Contact Tracing Incentive for COVID-19 and Other Pandemic Diseases From a Crowdsourcing Perspective

Pengfei Wang, *Member, IEEE*, Chi Lin, *Member, IEEE*, Mohammad S. Obaidat, *Life Fellow, IEEE*, Zhen Yu, Ziqi Wei, and Qiang Zhang, *Member, IEEE*

***Abstract*—Governments of the world have invested a lot of manpower and material resources to combat COVID-19 this year. At this moment, the most efficient way that could stop the epidemic is to leverage the contact tracing system to monitor people's daily contact information and isolate the close contacts of COVID-19. However, the contact tracing data usually contains people's sensitive information that they do not want to share with the contact tracing system and government. Conversely, the contact tracing system could perform better when it obtains more detailed contact tracing data. In this article, we treat the process of collecting contact tracing data from a crowdsourcing perspective in order to motivate users to contribute more contact tracing data and propose the incentive algorithm named CovidCrowd. Different from previous works where they ask users to contribute their data voluntarily, the government offers some reward to users who upload their contact tracing data to reimburse the privacy and data processing cost. We formulate the problem as a Stackelberg game and show there exists a Nash equilibrium for any user given the fixed reward value. Then, CovidCrowd computes the optimal reward value which could maximize the utility of the system. Finally, we conduct a large-scale simulation with thousands of users and evaluation with real-world data set. Both results show that CovidCrowd outperforms the benchmarks, e.g., the user participating level is improved by at least 13.2% for all evaluation scenarios.**

***Index Terms*—Contact tracing, COVID-19, incentive algorithm, Nash equilibrium, pandemic diseases, utility maximization.**

Manuscript received October 14, 2020; revised December 8, 2020; accepted December 23, 2020. Date of publication January 4, 2021; date of current version October 22, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0910500; in part by the Fundamental Research Funds for the Central Universities under Grant DUT20RC(3)039, Grant DUT20RC(4)005, Grant DUT19JC39, Grant N182608004, and Grant N180101028; in part by the Liaoning Key Research and Development Program under Grant 2019JH210100030 and Grant 2020JH210100046; in part by the National Key Research and Development Projects under Grant 2019YFB1802600; in part by the Liaoning United Foundation under Grant U1908214; in part by the National Natural Science Foundation of China under Grant 62002045, Grant 62072094, Grant 61872052, Grant 61872073, and Grant 61672148; in part by the Natural Science Foundation of Liaoning Province under Grant 2019-MS-055; in part by the Liaoning Province Science and Technology Fund Project under Grant 2020MS086; in part by the CERNET Innovation Project under Grant NGII20190504; in part by the Youth Science and Technology Star of Dalian under Grant 2018RQ45; in part by the Opening Project of Shanghai Trusted Industrial Control Platform under Grant TICPSH202003017-ZC; in part by the Program for Liaoning Innovative Research Term in University under Grant LT2016007; in part by the Program for the Liaoning Distinguished Professor, Program for Innovative Research Team in University of Liaoning Province; in part by the Science and Technology Innovation Fund of Dalian under Grant 2020JJ25CY001; and in part by the PR of China Ministry of Education Distinguished Possessor Grant given to Prof. Obaidat under Grant MS2017BJKJ003. *(Corresponding authors: Pengfei Wang; Ziqi Wei; Qiang Zhang.)*

Pengfei Wang, Zhen Yu, and Qiang Zhang are with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: wangpf@dlut.edu.cn; yuzhen@dlut.edu.cn; zhangq@dlut.edu.cn).

Chi Lin is with the School of Software, Dalian University of Technology, Dalian 116600, China (e-mail: c.lin@dlut.edu.cn).

Mohammad S. Obaidat is with the College of Computing and Informatics, University of Sharjah, Sharjah, UAE, also with the King Abdullah II School of Information Technology, University of Jordan, Amman 19328, Jordan, also with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with the School of Engineering, Amity University—A Global University, Noida 201313, India (e-mail: msobaidat@gmail.com).

Ziqi Wei is with the School of Software, Tsinghua University, Beijing 100084, China (e-mail: weizq@tsinghua.edu.cn).

Digital Object Identifier 10.1109/JIOT.2020.3049024

I. Introduction

THE PANDEMIC of COVID-19 has brought serious social and healthy problems to people in this year. To decrease its negative influence, a large amount of contact tracing systems and applications [1], [2] have been devised and implemented by governments of the world rapidly. For example, Singapore released a Bluetooth-based contact tracing system to monitor people's daily contact information, China leveraged the health QR code system to record people's identities and daily activities and Israel adopted a location-based tracing system to track people's movement, Korea posted the private tracing data of the infectors to the public. Meanwhile, there are also many novel contact tracing systems proposed by academia [3], [4] recently.

Nevertheless, the privacy concern from the public is still the most important barrier that prevents people adopting the contact tracing system, although many new techniques, which ensure the user privacy, are induced to the system, e.g., blockchain-based techniques [5], [6], data encryption techniques [7], [8], etc. It is mainly because the nature of contact tracing system which leverages the precise contact tracing information to stop the virus infection. Generally speaking, the performance of contact tracing system will be decreased when the system does not get enough precising contact information, and this is also an irreconcilable contradiction between the privacy protection and precising contact tracing.

To combat with COVID-19, current contact tracing systems require participants to devote their contact tracing data voluntarily, and all users sacrifice their privacy with no

reimbursement. The current process really demotivates users to adopt such systems. On the one hand, the system asks users to give up their privacy and contribute their contact tracing information. On the other hand, users also need other resources (e.g., communication, storage, power, etc.) to fulfill the contact tracing task.

Essentially, the process of collecting contact tracing data in the contact tracing system is quite similar to that in crowdsensing [9], [10] which requires large amounts of participants (e.g., normal smartphone users) to sense the surrounding environment via rich built-in sensors of mobile devices, and uploads these data to the system. In practice, crowdsensing also brings the privacy challenges to participants since they are also required to provide the sensitive information (e.g., location information, surrounding sensing information, etc.). To overcome the difficulties, the incentive frameworks and algorithms are usually devised to stimulate participants to provide these information to the system.

Inspired by crowdsensing, we devise the incentive framework, named CovidCrowd, for contact tracing system in this article. As far as we know, this is the first paper which treats the whole process of collecting contact tracing information from the prospect of crowdsourcing. In CovidCrowd, the system (i.e., data consumer) buys the contact tracing data from the public instead of asking them to provide them voluntarily. To achieve it, the government first chooses a total reward to motivate all users. Then, all users will select their own response strategies. The goal of CovidCrowd is to compute the optimal total reward value to maximize the system utility.

Toward a comprehensive solution for contact tracing incentive of COVID-19 and other pandemic diseases, we make the following contributions.

1) We devise CovidCrowd—an incentive framework for the contact tracing system to stimulate users to upload their contact tracing data. Creatively, we treat the privacy leakage as a kind of cost for users. As far as we know, this is the first paper that addresses the contact tracing problem with the incentive method (Section III).
2) The incentive problem of the contact tracing system is formulated as a Stackelberg game with two stages where the government first posts a total reward and users choose their response strategies (Section IV-A).
3) We show that the Nash equilibrium of participating users (Section IV-B), and CovidCrowd is also leveraged to find the optimal reward value which could maximize the system utility (Section IV-C).
4) To evaluate the effectiveness of CovidCrowd, we conduct a large-scale simulation and a real-world data set evaluation (Section V). Both results show that CovidCrowd outperforms the other two benchmarks. Specifically, the user participating level is improved by at least 13.2% under all scenarios.

The remainder of this article is organized as follows. Section II discusses the related work first. We introduce the system model from the crowdsourcing perspective and formulate the problem in Section III. Section IV presents the definition of CovidCrowd and detail of the whole game process. Evaluation settings and numerical results are presented in Section V, and we conclude this article finally in Section VI.

II. Related Work

In this section, we first investigate the related work on contact tracing of COVID-19. Then, the crowdsensing systems are introduced. We also elaborate on game theory-based incentive frameworks and algorithms for crowdsensing since our work is to devise an incentive framework for contact tracing system.

In order to prevent the spread of COVID-19, a number of contact tracing systems are designed by governments and companies. For example, Apple Google collaboration [11], Singapore's TraceTogether [12], European PEPP-PT [13], and MIT's PrivateKit [14] leverage the Bluetooth technique to get the contact information from users. In Israel, the government is allowed to obtain the private data of suspected infected persons and directly store the location information of people in a central database [15]. The health code system [16] is widely used in China, and it is an epidemic prevention measure based on mobile phones, 3-D face recognition and population management in multiple occasions.

In addition, researchers from academia also devise and propose many different contact tracing systems by taking different factors into considerations. Here, we introduce some recent work on it. Gupa *et al.* [3] proposed a method of contact tracing by collecting WiFi data and design a complete system through passively collected WiFi data to accurately identify users within the range of activities of confirmed patients. Altuwaiyan *et al.* [1] developed an EPIC system based on wireless communication between smartphones, server and wireless short-range devices to check whether the user has contact with infected person. Wang *et al.* [2] proposed to use big data analysis technology to classify infection risks of residents and generate real-time alerts.

Actually, directly collecting or even publishing personal private data is intrusive, potentially violates individual privacy rights, and is often subject to regulations and laws. Therefore, many privacy preserving oriented work are also conducted recently. For example, Reicherts *et al.* [7] leveraged the secure multiparty computation technology in classical cryptography to transmit tracing data. Gupta *et al.* [4] adopted the method of encrypting and transmitting the collected WiFi tracing data to ensure user privacy and data security. Torky and Hassanien [5] begin to explore the use of decentralized feature in blockchain technology to protect user privacy data. Lv *et al.* [17] proposed the use of blockchain combined with zero knowledge proof methods to achieve privacy protection. However, all these works assume that the central servers or the third parts are trustful, but both the disclosure of private key and attacks may lead to privacy exposure in practice. Thus, we utilize the idea of crowdsourcing to buy the contact tracing data in this article.

Crowdsourcing [18]–[20] is an idea of using the power of ordinary people to solve complex problems and a distributed problem-solving and business production model. Crowdsourcing has received extensive attention in various fields, including human–computer interaction [21], machine learning [22], computer theory [23], information retrieval [24],

databases [25], etc. Crowdsensing [26]–[28] is a kind of crowdsourcing-based system which leverages the power of the crowds to sense the surrounding areas, and many crowdsensing systems have solved a lot of realistic problems in real-world scenario. For example, FlierMeet [29] is mainly for the public information reporting and sharing, CrowdMonitor [30] is adopted to assess physical and digital activities of citizens, and the system named third eye [31] could monitor the air quality, etc.

Incentive algorithms could help crowdsensing systems motivate users to contribute more sensing data, and the game theory is a very important method. Game theory [32]–[35]-based incentive frameworks and algorithms are well studied in recent years. Among them, the incentive algorithms based on game theory mainly contain Stackelberg game, SPE-based game, and Bargaining game.

Stackelberg game [36] is widely used in the design of incentive mechanisms, and it is basic model includes a leader and several followers. The leader first takes action, and then the followers adjust their strategies according to the leader's actions to maximize their own utility. Duan *et al.* [37] used Stackelberg game to design a threshold revenue model, and show that if service providers can know the cost of users in advance and it can choose the users who participate and only need to provide a small amount of rewards. Luo *et al.* [38] adopted Stackelberg game to maximize user utility for user-centric incentive model, and show that contribution-dependent reward system perform better than optimal constant reward system. Xiao *et al.* [39] proposed a Stackelberg game incentive model to prevent forged sensing attacks, and using deep learning and reinforcement learning to obtain the optimal crowdsensing strategy.

Inspired by the above work from the contact tracing system and crowdsensing, we propose CovidCrowd which is the first incentive framework for COVID-19 and other pandemic diseases to motivate users to provide their private contact tracing data.

III. System Model

In this section, we devise the contact tracing system from the perspective of crowdsourcing and describe its basic components and workflow.

Generally speaking, the performance of contact tracing system could be more accurate if users provide more contact tracing data during their daily life [40]. However, it is impossible for rational users to contribute their data when their utilities are negative according to the game theory [41]. Therefore, we adopt the concept of crowdsourcing to stimulate users to contribute more private data in this article.

Fig. 1 is leveraged to aid our description of the contact tracing system from a crowdsourcing perspective, and we treat the contact tracing data collection as a task. To be specific, different from the general contact tracing system [27] where participants contribute their tracing data voluntarily, the government (i.e., data consumer) buys these data from participating users with the total reward within a specific time

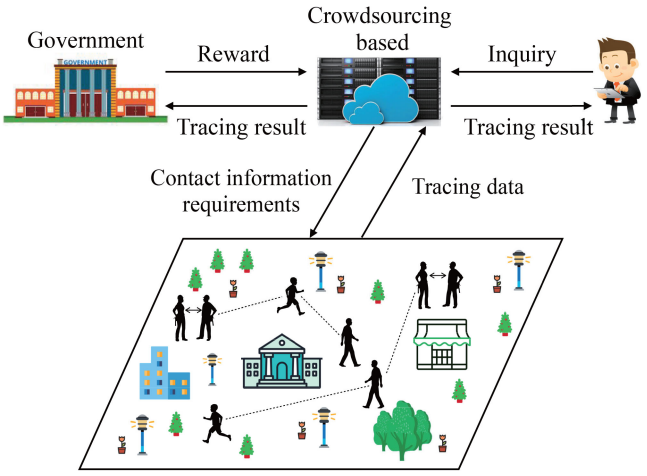


Fig. 1. Framework of contact tracing system from a crowdsourcing perspective.

period (e.g., one day, etc.). Then, users choose their participating levels/strategies (i.e., how long to participate) when using contact tracing system during the daily life. The user will choose to cooperate only when the received reward could cover his/her cost in both privacy leakage and data processing. Finally, the user could get the corresponding reward based on the participating levels of all users and the total reward of the system.

In essence, this is a Stackelberg game, including two stages where the government is the leader and the users are followers. In the first stage, the government first determines the total reward to motivate users. In the second stage, each user chooses a response strategy (i.e., participating level) to maximize his/her own utility based on the observed total reward given by the government.

IV. CovidCrowd

We elaborate on the details of the incentive framework (i.e., CovidCrowd) for the contact tracing system in this section. To achieve the goal, CovidCrowd first formulates the contact tracing incentive problem as a Stackelberg game. Then, we analyze the user response strategy based on the user utility function and show there exists a Nash equilibrium for any users with any reward value. Based on the above analysis, the Nash equilibrium of the contact tracing system is studied. Finally, CovidCrowd computes the optimal reward value to maximize the system utility and motivate users.

A. Problem Formulation

The contact tracing system contains one task during the daily life, i.e., collecting contact tracing data from users, and the government could monitor the close contacts of COVID-19 with the detailed contact tracing data. Specifically, the government offers a total reward R to attract users to upload their contact tracing data. At the same time, users also choose their participating levels (i.e., how long to devote the contact tracing data), and they will get the corresponding reward based

on the total reward R and participating levels (strategies) of other users.

Therefore, we can compute the utility of user i (i.e., u_i) by

$$u_i = \frac{t_i}{\sum_{j\in U} t_j} R - c_i \times t_i \tag{1}$$

where U is the user set, $c_i \in (0, c_{\max})$ denotes the unit cost of user i, and t_i represents the participating level (e.g., the participating time period, the uploading tracking data quality, etc.) of user i. In addition, $[t_i/(\sum_{j\in U} t_j)]R$ is the reward that user i will receive after joining the task with participating level t_i. It should be noticed that a rational user will not cooperate when the utility is negative. In other words, user i will do nothing (i.e., $t_i = 0$) if $[t_i/(\sum_{j\in U} t_j)]R < c_i t_i$. Therefore, We could rewrite (1) as follows:

$$u_i = \begin{cases} \frac{t_i}{\sum_{j\in U} t_j} R - c_i t_i, & c_i < \frac{R}{\sum_{j\in U} t_j} \\ 0, & c_i \geq \frac{R}{\sum_{j\in U} t_j}. \end{cases} \tag{2}$$

Differing from the general crowdsensing system where the total participating cost of a user is closely related with the data size and user ability [42]–[44], we also creatively treat the user privacy as a kind of cost in the contact tracing system. Next, we define the unit cost function of user i (i.e., c_i) in the contact tracing system by

$$c_i = g(\alpha_i, \beta_i, \gamma_i) \tag{3}$$

where α_i represents the unit privacy cost for user i, the unit data processing cost (including generating, communication cost, etc.) for user i is denoted by β_i, and γ_i is the average data size generated by user i (e.g., the average contact tracing data size generated by user i every hour). In addition, it is assumed that α_i and β_i usually follow the Gaussian distribution, which is a very common assumption [45]. Obviously, it is not hard to get that the partial derivatives of function $g(\alpha_i, \beta_i, \gamma_i)$ should satisfy the following equations:

$$\frac{\partial g(\alpha_i, \beta_i, \gamma_i)}{\partial \alpha_i}, \frac{\partial g(\alpha_i, \beta_i, \gamma_i)}{\partial \beta_i}, \frac{\partial g(\alpha_i, \beta_i, \gamma_i)}{\partial \gamma_i} > 0. \tag{4}$$

The unit cost will be higher when they have a greater unit privacy cost, data processing cost, or larger unit generated contact tracing data. Besides, the unit cost function $g(\alpha_i, \beta_i, \gamma_i)$ is usually related with the system attributes. We give a concrete $g(\alpha_i, \beta_i, \gamma_i)$ for the simulation in Section V.

Next, we define some essential variables for CovidCrowd to compute the system utility. Equation (2) depicts that users would like to choose the same strategy to maximize their utility if their unit costs are same. Furthermore, the unit cost interval $(0, c_{\max})$ is defined where $c_{\max}$ is the maximum unit cost for users in the contact tracing system, and it is easy to conclude that $0 < c_i \leq c_{\max}$ for any user i. It is assumed that the unit cost set of all users is $S = \{s_1, s_2, \ldots, s_X\}$ in the contact tracing system, and it has the following constraints $0 < s_x \leq c_{\max}(1 \leq x \leq X), c_i \in S(0 \leq i \leq n)$.

Moreover, the expected participating level for specific unit cost s_x is denoted by $\hat{t}_x(1 \leq x \leq X)$, and we also give the expected participating level set $\hat{T} = \{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_X\}$ correspondingly. In addition, the number of users whose unit costs are same is represented as $n_x(1 \leq x \leq X)$, and the set of the number of users for $\hat{T}$ is denoted by $N = \{n_1, n_2, \ldots, n_X\}$. According to the above definitions, the utility function of contact tracing system can be computed as

$$u_0 = f\big(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_X; n_1, n_2, \ldots, n_X\big) - R \tag{5}$$

where $f(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_X; n_1, n_2, \ldots, n_X)$ represents the valuating function of participating levels of all users in the contact tracing system. It is also assumed that the utility function of contact tracing system is a strictly concave function in variables T for any fixed N, and it is increasing monotonically for every t_i of user i. This is a general assumption in many other related papers [46]. Each user i decides its strategy (i.e., t_i) to maximize (2) with a given reward value R.

The goal of CovidCrowd is to compute the optimal reward value (i.e., R^*) which maximizes the system utility [i.e., (5)], and this is also the Nash equilibrium status of the contact tracing system. To achieve the goal, we first analyze the Nash equilibrium of users (i.e., response strategies), then elaborate on how to get the optimal reward value of the contact tracing system in the next. Moreover, detailed explanations of employed notations in this article are also concluded in Table I.

TABLE I
MAJOR NOTATIONS EMPLOYED IN THIS ARTICLE

Notation	Explanation
R	total reward of the contact tracing system
R^*	optimal reward value
R_{others}	other feasible reward values
U	user set
U_{-i}	user set except user i
u_i	utility function of user i
u_0	utility function of the contact tracing system
n	number of users
t_{-i}	strategy profile of all users except user i
t_{-i}^{ne}	strategy profile of all users except user i in Nash equilibrium
t_i	participating level (strategy) of user i
t_i^{ne}	participating level (strategy) of user i in Nash equilibrium
c_i	unit cost of user i
c_{max}	maximum unit cost
S	unit cost level set
X	number of unit cost levels
s_x	xth unit cost level
$\hat{T}$	expected participating time duration set
$\hat{t}_x$	expected participating time duration for s_x
N	set of the user number at different unit cost levels
n_x	number of users whose unit costs are same
B_i	best response strategy of user i
α_i	unit privacy cost of user i
β_i	unit data processing cost of user i
γ_{max}	maximum encounter times of all users
γ_i	encounter times of user i

B. User Nash Equilibrium

Given a reward R from the government, user i would like to choose the best response strategy to maximize its utility [i.e., (2)]. Therefore, a user would like to play its best response strategy in the Nash equilibrium.

To study the user response strategy, we first define the Nash equilibrium of user i as below:

Definition 1 (Nash Equilibrium of Users): A strategy set $(t_1^{ne}, t_2^{ne}, \ldots, t_n^{ne})$ is a Nash equilibrium when any user i

satisfies (20)

$$u\left(t_i^{ne}, t_{-i}^{ne}\right) \geq u\left(t_i, t_{-i}^{ne}\right) \tag{6}$$

where $u(t_i^{ne}, t_{-i}^{ne})$ denotes the utility that user i adopts the strategy t_i^{ne}, and $u(t_i, t_{-i}^{ne})$ is the user utility that user i adopts the other strategy t_i. Then, the best response strategy of user i is also defined as following.

Definition 2 (Best Response Strategy of User): Strategy t_i for user i, denoted by B_i, is the best response strategy if it maximizes $u(t_i, t_{-i})$ over all $t_i \geq 0$.

To further study the best response strategy of user i, we compute the first- and second-derivatives of utility function u_i with respect to t_i as depicted in the following equation:

$$\frac{\partial u_i}{\partial t_i} = \frac{R}{\sum_{j\in U} t_j} - \frac{t_i R}{\left(\sum_{j\in U} t_j\right)^2} - c_i \tag{7}$$

$$\frac{\partial^2 u_i}{\partial t_i^2} = -\frac{2Rt_i}{\left(\sum_{j\in U} t_j\right)^3}. \tag{8}$$

It is obvious that the utility function of user i is a strictly concave function since the second-order derivative $[(\partial^2 u_i)/(\partial t_i^2)] \leq 0$. To compute the best response strategy of user i for his/her Nash equilibrium, we can have (9) by setting the first-order derivative $[(\partial u_i)/(\partial t_i)] = 0$

$$\frac{R}{\sum_{j\in U} t_j} - \frac{t_i R}{\left(\sum_{j\in U} t_j\right)^2} - c_i = 0. \tag{9}$$

We can get (10) by sorting out (9)

$$t_i = \sqrt{\frac{\left(\sum_{j\in U_{-i} t_j}\right)R}{c_i}} - \sum_{j\in U_{-i}} t_j. \tag{10}$$

Any user i will not cooperate when the utility is lower than 0. Thus, the best response strategy of user i can be summarized as in

$$B_i = \begin{cases} \sqrt{\frac{\left(\sum_{j\in U_{-i}} t_j\right)R}{c_i}} - \sum_{j\in U_{-i}} t_j, & R \geq c_i \times \sum_{j\in U_{-i}} t_j \\ 0, & R < c_i \times \sum_{j\in U_{-i}} t_j. \end{cases} \tag{11}$$

The above analysis shows that any user $i(0 < i < n)$ has its best response strategy B_i for any given reward $R > 0$ and strategy profile t_{-i} of other users. Adding up (9) over all users in the contact tracing system, we can obtain

$$-R + |U| \times R - \left(\sum_{j\in U} c_j\right) \times \left(\sum_{j\in U} t_j\right) = 0. \tag{12}$$

Furthermore, we could obtain

$$\sum_{j\in U} t_j = \frac{(|U|-1)R}{\sum_{j\in U} c_j}. \tag{13}$$

To calculate t_i for user i, (9) can also be rewritten as

$$t_i = \sum_{j\in U} t_j - \frac{c_i}{R} \times \left(\sum_{j\in U} t_j\right)^2. \tag{14}$$

Algorithm 1 Computing the User Best Response Strategy in Nash Equilibrium

Require: Reward R, user set U.
Ensure: Nash equilibrium strategy set for all users $\hat{T} = \{t_1^{ne}, t_2^{ne}, ..., t_n^{ne}\}$

```
t^ne ← Φ;
for i ← 1 to n do
    t_tmp ← (|U|−1)R / Σ_{j∈U} c_j − c_i × R × (( |U|−1) / Σ_{j∈U} c_i)^2;
    if t_tmp > 0 then
        t_i^ne ← t_tmp;
    else
        t_i^ne ← 0;
    end if
    t^ne ← t^ne.add(t_i^ne);
end for
return t^ne;
```

Substituting (13) into (14), we can calculate the best response strategy for user i as

$$t_i = \frac{(|U|-1)R}{\sum_{j\in U} c_j} - c_i \times R \times \left(\frac{(|U|-1)}{\sum_{j\in U} c_j}\right)^2. \tag{15}$$

According to (15), we can compute the Nash equilibrium of users, and Algorithm 1 depicts the details.

So far, we have known how to calculate the Nash equilibrium of any user i with any given reward value R in the contact tracing system. In the next section, we will elaborate on how to choose the optimal reward value R^* to maximize the contact tracing system's utility.

C. Maximizing the Utility of Contact Tracing System

Obviously, the contact tracing system, and leader of the Stackelberg game, can know the existing user Nash equilibrium based on the above analysis. In this case, the contact tracing system can select the optimal reward value R^* to maximize its utility based on the following:

$$u_0 = \left(f\left(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_X; n_1, n_2, \ldots, n_X\right)/(R-1)\right)R \tag{16}$$

where

$$\hat{T} = \left\{t_1^{ne}, t_2^{ne}, \ldots, t_X^{ne}\right\} \tag{17}$$

$$N = \{n_1, n_2, \ldots, n_X\}. \tag{18}$$

For each $t_i^{ne} \in \hat{T}$, we have

$$t_i^{ne}/R = \frac{(|U|-1)}{\sum_{j\in U} c_j} - c_i \times \left(\frac{(|U|-1)}{\sum_{j\in U} c_j}\right)^2 \tag{19}$$

where u_0 is a strictly concave function, $\hat{T}$ could be obtained by Algorithm 1, and N could be calculated according to the previous statistics by analyzing the historical data.

Next, we define the Nash equilibrium of contact tracing system as follows.

Definition 3 (Nash Equilibrium of Contact Tracing System): The chosen reward value R is the Nash equilibrium of contact tracing system if it can maximize the system utility.

Besides, the best response strategy of contact tracing system is defined as follows.

Definition 4 (Best Response Strategy of Contact Tracing System): The optimal reward value R^* is the best response strategy with the given user Nash equilibrium strategy set $(t_1^{ne}, t_2^{ne}, \ldots, t_n^{ne})$, and R^* meets

$$u_0\left(R^*\right) \geq u_0(R_{\text{others}}). \tag{20}$$

In summary, it can be concluded that the Nash equilibrium exists in this Stackelberg game, and the optimal reward R^* can maximize the utility function u_0 in (16) over $R \in [0, +\infty)$. To calculate R^*, many efficient methods could be leveraged, such as Newton's method [47], etc.

V. Performance Evaluation

To evaluate the performance of CovidCrowd, two experiments, including a large-scale simulation with thousands of users and an evaluation with the real-world data set, are conducted in this section. Next, we first introduce the default evaluation settings of both two experiments.

A. Default Evaluation Settings

1) Utility Functions: In the contact tracing system, the unit cost function is set to (22)

$$g(\alpha_i, \beta_i, \gamma_i) = \left(\frac{\gamma_i}{\gamma_{\max}} \times c_{\max}\right) \times \left(1 - e^{-\alpha_i}\right) \times \left(1 - e^{-\beta_i}\right) \tag{21}$$

where γ_i and $\gamma_{\max}$ represent the encounter times of user i and the maximum encounter times of all users, respectively. Both of α_i and β_i follow the Gaussian distribution with mean 5 and variance 1 in the interval $(0, +\infty)$, and this is because the privacy and communication costs of most users are concentrated in one area.

In terms of the utility function of contact tracing system, we set the valuating function to

$$f\left(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_X; n_1, n_2, \ldots, n_X\right) = \lambda \log\left(1 + \sum_{c_j \in S} \log\left(1 + t_j\right)\right) \tag{22}$$

where $t_j \in \hat{T}$ and $\lambda = 10$ is a system parameter, since $\log(1 + t_j)$ represents the contact tracing system's diminishing return on the work of a user with unit cost c_j, and $\log\left(1 + \sum_{c_j \in S} \log(1 + t_j)\right)$ denotes the contact tracing system's diminishing return on the number of participating users.

2) Compared Strategies: In order to evaluate the performance of CovidCrowd, we adopt two compared strategies (benchmarks)—best effort strategy and random strategy. User i is always trying his best to upload the tracing data (i.e., $t_i = t_{\max}$) if one can obtain the profit (i.e., the utility of user is larger than zero) from the contact tracing system when adopting the best effort strategy. In addition, we can get the value of $t_{\max}$ by computing the maximum t_i of all users. Random algorithm is a strategy that allows user i to select t_i based on his/her own preferences at random. User i can choose the random participating level value in $[0, t_{\max}]$.

3) Metrics: Four metrics are leveraged to measure the effectiveness of CovidCrowd, including the system utility, total participating level, average participating level, and the number of participants during the whole Stackelberg game process. Specifically, the system utility u_0 has been introduced in Section III, and it is the key metric to measure the algorithm performance; the total participating level is defined as the sum of user t_i; the average participating level can be computed by the total participating level divides the number of users; and the number of participants is the number of users whose $t_i > 0$ during the game process.

B. Simulation

In this section, we conduct the simulation with thousands of users. We first show the basic scenario settings, then validate CovidCrowd by varying the number of users and the maximum unit cost of users.

1) Simulation Settings: The default values of simulation settings are defined as below. It is assumed that the contact tracing system is applied in a 10 km $\times$ 10 km region (i.e., 100 km^2). The default number of users, who participate the tracing task and upload their tracing data to improve tracing capacity of system with a given reward value R, is 1000. All users are randomly distributed in the area, and they are stationary or moving in all directions at a speed of 1 m/s. A distance of 1.5 m between users is a relatively safe distance, and a distance within 1.5 m is counted as mutual contact [48]. Notice that, we use a day as the unit time of evaluation. Unless otherwise specified, we use the default values to conduct the following simulations.

2) Performance With Different Numbers of Users: By changing the number of users, we show the performance of CovidCrowd and the other two benchmarks. In particular, the number of users is changed from 1000 to 6000 with the increment of 1000 users, and the results are depicted in Fig. 2.

Fig. 2(a) depicts the system utility performance with different numbers of users. CovidCrowd is 2.92% higher than best effort and 35.5% higher than random by on average, respectively. System utility of CovidCrowd is increased with the number of users going up, and the reason here is that the tracing performance of CovidCrowd system could be improved when more users upload their tracing data. Notice that, CovidCrowd has the highest system utility contrasting with the other algorithms, and this is our key goal to design CovidCrowd which chooses the best response strategy to maximize the system utility.

Fig. 2(b) describes the total participating level result with different numbers of users. On average, the total participating level of CovidCrowd is 16.3% better than best effort and 81.4%better than random. As it is expected, CovidCrowd is better than best effort and random algorithm in total participating level, and it is more obvious with the increase of the number of users. This results in that CovidCrowd can use the best strategy to get more user's participating level.

In terms of average participating level depicted in Fig. 2(c), CovidCrowd has an improvement of 17.1% compared

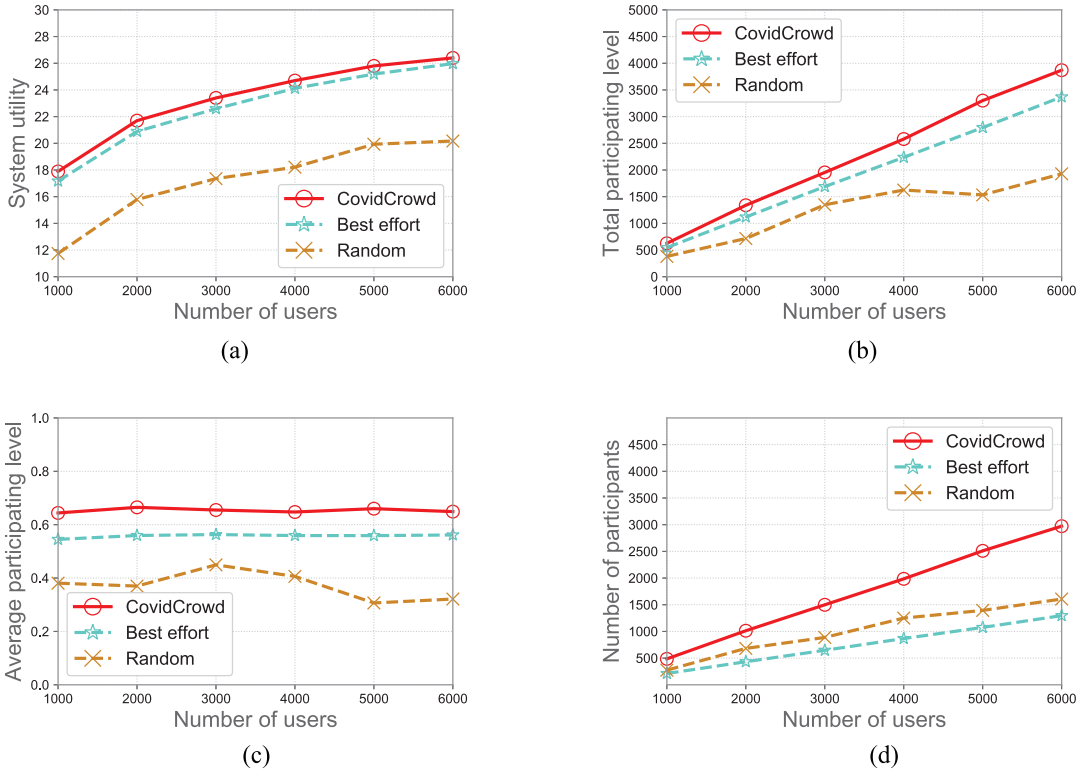


Fig. 2. Performance with different numbers of users. (a) System utility. (b) Total participating level. (c) Average participating level. (d) Number of participants.

with best effort and 75.4% compared with random. Both CovidCrowd and best effort algorithm reach to a stable state. In contrast, the average participating level of random algorithm varies as the number of users increases, and this is mainly because users randomly choose how to cooperate (e.g., whether participate or not, how to participate, etc.). In addition, the random algorithm is still the worst case among three algorithms.

Fig. 2(d) depicts the number of participants performance with different numbers of users. Similar to the result of system utility, the number of participants for CovidCrowd also achieves the highest as the number of users increasing. On average, the number of participants of CovidCrowd is 131% higher than best effort and 71.7% higher than random, respectively. The main reason is that CovidCrowd is a Nash equilibrium status and user will cooperate only if it can obtain profits from CovidCrowd.

3) Performance With Different Maximum Costs of Users: Then, we evaluate all three algorithms with different number of user's maximum unit cost. In particular, the maximum unit cost value is varied from 8 to 13 with the increment of 1, and Fig. 3 depicts the final results.

From the perspective of system utility [i.e., Fig. 3(a)], CovidCrowd still outperforms the other algorithms since it is the Stackelberg Nash equilibrium status. Specifically, we can conclude that CovidCrowd outperforms both best effort (4.3% improvement on average) and random (51.4% improvement on average) in all scenarios. Furthermore, the best effort algorithm is better than the random algorithm in system utility with different maximum costs, and this is mainly because that users with best effort algorithm are always doing their best to provide tracing data.

The participating level [i.e., Fig. 3(b) and Fig. 3(c)] of three algorithms shows the same trends that CovidCrowd is 20.7% better than best effort and 95.2% better than random on average. In addition, both total participating level and average participating level of CovidCrowd and best effort could become lower when the number of maximum costs increases. This is due to that the increase in the maximum costs improves the unit cost of user, which leads to a decrease in user's participating level.

The number of participants is depicted in Fig. 3(d). On average, the number of participants of CovidCrowd is 131.5% higher than best effort algorithm and 61.9% higher than random algorithm. It is concluded that CovidCrowd also can stimulate more users to participate in providing extra tracing data under different maximum costs of users. Specifically, CovidCrowd can reach to a Stackelberg Nash equilibrium where all participated users cannot obtain more extra profits if its derivatives are from current algorithm.

To summarize, we can see that CovidCrowd has the highest system utility, participating level and the number of participants comparing with the other two benchmarks. Although the system utility of best effort is close to CovidCrowd, it is difficult to reach since users are rational (selfish) in practice. Thus, CovidCrowd outperforms the other algorithms with the increase of the number of users and maximum unit cost.

C. Real-World Data Set Evaluation

1) Data Set Description: We also conduct the simulation with a real-world trace data set to evaluate the performance of CovidCrowd. The contact tracing application sample data set [49] is leveraged, and it is the latest data set that records

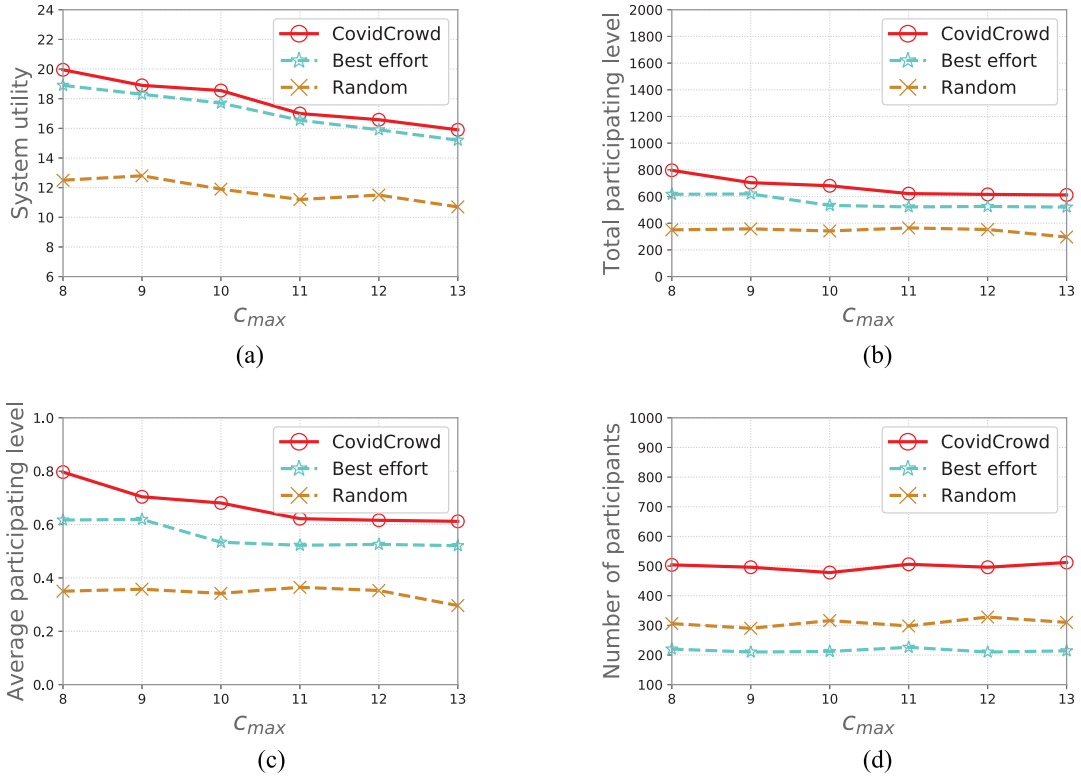


Fig. 3. Performance with different maximum unit costs. (a) System utility. (b) Total participating level. (c) Average participating level. (d) Number of participants.

contact tracing information in August 2020. Data set includes basic information of users and their corresponding contact information from August 1st to August 31st. To be specific, user's basic information includes name, birthday, gender, phone number and infection information (i.e., whether the user is infected and if so his or her infection date), etc., and contact information includes contact users (i.e., the user who reported the contact and the users who had contact with him) and contact start and end time.

2) Simulation Settings: The default value of $c_{\max}$ is 10, and we still adopt one day as the unit time period for the evaluation. We use a day as the unit time for evaluation, and choose to show the results of one week (i.e., 7 evaluations in 7 days) from the data set. This is mainly because that the result trends of other days are similar after we conduct the evaluation, and one week is representative enough. Other default function settings are the same as Section V-A.

3) Performance With Different Days: We conduct a total of one-week evaluation in days (i.e., from Monday to Sunday), and the final results are depicted in Fig. 4. From the Figure, we can get that the broken lines fluctuate without significant changes and the performance of CovidCrowd is always better than the best effort and random as the days changes.

To be concrete, In terms of system utility [i.e., Fig. 4(a)], CovidCrowd outperforms both best effort (7.1% improvement on average) and random (61.6% improvement on average). CovidCrowd also achieves the highest participating level compared to best effort (13.2% improvement on average) and random (114% improvement on average) from Fig. 4(b) and (c). In addition, the number of participants depicted in Fig. 4(d) shares CovidCrowd has 133% more participating users than best effort and 79.5% more than random.

4) Performance With Different Maximum Unit Costs: In this section, we only show the result of one day's evaluation due to the page limits, and the trend of metrics is quite similar in different days. The maximum unit cost of users is varied from 8 to 13, and the result is shown in Fig. 5.

Similarly with the simulation, the real-world use case shows that CovidCrowd could stimulate more users to upload their tracing data with higher participating level and improve system utility for contact tracing system comparing with the other two benchmarks. On average, CovidCrowd outperforms both best effort (6.9% improvement for system utility, 19.2% improvement for participating level, and 134% improvement for number of participants) and random (44.8% improvement for system utility, 97% improvement for participating level, and 81.7% improvement for number of participants) in all scenarios.

5) Performance of Random Users: In this section, we randomly select four users from participating users (i.e., $t_i > 0$) to analyze the user participating level. To be specific, We stand from the perspective of users to analyze their participating strategy under different algorithms, so we leverage participating level to evaluate the participating strategy of users with changing time.

The result of participating level with different time is depicted in Fig. 6. The most obvious finding here is that the user utility of best effort algorithm is the highest and the

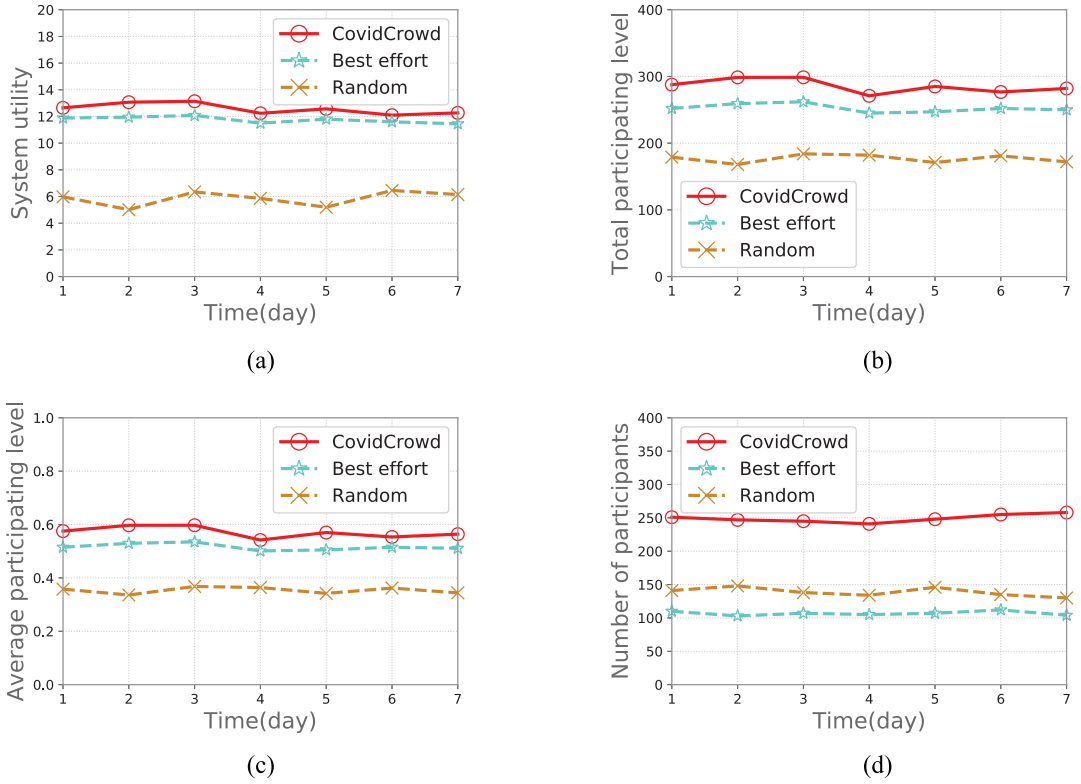


Fig. 4. Performance at different time. (a) System utility. (b) Total participating level. (c) Average participating level. (d) Number of participants.

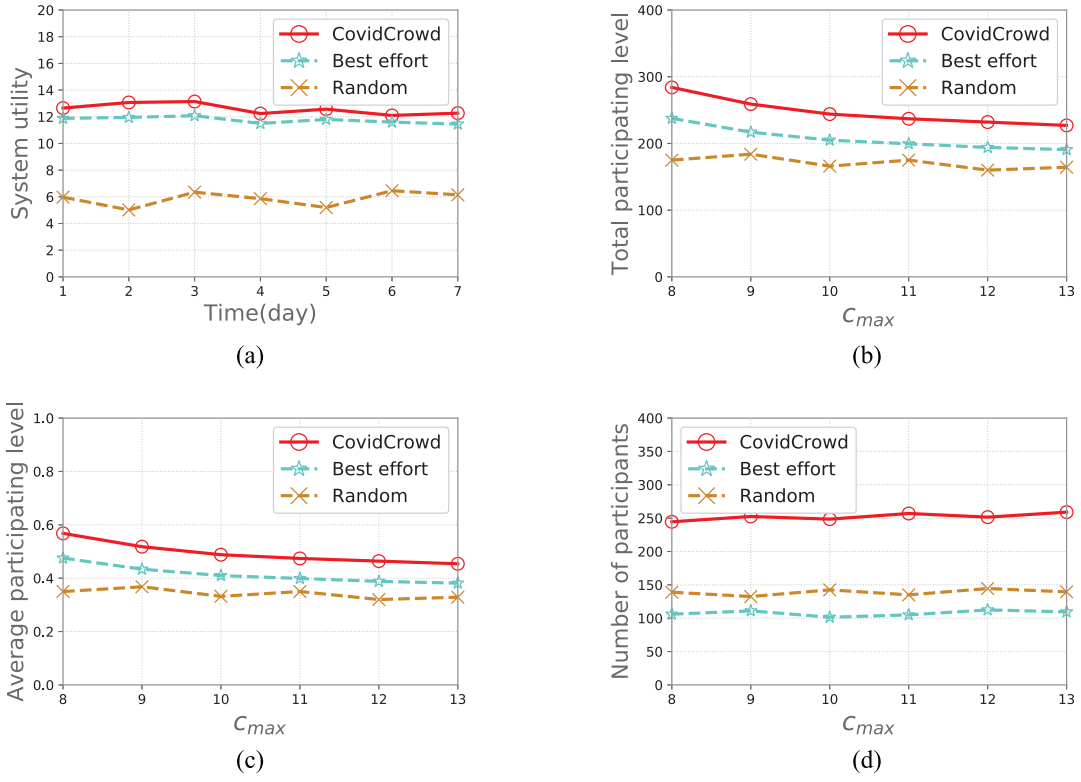


Fig. 5. Performance with different maximum unit costs. (a) System utility. (b) Total participating level. (c) Average participating level. (d) Number of participants.

random algorithm is the lowest in most cases. The main reason here is that the best effort strategy is always trying best to fulfill the task, however, the participating level of CovidCrowd is better than best effort when we consider all users in the contact tracing system. In addition, the system utility will not reach to the maximum, even though the participating level of single

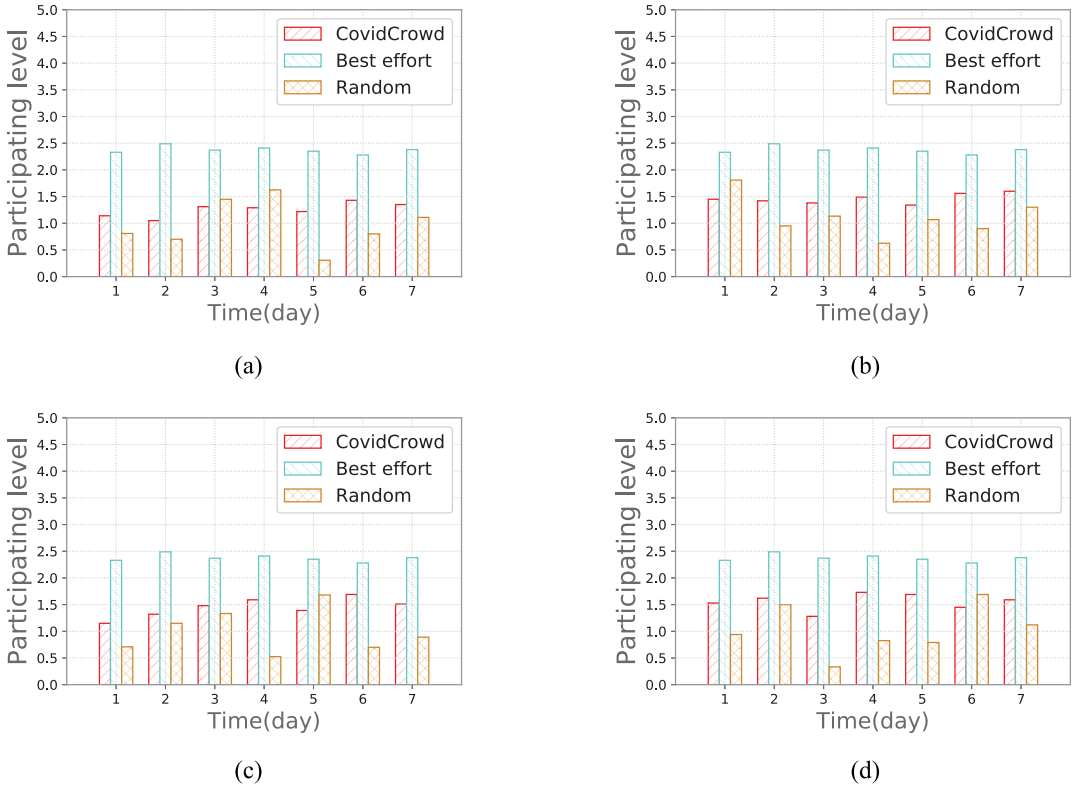


Fig. 6. Participating level with random selected users. (a) Random user A. (b) Random user B. (c) Random user C. (d) Random user D.

participating users is high. This is mainly because that the best effort strategy is not the Nash equilibrium of users, and the goal of CovidCrowd is to encourage more users joining the system and share their contact tracing information instead of fewer ones.

In real-world tracing data set, CovidCrowd can achieve the highest system utility, participating level and the number of participants, the three key metrics of this article, contrasting with the other two benchmarks under various settings based on realistic scenario. Although best effort algorithm requires all users to try their best to do the task, this is not a Nash equilibrium status in practice and users are usually rational. CovidCrowd performs better than other algorithms, the main reason is that CovidCrowd use Nash equilibrium of Stackelberg game to choose the best response strategy for both contact tracing system and users.

VI. Conclusion

In this article, we studied the process of collecting contact tracing data for COVID-19 and other pandemic diseases from a crowdsourcing perspective. Motivating users to contribute more contact tracing data, the contact tracing system reimburses users' privacy and data processing cost with the total reward value, and users will choose their response strategies based on it. We formulate the problem as a Stackelberg game and show there exists the Nash equilibrium for any user given any fixed reward value. Then, CovidCrowd computes the optimal reward value which could maximize the utility of the system. Finally, we conduct a large-scale simulation and real-word data set evaluations. Both results show that CovidCrowd outperform the benchmarks in maximizing the system utility and improving the total participating levels.

Acknowledgment

The authors thank the reviewers and editors for their careful reading of this article and their many insightful comments and suggestions.

References

[1] T. Altuwaiyan, M. Hadian, and X. Liang, "EPIC: Efficient privacy-preserving contact tracing for infection detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, 2018, pp. 1–6.

[2] C. J. Wang, C. Y. Ng, and R. H. Brook, "Response to COVID-19 in Taiwan: Big data analytics, new technology, and proactive testing," *JAMA Netw.*, vol. 323, no. 14, pp. 1341–1342, 2020.

[3] P. Gupta, S. Mehrotra, N. Panwar, S. Sharma, N. Venkatasubramanian, and G. Wang, "Quest: Practical and oblivious mitigation strategies for COVID-19 using WiFi datasets," 2020. [Online]. Available: arXiv:2005.02510.

[4] D. Gupta, S. Bhatt, M. Gupta, and A. S. Tosun, "Future smart connected communities to fight COVID-19 outbreak," 2020. [Online]. Available: arXiv:2007.10477.

[5] M. Torky and A. E. Hassanien, "COVID-19 blockchain framework: Innovative approach," 2020. [Online]. Available: arXiv:2004.06081.

[6] H. Xu, L. Zhang, O. Onireti, Y. Fang, W. B. Buchanan, and M. A. Imran, "BeepTrace: Blockchain-enabled privacy-preserving contact tracing for COVID-19 pandemic and beyond," 2020. [Online]. Available: arXiv:2005.10103.

[7] L. Reichert, S. Brack, and B. Scheuermann, "Privacy-preserving contact tracing of COVID-19 patients," IACR Cryptol. ePrint Archives, Lyon, France, Rep. 2020/375, 2020.

[8] S. Brack, L. Reichert, and B. Scheuermann, "Decentralized contact tracing using a DHT and blind signatures," IACR Cryptol. ePrint Archives, Lyon, France, Rep. 2020/398, 2020.

[9] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.

[10] X. Zhang *et al.*, "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 54–67, 1st Quart., 2016.

[11] *Privacy-Preserving Contact Tracing*, Apple, Cupertino, CA, USA, 2020. [Online]. Available: https://covid19.apple.com/contacttracing

[12] J. Bay *et al.*, "BlueTrace: A privacy-preserving protocol for community-driven contact tracing across borders," Government Technol. Agency, Singapore, Rep., 2020. [Online]. Available: https://bluetrace.io/static/bluetrace_whitepaper-938063656596c104632def383eb33b3c.pdf

[13] *Pan-European Privacy-Preserving Proximity Tracing*, PEPP-PT, Tel Aviv-Yafo, Israel, 2020. [Online]. Available: https://www.pepp-pt.org/

[14] R. Raskar *et al.*, "Apps gone rogue: Maintaining personal privacy in an epidemic," 2020. [Online]. Available: arXiv:2003.08567.

[15] J. Tidy, *Coronavirus: Israel Enables Emergency Spy Powers*, BBC News, London, U.K., Mar. 2020.

[16] P. Mozur, R. Zhong, and A. Krolik, *In Coronavirus Fight, China Gives Citizens A Color Code, With Red Flags*, vol. 1, New York Times, New York, NY, USA, 2020.

[17] W. Lv, S. Wu, C. Jiang, Y. Cui, X. Qiu, and Y. Zhang, "Decentralized blockchain for privacy-preserving large-scale contact tracing," 2020. [Online]. Available: arXiv:2007.00894.

[18] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, 2006.

[19] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Trans. Mobile Comput.*, early access, Jul. 28, 2020, doi: 10.1109/TMC.2020.3012509.

[20] T. S. Behrend, D. J. Sharek, A. W. Meade, and E. N. Wiebe, "The viability of crowdsourcing for survey research," *Behav. Res. Methods*, vol. 43, no. 3, p. 800, 2011.

[21] J. Noronha, E. Hysen, H. Zhang, and K. Z. Gajos, "Platemate: Crowdsourcing nutritional analysis from food photographs," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 1–12.

[22] J. C. Chang, S. Amershi, and E. Kamar, "Revolt: Collaborative crowdsourcing for labeling machine learning datasets," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2017, pp. 2334–2346.

[23] S. Chawla, J. D. Hartline, and B. Sivan, "Optimal crowdsourcing contests," *Games Econ. Behav.*, vol. 113, pp. 80–96, Jan. 2019.

[24] E. Maddalena, S. Mizzaro, F. Scholer, and A. Turpin, "On crowdsourcing relevance magnitudes for information retrieval evaluation," *ACM Trans. Inf. Syst.*, vol. 35, no. 3, pp. 1–32, 2017.

[25] C. Chai, J. Fan, G. Li, J. Wang, and Y. Zheng, "Crowdsourcing database systems: Overview and challenges," in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, 2019, pp. 2052–2055.

[26] J. Liu, H. Shen, H. S. Narman, W. Chung, and Z. Lin, "A survey of mobile crowdsensing techniques: A critical component for the Internet of Things," *ACM Trans. Cyber Phys. Syst.*, vol. 2, no. 3, pp. 1–26, 2018.

[27] Z. Ning *et al.*, "Intelligent edge computing in Internet of vehicles: A joint computation offloading and caching solution," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 5, 2020, doi: 10.1109/TITS.2020.2997832.

[28] F. Restuccia, N. Ghosh, S. Bhattacharjee, S. K. Das, and T. Melodia, "Quality of information in mobile crowdsensing: Survey and research challenges," *ACM Trans. Sens. Netw.*, vol. 13, no. 4, pp. 1–43, 2017.

[29] B. Guo, H. Chen, Z. Yu, X. Xie, S. Huangfu, and D. Zhang, "FlierMeet: A mobile crowdsensing system for cross-space public information reposting, tagging, and sharing," *IEEE Trans. Mobile Comput.*, vol. 14, no. 10, pp. 2020–2033, Oct. 2015.

[30] T. Ludwig, C. Reuter, T. Siebigteroth, and V. Pipek, "CrowdMonitor: Mobile crowd sensing for assessing physical and digital activities of citizens during emergencies," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, 2015, pp. 4083–4092.

[31] L. Liu, W. Liu, Y. Zheng, H. Ma, and C. Zhang, "Third-eye: A mobilephone-enabled crowdsensing system for air quality monitoring," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–26, 2018.

[32] X. Xie, H. Chen, and H. Wu, "Bargain-based stimulation mechanism for selfish mobile nodes in participatory sensing network," in *Proc. 6th Annu. IEEE Commun. Soc. Conf. Sens. Mesh Ad Hoc Commun. Netw.*, Rome, Italy, 2009, pp. 1–9.

[33] Z. Ning *et al.*, "Mobile edge computing enabled 5G health monitoring for Internet of medical things: A decentralized game theoretic approach," *IEEE J. Sel. Areas Commun.*, early access, Dec. 25, 2020, doi: 10.1109/JSAC.2020.3020645.

[34] T. Luo, S. S. Kanhere, J. Huang, S. K. Das, and F. Wu, "Sustainable incentives for mobile crowdsensing: Auctions, lotteries, and trust and reputation systems," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 68–74, Mar. 2017.

[35] Z. Ning *et al.*, "Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks," *IEEE Trans. Mobile Comput.*, early access, Sep. 18, 2020, doi: 10.1109/TMC.2020.3025116.

[36] J. Nie, J. Luo, Z. Xiong, D. Niyato, and P. Wang, "A Stackelberg game approach toward socially-aware incentive mechanisms for mobile crowdsensing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 724–738, Jan. 2019.

[37] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 1701–1709.

[38] T. Luo, H.-P. Tan, and L. Xia, "Profit-maximizing incentive for participatory sensing," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Toronto, ON, Canada, 2014, pp. 127–135.

[39] L. Xiao, Y. Li, G. Han, H. Dai, and H. V. Poor, "A secure mobile crowdsensing game with deep reinforcement learning," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 35–47, 2018.

[40] E. Garman, "Cool data: Quantity and quality," *Acta Crystallographica Sect. D, Biol. Crystallogr.*, vol. 55, no. 10, pp. 1641–1653, 1999.

[41] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. Cambridge, MA, USA: MIT Press, 1994.

[42] L. Wang, D. Zhang, and H. Xiong, "effSense: Energy-efficient and cost-effective data uploading in mobile crowdsensing," in *Proc. ACM Conf. Pervasive Ubiquitous Comput. Adjunct Publ.*, 2013, pp. 1075–1086.

[43] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2015, pp. 177–186.

[44] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2015, pp. 167–176.

[45] S. Park, E. Serpedin, and K. Qaraqe, "Gaussian assumption: The least favorable but the most useful [lecture notes]," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 183–186, May 2013.

[46] P. Loiseau, G. Schwartz, J. Musacchio, S. Amin, and S. S. Sastry, "Incentive mechanisms for Internet congestion management: Fixed-budget rebate versus time-of-day pricing," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 647–661, Apr. 2014.

[47] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

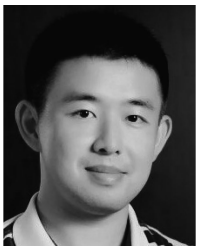
[48] L. Setti *et al.*, "Airborne transmission route of COVID-19: Why 2 meters/6 feet of inter-personal distance could not be enough," *Int. J. Environ. Res. Public Health* vol. 17, no. 8, p. 2932, 2020.

[49] A. Anfigeno. (2020). *Contact Tracing Application Sample Dataset*. https://www.kaggle.com/aleanfigeno/contact-tracing-application-sample-datasets/

Pengfei Wang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in software engineering from Northeastern University, Shenyang, China, in 2013, 2015, and 2020, respectively.

From 2016 to 2018, he was a visiting Ph.D. student with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA. He is currently an Associate Professor with the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. He has authored more than ten papers on high-quality journals and conferences, such as IEEE INFOCOM, IEEE ICDCS, IEEE Global Internet Symposium, and IEEE INTERNET OF THINGS. He also holds a series of patents in U.S. and China. His research interests are in the area of computer networking with emphasis on design, measurements, and prototype implementation of protocols and algorithms for the Internet and IoT.

Dr. Wang was awarded ACM Academic Rising Star in 2020.



Chi Lin (Member, IEEE) received the B.E. and Ph.D. degrees from Dalian University of Technology (DUT), Dalian, China, in 2008 and 2013, respectively.

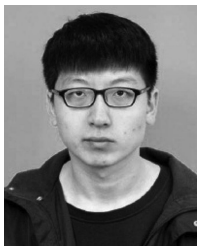
He has been an Assistant Professor with the School of Software, DUT since 2014, where he has been an Associate Professor with the School of Software since 2017. He has authored over 50 scientific papers, including INFOCOM, SECON, ICDCS, IEEE TRANSACTIONS ON MOBILE COMPUTING, and *ACM Transactions on Embedded Computing Systems*. His research interests include pervasive computing, cyber–physical systems, and wireless sensor networks.

Dr. Lin awarded ACM Academic Rising Star in 2015.

Mohammad S. Obaidat (Life Fellow, IEEE) received the Ph.D. degree in computer engineering with a minor in computer science from Ohio State University, Columbus, OH, USA, in 1986.

He is currently the Founding Dean and a Professor with the College of Computing and Informatics, University of Sharjah, Sharjah, UAE. He has worked as a Tenured Full Professor with the King Abdullah II School of Information Technology, University of Jordan, Amman, Jordan; the People's Republic of China Ministry of Education Distinguished Overseas Professor with the University of Science and Technology Beijing, Beijing, China; and an Honorary Distinguished Professor with the Amity University—A Global University, Noida, India. He has authored and coauthored about 1000 refereed technical articles—about half of them are journal articles, more than 70 books, and about 70 book chapters.

Prof. Obaidat is the Editor-in-Chief for three scholarly journals and an Editor of many other international journals. He is the founding Editor-in-Chief for *Security and Privacy* (Wiley). Previously, he was the President and a Chair of the Board of Directors and the Senior Vice President of the Society for Molding and Simulation International. He has chaired more than 160 international conferences and has given more than 160 keynote speeches worldwide.



Zhen Yu received the B.S. degree from the North China University of Water Resources and Electric Power, Zhengzhou, China, in 2020. He is currently pursuing the master's degree with the School of Computer Science and Technology, Dalian University of Technology, Dalian, China.

He has published one paper in ICPADS conference, and applied one patent. His current research interest is crowdsourcing-based urban IoT systems.

Ziqi Wei received the B.S. degree in computing science and technology from the Renmin University of China, Beijing, China, in June 2009, and the Ph.D degree from the University of Alberta, Edmonton, AB, Canada, in November 2018.

He is doing his Postdoctoral Researcher with Tsinghua University, Beijing. He has (co) authored about five papers published. His research areas include wireless sensor network, computing theory, heuristic algorithms, health ageing, and big data techniques in healthcare.

Qiang Zhang (Member, IEEE) received the B.S. degree in electronic engineering and the M.S. and Ph.D. degrees in circuits and systems from the School of Electronic Engineering, Xidian University, Xi'an, China, in 1994, 1999, and 2002, respectively.

He is currently the Dean and a Professor with the College of Computer Science and Technology, Dalian University of Technology, Dalian, China. His research interests are artificial intelligence, neural networks, DNA computing, optimization, and intelligent robots.