Truthful Incentive Mechanism for Nondeterministic Crowdsensing with Vehicles

Guojun Gao, Mingjun Xiao, Member, IEEE, Jie Wu, Fellow, IEEE, Liusheng Huang, Member, IEEE, and Chang Hu

Abstract—In this paper, we focus on the incentive mechanism design for a vehicle-based, nondeterministic crowdsensing system. In this crowdsensing system, vehicles move along their trajectories and perform corresponding sensing tasks with different probabilities. Each task may be performed by multiple vehicles jointly so as to ensure a high probability of success. Designing an incentive mechanism for such a crowdsensing system is challenging since it contains a non-trivial set cover problem. To solve this problem, we propose a truthful, reverse-auction-based incentive mechanism that includes an approximation algorithm to select winning bids with a nearly minimum social cost and a payment algorithm to determine payments for all participants. Moreover, we extend the problem to a more complex case in which the Quality of sensing Data (QoD) of each vehicle is taken into consideration. For this problem, we propose a QoD-aware incentive mechanism, which consists of a QoD-aware winning-bid selection algorithm and a QoD-aware payment determination algorithm. We prove that the proposed incentive mechanisms have truthfulness, individual rationality, and computational efficiency. Moreover, we analyze the approximation ratios of the winning-bid selection algorithms. The simulations, based on a real vehicle trace, also demonstrate the significant performances of our incentive mechanisms.

Index Terms—Incentive mechanism, nondeterministic crowdsensing, quality of data, reverse auction, truthful

1 INTRODUCTION

In recent years, vehicles have been equipped with more and more components that can provide a better user experience such as wireless network interfaces, event data recorders, vehicular computers, etc. Vehicles that carry these components can be considered programmable and powerful mobile sensors which are able to communicate with the Internet and with each other. Furthermore, they move along roads day after day, and thus have the potential to collect data and permit the enabling of numerous novel applications such as traffic management [2], mobile advertising [3], [4], environment monitoring [5], [6], etc. All of these applications can be formalized as outsourcing location-based sensing tasks to mobile vehicles, which are also called vehicle-based crowdsensing [7], [8]. Roughly speaking, vehicle-based crowdsensing involves a platform that receives task requests from platform users and dispatches sensing tasks to mobile vehicles that are willing to serve.

Stimulating enough vehicles to participate in the crowdsensing is one of the most critical issues since it determines whether the crowdsensing can provide adequate sensing quality. While performing sensing tasks, participants may consume some resources, such as battery, storage, cpu, etc., and may even suffer threats to their privacy [9], [10], [11]. These factors could discourage them from participating in crowdsensing unless they receive sufficient rewards to compensate for the expenditures and the risks. Hence, an incentive mechanism that determines which participants should be recruited and how much reward should be paid to each of them is necessary. However, what makes the incentive mechanism design highly complicated is that a participant might strategically claim a higher cost than the real one to increase his/her payoff. Additionally, the mechanism also needs to minimize the social cost (i.e., the total sensing cost) and ensure the successful probability of performing tasks, which all contribute to the challenge.

In this paper, we focus on the incentive mechanism design for vehicle-based, nondeterministic crowdsensing. Consider a typical vehicle-based, nondeterministic crowdsensing system like [8], which consists of a platform, several platform users, and many mobile vehicles. The platform receives the sensing tasks associated with some Places of Interest (Pols) from the platform users. The vehicles move along streets and can communicate with the platform via road-side infrastructures or cellular networks so that the platform can select vehicles to perform the sensing tasks. In general, these tasks are associated with different Pols. Vehicles might be selected to perform a task only when they pass by the corresponding Pol, as illustrated in Fig. 1. Real vehicular trace analysis has shown that the mobile trajectory of each vehicle in real applications is nondeterministic [3], [8], [12], [13], [14]. Therefore, it is a probabilistic event that each vehicle will pass by a Pol and will perform the related task (i.e., covers this task). Due to this nondeterminacy, the platform user will require that the probability of each task being performed successfully is no less than a specified
We design a reverse-auction-based incentive mechanism to guarantee the total sensing quality of the crowdsensing. The platform, therefore, needs to stimulate enough vehicles to participate in the crowdsensing by using an incentive mechanism. This incentive mechanism should consider the truthfulness, individual rationality, computational efficiency, minimum social cost, and the constraints of vehicles’ possible mobile trajectories simultaneously.

Currently, there have been many incentive mechanisms designed for crowdsensing/crowdsourcing \cite{15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33}. In \cite{15}, Yang et al. design two incentive mechanisms, CCM and UCM, to maximize the utility of the crowdsourcer. They build the incentive mechanisms based on Stackelberg game and reverse auction, respectively. In \cite{16}, Zhao et al. propose an online incentive mechanism to maximize the utility of the crowdsourcer within a given budget. In \cite{17}, Wei et al. tend to stimulate both workers and crowdsourcers to participate in the dynamic mobile crowdsensing. They design a general incentive mechanism framework based on double auction to ensure the budget balance. In \cite{24}, Feng et al. design a social cost efficient incentive mechanism, TRAC, for the location-aware collaborative crowdsensing. In TRAC, each mobile user can perform a few location-related tasks and multiple users are stimulated to cooperatively perform a group of tasks. However, all of these incentive mechanisms only involve deterministic crowdsensing in which each user is assumed to perform a task either with a probability of 100 percent or with a zero probability. In contrast, we focus on the nondeterministic crowdsensing where tasks are performed by each vehicle/user with different probabilities from zero to 100 percent. Such a nondeterministic crowdsensing scenario is more consistent with the real world. Nevertheless, it will result in a non-integer set cover problem with non-linear constraints. The aforementioned incentive mechanisms for deterministic crowdsensing are not competent for this novel nondeterministic crowdsensing.

In this paper, we design a truthful, reverse-auction-based incentive mechanism to meet the requirements of our scenario. Since our mechanism uses a similar process for each platform user, we will only consider the scenario that consists of a single platform user. First, the platform receives the tasks, probability thresholds, and the time intervals from the platform user. It then sends them to the vehicles registered in the platform. After that, the vehicles reply with bids that contain the tasks on their possible trajectories and the corresponding costs. Next, the platform decides which bids should be selected to minimize the social cost, while ensuring that the joint probability of each task being successfully performed is no less than the given threshold. The platform also determines a payment for each bid in order to guarantee that each vehicle will report the real costs (i.e., the truthfulness) and that the payoff of each bid is non-negative (i.e., the individual rationality). Finally, the platform will notify the vehicles of their winning bids, pay the vehicles after receiving sensing data from them, charge the platform user for the payments, and send the platform the sensing data. In summary, the incentive mechanism mainly consists of a winning-bid selection algorithm and a payment determination algorithm. Indeed, the designed incentive mechanism can be applied in many fields in real life, such as environment and noise monitoring \cite{5}, \cite{6}, crowd labeling \cite{19}, \cite{34}, social networks \cite{35} and so on.

We highlight the main contributions as follows:

- We design a reverse-auction-based incentive mechanism for vehicle-based, nondeterministic crowdsensing. To the best of our knowledge, this is the first work on crowdsensing incentive mechanism design that takes into consideration truthfulness, individual rationality, computational efficiency, social cost efficiency, and the nondeterministic crowdsensing scenario, where each task is performed with a joint probability, simultaneously.
- We prove that the winning-bid selection problem in our scenario is NP-hard since it leads to a non-trivial set cover problem with non-linear constraints. To solve the problem efficiently, we propose a nearly optimal winning-bid selection algorithm and analyze the approximation ratio.
- We propose an algorithm to determine the payments for the winning bids and theoretically prove that these payments can ensure the truthfulness and individual rationality of the mechanism.
- We extend the incentive mechanism design problem to a case where the Quality of Data (QoD) is taken into consideration. We propose a QoD-aware incentive mechanism consisting of a QoD-aware winning-bid selection algorithm and payment determination algorithm. We prove that the QoD-aware incentive mechanism is truthful and individually rational. Moreover, we analyze the approximation ratio of the QoD-aware winning-bid selection algorithm.
- We conduct extensive simulations on a real vehicle trace to demonstrate the significant performances of the proposed incentive mechanisms.

The rest of the paper is organized as follows. In Section 2, we introduce the system model and the problem. Then, we describe the detailed design of our mechanism and the related extension in Sections 3 and 4, respectively. The theoretical analysis and the evaluation of the incentive mechanisms are presented in Sections 5 and 6, respectively. After reviewing the related work in Section 7, we finally conclude the paper in Section 8.

2 SYSTEM MODEL AND PROBLEM

2.1 System Model

We consider a vehicle-based, nondeterministic crowdsensing system which consists of a platform, several platform users, and many vehicles. The platform accepts sensing requests from platform users who connect to the platform via Internet and negotiates with the vehicles either via...
cellular networks or road-side infrastructures [36], [37], [38], [39], which are left up to their preference. The platform will pay the rewards to vehicles after receiving data from them and will charge the platform users for the payments.

The system might contain several platform users. Since the sensing tasks for different platform users are distinct, the submitted bids are also separate. Here, the bid submitted by each vehicle indicates the cost of performing one platform user’s sensing tasks, instead of the cost of performing all platform users’ tasks on this trajectory. For the simplicity of descriptions, we only consider one platform user, and we let the platform conduct the same operations for other platform users in the practical scenario. The platform user wants to collect data (e.g., traffic congestion, noise pollution, air quality, etc.) from m Pols which are distributed in different streets. Hence, it produces m Pol-related sensing tasks. These sensing tasks are denoted as a task set \( S = \{s_1, s_2, \ldots, s_m\} \) in which \( s_i \) is the \( i \)th task (\( 1 \leq i \leq m \)). Since the sensing data is time-sensitive, the platform user requires that the tasks are performed within a time interval \([t_1, t_2]\). The sensing data is meaningful only when the tasks are completed within the time interval \([t_1, t_2]\). Without Loss Of Generality (WLOG), we assume that \( t_1 = 0 \) and \( t_2 = D \). Since each task is performed with a probability in the nondeterministic crowdsensing, the platform demands that the probability of each task in \( S \) being successfully performed be no less than a threshold \( \eta \).

The vehicle-based, nondeterministic crowdsensing system also includes \( n \) mobile vehicles, denoted as \( V = \{v_1, v_2, \ldots, v_n\} \). These mobile vehicles move around different streets so that they might cover and perform the sensing tasks in \( S \). In general, the mobile behaviors of vehicles are uncertain. This characteristic has been captured by many real vehicular traces [3], [8], [12], [13], [14], [38]. Based on this observation, we assume that each vehicle has multiple possible trajectories and that each trajectory has a probability. More specifically, vehicle \( v_j \) has \( l_j \) possible trajectories. Each trajectory might cover a group of tasks. The tasks covered by vehicle \( v_j \) are denoted as \( \{S_{j}^1, S_{j}^2, \ldots, S_{j}^l_j\} \) in which \( S_{j}^k \) is the set of tasks covered by the \( k \)th trajectory (\( 1 \leq k \leq l_j \)). Since the execution time of tasks is much smaller than the driving time of vehicles and each vehicle can obtain more extra income by performing more sensing tasks on a trajectory, each vehicle is always willing to perform all sensing tasks on its one driving trajectory. That is, the tasks in \( S_{j}^k \), covered by the same trajectory, will be performed as a whole which means that either all of the tasks are performed or none of them are. Moreover, the probability of tasks in \( S_{j}^k \) being performed is the probability of the \( k \)th trajectory, denoted by \( q_{j}^k \).

When a vehicle performs sensing tasks, it will consume battery, storage, cpu, and so on, which will result in a cost. The cost of \( v_j \) performing all tasks in \( S_{j}^k \) is denoted by \( c_{j}^k \).

In the above system, the vehicles are deemed as common office workers. That is, the primary goal of the vehicles is to drive to their destinations, and the secondary goal is to perform some sensing tasks on their trajectories. All vehicles may submit bids for their possible trajectories. However, they select the trajectories according to the practical situation (e.g., traffic condition) at that time, instead of selecting the trajectories that maximize their benefits. Here, we assume that the platform can derive the value of \( q_{j}^k \) from \( v_j \)'s historic movement records. This assumption is reasonable since the platform can trace the daily movements of the vehicles, and therefore, the platform can derive the trajectories and probabilities of vehicles. We also assume that the time it takes each vehicle to perform a task can be ignored since it is far less than that of the vehicle visiting a PoI in magnitudes. Here, we only consider the tasks that can be completed before \( D \). If a trajectory covers some extra tasks whose performing times are beyond \( D \), we will deem that these extra tasks cannot be performed by the corresponding vehicle.

2.2 Reverse-Auction-Based Incentive Mechanism

In the vehicle-based, nondeterministic crowdsensing system, the platform adopts a reverse-auction-based incentive mechanism to select participants and to determine the payments for them after receiving the request from the platform user. Since the tasks on a trajectory will be performed together, the mechanism is actually based on a reverse and combinatorial auction. The whole incentive mechanism mainly includes five rounds of interactions between the platform and vehicles, as illustrated in Fig. 2:

(1) The platform announces all Pol-related sensing tasks in \( S \) as well as \( D \) to the vehicles in \( V \) after receiving from the platform user.

(2) Each vehicle \( v_j \) will reply to the platform with a set of bids, each of which is a tasks-bid pair \( \beta_j = (S_{j}^k, b_{j}^k) \), in which \( S_{j}^k \) is the set of tasks covered by \( v_j \)'s \( k \)th trajectory and \( b_{j}^k \) is the cost claimed by \( v_j \) for performing all tasks in \( S_{j}^k \). Here, the \( \beta_j \) is valid only when \( v_j \) moves along the \( k \)th trajectory and performs the tasks in \( b_{j}^k \). (Note that, due to the nondeterminacy of \( v_j \)'s mobile behaviors, it cannot guarantee that the claimed tasks in \( b_{j}^k \) will be performed indeed).

(3) After receiving replies from all vehicles in \( V \), the platform selects a set of winning bids, denoted by \( \Phi \), from the received bids, denoted by \( \Gamma \), to guarantee that all tasks in \( S \) are performed with a probability no less than a specified threshold \( \eta \). A bid \( \beta_j \in \Phi \) indicates that \( v_j \) will be selected to perform the tasks in \( S_{j}^k \). After determining \( \Phi \), the platform will notify the vehicles of the corresponding winning bids. Also, the platform will determine the payment \( p_{j}^k \) for each bid \( \beta_j \) in \( \Phi \).

(4) After receiving the winning bids, vehicle \( v_j \) will perform the tasks in each winning bid \( \beta_j \) until \( D \), when \( v_j \) moves along the \( k \)th trajectory. Moreover, \( v_j \) will send the results back to the platform after it completes all tasks in \( S_{j}^k \).

(5) The platform will pay vehicle \( v_j \) with the money \( p_{j}^k \) for the bid \( \beta_j \) after receiving the results of \( S_{j}^k \).

Note that the payment should be fair and the payoff of a bid \( \beta_j \), which is defined as \( p_{j}^k - c_{j}^k \), should be reasonable to attract more vehicles. However, the platform only knows the claimed cost \( b_{j}^k \) instead of the real cost \( c_{j}^k \) that \( v_j \) spends
Individual Rationality. Individual rationality indicates that each winner should be paid with a value no less than its real cost, which implies that each winner’s payoff is negative. Due to the non-negative payoff, each winner is willing to participate in the crowdsensing.

Truthfulness. Truthfulness means that no bidder can improve his or her payoff by submitting different costs from the true values. According to Myerson’s Theorem [41], a mechanism is truthful if and only if: the winner selection rule is monotone and each winner is paid with a critical value. The monotonicity indicates that if \( v_j \) wins the bid \( \beta_k \) when it claims a cost \( b_k \) for performing \( S_k \), it will still win the bid when claiming a smaller cost \( b_k' \)(\( \leq b_k \)). The critical value is the maximum bid value for a bid to win.

Additionally, the frequently used notations in this paper are summarized in Table 1.

### 3 Design of Incentive Mechanism

In this section, we propose a reverse-auction-based incentive mechanism consisting of solutions to the MCBS problem and the payment determination problem. We first analyze the NP-hardness of the MCBS problem, and then propose an approximation algorithm to resolve MCBS efficiently. Next, we propose another algorithm to compute the payments for all winning bids, which will induce the vehicles to report their costs truthfully.

#### 3.1 NP-Hardness of MCBS

First, we analyze the complexity of the MCBS problem and derive the following theorem.

**Theorem 1.** The MCBS problem is NP-hard.

**Proof.** To prove the NP-hardness of the MCBS problem, we first prove that a special case of MCBS, where the probabilities of all trajectories are same (i.e., \( q_k = \eta \) for all \( k \)), is NP-hard. Since the constraint (3) can always be met, the special MCBS problem is actually to select a set of bids with least social cost to cover \( S \). Now, we introduce a well-known NP-hard problem, i.e., minimum weighted set cover (MWSC) problem. Given a set of elements \( S = \{s_1, \ldots, s_m\} \) and the collection of some subsets \( \Gamma = \{S_1, \ldots, S_n\} \) in which \( S_i \subseteq S \) for all \( i \) and \( S_i \) has a weight \( c_i \), find a least weight collection \( \Phi \) of subsets of \( \Gamma \) such that \( \Phi \) covers all elements in \( S \). By mapping \( S_i \) and \( c_i \) in MWSC to \( S_k \) and \( c_k \) in the special case of MCBS, respectively, we reduce the MWSC problem to the special MCBS problem in constant time. Obviously, the MWSC problem is actually equivalent to the special case of MCBS. Therefore, the special MCBS problem is NP-hard. Further, the general MCBS problem is at least NP-hard and the theorem holds.

---

**Table 1: Frequently Used Notations**

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S, V )</td>
<td>the sets of tasks and vehicles, respectively.</td>
</tr>
<tr>
<td>( i, j, k )</td>
<td>the indexes for tasks, vehicles and trajectories, respectively.</td>
</tr>
<tr>
<td>( S_k^j )</td>
<td>the set of tasks covered by ( v_j )'s ( k )th trajectory.</td>
</tr>
<tr>
<td>( q_k^j )</td>
<td>the probability of ( v_j )'s ( k )th trajectory.</td>
</tr>
<tr>
<td>( c_k^j )</td>
<td>the real cost that ( v_j ) spends on performing ( S_k^j ).</td>
</tr>
<tr>
<td>( b_k^j )</td>
<td>the claimed cost (i.e., bid) for ( v_j ) on ( S_k^j ).</td>
</tr>
<tr>
<td>( \beta_k^j )</td>
<td>the bid of ( v_j ), containing ( S_k^j ) and ( b_k^j ).</td>
</tr>
<tr>
<td>( p_k^j )</td>
<td>the payment for the bid ( \beta_k^j ).</td>
</tr>
<tr>
<td>( \Gamma, \Gamma - \beta_k^j )</td>
<td>the set of bids received by the platform, and the set of bids except ( \beta_k^j ) respectively.</td>
</tr>
<tr>
<td>( \Phi, \Phi^i )</td>
<td>the set of winning bids selected from ( \Gamma ) and ( \Gamma - \beta_k^j ) respectively.</td>
</tr>
<tr>
<td>( \rho_i^\Phi )</td>
<td>the joint probability of ( s_i ) being successfully performed based on the winning bid set ( \Phi ).</td>
</tr>
<tr>
<td>( \eta )</td>
<td>the threshold of the joint probability.</td>
</tr>
</tbody>
</table>

---

Reasonable since we can add more vehicles to \( V \) until the problem is solvable.

Second, the payment determination problem is to compute payments for winning bids. The payments should ensure that the vehicles are willing to participate in the crowdsensing and that they will not manipulate their claimed costs. Thus, the payment computation needs to make the whole incentive mechanism satisfy the following properties:

- **Individual Rationality.**
- **Truthfulness.**
Here, we emphasize that the MCBS problem is actually a non-trivial set cover problem. This is because the objective function of this problem is not an integer function. Furthermore, the constraint of this problem is a group of non-linear real functions. Owing to these two characteristics, the set cover approximation algorithms in [12], [15], [24], [42] cannot be utilized directly to solve the MCBS problem.

3.2 Winning-Bid Selection Algorithm

Since the MCBS problem is NP-hard, we propose a winning-bid selection approximation algorithm to solve it. Here, we first assume that all vehicles report their costs truthfully, i.e., \( \beta_i = c_i \). We will prove that this truthful assumption is reasonable and correct in the next section.

The approximation algorithm is based on a utility function and the marginal contributions of all bids in \( \Gamma \). The utility function \( U(\Phi) \) is the sum of the probabilities that all tasks in \( S \) are successfully performed with when selecting all bids in \( \Phi \), defined as follows:

\[
U(\Phi) \triangleq \sum_{i=1}^{m} \min\{\rho_i, \eta\}.
\]  

The marginal contribution function of a bid \( \beta_k \in \Gamma - \Phi \) is the increased utility after adding \( \beta_k \) into \( \Phi \), defined as follows:

\[
U_{\beta_k}(\Phi) \triangleq U(\Phi \cup \{\beta_k\}) - U(\Phi) = \sum_{i \in S_k} \left( \min\{\rho_i, \beta_k\} - \min\{\rho_i, \eta\} \right).
\]

The approximation algorithm mainly contains an iterative bid selection process. In each round of iteration, we greedily select the bid whose Marginal Contribution Per Cost (MPC) is the maximal to expand the winning set until the utility of the whole winning set reaches the maximum value. The detailed algorithm is presented in Algorithm 1. At the beginning, the winning-bid set \( \Phi \) is initialized as \( \emptyset \) in Step 1. From Step 2 to Step 7, the greedy strategy, based on the utility function and the marginal contribution function, is adopted to select the winning bids in Step 3.

Here, we point out that although the basic formalism of our algorithm looks similar to the traditional set cover approximation algorithms (e.g., [15], [24]), our algorithm is intrinsically different from them. In general, the greedy criterion of the traditional algorithms is directly the value of the optimization objective defined in the problem. In contrast, our greedy criterion is the special utility function that we build for MCBS. This utility function not only contains the optimization objective of the MCBS problem, but also takes the non-linear constraint (3) into consideration. Furthermore, unlike existing algorithms that only deal with the integer set cover problems, our algorithm can solve the set cover problem with a real optimization objective function, such as MCBS.

The correctness of Algorithm 1 is supported by the following theorem.

**Theorem 2.** Algorithm 1 can always produce a feasible solution if MCBS is solvable.

**Proof.** If MCBS is solvable, selecting all bids in \( \Gamma \) will meet the constraints of MCBS. That is to say, \( \Gamma \) is at least a feasible solution, i.e., \( \rho_i \geq \eta \) for all \( s_i \in S \), and \( U(\Gamma) = m\eta \). Hence, Algorithm 1 will terminate for sure, either before or when all bids in \( \Gamma \) are added into \( \Phi \). When Algorithm 1 terminates, \( U(\Phi) = m\eta \), indicating that \( \min\{\rho_i, \eta\} = \eta \) for all \( s_i \in S \). Then, \( \rho_i \geq \eta \) for all \( s_i \in S \), meeting the constraint (3). Therefore, Algorithm 1 produces a feasible solution for MCBS after it terminates, and the theorem holds.

### Algorithm 1. Winning-Bid Selection Algorithm

**Input:** \( \Gamma, m, \eta, \{q_i|s_i \in \Gamma\} \) before \( D \)

**Output:** winning bid set \( \Phi \), social cost \( C(\Phi) \)

1. \( \Phi \leftarrow \emptyset, U(\Phi) \leftarrow 0, C(\Phi) \leftarrow 0 \)
2. while \( U(\Phi) < m\eta \) do
3. Select a bid \( \beta_k \) from \( \Gamma - \Phi \) to maximize \( \frac{U_{\beta_k}(\Phi)}{\beta_k} \)
4. \( U(\Phi) \leftarrow U(\Phi) + U_{\beta_k}(\Phi) \)
5. \( \Phi \leftarrow \Phi \cup \{\beta_k\} \)
6. \( C(\Phi) \leftarrow C(\Phi) + \beta_k \)
7. return \( \Phi, C(\Phi) \)

### 3.3 Payment Determination Algorithm

Besides the MCBS problem, the incentive mechanism also needs to solve the payment determination problem, i.e., it must decide how much to pay for each bid that is selected by Algorithm 1 while ensuring that the mechanism is truthful and individually rational.

To guarantee the truthfulness of the mechanism, the payment \( p_i \) for a winning bid \( \beta_i \) should depend on other bids in \( \Gamma \) instead of \( \beta_i \) itself. Therefore we first remove \( \beta_k \) from \( \Gamma \) to get a new bid set, denoted by \( \Gamma - \beta_k \). Based on \( \Gamma - \beta_k \), we reselect a new winning-bid set that is denoted by \( \Phi' \). Assume that in the \( q \)-th round of iteration of this new selection, \( \Phi' \) is the winning set, and the winning bid \( \beta_i \) is in \( \Phi' \). If \( \beta_i \) wants to win in the \( q \)-th round, the related cost must be no more than the value \( \beta_i \cdot U_{\beta_i}(\Phi'_{q-1}) / U_{\beta_i}(\Phi'_{q-1}) \). Otherwise, the MPC of \( \beta_i \) will not be the maximal. We therefore determine \( p_i \) based on this critical value. To guarantee the individual rationality, the payment \( p_i \) must be no less than the real cost \( c_i \). Hence, we set \( p_i \) as the maximum critical value

\[
p_i = \max \left\{ \frac{U_{\beta_i}(\Phi'_{q-1})}{U_{\beta_i}(\Phi'_{q-1})}, \beta_i|q = 1, 2, \ldots \right\}.
\]

The detailed process is presented in Algorithm 2. The algorithm traverses all bids in \( \Gamma \) to decide the payment for each bid in Step 1. In Step 2, we initialize the payment \( p_i \) for \( \beta_i \) as 0 and the new winning set \( \Phi' \) as \( \emptyset \). If \( \beta_i \) is a winning bid in Step 3, we expand \( \Phi' \) according to the greedy strategy that is also used in Algorithm 1, and we set the payment as the maximal critical value in Steps 4-9. Note that we use an equivalent expression instead of (6) in Step 8.

### 3.4 A Walk-Through Example

To better understand the two algorithms in our incentive mechanism, we use an example in Fig. 3 to show the processes of Algorithms 1 and 2. In the example, \( m = 4, \eta = 0.6, m\eta = 2.4, q_1 = 0.35, q_2 = 0.4, q_3 = 0.4, q_4 = 0.45, \) \( q_5 = 0.5 \), and \( \Gamma \) is marked in Fig. 3. Algorithm 1 is conducted as follows:

- **First round:** \( \Phi = \emptyset, U(\Phi) = 0, C(\Phi) = 0 \)
- **Second round:** Due to \( U(\Phi) < 2.4 \), we compute \( \frac{U_{q_1}(\Phi)}{q_1} = 0.26, \frac{U_{q_2}(\Phi)}{q_2} = 0.27, \frac{U_{q_3}(\Phi)}{q_3} = 0.4, \frac{U_{q_4}(\Phi)}{q_4} = 0.34 \).
Fig. 3. An example: \(v_1, v_2, v_3\) send their bids to the platform.

\[
\frac{U_{\beta_i}(\Phi)}{b_i} = 0.5. \text{ Since } \frac{U_{\beta_1}(\Phi)}{b_1} \text{ is the maximal, we update } \\
\Phi = \{\beta_1\}. U(\Phi) = 1.5, C(\Phi) = 3.
\]

- **Third round:** Since \(U(\Phi) = 1.5 < 2.4\), we continue to compute \(\frac{U_{\beta_1}(\Phi)}{b_1} = 0.14, \frac{U_{\beta_2}(\Phi)}{b_2} = 0.07, \frac{U_{\beta_3}(\Phi)}{b_3} = 0.2, \frac{U_{\beta_4}(\Phi)}{b_4} = 0.16. \text{ As } \frac{U_{\beta_2}(\Phi)}{b_2} \text{ is the maximal, we update } \\
\Phi = \{\beta_1, \beta_2\}, U(\Phi) = 2.1, C(\Phi) = 6.
\]

- **Fourth round:** Owing to \(U(\Phi) = 2.1 < 2.4\), we compute \(\frac{U_{\beta_1}(\Phi)}{b_1} = 0.08, \frac{U_{\beta_2}(\Phi)}{b_2} = 0.03, \frac{U_{\beta_3}(\Phi)}{b_3} = 0.08. \text{ WLOG, we select } \beta_1 \text{ as the winner and update } \\
\Phi = \{\beta_1, \beta_2, \beta_3\}, U(\Phi) = 2.4, C(\Phi) = 10. \text{ Now } U(\Phi) = mn \text{ and Algorithm 1 terminates.}
\]

Algorithm 2. Payment Determination Algorithm

Input: \(\Gamma, \Phi, m, n, \{q_k\}, \{s_i\} \in \Gamma\) before \(D\)

Output: the payment \(p_i^k\) for all \(v_j \in [1, n]\) and all \(k \in [1, t_j]\)

1: for all \(\beta_i \in \Gamma\) do
2: \(\Phi' \leftarrow \emptyset, p_i^k \leftarrow 0\)
3: if \(\beta_i \in \Phi\) then
4: while \(U(\Phi') < mn\) do
5: \(\text{Select } \beta_i' \text{ from } \Gamma - \beta_i - \Phi' \text{ to maximize } \frac{U_{\beta_i}(\Phi')}{b_i}\)
6: \(U(\Phi') \leftarrow U(\Phi') + U_{\beta_i}(\Phi')\)
7: \(\Phi' \leftarrow \Phi' \cup \{\beta_i\}\)
8: \(p_i^k \leftarrow \max\{p_i^k, \frac{U_{\beta_i}(\Phi')}{b_i}\}\)
9: return \(p_i^k\) for all \(v_j \in [1, n]\) and all \(k \in [1, t_j]\)

The above calculation shows that the set of winning bids is \(\Phi = \{\beta_1, \beta_2, \beta_3\}\). Based on this result, Algorithm 2 is conducted as follows:

- **First round:** Since \(\beta_1\) is a winning bid, \(\Gamma - \beta_1 = \{\beta_2, \beta_1, \beta_2, \beta_3\}\). In the rounds of iteration from Step 4 to Step 9, the selected bids are \(\beta_1, \beta_1, \beta_2, \beta_2\) in turn. Accordingly, \(U_{\beta_2}(\Phi) \cdot b_2 = 2.1, U_{\beta_3}(\Phi) \cdot b_3 = 2.75, U_{\beta_1}(\Phi) \cdot b_1 = 4.24, \).

- **Second round:** \(p_1^k = 0\), as \(\beta_1 \notin \Phi\).

- **Third round:** Since \(\beta_2 \notin \Phi, \Gamma - \beta_1 = \{\beta_1, \beta_2, \beta_2, \beta_3\}\). In the computations of Steps 4-9, the selected bids are \(\beta_1, \beta_1, \beta_2, \beta_2\) in turn. Then \(\frac{U_{\beta_1}(\Phi)}{b_1} = 2.4, \frac{U_{\beta_2}(\Phi)}{b_2} = 3.69, \frac{U_{\beta_3}(\Phi)}{b_3} = 4.1, \frac{U_{\beta_1}(\Phi)}{b_1} = 4.\)

- **Fourth round:** As \(\beta_2\) is not a winning bid, \(p_2^k = 0\).

- **Fifth round:** Since \(\beta_1\) is a winning bid, \(\Gamma - \beta_1 = \{\beta_1, \beta_2, \beta_2, \beta_3\}\). In the rounds of iteration from Step 4 to Step 9, the selected bids are \(\beta_1, \beta_1, \beta_2, \beta_2\) in turn. Accordingly, \(U_{\beta_2}(\Phi) \cdot b_2 = 3.75, U_{\beta_1}(\Phi) \cdot b_1 = 4.24, U_{\beta_3}(\Phi) \cdot b_3 = 3. \text{ Hence, } p_1^k = 4.24. \)

We can find that the payment for each winning bid is no less than the related cost. Then \(v_1\) and \(v_2\) will perform the tasks successfully and will send the sensing data back to the platform. Hence, the platform only pays for \(\beta_1\) and \(\beta_2\) (as shown in Fig. 4). In this example, since \(v_3\) does not perform the tasks and send data back, it gets nothing.

4 QoD-Aware Incentive Mechanism

In this section, we extend our problem to a case in which the Quality of Data (QoD) is taken into consideration. We first introduce the extended problem, where we integrate the constraint (3) and the QoD constraint. After that, we propose the QoD-aware incentive mechanism consisting of a winning-bid selection algorithm and a payment determination algorithm for the extended problem.

4.1 The Extended Problem

In addition to the successful probability, the QoD is also a major factor that needs to be considered in mechanism design, as pointed out by [20], [21]. The QoD indicates the usability of the sensing data. It might be affected by various factors including poor sensor quality, noise, lack of sensor calibration, and so forth. We consider a general class of crowdsensing applications in which the availability and preciseness of services significantly depend on the QoD, e.g., noise pollution monitoring and air quality monitoring. In these cases, the qualities of the sensing data collected by vehicles are different, and the platform user requires that the total quality of the received data for each task be no less than a threshold. More specifically, we assume that the platform not only requires that the probability of each task being successfully performed be no less than \(q\), but also that the QoD for task \(s_i\) be no less than \(\xi\). Additionally, just like [21], we assume that the QoD profile of the vehicles, denoted by \(R = \{r_1, r_2, \ldots, r_n\}\), is known to the platform since the platform maintains a historical record of vehicles’ QoD profile \(R\). \(R\) can be calculated from the ground truth data or by utilizing algorithms such as those proposed in [20], [43], [44], [45]. Thus, the platform will receive a request including \(S, q, \{\xi_1, \xi_2, \ldots, \xi_m\}\) from the user.

To design a truthful incentive mechanism for this scenario, we need to solve both the winning-bid selection
problem and the payment determination problem. In addition to the probability constraints, the winning-bid selection must also meet the QoD requirements. As vehicle \( v_i \in \Phi \) performs each task in \( S^i \), with a probability of \( q^i_j \), we can calculate the QoD expectation value \( \mu_j^\Phi \) of task \( s_i \) as follows:

\[
\mu_j^\Phi = \frac{1}{\Phi \in \Phi \cap \Phi^i} (r_j q^i_j).
\]  

(7)

In the definition, the expectation value of QoD is derived from the real-world example, e.g., crowd labeling \([19],[34]\), where the sum of expectation values outputted by all winning vehicles denotes the final expected QoD. Then, we can formalize the QoD-MCBS problem as follows:

\[
\begin{align*}
\min C(\Phi) &= \sum_{\Phi^k \in \Phi} c_k^j \\
\text{s.t.} \quad &\Phi \subseteq \Gamma, \\
&\mu_i^\Phi \geq g, \\
&\mu_i^\Phi \geq q, \\
&1 \leq i \leq m.
\end{align*}
\]  

(8)

(9)

(10)

(11)

This QoD-MCBS problem can be simplified to MCBS when we let \( r_j \geq \max \{\xi_1, \ldots, \xi_m\} \) for \( \forall i \leq j \leq n \). Hence, QoD-MCBS is also NP-hard and an approximation algorithm is needed to solve this problem in polynomial time.

**Algorithm 3. QoD-Aware Winning-Bid Selection Algorithm**

**Input:** \( \Gamma, m, n, \mathcal{R}, \{\xi_1, \xi_2, \ldots, \xi_m\}, \{q^i_j\}_{i \in S^i} \in \Gamma \) before \( D \)

**Output:** winning bid set \( \Phi \), social cost \( C(\Phi) \)

1: \( \Phi \leftarrow \emptyset, G(\Phi) \leftarrow 0, C(\Phi) \leftarrow 0 \)
2: while \( G(\Phi) \leq mn + \sum_{i=1}^m \xi_i \) do
3: Select a bid \( \beta^i_k \) from \( \Gamma - \Phi \) to maximize \( G_j^\Phi(\Phi) \)
4: \( G(\Phi) \leftarrow G(\Phi) + G_j^\Phi(\Phi) \)
5: \( \Phi \leftarrow \Phi \cup \{\beta^i_k\} \)
6: \( C(\Phi) \leftarrow C(\Phi) + b_k^i \)
7: return \( \Phi, C(\Phi) \)

4.2 QoD-Aware Winning-Bid Selection Algorithm

To design an appropriate approximation algorithm, we first propose a QoD utility function and a QoD marginal contribution function. The QoD utility function \( F(\Phi) \), which is the sum of the QoD for all tasks, is defined as follows:

\[
F(\Phi) = \sum_{i=1}^m \min \{\mu_i^\Phi, \xi_i\}.
\]  

(12)

Based on \( F(\Phi) \), the QoD marginal contribution of a bid \( \beta^i_k \in \Gamma - \Phi \) is the increased QoD utility after adding \( \beta^i_k \) into \( \Phi \), defined as follows:

\[
F_{\beta^i_k}(\Phi) = F(\Phi \cup \{\beta^i_k\}) - F(\Phi) = \sum_{s_i \in S^i} (\min \{\mu_i^\Phi(\beta^i_k), \xi_i\} - \min \{\mu_i^\Phi, \xi_i\}).
\]  

(13)

To combine the constraints (10) and (11), we define the combination utility function \( G(\Phi) \) as follows:

\[
G(\Phi) = U(\Phi) + F(\Phi).
\]  

(14)

The corresponding combination marginal contribution is

\[
G_{\beta^i_k}(\Phi) = U_{\beta^i_k}(\Phi) + F_{\beta^i_k}(\Phi).
\]  

(15)

Based on the combination utility function and combination marginal contribution function, we propose the QoD-aware winning-bid selection algorithm in Algorithm 3. Similar to Algorithm 1, Algorithm 3 utilizes a greedy process in which the bid, whose combination MPC is the largest in each round of iteration, is added into the winning-bid set until the utility function achieves the maximum value \( m \eta + \sum_{i=1}^m \xi_i \). We prove the correctness of Algorithm 3 in the following theorem.

**Theorem 3.** Algorithm 3 can always produce a feasible solution if QoD-MCBS is solvable.

**Proof.** Similar to Theorem 2, \( \Gamma \) is at least a feasible solution. Hence, Algorithm 3 will terminate for sure either before or after adding all bids in \( \Gamma \) into a \( \Phi \). When it terminates, \( G(\Phi) = U(\Phi) + F(\Phi) = m \eta + \sum_{i=1}^m \xi_i \). According to (4) and (12), \( U(\Phi) \leq m \eta \) and \( F(\Phi) \leq \sum_{i=1}^m \xi_i \). Hence, when Algorithm 3 terminates, \( U(\Phi) = m \eta \) and \( F(\Phi) = \sum_{i=1}^m \xi_i \), i.e., \( \mu_i^\Phi \geq g \) and \( \mu_i^\Phi \geq \xi_i \) for \( \forall i \in S \). The theorem holds. \( \square \)

4.3 QoD-Aware Payment Determination Algorithm

To attract more vehicles and prevent them from manipulating the claimed costs, we need to decide a proper payment for each bid that is selected by Algorithm 3.

Actually, the idea of QoD-aware payment determination is similar to the basic payment determination in Section 3.3. We set the payment for each winning bid as the maximum critical value, i.e.,

\[
p_k^j = \max \left\{ \frac{G_j^\Phi(\Phi')}{G_{\beta^i_k}(\Phi')}, b_k^i \right\}, q = 1, 2, \ldots, \}
\]  

(16)

where the implications of the notations are the same as (6).

The detailed computation is presented in Algorithm 4. The algorithm first checks whether a bid \( \beta^i_k \) is winning in Step 3 and then recalculates the winning set when \( \beta^i_k \) is excluded from \( \Gamma \) from Steps 4-9. For the winning bids, the payments are the maximal critical values. For the losing bids, the payments are exactly 0.

**Algorithm 4. QoD-Aware Payment Determination Algorithm**

**Input:** \( \Gamma, \Phi, m, \eta, \mathcal{R}, \{\xi_1, \xi_2, \ldots, \xi_m\}, \{q^i_j\}_{i \in S^i} \in \Gamma \) before \( D \)

**Output:** the payment \( p_k^j \) for \( \forall j \in [1, n] \) and \( \forall k \in [1, l] \)

1: for all \( \beta^i_k \in \Gamma \) do
2: \( \Phi' \leftarrow \emptyset, p_k^j \leftarrow 0 \)
3: if \( \beta^i_k \in \Phi \) then
4: while \( G(\Phi') < m \eta + \sum_{i=1}^m \xi_i \) do
5: Select \( \beta^i_k \) from \( \Gamma - \Phi' \) to maximize \( G_j^\Phi(\Phi') \)
6: \( G(\Phi') \leftarrow G(\Phi') + G_j^\Phi(\Phi') \)
7: \( \Phi' \leftarrow \Phi' \cup \{\beta^i_k\} \)
8: \( p_k^j \leftarrow \max \{p_k^j, G_j^\Phi(\Phi') - \frac{G_{\beta^i_k}(\Phi')}{G_{\beta^i_k}(\Phi')}, b_k^i \} \)
9: return \( p_k^j \) for \( \forall j \in [1, n] \) and \( \forall k \in [1, l] \).
5 MECHANISM ANALYSIS

In this section, we prove that both the basic and QoD-aware incentive mechanisms have the properties of truthfulness, individual rationality and computational efficiency. We also analyze the approximation ratios of Algorithms 1 and 3. In each section, we first analyze the property of the basic incentive mechanism and then analyze the QoD-aware incentive mechanism based on the related property of the basic incentive mechanism.

WLOG, we consider the qth round of iteration in Algorithm 1. In this round, $\beta^*_k$ is the winning bid and is added into the winning set $\Phi_q$ (a winning subset of $\Gamma$). Moreover, we also consider the qth round of iteration for the payment computation of $\beta^*_k$ in Algorithm 2 (See Steps 4-9), in which the winning set is denoted as $\Phi_q^*$ (a winning subset of $\Gamma$).

Then, we prove the truthfulness and individual rationality properties as follows.

5.1 Truthfulness

To prove the truthfulness of the mechanisms, we first give a lemma on the winning set.

**Lemma 1.** $\Phi_t = \Phi^*_t$, for $\forall t \in [0, q-1]$.

**Proof.** Note that $\Phi$ and $\Phi^*_t$ are the winning sets selected from $\Gamma$ and $\Gamma^*_t$ respectively through the same greedy strategy. Since $\beta^*_k$ is the winner of the qth round, it must have not been selected as a winning bid before the qth round. Therefore, although $\Gamma = \Gamma^*_t \cup \{\beta^*_k\}$, the winning bids selected from $\Gamma$ and $\Gamma^*_t$ before the qth round are the same. Hence, the lemma holds.

Second, we prove that our mechanisms are bid-monotonic and payment-critical, which implies that the incentive mechanisms are truthful.

**Lemma 2.** Algorithm 1 is bid-monotonic.

**Proof.** Since $\beta^*_k$ wins in the qth round of iteration of Algorithm 1, the MPC of $\beta^*_k$, i.e., $U_{\beta^*_k}(\Phi_{q-1})/b^*_k$, is the maximal in this round, and it is no more than the MPC values of all bids in $\Phi_{q-1}$. Assume that $\beta^*_k$ reports a smaller cost $b^*_k$ ($< b^*_k$). As $U_{\beta^*_k}(\Phi_{q-1})/b^*_k < U_{\beta^*_k}(\Phi_{q-1})/b^*_k$, $\beta^*_k$ still wins or even before the qth round according to the greedy selection rule in Algorithm 1. Lemma 2 holds.

**Lemma 3.** The payments determined by Algorithm 2 for all winning bids are critical.

**Proof.** Assume that the winning bid $\beta^*_k$ reports a cost $b^*_k$ instead of $b^*_k$. To prove the criticality of the payment $p^*_k$, we need to prove that $\beta^*_k$ will fail if $b^*_k > p^*_k$, and that it will win if $b^*_k \leq p^*_k$.

Case 1: $b^*_k > p^*_k$. We consider the qth round of iteration in Algorithm 1 and derive that

$$\frac{U_{\beta^*_k}(\Phi_{q-1})}{b^*_k} = \frac{U_{\beta^*_k}(\Phi_{q-1})}{b^*_k} < \frac{U_{\beta^*_k}(\Phi^*_{q-1})}{p^*_k} \leq \frac{U_{\beta^*_k}(\Phi^*_{q-1})}{b^*_k},$$

where $\beta^*_k$ is a winning bid, so that $\Phi^*_q = \Phi^*_{q-1} \cup \{\beta^*_k\}$. In (17), the first equality depends on $\Phi_{q-1} = \Phi^*_{q-1}$ in Lemma 1 and the last inequality depends on $p^*_k \geq b^*_k \cdot U_{\beta^*_k}(\Phi^*_{q-1})/

$U_{\beta^*_k}(\Phi_{q-1})$ according to (6). Hence, $\beta^*_k$ is selected as the winner instead of $\beta^*_k$ in the qth round of iteration. Moreover, $\Phi_q = \Phi_{q-1} \cup \{\beta^*_k\} = \Phi_q^*$. Repeating the above analysis, we can see that $\beta^*_k$ will fail in all rounds of iteration of Algorithm 1.

Case 2: $b^*_k \leq p^*_k$. WLOG, assume that Algorithm 1 runs when the input bid set is $\Gamma = \beta^*_k$, which is exactly the process of computing the payment for $p^*_k$. According to (6), we assume that $\beta^*_k = b^*_k \cdot U_{\beta^*_k}(\Phi^*_{q-1})/U_{\beta^*_k}(\Phi_{q-1})$, where $\beta^*_k$ is the winner in the qth round of iteration of this process. Now, we run Algorithm 1 again with the input bid set $\Gamma$. We discuss two subcases of this new process: 1) $\beta^*_k$ wins before the qth round; 2) $\beta^*_k$ does not win before the qth round. In the qth round of iteration

$$\frac{U_{\beta^*_k}(\Phi^*_{q-1})}{b^*_k} \geq \frac{U_{\beta^*_k}(\Phi_{q-1})}{p^*_k} \geq \frac{U_{\beta^*_k}(\Phi^*_{q-1})}{b^*_k}.$$  

Therefore, $\beta^*_k$ wins in this round. Synthesizing both subcases, $\beta^*_k$ wins when $b^*_k \leq p^*_k$.

In conclusion, the payments for all winning bids are critical, and the lemma holds.

**Theorem 4.** Our basic incentive mechanism consisting of Algorithms 1 and 2 is truthful.

**Proof.** According to Myerson’s theorem[41], our incentive mechanism is truthful since the winning-bid selection rule is monotone (i.e., Lemma 2) and each winning bid is paid with a critical value (i.e., Lemma 3).

**Theorem 5.** Our QoD-aware incentive mechanism consisting of Algorithms 3 and 4 is truthful.

**Proof.** Despite of the utility function and the termination threshold, the winning-bid selection methods in Algorithms 1 and 3 are the same. Hence, Algorithm 3 is also bid-monotonic according to Lemma 2. The payment determination criteria and methods are also the same in Algorithms 2 and 4. Thus the payments determined by Algorithm 4 for all winning bids are also critical according to Lemma 3. Then, based on Myerson’s theorem[41], our QoD-aware incentive mechanism is also truthful.

5.2 Individual Rationality

In this section, we prove the individual rationality of the incentive mechanisms and have the following theorems:

**Theorem 6.** Our basic incentive mechanism consisting of Algorithms 1 and 2 is individually rational.

**Proof.** We assume that $\beta^*_k$ wins in the qth round of iteration of the inner loop (Steps 4-9) in Algorithm 2. Since $\beta^*_k$ is the winner in the qth round of Algorithm 1

$$\frac{U_{\beta^*_k}(\Phi_{q-1})}{b^*_k} \leq \frac{U_{\beta^*_k}(\Phi_{q-1})}{b^*_k}.$$  

According to (6) and Lemma 1

$$b^*_k \leq \frac{U_{\beta^*_k}(\Phi_{q-1})}{b^*_k} \leq \frac{U_{\beta^*_k}(\Phi^*_{q-1})}{p^*_k} \leq b^*_k \leq p^*_k.$$  


As each vehicle reports its real cost in the truthful incentive mechanism, we get $c_k^i = b_k^i$. Hence, $p_k^i \geq c_k^i$, indicating that the payoff of $p_k^i$ is non-negative. The theorem holds.

**Theorem 7.** Our QoD-aware incentive mechanism consisting of Algorithms 3 and 4 is individually rational.

**Proof.** Since the payment for each winning bid determined by Algorithm 4 is the maximum critical value in (16), the QoD-aware incentive mechanism is also individually rational according to Theorem 6. The theorem holds.

### 5.3 Computational Efficiency

We prove the computational efficiency of the two mechanisms in the following theorem.

**Theorem 8.** The basic and QoD-aware incentive mechanisms have a polynomial-time computational complexity.

**Proof.** 1) The basic incentive mechanism is composed of Algorithms 1 and 2. The computation overhead of Algorithm 1 is dominated by Step 3, which can be denoted by $O(|S|^{2}||\Gamma^{2}|)$. Since Algorithm 1 loops at most $|\Gamma|$ times, its computational complexity is denoted by $O(|S|^{2}||\Gamma^{2}|)$.

2) The QoD-aware incentive mechanism consists of Algorithms 3 and 4. The computational complexity of Algorithm 3 is the same as that of Algorithm 1, i.e., $O(|\Gamma|^{2})$. Meanwhile, similar to Algorithm 2, the computational complexity of Algorithm 4 is denoted as $O(|\Gamma|^{2})$.

Therefore, the theorem holds.

### 5.4 Approximation Ratio Analysis

In this section, we first analyze the approximation ratio of Algorithm 1, followed by Algorithm 3.

#### 5.4.1 Approximation Ratio of Algorithm 1

To figure out the approximation ratio of Algorithm 1, we first analyze the properties of the utility function $U(\Phi)$ and the optimization objective function $C(\Phi)$. For simplicity of description, we define two notations

$$
\pi(i|\Phi) \triangleq \prod_{\Phi \in S_i \in S^j} (1 - q_i^j),
$$

$$
U(i|\Phi) \triangleq \min\{p_i^r, \eta\}.
$$

In addition, we consider two arbitrary bid sets, $S_1$ and $S_2$, $S_1 \subseteq S_2 \subseteq \Gamma$ and a bid $p_k^i \in (\Gamma - S_2)$. Then, we have:

**Lemma 4.** $U(\Phi)$ is an increasing function.

**Proof.** According to the decreasing property of $\pi(i|\Phi)$, $p_i^r$ is increasing. Therefore, $p_i^{r_1} \leq p_i^{r_2}$ and $U(i|\Phi_1) \leq U(i|\Phi_2)$ for $\forall \phi \in S$. Then $U(\Phi_1) \leq U(\Phi_2)$ when $\Phi_1 \subseteq \Phi_2$, which implies that $U(\Phi)$ is increasing.

**Lemma 5.** $U(\Phi)$ is submodular.

**Proof.** WLOG, we assume that for two arbitrary bid sets $X$ and $Y$, $X \subseteq \Gamma$ and $Y \subseteq \Gamma$. For $\forall \phi_i \in S$, we have the following conclusions:

1. $\pi(i|X \cup Y) \leq \pi(i|X), \pi(i|Y) \leq \pi(i|X \cap Y) \leq 1$;
2. $\rho_i^{X/Y} \leq \rho_i^X, \rho_i^Y \leq \rho_i^{X/Y} \leq 1$.

Based on these, we can get that

$$
\rho_i^X + \rho_i^Y - \rho_i^{X/Y} = \pi(i|X \cup Y) + \pi(i|X \cap Y) - \pi(i|X) - \pi(i|Y) \\
= \pi(i|X \cap Y)(1 - \pi(i|X - Y))(1 - \pi(i|Y - X)) \geq 0.
$$

Hence

$$
\rho_i^X + \rho_i^Y \geq \rho_i^{X/Y}.
$$

Now to prove that $U(\Phi)$ is submodular, we consider the relationships between $\rho_i^{X/Y}$, $\rho_i^X$, $\rho_i^Y$, and $\eta$, which can be divided into the following four cases:

**Case 1:** $\rho_i^{X/Y} \leq \eta$. Then

$$
U(i|X) + U(i|Y) - U(i|X \cup Y) - U(i|X \cap Y) = \rho_i^X + \rho_i^Y - (\rho_i^{X/Y} + \rho_i^{X/Y}) \geq 0.
$$

Consequently

$$
U(i|X) + U(i|Y) \geq U(i|X \cup Y) + U(i|X \cap Y),
$$

when $\rho_i^{X/Y} \leq \eta$.

**Case 2:** $\rho_i^{X/Y} > \eta$ while $\rho_i^X, \rho_i^Y \leq \eta$. Then

$$
U(i|X) + U(i|Y) - U(i|X \cup Y) - U(i|X \cap Y) > \rho_i^X + \rho_i^Y - (\rho_i^{X/Y} + \rho_i^{X/Y}) \geq 0.
$$

Hence, (22) also holds in this case.

**Case 3:** $\rho_i^X > \eta$ or $\rho_i^Y > \eta$ while $\rho_i^{X/Y} \leq \eta$. Consider $\rho_i^X$ and $\rho_i^Y$:

1. If one of them is larger than $\eta$, (22) holds because both $\rho_i^X$ and $\rho_i^Y$ are no less than $\rho_i^{X/Y}$;
2. If they are both larger than $\eta$, (22) is still correct because $\rho_i^{X/Y} \leq \eta$.

Therefore, (22) is true in this case.

**Case 4:** $\rho_i^{X/Y} > \eta$. Now $U(i|X) + U(i|Y) - U(i|X \cup Y) - U(i|X \cap Y) = 0$, and (22) is valid.

In summary, (22) is valid in all cases. Hence, according to the above analysis and the fact that $U(\Phi) = \sum_{i=1}^{m} U(i|\Phi)$

$$
U(X) + U(Y) \geq U(X \cup Y) + U(X \cap Y),
$$

which indicates that $U(\Phi)$ is submodular.

**Theorem 9.** $U(\Phi)$ is a polymatroid function on $2^\Gamma$.

**Proof.** We have that $U(\Phi) = 0$ when $\Phi = \emptyset$. According to Lemmas 4 and 5, the theorem holds.

**Theorem 10.** $C(\Phi)$ is a polymatroid function on $2^\Gamma$.

**Proof.** According to $C(\Phi) = \sum_{p_k^i \in \Phi} c_k^i$, $C(\Phi)$ is an increasing function with $C(\emptyset) = 0$. As $C(\Phi_1 \cup \{p_k^i\}) - C(\Phi_1) = c_k^i \geq C(\Phi_2 \cup \{p_k^i\}) - C(\Phi_2)$, we have that $C(\Phi)$ is submodular. Therefore, $C(\Phi)$ is a polymatroid function on $2^\Gamma$. 

Suppose $\phi$ is the winning bid set after Algorithm 1 terminates in the $q$th round of iteration. Let $\theta_1 = \min\{\frac{C(\Phi_{q-1})}{m_{\phi_{q-1}}} | q = 1, 2, 3, \ldots, Q\}$, where $\beta_q$ is the winning bid in the $q$th round, $\theta_2 = \frac{C(\Phi)}{m_{\phi}}$, and $\theta = \max(\theta_1, \theta_2)$. Utilizing $\theta$, we derive a new utility function, $U'(\Phi) = \theta U(\Phi)$, and the new marginal contribution of $\beta_k \notin \Phi$, $U'_{\beta_k}(\Phi) = \theta U_{\beta_k}(\Phi)$. Moreover, we derive the following theorems:

**Theorem 11.** $U'(\Phi)$ is a polymatroid function on $2^\Phi$.

**Proof.** According to Theorem 9 and the fact that $\theta$ is a constant, the theorem holds. □

**Theorem 12.** The MCBS problem can be equivalently re-formalized as

$$\min\{C(\Phi)|U'(\Phi) = U'(\Gamma), \Phi \subseteq \Gamma\}.$$ (23)

**Proof.** On one hand, the constraint (3) are met when $U(\Phi) = mn(U'(\Phi) = \theta mn)$ according to Theorem 2. On the other hand, for $\forall \phi_i \in S$, $\min(\beta_i, \eta) = \eta$, and $U'(\Phi) = \theta mn$, if $\beta_i \geq \eta$, i.e., (3). That is to say, $U'(\Phi) = \theta mn$ is equivalent to the constraint (3), and we can replace (3) with $U'(\Phi) = \theta mn$. Since $U'(\Gamma) = \theta mn$, we can equivalently replace (3) with $U'(\Phi) = U'(\Gamma)$, the theorem holds. □

According to Theorems 10, 11, and 12, the MCBS problem can be deemed as a minimum submodular cover with submodular cost problem [46]. Additionally, we can derive a new greedy algorithm that shares the same solution as Algorithm 1 by replacing $U(\Phi), U_{\beta_k}(\Phi)$, and $mn$ in Algorithm 1 with $U'(\Phi), U'(\Phi), \phi_i \notin \Phi$, respectively. The derived algorithm also greedily selects the bid $\beta_i$, whose MPC, i.e., $U'(\Phi)/\beta_i$ is the maximal in each round of iteration. Then we can analyze the approximation ratio of the derived algorithm based on the theorem in [46]:

**Theorem 13.** If in each iteration of a greedy algorithm, the selected bid $\beta_i$ always satisfies that $U'(\Phi)/\beta_i \geq 1$, then the greedy solution is a $(1 + \ln(\frac{U'(\Gamma)}{\text{opt}}))$-approximation, where $U'$ is a polymatroid function on $2^\Phi$, opt is the cost of a minimum submodular cover, and $U'(\Gamma) \geq \text{opt}$. 

**Theorem 14.** The derived algorithm achieves the $(1 + \ln(\frac{\text{opt}}{\text{opt}}))$-approximation of the optimal social cost, where opt is the cost of the optimal solution to MCBS.

**Proof.** Note that $U'(\Gamma) = \theta U(\Gamma) = \theta mn$. Consequently, $U'(\Gamma) \geq mn\theta_2 = C(\Phi) \geq \text{opt}$. Additionally, $U'_{\beta_i}(\Phi)/\beta_i \geq U'_{\beta_i}(\Phi)/\beta_i \geq 1$ for $\forall \beta_i \in \Phi$ according to the definition of $\theta_1$. Hence, the approximation ratio of the derived algorithm is $(1 + \ln(\frac{\text{opt}}{\text{opt}}))$ based on Theorems 10, 11, 12, and 13. □

**Theorem 15.** Algorithm 1 achieves the $(1 + \ln(\frac{\text{opt}}{\text{opt}}))$-approximation of the optimal social cost, where opt is the cost of the optimal solution to the MCBS problem.

**Proof.** Since the derived algorithm shares the same solution with Algorithm 1, Algorithm 1 approximates the optimal solution of MCBS within a factor of $(1 + \ln(\frac{\text{opt}}{\text{opt}}))$ according to Theorem 14. □

**5.4.2 Approximation Ratio of Algorithm 3**

After getting the approximation ratio of Algorithm 1, we analyze Algorithm 3. We also define a notation

$$F(i(\Phi)) \triangleq \min\{\xi_i, \xi_i\}.$$ 

Consider two arbitrary bid sets, $\Phi_1$ and $\Phi_2$, $\Phi_1 \subseteq \Phi_2 \subseteq \Gamma$, and a bid $\beta_k \in (\Gamma - \Phi_2)$. Then, we have:

**Lemma 6.** $F(\Phi)$ is an increasing function.

**Proof.** Considering the definition in (7), we get $\mu_i(\Phi_1) \leq \mu_i(\Phi_2)$ and $F(i(\Phi_1)) \leq F(i(\Phi_2))$ for $\forall s_i \in S$. Then, $F(\Phi_1) \leq F(\Phi_2)$ if $\Phi_1 \subseteq \Phi_2$, which implies that $F(\Phi)$ is increasing. □

**Lemma 7.** $F(\Phi)$ is a submodular function.

**Proof.** WLOG, we assume that for two arbitrary bid sets $X$ and $Y$, $X \subseteq \Gamma$, and $Y \subseteq \Gamma$. For $\forall s_i \in S$, we have the following conclusions:

1. $\mu_i^{X \cap Y} \leq \mu_i^X$, $\mu_i^X \leq \mu_i^{X \cup Y}$;
2. $\mu_i^{X \cap Y} + \mu_i^Y = \mu_i^{X \cup Y} + \mu_i^{X \cap Y}$.

Similar to the proof of Lemma 5, we have that $F(i(X) + F(i(Y)) \geq F(i(X \cup Y)) + F(i(X \cap Y))$ is valid in the following four cases:

1. $\mu_i^{X \cap Y} \leq \xi$;
2. $\mu_i^{X \cup Y} > \xi$ while $\mu_i^X, \mu_i^Y \leq \xi$;
3. $\mu_i^X > \xi_i$ or $\mu_i^Y > \xi_i$ while $\mu_i^{X \cap Y} \leq \xi_i$;
4. $\mu_i^{X \cap Y} > \xi_i$.

Hence, $F(X) + F(Y) \geq F(X \cup Y) + F(X \cap Y)$, and the theorem holds. □

**Lemma 8.** $G(\Phi)$ is a polymatroid function on $2^\Phi$.

**Proof.** Since both $F(\Phi)$ and $U(\Phi)$ are submodular based on Lemmas 5 and 7, $G(\Phi)$ is submodular according to (14). Meanwhile, $G(\Phi)$ is also an increasing function because of the increasing property of $F(\Phi)$ and $U(\Phi)$. As $G(\emptyset) = 0$, the theorem holds. □

Suppose $\phi$ is the winning bid set after Algorithm 3 terminates in the $q$th round of iteration. Let $\alpha_1 = \min\{\frac{G(\phi_{q-1})}{\beta_i} | q = 1, 2, 3, \ldots, Q\}$, where $\beta_q$ is the winning bid in the $q$th round, $\alpha_2 = \frac{C(\Phi)}{m_{\phi} \sum_{i=1}^{\Phi} \xi_i}$, and $\alpha = \max\{\frac{1}{\alpha_1}, \alpha_2\}$. Utilizing $\alpha$, we derive a new utility function, $G'(\Phi) = aG(\Phi)$, and a new marginal contribution of $\beta_k \notin \Phi$, $G'_{\beta_k}(\Phi) = aG_{\beta_k}(\Phi)$. Similar to the proofs in Section 5.4.1, we can have the following two theorems:

**Theorem 16.** $G'(\Phi)$ is a polymatroid function on $2^\Phi$.

**Proof.** According to Lemma 8 and the formula $G'(\Phi) = aG(\Phi)$ in which $a$ is a constant, the theorem holds. □

**Theorem 17.** The QoD-MCBS problem can be equivalently reformalized as

$$\min\{C(\Phi)|G'(\Phi) = G'(\Gamma), \Phi \subseteq \Gamma\}.$$ 

**Proof.** The constraints (10) and (11) are simultaneously met when $U(\Phi) = mn$ and $F(\Phi) = \sum_{i=1}^{\Phi} \xi_i$, that is, $G(\Phi) = mn + \sum_{i=1}^{\Phi} \xi_i$ and $G'(\Phi) = a(mn + \sum_{i=1}^{\Phi} \xi_i)$. On the other
hand, if \( G'(\Phi) = a(m\eta + \sum_{i=1}^{m} \xi_i) \), i.e., \( U(\Phi) = m\eta \) and \( F(\Phi) = \sum_{i=1}^{m} \xi_i \), for \( \forall s_i \in S \), we have \( \mu^k_i \geq \eta (10) \) and \( \mu^k_i \geq \xi_i \) (11). This indicates that \( G'(\Phi) = a(m\eta + \sum_{i=1}^{m} \xi_i) \) is equivalent to the constraints (10) and (11). Based on \( G'(\Gamma) = a(m\eta + \sum_{i=1}^{m} \xi_i) \), we can equivalently replace (10) and (11) with \( G'(\Phi) = G'(\Gamma) \). The theorem holds. \( \square \)

Based on Theorems 10, 16, and 17, QoD-MCBS can also be deemed a minimum submodular cover with submodular cost problem [46]. Similar to Section 5.4.1, when replacing \( G(\Phi) \), \( G_{jk}(\Phi) \), and \( m\eta + \sum_{i=1}^{m} \xi_i \) in Algorithm 3 with \( G'(\Phi) \), \( G_{jk}'(\Phi) \), and \( a(m\eta + \sum_{i=1}^{m} \xi_i) \) respectively, we can get a new algorithm which shares the same solution with Algorithm 3 and has the approximation ratio of \( (1 + \ln \frac{a(m\eta + \sum_{i=1}^{m} \xi_i)}{opt}) \) where \( opt \) is the cost of the optimal solution to the QoD-MCBS problem. Then, we have another theorem:

**Theorem 18.** Algorithm 3 achieves the \((1 + \ln \frac{a(m\eta + \sum_{i=1}^{m} \xi_i)}{opt})\) approximation of the optimal social cost, where \( opt \) is the cost of the optimal solution to the QoD-MCBS problem.

**Proof.** Since the derived algorithm of Algorithm 3 shares the same solution to the QoD-MCBS problem, Algorithm 3 approximates the optimal solution of QoD-MCBS with a factor of \((1 + \ln \frac{a(m\eta + \sum_{i=1}^{m} \xi_i)}{opt})\). The theorem holds. \( \square \)

### 6 Evaluation

We conduct extensive simulations to evaluate the performances of the proposed incentive mechanisms. The trace that we used, the simulation settings, the metrics, and the results are presented as follows.

#### 6.1 The Trace and Settings

We adopt the widely-used trace in [14] which contains the GPS coordinates of approximately 320 taxi cabs collected over 30 days in Rome, Italy. All of the taxi cabs in the trace move along different streets in Rome day after day. In our simulations, we select 50 main streets from the trace, as illustrated in Fig. 5. In the selected streets, we randomly deploy \{100, 200, 300, 400\} sensing tasks. Furthermore, we choose 316 vehicles from the trace for our simulations by excluding those vehicles that visit the selected streets with low frequency.

In our simulations, we select 30 days’ records of GPS coordinates for the chosen vehicles. We divide each day into two equal-length sensing periods, i.e., \([0, 12]\) and \([12, 24]\), and thus divide 30 days into 60 periods. Moreover, we let the sensing tasks be distributed in the same period and let the \( D \) of the tasks be set as 12 hours. The probability of each vehicle visiting a street (i.e., the probability of a trajectory) is estimated as follows. First, we determine whether a vehicle has visited a street in a sensing period by testing whether the coordinates of this vehicle located in the street during this period. Then, we count the number of sensing periods during which a vehicle has visited a street. This number divided by 60 is viewed as the average probability of the vehicle visiting the street. Additionally, the real costs of bids are generated based on three distributions, i.e., uniform distribution (UNM), normal distribution (NORM) and exponential distribution (EXP). Each simulation is conducted with the three distributions. All simulations in this section are performed in JAVA on a Windows PC with a 3.2 GHz Intel Core i5 CPU and 8GB memory.

#### 6.2 The Evaluation Metrics, Methodology and Results

To evaluate the performance of our mechanisms, we use the following metrics: number of winning bids, successful performing ratio, social cost, overpayment ratio, truthfulness, and individual rationality. The Number of Winning Bids (NWB) measures the scale of crowdsensing. The Successful Performing Ratio (SPR) is the ratio of the number of the successfully performed crowdsensing tasks and the number of all tasks. The overpayment ratio is defined as

\[
\lambda = \frac{(P - C(\Phi))}{C(\Phi)},
\]

where \( P \) is the total payment and \( C(\Phi) \) is the total cost. It measures the cost paid by the platform user to induce the truthfulness of all vehicles. Truthfulness is the property ensuring that no bidder can improve his or her payoff by submitting a different cost from the real one. Individual rationality is the property which ensures that the payoff of each bid is non-negative.

The default settings of our simulations are shown in Table 2. We set the default QoD values \( Q \) and \( \mathcal{R} \) as 0, which means that we conduct simulations for the basic incentive mechanism. The results are shown as follows.

**Number of winning bids:** We depict the evaluation on the NWB in Figs. 6, 10 and 16. We increase the number of tasks to verify the impact on the NWB, and the results show that the NWB will increase since we need more vehicles and bids to perform more tasks. Additionally, more bids are needed to meet constraint (3) when \( \eta \) increases, as shown in Fig. 16. However, when the average of the real cost increases, the NWB does not change much (see Fig. 10). This is because we have to keep the joint probability no less than \( \eta \) no matter how much it will cost. If \( \mathcal{C} \) expands (i.e., the average cost increases), the MPCs of all bids will decrease, leading to results with fewer changes. We also find that the UNM needs the most bids and that the EXP needs the fewest.

**Successful performing ratio:** Fig. 7 plots the successful performing ratio when the number of tasks (i.e., \( m \)) changes from 100 to 400. With the increase of \( m \), more bids will be selected, spontaneously leading to the increase of successful performing ratios. Figs. 11 and 17 plot the SPR when the cost range \( \mathcal{C} \) and the threshold \( \eta \) increase, respectively. The SPR does not change much in Fig. 11, since the number of the winning bids changes little when \( \mathcal{C} \) increases (see Fig. 10). The results in Fig. 17 show that the SPR is
slightly larger than $\eta$. These results are reasonable, since the SPR is in the constraint (3). Additionally, the distribution of costs has little influence on the SPR based on the above figures. This is because we have guaranteed the successful performing probability of each task in the bid selection process.

**Social cost:** We verify the performance of the social cost by changing $m$, $C$, and $h$, and present the results in Figs. 8, 12, and 18, respectively. These figures show that the social cost will increase if we increase $m$, $C$, and $h$ separately. This is because with the increase of $m$ and $h$, the mechanism will select more bids to meet the constraint (3). If we increase the average cost (i.e., expand $C$), the social cost will spontaneously increase. Additionally, the results in the three figures show that the social cost of UNM is larger than those of NORM and EXP. This is because the UNM produces greater costs than the others do. Since NORM produces more middle-range costs, the social cost of NORM is always less than that of the other two distributions.

**Overpayment ratio:** We depict the evaluation on overpayment ratio $\lambda$ in Figs. 9, 13, and 19. The results show that $\lambda$ is always less than 0.6, which means that the platform user does not have to pay much extra money to induce truthfulness. If we increase $m$ and $h$ in Figs. 9 and 19, $\lambda$ will increase in accordance. This is because more vehicles and bids will be recruited and the increments of the payments are greater than those of the costs. We find that when $C$ increases in Fig. 13, $\lambda$ will also increase since the truthfulness guarantees that each payment is larger than the related cost and that the increments of the payments are larger than those of the costs.

**Truthfulness and individual rationality:** To verify the truthfulness of our incentive mechanism, we randomly pick a winning bid, change its claimed cost, and recalculate the related payments as well as the payoffs. The results are illustrated in Fig. 14. The payment is 25.2, and the real cost

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tasks $m$</td>
<td>100</td>
</tr>
<tr>
<td>Cost range $C$</td>
<td>[10, 20]</td>
</tr>
<tr>
<td>Threshold of successful performing probability $\eta$</td>
<td>0.6</td>
</tr>
<tr>
<td>Variance of NORM $\sigma$</td>
<td>10</td>
</tr>
<tr>
<td>Quality Range of tasks $Q$</td>
<td>[0, 0]</td>
</tr>
<tr>
<td>Quality Range of vehicles $R$</td>
<td>[0, 0]</td>
</tr>
</tbody>
</table>

**Fig. 6. NWB versus $m$.**

**Fig. 7. SPR versus $m$.**

**Fig. 8. Social cost versus $m$.**

**Fig. 9. Overpayment ratio versus $m$.**

**Fig. 10. NWB versus $C$.**

**Fig. 11. SPR versus $C$.**

**Fig. 12. Social cost versus $C$.**

**Fig. 13. Overpayment ratio versus $C$.**

**Fig. 14. Truthfulness and individual rationality.**
is 16. Then, the payoff is 9.2. We find that the payoff remains unchanged if the claimed cost is no more than the payment. Moreover, when the claimed cost is larger than the critical value 25.2, the payoff becomes zero. We also verify the individual rationality by comparing the real EXP cost of each bid and the related payment, which is calculated when $m = 100$, $\eta = 0.6$, and $C = [10, 30]$. Then we find that each payment is greater than the related cost (see Fig. 15).

We also conduct simulations under the condition that $Q$ and $R$ are not 0. More specifically, the range of $Q$ is set as $[10, 15], [10, 20], [10, 25]$, the range of $R$ is set as $[10, 20]$, and the other parameters are set as the default values. The results are shown in Figs. 20, 21, 22, and 23. Compared to the results of the basic mechanism in Figs. 6, 8, and 9 when $m = 100$, the results in Figs. 20, 22, and 23 show that more winning bids are needed, more social costs are spent, and higher overpayment ratios are produced since we have to meet the QoD constraint (11) besides the probability constraint. We also find that some SPRs in Fig. 21 are less than the related value of $\eta$, e.g., SPR of UNM is 0.51 when $\eta = 0.6$ and $Q = [10, 15]$. This is because if the QoD constraints of some tasks are not met, we deem that these tasks are incomplete, even if they have been performed by a few vehicles.

7 RELATED WORK

There have been a few works [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [28], [29], [30], [31], [32], [33] on the incentive mechanism design for crowdsensing/crowdsourcing, which can be divided into two categories: offline and online.

Offline Incentive Mechanism. In an offline scenario, the crowdsourcer/platform is aware of all users, and no users will come/leave during the process. In [15], Yang et al. proposed two types of mechanisms to maximize the utility of the corwdsource: CCM and UCM. They have shown that their mechanisms perform well by proving the unique Stackelberg Equilibrium of IMCC in CCM and giving the approximation ratio of IMCU in UCM. Feng
et al. solve the problem of location-aware collaborative sensing in mobile crowdsourcing in [24] by proposing TRAC. They formulate WBDP as a linear set cover problem with minimum social cost and present an approximation algorithm that can approach the optimum solution within a factor of $1 + \ln(n)$. Zhang et al. [18] consider a scenario where a crowdsourcing job requires the collective effort of multiple participants. They propose incentive mechanisms for three models: SS-Model, SM-Model, and MM-Model. In [25], Luo et al. propose a different incentive mechanism based on an all-pay auction model, in which every bidder must pay for his bid regardless of whether he wins or loses the auction.

In addition, some works focus on the mechanism design for special problems, e.g., image labeling and noise sensing. In [19], Zhang et al. study the problem that how to stimulate workers to perform binary labeling tasks while maximizing the utility of the platform and meeting budget constraints. They profile the quality of the workers with Beta distribution functions and calculate them by Bayes’ rule. In [20], [21], the authors incorporate the consideration of data quality into the design of incentive mechanism for noise sensing. Jin et al. [21] prefer to maximize social welfare while ensuring that the quality of each task is no less than a given threshold. Peng et al. [20] propose a mechanism to maximize the platform’s profit. They estimate the quality of sensing data via the well-known expectation maximization algorithm and quantify the participants’ contributions through information uncertainty reduction.

**Online Incentive Mechanism.** Different from the offline scenario, in an online scenario the statuses of users are dynamic. That is to say, the users and tasks come and leave randomly, and the crowdsourcer/platform only sees part of its users in one time interval. Considering this scenario, Zhao et al. [16] propose two online mechanisms, OMZ and OMG, that adopt a multiple-stage sampling acceptance process to maximize the value of the crowdsourcer without sacrificing utility. To satisfy the budget constraint, their mechanisms utilize the density threshold to filter out inapposite users. Moreover, both OMZ and OMG are competitive with the offline scenario. Zhang et al. [22] design three online, reverse-auction-based incentive mechanisms: TBA, TOIM, and TOIM-AD. TBA uses the first batch of bidders as a sample and makes decisions on the second batch of bidders. It is designed to pursue the maximization of the platform utility. TOIM is a truthful online mechanism which is highly competitive with the optimal solution in the zero arrival-departure model. TOIM-AD extends TOIM to the non-zero arrival-departure model. In [23], Zhu et al. first propose an offline mechanism where the platform knows the active time and the arrival time of each task at the beginning of the crowdsourcing. Based on this offline mechanism, they propose the online social welfare maximization mechanism which divides time into slots, finds the near-optimal solution, and decides the payments in each slot. The task allocation algorithm in the online mechanism achieves a constant competitive ratio of $\frac{1}{3}$. In [17], Wei et al. consider stimulating both service users and providers to participate in mobile crowdsourcing and model the interactions as an online double auction. They propose an expressive general framework that is suitable for different price schedules.

### 8 Conclusion

In this paper, we first design a truthful incentive mechanism for vehicle-based, nondeterministic crowdsensing, where the sensing tasks are performed with different probabilities and the probability of each task being successfully performed is no less than a threshold. After considering a more complex scenario where the platform has a requirement on the QoD, we also design a QoD-aware incentive mechanism. Through rigorous theoretical analysis, we prove that both incentive mechanisms have the properties of truthfulness, individual rationality, computational efficiency and social cost efficiency. Finally, we conduct lots of simulations on a real trace to verify the significant performances of our incentive mechanisms.

### Acknowledgments

This paper is an extended version of the conference paper [1] published in IEEE/ACM IWQoS 2016. This research was supported in part by the National Natural Science Foundation of China (NSFC) (Grant No. 61572457, 61379132, 61502261, 61303206, 61572342), US National Natural Science Foundation grants CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, CNS 1156574, the Natural Science Foundation of Jiangsu Province in China (Grant No. BK20131174, BK2009150), and the Anhui Province Guidance Funds for Quantum Communication and Quantum Computers.
Guoju Gao received the BS degree in information security from the University of Science and Technology of Beijing, Beijing, China, in 2014. He is currently working toward the PhD degree in computer science and technology with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include mobile data offloading, vehicle ad hoc networks, cloud computing, and auction mechanisms.

Mingjun Xiao received the PhD degree from the University of Science and Technology of China (USTC), in 2004. He is an associate professor with the School of Computer Science and Technology, University of Science and Technology of China. In 2012, he was a visiting scholar with Temple University, under the supervision of Dr. Jie Wu. He is a TPC member of many conferences, including IEEE INFOCOM 2018, IEEE ICDCS 2015, ACM Mobihoc 2014, etc, and has served as a reviewer for many journal papers. His main research interests include mobile crowdsensing, mobile social networks, and vehicular ad hoc networks. He has published more than 50 papers in refereed journals and conferences, including the IEEE/ACM Transactions on Networking, the IEEE Transactions on Mobile Computing, the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, INFOCOM, etc. He is a member of the IEEE.

Liusheng Huang received the MS degree in computer science from the University of Science and Technology of China, Hefei, P.R. China, in 1988. He is currently a professor and PhD supervisor with the School of Computer Science and Technology, University of Science and Technology of China. He has published six books and more than 200 papers. His research interests include the areas of wireless networks, information security, VANET, and algorithms. He is a member of the IEEE.

Chang Hu received the BS degree in computer science and technology from Sichuan University, Chengdu, P.R. China, in 2014, and the MS degree in computer science from the University of Science and Technology of China, Hefei, P.R. China, in 2017. His research interests include the areas of mobile crowdsensing, vehicular ad hoc networks, privacy preserving, and auction mechanisms.

Jie Wu is the director of the Center for Networked Computing and Laura H. Carnell professor with Temple University. He also serves as the director of international affairs at the college of Science and Technology. He served as chair of the Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and associate vice provost for international affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director with the National Science Foundation and was a distinguished professor with Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. He regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including the IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. He was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, IEEE ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society distinguished visitor, ACM distinguished speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). He is a CCF distinguished speaker and a fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.