QUAD: A Quality Aware Multi-Unit Double Auction Framework for IoT-Based Mobile Crowdsensing in Strategic Setting

Vikash Kumar Singh∗  Anjani Samhitha Jasti†  Sunil Kumar Singh‡  Sanket Mishra§

March 18, 2022

Abstract

Crowdsourcing with the intelligent agents carrying smart devices is becoming increasingly popular in recent years. It has opened up meeting an extensive list of real life applications such as measuring air pollution level, road traffic information, and so on. In literature this is known as mobile crowdsourcing or mobile crowdsensing. In this paper, the discussed set-up consists of multiple task requesters (or task providers) and multiple IoT devices (as task executors), where each of the task providers is having multiple homogeneous sensing tasks. Each of the task requesters report bid along with the number of homogeneous sensing tasks to the platform. On the other side, we have multiple IoT devices that reports the ask (charge for imparting its services) and the number of sensing tasks that it can execute. The valuations of task requesters and IoT devices are private information, and both might act strategically. One assumption that is made in this paper is that the bids and asks of the agents (task providers and IoT devices) follow decreasing marginal returns criteria. In this paper, a truthful mechanism is proposed for allocating the IoT devices to the sensing tasks carried by task requesters, that also keeps into account the quality of IoT devices. The mechanism is truthful, budget balance, individual rational, computationally efficient, and prior-free. The simulations are carried out to measure the performance of the proposed mechanism against the benchmark mechanisms. The code and the synthetic data are available at https://github.com/Samhitha-Jasti/QUAD-Implementation.

Keywords: Crowdsensing; Internet of Things; Auctions; Strategic; Quality; Truthfulness

1 Introduction

Over the past decades, there has been an unprecedented growth of the mobile users with smartphones (mobile phones with embedded-sensors). As per the Ericsson mobility report of 2017 Heuveldop

∗School of Computer Science and Engineering, VIT-AP University, Amaravati, India, Email: vikash.singh@vitap.ac.in
†School of Computer Science and Engineering, VIT-AP University, Amaravati, India, Email: samhitha.20bcd7070@vitap.ac.in
‡School of Computer Science and Engineering, VIT-AP University, Amaravati, India, Email: sunil.singh@vitap.ac.in
§School of Computer Science and Engineering, VIT-AP University, Amaravati, India, Email: sanket.mishra@vitap.ac.in
the number of worldwide mobile subscriptions is growing at around 4% every year, reaching to 9.1 billion in 2022. As the mobile users (or crowd workers) are equipped with sensing devices, so the researchers thought of utilizing these mobile users for sensing and collecting data for several real-world applications and then distributing it to the community or organization. For example, measuring the air pollution level across the cities Pan et al. (2017); Samulowska et al. (2021); Li et al. (2019), giving information about the road traffic Staniek (2021); Aubry et al. (2014), noise pollution assessment Pödör and Szabó (2021); Rana et al. (2010), information about the potholes Enigo et al. (2016); Wu et al. (2020), and many more Nagatani et al. (2013); Poblet et al. (2014). The process of completion of task(s) by the crowd workers or group of common people equipped with sensing devices in the form of an open call give rise to a new paradigmatic field of study termed as mobile crowdsensing (a.k.a. mobile crowdsourcing or crowdsensing) Hou et al. (2019); Ang et al. (2022); Peng et al. (2015); Phuttharak and Loke (2019); Dasari et al. (2020). The general framework of the mobile crowdsensing consists of three entities: (1) task requester(s) or task provider(s), (2) platform (or third party), and (3) task executors (crowd workers with smart devices). First the task provider(s) provide the sensing tasks to the platform. Once the platform receives the sensing tasks, it is supplied to the task executor(s) that are present on the other side of the crowdsensing market for execution purpose. The task executors equipped with sensing devices completes the tasks and submits the completed tasks to the platform. The platform gives back the completed tasks to the task provider(s). The task executors receives the incentive in return of their services. In mobile crowdsensing market Singh et al. (2020b); Dasari et al. (2020); Abualsaud et al. (2019), the two challenges that are of major concern in strategic setting are:

1. Which set of task executors should be selected for task execution purpose?

2. What incentives is to be given to the task executors in exchange of their services?

Most of the works in mobile crowdsensing is carried out answering the questions raised above in points 1 and 2 Mukhopadhyay et al. (2022); Singh et al. (2020a,b); Tan and Jiang (2019), in strategic setting. Apart from the challenges mentioned above, another challenge that persists in mobile crowdsourcing system is to have a large group of crowd workers in crowdsensing market. But, the question is: how to drag more number of crowd workers to the mobile crowdsensing market? One of the plausible solutions is to provide incentives to the crowd workers in exchange of their services. In past, several works have been carried out that design the schemes such that the task executors receive incentives in some terms (may be money Mukhopadhyay et al. (2022); Singh et al. (2020a); Phuttharak and Loke (2019); Singh et al. (2020b), or some social recognition Singh and Mishra (2022)). Another challenging aspect in crowdsensing environment is to get the quality data from the task executors Singh et al. (2020a); Liu and Chen (2016); Yin and Chen (2015). For this purpose, in past, the mechanisms are proposed that along with truthfulness keeps track of quality task executors Singh et al. (2020b); Mukhopadhyay et al. (2022); Singh et al. (2020a); Yin and Chen (2015). In Singh et al. (2020a) a truthful mechanism is proposed for the crowdsourcing set-up in combinatorial environment that also ensure that the task requesters receive quality data. In Mukhopadhyay et al. (2022), a budget feasible truthful mechanism is proposed for the set-up with single task requester and multiple IoT devices (as task executors) in strategic

---

1 By strategic, it is meant that the participating agents will try to maximize their utility by mis-reporting their private information. Here, utility is quasi-linear utility, means the difference between the true valuation and the payment. By private it is meant that the bids and asks are only known to the respective agents and not known to others.
setting. Here, the task requesters have limited budget along with the constraint that the overall budget is not available apriori but arrive in the system on incremental basis in several rounds. The set-up with multiple task requesters and multiple IoT devices with each of the task requesters is having a single task along with the budget, is discussed in Singh et al. (2020b). For this, a truthful mechanism is proposed that also take into account the quality of IoT devices along with satisfying the constraint that the total payment made to the IoT devices is within the budget of the respective task requesters.

Motivated from the above discussed scenarios, in this paper, one of the scenarios of IoT-based mobile crowdsensing is investigated using double auction framework. The detailed overview of the proposed framework is shown in Figure 1. In this, there are multiple task requesters and multiple IoT devices. Each of the task requesters is having multiple homogeneous tasks and each IoT device is capable of performing subset of homogeneous tasks\(^2\). Based on the type of tasks floated by the task requesters and the type of tasks the IoT devices are capable of performing, the task requesters and IoT devices are categorized into multiple different categories. In each category, the task requesters submit the tasks along with the bids to the platform. On the other side, there are multiple task executors that reveals the number of tasks that they can perform along with the cost (or \textit{ask}) for executing the tasks. The bids and asks of the agents are termed as valuations. The valuation of the agents are \textit{private} information. As there are multiple task requesters and multiple task executors, so the problem can be modeled using double auction. It is to be noted that the proposed framework is a two-phase process. In the first phase, the set of quality IoT devices is determined. For this purpose, the platform gives the part of the available tasks to the IoT devices for execution purpose. Each of the IoT devices execute the tasks and submits the completed tasks back to the platform. Once the platform receives the completed tasks, it distributes the completed tasks to other IoT devices for grading (or ranking) purpose. The IoT devices provide the ranking over the completed tasks and submit the ranks of the tasks to the platform. Based on the ranking received, the set

\[ \text{For example, one task requester may have tasks of measuring air pollution at different locations, other task requester may have tasks of detecting road traffic through different routes in the city, and so on. On the other hand, a particular set of IoT devices will be capable of measuring the air pollution level, or some may be having capability of getting information about the road traffic and so on. The agents participating in measuring air pollution level belongs to one category and the agents participating in determination of road traffic condition belongs to other category and likewise.} \]
of quality IoT devices is selected among the available IoT devices by the platform. The output of the first phase is a set of quality IoT devices. In the second phase, the platform allocate the tasks to the quality IoT devices for execution purpose. The quality IoT devices execute the tasks and submit it back to the platform. The platform gives the completed tasks to the respective task requesters and the IoT devices get paid in return of their services. The bid and ask values reported by the task requesters and IoT devices respectively are private information.

In this paper, for the problem under consideration a two-fold mechanism namely quality aware multi-unit double auction mechanism (QUAD) is proposed. In the first fold, the set of quality IoT devices is determined using the subroutine IoT-devices quality determination mechanism motivated by Nisan et al. (2007); Roughgarden (2016). After that in the second fold from among the set of quality IoT devices, the subset of IoT devices are selected as winners and payment are made using allocation and pricing rule motivated by Segal-Halevi et al. (2018b); Roughgarden (2014). In the upcoming subsection the contributions made in this paper are listed.

1.1 Contributions

In the presence of strategic agents, the objective is to design a truthful mechanism that will do the following: (1) determine the quality IoT devices, (2) which IoT devices will execute the tasks?, and (3) what price will be paid to the winning IoT devices? In particular, our contributions are:

- One of the scenarios of crowdsensing is investigated in strategic setting. The set-up consists of multiple task requesters and multiple IoT devices with each of the task requesters have multiple sensing tasks that are to be executed. The objective is to assign the floated sensing tasks to the quality IoT devices and to incentivize the IoT devices in return of their services.

- A two-fold truthful mechanism is proposed namely QUAD for allocating tasks to the IoT devices. In first fold, a very small part of the tasks from the available tasks is given to the IoT devices for execution purpose for estimating the quality of the IoT devices. It is required as it will help us to assure that the quality sensing data is provided by the IoT devices to the task providers. In second fold, the set of quality IoT devices is considered as winners and the payment is decided by utilizing the concept of mechanism design.

- Theoretical analysis is carried out in Section 5 to show that the proposed mechanism is computationally efficient (Lemma 5.1), correct (Lemma 5.2), prior free (Lemma 5.3) truthful (Lemma 5.4), individual rational (Lemma 5.5), and budget balance (Lemma 5.6). Further probabilistic analysis is carried out to have an estimate on the number of tasks that will be executed by the quality IoT devices for any task provider (Lemma 5.7 and Lemma 5.8).

- In the simulation results it is shown that QUAD is not vulnerable to manipulation and is compared with the benchmark mechanism namely posted price mechanism (PPM) Roughgarden (2014) and McAfee double auction (DA) McAfee (1992); Bredin and Parkes (2005). QUAD is compared with PPM on the ground of truthfulness and budget balance. Further, QUAD is compared with McAfee DA on the basis of satisfaction level of the agents.

1.2 Paper Organization

The remainder of the paper is structured as follows. In Section 2 related works in the fields of double auction, mobile crowdsourcing, and IoT is discussed. Section 3 describes our proposed
model and discuss about the game theoretic properties. The proposed mechanism, namely, QUAD is illustrated in section 4. In section 5 the theoretical and probabilistic analysis of QUAD is carried out. The simulation results are depicted in section 6. The paper is concluded with the possible future directions in section 7.

2 Related Prior Works

In this section, the related works in the areas of double auction, mobile crowdsensing, and Internet of Things (IoT) are discussed one by one in the given order.

2.1 Double Auction

The double auction mechanism is a subroutine in QUAD; therefore, the related works in double auction is discussed in this section. In a double auction, there are multiple sellers who are ready to sell their items, multiple buyers who are ready to buy the available items, and an auctioneer (or third-party). The double auction gives the flexibility to the two parties (buyers and sellers) to reveal their privately held type simultaneously and can act strategically McAfee (1992); Nisan et al. (2007); Myerson and Satterthwaite (1983). A double auction is a mechanism that provide the platform for deciding that who will buy a particular item and at what price? It is applicable to several application domains such as spectrum market Zhou and Zheng (2009); Li et al. (2016); Leyton-Brown et al. (2017), Internet advertisement Feldman and Gonen (2016), emission trading market Sun and Li (2020), etc. For two-sided market, the impossibility result by Myerson and Satterthwaite (1983) states that any mechanism that is incentive compatible (IC), budget balance (BB), and individual rational (IR) cannot maximize Gain-from-trade\(^3\) (or efficiency).

In past, several works have been carried out in double auction Plott and Gray (1990); McAfee (1992); Deshmukh et al. (2002). In McAfee (1992) the set-up is, there are multiple sellers and multiple buyers. Each seller wants to sell a single unit of good and each buyer wants to buy a single unit of good. A truthful mechanism is proposed for matching buyers to sellers. In addition to truthfulness, McAfee’s mechanism is individual rational, budget balance, and prior-free. By prior-free, it is meant that the mechanism does not make any assumption on the type of valuations the agents reveal. Some double auction mechanism makes assumption on the valuation of the agents such as agent’s valuation as decreasing marginal returns (DMR) Blumrosen and Dobzinski (2014); Segal-Halevi et al. (2018b), additive valuation Hirai and Sato (2021); Goel et al. (2016); Feldman and Gonen (2016), gross substitute Segal-Halevi et al. (2018a) or their valuation is represented by single parameter Gonen and Egri (2017); Gonen et al. (2007). In Blumrosen and Dobzinski (2014) a mechanism is proposed for multi parametric agents with DMR valuation but is not asymptotically efficient. The competitive ratio achieved is \(\frac{1}{\ln m}\). In Segal-Halevi et al. (2018b) a truthful double auction mechanism is proposed that allows trading of multiple units of item per agent but with the constraint that the agents have decreasing marginal return valuation. Theoretical analysis shows that the competitive ratio of the proposed mechanism is at least \(1 - O(M \sqrt{\frac{\ln mk}{mk}})\). In Segal-Halevi et al. (2018a) a truthful mechanism is designed for the set-up with multiple sellers and multiple buyers. Each of the sellers have multiple distinct goods and each buyer have infinite amount of money. A double auction based truthful mechanism is proposed that allow trading of multiple units of distinct goods per agents but with the constraint that the agents have gross substitute valuation.

\(^3\)It is defined as the difference between the total valuations of the buyers to the total asks of the sellers.
In this paper, we have utilized the idea presented in Segal-Halevi et al. (2018b) for allocating the tasks to the IoT devices and deciding their payment, once the quality IoT devices are determined.

2.2 Mobile Crowdsensing

In order to get an overview on the current trend in mobile crowdsensing the readers can follow Liu et al. (2018); Abualsaud et al. (2019); Dasari et al. (2020); Kong et al. (2019); Hou et al. (2019). As discussed above, one of the major challenges in mobile crowdsensing is how to motivate the group of workers carrying smartphones towards the crowdsourcing market?

For this purpose, several incentive based schemes are proposed for incentivizing the crowd workers Mukhopadhyay et al. (2022); Feng et al. (2014); Yang et al. (2012); Tan and Jiang (2019); Wang et al. (2018). In Mukhopadhyay et al. (2022) the set-up with single task requester and multiple task executors in mobile crowdsourcing is investigated in strategic setting. Here, the task requester has single task and the limited budget associated with that task. However, the constraint that is preserved in the discussed set-up is that the overall budget is not available apriori but comes in incremental manner in the system in several rounds. Given the above discussed set-up a truthful mechanism is designed that ensures that the total payment made to the task executors is within the available budget. In Feng et al. (2014) the set-up is such that the platform float the sensing tasks alongwith its location information, and on the other side of the MCS market the smartphone users provide the bids for the sensing tasks falling within their coverage area. The bids of the smartphone users are private information. For this purpose, a reverse auction based mechanism is proposed namely TRAC. TRAC is a two-phase mechanism. In the first phase, a near-optimal approximation algorithm is developed for determining the winners with low computation complexity. The second phase determines the payment of the winners. Wang et al. (2018) proposed a truthful mechanism that selects the quality crowd workers for completing the tasks. One of the constraint that is preserved in the set-up is that the quality of the crowd workers is changing frequently. In Tan and Jiang (2019) auction is coupled with the experience model to have a fair competition among the crowd workers. In Yang et al. (2012) the two separate incentive mechanisms are proposed for user-centric model and platform-centric model respectively. In case of platform-centric model, only one sensing task is floated and the total reward will be shared among the winning mobile phone users. For designing an incentive mechanism Stackelberg game is used, where the utility of the platform is maximized by calculating the unique Stackelberg equilibrium. For the user-centric model an auction-based incentive mechanism is proposed where the mobile phone users can select multiple tasks and can report their bid price to the platform. Based on the reported bid prices, the platform will decide the winners.

Also, in some cases the payment made to the task executors depend on the quality of the work performed by Yin and Chen (2015); Liu and Chen (2016). In Yin and Chen (2015) the prime focus is to provide the bonuses to the task executors for their exceptional work, so as to improve the overall utility of a task requester. In order to have the information about the impact of bonuses on the quality of work supplied by the task executors, the hidden Markov model is used. In Liu and Chen (2016), the output agreement mechanisms are used to have the true answers from the group of task executors. In Huang et al. (2017) the set-up consists of multiple task consumers and multiple smartphone users, where each of the task consumers have multiple tasks to complete. The objective is to assign the sensing tasks to the smartphone users for execution purpose. For the above discussed set-up a truthful mechanism is proposed that achieves max-min fairness. In Wei et al. (2015) the focus is on two-sided mobile crowdsourcing market that consists of multiple...
service users, platform, and multiple service providers. Here, each of the service users can request for single service and each of the service providers can provide single service. The above discussed set-up is modeled using the double auction framework and a truthful mechanism is proposed that also satisfies individual rationality, and budget balance. In Hamrouni et al. (2021) the focus is on collaborative mobile crowdsourcing, where the tasks are accomplished by the group of IoT devices that communicate among themselves and share their operational activities.

From the above discussed literature reviews it can be seen that the set-up discussed in this paper in IoT-based mobile crowdsensing in strategic setting is not considered. In this paper, for the discussed set-up a truthful mechanism is proposed that allocates the tasks provided by the task requesters to the quality IoT devices for execution purpose.

2.3 Internet of Things (IoT)

For detailed overview in IoT the readers can refer the following research works Liu et al. (2018); Atzori et al. (2010); Ray (2018); Ray et al. (2018); Al-Fuqaha et al. (2015); Ang et al. (2022). The term ‘Internet of Things’ first came into picture in the year 1998 Ashton (1999). Later on the International Telecommunication Union (ITU) formally gave the concept of IoT in the year 2005.

| Research paper                  | Research directions in IoT               |
|---------------------------------|-----------------------------------------|
| Abraham et al. (2021)           | Smart agriculture                       |
| Baccour et al. (2020)           | Privacy                                 |
| Qolomany et al. (2021)          | Smart cities                            |
| Sun et al. (2020)               | Privacy and transportation              |
| Siarry et al. (2021)            | Healthcare                              |
| Zhou et al. (2017)              | Security and privacy                    |
| Rahman et al. (2019)            | Healthcare and well being               |
| Talavera et al. (2017)          | Industries, agriculture, and environment|
| Alavi et al. (2018)             | Transportation, health, and smart cities|
| García et al. (2017)            | Security, privacy, and architecture      |
| Pereira and Aguiar (2014)       | Quality of service, scalability, and interoperability |
| Li et al. (2015)                | Security, privacy, and reliability.     |
| Zhang (2011)                    | Data processing, security, and privacy  |
| Zanella et al. (2014)           | Transportation, health, and smart cities|
| Yan et al. (2014)               | Quality of service (QoS), identification, and authentication |
| Khajenasiri et al. (2017)       | Energy and environment                  |

It is estimated by the large number of companies and the research professionals that the IoT will contribute 4%-11% of global GDP in the year 2025 Index (2015). Huawei predicts that by 2025 there will be approximately 100 billion IoT connections Manyika et al. (2015). Internet and
the advancement of recent technologies have been the catalyst for the research in IoT. The IoT have many application areas but not limited to healthcare Alavi et al. (2018), agriculture Qiu et al. (2013), environment Talavera et al. (2017) etc. Table 2 depicts the various application areas of IoT.

In the upcoming section the problem of mobile crowdsensing is formulated using double auction framework and is discussed in detailed manner.

3 Preliminaries

First the model and notations are discussed. After that game theoretic properties are discussed.

3.1 Model and Notation

In this section, the problem is formulated by utilizing the concept of double auction. There are \( m \) task requesters and \( n \) IoT devices, such that \( m < n \). In this model it is considered that task requesters and IoT devices are heterogenous by nature. By heterogeneity it is meant that the task requesters may float different types of tasks (such as measuring air pollution, road condition, and so on) and IoT devices may vary in terms of type of tasks they are capable to perform. However, a particular task requester will be endowed with similar type of tasks and particular IoT device will be capable of performing one type of task. Based on the type of tasks the task requesters are floating and the type of tasks the IoT devices can perform, the task requesters and IoT devices are categorized into different categories. Let us say, we have \( k \) different categories and is given as \( w = \{ w_1, w_2, \ldots, w_k \} \), where \( w_i \) represents \( i^{th} \) category.

In any \( w_i \) category, say, we have \( m_i \) task requesters and \( n_i \) IoT devices. The set of task requesters is given as \( r_i = \{ r_{i1}, r_{i2}, \ldots, r_{im_i} \} \), where \( r_{ij} \) represents \( j^{th} \) task requester in \( w_i \) category. The set of IoT devices is given as \( I_i = \{ I_{i1}, I_{i2}, \ldots, I_{in_i} \} \), where \( I_{ik} \) represents \( k^{th} \) IoT device in \( i^{th} \) category. The set of task requesters and IoT devices in all the \( k \) categories is given as \( r = \{ r^1, r^2, \ldots, r^k \} \) and \( I = \{ I^1, I^2, \ldots, I^k \} \) respectively. The task requesters are endowed with multiple similar tasks and the bids (maximum price he is ready to pay). Let us say that any \( k^{th} \) task requester is endowed with at most \( Q_i^k \) number of similar tasks. On the other side of MCS market, the IoT devices report the number of tasks that they can execute along with the asks (charge for imparting its services). Any \( j^{th} \) IoT device can execute at most \( Q_j^i \) number of similar tasks. In our case, the task requesters and IoT devices taken together will be termed as agents and both may behave strategically. The bids and asks of the agents will be termed as valuation as and when required.

In our set-up the discussed problem is studied as a two-phase process. In the first phase, among the available IoT devices, the set of quality IoT devices is determined for executing the tasks. For this purpose, firstly, a part of available tasks is given to the IoT devices by the platform for execution. On receiving the tasks the IoT devices execute the tasks and submits the completed tasks back to the platform. Now, each of the completed tasks is given to the peers (other IoT devices) for the review purpose. In any category \( w_i \), if \( j^{th} \) IoT device prefers \( k^{th} \) IoT device over \( l^{th} \) IoT device then it is represented as \( I_i^k \succ_j I_i^l \). For all the IoT devices in \( i^{th} \) category the rank profile is represented as \( \succ^i = \{ \succ_i^1, \succ_i^2, \ldots, \succ_i^{n_i} \} \). For all the \( k \) categories, the rank profile is given as \( \succ = \{ \succ^1, \succ^2, \ldots, \succ^k \} \). After reviewing the completed tasks, the review report is submitted to the platform and the quality IoT devices are determined. Once the quality IoT devices are determined, next, the below mentioned challenges are handled:

- Which quality IoT devices should be selected for tasks execution purpose?
• What will be the payment made to the selected quality IoT devices?

In the second fold, the above two issues are resolved. In any category $w_i$, each agent $k$ has a valuation function $\nu^i_k$ that returns, for every integer $f$ ($0 < f \leq Q^i_k - 1$), the agent’s value for owing $f$ units. It is to be noted that the valuation for owing zero unit is 0. In our discussed setup, all the agents posses DMR valuation function. For any agent $j$ in $i^{th}$ category, by DMR valuation function $\nu^i_j$ we mean that $\nu^i_j(f) - \nu^i_j(f - 1) \geq \nu^i_j(f + 1) - \nu^i_j(f)$. It means that, marginal utility for an agent from having one more task is weakely-decreasing in his current number of tasks.

Table 2: Notations used

| Symbols | Description |
|---------|-------------|
| $m$ | Number of task requesters |
| $n$ | Number of IoT devices |
| $w$ | $w = \{w_1, w_2, \ldots, w_k\}$: Set of $k$ different categories |
| $m_i$ | Number of task requesters in $w_i$ category |
| $n_i$ | Number of IoT devices in $w_i$ category |
| $\mathcal{I}$ | $\mathcal{I} = \{\mathcal{I}^1, \mathcal{I}^2, \ldots, \mathcal{I}^k\}$: Set of IoT devices in all the $k$ categories |
| $\mathcal{I}_i^i$ | $\mathcal{I}_i = \{\mathcal{I}_i^1, \mathcal{I}_i^2, \ldots, \mathcal{I}_i^{n_i}\}$: Set of IoT devices in $w_i$ category |
| $r$ | $r = \{r^1, r^2, \ldots, r^k\}$: Set of task requesters in all the $k$ categories |
| $r^i$ | $r^i = \{r^i_1, r^i_2, \ldots, r^i_{m_i}\}$: Set of task requesters in $w_i$ category |
| $\succ^i_k$ | Rank list of $k^{th}$ IoT device in $i^{th}$ category. |
| $\succ^i$ | $\succ^i = \{\succ^i_1, \succ^i_2, \ldots, \succ^i_{n_i}\}$: Rank profile of all the IoT devices in $i^{th}$ category. |
| $Q^i_k$ | Maximum number of tasks held by any $k^{th}$ agent in $i^{th}$ category |
| $\nu^i_j$ | Valuation function of agent $j$ in $i^{th}$ category |
| $\nu^i_{j,f}$ | Agent $j$ marginal value for having $f^{th}$ task in $i^{th}$ category |
| $p$ | Equilibrium price |
| $u^i_j(f, p)$ | Utility of task requester $j$ from buying $f$ units in $i^{th}$ category at price $p$ |
| $z^i_j(f, p)$ | Utility of IoT device $j$ from selling $f$ units in $i^{th}$ category at price $p$ |
| $d^i_k(p)$ | Demand of any task requester $k$ at a given price $p$ in $i^{th}$ category |
| $s^i_j(p)$ | Supply of any IoT device $j$ at a given price $p$ in $i^{th}$ category |
| $d^R$ | Total demand at a given price $p$ in right crowdsensing arena in $i^{th}$ category |
| $s^R$ | Total supply at a given price $p$ in left crowdsensing arena in $i^{th}$ category |

The reason behind restricting the valuation function to DMR is that, in our discussed set-up
the equilibrium price vector\(^4\) exists only when we have a DMR valuation. In any category \(w_i\), if any agent \(k\) have \(Q^i_k\) homogeneous tasks, then it is represented as \(Q^i_k\) single unit virtual agents. The value of virtual-agent \(f\) of agent \(k\) is the agent’s marginal value for having the \(f\)th unit and is given as \(\nu^i_{k,f} = \nu^i_k(f) - \nu^i_k(f - 1)\) for \(f \in \{1, \ldots, Q^i_k\}\). The ties between the marginal values of the virtual agents are broken randomly. In any category \(w_i\), given an equilibrium price \(p\) for each task, the utility of task requester \(k\) resulted by buying \(f\) units of completed task is:

\[ u^i_k(f, p) = \begin{cases} \nu^i_k(f) - p \cdot f, & \text{if agent } k \text{ receives } f \text{ executed tasks} \\ 0, & \text{otherwise} \end{cases} \]  

Similarly, the utility of IoT device \(j\) by supplying sensed data for \(f\) tasks is:

\[ z^j(f, p) = \begin{cases} f \cdot p - (\nu^j_1(Q^j) - \nu^j_1(Q^j - f)), & \text{if agent } j \text{ provides sensed data for } f \text{ tasks} \\ 0, & \text{otherwise} \end{cases} \]  

In any category \(w_i\), the demand of any task requester \(k\) at an equilibrium price \(p\) is the set of tasks that maximizes the value \(u^i_k(f, p)\) and is given as:

\[ d^i_k(p) = \arg\max_{f \in [0, Q^i_k]} u^i_k(f, p) \]  

Here, if the valuation of any \(k\)th task requester is following DMR, then \(d^i_k(p)\) is just the number of virtual task requesters whose bid value is greater than equilibrium price \(p\) i.e. \(\nu^i_{k,j} > p\). The total demand at an equilibrium price \(p\) is the sum of the demands of all the task requesters and is given as \(d^i = \sum_{k=1}^{m_i} d^i_k(p)\). Similarly, the supply from any \(j\)th IoT device at an equilibrium price \(p\) is the set of tasks that maximizes the value \(z^j(f, p)\) and is given as

\[ s^j_j(p) = \arg\max_{f \in [0, Q^j_j]} z^j_j(f, p) \]  

Here, if the valuation of any \(j\)th IoT device is following DMR, then \(s^j_j(p)\) is just the number of virtual IoT devices whose ask value is less than equilibrium price \(p\) i.e. \(\nu^i_{j,f} < p\). The total supply at an equilibrium price \(p\) is the sum of the supply from all the IoT devices and is given as \(s^i = \sum_{j=1}^{n_i} s^j_j(p)\).

Let us try to understand DMR valuation with the help of an example given below.

**Example 1.** In any category \(w_i\), the bid value reported by any agent \(k\) for 3 units of a task is 10. For 1st unit the value is say 5 i.e. \(\nu^i_{1,1} = 5\). Similarly, for 2nd unit and 3rd unit the value is given as \(\nu^i_{1,2} = 4\) and \(\nu^i_{1,3} = 1\) respectively. Here, the valuation function \(\nu^i_k\) of agent \(k\) is DMR, if the equilibrium price is set as 3 then the demand of \(k\) will be 2 units, as \(\nu^i_{1,1}\) and \(\nu^i_{1,2}\) values are above equilibrium price. However, if the bid value configuration is such that \(\nu^i_{1,1} = 2, \nu^i_{1,2} = 5, \) and \(\nu^i_{1,3} = 3\) then the demand will be 0 even if the marginal values reported by the virtual agents are above the equilibrium price. The reason is that the valuation function here is not following DMR criteria. Similar argument can be given for IoT devices.

\(^{4}\)By *equilibrium price* it is meant that, a price at which the number of tasks floated by the task requesters is equal to the number of tasks executed by the IoT devices. In other words, we say it as supply equals demand.
3.2 Game Theoretic Properties

In this section, the five game theoretic properties that will be utilized in this paper are discussed.

**Definition 1 (Prior-free).** A mechanism is prior-free, if it does not use any statistical information on the valuation of the agents.

**Definition 2 (Truthful or Dominant Strategy Incentive Compatible (DSIC)).** A mechanism is truthful, if no matter how other agents are bidding, no agent $j$ can improve his utility by mis-reporting his valuation i.e. $u_j^i(f,p) \leq \hat{u}_j^i(f,p)$ in case of task requester and $z_j^i(f,p) \leq \hat{z}_j^i(f,p)$ in case of task executor.

**Definition 3 (Individual Rationality (IR)).** A mechanism is IR, if no winning task executor is paid less than his ask value and no winning task provider pays more than his bid value. It means that the utility of the agents should be at least 0.

**Definition 4 (Strongly Budget-Balanced (SBB)).** A mechanism is SBB, if the total payment of the agents is exactly 0. It means that the mechanism neither incurs surplus nor incurs deficit.

**Definition 5 (Weakly Budget-Balanced (WBB)).** A mechanism is WBB, if the total payment of the participating agents is positive. It means that the mechanism incurs a surplus.

4 Proposed DSIC Mechanism

In this section, a truthful mechanism namely QUAD motivated by Nisan et al. (2007); Roughgarden (2016, 2014); Segal-Halevi et al. (2018b) is presented and discussed. QUAD determines the quality of the IoT devices along with deciding which of the quality IoT devices will be hired for task execution and what will be their payment? The QUAD consists of three components:

- **k-category procedure** – The reason for developing $k$-category procedure is to process all the $k$ different categories of the task requesters and IoT devices present in MCS.

- **IoT-devices quality determination mechanism** (motivated by Nisan et al. (2007); Roughgarden (2016)) – To hire quality IoT devices.

- **Allocation and pricing rule** (motivated by Segal-Halevi et al. (2018b); Roughgarden (2014)). The allocation and pricing rule consists of three subroutine, namely
  1. Splitting and equilibrium price determination,
  2. Demand and supply calculation, and
  3. Winner determination and payment

In the upcoming subsections, the components of the proposed mechanism are discussed.

4.1 $k$-category Procedure

In $k$-category procedure, for each category, in lines 2-5: (a) call to subroutine IoT-QDBC is made – for selecting the quality IoT devices from the available ones, (b) call to split and equilibrium price determination subroutine is made – to distribute the agents into two different MCS arenas
and determine the equilibrium price in both the arenas, (c) call to demand and supply calculation is made to determine the demand and supply of the agents in the respective MCS arenas by using the equilibrium prices of the opposite MCS arena, and (d) call to winner determination and payment is made to decide the winning agents and their payment. For each category $w_i$, in line 6, the winning IoT devices and winning task requesters are captured in $I^w(i)$ and $r^w(i)$ respectively. In line 7 the equilibrium price of right mobile crowdsourcing arena is stored in $p_f$ and after that $p_R$ is reset to $\phi$. Similarly, the equilibrium price of left crowdsourcing arena can also be calculated and stored. In line 9, the list of winning task requesters i.e. $r^w$ and list of winning IoT devices i.e. $I^w$ for all the categories are returned.

**ALGORITHM 1:** k-category procedure $(\mathcal{I}, r, \succ, w)$

Output: $I^w \leftarrow \phi$, $r^w \leftarrow \phi$, $p_f \leftarrow \phi$

1. foreach $w_i \in w$ do
   2. $\tilde{Q}^i \leftarrow$ IoT-QDBC $(\mathcal{I}^i, \succ^i)$ // Call to IoT-QDBC subroutine is made and the quality IoT devices are held in $\tilde{Q}^i$.
   3. $(I^i_L, r^i_L, p_R) \leftarrow$ Splitting and equilibrium price determination $(\tilde{Q}^i, r)$ // Call to Splitting and equilibrium price determination subroutine is made.
   4. $(d^L, s^L) \leftarrow$ Demand and supply calculation $(I^i_L, r^i_L, p_R)$ // Call to Demand and supply calculation subroutine is made.
   5. $(I^w(i), r^w(i)) \leftarrow$ Winner determination and payment $(d^L, s^L, \mathcal{I}^L_i, r^i_L, p_R)$ // Call to Winner determination and payment subroutine is made.
   6. $I^w \leftarrow I^w \cup I^w(i)$; $p_f \leftarrow p_f \cup r^w \cup r^w(i)$
   7. $p_f \leftarrow p_f \cup p_R; p_R \leftarrow \phi$
8. end
9. return $I^w, r^w, p_f$

### 4.2 IoT-devices Quality Determination Mechanism

In this, for hiring the quality IoT devices among the available IoT devices, a subroutine namely IoT-devices quality determination using borda count (IoT-QDBC) motivated by Nisan et al. (2007); Roughgarden (2016) is proposed.

#### 4.2.1 Outline of IoT-QDBC

In this section, the main idea of the IoT-QDBC is illustrated below.

**IoT-QDBC**

Fix a category $w_i$:

1. Each time $\gamma$ IoT devices are picked up randomly that provide full rank list over the $\beta$ other IoT devices that are chosen randomly from the $(n - \gamma)$ IoT devices.

2. Now, based on the rank lists of the IoT devices, each IoT device $I^i_j$ gets $\beta$ points for each first preference, $\beta - 1$ points for each second preference, and so on, with 1 point for each last preference.

3. The process iterates until all the IoT devices are not ranked.
4.2.2 Detailing of IoT-QDBC

The detailing of IoT-QDBC is depicted in below listing. In line 1 of Alg. 2, the lists \( g \) and \( s \) are initialized to \( \phi \). In line 2, the copies of the set of IoT devices in \( i \)th category are maintained in \( Z \) and \( J \). The \( \text{while} \) loop in line 3-16 determines the quality IoT devices. In line 4, \( \gamma \) IoT devices are picked up randomly from the list of IoT devices \( Z \) using \( \text{RSELECT} \) method and held in \( g \). \( s \) holds randomly selected \( \beta \) IoT devices from the list of IoT devices remaining after picking \( \gamma \) IoT devices in line 4. In line 6, each of the IoT devices in \( g \) provides the full rank list over the IoT devices in \( s \). Lines 7-13 determines the total point obtained by each of the IoT devices \( I \in s \) and held in \( c \). In line 14, for each iteration of \( \text{while} \) loop the IoT device having the maximum point is kept in \( Q \). The set of IoT devices that are ranked in the current iteration are removed from \( J \) in line 15. The process iterates until \( J \) becomes empty. Line 17 returns the set of quality IoT devices obtained in \( w_i \) category.

**Algorithm 2:** IoT-QDBC (\( I^i, \succ^i \))

Output: \( Q^i \leftarrow \phi \)

\[
\begin{align*}
1 & \quad g \leftarrow \phi, \ s \leftarrow \phi \quad \text{// \( g \) and \( s \) lists are initialized to \( \phi \).} \\
2 & \quad Z = J = I^i \quad \text{// Set of IoT devices in \( i \)th category are held in \( Z \) and \( J \).} \\
3 & \quad \text{while } J \neq \phi \text{ do} \\
4 & \quad \quad g \leftarrow \text{RSELECT}(Z, \gamma) \quad \text{// \( \gamma \) IoT devices are picked up from \( Z \) and held in \( g \).} \\
5 & \quad \quad s \leftarrow \text{RSELECT}(J \setminus g, \beta) \quad \text{// \( \beta \) IoT devices are picked up from \( J \setminus g \) and held in \( s \).} \\
6 & \quad \quad \text{Each } I^i_k \in g \text{ provide a full rank list } i.e. \succ^i_k \text{ over all the IoT devices in } s. \\
7 & \quad \quad \text{foreach } I^i_k \in s \text{ do} \\
8 & \quad \quad \quad c_k \leftarrow 0 \quad \text{// \( c_k \) holds the point of each IoT device and is initialized to 0.} \\
9 & \quad \quad \quad \text{foreach } I^i_j \in g \text{ do} \\
10 & \quad \quad \quad \quad c_k \leftarrow c_k + (\beta - \ell) \quad \text{// where } \ell = 0 \text{ for first preference, } \ell = 1 \text{ for second preference, likewise } \ell = (\beta - 1) \text{ for last preference.} \\
11 & \quad \quad \quad \text{end} \\
12 & \quad \quad \quad c \leftarrow c \cup \{c_k\} \quad \text{// \( c \) holds the point received by each IoT device in } s. \\
13 & \quad \quad \text{end} \\
14 & \quad Q^i \leftarrow Q^i \cup \text{argmax}_{I^i_k \in s} \{c\} \quad \text{// Selects the IoT device with maximum points and store it in } Q^i. \\
15 & \quad J \leftarrow J \setminus \{I^i_k\} \quad \text{// Removes the set of IoT devices from } J \text{ that are already ranked.} \\
16 & \quad \text{end} \\
17 & \quad \text{return } Q^i
\end{align*}
\]

**Example 2.** Let us understand IoT-QDBC algorithm with the help of an example for \( w_2 \) category. In our running example, there are 3 task requesters and 9 IoT devices. Here, \( \beta \) and \( \gamma \) values are taken as 3. Following line 4 of Alg. 2, 3 IoT devices are picked up randomly for providing the ranking. Next, the remaining 3 IoT devices are chosen that are to be ranked. The full rank lists of the IoT devices for the first iteration are depicted in Figure 3a. Now, following line 7-13 of Alg. 2 the points gained by each of the IoT devices will be calculated. As IoT device \( I^2_3 \) is ranked first by two of the IoT devices (\( I^2_2 \) and \( I^2_6 \)), and ranked second by one of the IoT device (\( I^2_7 \)), due to this reason it receives 3 points for each first rank and 2 points for second rank. So, IoT device \( I^2_3 \) gains a total of 8 points as shown in Figure 2d. In the similar fashion, the IoT devices \( I^2_1 \) and \( I^2_5 \)
gains a total of 7 points each depicted in Figure 2d. So, from the first iteration $I_7^2$ is placed in the list of quality IoT devices.

In the next iteration, again 3 IoT devices will be picked up randomly for providing the ranking over the other 3 IoT devices as shown in Figure 2b. As the IoT device $I_2^2$ is ranked first by two IoT devices ($I_2^2$ and $I_5^2$), and ranked second by one IoT device ($I_2^2$) due to this it receives 3 points for each first rank and 2 points for a second rank. So, IoT device $I_2^2$ also gains a total point of 8 as shown in Figure 2e. In the similar fashion, the IoT devices $I_1^3$ and $I_9^2$ gains a total point of 6 and 4 respectively. So, in the second iteration $I_2^2$ is placed in the list of quality IoT devices.

\[
\begin{array}{c}
\text{I}_2^2 \quad \text{I}_5^2 \quad \text{I}_1^2 \quad \text{I}_3^2 \\
\text{I}_1^3 \quad \text{I}_2^1 \quad \text{I}_3^1 \quad \text{I}_5^1 \\
\text{I}_6^3 \quad \text{I}_3^3 \quad \text{I}_2^3 \quad \text{I}_1^3
\end{array}
\]

(a) Rank list of $I_2^2$, $I_5^2$, $I_6^3$ (b) Rank list of $I_1^2$, $I_2^1$, $I_8^1$ (c) Rank list of $I_3^3$, $I_5^3$, $I_2^3$

\[
\begin{array}{c}
\text{I}_3^2 \quad \text{I}_2^2 \quad \text{I}_1^2 \quad \text{I}_6^2 \\
\text{I}_4^2 \quad \text{I}_2^2 \quad \text{I}_1^2 \quad \text{I}_6^2
\end{array}
\]

(d) Point calculation of $I_3^2$, $I_3^3$, $I_5^2$ (e) Point calculation of $I_2^2$, $I_1^2$, $I_6^2$ (f) Point calculation $I_4^2$, $I_8^2$, $I_9^2$

Figure 2: Detailed Illustration of IoT-QDBC mechanism.

In the last iteration, again 3 IoT devices are picked up randomly and rankings are provided. In this iteration, IoT device $I_3^2$ is ranked first by one IoT device ($I_3^2$), ranked second by one IoT device ($I_3^2$), and ranked third by the IoT device $I_2^3$. Due to this it receives 3 points for a first rank, 2 points for a second rank, and 1 point for a third rank. So, IoT device $I_3^2$ gains a total points of 6. In the similar fashion, the IoT devices $I_2^2$ and $I_9^2$ gains a total points of 6 each. So, in the this iteration, we can select any of the IoT devices as the quality IoT device, say, $I_3^2$ is considered in the list of quality IoT devices. Alg. 2 returns $I_3^2$, $I_2^2$, and $I_9^2$ as the list of quality IoT devices for the running example.

4.3 Allocation and Pricing Rule

Once the quality IoT devices are determined in the first phase of the proposed framework, now the goal is to tackle the issues raised in points 1 and 3 in Section 1. For this purpose, a subroutine of QUAD namely allocation and pricing rule motivated by Segal-Halevi et al. (2018b); Roughgarden (2014) is proposed. The allocation and pricing rule consists of three subroutines:

- Splitting and equilibrium price determination,
- Demand and supply calculation, and
- Winner determination and payment

In the upcoming subsections, each of the subroutines is discussed in a detailed manner one-by-one.
### 4.3.1 Splitting and Equilibrium Price Determination

Using line 1 of Algorithm 3 the set of task requesters and the IoT devices are divided into two arenas called *left mobile crowdsourcing arena (LMCA)* and *right mobile crowdsourcing arena (RMCA)*.

**ALGORITHM 3:** Splitting and equilibrium price determination $(T^i, r^i)$

```plaintext
/* Split mobile crowdsensing arena */
1 Divde the mobile crowdsensing arena into two sub-arenas namely left mobile crowdsensing arena (LMCA) and right mobile crowdsensing arena (RMCA).
2 With probability 1/2 place the task requesters and IoT devices into LMCA and RMCA independently. The task requesters and IoT devices in LMCA are held in $r^i_L$ and $I^i_L$ respectively. For RMCA, $r^i_R$ and $I^i_R$ used to hold task requesters and IoT devices respectively.

/* Equilibrium price determination in RMCA */
3 $p \leftarrow 0, d^R = \infty, s^R = 0$  // The variables are initialized to 0 and $\infty$.
4 while $d^R \neq s^R$ do
5     $p \leftarrow p + \epsilon$  // Each time the price $p$ is incremented by $\epsilon$.
6     foreach $r^i_j \in r^i_R$ do
7         $d^i_j(p) = \text{argmax}_{f \in [0, Q^i_j]} u^i_j(f, p)$  // Determines the maximum demand of task requester $r^i_j$ at price $p$ and hold the demand in $d^i_j(p)$.
8     endforeach
9     foreach $I^i_k \in I^i_R$ do
10        $s^i_k(p) = \text{argmax}_{f \in [0, Q^i_k]} z^i_k(f, p)$  // Determines the maximum supply of IoT device $I^i_k$ at price $p$ and hold the supply in $s^i_k(p)$.
11    end
12    $d^R = \sum_{j=1}^{n_i} d^i_j(p)$  // Total demand in $i^{th}$ category at price $p$ is calculated for all the $n_i$ task requesters and is stored in $d^R$.
13    $s^R = \sum_{k=1}^{m_i} s^i_k(p)$  // Total supply in $i^{th}$ category at price $p$ is calculated for all the $m_i$ IoT devices and is stored in $s^R$.
14 end
15 $p^R \leftarrow p$  // The equilibrium price of RMCA is stored in $p^R$.
16 return $I^i_L, r^i_L, p^R$  // Returns the list of IoT devices, list of task requesters in LMCA and equilibrium price of RMCA.
```

Each time a task requester or an IoT device is placed to LMCA/RMCA with probability 1/2, independent of others, as depicted in line 2. Here, for determining the equilibrium price we are considering RMCA. However, in the similar fashion one can determine the equilibrium price of LMCA. In line 3, initially, the equilibrium price $p$ and supply $s^R$ is initialized to 0, and the demand $d^R$ at price $p = 0$ is set to $\infty$. The *while* loop in line 4–14 takes care about determining the demand and supply of each agent and terminates once the total demand becomes equal to total supply. For each iteration of *while* loop the price $p$ is incremented by $\epsilon$. (some small constant value)Line 6-8 calculates the demand of each task requester present in RMCA at price $p$. $r^i_j$ captures the set of task requesters present in RMCA. Line 9-11 calculates the supply of each IoT device present in RMCA at price $p$. $I^i_R$ captures the set of IoT devices present in RMCA. Line 12 and 13 holds the total demand and total supply of agents in $d^R$ and $s^R$ respectively. In line 15, the equilibrium price
\( p \) is held in \( p_R \). Line 16 returns the list of IoT devices and list of task requesters in LMCA, and equilibrium price \( p_R \) of RMCA.

### 4.3.2 Demand and Supply Calculation

In this section, the demand and supply of task requesters and IoT devices respectively are determined in LMCA by using the equilibrium price of RMCA (determined using Algorithm 3). In line 1 – 7, for each task requester \( r^i_j \in r^i_L \), the demand is calculated and the task requesters whose demands are positive at price \( p_R \) is held in \( \tilde{r}^i_L \). In line 5, the total demand in LMCA is calculated and stored in \( d^L \). In line 8 – 14, for each IoT device \( I^i_k \in I^i_L \), the supply is calculated and the IoT devices whose supply are positive at price \( p_R \) is held in \( \tilde{I}^i_L \). In line 12, the total supply in LMCA is calculated and stored in \( s^L \). Line 15 returns total supply and total demand in LMCA at \( p_R \).

**ALGORITHM 4:** Demand and supply calculation \((I^i_L, r^i_L, p_R)\)

Output: \( d^L \leftarrow 0, s^L \leftarrow 0 \)

/* Demand of task requesters in LMCA at price \( p_R \) */

1. *foreach* \( r^i_j \in r^i_L \) *do*

   2. \( d^i_j(p_R) = \arg \max_{f \in [0, Q^i_j]} u^i_j(f, p_R) \) // Calculating the demand of \( r^i_j \) at equilibrium price \( p_R \).

   3. *if* \( d^i_j(p_R) > 0 \) *then*

      4. \( \tilde{r}^i_L \leftarrow \tilde{r}^i_L \cup \{r^i_j\} \) // Each time \( \tilde{r}^i_L \) holds the task requester \( r^i_j \) if criteria in line 3 is satisfied.

   5. \( d^L = d^L + d^i_j(f, p_R) \) // Total demand in LMCA is calculated and held in \( d^L \).

6. *end*

/* Supply of IoT devices in LMCA at price \( p_R \) */

8. *foreach* \( I^i_k \in I^i_L \) *do*

   9. \( s^i_k(p_R) = \arg \max_{f \in [0, Q^i_k]} z^i_k(f, p_R) \) // Calculating the supply of \( I^i_k \) at equilibrium price \( p_R \).

   10. *if* \( s^i_k(p_R) > 0 \) *then*

      11. \( \tilde{I}^i_L \leftarrow \tilde{I}^i_L \cup \{I^i_k\} \) // Each time \( \tilde{I}^i_L \) holds the task requester \( I^i_k \) if criteria in line 10 is satisfied.

   12. \( s^L = s^L + s^i_k(f, p_R) \) // Total supply in LMCA is calculated and held in \( d^L \).

13. *end*

14. *end*

15. *return* \( d^L, s^L \) // Returns, total demand and supply from LMCA.

### 4.3.3 Winner Determination and Payment

In this, the winners and their payment are determined in LMCA. Similarly, we can determine the set of winners and their payment in RMCA by doing notational modifications in Algorithm 5. On determining the demand and supply of the task requesters and task executors respectively in LMCA based on \( p_R \), the three cases may arise: (1) \( d_L = s_L \), (2) \( d_L > s_L \), and (3) \( d_L < s_L \). In line 1-12, the case with \( d_L = s_L \) is tackled and the set of winning task requesters \( r^{w(i)} \), winning IoT devices \( I^{w(i)} \), and their payment are determined.
Algorithm 5: Winner determination and payment ($d^L$, $s^L$, $I^L_i$, $r^L_i$, $p_R$)

1. if $d^L = s^L$ then
   2. foreach $r^i_j \in r^L_i$ do
      3.   foreach $r^i_{j,l} \in r^i_j$ do
         4.      $r^{w(i)} \leftarrow r^{w(i)} \cup \{r^i_{j,l}\}$  \(// r^{w(i)} \) holds the winning virtual task requesters (v.r's).
      5.   end
   6. end
   7. foreach $I^i_k \in I^L_i$ do
      8.   foreach $I^i_{k,v} \in I^i_k$ do
         9.      $I^{w(i)} \leftarrow I^{w(i)} \cup \{I^i_{k,v}\}$  \(// I^{w(i)} \) holds the winning virtual IoT devices (v.I's).
      10. end
   11. end
   12. else if $d^L > s^L$ then
      13.   $\tilde{r}_L \leftarrow \text{Sort} \ (r^L_i)$ \(// \text{Sort v.r's in decreasing order of marginal valuation.}\)
      14.   while $|r^{w(i)}| \neq s^L$ do
      15.      $r^{*} \leftarrow \text{Pick} \ (\tilde{r}_L)$ \(// \text{Picks v.r's from sorted ordering of virtual task requesters.}\)
      16.   foreach $r^i_j \in r^{*}$ do
      17.         while $\nu^i_j < p_R \text{ and } |r^{w(i)}| \neq s^L$ do
      18.            $r^{w(i)} \leftarrow r^{w(i)} \cup \{r^i_{j,l}\}$  \(// r^{w(i)} \) holds the winning v.r's in $i^{th}$ category.
      19.         end
      20.      end
      21.   end
      22.   foreach $I^i_{k,v} \in I^L_i$ do
      23.      $I^{w(i)} \leftarrow I^{w(i)} \cup \{I^i_{k,v}\}$  \(// I^{w(i)} \) holds the winning v.I's in $i^{th}$ category.
      24.   end
  25. else
      26.   $\tilde{I}_L \leftarrow \text{Sort} \ (I^L_i)$ \(// \text{Sort v.I's in increasing order of marginal valuation.}\)
      27.   while $|I^{w(i)}| \neq d^L$ do
      28.      $I^{*} \leftarrow \text{Pick} \ (\tilde{I}_L)$ \(// \text{Picks v.I's from sorted ordering of virtual IoT devices.}\)
      29.   foreach $I^i_{k,v} \in I^{*}$ do
      30.         while $\nu^i_{k,v} > p_R \text{ and } |I^{w(i)}| \neq d^L$ do
      31.            $I^{w(i)} \leftarrow I^{w(i)} \cup \{I^i_{k,v}\}$  \(// I^{w(i)} \) holds the winning v.I's in $i^{th}$ category.
      32.         end
      33.      end
      34.   end
      35.   foreach $r^i_{j,l} \in r^L_i$ do
      36.      $r^{w(i)} \leftarrow r^{w(i)} \cup \{r^i_{j,l}\}$  \(// r^{w(i)} \) holds the winning v.r's in $i^{th}$ category.
      37.   end
  38. return $r^{w(i)}$, $I^{w(i)}$, $p_R$
Using line 3-5 the set of winning virtual task requesters are placed into \( r^{w(i)} \). Line 7-11 determines the set of winning virtual IoT devices and are placed into \( I^{w(i)} \). In line 13-25, the case with \( d_L > s_L \) is considered. In line 14 the virtual task requesters are sorted in decreasing order based on their valuation. The \( \text{while} \) loop in line 15-22 terminates when the set of selected virtual task requesters becomes equal to the total supply in \( w_i \) category. Using line 16, each time a task requester is picked up from the sorted ordering, sequentially. Now for each of the virtual task requesters for the picked up task requester the stopping condition in line 18 is checked. If it is satisfied then the virtual task requester is placed in the winning set otherwise not. The for loop in line 23-25 iterates through all the virtual IoT devices in LMCA and place it into \( I^{w(i)} \). Further, the case with \( d_L < s_L \) is considered in line 27-40. In line 28 the virtual IoT devices are sorted in increasing order based on their valuation. The \( \text{while} \) loop in line 29-36 terminates when the set of selected virtual IoT devices becomes equal to the total demand in \( w_i \) category. Using line 30, each time an IoT device is picked up from the sorted ordering, sequentially. Now for each of the virtual IoT devices for the picked up IoT device the stopping condition in line 32 is checked. If it is satisfied then the virtual IoT device is placed in the winning set otherwise not. The for loop in line 37-39 iterates through all the virtual task requesters in LMCA and place it into \( r^{w(i)} \). Finally in line 41 the winning task requesters, winning IoT devices, and price of the agents in \( i^{th} \) category are returned.

**Example 3.** Let us understand the **allocation and pricing rule** subroutine of QUAD with the help of an example. The category of the task requesters and IoT devices is considered as \( w_2 \). The \( \epsilon \) value (increase in price \( p \) in each iteration) is taken as 3. For the running example it is assumed that the agents are already distributed into two different mobile crowdsourcing arenas \( i.e. \) LMCA and RMCA. Let us consider each of the MCS arenas one-by-one.

- **LMCA:** We have 3 task requesters and 3 IoT devices as shown in Figure 3a. For the given set-up first goal is to determine the equilibrium price \( p \). So, let us say the initial price \( p \) is set to 0. At this price, all the task requesters wants that their tasks get executed \( i.e. d^L \) is 6) and on the other hand no IoT device wants to execute the tasks \( i.e. s^L \) is 0).

![Figure 3: Detailed Illustration of Allocation and Pricing Rule in LMCA.](image-url)
Line 4 of Algorithm 3 is satisfied and $p$ value is increased by 3. Now, at price $p = 3$ the demand from the task requesters i.e. $d^L$ is 6 and the supply of the IoT devices i.e. $s^L$ is 0. Again the condition in line 4 of the Algorithm 3 is satisfied, so $p$ value is increased by 3 again and now $p$ is 6. At $p = 6$, the $d^L$ of task requesters is 6 and $s^L$ of IoT devices is 1, so demand is not equal to supply still. Again, the price value is increased by 3 and now at $p = 9$, the $d^L$ of task requesters is 5 and $s^L$ of the IoT devices is 1. Similarly, we keep on increasing the price and at $p = 15$ the $s^L$ becomes equal to $d^L$. So the equilibrium price in the LMCA is 15. In Figure 3b, at $p = 15$, the set of selected task requesters and IoT devices are shown in square box.

- **RMCA:** In RMCA, we have 2 task requesters and 3 IoT devices as shown in Figure 4a. Let us first determine the equilibrium price $p$ in RMCA. So, let us say the initial price $p$ is set to 0. At this price, all the task requesters wants that their tasks get executed (i.e. $d^R$ is 4) and on the other hand no IoT device wants to execute the tasks (i.e. $s^R$ is 0). Line 4 of Algorithm 3 is satisfied and $p$ value is increased by 3. Now, at price $p = 3$ the demand from the task requesters i.e. $d^R$ is 4 and the supply of the IoT devices i.e. $s^R$ is 0. Again the condition in line 4 of the Algorithm 3 is satisfied, so $p$ value is increased by 3 again and now $p$ is 6. At $p = 6$, the $d^R$ of task requesters is 4 and $s^R$ of IoT devices is 1, so demand is not equal to supply still. Again, the price value is increased by 3 and now at $p = 9$, the $d^R$ of task requesters is 4 and $s^R$ of the IoT devices is 2. Similarly, we keep on increasing the price and at $p = 12$ the $s^R = 3$ becomes equal to $d^R = 3$. So the equilibrium price in the RMCA is 12. In Figure 4b, at $p = 12$, the set of selected task requesters and IoT devices are shown in square box.

From the above discussion, we get the equilibrium price in LMCA as $p_L = 15$ and the equilibrium price in RMCA as $p_R = 12$. Once the equilibrium price in each of the markets is determined after that Algorithm 4 will be applied to our running example. Following Algorithm 4, in RMCA, at $p_L = 15$, $d^R$ is 3 and $s^R$ is 5. On the other hand, in LMCA, at $p_R = 12$, $d^L$ is 5 and $s^L$ is 3. Now applying Algorithm 5 to LMCA and RMCA one-by-one. For LMCA, line 13-26 of Algorithm 5 will

![Figure 4: Detailed Illustration of Allocation and Pricing Rule in RMCA.](image-url)
be activated, as \( d^L > s^L \). Following line 14, the virtual task requesters are sorted in decreasing order of their valuation. After that using line 15-22 the winning virtual task requesters are decided and following line 23-25 winning virtual IoT devices are decided as shown in Figure 3b. The price at which the trading took place is \( p_R = 12 \). For RMCA, line 27-40 of Algorithm 5 will be activated, as \( d^R < s^R \). Following line 28, the virtual task requesters are sorted in increasing order of their valuation. After that using line 29-36 the winning virtual IoT devices are decided and following line 37-39 winning virtual task requesters are decided as shown in Figure 4c. The price at which the trading took place is \( p_L = 15 \).

5 Analysis of QUAD

In this section, the analysis of QUAD is carried out. In Lemma 5.1 it is shown that the QUAD runs in polynomial time. In Lemma 5.2 it is proved that QUAD is correct. In order to prove that QUAD is correct, it is shown that the agents pay some amount to the platform. Lemma 5.3 proves that QUAD is weakly budget balanced. The reason behind proving QUAD as WBB is to show that the agents pay some amount to the platform.

Lemma 5.1. QUAD is computationally efficient.

Proof. The running time of QUAD is the sum of the running time of Algorithm 1, Algorithm 2, Algorithm 3, Algorithm 4, and Algorithm 5. As Algorithm 1 will iterate for \( k \) times, so let us first determine the time taken by each of the subroutines of Algorithm 1. After that the running time for \( k \) different categories will be decided.

In Algorithm 2 lines 1 and 2 will take \( O(1) \) time. The while loop in line 3-16 will terminate only when all the IoT devices are ranked. So, say the while loop iterates for \( \lfloor \frac{n}{3} \rfloor \). For each iteration of while loop, line 4 and line 5 will take \( O(\gamma) \) and \( O(\beta) \) respectively. Line 6 is bounded above by \( O(\gamma \beta) \). Line 7-13 will take \( O(\gamma \beta) \). In line 14, the IoT device with maximum point is selected and will take \( O(\beta) \) time. Therefore, line 3-16 is bounded above by \( \frac{n}{3} \cdot O(\gamma \beta) = n_1 \gamma \). So, the time taken by Algorithm 2 is \( O(n_1 \gamma) \).

In Algorithm 3, line 1 and 2 will take \( O(m_i + n_i) \) time. Line 3 will take constant time. The execution of Lines 5-13 depends on the condition in line 4. For each iteration of while loop, line 5 will take \( O(1) \). Line 6-8 is bounded above by \( O(m_i Q_j^i) \) and line 9-11 is bounded above by \( O(n_i Q_j^i) \). In the worst case, line 12 will sum the demands of all the \( m_i \) task requesters at equilibrium price \( p \) and will take \( O(m_i) \) time. Similarly, line 13 will take \( O(n_i) \) time. So, for each iteration of while loop, line 5-13 is bounded above by \( O(1) + O(n_i Q_j^i) + O(m_i Q_j^i) + O(m_i) = O(n_i^2) \), if \( Q_j^i, Q_k^i \), and \( m_i \) is a function of \( n_i \). If the number of iteration of while loop in line 4-14 is a function of \( n_i \), then line 4-14 is bounded above by \( O(n_i^2) \). Line 15 does assignment so it will take constant time. The return statement in line 16 will take \( O(1) \). So, Algorithm 3 takes \( O(n_i^2) \).

In Algorithm 4 for each iteration of for loop in line 1-7, line 2 will take \( O(Q_j^i) \). Line 3-6 will take \( O(1) \). Now, if the for loop iterates for the number of times that is a function of \( m_i \), then line 1-7 of Algorithm 4 is bounded above by \( O(m_i^2) \), if \( Q_j^i \) is a function of \( m_i \). Similar argument can
be given for line 8-14 and is bounded above by $O(n^2)$. The return statement in line 15 will take constant time. So, Algorithm 4 takes $O(m^2) + O(n^2)$ time.

Now, let us determine the time taken by Algorithm 5 for each iteration of Algorithm 1. From the construction of Algorithm 5 it can be seen that at a time either line 1 – 11 will be executed or line 12 – 24 will be executed or line 26 – 38 will be executed. Let us say line 1 – 12 gets executed. Line 1 will take constant time. Line 2 – 6 takes $O(m^2)$ in worst case. Similarly, line 7 – 11 can take $O(n^2)$ in worst case. So line 1 – 12 of Algorithm 5 is bounded above by $O(n^2) + O(m^2)$. Line 12 – 24 executes when the demand of the task requesters is more than the supply from the IoT devices. In line 13 sorting takes place and will take $O(m_i \log m_i)$, where $\log m_i$ stands for $\log_{10} m_i$. Line 14 – 21 will take $O(m^3)$ in worst case. Line 22-24 will take $O(n_i)$. Similarly, line 25 – 38 executes when the demand of the task requesters is less than the supply from the IoT devices. In line 26 sorting takes place and will take $O(n_i \log n_i)$. Line 27 – 34 will take $O(n^3_i)$ in worst case. Line 35-37 will take $O(m_i)$. So the running time of Algorithm 5 is $O(m^2_i) + O(m_i \log m_i) + O(n^3_i) + O(n_i \log n_i) + O(n^3_i) = O(m^3_i) + O(n^3_i) + O(m^3_i) = O(n^3_i) + O(m^3_i)$. So, for $k$ different categories the QUAD will be bounded above by $km^3_1 + km^3_i = k(n^3_i + m^3_i)$ i.e. $O(n^3_i + m^3_i)$, if $k$ is constant. Hence, QUAD is computationally efficient.

Lemma 5.2. QUAD works correctly.

Proof. We prove the correctness of QUAD using loop invariant technique Cormen et al. (2009). In order to prove that QUAD is correct, it is to be shown that each of the algorithms acting as subroutine are also correct. Below the correctness of each of the subroutines is discussed one-by-one:

Proof of correctness of Algorithm 2 (IoT-QDBC): In order to prove that IoT-QDBC is correct, the following loop invariant is considered:

**Loop invariant:** In each iteration of while loop of lines 3-16, a quality IoT device in $i^{th}$ category is added into the output array $Q^i$. 

- **Initialization:** We can start by showing that loop invariant hold before the first iteration of while loop, when $Q^i = \phi$. The output array $Q^i$ have no quality IoT devices before the first iteration. So, loop invariant holds.

- **Maintenance:** For the loop invariant to be true, it is to be shown that before any $k^{th}$ iteration of the while loop and after $k^{th}$ iteration of the while loop the loop invariant holds. Before $k^{th}$ iteration, i.e. till $(k-1)^{th}$ iteration there will be $(k-1)$ quality IoT devices in an array $Q^i$. After $k^{th}$ iteration, the number of quality IoT devices will be $\sum_{i=1}^{k} 1 = k$ in $Q^i$. So, loop invariant holds.

- **Termination:** From the construction of IoT-QDBC, it is clear that the while loop will terminate only when there exists no IoT devices whose executed tasks are to be ranked. It means that once while loop terminates $Q^i$ contains all the quality IoT devices. 

Hence, IoT-QDBC is correct.

Proof of correctness of Algorithm 3 (Splitting and Equilibrium Price Determination): In order to prove that Algorithm 3 is correct, we use the following loop invariant:
Loop invariant: At the start of while loop in lines 4-14, the price $p$ is increased by $\epsilon$. So, with the increase in price $p$ the demand $d_R^R$ of the task requesters will be decreasing or remains same and supply $s_R^R$ of the IoT devices will be increasing or remains same.

- **Initialization:** Before the 1st iteration of while loop, the price $p$ is zero. In this case, the demand $d_R^R$ of the task requesters will be very high (say $\infty$) and supply $s_R^R$ of the IoT devices will be very low (say 0). So, it trivially satisfies the loop invariant.

- **Maintenance:** In the $k^{th}$ iteration of while loop, the price $p$ will be some non-zero value. Depending on the price value at $k^{th}$ iteration, the demand of task requester may remain same or may decrease and the supply of IoT devices may remain same or may increase. After $k^{th}$ iteration the price $p$ will be $\epsilon$ greater than it was before $k^{th}$ iteration. Now, with the increase in price the demand may decrease or remains same and supply may increase or remains same. So, the loop invariant may hold.

- **Termination:** From the construction of Algorithm 3, at termination, the demand of task requesters and supply by the IoT devices becomes equal at the equilibrium price. So, the algorithm provide the equilibrium price on termination.

Hence, Algorithm 3 is correct.

**Proof of correctness of Algorithm 4 (Demand and Supply Calculation):**

Loop invariant: In each iteration of for loops in line 1-7 and in line 8-14 the $d^L$ and $s^L$ respectively will be either zero or positive.

- **Initialization:** Before the 1st iteration of for loops in line 1-7 and line 8-14, the $d^L$ and $s^L$ values respectively are 0. So, it trivially satisfies the loop invariant.

- **Maintenance:** Before any $k^{th}$ iteration of for loops, in line 1-7 and line 8-14, the $d^L$ and $s^L$ values respectively will be either zero or some positive value. After the $k^{th}$ iteration of for loops, in line 1-7 and line 8-14 also, the $d^L$ and $s^L$ values respectively will be either zero or some positive value. So, the loop invariant may hold.

- **Termination:** From the construction of Algorithm 3, at termination the demand of task requesters and supply by the IoT devices will be non-negative. So, on termination also the loop invariant holds for Algorithm 4.

Hence, Algorithm 4 is correct.

**Proof of correctness of Algorithm 5 (Winner Determination and Payment):** The case with $d^L = s^L$ is trivial. In order to prove correctness of Algorithm 5, the non-trivial cases are considered i.e. $d^L > s^L$ or $d^L < s^L$. Let us $d^L > s^L$ case, similarly $d^L < s^L$ can be handled.

Loop invariant: In each iteration of while loop in line 14-21 the task requesters are added in the winning set $r^{w(i)}$. 

22
Initialization: We can start by showing that loop invariant holds prior to first iteration of 
while loop in line 14-21, when \( r^{w(i)} = \phi \). The winners list \( r^{w(i)} \) have no task requesters. So, loop invariant holds.

Maintenance: For the loop invariant to be true, it is to be shown that before any \( k^{th} \) iteration of the while loop in line 14-21 and after any \( k^{th} \) iteration of the while loop in line 14-21 the loop invariant holds. Before \( k^{th} \) iteration, i.e. till \((k - 1)^{th}\) iteration there will be \((k - 1)\) winning task requesters in \( r^{w(i)} \). After \( k^{th} \) iteration, one more task requester will be added to \( r^{w(i)} \) leads to \( k \) task requesters. So, loop invariant holds.

Termination: From the construction of Algorithm 5, it is clear that the while loop in line 14-21 will terminate only when \( r^{w(i)} \) becomes equal to \( s^L \). It means that once while loop terminates \( r^{w(i)} \) contains all the possible winning task requesters at equilibrium price. On termination the loop invariant holds for Algorithm 5.

Hence, Algorithm 5 is correct.

As Algorithm 2, Algorithm 3, Algorithm 4, and Algorithm 5 are correct for any category \( w_i \) and so as the QUAD. So, QUAD will be correct for all the \( k \) categories in the system. Hence QUAD is correct.

Lemma 5.3. QUAD is prior-free.

Proof. From the construction of QUAD it can be seen that it is not using any statistical information on valuations of the agents. So, QUAD is prior-free.

Observation 1. QUAD mechanism consists of two components: IoT-QDBC mechanism, and allocation and pricing rule. Here, IoT-QDBC mechanism is independent of the bid values of the IoT devices and will not influence the truthfulness of QUAD.

Lemma 5.4. QUAD is truthful or DSIC.

Proof. Fix category \( w_i \) and task requester \( r^j_i \). To prove that QUAD is truthful, the two cases are considered: (1) early exit, and (2) late exit. In the first case, it may happen that task requester \( r^j_i \) bids less than his true valuation and can come out of the market early, even if he can participate in further iterations. In late exit, it may happen that the task requester \( r^j_i \) can bid more than his true valuation and can stay for longer time in the market. The proof is carried out considering the task requesters. However, in the similar fashion the truthfulness of the QUAD can be proved for the IoT devices. Let us discuss the two cases one-by-one.

- Early exit: Let us suppose that a task requester \( r^j_i \) misreports his bid value for buying \( t \) units of completed tasks such that \( \hat{\nu}^i_j(t) < \nu^i_j(t) \). Now, in this situation the two scenarios can happen. One scenario could be that the \( t^{th} \) iteration is the last iteration of the mechanism i.e. demand and supply becomes equal and the mechanism terminates. In this case, the task requester \( r^j_i \) wins and his utility will be same as his utility when he would have reported truthfully i.e. \( \hat{u}^i_j(t, p) = \nu^i_j(t) - p \cdot t = u^i_j(t, p) \). Another scenario could be the mechanism can go further for \((i + 1)^{th}\) iteration and many more. Now, in this scenario as the task requester has misreported (lowered his bid value) and exited from the market, so he is a loser and his utility will be 0.
Late exit: Let us suppose that a task requester \( r^j_t \) misreports his bid value for buying \( t \) units of completed tasks such that \( \hat{\nu}^j_t(t) > \nu^j_t(t) \). Now, in this situation the two scenarios can happen. One scenario could be that the \( i^{th} \) iteration is the last iteration of the mechanism and after that the mechanism terminates. In this case, the task requester \( r^j_t \) wins and his utility will be same as his utility when he would have reported truthfully i.e. \( \hat{u}^j_t(t, p) = \nu^j_t(t) - p \cdot t = u^j_t(t, p) \). Another scenario could be the mechanism can go further for \( (i + 1)^{th} \) iteration and many more. Now, in this scenario again the two cases can happen. The task requester \( r^j_t \) can participate in further iterations by misreporting his bid value i.e. \( \hat{\nu}^j_t(t) > \nu^j_t(t) \) and he wins. In this case, his utility will be \( \hat{u}^j_t(t, p) = \nu^j_t(t) - p \cdot t < u^j_t(t, p) \). Another case could be with the increase in his bid value the task requester \( r^j_t \) made his presence for some additional round but dropped from the market in the later rounds. In this case he loses and his utility will be 0.

So, from above discussion it can be concluded that the task requester cannot gain by misreporting his bid value. In the similar line it can be shown that the IoT devices cannot gain by misreporting its bid value. Hence, QUAD is truthful for \( w_i \) category and so is for all the \( k \) categories.

Lemma 5.5. QUAD satisfies individual rationality.

Proof. Fix a category \( w_i \). In QUAD each time a price \( p \) is set and then it is asked from the agents that whether you are ready to trade at price \( p \) or not? Now, if for any task requester \( r^j_t \) the bid value is more than or equal to the current price then his answer will be YES, otherwise his answer will be NO. It is obvious that the task requester \( r^j_t \) will reply YES only when \( \nu^j_t(t) > p \). In such case \( r^j_t \) has to pay less than his valuation and achieves positive utility. Else if \( \nu^j_t(t) < p \) then \( r^j_t \) will drop from the crowdsensing market and in that case his utility will be zero. So, the task requester will be having a non negative utility. Similar argument can be given for the IoT devices. As the utility of the participating agents are non negative, so from the definition of individual rationality (see Definition 3) it can be said that QUAD is individually rational for \( w_i \) category. Hence, QUAD is individually rational for all the \( k \) categories.

Lemma 5.6. The allocation resulted by QUAD is weakly budget balanced.

Proof. Fix a category \( w_i \). Let us say some virtual task requesters of \( r^j_t \) is present in the winning set. Now, from the construction of QUAD the trading fee paid by any task requester \( r^j_t \) is the utility that would have been achieved by the virtual task requesters that are not in the winning set because of the presence of virtual task requesters of task requester \( r^j_t \) in the winning set. In Lemma 5.5 it is already shown that the agents in the winning set is having a non-zero utility, so it can be said that the trading fees paid by any task requester \( r^j_t \) will be a non-zero value. So, the platform will always receive non zero fees from the task requesters. Similar argument can be given for the IoT devices. QUAD is weakly budget balanced for \( w_i \) category. Hence, QUAD is weakly budget balance for all the \( k \) different categories.

5.1 Probabilistic Analysis of QUAD

In this section, the probabilistic analysis of QUAD is carried out. Lemma 5.7 provide an estimate that, on an average how many tasks are executed for any task requester? It is shown that for any task requester \( r^j_t \) the expected number of his endowed tasks will be executed by the quality IoT
devices will be \( \Lambda_i \cdot \left( \frac{1}{\lg \Lambda_i} \right) \). Here, \( \Lambda_i \) is the total number of tasks carried by any task requester \( r^i_j \).

Further in Lemma 5.8, it is estimated that at least \( 0.9 \lg \Lambda_i \) tasks of any task requester \( r^i_j \) will be considered out of \( \Lambda_i \) tasks is at most \( \frac{10}{9 \lg \Lambda_i} \).

**Lemma 5.7.** In category \( w_i \), for any task requester \( r^i_j \) the expected number of his endowed tasks executed by the quality IoT devices out of \( \Lambda_i \) is given as:

\[
E[Z] = \Lambda_i \cdot \left( \frac{1}{\lg \Lambda_i} \right)
\]

where, \( \Lambda_i \) is the number of tasks held by task requester \( r^i_j \). \( Z \) is the indicator random variable measuring the number of tasks executed by the quality IoT devices out of \( \Lambda_i \) tasks.

**Proof.** Fix category \( w_i \) and the task requester \( r^i_j \). In this lemma, we are trying to show that, in expectation, how many tasks of the task requester \( r^i_j \) gets executed by the quality IoT devices out of \( \Lambda_i \)? It is represented as \( E[Z] \). Before moving forward, it is important to decide the sample space associated with the task requester \( r^i_j \) and is given as:

\[
S = \{ i^{th} \text{ task of } r^i_j \text{ gets executed, } i^{th} \text{ task of } r^i_j \text{ does not get executed} \}
\]

Now, let us determine the probability that \( i^{th} \) task of \( r^i_j \) gets executed and is given as \( q \). So, the probability that \( i^{th} \) task of \( r^i_j \) do not get executed is given as \( 1 - q \). Let us define an indicator random variable \( Z_i \), associated with \( i^{th} \) task of \( r^i_j \) gets executed, which is the event \( R \). The variable \( Z_i \) captures that the \( i^{th} \) task of \( r^i_j \) gets executed or not. If it gets executed then \( Z_i \) is 1 otherwise 0. We write

\[
Z_i = I\{R\} = \begin{cases} 
1, & \text{if } R \text{ happens,} \\
0, & \text{otherwise} 
\end{cases}
\]

The expected value of the random variable \( Z_i \) gives us the expected value that the \( i^{th} \) task of \( r^i_j \) gets executed. Taking expectation both side of equation 5, we get

\[
E[Z_i] = E[I\{R\}] = 1 \cdot q + 0 \cdot (1 - q) = q
\]

Now, let \( Z \) be the random variable denoting the total number of tasks executed by the quality IoT devices out of \( \Lambda_i \) tasks. It is formulated as

\[
Z = \sum_{i=1}^{\Lambda_i} Z_i
\]

As our aim is to compute the expected number of tasks executed by the quality IoT devices out of \( \Lambda_i \) tasks and can be obtained by taking expectation both side of the equation 6. So, we get

\[
E[Z] = E\left[ \sum_{i=1}^{\Lambda_i} Z_i \right]
\]
By linearity of expectation and then substituting the value of \( E[\mathcal{Z}_i] \), we get

\[
E[\mathcal{Z}] = \sum_{i=1}^{\Lambda_i} E[\mathcal{Z}_i] = \sum_{i=1}^{\Lambda_i} q = \Lambda_i \cdot q \tag{8}
\]

Now, if the probability that \( i^{th} \) task of \( r_j^i \) task requester will get executed is taken as \( \frac{1}{\lg \Lambda_i} \) then the value \( E[\mathcal{Z}] \) boils down to \( \Lambda_i \cdot \left( \frac{1}{\lg \Lambda_i} \right) \). It can be written as:

\[
E[\mathcal{Z}] = \Lambda_i \cdot \left( \frac{1}{\lg \Lambda_i} \right) \tag{9}
\]

Hence proved.

**Observation 2.** If we consider the value of \( \Lambda_i \) as 100 then \( E[\mathcal{Z}] = \frac{\Lambda_i}{\lg \Lambda_i} = \frac{100}{\lg 100} = \frac{100}{2} = 50 \). It means that, for task requester \( r_j^i \) on an average 50 of his tasks will be executed by the IoT devices out of his 100 sensing tasks.

**Lemma 5.8.** The probability that at least 0.9\( \Lambda_i \) tasks of any task requester \( r_j^i \) will be executed out of \( \Lambda_i \) tasks by the quality IoT devices is less than or equals to \( \left( \frac{10}{9 \lg \Lambda_i} \right) \). More formally, we can write

\[
P\{ \mathcal{Z} \geq 0.9 \Lambda_i \} \leq \left( \frac{10}{9 \lg \Lambda_i} \right) \tag{10}
\]

**Proof.** Fix a category \( w_i \) and a task requester \( r_j^i \). Here, we are trying to prove that at least 0.9\( \Lambda_i \) tasks of task requester \( r_j^i \) will be executed out of \( \Lambda_i \) tasks is bounded above by \( \left( \frac{10}{9 \lg \Lambda_i} \right) \) quantity. For this purpose we will take the help of the random variable \( \mathcal{Z} \) defined in Lemma 5.7. Following Lemma 5.7, we can write

\[
I = \begin{cases} 
1, & \text{if } \mathcal{Z} \geq 0.9 \Lambda_i, \\
0, & \text{otherwise} 
\end{cases} \tag{10}
\]

Now, considering the case we are interested in from the above case representation form \( \mathcal{Z} \geq 0.9 \Lambda_i \). We can rewrite it as \( \frac{Z}{0.9 \Lambda_i} \geq 1 \). From equation 10, \( I \) is same as 1, so we get

\[
\frac{Z}{0.9 \Lambda_i} \geq I \tag{11}
\]

Taking expectation both side in equation 11, we get

\[
E\left[ \frac{Z}{0.9 \Lambda_i} \right] \geq E[I] \tag{12}
\]

or

\[
E[I] \leq E\left[ \frac{Z}{0.9 \Lambda_i} \right] = \frac{1}{0.9 \Lambda_i} E[Z] \tag{12}
\]
From the definition of expectation, equation 14 can be rewritten as

\[ P\{ Z \geq 0.9\Lambda_i \} \cdot 1 \leq \left( \frac{10}{9\Lambda_i} \right) \cdot E[Z] \]  

Substituting the value of \( E[Z] \) in equation 14 from Lemma 5.7, we get

\[ P\{ Z \geq 0.9\Lambda_i \} \leq \left( \frac{10}{9\Lambda_i} \right) \cdot \left( \frac{\Lambda_i}{\lg \Lambda_i} \right) \]

\[ = \frac{10}{9\lg \Lambda_i} \]

Hence proved. \( \Box \)

Observation 3. If we consider the value of \( \Lambda_i \) as 1000, then \( P\{ Z \geq 900 \} \leq \frac{10}{9\lg 1000} = \frac{10}{27} = 0.37. \)

So, the probability that at least 900 tasks of \( r_i^j \) will be executed out of 1000 will be less than or equal to 0.37, which is quite low.

6 Experiments and Results

To complement our theoretical analysis, the QUAD is compared with the benchmark mechanisms namely PPM Roughgarden (2014) and McAfee double auction (DA) McAfee (1992); Bredin and Parkes (2005). In order to measure the performance of QUAD in terms of truthfulness and budget balanced properties, it is compared with PPM on the following evaluation metrics: (1) Utility of agents, and (2) Utility of platform. The evaluation metric mentioned in point 1 above will be helpful in showing that QUAD is not vulnerable to manipulation whereas in case of PPM the participating agents gain by misreporting their true values. On the other hand the metric mentioned in point 2 will help us to show that QUAD is weakly budget balanced. It means that platform will have non-zero utility. Another direction of performance measurement of QUAD is in terms of satisfaction level of the IoT devices. Higher the incentives given to the IoT devices more satisfied and motivated the IoT devices will be in MCS market. In order to measure the performance of QUAD on the basis of satisfaction level, it is compared with McAfee DA on the ground of total charge. Higher the value of charges paid to the IoT devices, higher will be the satisfaction level of the IoT devices. The unit of bid values of the task requesters and IoT devices is taken as $\text{.}$ The experiments are carried out in Java.

6.1 Simulation Set-up

For the simulation purpose, the task requesters and IoT devices are considered from 5 different categories. In each category, the number of task requesters, the number of IoT devices, and the bid value ranges of the task requesters and the IoT devices are considered as depicted in Table 3 below. In Table 3 the symbol \( \mu \) represents the mean and symbol \( \sigma \) represents the standard deviation of the bid values generated using normal distribution. In the experiments, the valuation of the agents are distributed among their respective virtual agents in such a way that it follows decreasing marginal
return. In our case, the experiment is executed for all the 5 categories for 100 times and the required values are plotted by taking the average over these 100 times. In order to strengthen our claim the simulation is carried out for two different probability distributions, namely random distribution (RanD) and normal distribution (NanD).

Table 3: Data set utilized for Simulation

| Parameters | Values          | Description                                               |
|------------|-----------------|-----------------------------------------------------------|
| $m_i$      | {5, 10, 15, 20, 25, 30} | Number of task requesters in each category.               |
| $n_i$      | {15, 30, 45, 60, 75, 90} | Number of IoT devices in each category.                    |
| $\nu^r$   | [8, 30]         | Bid value range of task requesters for RanD.               |
| $\nu^d$   | [5, 25]         | Bid value range of IoT devices for RanD.                   |
| $\nu^r$ (for NanD) | $[\mu = 15, \sigma = 4]$ | Bid value determination of task requesters for NanD.      |
| $\nu^d$ (for NanD) | $[\mu = 16, \sigma = 5]$ | Bid value determination of IoT devices for NanD.           |

For comparing the QUAD and PPM on the ground of truthfulness, it is considered that 50% of the agents misreport their true bid values in case of PPM and is represented as PPM-D in the simulation results. By misreporting the bid value it is meant that the agents will report the increased bid price so as to stay for longer duration of time in the MCS market.

6.2 Results and Comparison

In this section, the discussion will mainly circumvent around the evaluation metrics depicted above. Figure 5 compares the two mechanisms namely QUAD and PPM on the ground of truthfulness of the agents for RanD and NanD cases. From the construction of QUAD and PPM, the utility of the participating agents in case of QUAD can be more or less than the utility of the agents in case of PPM as depicted in Figure 5a and 5b. However, in case of PPM, when 50% of the agents misreport their true bid values then the utility of the agents is more as compared to the case when all the agents report truthfully. It is natural as PPM is vulnerable to manipulation, so the participating agents gain by misreporting their true bid values.

Figure 5: Comparison of Utility of Agents for RanD and NanD cases
In Figure 5, in some cases it can be seen that the utility of the agents in case of PPM with 50% deviation is even more than the utility of the agents in case of QUAD. On the other hand, as QUAD works on the principle of Vickery auction W.Vickery (1961) and so is not vulnerable to manipulation. It means that the participating agents can maximize their utility by only reporting truthfully in case of QUAD and is seen in Figure 5a and 5b.

Considering the second evaluation metric i.e. utility of platform. In Figure 6a and 6b it can be seen that the utility of platform in case of QUAD could be more or less as compared to the utility of platform in case of PPM. However, for both the mechanisms i.e. QUAD and PPM the utility of the platform will be non-negative irrespective of the type of distribution followed by the bid values of the agents. In such case it can be inferred from Figure 6a and Figure 6b that the two mechanisms is weakly budget balanced (see Definition 5). From the above discussion it can be inferred that the QUAD is truthful and both the mechanism satisfies one of the economic properties called budget balance.

![Figure 6: Comparison of Utility of Platform for RanD and NanD cases](a) Utility of Platform (RanD)  
(b) Utility of Platform (NanD)

![Figure 7: Comparison of Charge received by IoT Devices for RanD and NanD cases](a) Total Charge Received by IoT Devices (RanD)  
(b) Total Charge Received by IoT Devices (NanD)
Figure 7a and 7b compares QUAD and McAfee double auction (DA) on the basis of total charges made to the IoT devices for both RanD and NanD. It can be seen that the total charge of the IoT devices is high in case of QUAD a compared to McAfee DA in both RanD and NanD. As it is obvious because in case of posted price ascending auctions the selling price is generally high as compared to the sealed bid auction and so the IoT devices are paid high. Due to this reason the satisfaction level of winning IoT devices in case QUAD will be more than the satisfaction level of the winning task requesters in case of McAfee DA. So, if it is seen from the perspective of IoT devices then QUAD will be more viable mechanism as compared to McAfee DA.

The simulation results in Figure 8a and 8b shows that in case of QUAD the average number of tasks executed by the quality IoT devices is almost half the total number of available tasks. The x-axis of the graphs shown in Figure 8a and 8b represents the task requesters and y-axis represents the number of tasks. For all the 5 task requesters the number of tasks that got executed is almost half the number of tasks endowed by the respective task requesters irrespective of the distribution of data. The similar results we obtained from the probabilistic analysis carried out in Lemma 5.7.

7 Conclusion and Future Directions

In this paper, for the discussed scenario in IoT-based mobile crowdsensing, a truthful mechanism is proposed that along with satisfying other economic properties such as individual rationality, and budget balance keeps into account the quality of IoT devices. By analysis it is shown that QUAD is computationally efficient, truthful, individual rational, and prior free. Further the probabilistic analysis is carried out to have an estimate on the number of tasks of any task requester got executed by the quality IoT devices among the available ones. The simulation results compares QUAD with the benchmark mechanisms on the ground of truthfulness, budget balance, and satisfaction level of the IoT devices.

In future, one of the challenges could be to design a truthful mechanism for the set-up discussed in this paper for general valuation (not DMR). Another direction could be, considering the set-up with multiple task providers and multiple IoT devices, where the task providers will provide the
heterogeneous tasks. Here, the challenge will be to design a truthful mechanism for the cases when the participating agents have DMR valuation or general valuation.

References

Abraham, A., Dash, S., Rodrigues, J., Acharya, B., and Pani, S. 2021. AI, Edge and IoT-based Smart Agriculture. Intelligent Data-Centric Systems. Elsevier Science, Barcelona, Spain.

Abualsaud, K., Elfouly, T. M., Khattab, T., Yaacoub, E., Ismail, L. S., Ahmed, M. H., and Guizani, M. 2019. A survey on mobile crowd-sensing and its applications in the IoT era. IEEE Access 7, 3855–3881.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. 2019. Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys Tutorials 17, 4, 2347–2376.

Alavi, A. H., Jiao, P., Buttlar, W. G., and Lajnef, N. 2018. Internet of Things-enabled smart cities: State-of-the-art and future trends. Measurement, Elsevier 129, 589–606.

Ang, K. L. M., Seng, J. K. P., and Ngharamike, E. 2022. Towards crowdsourcing internet of things (crowd-iot): Architectures, security and applications. Future Internet, MDPI 14, 2.

Ashton, K. 1999. That ‘internet of things’ thing.

Atzori, L., Iera, A., and Morabito, G. 2010. The internet of things: A survey. Computer Networks, Elsevier, 54, 15, 2787–2805.

Aubry, E., Silverston, T., Lahmadi, A., and Festor, O. 2014. Crowdout: A mobile crowdsourcing service for road safety in digital cities. In 2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS). Budapest, Hungary, 86–91.

Baccour, E., Erbad, A., Mohamed, A., Hamdi, M., and Guizani, M. 2020. Distprivacy: Privacy-aware distributed deep neural networks in iot surveillance systems. In GLOBECOM 2020 - 2020 IEEE Global Communications Conference. Taipei, Taiwan, 1–6.

Blumrosen, L. and Dobzinski, S. 2014. Reallocation mechanisms. In Proceedings of the Fifteenth ACM Conference on Economics and Computation. EC’14. Association for Computing Machinery, New York, NY, USA, 617.

Bredin, J. and Parkes, D. C. 2005. Models for truthful online double auctions. In 21st International Conference on Uncertainty in Artificial Intelligence (UAI). AUAI press, 50–59.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. 2009. Introduction to algorithms. MIT press.

Dasari, V. S., Kantarci, B., Pouryazdan, M., Foschini, L., and Girolami, M. 2020. Game theory in mobile crowdsensing: A comprehensive survey. Sensors 20, 7.
Deshmukh, K., Goldberg, A. V., Hartline, J. D., and Karlin, A. R. 2002. Truthful and competitive double auctions. In *Algorithms — ESA 2002*, R. Möhring and R. Raman, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 361–373.

Enigo, V. F., Kumar, T. V., Vijay, S., and Prabu, K. 2016. Crowdsourcing based online petitioning system for pothole detection using android platform. *Procedia Computer Science* 87, 316–321. Fourth International Conference on Recent Trends in Computer Science & Engineering (ICRTCSE 2016).

Feldman, M. and Gonen, R. 2016. Markets with strategic multi-minded mediators. *CoRR* abs/1603.08717.

Feng, Z., Zhu, Y., Zhang, Q., Ni, L. M., and Vasilakos, A. V. 2014. TRAC: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. Toronto, ON, Canada, 1231–1239.

García, P. A. G., Marin, C., Prieto, J. D., and Nieto, Y. V. 2017. Analysis of security mechanisms based on clusters iot environments. *Int. J. Interact. Multim. Artif. Intell.* 4, 55–60.

Goel, G., Leonardo, S., Mirrokni, V., Nikzad, A., and Paes-Leme, R. 2016. Reservation exchange markets for internet advertising. *LIPIcs* 55, 142:1–142:13.

Gonen, M., Gonen, R., and Pavlov, E. 2007. Generalized trade reduction mechanisms. In *Proceedings of the 8th ACM Conference on Electronic Commerce*. EC ’07. Association for Computing Machinery, New York, NY, USA, 20–29.

Gonen, R. and Egri, O. 2017. Dycom: A dynamic truthful budget balanced double-sided combinatorial market. In *AAMAS*.

Hamrouni, A., Alelyani, T., Ghazzai, H., and Massoud, Y. 2021. Toward collaborative mobile crowdsourcing. *IEEE Internet of Things Magazine* 4, 2, 88–94.

Heuveldop, N. 2017. Ericsson mobility report.

Hirai, H. and Sato, R. 2021. Polyhedral clinching auctions for two-sided markets. *Mathematics of Operations Research* 0, 0, null.

Hou, F., Pei, Y., and Sun, J. 2019. *Mobile Crowd Sensing: Incentive Mechanism Design*. Springer, Cham.

Huang, H., Xin, Y., Sun, Y.-E., and Yang, W. 2017. A truthful double auction mechanism for crowdsensing systems with max-min fairness. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. 1–6.

Index, G. C. 2015. Huawei technologies coo., ltd.,

Khajenasiri, I., Estebbsari, A., Verhelst, M., and Gieelen, G. 2017. A review on internet of things solutions for intelligent energy control in buildings for smart city applications. *Energy Procedia* 111, 770–779. 8th International Conference on Sustainability in Energy and Buildings, SEB-16, 11-13 September 2016, Turin, Italy.
Kong, X., Liu, X., Jedari, B., Li, M., Wan, L., and Xia, F. 2019. Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges. *IEEE Internet of Things Journal* 6, 5, 8095–8113.

Leyton-Brown, K., Milgrom, P., and Segal, I. 2017. Economics and computer science of a radio spectrum reallocation. *Proceedings of the National Academy of Sciences* 114, 28, 7202–7209.

Li, G., Wang, J., Zheng, Y., and Franklin, M. J. 2016. Crowdsourced data management: A survey. *IEEE Transactions on Knowledge and Data Engineering* 28, 9, 2296–2319.

Li, H., Wang, H., Yin, W., Li, Y., Qian, Y., and Hu, F. 2015. Development of a remote monitoring system for henhouse environment based on iot technology. *Future Internet* 7, 3, 329–341.

Li, J., Wu, J., and Zhu, Y. 2019. Selecting optimal mobile users for long-term environmental monitoring by crowdsourcing. In *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*. 1–10.

Li, J., Shen, H., Narman, H. S., Chung, W., and Lin, Z. 2018. A survey of mobile crowdsensing techniques: A critical component for the internet of things. *ACM Trans. Cyber-Phys. Syst.* 2, 3.

Liu, Y. and Chen, Y. 2016. Learning to incentivize: Eliciting effort via output agreement. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI’16. AAAI Press, 3782–3788.

Manyika, J., Chui, M., and Bisson, P. 2015. The internet of things: mapping the value beyond the hype.

McAfee, R. P. 1992. A dominant strategy double auction. *Journal of Economic Theory* 56, 2, 434–450.

Mukhopadhyay, J., Singh, V. K., Pal, A., and Kumar, A. 2022. A truthful budget feasible mechanism for iot-based participatory sensing with incremental arrival of budget. *Journal of Ambient Intelligence and Humanized Computing* 13, 1107–1124.

Myerson, R. B. and Satterthwaite, M. A. 1983. Efficient mechanisms for bilateral trading. *Journal of Economic Theory* 29, 2, 265–281.

Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., and Kawatsuma, S. 2013. Emergency response to the nuclear accident at the fukushima daiichi nuclear power plants using mobile rescue robots. *Journal of Field Robotics* 30, 1, 44–63.

Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. 2007. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA.

Pan, Z., Yu, H., Miao, C., and Leung, C. 2017. Crowdsensing air quality with camera-enabled mobile devices. In *AAAI*.
Peng, D., Wu, F., and Chen, G. 2015. 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*. MobiHoc ’15. Association for Computing Machinery, New York, NY, USA, 177–186.

Pereira, C. and Aguiar, A. 2014. Towards efficient mobile m2m communications: Survey and open challenges. *Sensors* 14, 10, 19582–19608.

Phuttharak, J. and Loke, S. W. 2019. A review of mobile crowdsourcing architectures and challenges: Toward crowd-empowered internet-of-things. *IEEE Access* 7, 304–324.

Plott, C. R. and Gray, P. 1990. The multiple unit double auction. *Journal of Economic Behavior & Organization* 13, 2, 245–258.

Poblelet, M., García-Cuesta, E., and Casanovas, P. 2014. Crowdsourcing tools for disaster management: A review of platforms and methods. In *AI Approaches to the Complexity of Legal Systems*, P. Casanovas, U. Pagallo, M. Palmirani, and G. Sartor, Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 261–274.

Pődör, A. and Szabó, S. 2021. Geo-tagged environmental noise measurement with smartphones: Accuracy and perspectives of crowdsourced mapping. *Environment and Planning B: Urban Analytics and City Science* 48, 9, 2710–2725.

Qiu, T., Xiao, H., and Zhou, P. 2013. Framework and case studies of intelligence monitoring platform in facility agriculture ecosystem. *2013 Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 522–525.

Qolomany, B., Mohammed, I., Al-Fuqaha, A., Guizani, M., and Qadir, J. 2021. Trust-based cloud machine learning model selection for industrial iot and smart city services. *IEEE Internet of Things Journal* 8, 4, 2943–2958.

Rahman, M. A., Rashid, M., Barnes, S., Hossain, M. S., Hassanain, E., and Guizani, M. 2019. An iot and blockchain-based multi-sensory in-home quality of life framework for cancer patients. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*. 2116–2121.

Rana, R. K., Chou, C. T., Kanhere, S. S., Bulusu, N., and Hu, W. 2010. 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*. IPSN ’10. Association for Computing Machinery, New York, NY, USA, 105–116.

Ray, P. 2018. A survey on internet of things architectures. *Journal of King Saud University - Computer and Information Sciences* 30, 3, 291–319.

Ray, S., Adhikary, P., Lana, S., Mitra, S., Halder, T., Paul, M., Mukherjee, A., Koshika, De, A., Chakravorty, N., Goswami, R., Kundu, V., and Sarkar, D. 2018. A survey paper on architecture of internet of things. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 908–913.
Roughgarden, T. 2014. CS364B: Frontiers in mechanism design, (Stanford University Course), Lecture #4: The clinching auction.

Roughgarden, T. 2016. CS269I: Incentives in computer science (Stanford University course). Lecture 3: Strategic Voting.

Samulowska, M., Chmielewski, S., Raczkó, E., Lupa, M., Myszkowska, D., and Zagajewski, B. 2021. Crowdsourcing without data bias: Building a quality assurance system for air pollution symptom mapping. *ISPRS International Journal of Geo-Information* 10, 2, 1–28.

Segal-Halevi, E., Hassidim, A., and Aumann, Y. 2018a. Double auctions in markets for multiple kinds of goods. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 489–497.

Segal-Halevi, E., Hassidim, A., and Aumann, Y. 2018b. MUDA: A truthful multi-unit double-auction mechanism. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI18/IAAI18/EAAI18. AAAI Press.

Siarry, P., Jabbar, M., Aluvalu, R., Abraham, A., and Madureira, A. 2021. The Fusion of Internet of Things, Artificial Intelligence, and Cloud Computing in Health Care. Internet of Things, Technology, Communications and Computing. Springer International Publishing.

Singh, V. K. and Mishra, S. 2022. A truthful mechanism for time-bound tasks in IoT-based crowdsourcing with zero budget.

Singh, V. K., Mukhopadhyay, S., Xhafa, F., and Krause, P. 2020a. A quality-assuring, combinatorial auction based mechanism for IoT-based crowdsourcing. In *Advances in Edge Computing: Massive Parallel Processing and Applications*. Vol. 35. IOS Press, 148–177.

Singh, V. K., Mukhopadhyay, S., Xhafa, F., and Sharma, A. 2020b. A budget feasible peer graded mechanism for iot-based crowdsourcing. *Journal of Ambient Intelligence and Humanized Computing* 11, 4, 1531–1551.

Staniek, M. 2021. Road pavement condition diagnostics using smartphone-based data crowdsourcing in smart cities. *Journal of Traffic and Transportation Engineering (English Edition)* 8, 4, 554–567.

Sun, G., Sun, S., Yu, H., and Guizani, M. 2020. Toward incentivizing fog-based privacy-preserving mobile crowdsensing in the internet of vehicles. *IEEE Internet of Things Journal* 7, 5, 4128–4142.

Sun, J. and Li, G. 2020. Designing a double auction mechanism for the re-allocation of emission permits. *Annals of Operations Research* 291, 847–874.

Talavera, J. M., Tobón, L. E., Gómez, J. A., Culman, M. A., Aranda, J. M., Parra, D. T., Quiroz, L. A., Hoyos, A., and Garreta, L. E. 2017. Review of iot applications in agro-industrial and environmental fields. *Computers and Electronics in Agriculture* 142, 283–297.
TAN, W. and JIANG, Z. 2019. A novel experience-based incentive mechanism for mobile crowd-sensing system. In Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing. AIIPCC ’19. Association for Computing Machinery, New York, NY, USA.

WANG, H., GUO, S., CAO, J., and GUO, M. 2018. Melody: A long-term dynamic quality-aware incentive mechanism for crowd-sourcing. IEEE Trans. Parallel Distrib. Syst. 29, 4, 901–914.

WEI, Y., ZHU, Y., ZHU, H., ZHANG, Q., and XUE, G. 2015. Truthful online double auctions for dynamic mobile crowd-sourcing. In 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, Hong Kong, China, 2074–2082.

WU, C., WANG, Z., HU, S., LEPINE, J., NA, X., AINALIS, D., and STETTLER, M. 2020. An automated machine-learning approach for road pothole detection using smartphone sensor data. Sensors 20, 19, 1–23.

V. WICKERY. 1961. Counterspeculation, auctions and competitive sealed tenders. Journal of Finance. 16, 1, 837.

YAN, Z., ZHANG, P., AND VASILAKOS, A. V. 2014. A survey on trust management for internet of things. Journal of Network and Computer Applications 42, 120–134.

YANG, D., XUE, G., FANG, X., AND TANG, J. 2012. Crowd-sourcing to smart-phones: Incentive mechanism design for mobile phone sensing. In Proceeding of 18th Annual International Conference on Mobile Computing and Networking. ACM, Istanbul, Turkey, 173–184.

YIN, M. AND CHEN, Y. 2015. Bonus or not? learn to reward in crowd-sourcing. In Proceedings of the 24th International Conference on Artificial Intelligence. IJCAI’15. AAAI Press, Buenos Aires, Argentina, 201–207.

ZANELLA, A., BUI, N., CASTELLANI, A., VANGELISTA, L., AND ZORZI, M. 2014. Internet of things for smart cities. IEEE Internet of Things Journal 1, 1, 22–32.

ZHANG, L. 2011. An iot system for environmental monitoring and protecting with heterogeneous communication networks. In 2011 6th International ICST Conference on Communications and Networking in China (CHINACOM). IEEE, Harbin, China, 1026–1031.

Zhou, J., CaO, Z., Dong, X., and VASILAKOS, A. V. 2017. Security and privacy for cloud-based iot: Challenges. IEEE Communications Magazine 55, 1, 26–33.

ZHOU, X. AND ZHENG, H. 2009. Trust: A general framework for truthful double spectrum auctions. In IEEE INFOCOM 2009. IEEE, Rio de Janeiro, Brazil, 999–1007.