interactions among Rs, P and Ws and major events in one task scheduling round. The events r_1 - r_N are sensing service requests from a number of data requesters, Rs. On reception of such requests, the platform P advertises a sensing task set, $\mathcal{T}$ among all the worker devices, denoted by the events a_1 - a_N. Each worker $m \in \mathcal{M}$ then bids for a subset of tasks (One bid for one task) in events b_1 - b_M and let $\mathscr{B}_m$ denotes the set of all bids submitted by a worker $m \in \mathcal{M}$. Each bid, Γ_m^t is characterized by a five parameter tuple $< m, t, w_m^t, c_m^t, d_m^t >$ $(m \in \mathcal{M},\ t \in \mathcal{T})$, where, w_m^t is the offered workload, c_m^t is the claimed cost and d_m^t is the task completion delay. Each worker, $m \in \mathcal{M}$ also send a maximum number, n_m^{max} of winning bids to the platform. Though having heterogeneous sensors, a worker m may not complete all the tasks due to its availability in the task's AOI and deadline of the task which in turns decreases the reputation of the worker. Thus it is rational that the numbers of winning bids of worker m is limited by n_m^{max}. Let $\mathscr{B} = \{\mathscr{B}_1 \cup \mathscr{B}_2 \cup ... \cup \mathscr{B}_M\}$ denotes the set of all bids received by the platform. Platform then runs PQ-Trade worker selection method (to be presented in Section 3.3.2) in event e_{trade} and selects workers along with their winning bids. Then the selected workers are notified about their winning bids in events s_1 - s_N. Each worker then completes tasks in event $e_{sensing}$ for which it has a winning bid. After completion of $e_{sensing}$, sensed data are provided to the platform in events d_1 - d_W. The platform provides services to Rs in events d_1 - d_N. Rs pay for the services in events p_1 - p_N. In event $e_{payment}$, P determines the payment of each worker considering the quality of sensing data and calculate a payment vector. After completion of $e_{payment}$, platform finally pays the Ws for their contribution in events p_1 - p_W. Service request events r_{N+1} - r_{N+3}, which arrive after T_1 waits until T_2, starting of next scheduling round. The major notations used to design PQ-Trade are listed in Table 1.

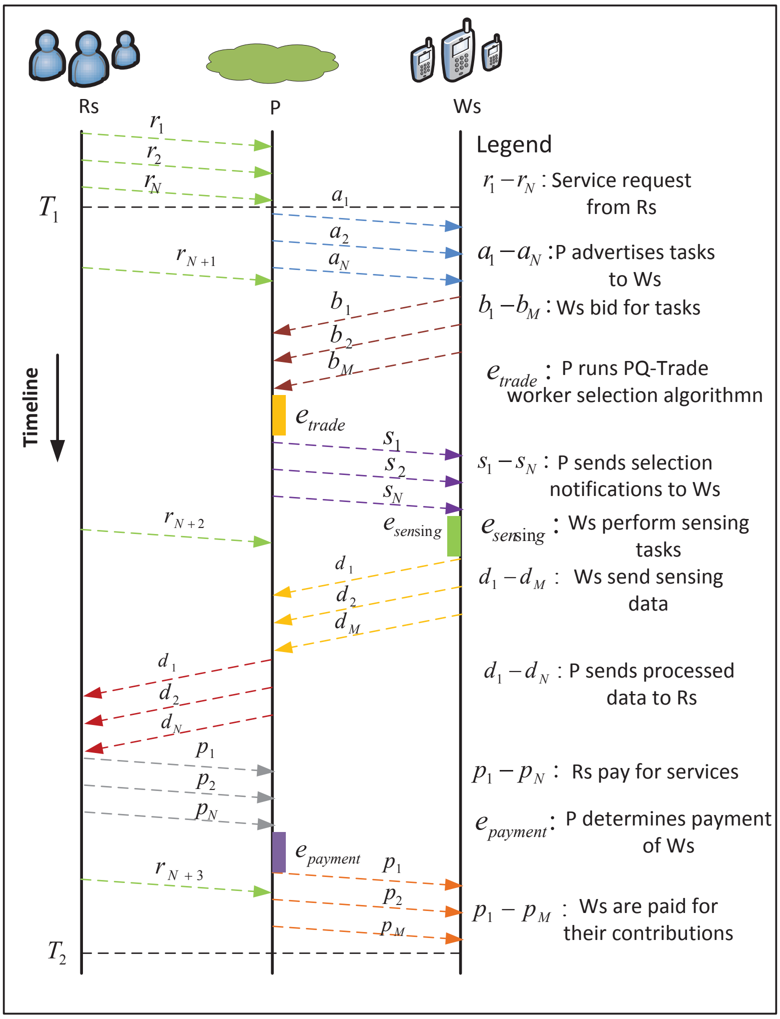


Figure 3.2: Interactions among Rs, P and Ws.

3.3 Design of PQ-Trade

In this section, we present the computational model of proposed PQ-Trade platform, formulate optimal worker selection problem as a MONLP, develop two greedy first-fit algorithms and a payment policy for worker devices.

Table 3.1: Notation Table

Symbol	Meaning
$\mathcal{T}$	Set of tasks advertised by the platform P
l_t	Location of a task $t \in \mathcal{T}$
r_t	Radius of task t's AOI
$\mathcal{D}_t$	Delay deadline of task $t \in \mathcal{T}$
$\mathscr{W}_t$	Total workload of task $t \in \mathcal{T}$
V_t	Value of task $t \in \mathcal{T}$
$\mathcal{M}$	Set of available worker devices
l_m	Current location of worker $m \in \mathcal{M}$
$\mathscr{B}_m$	Set of bids sent by a worker $m \in \mathcal{M}$
w_m^t	Offered workload of task t from a worker $m \in \mathcal{M}$
c_m^t	Claimed cost of worker m for a task $t \in \mathcal{T}$
d_m^t	Task completion delay for task $t \in \mathcal{T}$, $m \in \mathcal{M}$
$\mathscr{B}'$	Set of winning bids
$\mathscr{L}_m^t$	Sojourn time of worker m in task t's AOI
$\mathcal{U}_M$, $\mathcal{U}_D$, $\mathcal{U}_Q$	Workers' utility for mobility, location and past sensing quality, respectively
$\mathcal{U}_m^t$	Combined utility of worker m for task t
$\mathcal{V}_m^t$	Monetary value of workers' utility
$\mathcal{P}_m^t$, ρ_m^t	Platform total and normalized profit for allocating task t to worker m

3.3.1 Computational Model of PQ-Trade Platform

Figure 3.3 shows the functional modules of our proposed PQ-Trade platform, where an individual module is responsible to perform designated tasks, as presented below:

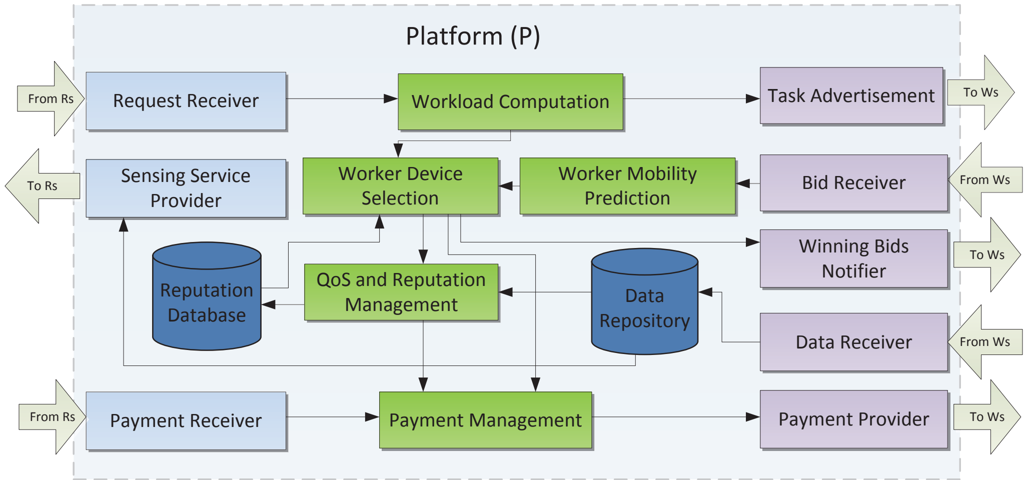


Figure 3.3: Computational model of PQ-Trade platform

- **Request receiver** module delivers summary of the task specification sheets, received from sensing service requesters Rs, to workload computation module. Each task sheet contains specifications of desired sensing services (e.g., location, type of sensing task, deadline of task, budget, etc.) [62].

- **Workload computation** module counts the number of unit subtasks in each task $t \in \mathcal{T}$ to quantify its workload, $\mathscr{W}_t$, as follows,

$$\mathscr{W}_t = \frac{\mathscr{V}_t}{\mathscr{S}_t}, \tag{3.1}$$

  where, $\mathscr{V}_t$ is the task volume and $\mathscr{S}_t$ is the size of unit subtask for t. Recall that each task $t \in \mathcal{T}$ is characterized by a five parameter tuple, stored in the task pool. The tasks wait in the pool till the next scheduling round starts.

- **Task advertisement** module, at the start of a scheduling round, pulls the set of available tasks, $\mathcal{T}$, from task pool of workload computation module and advertises the tasks to all worker devices, Ws.

- **Bid receiver** module, after advertisement of available tasks set, $\mathcal{T}$ among all the worker devices, waits for a certain amount of time, T_{bid} and forward the received bid set, $\mathscr{B}$ to worker mobility prediction module for further processing. Recall that each bid from a worker $m \in \mathcal{M}$ is characterized by a five parameter tuple and a worker can only submit one bid for each task $t \in \mathcal{T}$.

- **Worker mobility prediction** module extracts the bid tuple, Γ_m^t of each worker-task pair, (m, t) and calculates the sojourn time, $\mathscr{L}_m^t$ of a worker $m \in \mathcal{M}$ in the AOI of task $t \in \mathcal{T}$ using its location information and predicted velocity. In PQ-Trade, we exploit smooth random mobility model to calculate $\mathscr{L}_m^t$ for a worker device that can predict more accurately [64]. Finaly, this new parameter, $\mathscr{L}_m^t$ is added with each bid tuple, $b_m^t \in \mathscr{B}$.

- **Worker selection** module extracts the bid tuple, Γ_m^t of each (m, t) pair from $\mathscr{B}$, calculates worker's mobility, distance and past sensing quality based utilities, $\mathcal{U}_M$, $\mathcal{U}_D$ and $\mathcal{U}_Q$, respectively and combines these to calculate the total utilities, $\mathcal{U}_m^t$ for each (m, t) pair discussed in section 3.3.2.1. Worker selection module then calculates the corresponding profit, $\mathcal{P}_m^t$ and normalize profit, ρ_m^t of the platform and selects optimal workers, details are found in sections 3.3.2.2 and 3.3.3.

- **Winning bid notifier** module receives winning bids set, $\mathscr{B}^w$ from the worker selection module and notifies the workers about their winning bids.

- **Data receiver** module collects sensing data from the selected worker devices, Ws. After being notified by the winning bid notifier module, each selected worker device $m \in \mathcal{M}$ performs sensing tasks and submits data to the data receiver module. Data receiver stores the data into a data repository.

- **QoS and reputation management** module evaluates the quality of the sensed data provided by the workers $m \in \mathcal{M}$ using methods proposed in [37] and updates the standard deviation of sensing quality of the worker, q_m. The platform maintains a a database to record the q_m value of past sensing responsibilities of a worker devices $m \in \mathcal{M}$. Whenever a new worker registers, the platform creates a new entry in the database and gives the worker an initial quality value, q_m which is updated over time as tasks are assigned to the worker and sensing data are received by the platform. The details of calculation of q_m can be found in section, 3.3.2.1.

- **Sensing service provider** module provides processed data as services to the requesters, Rs. Data are represented in application specific format (e.g., creates WiFi coverage map, shows traffics in a map, etc.,).
- **Payment receiver** module charges money to data requesters, Rs for rendering services and Rs pay to the platform according to their service demand.
- **Payment management** module calculates the payment of each selected worker device $m \in \mathcal{M}$ using PQ-Trade payment policy after evaluating the quality of their sensing data and constructs a payment vector, $\mathscr{P}$.
- **Payment provider** module pays each selected worker device $m \in \mathcal{M}$ according to the calculated payment vector for their contributions.

3.3.2 Problem Formulation

In this section, we first quantify achievable utility, $\mathcal{U}_m^t$ of a worker device $m \in \mathcal{M}$ for performing a certain task $t \in \mathcal{T}$ and then we calculate the corresponding profit, $\mathcal{P}_m^t$ of the platform and finally, we formulate PQ-Trade worker selection problem as a Multi-Objective Non-Linear Programming (MONLP) problem.

3.3.2.1 Worker Utility

For each candidate worker $m \in \mathcal{M}$ and a certain task $t \in \mathcal{T}$, we compute an integrated utility metric, $\mathcal{U}_m^t$, which is a linear combination of three sub-metrics - $\mathcal{U}_M$, $\mathcal{U}_D$, and $\mathcal{U}_Q$

due to its mobility, location and past sensing quality, respectively.

Calculation of $\mathcal{U}_M$: For a location-aware crowdsourcing system, assignment of a task to a worker highly depends on the availability of the worker in the task's Area of Interest (AOI). Therefore, user's velocity and direction of movement jointly determine the available of the worker within the area. Thus, we introduce the sojourn of a worker, $\mathscr{L}_m^t$, defined as the amount of time the worker m is expected to stay in the AOI. Higher value of $\mathscr{L}_m^t$ indicates availability of the worker for a longer duration, which is beneficial for quality sensing.

The utility $\mathcal{U}_\mathcal{M}$ of worker m for performing task t based on its mobility pattern is quantified as follows,

$$\mathcal{U}_\mathcal{M} = \begin{cases} 1 - e^{(d_m^t - min(\mathscr{L}_m^t, \mathcal{D}_t))} & d_m^t < min(\mathscr{L}_m^t, \mathcal{D}_t), \\ 0 & otherwise, \end{cases} \tag{3.2}$$

where, $\mathscr{L}_t^m$ can be calculated using workers mobility information [65]. Note that, we use similar approch of [65] which calculates the sojourn time of a mobile device within the coverage area of a cloudlet; The difference is that, instead of the coverage area of a cloudlet we consider the AOI of a task. Here, $\mathcal{D}_t$ is the delay-deadline of task t, defined by an application; and, d_m^t is the task completion delay. Note that, the value of $\mathcal{U}_\mathcal{M}$ ranges between 0 and 1 and the exponent function in Eq. 3.2 decreases its value sharply when the task completion delay, d_m^t is close to $min(\mathscr{L}_m^t, \mathcal{D}_t)$.

Calculation of $\mathcal{U}_D$: Quality of sensing a task highly depends on the distance of a worker from the center of it's AOI for many applications including measurement of temperature, light intensity, Wi-Fi signal strenth, etc. Further away a worker is from the center of a task, less the sensing quality it can provide. Thus, we define a utility function based on Eucledian distance, $||l_m - l_t||_2$ between the current location, l_m of a worker $m \in \mathcal{M}$ and the center location, l_t of the task $t \in \mathcal{T}$, as follows,

$$\mathcal{U}_\mathcal{D} = \begin{cases} (1 - \frac{||l_m - l_t||_2}{r_t})^\delta & ||l_m - l_t||_2 \leq r_t, \\ 0 & otherwise, \end{cases} \tag{3.3}$$

where, r_t is the radius of task t's AOI and δ is the distance factor. Now, combining $\mathcal{U}_M$ and $\mathcal{U}_D$, we define a joint utility, $\mathcal{U}_{MD}$ due to current location as follows,

$$\mathcal{U}_{MD} = \mathcal{U}_M \times \mathcal{U}_D. \tag{3.4}$$

Note that, the Eq. 3.4 gives higher utility to a worker that stays longer within the AOI of a task and is located closer to center of AOI.

Calculation of $\mathcal{U}_Q$: Alongside the worker utility due to current position and movement, its historical sensing quality is also an important parameter to consider. To estimate sensing accuracy of a worker from its historical sensing quality, we use *quality indicator* proposed in ABSee [37]. It helps us to discourage selfish nodes that demand high payment for services. In PQ-Trade, the QoS management module determines the sensing quality as the standard deviation of data qualities sensed by the workers. Assume that, ${\psi^t_m}^{(k)}$ and $\psi^{t(k)}$ denote the value of task $t \in \mathcal{T}$ obtained from worker $m \in \mathcal{M}$ and estimated value of task t performed by the winners, respectively, in the k^{th} task allocation round. Now, we calculate the *quality indicator*, $q_m^{(k)}$ of worker $m \in \mathcal{M}$ for performing tasks $\mathcal{T}' \subseteq \mathcal{T}$ in k^{th} round as follows,

$$q_m^{(k)} = \sqrt{\frac{1}{|\mathcal{T}'|} \times \sum_{\forall t \in \mathcal{T}'} ({\psi^t_m}^{(k)} - \psi^{t(k)})^2}, \tag{3.5}$$

where, $\mathcal{T}' = \{t : \text{m has a winning bid for } t\}$. $\psi^{t(k)}$ can be estimated accurately using truth discovery methods proposed in [53]. Now, q_m is updated using EWMA as follows,

$$q_m = \gamma \times q_m^{(k)} + (1 - \gamma) \times q_m^{(k-1)}, \tag{3.6}$$

where, γ $(0 \leq \gamma \leq 1)$ is the weight factor for the most recent value of q_m. Note that, $q_m > 0$ and a smaller value indicates higher quality of sensed data. Initialy, platform sets

$q_m = q_0$ for each worker $m \in \mathcal{M}$ and updates the value of q_m using Eq. 3.6 after each task allocation round based on the quality of data sensed by the workers. Note that, a more accurate value of q_m can be estimated after running a number of task allocation rounds.

Now, PQ-Trade platform calculates the third utility, $\mathcal{U}_Q$ based on q_m as follows,

$$\mathcal{U}_Q = -2 \times (\frac{1}{1 + e^{-\vartheta q_m}} - 1). \tag{3.7}$$

Here, $\mathcal{U}_Q \in [1, 0]$ and ϑ is a scalling factor. By adjusting the value of ϑ, the shape of $\mathcal{U}_Q$ can be adaptive according to the application requirements. For a newly selected worker (i.e., $q_m = q_0$), we recommend, $q_0 = \frac{1}{\vartheta} \times ln(3)$ to set a fair utility (i.e., $\mathcal{U}_Q = 0.5$).

Calculation of $\mathcal{U}_m^t$: Now, we define a combined utility function $\mathcal{U}_m^t$, for selecting the bid, Γ_m^t of a worker m for task t as follows,

$$\mathcal{U}_m^t = \alpha \times \mathcal{U}_{MD} + (1 - \alpha) \times \mathcal{U}_Q. \tag{3.8}$$

Note that, $\mathcal{U}_m^t$ is a linear combination of worker utilities due to location and sensing reputation, weighted by α $(0 \leq \alpha \leq 1)$. By tuning the value of α, the platform can achieve different objectives according to application preferences.

3.3.2.2 Platform Profit

A platform tries to maximize its profit while providing quality sensed data services to its customers. To calculate the profit of the platform for selecting a bid, Γ_m^t of a worker $m \in \mathcal{M}$ for a task $t \in \mathcal{T}$, we calculate monetary value, $\mathcal{V}_m^t$, as follows,

$$\mathcal{V}_m^t = V_t \times \frac{w_m^t}{\mathscr{W}_t}, \tag{3.9}$$

where, V_t is the value of task t to the platform.

Recall that the measurement processes for the parameters V_t, $\mathscr{W}_t$ and w_m^t have been discussed in sections previous sections. Now, the profit of the platform, $\mathcal{P}_m^t$ is determined as,

$$\mathcal{P}_m^t = \mathcal{V}_m^t - c_m^t, \tag{3.10}$$

where, c_m^t is the claimed cost of a worker $m \in \mathcal{M}$ for performing a task $t \in \mathcal{T}$. Now, the normalized profit, ρ_m^t of the platform can be calculated as,

$$\rho_m^t = \frac{\mathcal{P}_m^t}{V^{max}}, \tag{3.11}$$

where, $0 \leq \rho_m^t \leq 1$ and $V^{max} = \max_{t \in \mathcal{T}} V_t$.

3.3.2.3 Objective Function

Note that, profit, $\mathcal{P}_m^t$ made by the platform for allocating task $t \in \mathcal{T}$ to worker $m \in \mathcal{M}$ depends on worker's claimed cost, c_m^t and monetary value, V_t of that task. Thus selecting workers with higher utility values may end up with decreasing platform profit and vise-versa. In this work, our goal is to make a trade-off between these two. That is, in addition to maximize the platform profit while maintaining required quality of sensing data values, we carry out boundary analysis both for the data quality and the platform profit. The proposed PQ-Trade system is formulated as a Multi-Objective Non Linear Programming (MONLP) problem as follows:

$$\mathscr{B}' = \underset{b \in \mathcal{P}(\mathscr{B})}{\operatorname{argmax}} \sum_{\forall \Gamma_m^t \in b} \{\omega \times \mathcal{U}_m^t + (1-\omega) \times \rho_m^t\} \tag{3.12}$$

s.t.

$$\sum_{\forall \Gamma_m^t \in b} w_m^t \leq \mathscr{W}_t, \quad \forall t \in \mathcal{T} \tag{3.13}$$

$$|\mathscr{B}' \bigcap \mathscr{B}_m| \leq n_m^{max}, \quad \forall \Gamma_m^t \in b \tag{3.14}$$

$$\mathcal{U}_{MD} > 0, \quad \forall \Gamma_m^t \in b, \forall t \in \mathcal{T} \tag{3.15}$$

$$\mathcal{U}_m^t \geq \mathcal{U}_{th}^t, \quad \forall \Gamma_m^t \in b, \forall t \in \mathcal{T} \tag{3.16}$$

$$\rho_m^t \geq \rho_{th}^t, \quad \forall \Gamma_m^t \in b, \forall t \in \mathcal{T} \tag{3.17}$$

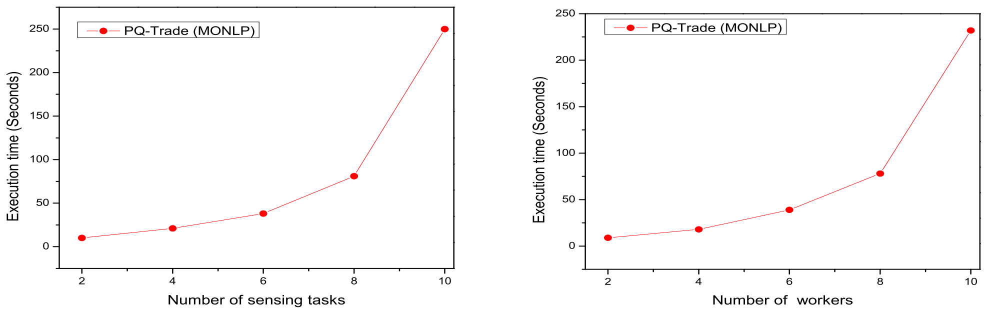
Here, the weight parameter ω $(0 \leq \omega \leq 1)$ controls the objective of the optimization function, i.e., setting $\omega = 1$ makes the problem as a utility (i.e., data quality) maximization problem and $\omega = 0$ converts the problem into a profit maximization problem, while the other values make the desired trade-off we want to achieve from the system.

The constraint (13) is the workload assignment constraint and it indicates that for each task $t \in \mathcal{T}$, workload offered by all selected workers, $\forall m \in b$ must be less than or equal to the total workload of that task. The constraint (14) is the maximum winning bids constraint and it refers that, the total number of winning bids of a worker $m \in b$ must be less than or equal to n_m^{max}. The constraint (15) ensures that the workers outside the task's AOI or having remaining lifetime less then the completion delay of the task can't be selected. The constraint (16) refers that the total utility, $\mathcal{U}_m^t$ achieved from worker $m \in b$ for task $t \in \mathcal{T}$ must be greater than or equal to some threshold, $\mathcal{U}_{th}^t$. The constraint (17) indicates that each worker $m \in b$ selected for task $t \in \mathcal{T}$ must have some profit greater than or equal to ρ_{th}^t. Note that, constraints (16) and (17) ensures the marginal level of the utility and the profit, respectively.

Note that the worker selection problem in PQ-Trade, formulated in Eq. (3.12), is a multi-objective non-linear programming (MONLP) problem. In the literature this is known as classical *Maximum Weight Subset Selection Problem* that aims to select a subset of elements from the universal set, $S \subset U$, satisfying certain properties so as to maximize some objective function $f(S)$. Similarly, in PQ-Trade, selecting a subset b from the universe of discourse $\mathcal{P}(\mathscr{B})$, i.e., $b \in \mathcal{P}(\mathscr{B})$ that maximizes the objective function in Eq. (3.12) satisfying constraints (11) to (17), is a *Maximum Weight Subset Selection Problem*, which is a well known NP-hard problem [66]. So PQ-Trade worker selection problem is NP-Hard and no solution can be found in polynomial time.

3.3.3 Greedy Solutions

The optimal worker selection problem, presented in the previous section is proved to be $NP - hard$. In a practical crowdsourcing application, thousands of users may co-exist and bid for number of tasks at a time. Thus, even if the higher computing facilities are available in cloud platform, no solutions can be found in polynomial time. To justify the above statements we simulate the objective function of PQ-Trade in NEOS optimization server (2x Intel Xeon E5-2698 @ 2.3GHz CPU and 92GB RAM) both for varying number of sensing tasks and worker devices, and the result is shown in Fig. 3.4. The result reveals that, on an average, several seconds are required to run PQ-Trade worker selection algorithm for around $5 \sim 6$ number of tasks and $5 \sim 7$ number of workers. As the number of tasks or workers increases it requires exponentially high execution time. The reason is that, PQ-Trade system, generates subsets of the bids which increases as a power of total number of bids. In this section, we develop two greedy solutions with different objectives for worker selection problem which can select workers in polynomial time.



(a) Execution time vs. number of sensing tasks (b) Execution time vs. number of workers

Figure 3.4: Impacts of number of sensing tasks and workers on execution time

At first, we develop a *First-Fit Utility Maximization (FFU)* algorithm that aims to select each bid $q_m^t \in \mathscr{B}$ of worker $m \in \mathcal{M}$ for task $t \in \mathcal{T}$ so as to maximize its utility, $\mathcal{U}_m^t$ while keeping profit, $\mathcal{P}_m^t$ at a fixed level. The later one, *Fast-Fit Profit Maximization*

(FFP) algorithm greedily maximizes platform's profit while selecting workers offering minimal utility values.

3.3.3.1 First-Fit Utility Maximization

In this strategy, for each task $t \in \mathcal{T}$, the platform fixes a minimum profit target, $\mathcal{P}^t_{min}$ and aims at maximizing the utility of each selected worker $m \in \mathcal{M}$. That is, the system emphasizes here on increasing the sensing data quality.

Algorithm 1 summarizes the steps of First-Fit Utility Maximization. For each candidate bid $q \in \mathscr{B}_t$, $\mathcal{U}^t_m$, $\mathcal{V}^t_m$ and $\mathcal{P}^t_m$ are calculated (line no. 9) and $\mathscr{B}$ is sorted in the decreasing order of $\mathcal{U}^t_m$. After that, each candidate bid, $q \in \mathscr{B}$ is choosen greedily and added to the winning bid set, $\mathscr{B}'$ if $\mathcal{P}^t_m$ is greater than or equal to $\mathcal{P}^t_{min}$, t has unassigned workload and the worker won permitted number of bids (line no. 15). This process repeats until $\mathscr{B}$ is empty.

The complexity of Algorithm 1 is quite straightforward to follow. The statement 3 is enclosed in a loop that iterates $|\mathcal{M}|$ times. The statement 6 is also enclosed in another loop that iterates $|\mathcal{T}|$ times. The statement 9 is enclosed in a loop that iterates $|\mathscr{B}|$ times and has the complexity of $O(|\mathcal{T}| \times |\mathcal{M}|)$ in the worst case. In statement 11, we use quick sort having worst case complexity $O(|\mathcal{T}|^2 \times |\mathcal{M}|^2)$. The statements 13 $\sim$ 19 are enclosed in a loop that iterates $|\mathscr{B}|$ times having worst case complexity of $O(|\mathcal{T}| \times |\mathcal{M}|)$. The rest of the statements have constant unit time complexities. Therefore, the overall computational complexity of the algorithm is $O(|\mathcal{M}| + |\mathcal{T}| + |\mathcal{M}| \times |\mathcal{T}| + |\mathcal{M}|^2 \times |\mathcal{T}|^2 + |\mathcal{M}| \times |\mathcal{T}|) \approx O(|\mathcal{M}| + |\mathcal{T}| + |\mathcal{M}|^2 \times |\mathcal{T}|^2) \approx O(|\mathcal{M}|^2 \times |\mathcal{T}|^2)$.

3.3.3.2 First-Fit Profit Maximization

In this strategy, the cloud platform aims at maximizing its profit while maintaining a minimum quality of sensed data by the workers.

Algorithm 2 summarizes the steps of First-Fit Profit Maximization Algorithm. All

Algorithm 1 First-Fit Utility Maximization Algorithm

INPUT: Set of bids of all workers, $\mathscr{B} \leftarrow \bigcup_{\forall m \in \mathcal{M}} \mathscr{B}_m$

OUTPUT: Set of winning bids, $\mathscr{B}'$

$\mathscr{B}' \leftarrow \phi$
for all $m \in \mathcal{M}$ **do**
    $n_m \leftarrow 0$
end for
for all $t \in \mathcal{T}$ **do**
    $w_t \leftarrow 0$, $c_t \leftarrow 0$
end for
for all $q \in \mathscr{B}$ **do**
    Calculate $\mathcal{U}_m^t$, $\mathcal{V}_m^t$ and $\mathcal{P}_m^t$ using Eq. (3.8), (3.9) and (3.10), respectively
end for
Sort $\mathscr{B}$ in descending order of $\mathcal{U}_m^t$
while $(\mathscr{B} \neq \phi)$ **do**
    $q \leftarrow$ First element of $\mathscr{B}$
    if $(\mathcal{P}_m^t \geq \mathcal{P}_{min}^t$ && $(\mathscr{W}_t - w_t) \geq w_m^t$ && $n_m < n_m^{max}$ && $(V_t - c_t) \geq c_m^t)$ **then**
        $\mathscr{B}' \leftarrow \mathscr{B}' \bigcup q$
        $w_t \leftarrow w_t + w_m^t$, $c_t \leftarrow c_t + c_m^t$
        $n_m \leftarrow n_m + 1$
    end if
    $\mathscr{B} \leftarrow \mathscr{B} \setminus q$
end while
return $\mathscr{B}'$

Algorithm 2 First-Fit Profit Maximization Algorithm

INPUT: Set of bids of all worker, $\mathscr{B} \leftarrow \bigcup_{\forall m \in \mathcal{M}} \mathscr{B}_m$

OUTPUT: Set of winning bids, $\mathscr{B}'$

$\mathscr{B}' \leftarrow \phi$
for all $m \in \mathcal{M}$ **do**
    $n_m \leftarrow 0$
end for
for all $t \in \mathcal{T}$ **do**
    $w_t \leftarrow 0$, $c_t \leftarrow 0$
end for
for all $q \in \mathscr{B}$ **do**
    Calculate $\mathcal{U}_m^t$, $\mathcal{V}_m^t$ and $\mathcal{P}_m^t$ using Eq. (3.8), (3.9) and (3.10), respectively
end for
Sort $\mathscr{B}$ in descending order of $\mathcal{P}_m^t$
while $(\mathscr{B} \neq \phi)$ **do**
    $q \leftarrow$ First element of $\mathscr{B}$
    if $(\mathcal{U}_m^t \geq \mathcal{U}_{min}^t$ && $(\mathscr{W}_t - w_t) \geq w_m^t$ && $n_m < n_m^{max}$ && $(V_t - c_t) \geq c_m^t)$ **then**
        $\mathscr{B}' \leftarrow \mathscr{B}' \bigcup q$
        $w_t \leftarrow w_t + w_m^t$, $c_t \leftarrow c_t + c_m^t$
        $n_m \leftarrow n_m + 1$
    end if
    $\mathscr{B} \leftarrow \mathscr{B} \setminus q$
end while
return $\mathscr{B}'$

the statements are similar to algorithm 1 excepts, in line no. 11, $\mathscr{B}$ is sorted in the decreasing order of $\mathcal{P}_m^t$ and, instead of checking minimum utility condition, it checks for minimum profit condition (*i.e.*, $\mathcal{U}_m^t \geq \mathcal{U}_{min}$) in line no. 14. The computational complexity of Algorithm 2 is same as that of Algorithm 1.

3.3.4 Worker Payment Policy

We develop a payment policy for the workers won for performing different tasks. The payment policy not only gives a share of the profit to the workers but also it penalizes a worker that fails to provide required quality of data. Such a policy stimulates workers to join the crowdsourcing system with good contribution. We define the payment P_m^t of a worker $m \in \mathcal{M}$ for performing a task $t \in \mathcal{T}$ as follows,

$$P_m^t = \begin{cases} c_m^t + \varphi & q_m^t \leq q_{max}^t \;\&\&\; \mathcal{P}_m^t \geq \mathcal{P}_{min}^t, \\ \varepsilon \times c_m^t & q_m^t > q_{max}^t, \\ c_m^t & Otherwise, \end{cases} \tag{3.18}$$

where, q_{max}^t is the maximum allowable standard deviation of sensing quality for task t. In the case $q_m^t \leq q_{max}^t$ and $\mathcal{P}_m^t \geq \mathcal{P}_{max}^t$ (i.e., platform makes more profit than expected from the worker with high quality data) the worker is considered as a valuable one and φ amount of extra profit is added to the claimed cost of the worker. On the other hand, $q_m^t > q_{max}^t$ indicates poor quality of the sensed data, resulting in reduced the payment of the worker by a factor of ε. Otherwise, the payment of the worker is simply equal to its claim cost. Different platforms can adapt their own strategies for calculating reward amount, φ and penalty factor, ε; we calculate φ using eq. 3.19 as follows,

$$\varphi = (\mathcal{P}_m^t - \mathcal{P}_{min}^t) \times \{1 - e^{(q_m^t - q_{max}^t) \times \theta_1}\} \tag{3.19}$$

.

To calculate penalty factor, ε we use simular equation of [36] as follows,

$$\varepsilon = e^{(q_{min}^t - q_m^t) \times \theta_2}. \tag{3.20}$$

Note that, θ_1 and θ_2 in eq. 3.19 and eq. 3.20, respectively are two scaling factors whose values can be adjusted to control the effect of sensing quality deviation. Now, we develop a payment vector $\mathscr{P}$ for all the selected workers, following the above payment policy, using Algorithm 3.

Algorithm 3 Determination of Payment Vector

INPUT: Set of winning bids, $\mathscr{B}'$

OUTPUT: Payment vector, $\mathscr{P}$

1: $\mathcal{M}' \leftarrow \{m : q_m^t \in \mathscr{B}'\}$
2: **for all** $m \in \mathcal{M}'$ **do**
3: $\quad \mathscr{P}_m \leftarrow 0$
4: **end for**
5: **for all** $q \in \mathscr{B}'$ **do**
6: $\quad$ Calculate p_m^t using Eq. (3.18)
7: $\quad \mathscr{P}_m \leftarrow \mathscr{P}_m + p_m^t$
8: **end for**
9: **return** $\mathscr{P}$

The complexity of Algorithm 3 is quite straight forward to follow. The statement 3 is enclosed in a loop which iterates for $|\mathcal{M}'|$ times having the worst case complexity $O(|\mathcal{M}|)$. Statements 6 $\sim$ 7 are enclosed in another loop which iterates for $|\mathscr{B}'|$ times having the worst case complexity $O(|\mathcal{T}| \times |\mathcal{M}|)$. Other statements have constant time complexities. Thus, the total complexity of Algorithm 3 is $O(|\mathcal{M}| + (|\mathcal{T}| \times |\mathcal{M}|)) \approx O(|\mathcal{M}| \times (|\mathcal{T}| + 1))$.

3.4 Conclusion

In this chapter, we have presented the system model and design components of our proposed PQ-Trade system. After that, we have given the work flow of our proposed work on how the PQ-Trade platform takes optimal workload allocation decisions. In the next chapter, we show the experimental results of our PQ-Trade workload allocation policy and compare the obtained results with some of the state-of-the-art works.

Chapter 4

Performance Evaluation

In the previous chapter, we have discussed on the formulation of the PQ-Trade worker selection problem that selects the optimal workers balancing their utilities and platform profit. In this chapter, we present the detail performance evaluation results of our proposed PQ-Trade system and analyze its effectiveness by comparing it with state-of-the-art workers.

4.1 Introduction

The actual quality of a research work can be judged by its performance in realistic environment. In this chapter we describe how we evaluate the performance of our proposed PQ-Trade system. We perform synthetic analyzing of our proposed system on different simulation scenario. We analyze the performance of our proposed system based on different metrics such as, platform profit, average utility, request service satisfaction, etc. To realize the effectiveness of PQ-Trade system, we implement three versions of our greedy solution - *FFP*, *FFU* and *PQ-Trade* using a commercial software MATLAB [67] and compare their performances with two state-of-the-art mechanisms MSC [46] and SACRM [36]. The PQ-Trade greedy solution sorts bids in the decreasing order of objective function decribed in Eq. 3.12. In the subsequent sections, we present the simulation environment, performance metrics and simulation results.

4.2 Simulation Environment

We assume that sensing tasks and worker devices are randomly distributed in a $1000 \times 1000\ m^2$ area. The arrivals of worker devices and sensing task requests are generated using Poisson distribution. Each sensing task has a circular coverage area, divided into several number of workloads and must be served within a delay deadline. The radius of coverage area, total number of workloads and delay deadline of each task are chosen randomly from 20-150m, 1-7, and 5-15s, respectively. The value (i.e., the monetary demand of the sensing service) of each task is distributed over 5 -15 with random uniform distribution. Each worker device can move at different velocities in random direction and bid for a subset of tasks. The number of workloads of each task offered by a worker is taken from 1-7 randomly and worker claimed cost varies uniformly over 1 - 20. For evaluating and updating quality of sensed data, we adopt similar philosophy of [36] where worker having combined utility, $\mathcal{U}_m^t < 0.3$ submits poor sensing results with a high probability of (1 - $\mathcal{U}_m^t$) and worker with $\mathcal{U}_m^t \geq 0.3$ provides poor sensing results with probability $0.4 \times$ (1 - $\mathcal{U}_m^t$). All the simulation parameters are summarized in table 4.1. Each simulation was run for 1000 seconds and graph data points are plotted for the average of the results from 50 simulation runs with different random seed values. We run all simulation experiments on a machine having 2.8 GHz Intel CPU and 4GB memory.

4.3 Performance Metrics

We analyze the performance of the studied systems *FFU*, *FFP*, *PQ-Trade*, MSC [68] and SACRM [36] on the following metrics.

- *Profit of the platform:* It is defined as the total amount of revenue received by the platform after completion of all allocated workloads for all the tasks by the selected worker devices.

Table 4.1: Simulation Parameters

Parameter	Value
Simulation area	$1000 \times 1000\ m^2$
Arrival rate of sensing tasks	$2 \sim 8$ tasks/sec
Arrival rate of worker devices	$2 \sim 8$ workers/sec
Workloads of task	$1 \sim 7$
Radius of task's AOI	$20 \sim 150$m
Task budget	$5 \sim 15$ units
Worker claimed cost	$1 \sim 20$ units
Task delay deadline	$5 \sim 15$s
Task completion time	$1 \sim 20$s
Worker mobility speed	$4.5 \sim 7$km/h
$\mathcal{U}_{min}$	0.3
$\mathcal{P}_{min}$	10%
α	0.6
ω	0.6
Simulation time	1000 Seconds

- *Average utility per worker:* It is measured as the ratio of total utility achieved from the selected workers to the total number of workers.
- *Request service satisfaction:* It is define as the ratio of total number of completed workloads of all the tasks to total workloads submitted by the data requesters.
- *Standard deviation of sensing quality:* It is defined as the average standard deviation of quality of sensed data received from the selected workers.
- *Average payment per worker:* It is measured as the ratio of total payments given to the selected workers to the total number of workers.
- *Execution time:* It is defined as the total time required to run the worker selection algorithm.

4.4 Simulation Results

In this section, we discuss on the results of performance evaluations for varying arrival rates of tasks, arrival rates of worker devices, worker velocity and worker claimed cost. We also study the impact of varying values of control parameter, ω, minimum profit threshold, $\mathcal{U}_{min}$ and minimum utility threshold, $\mathcal{P}_{min}$.

4.4.1 Impact of Varying Task Arrival Rates

In general increasing the task arrival rate increases the profit of the platform, decreases the utility of the selected workers and request service satisfaction. In this experiment, we varied the task arrival rates between 2∼8 tasks/second. The arrival rate of the workers is kept constant at 5 workers/second.

The graph of Fig. 4.1 states that the total profit of the platform increases with the increasing task arrival rates in all the studied systems. This is because a higher profit can be obtained by the cloud platform when more tasks are being crowdsourced to the

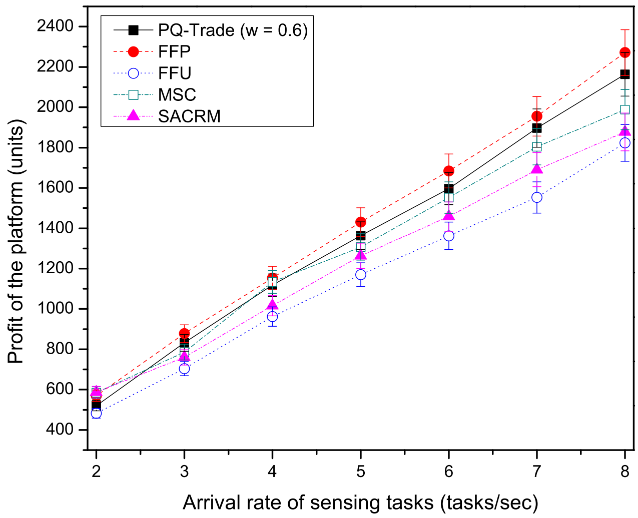


Figure 4.1: Profit of the platform for varying task arrival rates

worker devices. However, our proposed FFP algorithm provides the highest profit as it is designed to maximize the profit of the platform. The MSC [68] system makes more profit than SACRM [36] and FFU system due to the selection of workers minimizing social cost but can't outperform FFP as minimum social cost couldn't always ensure higher amount of profit. Though FFU aims to maximize the utilities of the selected workers, it performs better than SACRM because of maintaining the marginal profit. The SACRM system doesn't consider platform profit thus performs worst.

Average utility per worker device for varying task arrival rates is shown in Fig. 4.2. The graphs depict the fact that, the average utility increases with the increasing task arrival rates for all the studied systems. Such results are achieved by allocating more work loads of different sensing tasks to the suitable worker devices. However, it reaches a saturation point when task arrival rate crosses 7 tasks/sec. The obtained result clearly reveals the fact that, the proposed FFU system achieved the highest utility from the selected workers compared to other systems as it is designed to maximize the utility.

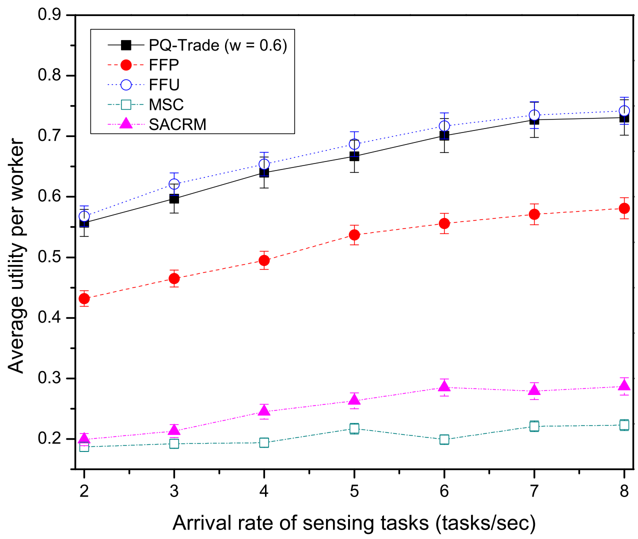


Figure 4.2: Average utility for varying task arrival rates

Inspite of maximizing the profit of the platform, the proposed FFP outperforms MSC and SACRM systems for maintaing minimal utility while selecting workers. The proposed PQ-Trade ($\omega = 0.6$) system performs in between FFU and FFP but closer to FFU. This is because it gives more priority to worker utility than the profit of the platform.

In Fig. 4.3, we observe that, the request service satisfaction decreases with the increasing task arrival rates and starts to fall rapidly after task arrival rate reaches to 6 tasks/sec. This is caused by the fact that all the workers get allocated with the maximum number of bids. All of our proposed systems give priority to the workers with longer sojourn period within the AOI of a task and thus it outperforms MSC and SACRM. It is interesting to observe that, FFP performs better than the proposed FFU as it maintains the minimal utility requirement of the application. On the other hand, while maximizing utility of the selected workers, FFU sometimes fail to ensure minimum utility requirement for each of the workers. The MCS and SACRM systems neither consider the sojourn period of the workers or maintain the minimum utility requirements.

Moreover, both of those can't handle the heterogeneous task requests simultaneously and thus sensing resources of the workers remain unused which in turns decreases the request service satisfaction.

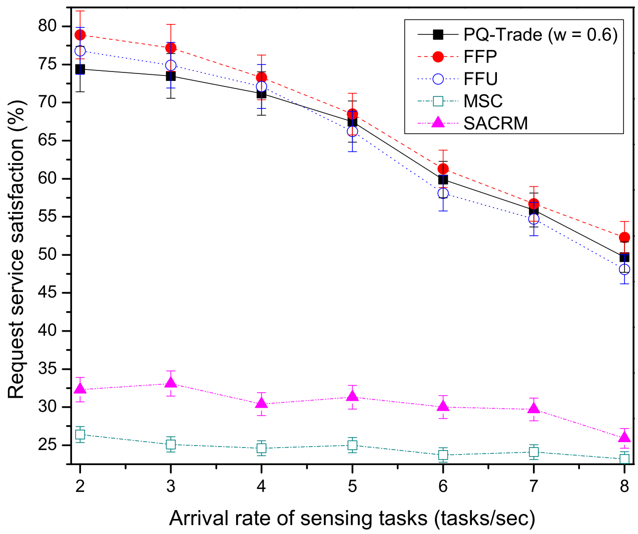


Figure 4.3: Request service satisfaction for varying task arrival rates

Standard deviation of sensing quality for varying task arrival rates for all the studied systems are shown in Fig. 4.4. The experimental outcomes reveal that, the standard deviation of sensing quality of the proposed systems are significantly lower than the MSC and SACRM systems. This is because proposed systems select workers that are probabilistically expected to provide accurate and high quality sensing results. The proposed FFU focuses on utility of the selected workers and thus it provides better sensing quality with least deviation compared to other systems. The proposed PQ-Trade (ω = 0.6) performs better than FFP as it gives more priority to utility. The MSC doesn't consider any of the quality parameters while selecting the workers, offering the worst results.

The average payment per worker device for successfully completing assigned tasks

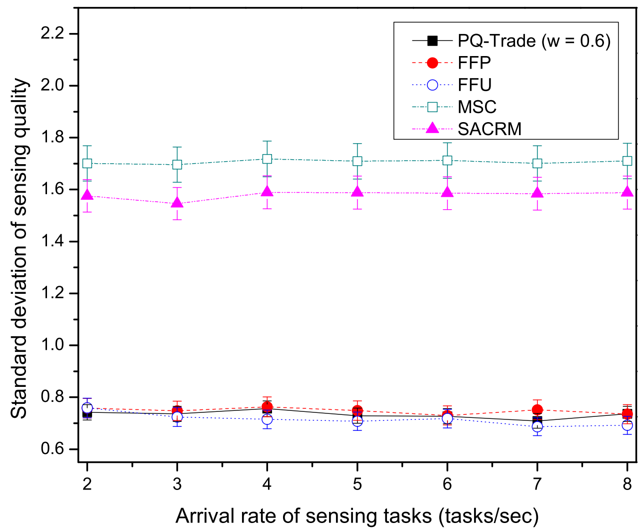


Figure 4.4: Standard deviation of sensing quality for varying task arrival rates

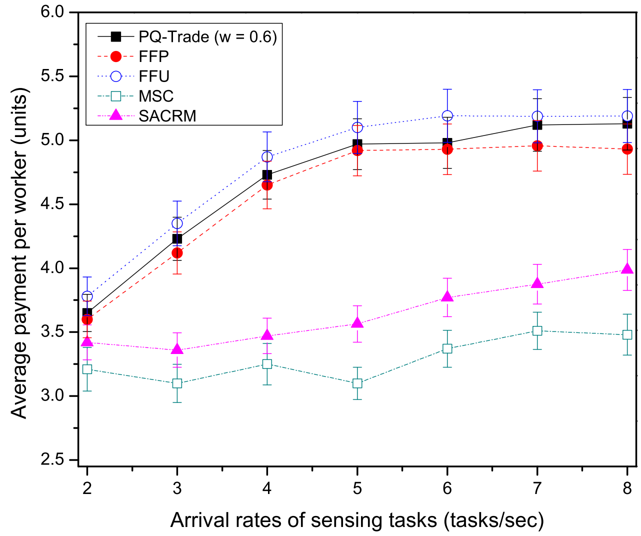


Figure 4.5: Average payment per worker

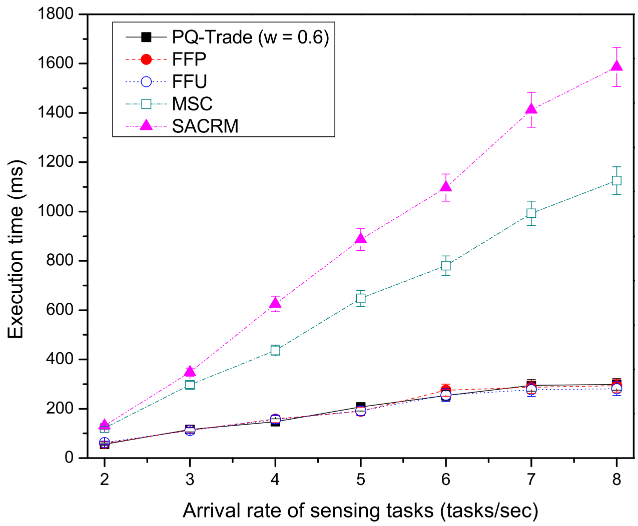


Figure 4.6: Execution time for varying task arrival rates

has been shown in Fig. 4.5. The graphs depict the fact that, the average payment of the selected workers are higher in proposed FFP, FFU and PQ-Trade ($\omega = 0.6$) compared to MSC and SACRM systems. The reason behind this result is articulated by the fact that, the proposed payment policy penalizes workers with poor sensing result at the same time gives rewards for good contribution which could even pay a worker more than its claimed cost if sufficient amount of profit is made from that worker. On the other hand, SACRM only penalizes workers thus payment of each worker can never exceed its claimed cost. In case of MCS rule of critical payment is followed as workers are paid according to their marginal contribution and paid less than or equal to their claim cost.

The graphs of Fig. 4.6 depict that the average execution time of worker selection algorithm increases in all the approaches with varying task arrival rates. The obtained result clearly reveals that the proposed FFP, FFU and PQ-Trade ($\omega = 0.6$) systems outperform MSC and SACRM systems in average execution time of worker selection algorithm. This is because proposed systems exploit a sorting based greedy algorithm which consider all the sensing task requests arrived within a scheduling interval. However, SACRM adopts a dynamic programming solution and runs worker selection algorithm whenever a new sensing request arrives to the system. Though MSC also exploits sorting based greedy algorithm it runs selection algorithm for each of the task individually. That is why both of MSC and SACRM systems experience higher average execution time.

4.4.2 Impact of Varying Worker Arrival Rates

Increasing arrival rates of worker devices, in general, facilitates the cloud platform in selecting more suitable workers. As a result platform's profit, worker utility and request service satisfaction increases with the increasing worker arrival rates. In this experiment we show the system performance for varying worker arrival rates, ranging from 2-8 workers/second. The arrival rate of sensing tasks is fixed at 5 tasks/second for measuring system efficiency.

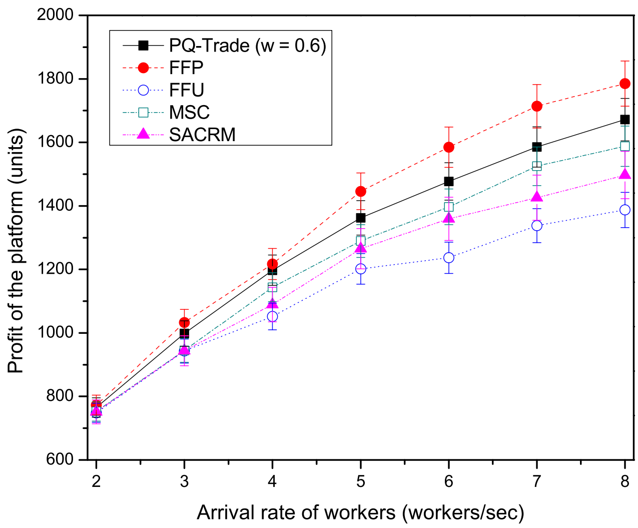


Figure 4.7: Profit of the platform for varying worker arrival rates

The graphs of Fig. 4.7 states that the total profit of the cloud platform increases with the increasing arrival rates of worker devices for all the studied system. This is caused by the fact that, from more available workers, bids for a single task increases, facilitating the platform to choose workers with additional profit. However, the rate of increase is slower as the system becomes congested with a large number of worker devices. The results also reveal that proposed FFP outperforms the other systems for the reason stated before. On the other hand, proposed FFU sacrifices profit in order to maximize the utility of selected workers.

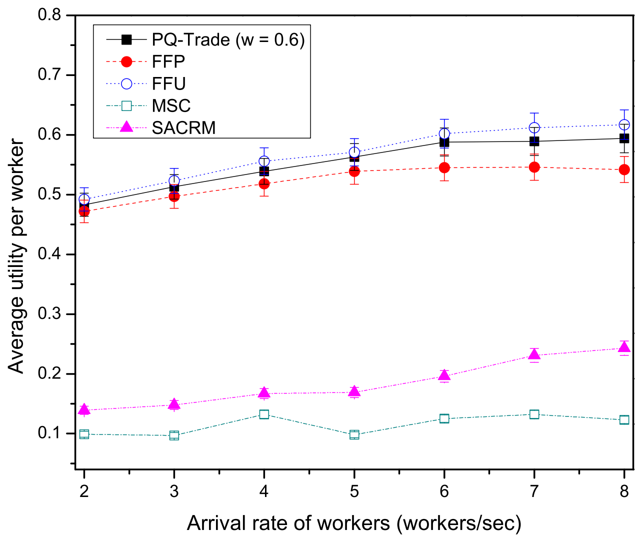


Figure 4.8: Average utility per worker for varying worker arrival rates

Average utility achieved from the selected worker devices for varying worker arrival rates is shown in Fig. 4.8 that depicts that the average utility of the proposed system increases with worker arrival rates and reaches to a saturation point. As the system becomes congested with a large number of worker devices, additional workers can contribute less in the utility gain. The FFU outperforms other systems for its design principle stated above. The SACRM system experience poor utility gain as it overlook worker's mobility and location information.

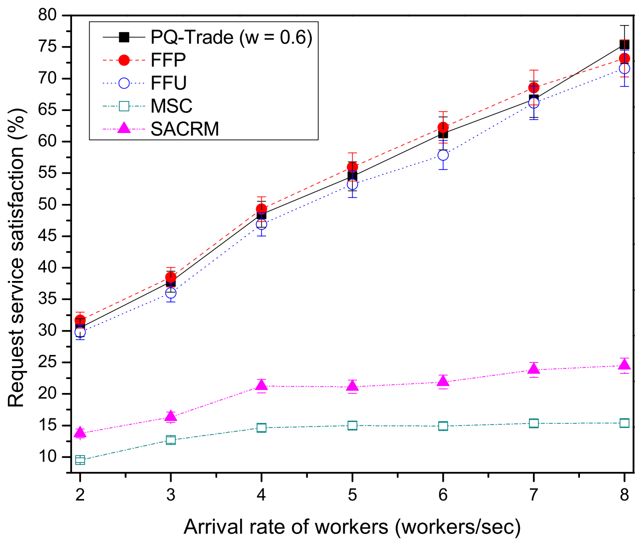


Figure 4.9: Request service satisfaction for varying worker arrival rates

Fig. 4.9 states that the request service satisfaction increases with the increasing worker arrival rates for all the studied system. However our proposed FFP, FFU and PQ-Trade ($\omega = 0.6$) outperform the MSC and SACRM system significantly. This is because PQ-Trade worker selection algorithm consider worker's sojourn time within the AOI of the task and workers with higher sojourn time get selected. As a result, task completion ratio increases significantly. It is an interesting observation that FFP performs better than FFU. This is caused by the fact that FFP maintains a minimum utility level of the application due to minimum utility constraint where FFU fails to ensure it in some cases. As expected PQ-Trade ($\omega = 0.6$) workers in between FFP and FFU. The rate of increase in utility with increasing worker arrival rates is slower for all the studied systems as the system gets saturated with sufficient workers.

Standard deviation of sensing qualities for varying worker arrival rates has been shown in Fig. 4.10. It decreases in all of the proposed systems as the platform gets the opportunity to select the workers with higher utility from a large number of worker devices.

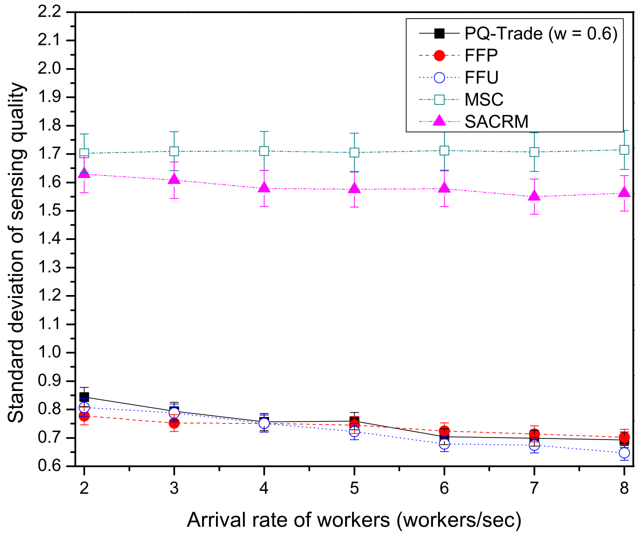


Figure 4.10: Standard deviation of sensing quality for varying worker arrival rates

However, the proposed FFU system shows better performance as it gives priority to the worker with higher utility. Moreover, the proposed FFP and PQ-Trade also gets benefit for marginal profit constraint and profit-utility trade off, respectively. All of the proposed systems outperform MSC and SACRM systems significantly for the reason stated before.

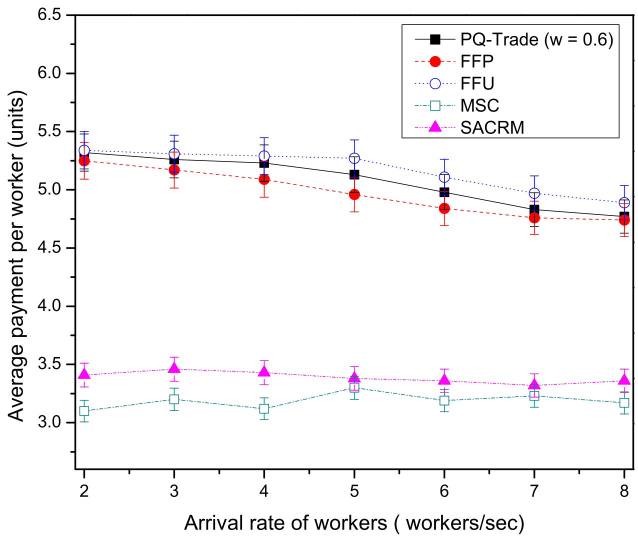


Figure 4.11: Average payment per worker for varying worker arrival rates

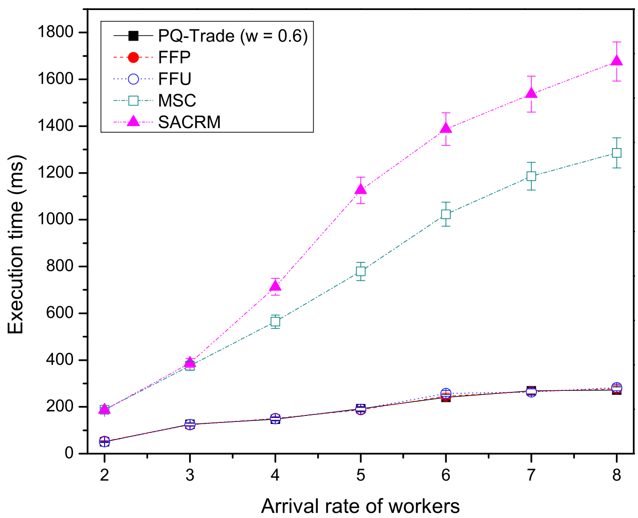


Figure 4.12: Execution time for varying worker arrival rates

Fig. 4.11 depicts the fact that average payment of the selected worker is higher in the proposed FFP, FFU, and PQ-Trade ($\omega = 0.6$) systems compared to MCS and SACRM systems. This is because of the proposed rewarding policy. From the graphs it is also clear that proposed FFP outperforms FFU for the reason stated above.

The graphs of Fig. 4.12 show the execution time of worker selection algorithm with varying worker arrival rates for all the studied systems. The the result reveals that execution time increases with the increasing worker arrival rates for all the systems. However, proposed FFP, FFU and PQ-Trade ($\omega = 0.6$) take significantly less time compared to MSC and SACRM systems. This is caused by the fact that MSC and SACRM both run worker selection algorithm each time a new task request arrives in the system. On the other hand both of the proposed algorithms consider all the tasks arrived within a task scheduling interval and exploit a sorting based greedy algorithm which significantly reduce the execution time.

4.4.3 Impact of Varying Worker Velocities

As the velocity of a worker device increases, its sojourn time within a task's AOI is reduced and thus number of workers with required service time becomes insufficient. As a result, platform's profit, utility of the selected workers and request service satisfaction are decreased with increasing velocity. In this section we show the system performance for varying worker velocities ranging from 5-65 km/h. For this experiment the arrival rates of sensing tasks and worker devices are fixed at 5 tasks/sec and 5 workers/sec respectively.

Figure 4.13 depicts the fact that with the increasing worker velocity the profit of the platform decrease for proposed systems. This is because with the increasing velocity a worker can provide sensing service for a shorter duration. As a result, the worker's mobility based utility drops to zero for the workers whose task completion delay exceeds worker's sojourn time. Thus proposed FFU gets insufficient workers to complete sensing

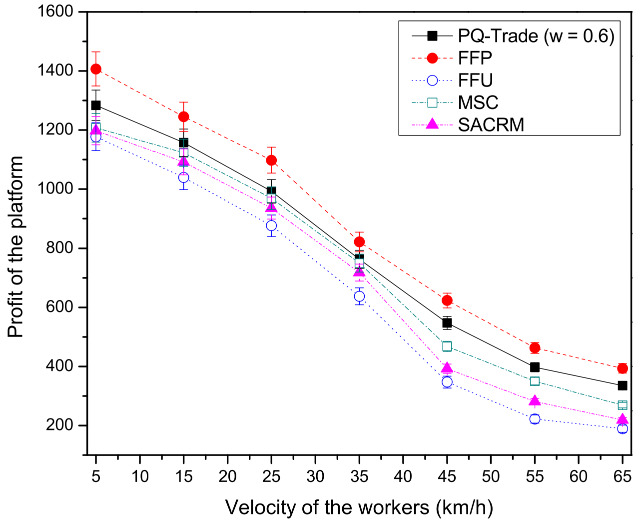


Figure 4.13: Profit of the platform for varying worker velocity

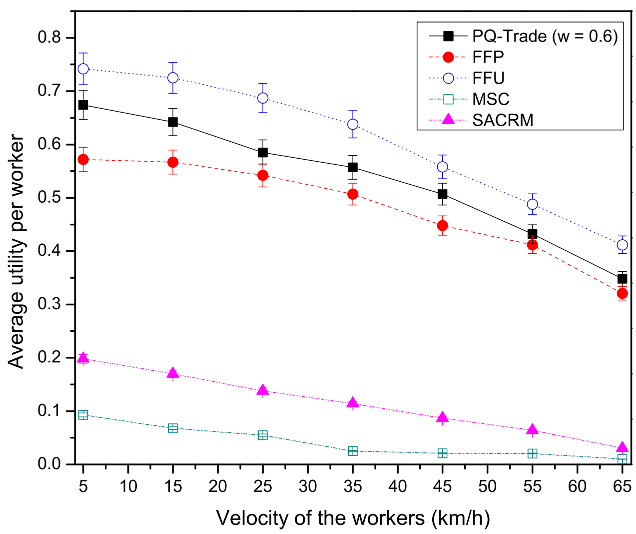


Figure 4.14: Average utility per worker for varying worker velocity

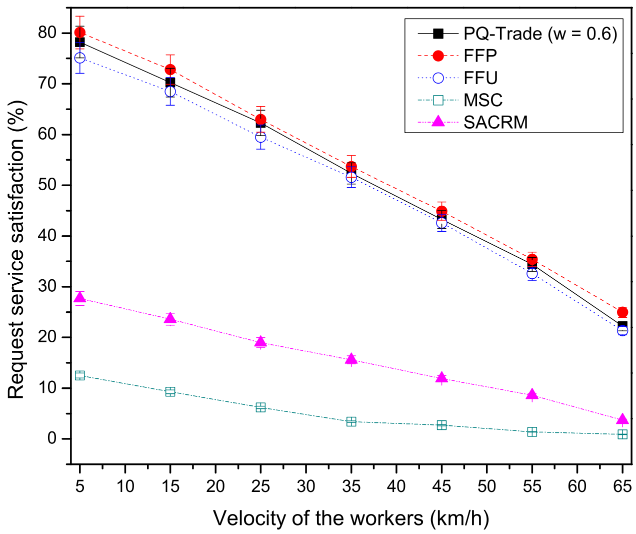


Figure 4.15: Request service satisfaction for varying worker velocity

tasks which in turns reduce the profit of the platform. For FFP the profit of the platform also decreases for the marginal utility constraint.

As shown in Fig. 4.14 average utility achieved from the selected workers decreases with the increasing worker velocity for all the studied system. This is caused by the fact that with the increasing velocity workers remaining time in a task's AOI decreases which in terns decreases workers mobility based utility. However, proposed FFP, FFU and PQ-Trade ($\omega = 0.6$) still select workers with higher utility. FFU outperforms FFP for the reason stated before.

The graphs of Fig. 4.15 depicts the fact that request service satisfaction decreases with the increasing worker velocity for all the studied systems. This is because as sojourn time of a worker decreases with the increasing worker velocity, the number of workers with required service time become insufficient. As a result, most of the workloads of the tasks remain unallocated which in turns decreases the request service satisfaction. However proposed systems outperforms MSC and SACRM systems as those selects workers

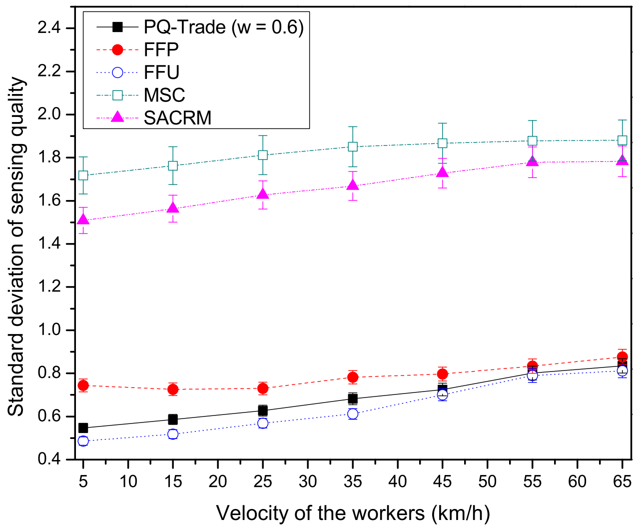


Figure 4.16: Standard deviation of sensing quality per worker

considering their sojourn time within the AOI of a task.

The standard deviation of sensing quality of the selected workers with varying worker velocity is shown in Fig. 4.16 With the increasing velocity workers with higher utility values becomes unavailable and less workers are selected. As a result, standard deviation of sensing quality increases. However, proposed systems are benefited by the utility aware worker selection polity.

4.4.4 Impact of Varying Worker Claimed Costs

We also evaluate the system performance with increasing worker claimed costs. In this experiment the arrival rate of sensing tasks and workers are fixed at 5 tasks/sec and 5 workers/sec, respectively.

Fig. 4.17 states that the profit of the platform decreases with the increasing worker claimed cost. The reason is quite straight forward.

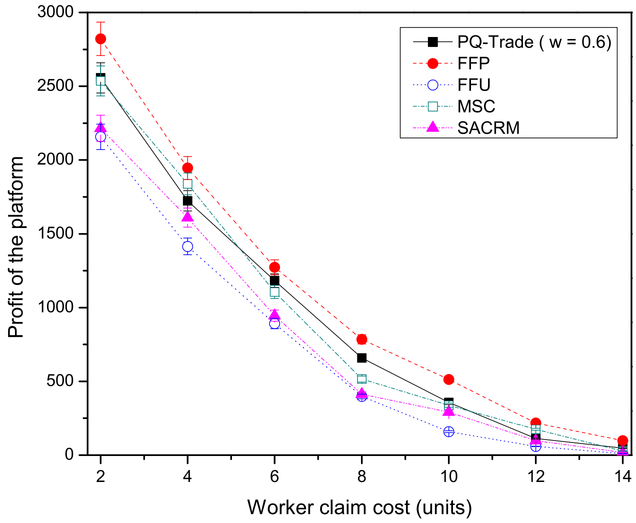


Figure 4.17: Profit of the platform for varying worker claim cost

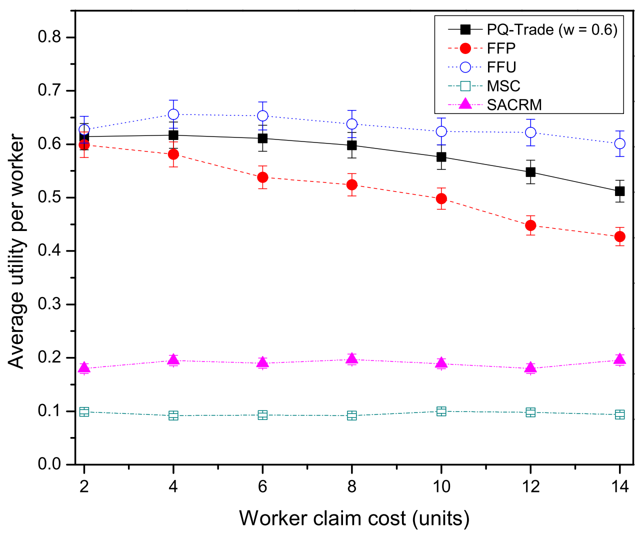


Figure 4.18: Average utility per worker for varying worker claim cost

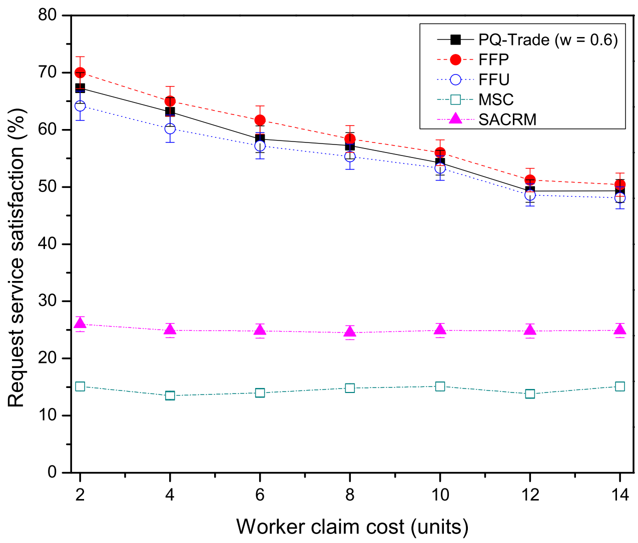


Figure 4.19: Request service satisfaction for varying worker claim cost

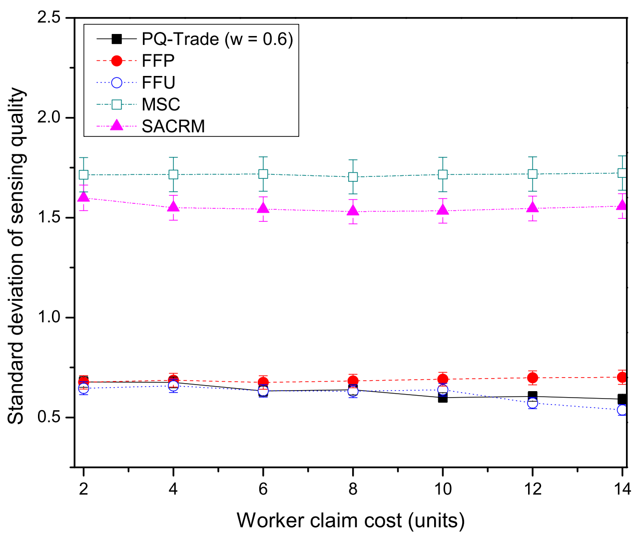


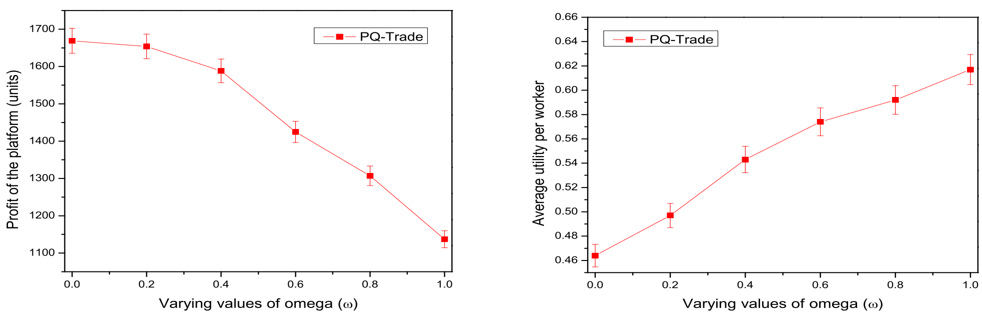
Figure 4.20: Standard deviation of sensing quality

The graphs in Fig. 4.18 depicts the fact that the average utility of the selected workers decreases as the claimed cost of the workers increases. This is because all of the proposed systems select less workers as their claimed cost exceeds the value of the tasks.

Fig. 4.19 states that request service satisfaction also decreases with the increasing claimed cost. This is caused by the fact stated above. With the increasing claimed cost less number of workers with higher utility get selected. Thus increases the standard deviation of the sensing quality which is shown in Fig. 4.20.

4.4.5 Impact of Varying Values of Control Parameter, ω

We have varied the value of ω and assessed the performances on platform profit and worker utility achieved by PQ-Trade system. The value of ω controls the level of importance the platform gives on profit and utility. For this experiment, arrival rates of sensing tasks and workers are kept constant at 5 tasks/sec and 5 workers/sec, respectively. From Fig. 4.21(a) and Fig. 4.21(b) it is clear that, platform profit and average utility of the workers is inversely related. PQ-Trade platform maximizes its profit when $\omega = 0$. Similarly the average utility of the selected workers is maximized when $\omega = 1$. However, other values of ω facilitate the system to make a trade off between profit and utility. Thus the value of ω is set by the platform following the requirement of the application.

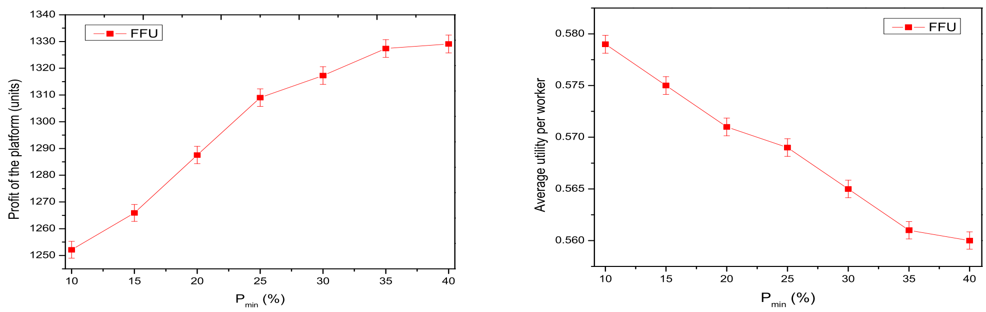


(a) Profit of the platform (b) Average utility per worker

Figure 4.21: Impact of varying values of ω

4.4.6 Impact of Marginal Profit Threshold, $\mathcal{P}_{min}$

We have observed the impact of varying values of $\mathcal{P}_{min}$ in FFU system. From the graphs of Fig. 4.22(a) and Fig. 4.22(b) it is clear that, with increasing values of $\mathcal{P}_{min}$, instead of maximizing utility of the selected workers FFU system maximizes profit and minimizes utility which is opposite to its design principle. Thus, an appropriate value should be chosen for $\mathcal{P}_{min}$ by the platform to maintain its minimum profit requirements while preserving the objectives of FFU systems.

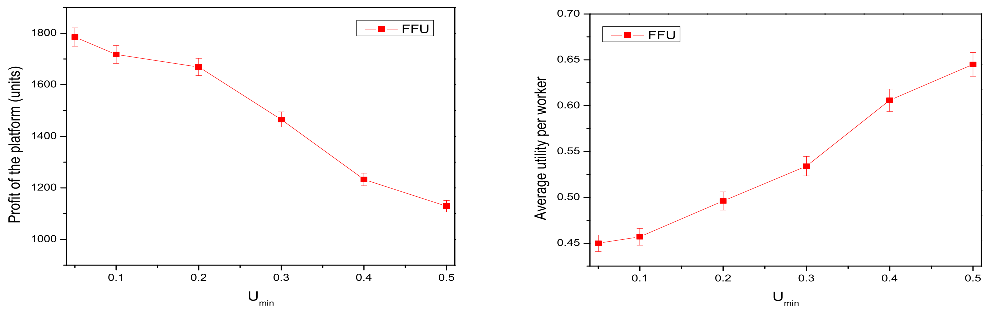


(a) Profit of the platform (b) Average utility per worker

Figure 4.22: Impact of varying values of $\mathcal{P}_{min}$

4.4.7 Impact of Marginal Utility Threshold, $\mathcal{U}_{min}$

We have also studied the impact of varying values of $\mathcal{U}_{min}$ in FFP system. The graphs of Fig. 4.23(a) and Fig. 4.23(b) depicts the fact that, with increasing values of $\mathcal{U}_{min}$, FFP maximizes average utility achieved from the selected workers and minimizes profit which is against its objectives. Thus, an appropriate value of $\mathcal{U}_{min}$ should be set by the platform such a way to maintain the minimum utility requirements of the MCS application while prioritizing the design goal of FFP.



(a) Profit of the platform (b) Average utility per worker

Figure 4.23: Impact of varying values of $\mathcal{U}_{min}$

4.5 Summary

The previous analysis shows that, our proposed PQ-Trade model performs better than MSC and SACRM in terms of platform profit, sensing quality, request service satisfaction and standard deviation of sensing quality. We have also compared the system performance in terms of average payment of the workers and execution time of worker selection algorithm. In every case we have found that,PQ-Trade system outperforms the MSC and SACRM system. Thus we can conclude that PQ-Trade system provide better solution of worker selection problem while balancing worker utility and platform profit and open a new window for further developments.

Chapter 5

Conclusion

In this chapter, we summarize the research results presented in this thesis and state few directions for future works.

5.1 Summary of Research

The key observation behind modeling the proposed PQ-Trade system was that, providing quality sensing services requires selection of high utility workers. At the same time, MCS model should be profit-aware for the platform to sustain in the long run. Thus, distribution of sensing responsibilities among the workers maintaining desired data quality and make a reasonable profit is a challenging one for the MCS platform. Although a number of works have been done in the literature, none of them consider the provider profit and the utility of the worker devices jointly. In this thesis, we addressed the problem of trade-off between the platform profit and worker utility. The key philosophy of our work was that while maximizing the profit or utility, the other one should be kept in a marginal level. Moreover, the payment policy should be adaptive according the to sensing data assessment report. To achieve these objectives, we defined the system components of the crowd platform and demonstrated their functionalities. To make the trade-off stated above, we formulated a multi-objective non-liner programming problem model with necessary constraints. The MONLP problem has been proved to be NP-hard and practical evaluation done in NEOS server suggests that, feasible solution can't be found in real time. As a result, we developed greedy first-fit solutions to select an optimal set of workers from the available workers in the MCS systems. Finally, we proposed our payment policy for the selected workers.

We have evaluated our proposed model in a distributed computation environment using MATLAB and compared the performances of our proposed system with state-of-the-art works MCS and SACRM. The most influential component of our proposed model was user utility model (i.e., worker mobility prediction module, quality assessment module, etc.). As depicted in simulation results, our proposed system outperforms the existing workers in terms of platform profit, worker utility and request service satisfaction. The proposed FFP system outperforms one of literature works in terms of profit as higher as 23.82 %, whereas, proposed FFU system achieved average utility gain which is approximately 2.29 times greater than the existing works. Our proposed system also achieved 1.62 times more request service satisfaction than the existing policies. Proposed PQ-Trade system ($\omega = 0.6$) makes 17.21% of more profit and achieved 2.22 times more utility gain. Here, profit loss and decrease in utility gain is caused due to the trade-off between profit and utility (e.g., $\omega = 0.6$ gives 60 % weight to worker utility).

5.2 Discussion

Getting in MCS research area was not an easy task. As MCS is an emerging area, our first challenge was to find a promising sector to work in MCS. We have studied MCS related research documents on various topics and found the worker selection and workload allocation problem to be most interesting. The primary hurdle we faced was analyzing the existing system and find out their drawbacks where we can put our contribution. However, the main challenge was to develop an alternate model that solves the existing problems and also compatible with MCS system.

After we developed our proposed system (PQ-Trade), the next challenge was to implement PQ-Trade and compare its performance with MCS and SACRM. Hence we needed to implement the proposed system using any simulation tool. We give a exhaustive search in the literature to find a suitable simulation tool for simulating dynamic MCS system. We hardy find any simulation tool dedicated for MCS system simulation. We chose MATLAB as most of literature works use it to implement their mechanism. During implementation phase, one of the primary challenges was to develop a realistic simulation environment. After successful implementation of PQ-Trade, we extensively evaluated their performance and at the end, the work is now presentable.

5.3 Future Plan

Despite the fact that this thesis gives an exhaustive study on MCS from different viewpoints, there are still some open issues and several research directions that can be sought after to improve the performance of proposed PQ-Trade. Making a trade off in between profit and quality, considering the presence of multiple MCS systems before the worker devices would be an interesting future work. Another interesting problem includes the consideration of the number of workers for number of participant aware MCS system. Trade off is also required in this case as the higher participation results in decreasing platform profit. However, making reasonable balance in between platform profit and number of participants will be another scope of contribution.

Bibliography

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Comm. Mag.*, vol. 48, no. 9, pp. 140–150, Sep 2010.

[2] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir: the personal environmental impact report, as a platform for participatory sensing systems research," in *in Proc. ACM/USENIX Int. Conf. Mobile Systems, Applications, and Services (MobiSys) Krakow*, 2009.

[3] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '08. New York, NY, USA: ACM, 2008, pp. 323–336.

[4] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: Accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. ACM, 2009, pp. 85–98.

[5] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10. New York, NY, USA: ACM, 2010, pp. 105–116.

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

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

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

[9] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 173–184.

[10] S. Sarker, A. K. Nath, and A. Razzaque, "Tradeoffs between sensing quality and energy efficiency for context monitoring applications," in *2016 International Conference on Networking Systems and Security (NSysS)*, Jan 2016, pp. 1–7.

[11] C. Laoudias, A. Konstantinidis, G. Chatzimilioudis, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, vol. 16, pp. 36–44, 2012.

[12] "Mobile technology fact sheet," Available online: http://www.pewinternet.org/fact-sheets/mobiletechnology-fact-sheet, accessed on 6 November 2016.

[13] Y. Wang, X. Jia, Q. Jin, and J. Ma, "Mobile crowdsourcing: Architecture, applications, and challenges," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing*

and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Aug 2015, pp. 1127–1132.

[14] J. Howe, "The rise of crowdsourcing," *Wired Magazine*, vol. 14, no. 6, 06 2006. [Online]. Available: http://www.wired.com/wired/archive/14.06/crowds.html

[15] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: challenges and opportunities," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 75–81, August 2015.

[16] M. Stevens, "Crowdsourcing of pollution data using smartphones."

[17] K. Frkas, . Z. Nagy, T. Toms, and R. Szab, "Participatory sensing based real-time public transport information service," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, March 2014, pp. 141–144.

[18] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The pothole patrol: Using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 29–39. [Online]. Available: http://doi.acm.org/10.1145/1378600.1378605

[19] D. Kwak, D. Kim, R. Liu, L. Iftode, and B. Nath, "Tweeting traffic image reports on the road," in *6th International Conference on Mobile Computing, Applications and Services*, Nov 2014, pp. 40–48.

[20] Y. C. Fan, C. T. Iam, G. H. Syu, and W. H. Lee, "Teleeye: Enabling real-time geospatial query answering with mobile crowd," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, May 2013, pp. 323–324.

[21] A. Grazioli, M. Picone, F. Zanichelli, and M. Amoretti, "Collaborative mobile application and advanced services for smart parking," in *2013 IEEE 14th International Conference on Mobile Data Management*, vol. 2, June 2013, pp. 39–44.

[22] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, "Signalguru: Leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '11. New York, NY, USA: ACM, 2011, pp. 127–140. [Online]. Available: http://doi.acm.org/10.1145/1999995.2000008

[23] J. N. C. Victorino and M. R. J. E. Estuar, "Profiling flood risk through crowdsourced flood level reports," in *2014 International Conference on IT Convergence and Security (ICITCS)*, Oct 2014, pp. 1–4.

[24] Y. Yang, M. Sherman, and J. Lindqvist, "Disaster mitigation by crowdsourcing hazard documentation," in *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, Oct 2014, pp. 93–98.

[25] D. C. M, S. Sankaranarayanan, and C. Sharma, "Project jagriti: Crowdsourced child abuse reporting," in *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, Oct 2014, pp. 609–613.

[26] J. Reilly, S. Dashti, M. Ervasti, J. D. Bray, S. D. Glaser, and A. M. Bayen, "Mobile phones as seismologic sensors: Automating data extraction for the ishake system," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 242–251, April 2013.

[27] L. I. Besaleva and A. C. Weaver, "Crowdhelp: M-health application for emergency response improvement through crowdsourced and sensor-detected information," in *2014 Wireless Telecommunications Symposium*, April 2014, pp. 1–5.

[28] D. M. Punjabi, L. P. Tung, and B. S. P. Lin, "Crowdsmile: A crowdsourcing-based social and mobile integrated system for learning by exploration," in *2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing*, Dec 2013, pp. 521–526.

[29] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 42–47. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002736.2002747

[30] T. Eaglin, K. Subramanian, and J. Payton, "3d modeling by the masses: A mobile app for modeling buildings," in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, March 2013, pp. 315–317.

[31] S. Chen, M. Li, and K. Ren, "The power of indoor crowd: Indoor 3d maps from the crowd," in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2014, pp. 217–218.

[32] K. Farkas, A. Zsolt Nagy, T. Tomas, and R. Szabo, "Participatory sensing based real-time public transport information service," pp. 141–144, 03 2014.

[33] M. S. S. Tan and R. O. Atienza, "Librorum: A crowdsourcing filipino-english dictionary mobile application," in *TENCON 2014 - 2014 IEEE Region 10 Conference*, Oct 2014, pp. 1–6.

[34] Y. Liu, T. Alexandrova, T. Nakajima, and V. Lehdonvirta, "Mobile image search via local crowd: A user study," in *2011 IEEE 17th International Conference on Embedded and Real-Time Computing Systems and Applications*, vol. 2, Aug 2011, pp. 109–112.

[35] C. Laoudias, G. Constantinou, M. Constantinides, S. Nicolaou, D. Zeinalipour-Yazti, and C. G. Panayiotou, "The airplace indoor positioning platform for android smartphones," in *2012 IEEE 13th International Conference on Mobile Data Management*, July 2012, pp. 312–315.

[36] J. Ren, Y. Zhang, K. Zhang, and X. S. Shen, "Sacrm: Social aware crowdsourcing with reputation management in mobile sensing," *Computer Communications*, vol. 65, pp. 55 – 65, 2015, mobile Ubiquitous Sensing from Social Network Viewpoint. [Online]. Available: //www.sciencedirect.com/science/article/pii/S014036641500050X

[37] B. Song, H. Shah-Mansouri, and V. W. S. Wong, "Quality of sensing aware budget feasible mechanism for mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3619–3631, June 2017.

[38] I. Koutsopoulos, "Optimal incentive-driven design of participatory sensing systems," in *2013 Proceedings IEEE INFOCOM*, April 2013, pp. 1402–1410.

[39] H. Shah-Mansouri and V. W. S. Wong, "Profit maximization in mobile crowdsourcing: A truthful auction mechanism," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 3216–3221.

[40] L. G. Jaimes, I. Vergara-Laurens, and M. A. Labrador, "A location-based incentive mechanism for participatory sensing systems with budget constraints," in *2012 IEEE International Conference on Pervasive Computing and Communications*, March 2012, pp. 103–108.

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

[42] T. Luo, H. P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 127–135.

[43] A. Subramanian, G. S. Kanth, S. Moharir, and R. Vaze, "Online incentive mechanism design for smartphone crowd-sourcing," in *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2015, pp. 403–410.

[44] X. Zhang, G. Xue, R. Yu, D. Yang, and J. Tang, "Truthful incentive mechanisms for crowdsourcing," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 2830–2838.

[45] L. Gao, F. Hou, and J. Huang, "Providing long-term participation incentive in participatory sensing," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 2803–2811.

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

[47] D. Zhao, X. Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 1213–1221.

[48] L. Tran-Thanh, M. Venanzi, A. Rogers, and N. R. Jennings, "Efficient budget allocation with accuracy guarantees for crowdsourcing classification tasks," in

Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems, ser. AAMAS '13. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2013, pp. 901–908. [Online]. Available: http://dl.acm.org/citation.cfm?id=2484920.2485063

[49] Q. Zhang, Y. Wen, X. Tian, X. Gan, and X. Wang, "Incentivize crowd labeling under budget constraint," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 2812–2820.

[50] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '15. New York, NY, USA: ACM, 2015, pp. 167–176. [Online]. Available: http://doi.acm.org/10.1145/2746285.2746310

[51] M. Zhang, P. Yang, C. Tian, S. Tang, X. Gao, B. Wang, and F. Xiao, "Quality-aware sensing coverage in budget-constrained mobile crowdsensing networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7698–7707, Sept 2016.

[52] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '15. New York, NY, USA: ACM, 2015, pp. 177–186.

[53] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han, "Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 8, pp. 1986–1999, Aug 2016.

[54] S. Reddy, D. Estrin, and M. Srivastava, "Recruitment framework for participatory sensing data collections," in *Proceedings of the 8th International Conference on Per-*

vasive Computing, ser. Pervasive'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 138–155.

[55] H. Amintoosi and S. S. Kanhere, "A trust-based recruitment framework for multi-hop social participatory sensing," in *2013 IEEE International Conference on Distributed Computing in Sensor Systems*, May 2013, pp. 266–273.

[56] X. Zhang, Z. Yang, C. Wu, W. Sun, Y. Liu, and K. Xing, "Robust trajectory estimation for crowdsourcing-based mobile applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1876–1885, July 2014.

[57] X. Wang, W. Cheng, P. Mohapatra, and T. F. Abdelzaher, "Artsense: Anonymous reputation and trust in participatory sensing," in *INFOCOM*, 2013.

[58] D. Christin, C. Rokopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "Incognisense: An anonymity-preserving reputation framework for participatory sensing applications," in *2012 IEEE International Conference on Pervasive Computing and Communications*, March 2012, pp. 135–143.

[59] H. Amintoosi and S. S. Kanhere, "A reputation framework for social participatory sensing systems," *Mob. Netw. Appl.*, vol. 19, no. 1, pp. 88–100, Feb. 2014. [Online]. Available: http://dx.doi.org/10.1007/s11036-013-0455-x

[60] K. L. Huang, S. S. Kanhere, and W. Hu, "Are you contributing trustworthy data?: The case for a reputation system in participatory sensing," in *Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, ser. MSWIM '10. New York, NY, USA: ACM, 2010, pp. 14–22. [Online]. Available: http://doi.acm.org/10.1145/1868521.1868526

[61] A. R. Johnsen, P. J. Binning, J. Aamand, N. Badawi, and A. E. Rosenbom, "The gompertz function can coherently describe microbial mineralization of growth-

sustaining pesticides," *Environmental Science & Technology*, vol. 47, no. 15, pp. 8508–8514, 2013, pMID: 23796023.

[62] S. Chen, M. Liu, and X. Chen, "A truthful double auction for two-sided heterogeneous mobile crowdsensing markets," *Computer Communications*, vol. 81, no. Complete, pp. 31–42, 2016.

[63] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive gps-based positioning for smartphones," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 299–314.

[64] C. Bettstetter, "Smooth is better than sharp: A random mobility model for simulation of wireless networks," in *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWIM '01. ACM, 2001, pp. 19–27.

[65] M. H. G. F. Asma Enayet, Md. Abdur Razzaque, "A mobility-aware optimal resource allocation architecture for big data task execution on mobile cloud in smart cities," *IEEE Communications Magazine*, 2017.

[66] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness.* New York, NY, USA: W. H. Freeman & Co., 1990.

[67] Natick, "Matlab, version 7.10.0 (r2010a)," The MathWorks Inc., Massachusetts, Tech. Rep., 2010.

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

Appendix A

Computation of $\mathscr{L}_t^m$

In PQ-Trade, we assume that a worker device, $m \in \mathcal{M}$ can move at different velocities in random direction. To model this mobility characteristics of a worker we adopt smooth random mobility model [64]. According to smooth random mobility model, the velocity of a worker at different time slots are correlated, i.e., the current velocity of a worker $m \in \mathcal{M}$ depends on its previous n number of velocities. Thus using smooth random mobility model, for each worker $m \in \mathcal{M}$, PQ-Trade worker mobility prediction module estimates expected velocity, $\hat{E}_m(v)$ and expected direction, $\hat{E}_m(\emptyset)$ from the velocity and direction of previous n number of slots.

Now we can draw a line in the direction of a worker $m \in \mathcal{M}$ as follows,

$$y - y_m = \chi \times (x - x_m), \tag{A.1}$$

where, (x, y) is any co-linear point on the line, $l_m(x_m, y_m)$ is the current geographical location of that worker and $\chi = tan\emptyset$ is the slope of the line. Note that, we assume each task $t \in \mathcal{T}$ has a circular AOI centered at (x_t, y_t) having radius of r_t. Thus, AOI of a task $t \in \mathcal{T}$ can be represented as,

$$(x - x_t)^2 + (y - y_t)^2 = (r_t)^2. \tag{A.2}$$

By solving Eq. A.2 and Eq. A.1 we get two intersection points $A(x', y')$, and $B(x'', y'')$ as shown in Fig. A.1, which are co-linear with (x_m, y_m). From these two points we take the point A(x', y'), in the direction of that worker. After crossing this point the worker $m \in \mathcal{M}$ would leave the AOI of task $t \in \mathcal{T}$ thus can't provide further sensing services.

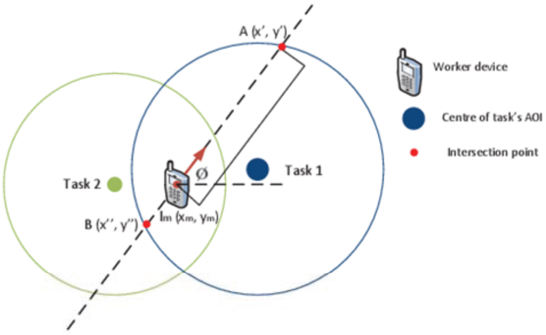


Figure A.1: Calculation of worker's sojourn time within a task's AOI

Now, the predicted time duration a worker $m \in \mathcal{M}$ stays within the AOI of task $t \in \mathcal{T}$ termed as worker sojourn time, $\mathscr{L}_t^m$ and can be calculated as follows,

$$\mathscr{L}_t^m = \frac{||A - l_m||_2}{\hat{E}_m(v)}, \tag{A.3}$$

where, $||A - l_m||_2 = \sqrt{(x' - x_t)^2 + (y' - y_t)^2}$ is the Eucledian distance between (x', y') and (x_t, y_t).

Appendix B

List of Acronyms

AOI	Area of Interest
FFP	First-Fit Profit
FFU	First-Fit Utility
GPS	Global Positioning System
ILP	Integer Linear Programming
LP	Linear Programming
MONLP	Multi-Objective Non-Linear Programming
MCS	Mobile Crowdsourcing

Appendix C

List of Notations

$\mathcal{T}$	Set of tasks advertised by the platform P
l_t	Location of a task $t \in \mathcal{T}$
r_t	Radius of task t's AOI
$\mathcal{D}_t$	Delay deadline of task $t \in \mathcal{T}$
$\mathscr{W}_t$	Total workload of task $t \in \mathcal{T}$
V_t	Value of task $t \in \mathcal{T}$
$\mathcal{M}$	Set of available worker devices
l_m	Current location of worker $m \in \mathcal{M}$
$\mathscr{B}_m$	Set of bids sent by a worker $m \in \mathcal{M}$
w_m^t	Offered workload of task t from a worker $m \in \mathcal{M}$
c_m^t	Claimed cost of worker m for a task $t \in \mathcal{T}$
d_m^t	Task completion delay for task $t \in \mathcal{T}$, $m \in \mathcal{M}$
$\mathscr{B}'$	Set of winning bids
$\mathscr{L}_m^t$	Sojourn time of worker m in task t's AOI
$\mathcal{U}_M$, $\mathcal{U}_D$, $\mathcal{U}_Q$	Workers' utility for mobility, location and past sensing quality, respectively
$\mathcal{U}_m^t$	Combined utility of worker m for task t
$\mathcal{V}_m^t$	Monetary value of workers' utility
$\mathcal{P}_m^t$, ρ_m^t	Platform total and normalized profit for allocating task t to worker m

Appendix D

List of Publications

International Journal Papers (SCI/SCIE-indexed)

1. ———, ———, ———, “A Multiconstrained QoS Aware MAC Protocol for Cluster-Based Cognitive Radio Sensor Networks”, International Journal of Distributed Sensor Networks (IJDSN), vol. 11, no. 5, May 13, 2015.

International Conference Papers

2. ———, ———, ———, “Tradeoffs Between Sensing Quality and Energy Efficiency in Context Monitoring Applications,” 2016 International Conference on Networking Systems and Security (NSysS), IEEE, pp. 1–7, January 2016.