Blockchain based Mobile Crowd Sensing for Reliable Data Sharing in IoT Systems

Zhenni Feng
Donghua University,China
fzn@dhu.edu.cn

Junchang Chen
Donghua University, China
2202500@mail.dhu.edu.cn

***Abstract*—Mobile crowd sensing (MCS) systems fully take advantage of wisdom of the crowd and benefit from low deployment cost and widely spatial coverage. We propose a practical decentralized MCS system based on a distributed auction process and the blockchain system, achieving the optimal social efficiency and individual rationality without disclosing privacy.**

***Index Terms*—decentralized mobile crowd sensing, blockchain, untrustworthy participants, optimal distributed auction.**

I. Introduction

Mobile crowd sensing (MCS) has become a promising paradigm [1], [2] to fully take advantage of ubiquitous sensors embedded at mobile devices, such as smartphones, wearables and unmanned vehicles. In a typical MCS system, three kinds of roles of requesters, workers, and a central platform form a triangular architecture. Mobile devices are regarded as *workers* and people who request sensing service or need sensory data are regarded as *requesters*. Requesters recruit workers to perform *sensing tasks* and offer *payment* to workers.

However, introduction of a central platform is problematic due to security issues of privacy breach and distributed-denial-of-service (DDoS) attack. Furthermore, either requesters or workers probably behave in an untrustworthy way, since they are owned by different organizations or private individuals.

There exist several technical challenges for *constructing an efficient MCS system with untrustworthy participants in a decentralized manner for data sharing*. *Firstly*, all participants are selfish which expect as much profit as possible. *Secondly*, both the cost information of workers and the valuation of sensing tasks for requesters are private. *Thirdly*, participants are usually untrustworthy, *e.g.*, requesters may deliberately deny receiving of solution or refuse to pay for workers.

Existing studies provide many incentive mechanisms for MCS systems [3]–[5], which have research the problem of finding an optimal match between tasks and workers to achieve optimal system efficiency (*e.g.*, social welfare) with a few constraints of privacy protection, budget, data quality etc. However, most existing approaches assume the existence of a central and neutral platform which manages and supervises the entire interactive process between workers and requesters.

To this end, we propose an efficient and practical decentralized MCS system on the basis of a distributed auction process and the blockchain system. *On the one hand*, we propose a distributed auction process which achieves optimal social profit iteratively and offers proper incentive to participants without disclosing their privacy as well. *On the other hand*, we design a reliable data sharing protocol based on the blockchain system and smart contracts, thus avoiding or mitigating false-reporting and free-riding threats.

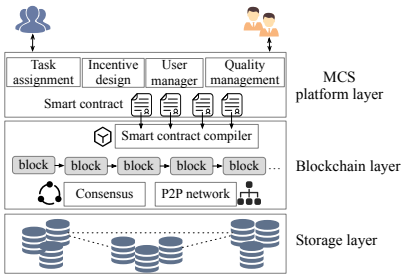

Fig. 1: The architecture of decentralized MCS systems.

We summarize main contributions of the paper as follows.

- We propose a decentralized MCS system considering untrustworthy and selfish participants, providing insights to implement practical and robust MCS applications.
- The distributed auction process is proved to converge at a stable state of optimal social profit, although participants only make simple strategies based on local information.
- Both theoretical analysis and simulations demonstrate that the proposed approach satisfies optimal social profit, convergence, individual rationality, privacy preserving.

II. Problem Formulation

A. Architecture of Decentralized MCS Systems

The architecture of a decentralized MCS system based on the blockchain system, as illustrated in Fig.1, has three layers of *storage layer*, *blockchain layer* and *MCS platform layer*. We concentrate on the design of the top layer since there are feasible solutions for underlying blockchain systems and distributed storage systems. There are two types of participants in decentralized MCS systems: *requesters* who post tasks and *workers* who perform tasks. A *task* usually has a series of requirements specified by its requester, *e.g.*, task category, geographical coverage, time limit, worker reliability and only workers who can satisfy all these requirements can be chosen.

B. Mathematical formulation

Both requesters and workers are *selfish* participants who try to maximize their utilities. The *utility* of a requester is net profit gained from completion of the task, which is the difference between the valuation v_k and the payment *payment*

Annex to ISBN 978-3-903176-39-3

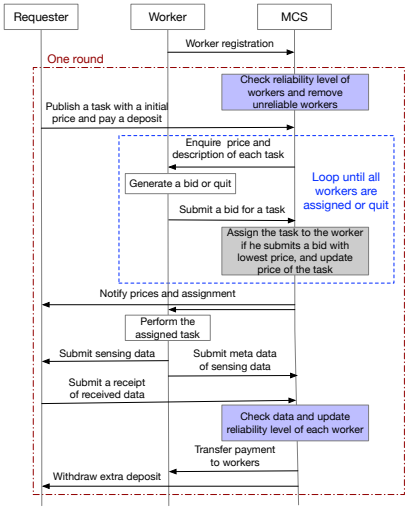

Fig. 2: The distributed auction process.

p_k to the worker. The utility of a worker w_j is net profit he gets from offering sensing service, which is exactly the difference between the payment p_k and his *cost* $c_{j,k}$ when task t_k is assigned to him. Both requesters and workers are *individually rational* since they would not accept an assignment except with nonnegative utility. Besides, both valuation and cost is *private* information which participants are reluctant to disclose.

The objective of designing decentralized MCS systems in Equation (1) is to achieve optimal system efficiency in spite of selfishness of participants. System efficiency is the summation of net profit generated by all assignments in a feasible assignment matrix. A feasible *assignment matrix* is defined as $\mathcal{A} = (a_{j,k}) \in \{1,0\}^{|\mathbf{N}|\times|\mathbf{M}|}$, where $a_{j,k}$ indicates whether a task t_k is assigned to a worker w_j. Furthermore, a feasible assignment matrix must satisfy the following two constraints: each worker can take at most one task, and each task can be assigned to at most one worker.

$$\begin{aligned} \mathcal{A}^* = \arg\max_{\mathcal{A}} S = \arg\max_{\mathcal{A}} \sum_{j\in\mathbf{N}}\sum_{k\in\mathbf{M}} a_{j,k}\cdot(v_k - c_{j,k}), \\ s.t.(i) \quad \sum_{k\in\mathbf{M}} a_{j,k} \le 1, \forall j \in \mathbf{N}, \\ (ii) \quad \sum_{j\in\mathbf{N}} a_{j,k} \le 1, \forall k \in \mathbf{M}, \\ (iii) \quad a_{j,k} \in \{1,0\}, \forall j \in \mathbf{N}, \forall k \in \mathbf{M}. \end{aligned} \tag{1}$$

However, existing algorithms such as the well-known Branch-and-Bound algorithm could be utilized in a decentralized scenario, since they are executed on complete information (including private information) and assume existence of a neutral, honest and nonprofitable platform.

III. Optimal Distributed Auction

A. Distributed Auction Process

A sequence map in Fig. 2 illustrates the complete process. Each participant registers in the platform and only *reliable* participants are allowed to participate in future rounds. During each round, three phases are listed as follow.

1) **Task publication**: Each requester broadcasts a task with its description and initial trading price along with an amount of deposit. Each worker decides whether he is capable of performing the task and estimates his cost.
2) **Bidding and task assignment**: An iterative procedure of bidding continues until there are no more unassigned workers. At the beginning, all workers are regarded as unassigned workers. At each iteration each unassigned worker queries current trading prices of all tasks and submits a bid with a strictly lower price for the "best" task. If the bidding price is lower than current trading price, the worker would be win the task replacing the former winner (if exists). Consequently, the former winner (if exists) is added into the set of unassigned workers and trading price of the task is decreased. Workers are removed from the set of unassigned workers if they are assigned or cannot get nonnegative utilities.
3) **Data transmission and payment**: A series of actions such as performing tasks, uploading solutions, quality management, paying to workers, management of deposit and reliability are carried out. Only meta data of the solution are recorded in the blockchain layer. Each requester submits a receipt indicating acceptance of the solution and then the worker is paid.

We call it distributed auction process since there is no central auctioneer to organize or supervise participants. Each worker determines his bids based on public information of current trading prices and local information of his cost, not necessarily disclosing private information.

Two smart contracts called *ParticipantManager* and *DistributedAuction* are deployed upon the blockchain layer. Related functionalities are reorganized and grouped into one smart contract due to expensive deploying cost. *ParticipantManager* is mainly responsible for auction initialization, deposit management, and updating worker reliability. *DistributedAuction* is responsible for organizing the trading process between requesters and workers, supporting two phases of task publication, bidding and task assignment.

B. Strategies of Participants

A requester sets the initial trading price to be his valuation minus a very small (randomly chosen) constant, instead of directly disclosing his valuation.

Each unassigned worker decides the target task (*i.e.*, "best" task) to bid as well as the bidding price repeatedly until he quits or has been assigned. Specifically, a worker must submit a bid to the target task with a strictly lower bidding price, otherwise his bid is invalid and would be ignored. We set the gap between the bidding price and the current trading, *i.e.*, $\Delta_{j,k^*} = p_{k^*} - c_{j,k^*} - (p_{k'} - c_{j,k'})^+ + \epsilon$, where k^* and k' are two tasks with the highest and the second highest net profit, p_{k^*} and $p_{k'}$ are the current trading prices of them, $\epsilon > 0$ is a very small positive constant, $(p_{k'} - c_{j,k'})^+$ is $\max\{p_{k'} - c_{j,k'}, 0\}$. ϵ is added to ensure $\Delta_{j,k^*} > 0$. Consequently, the bidding price is $b_{j,k^*} = p_{k^*} - \Delta_{j,k^*}$.

We demonstrate the desirable properties of the proposed algorithms theoretically from three different aspects. *Firstly*, we prove that the proposed algorithm converges in finite steps.

TABLE I: Default setting of parameters.

Notation	Meaning	Default value
$\|\mathbf{N}\|$,$\|\mathbf{M}\|$	No. of workers, tasks(requesters)	80, 24
ϵ	Minimum step size	$1/\max\{\|\mathbf{N}\|, \overline{m} \cdot \|\mathbf{M}\|\}$
	Range of costs	[0,10], integer
	Range of valuations	[10,20], integer

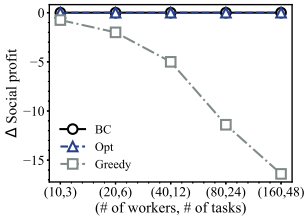

Fig. 3: Comparison on social profit.

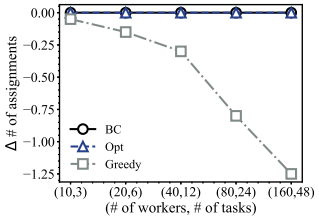

Fig. 4: Comparison on number of assignments.

Secondly, we further demonstrate that the proposed algorithm satisfies individual rationality which indicates every participant gets nonnegative utility. *Finally*, the final solution achieves optimal social profit. Due to space limit, we omit the proof.

C. Dealing with Untrustworthy Behavior

Two schemes of deposit and reliability are introduced to eliminate untrustworthy behavior. The deposit are prepaid by the requester when publishing a task and extra amount of deposit could be withdraw after he notifies the acceptance of solutions. Payment to workers are transferred automatically once solutions are accepted. Each registered worker has a reliability level indicating whether he is allowed to submit bids and to perform tasks. The reliability of a worker is increased if he finished a task according to task description and solutions were accepted; otherwise, the worker would not be paid and his reliability would be halved. A worker is unreliable if the reliability level drops below a predefined threshold.

IV. Performance Evaluation

We employ a local private blockchain as simulation environment via Ganache-cli. Smart contracts are written in Solidity using Remix. Parameters is randomly generated from a predefined range and default parameters are shown in Table I. Each group of simulations are executed 20 times and the average is computed. The method is evaluated based on the following performance metrics, *i.e.*, social profit, number of assignments, running time. The proposed method (BC) is compared with two centralized algorithms assuming the awareness of complete information. One is called Opt, which first transforms the original problem into the optimal matching problem and then employs Kuhn-Munkres algorithm to calculate the optimal social profit. The other is called Greedy which greedily selects the most profitable pair of an unassigned worker and an unassigned task until there is no available pairs.

In Fig. 3, the gap between Greedy and BC increases when the scale grows, indicating that it becomes more difficult for Greedy to approximate the optimal social profit. The y-axis of Δ social profit is the difference of social profit of each approach and Opt. Similarly, BC has the exactly the same performance on number of assignment as that of Opt in Fig. 4.

Comparison on running time are plotted in Fig. 5. Obviously, BC is less efficient that others especially in larger scale, since it is an iterative method. We compare performance of BC with different values of ϵ in Fig. 6. We find that default value of ϵ in Table I is small enough to generate the optimal solution.

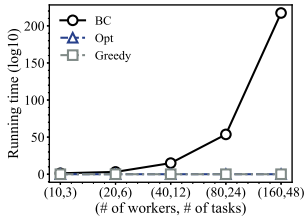

Fig. 5: Running time.

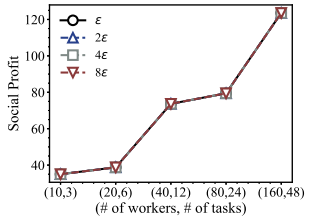

Fig. 6: Influence of ϵ.

V. Related Work

Existing papers [3]–[7] focus on designing incentive mechanism for MCS systems. They have investigated a lot of useful objectives, constraints or properties, *e.g.*, truthfulness, individual rationality, social welfare or revenue, budget, privacy preserving, data quality, location-awareness. CrowdBC [8] designs a MCS framework without any third-party trustful institutions. Another paper [9] designed a truthful incentive mechanism for distributed P2P applications. ZebraLancer [10] proposed a private and anonymous crowdsourcing system in despite of public property of the blockchain system. They could not be applied since we design a distributed auction process achieving optimal social efficiency as well as not disclosing private information.

VI. Conclusion

The proposed approach runs in a fully distributed manner where participants only makes simple decisions based on local information and public available information, providing valuable guidance to realization of MCS systems.

Acknowledgment

This research is partially supported by Shanghai Sailing Program (Grant No. 19YF1402200), Development of Shanghai Industrial Internet (Grant No. 2019-GYHLW-01004), and the Fundamental Research Funds for the Central Universities (Grant No. 2232021D-23).

References

[1] Y. Hu and R. Zhang, "Differentially-Private Incentive Mechanism for Crowdsourced Radio Environment Map Construction," in *proc. INFOCOM*, 2019.

[2] Y. Li, J. Sun, W. Huang, and X. Tian, "Detecting Anomaly in Large-scale Network using Mobile Crowdsourcing," in *proc. IEEE INFOCOM*, 2019.

[3] L. Jiang, X. Niu, J. Xu, D. Yang, and L. Xu, "Incentivizing the Workers for Truth Discovery in Crowdsourcing with Copiers," in *proc. IEEE ICDCS*, 2019.

[4] X. Wang, R. Jia, X. Tian, and X. Gan, "Dynamic Task Assignment in Crowdsensing with Location Awareness and Location Diversity," in *proc. IEEE INFOCOM*, 2018.

[5] S. Yang, K. Han, Z. Zheng, S. Tang, and F. Wu, "Towards Personalized Task Matching in Mobile Crowdsensing via Fine-Grained User Profiling," in *proc. IEEE INFOCOM*, 2018.

[6] Z. Duan, W. Li, X. Zheng, and Z. Cai, "Mutual-Preference Driven Truthful Auction Mechanism in Mobile Crowdsensing," in *proc. IEEE ICDCS*, 2019.

[7] Z. Wang, J. Hu, J. Zhao, D. Yang, H. Chen, and Q. Wang, "Pay On-Demand: Dynamic Incentive and Task Selection for Location-Dependent Mobile Crowdsensing Systems," in *proc. IEEE ICDCS*, 2018.

[8] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J. Liu, Y. Xiang, and R. Deng, "CrowdBC: A Blockchain-based Decentralized Framework for Crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1251–1266, 2019.

[9] Y. He, H. Li, X. Cheng, Y. Liu, C. Yang, and L. Sun, "A Blockchain based Truthful Incentive Mechanism for Distributed P2P Applications," *IEEE Access*, vol. 6, pp. 27 324–27 335, 2018.

[10] Y. Lu, Q. Tang, and G. Wang, "ZebraLancer: Private and Anonymous Crowdsourcing System atop Open Blockchain," in *proc. IEEE ICDCS*, 2018.