Social Crowdsourcing to Friends: An Incentive Mechanism for Multi-Resource Sharing

Xiaoying Gan, *Member, IEEE*, Yuqing Li, Weiwei Wang, Luoyi Fu, and Xinbing Wang, *Senior Member, IEEE*

***Abstract*— In this paper, we propose a novel game-based incentive mechanism for multi-resource sharing, where users are motivated to share their idle resources in view of conditional voluntary. Through social networking service platforms, such a crowdsourcing service fully explores the significant influence and computing potential of mobile social networks. Specifically, a combination of task allocation process, profit transfer process, and reputation updating process are involved in this sharing incentive mechanism, satisfying truthfulness, individual rationality, and robustness. To maintain the social fairness-efficiency tradeoff, we further develop a resource sharing algorithm on the basis of dominant resource fairness, revealing that the sacrifice of fairness properties is necessary for the improvement of efficiency. Real-world traces from Facebook are numerically studied, validating social fairness and efficiency of our social crowdsourcing mechanism.**

***Index Terms*— Social crowdsourcing, game-based incentive mechanism, multi-resource sharing, fairness-efficiency tradeoff.**

I. Introduction

Recent years have witnessed the explosive growth of data, which brings profound changes in people's daily life. According to the Cisco Global Cloud Index, the annual amount of global data will reach 10.4 ZB by 2019 [1]. Cloud computing is proposed as an efficient paradigm for processing the massive data, where Quality of Service (QoS) guaranteed services can be provided. However, the high maintenance costs and the failure to provide cooperation of distributed devices remain to be two drawbacks. Therefore, it's desirable to have computing service as a supplement to conventional cloud services, which may be realized by mobile and social computing.

With the progress of microelectronics, personal devices are equipped with powerful hardware and can perform complex computing tasks. However, such potential is still under utilized, which motivates us to design computing services in distributed personal devices. According to the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, it is estimated that 982 million smartphones will be shipped worldwide in 2015 [2]. Such service can be applied to collaborative sensing, collaborative localization [3] and distributed storage [4], etc., catering to users' *heterogeneous* computing demands.

The advent of social networks, such as Facebook, LinkedIn and Twitter, has greatly aroused people's enthusiasm in making social connections online. According to the statistics [5], the number of users remaining monthly active on Facebook has been tremendously growing, reaching 1.59 billion in 2015, indicating great computing capacities if resources can be shared between friends. Moreover, social networks are organized based on mutual friendship, providing users with enough incentives to share their resources.

To gather the help of friends in social networks corresponds to the idea of social crowdsourcing [6]–[9]. Crowdsourcing systems are extremely effective to perform large-scale data-processing tasks. Tang *et al.* [10] proposed a crowdsourced video streaming framework, which enables nearby mobile users to crowdsource their radio connections and resources for cooperatively video streaming. Duan *et al.* [11] utilized the collaboration of mobile devices to collect demanded data. Karger *et al.* [12] analyzed how to guarantee the reliability of crowdsourcing systems. Different from all previous works, however, we are the first to explore the benefit of computation crowdsourcing in social networks.

Even in social networks, it's more practical to consider users as *conditional* volunteers. Sharing idle resources may deplete remaining battery power sooner than expected since most mobile devices today are battery-constrained. In addition, if this sharing service is totally free, there exists an inevitable problem of "free-riding behavior", i.e., some users may simply consume idle resources of their friends and refuse to share their own. Thus it is necessary to develop an incentive mechanism to encourage users to participate in social crowdsourcing service. From the platform's perspective, how to allocate the tasks from the requesters to their idle friends *efficiently* has become an increasingly urgent issue. Game theory has naturally become a desired approach, which is effective in studying the strategic interactions process and designing optimal allocation schemes [13]. Zhang and Schaar [14] investigated the repeated game-based task allocation scheme with novel incentive mechanisms.

To this end, we propose a Vickrey-Clarke-Groves (VCG) game-based incentive mechanism for multi-resource sharing,

Manuscript received April 30, 2016; revised September 30, 2016; accepted January 24, 2017. Date of publication February 20, 2017; date of current version April 26, 2017. This work was supported in part by NSF China under Grant 61672342, Grant 61428205, Grant 91438115, Grant 61671478, Grant 61532012, Grant 61325012, Grant 61602303, and Grant 61521062, and in part by the China Postdoctoral Science Foundation.

X. Gan is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: ganxiaoying@sjtu.edu.cn).

Y. Li, W. Wang, L. Fu, and X. Wang are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: liyuqing@sjtu.edu.cn; qingniao93@sjtu.edu.cn; yiluofu@sjtu.edu.cn; xwang8@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2017.2672379

0733-8716 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

satisfying truthfulness, individual rationality and robustness. In addition to monetary reward, we also incorporate reputation reward into our incentive mechanism. Each user is assigned with a reputation recorded to reflect his/her ability. The SNS platform determines whether or not to allocate the task to the user by his/her reputation. In general, high reputation always can earn more payments for users and facilitate their requesting service afterwards.

In such multi-resource allocation, how to guarantee the fairness and efficiency has gained increasing attention. Previous research mainly focuses on the fairness issues in data centers [15]–[18]. Ghodsi *et al.* [19] first proposed DRF as a possible metric in measuring the fairness of multi-resource allocation in computer clusters, and then DRF was further implemented in packet queueing systems [20]. Joe-Wong *et al.* [21] followed up with a unifying framework for the fairness-efficiency tradeoff by introducing two families of allocations. Dolev *et al.* [22] proposed an alternative to DRF. These work make a great contribution to the fundamental knowledge and applications of DRF. However, these resource sharing schemes may not be compatible with complicated structures and characteristics of social networks, such as degrees of socialization, multiple resource pools and energy constraints.

Furthermore, we develop a DRF-based multi-resource sharing algorithm to facilitate this social crowdsourcing service. Specifically, we incorporate social connections into this sharing problem and introduce the concept of socially equivalent resource. To guarantee the social fairness in allocation, the resource of each user is constrained by the maximal dominant share, which is obtained from this DRF-based algorithm.

Our main contributions are highlighted as follows.

- We propose a novel game-based incentive mechanism for multi-resource sharing, fully realizing the social crowdsourcing service. Both users' social connections and participation incentives are involved.
- Considering users' *conditional voluntary*, we devise the incentive mechanism by utilizing the VCG game, guaranteeing truthfulness, individual rationality and robustness. Specifically, this mechanism involves a combination of task allocation process, profit transfer process, and reputation updating process.
- In view of social connection, we further develop a DRF-based multi-resource sharing algorithm, achieving the social fairness-efficiency tradeoff.
- Trace-driven simulations validate the theoretical analysis about social fairness and efficiency of our sharing incentive mechanisms.

In what follows, we provide the details of system model in Section II. In Section III, problem formulation for sharing incentive mechanism is given. To proceed, we first propose a DRF-based resource sharing algorithm in Section IV. We further design an efficient VCG-based incentive mechanism for resource sharing in Section V. Finally, discussions, simulations and conclusions are shown in Sections VI, VII and VIII, respectively.

II. System Model

In this section, we give an overview of our incentive mechanism, including a single SNS platform and N mobile users denoted by U_i ($i \in \mathcal{N} := \{1, \cdots, N\}$). Actually, there exists significant heterogeneity in users' resources, e.g., some users only have insufficient resources, while other users have much idle resources. With the SNS platform, the resources can be shared among users, i.e., users can either share idle resources or utilize their friends' resources to fulfill certain jobs. But in reality, such resource sharing always brings lots of challenges, such as power shortage or privacy leak, decreasing users' willingness to share.

A. Network and Service Model

Consider the social network consisting of multiple *ego networks*, where the "ego" refers to the central user in each friend circle [23]. For each user U_i, his/her ego network can be denoted by a graph $G_i = (V_i, E_i)$, where V_i is the set of nodes[1] in the ego network and E_i is the set of social ties between U_i and his/her friends. Besides, G_i could be either directed or undirected, depending on the context. For example, real-world traces of Facebook provided in [24] correspond to an undirected structure while the ego networks in Google+ and Twitter[2] are mostly directed. By the definition above, the entire social network can be represented by a larger graph $G = (V, E)$, where $V = \bigcup_{i \in \mathcal{N}} V_i$ and $E = \bigcup_{i \in \mathcal{N}} E_i$. Additionally, different users' ego networks can overlap with each other, i.e., they can have common friends, and we denote $\Gamma_\upsilon := \{i | \upsilon \in V_i\}$ as the set of users who have friendship with node υ. Figure 1 presents an example of such social network with 3 overlapping ego networks.

One of the most important traits of social networks is the *tie strength* between two individuals. We use $\phi_{\upsilon i}$ to indicate the tie strength between node $\upsilon \in V_i$ and user U_i (perceived by user U_i). Note that in the *directed* social network, $\phi_{\upsilon i}$ is not always equal to $\phi_{i \upsilon}$. In our model, it is assumed that people are more willing to share their idle resources with those who have stronger ties with them. Many social platforms, such as Facebook, Twitter and Google+, have developed their own ways to characterize this value depending on users' profiles, common friends, etc. Methods for quantifying the tie strength are beyond the scope of this paper.

The process of social crowdsourcing service is described as shown in Fig. 1: (1) A certain node first submits a computing job and the corresponding payment to the SNS platform[3]; (2) The platform selects qualified users as winning set according to user reputation, segments the job into multiple small tasks, and distributes them to the selected users through social network; (3) The users utilize their idle resources (e.g., CPUs, memory and storage, etc.) to accomplish tasks

[1] In the rest of this paper, we will use "nodes" and "friends" interchangeably. The same is with "user" and "ego".

[2] The datasets of Google+ and Twitter are publicly accessible data. They are available at http://snap.stanford.edu/data/.

[3] We assume that the SNS platform is not involved in the profit transfer process between users.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

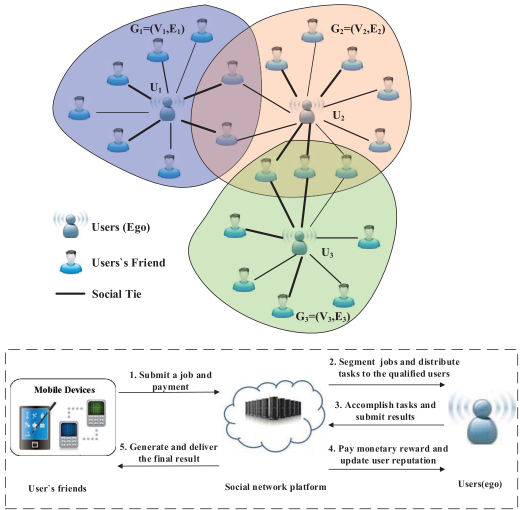


Fig. 1. General sharing incentive mechanism.

allocated and feed back intermediate results to the platform; (4) The platform pays monetary reward to the selected users and updates user reputation according to their performance; (5) The platform further compiles the data to generate the final result of the original problem and then delivers the result to the node. In this way, users can obtain monetary and reputation reward by sharing idle resources with their friends through the implementation of this mechanism.

Actually, our system model can be implemented in a variety of current real-world systems. Our first example is Open Garden, one typical crowdsourced user-provided networking system [25]. With MeshKit platform, users can post the resource information about their radio connection resources such as Bluetooth, peer-to-peer WiFi, and data resources such as news and files. Accordingly, users can share these resources with the requesting friends without an Internet connection. In addition, the second example is Waze, one collaborative localization system [26]. With Facebook platform, users can post and further share real-time traffic information (i.e., information resources) in their friend circle, realizing the vehicle dynamic navigation service.

B. Workload and Multi-Resource Model

In our proposed mechanism, each submitted job is segmented into multiple small tasks requiring the same amount of resources (or consuming the same cost). The number of allocated jobs denoted by W, is very large and it is acceptable not to enforce W to be integer-valued. Note that the above segmenting operation is a relatively mature technique in classical MapReduce systems, which can be directly applied in our mechanism.

Through the SNS platform, users are willing to share their idle resources to help their friends fulfill as many tasks as possible until the job is finished finally within the sharing period.[4] Each task demands a combination of multi-type resources. Formally, suppose there are m types of resources, and the resource requirement for per task of node $\upsilon \in V$ is denoted by a vector $\vec{r}_\upsilon = (r_\upsilon^{(1)}, r_\upsilon^{(2)}, \cdots, r_\upsilon^{(m)})$, where $r_\upsilon^{(k)}$ $(\forall k \in M =: \{1, \cdots, m\})$ is his/her demand for the type-k resource to process one task. The high heterogeneity of mobile devices also requires to characterize the resources of each node more specifically and flexibly. Hence, we introduce a multi-resource vector $\vec{c}_i = (c_i^{(1)}, c_i^{(2)}, \cdots, c_i^{(m)})$ to denote the *idle* resources of user U_i within the sharing period. Note that resources are referred as "idle" if the system can still operate normally when shared with others.

Denote an $N \times |V|$ matrix X as the workload allocation, whose element $x_{i\upsilon} \in R^N$ $(\forall i \in \mathcal{N}, \upsilon \in V)$ indicates the maximum amount of tasks from user U_i to node υ. For node $\upsilon \notin V_i$ $(i \in \mathcal{N})$, we have $x_{i\upsilon} = 0$. Under this definition, the amount of *normalized* resources that U_i shares with node υ can be represented as $\vec{\alpha}_{i\upsilon} := (\alpha_{i\upsilon}^{(1)}, \cdots, \alpha_{i\upsilon}^{(m)})$, where $\alpha_{i\upsilon}^{(k)} = \frac{r_\upsilon^{(k)} x_{i\upsilon}}{c_i^{(k)}}$ $(\forall k \in M)$.

C. Monetary and Reputation Reward Model

In such incentive mechanism, users can receive the payment by sharing their idle resources with friends, and they also needs to pay for each task their friends help them to fulfill. Each user U_i will consume α_i cost of idle resources to help his/her friend denoted as $\upsilon \in V_i$ accomplish tasks. Since the earned payments can be used for fulfilling their own tasks in the future, users are willing to share even though it may cost them some additional payment.

In addition, to encourage resource sharing and address "free-riding" problem, we incorporate user reputation into our mechanism, which is an indicator of user historical performance and reliability. Denote the reputation of user U_i as p_i, which is a natural number from the finite set $\mathcal{P} = \{0, 1, \cdots, P\}$. Here the reputation represents the social status of users. High reputation reflects good social status, meaning that the user has done very well in finishing tasks. As an administrator, the SNS platform needs to check user reputation to prevent job requests from malicious or low-reputation nodes, making it safer for users to share idle resources. The details of reputation scheme will be discussed later.

III. Problem Formulation

A. Incentive Ploblem

As mentioned previously, the social crowdsourcing service is enabled by motivating users to share and allocate idle resources reasonably. We assume that the SNS platform is authorized to manage nodes' payment and allocate the tasks to users. When any friend $\upsilon \in V_i$ submits a job request, user U_i gains the profit by helping friend υ fulfill the tasks, and then friend υ will pay user U_i for idle resources. In this process, users serve as the players with the goal of maximizing their own utility. In addition to monetary reward, each user is assigned with a reputation reflecting his/her ability.

Under the proposed incentive mechanism, we try to address the following problems:

[4]The sharing period is set by the user when he/she takes the service and decides to share the resources.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

• From the platform's perspective, how to allocate the tasks from the requesters to users *efficiently* (or how to guarantee the truthfulness of users).

• From the users' perspective, they need to decide how to allocate their idle resources *fairly* (or how to cope with social connections) and *efficiently* (or how to achieve *workload balance*).

The key to solving the above problems is to resolve the conflict between social connections and participation incentives. In the following, we will formally characterize these problems.

The objective of the platform is to improve overall service performance with the minimum cost, by distributing the tasks to users reasonably. Denote the outcome for task allocation as $y = \{y_i | i \in \Gamma_v\}$, where $y_i \in R^N$ is the number of tasks allocated to user U_i within limited time T_v and $\sum_{i\in\Gamma_v} y_i = W$. We suppose that the number of tasks that user U_i is capable of finishing within the time limit is z_i. To evaluate the performance of each user's service, we define *cost-effectiveness ratio* as $s_i = \alpha_i / p_i$, where α_i and p_i are the corresponding cost and reputation. For a given allocated task, the higher user reputation is, the more effectively the task will be finished, meaning that the lower cost-effectiveness ratio becomes. Here the effectiveness mainly characterizes how the allocated tasks can be finished within time limitation. Intuitively, choosing a user with lower reputation will increase the probability of failing to finish the tasks, which may deteriorate service performance and bring more loss.

User U_i will charge different payments for their idle resources due to the difference in cost and completed tasks. Hence, the aggregated cost-effectiveness ratio of the service requested by node v can be calculated as

$$C_v = \sum_{i\in\Gamma_v} y_i s_i. \tag{1}$$

In our proposed incentive mechanism, the task allocation (performed by the SNS platform) is based on a "social decision", minimizing the aggregated cost-effectiveness ratio (or equivalently, maximizing the aggregated utility). Accordingly, the social decision problem is to find an optimal allocation $y^* = \{y_i^* | i \in \Gamma_v\}$ that

$$\begin{aligned} \mathrm{y}^* = \arg\min_{y} & \sum_{i\in\Gamma_v} y_i s_i \\ s.t. & \sum_{i\in\Gamma_v} y_i = W. \\ & 0 \le y_i \le z_i, \quad y_i \in R^N. \end{aligned} \tag{2}$$

Moreover, this problem is similar to the knapsack and the capacitated lot-size problems, which have been proved to be $\mathcal{NP}$-hard. However, we can solve this problem easily through the implementation of a dynamic programming approach within pseudopolynomial time [28]. To facilitate reading, we list the major notations in Table I.

B. Our Solution

In essence, our incentive mechanism involves a combination of task allocation process, profit transfer process, and reputation updating process.

TABLE I
MAJOR NOTATIONS

V_i	the set of nodes in user U_i's ego network
Γ_v	the set of users who have social ties with node v
ϕ_{vi}	the tie strength between v and U_i (perceived at U_i)
$r_v^{(k)}$	node v's demand for resource k to process one job
$c_i^{(k)}$	user U_i's idle capacity of resource k
x_{iv}	the maximum amount of allocated tasks from U_i to v
$\alpha_{iv}^{(k)}$	the real share of resource k that U_i shares with v
α_i	the real cost of idle resource for user U_i
p_i	the reputation of user U_i

To guarantee the effectiveness of social crowdsourcing service, we aim at designing a VCG game-based incentive mechanism for multi-resource sharing, so as to stimulate users' willingness to share idle resources and provide high-quality task finishing. As an applied branch of game theory, the VCG mechanism seeks socially optimal solutions and possesses several desirable economic properties, especially for task allocation [13], [28], [29].

In VCG game, user U_i in Γ_v will make a joint strategy including two aspects. That is (i) the cost $\hat{\alpha}_i$ that the task allocated by platform takes to be completed, and (ii) the promised sharing time calculated with which the promised number of tasks is $\hat{z}_i$. Thus we denote the joint strategy as $(\hat{\alpha}_i, \hat{z}_i)$. For personal profit, these friends may misreport the real cost α_i and real sharing time characterized by z_i.

As mentioned previously, the above social decision problem involves both user social connection and participation incentive, respectively. The latter can be represented by the definition of s_i in the formulation of optimal allocation y^*. Considering user social connection, the tasks need to be allocated reasonably and effectively in the context of social networks. Actually, this is different from traditional task allocation schemes in mobile networks. In particular, it is not reasonable to only take reward into account in allocating resources to friends, with user social connection ignored. To guarantee the social fairness in allocation, we first study a DRF-based resource sharing algorithm, where the maximal dominant share is obtained to limit the resources allocated to each friend.

IV. DRF-Based Multi-Resource Sharing Algorithm

A. Fairness and Efficiency Measures in Social Networks

We adopt DRF as the basis of fairness measure in multi-resource scenario. Briefly, DRF achieves the max-min fairness of users' dominant shares (i.e., the shares of their most highly demanded resource among all types) [18]–[21]. As shown in [19], DRF has many properties and it is easy to implement, which yields low overheads and is particularly suitable for large-scale social networks.

However, the heterogeneity of users' tie strength with friends could make traditional DRF undesirable. Consider a simple case where a certain node A has 1GB ROM that can be shared with two users B and C. According to the philosophy of

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

DRF, the fairest way for B and C is to receive an equal amount of resources. As for social networks, however, allocation above could be highly unfair since B and C may differ in tie strength with A. For example, if B is A's boyfriend while C is only a distant schoolmate of A, B will lose the sharing incentive and envy C under such an equal allocation, which violates the economic propositions of DRF. From A's perspective, she may also prefer to allocate more resources to B and regard a biased resource distribution as a fairer result.

Therefore, fairness in social networks must be established on tie strength, and then we introduce a concept called *Socially Equivalent Resources* (SER) to quantize this feature. Generally, SER reflects the amount of allocated resources that users share with friends. Specifically, we denote $\pi(\alpha_{iv}^{(k)}, \phi_{vi})$ as the amount of type-k SER share that U_i shares with node v, which is a function of the *real allocated resources* and their tie strength. For simplicity of analysis, we assume $\pi(x, y)$ is linear of x. Besides, $\pi(\alpha_{iv}^{(k)}, \phi_{vi})$ is monotonically increasing with $\alpha_{iv}^{(k)}$ so that more *real* allocations can correspond to more SER shares, and this function also decreases with ϕ_{vi} to ensure that users share more *real* resources with the friends having stronger ties with them. One simple form is $\pi(\alpha_{iv}^{(k)}, \phi_{vi}) = \frac{\alpha_{iv}^{(k)} \sum_{i\in\Gamma_v} \phi_{vi}}{\phi_{vi}}$.

The concept of *dominant resources* is naturally extended to the *dominant SER*. Let

$$\mu_{i,v} = \max_{k\in M} \pi\left(\frac{r_v^{(k)}}{c_i^{(k)}}, \phi_{vi}\right) \tag{3}$$

denote the dominant SER share allocated by user U_i to process one task of node v, and then $\mu_{i,v} x_{iv}$ is the total dominant SER share allocated by U_i to v.

Based on the discussions above, we can further formally define DRF in social networks.

Definition 1 (DRF in Social Networks): A multi-resource allocation X satisfies DRF in social networks if it is *feasible* and for any $i \in \mathcal{N}$ and $v \in V$, user U_i cannot allocate higher dominant SER share to node v while maintaining feasibility without decreasing the dominant SER share to any other node v' $(v' \in V_i)$ whose $x_{iv'}\mu_{i,v'} \le x_{iv}\mu_{i,v}$.

Besides the consideration for fairness, from the global view, we also wish the users to allocate their resources in an *efficient* manner, or to maintain a certain degree of workload balance. Similar to DRF, our metric of interests is *Dominant Workload Imbalance* denoted by θ, i.e.,

$$\theta = \max_{i,l\in V} \frac{\tau_i}{\tau_l}, \tag{4}$$

where $\tau_i = \max_{k\in M} \sum_{v\in V_i} \alpha_{iv}^{(k)}$ reflects the dominant workload at user U_i. Compared to other workload measures (e.g., the average workload of all types [22]), the dominant workload is particularly important since it reflects the bottleneck of a system. Therefore, our scheme should lower the dominant workload imbalance as much as possible.

B. Resource Sharing Algorithm

Before showing this algorithm, a concept called *bottleneck resource* needs to be introduced.

Definition 2 (Bottleneck Resource): Resource k is the bottleneck resource shared by user U_i with node v if $\sum_{v\in V_i} \alpha_{iv}^{(k)} = 1$ and $x_{iv}\mu_{i,v} \ge x_{iv'}\mu_{i,v'}$ for any other node v' who demands resource k at U_i.

Theorem 1: The necessary and sufficient condition for an allocation X to achieve DRF in social networks is that each user U_i $(i \in \mathcal{N})$ has a *bottleneck resource* shared with node $v \in V_i$.

Proof: See Appendix A. □

According to Theorem 1, any algorithm to achieve DRF in the proposed mechanism must ensure that each user has at least one bottleneck resource shared with any node in his/her ego network. Algorithm 1 is proposed to realize this goal.

Algorithm 1 DRF-Based Resource Sharing Algorithm

1: Initialize $x_{iv} = 0$ $(\forall i \in \mathcal{N}, v \in V)$;
2: $S_i = V_i$ and $K_i = \emptyset$ $(\forall i \in \mathcal{N})$;
3: $L_i^{(k)} = c_i^{(k)}$ $(\forall k \in M, i \in \mathcal{N})$;
4: **for** each $i \in \mathcal{N}$ **do**
5: **repeat**
6: $x_{iv} \leftarrow x_{iv} + \Delta_{iv}$ $(\forall v \in S_i)$;
7: $K_i \leftarrow K_i \bigcup \{\widehat{k}\}$;
8: $L_i^{(k)} \leftarrow L_i^{(k)} - \sum_{v\in S_i} r_v^{(k)} \Delta_{iv}$ $(\forall k \in M)$;
9: $S_i = \{v | \forall k \in K_i, r_v^{(k)} = 0, v \in S_i\}$;
10: **until** $S_i = \emptyset$
11: **end for**

Briefly, for each user $U_i, i \in \mathcal{N}$, Algorithm 1 allocates an equal dominant SER share to all nodes who have social ties with U_i and do not use any saturated resources until U_i have a bottleneck resource at all nodes. In each "Repeat-Until" loop of user U_i, K_i is the set of resources that have been used up, S_i is the set of friends who have social ties with U_i and do not demand any saturated resources $k \in K_i$, and $L_i^{(k)}$ is the amount of left idle resources of type k at user U_i at the beginning of that loop. Moreover, Δ_{iv} in step 6 is the increment of x_{iv} in that loop, i.e.,

$$\Delta_{iv} = \frac{1}{\mu_{i,v}} \min_{k\in M} \left(\frac{L_v^{(k)}}{\sum_{v'\in S_i} \frac{r_v^{(k)}}{\mu_{i,v'}}}\right). \tag{5}$$

In step 7, $\widehat{k}$ is the type of resources which are exhausted in that loop, which is shown as

$$\widehat{k} = \arg\min_{k\in M} \left(\frac{L_i^{(k)}}{\sum_{v'\in S_i} \frac{r_v^{(k)}}{\mu_{i,v'}}}\right). \tag{6}$$

We will present a further explanation of equations (5) and (6) in the proof of Theorem 2.

Theorem 2: Algorithm 1 can achieve the DRF in social networks.

Proof: See Appendix B. □

As for the time complexity of Algorithm 1, the number of "Repeat-Until" loops will be at most m since S_i must be empty when all m types of resources are drained out. Since we focus on how time complexity scales with the network scale and the

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

number of users, the algorithm is of $O(1)$ at each user, and thus the overall time complexity of Algorithm 1 is $O(|V|)$. Low time complexity makes Algorithm 1 very suitable for handling large-scale social networks.

C. Property Analysis

We first investigate economic properties of Algorithm 1.

Definition 3 (Pareto Efficiency): A resource allocation satisfies Pareto Efficiency (PE) if it is impossible to increase any user's dominant SER share with any node without decreasing the dominant SER shares with any other nodes.

Definition 4 (Sharing Incentive): An allocation holds Sharing Incentive (SI) if the node is better off than they would be under an equal SER share of all resources at any user.

Definition 5 (Envy-Freeness): A resource distribution pattern satisfies Envy-Freeness (EF) if no node prefers the allocation of others from any user.

Definition 6 (Strategy-Proofness): The Strategy-Proofness (SP) is maintained if the node has no incentives to submit untruthful demands of multi-type resources from any user.

According to the definitions above, we can further obtain the following propositions.

Proposition A.1: The allocation X obtained through Algorithm 1 satisfies Pareto Efficiency, Sharing Incentive, Envy-Freeness and Strategy-Proofness.

Proof: See Appendix C. □

Proposition A.2: Algorithm 1 achieves the Dominant Workload Balance ($\theta = 1$).

This proposition holds because each user has at least one saturated resource by Theorems 1 and 2. Hence, the dominant workload of each user is 100%, and then the dominant workload imbalance θ achieves the minimum value 1.

As for the workloads of each single-type resource, we have the following result.

Proposition A.3: For a specific resource type k, the workload imbalance across all users is upper-bounded by $\max_{i\in\mathcal{N}} \sum_{\upsilon\in V_i} \frac{A_{i\upsilon} c_i^{(k)}}{r_\upsilon^{(k)}}$, where $A_{i\upsilon} = \max_{k\in M} \frac{r_\upsilon^{(k)}}{c_i^{(k)}}$.

Proof: See Appendix D. □

Finally, we investigate the *efficiency* of Algorithm 1. In the multi-resource settings, efficiency can be represented in different ways, such as the average resource utilization [22] and the sum of dominant shares [18], [21]. In this paper, we mainly consider the latter one. The following proposition bounds the efficiency obtained by Algorithm 1.

Proposition A.4: Let $B = \min_{i\in\mathcal{N}} \frac{\min_{\upsilon\in V_i} \pi(1,\phi_{\upsilon i})}{\max_{\upsilon\in V_i} \pi(1,\phi_{\upsilon i})}$. The lower bound for the sum of users' dominant SER shares with all their friends is $\frac{B}{m}$, i.e.,

$$\sum_{i\in\mathcal{N}} \sum_{\upsilon\in V_i} \mu_{i,\upsilon} x_{i\upsilon} \geq \frac{B}{m}.$$

Proof: See Appendix E. □

D. Improving the Efficiency

Proposition A.4 shows that the efficiency of Algorithm 1 is relatively low, which is the inherent drawback of DRF [18], [21]. Although Joe-Wong *et al.* [21] provided metrics for the fairness-efficiency tradeoff, it may not be suitable for large-scale social networks due to computational intractability. Thus, we introduce the concept of ϵ-DRF so as to improve the efficiency while maintaining the implementation.

Definition 7 (ϵ-DRF): A multi-resource allocation X satisfies ϵ-DRF if for any $i \in \mathcal{N}$ and $\upsilon \in V$, node υ cannot increase his/her dominant SER share at user U_i without decreasing that of any other node υ' in V_i whose $x_{i\upsilon'}\mu_{i,\upsilon'} \leq x_{i\upsilon}\mu_{i,\upsilon} - \epsilon$.

This definition resembles that of DRF except that we relax the condition $x_{i\upsilon'}\mu_{i,\upsilon'} \leq x_{i\upsilon}\mu_{i,\upsilon}$ to $x_{i\upsilon'}\mu_{i,\upsilon'} \leq x_{i\upsilon}\mu_{i,\upsilon} - \epsilon$. System efficiency varies under different values of ϵ. Specifically, $\epsilon = 0$ achieves the exact DRF while $\epsilon = \max_{\upsilon,i} \pi(1,\phi_{\upsilon i})$ is efficiency-optimal. However, ϵ-DRF improves the efficiency at the cost of a certain degree of fairness, especially some economic properties. Before detailed analysis, we first investigate how to obtain ϵ-DRF in social networks. Similar to DRF, there is a sufficient and necessary condition for ϵ-DRF.

Theorem 3: An allocation X maintains ϵ-DRF if and only if each user U_i $(i \in \mathcal{N})$ has an *ϵ-bottleneck resource* shared with any node $\upsilon \in V_i$.

The concept of ϵ-bottleneck resources is similar to Definition 2 except that we modify $x_{i\upsilon}\mu_{i,\upsilon} \geq x_{i\upsilon'}\mu_{i,\upsilon'}$ to $|x_{i\upsilon}\mu_{i,\upsilon} - x_{i\upsilon'}\mu_{i,\upsilon'}| \leq \epsilon$. The proof of Theorem 3 directly follows that of Theorem 1, so we do not present it here. Based on Theorem 3, we seek to find an allocation X that leaves each user an *ϵ-bottleneck resource* at each node in his/her ego network, which is equivalent to the following linear programming (LP) problem, i.e.,

$$\begin{aligned} &\max_{X\geq 0} \ \sum_{i\in\mathcal{N}} \sum_{\upsilon\in V_i} \mu_{i,\upsilon} x_{i\upsilon} \\ &s.t. \ \sum_{\upsilon\in V_i} x_{i\upsilon} r_\upsilon^{(k)} \leq c_i^{(k)}, \quad \forall i \in \mathcal{N},\ k \in M, \\ &\qquad |x_{i\upsilon}\mu_{i,\upsilon} - x_{i\upsilon'}\mu_{i,\upsilon'}| \leq \epsilon, \quad \forall i \in \mathcal{N},\ \upsilon, \upsilon' \in V_i. \end{aligned} \tag{7}$$

Solving this LP problem is very easy through implementation of some polynomial-time LP algorithms [27]. The correctness of such a transformation is as follows. At first, the solution to this LP problem ensures that a certain user U_i can have at least one saturated resource $\widehat{k}$ shared with any node $\upsilon \in V_i$; otherwise we can increase the value of the objective function in (7) by allocating more jobs to υ using unsaturated resources. However, this contradicts the optimality of the solution to (7). In addition, the second constraint of the LP problem above ensures $|x_{i\upsilon}\mu_{i,\upsilon} - x_{i\upsilon'}\mu_{i,\upsilon'}| \leq \epsilon$ holds for υ and any other node υ' in V_i. Consequently, the solution to the LP problem gives each user an *ϵ-bottleneck resource* at each node and further achieves ϵ-DRF in social networks according to Theorem 3.

We can validate the above propositions by the previous example with $\epsilon = 0.2$. Under our scheme, the result is that A receives $\langle 68.6\%, 51.4\% \rangle$ of all ROM and CPU resources while B gets $\langle 8.1\%, 48.6\% \rangle$. Thus B's dominant SER share is below 50% and A has a strictly higher share of all resources than B, violating the Sharing Incentive and Envy-Freeness. However, when user B untruthfully submits his/her demands

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

as ⟨5GB ROM, 3-unit CPU⟩. The result becomes $\langle 45.5\%, 34.1\% \rangle$ for A and $\langle 54.5\%, 65.4\% \rangle$ for B, where B's dominant SER share increases and the Strategy-Proofness does not hold.

Proposition B.1: ϵ-DRF is Pareto Efficient.

The proof of this property resembles to that of Proposition A.1, so we omit it for the brevity.

Proposition B.2: ϵ-DRF achieves the Dominant Workload Balance.

This conclusion can be directly derived from the discussion about the correctness of our LP formulation. Unfortunately, some other economic properties of fairness do not always hold for any $\epsilon > 0$. In fact, such a result is not unexpected since Parkes *et al.* [17] have demonstrated that it is impossible to improve the sum of dominant shares while maintaining any one of the following three properties. In other words, there exists a tradeoff between fairness and efficiency.

Proposition B.3: ϵ-DRF does *not* always satisfy Sharing Incentives for any $\epsilon > 0$.

Proposition B.4: ϵ-DRF does *not* always satisfy Envy-Freeness for any $\epsilon > 0$.

Proposition B.5: ϵ-DRF does *not* always maintain Strategy-Proofness for any $\epsilon > 0$.

V. VCG Game-Based Incentive Mechanism for Multi-Resource Sharing

A. Reputation Updating Policy

To make task allocation with high quality and efficiency, a differential reputation scheme is performed by the platform: users with higher reputation will be given higher chance to receive allocated tasks and further obtain more payments.

Assume that there exists a threshold reputation $p_{th} \in \mathcal{P} = \{0, 1, \cdots, P\}$. Specifically, if user reputation is higher than p_{th} at the beginning of a time slot, the platform will deliver a task to user at this time slot. We call this kind of workers *active worker*. While if the reputation is lower than p_{th}, the worker won't receive any task. We call this kind of workers *isolated worker*. Since the model is dynamic, user reputation changes at the end of every time slot. Hence, active workers have the chance to be isolated when failing to accomplish tasks. While isolated workers also have chance to get over the threshold p_{th} and further increase the reputation. The updating rule of user reputation can be expressed as follows:

$$p_i = \begin{cases} \min\{P, p_i + 1\} & \text{if } k_i p_i \geq p_{th} \\ p_i - 1 & \text{if } k_i = 0 \text{ and } p_i \geq p_{th} \\ 0 & \text{if } k_i = 0 \text{ and } p_i = p_{th} \\ p_i + 1 & \text{if } p_i \leq p_{th}. \end{cases} \tag{8}$$

where k_i is a binary variable indicating whether user U_i has completed all the tasks ($k_i = 0$ means failing to finish all the tasks, and vice versa).

Under this reputation scheme, when a worker is active and finishes his/her task at the last time slot, his/her reputation increases by 1 while not exceeding P after one time slot. While if he/she does not finish the task, his/her reputation decreases by 1. When a worker's reputation falls to p_{th} and he/she does not finish a task last time, the platform will set the reputation to 0. An isolated worker, on the other hand, increases his/her reputation by 1 each period until the reputation reaches p_{th} and he/she is activated again.

B. VCG Game-Based Incentive Mechanism

At a certain time, the SNS platform receives sharing service request as well as the period of limited time T_υ from node υ. After job division, a total of W tasks are to be done. In addition, the platform needs to determine a winning set $\Gamma'_\upsilon \subseteq \Gamma_\upsilon$ according to user reputation. Specifically, each user U_i in winning set Γ'_υ should keep active so as to help fulfill these tasks with high efficiency. Hence, we obtain

$$\Gamma'_\upsilon = \{i | p_i \geq p_{th}, i \in \Gamma_\upsilon\}. \tag{9}$$

Accordingly, as for node υ, there are $|\Gamma'_\upsilon|$ users altogether who would offer idle resources. For each user U_i who has received allocated tasks, i.e., $i \in \Gamma'_\upsilon$, we assume that his/her helping node υ fulfill the unit task will incur α_i cost of idle resource, which is private information of U_i.

In a heterogeneous social network, each selected user will allocate a different amount of resource. In order to finish the job more efficient (using less time), it is intuitive that we should allocate those who can allocate more resources more tasks. In order to achieve DRF fairness, we assume that users allocate their idle resources according to Algorithm 1 introduced in Section IV. It is worth noting that once the job request of node υ is satisfied, resource allocation will end so as to improve efficiency. Suppose the number of tasks that user U_i is capable of finishing within the time limit is z_i, which is associated with the resource allocation in Algorithm 1.

For user U_i who has received tasks, he/she will obtain the corresponding payment from node υ once successfully returning the result to the platform. Based on the optimal task allocation y^*, a transfer which presents the marginal contribution of users to the society is imposed to guarantee the truthfulness of users. Here, the truthfulness has the following two meanings. One is that the user charges with a reasonable price according to the cost of providing idle resources. The other is that the user holds to supply idle resources until the allocated tasks are all done. In particular, we denote the transfer associated with user U_i as τ_i, i.e.,

$$\tau_i = \left[\min_{y_l \leq z_l} \sum_{l \in \Gamma'_\upsilon / i} y_l s_l \right] - \left[\sum_{l \in \Gamma'_\upsilon / i} y_l^* s_l \right]. \tag{10}$$

Note that the first part in the summation is the minimum aggregated costs that other users can derive if user U_i does not participate in this VCG mechanism. The second part is sum of aggregated costs of the other users except user U_i under the optimal task allocation in the presence of U_i. Thus, equation (10) represents the marginal contribution of user U_i to reducing the total cost of the optimal allocation. It is easy to see that τ_i is always nonnegative.

As mentioned in reputation scheme design, choosing a user with higher reputation will increase the probability of finishing the tasks successfully, which may bring more profit. Users with

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

higher reputation will receive more payments than those with lower reputation upon the completion of equal tasks. Thus for each user U_i, we incorporate the reputation p_i into the his/her reward as a weighting factor, providing the incentive to improve user reputation. In case that users may not be truthful, there is another punishment scheme for those who quit before finishing the tasks. Assume that user U_i finally finishes $y_i{}'$, so the final payment to U_i can be expressed as the difference between the corresponding reward for the true amount of finished tasks minus a penalty γ in punishment scheme, which is determined by how critical the tasks demand to be met. Therefore, the payment to user U_i after he/she submits the results to the platform or after the time limitation T_v is $u_i = p_i \cdot \tau_i|_{z_i=y_i'} - \kappa\gamma$, where $\tau_i|_{z_i=y_i'}$ is the transfer of user U_i when z_i equals $y_i{}'$. Similar to the definition of k_i, κ is also a binary variable indicating whether user U_i has completed all the tasks within time limitation T_v ($\kappa = 1$ means failing to finish all the tasks, and vice versa). On the other hand, the total cost of user U_i for accomplishing the allocated tasks can be characterized as $c_i = y_i^*\alpha_i$.

Based on the above fair resource allocation, user U_i will make a joint strategy $(\hat{\alpha}_i, \hat{z}_i)$. Accordingly, we define the utility of user U_i, denoted by f_i, as his/her profit. In detail, the utility of U_i is $u_i - c_i$ if he/she receives allocated tasks, i.e., $i \in \Gamma_v'$, and 0 otherwise. Accordingly, the utility function of each user can be formulated as

$$f_i(\hat{\alpha}_i, \hat{z}_i) = \begin{cases} u_i - y_i^*\alpha_i & \text{if } i \in \Gamma_v', \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

We described the detailed VCG game-based incentive mechanism in Algorithm 2. Whether or not a user is selected only depends on his/her joint strategy $(\hat{\alpha}_i, \hat{z}_i)$ and reputation p_i. The final optimal task allocation y^* can also be regarded as users' available resource constrained by z_i. min_value is a temp value in "repeat-until" loop to store the aggregate cost-effectiveness ratio. User utility can be calculated as the difference between payment and real cost.

Algorithm 2 VCG-Based Incentive Mechanism

Require:
User set Γ_v; Bid $\{(\alpha_i, z_i)|i = 1, 2, ..., |\Gamma_v|\}$;
Reputation $\{p_i|i = 1, \cdots, |\Gamma_v|\}$; W;
Ensure:
Winning set Γ_v'; Available resource y^*;
User utility $f_i(\alpha_i, z_i)$;
Initialize $y_i^* = 0$; $f_i = 0$; $min\_value = 0$; $\Gamma_v' = \emptyset$;
$s_i \leftarrow \alpha_i / p_i$;
repeat
  $i \leftarrow arg\,\min_i\{s_i|\ p_i \ge p_{th},\ i \in \Gamma_v \setminus \Gamma_v'\}$;
  $y_i^* \leftarrow min(z_i, W)$;
  $W \leftarrow W - y_i^*$;
  $min\_value \leftarrow min\_value + y_i^* s_i$;
  $\Gamma_v' \leftarrow \Gamma_v' \bigcup \{i\}$;
until $W = 0$
for $i \in \Gamma_v'$ **do**
  $\tau_i \leftarrow \left[\min_{y_l \le z_l} \sum_{l \in \Gamma_v'/i} y_l s_l\right] - \left[\sum_{l \in \Gamma_v'/i} y_l^* s_l\right]$;
  $f_i(\hat{\alpha}_i, \hat{z}_i) \leftarrow p_i \cdot \tau_i - y_i^*\alpha_i$;
end for

C. Property Analysis

Several favored properties of VCG mechanism [13] are expected to satisfy:

Definition 8 (Truthfulness): Truthfulness is satisfied when the dominant strategy of user U_i is to submit true cost α_i and sharing time characterized by z_i, i.e., $f_i(\alpha_i, z_i) \ge \max\{f_i(\hat{\alpha}_i), f_i(\hat{z}_i)\}, \forall i \in \Gamma_v'$.

Definition 9 (Individual Rationality): Individual Rationality holds when user utility is not less than 0, i.e., $f_i(\alpha_i, z_i) \ge 0$, $\forall i \in \Gamma_v'$.

Next, we will analyze the properties of our proposed VCG game-based mechanism.

Proposition C.1: The VCG game-based incentive mechanism is *truthful*.

Proof: See Appendix F. □

Proposition C.2: The VCG game-based incentive mechanism is *individual rational*.

Proof: See Appendix G. □

Remark: Propositions C.1 and C.2 demonstrate our proposed VCG game-based incentive mechanism can guarantee the desired economic properties, i.e., Truthfulness and Individual Rationality. In that way, users will have the incentives to participate and submit true cost and sharing time, making it efficient to allocate tasks. Accordingly, without malicious users considered, all the users should have a reputation close to 1.

Proposition C.3: The VCG game-based incentive mechanism is robust to the uncertainty of sharing time.

Proof: See Appendix H. □

Remark: Proposition C.3 demonstrates that for a given sharing period, the platform can make user utility always below zero by choosing the value of penalty γ. Therefore, users will lose the incentive to submit a longer sharing time.

VI. Discussions

A. Impact of Wireless Environments

Unlike wired equipment, network resources possessed by mobile devices are spatially and temporally diverse due to the location-dependent and time-varying properties of wireless links. For example, the network performance of mobile devices in an indoor environment with many obstacles tends to be poorer than that in an open area. Users may measure and post their idle resources on the platform. Such a method is simple but can yield rough estimation of available resources within the sharing period, making it suitable for some dynamic environments. However, since the sharing period could be relatively short in our system, the above estimation may be enough to handle most practical scenarios. Besides, user mobility further strengthens the uncertainty of network resources in wireless environments. In our model, such diversity is reflected in the heterogeneity of network resources in different mobile devices.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

B. Impact of Power Consumption

In our model, users can connect to the social network through either wired equipment (e.g., PCs and laptops) or mobile devices, wherein significant difference exists among them in some aspects. Mobile devices are usually driven by batteries, where the power supply is very limited. Generally, failing to make rational use of energy may cause undesirable and even fatal consequences to users. For any rational user, the sharing resources will impose a power constraint, i.e., the maximum power consumption allowed for users' computing jobs. Such a constraint is mainly dependent on the residual energy but may also account for other factors like the exclusive or emergent usage of mobile devices. But for wired equipment, the power constraint is very large, which can be neglected in our model. Hence, it's worthy mentioning that our mechanism is completely compatible with the heterogeneous power consumption patterns.

C. Impact of User Heterogeneity

In this section, we study the impact of user heterogeneity in tie strength and reputation. Each user specifies a certain tie strength and reputation. Thus our sharing mechanism based on user heterogeneity characterizes a more realistic scenario. Moreover, in social networks, traditional definitions about fairness and efficiency are not applicable any more. The heterogeneity in tie strength and reputation constitutes the essence of our incentive mechanism and we can explore economic potential from it. Specifically, our mechanism can encourage users with strong social relationship and high reputation share idle resources, which are validated by simulation results.

VII. Case Study

In this section, we will conduct trace-driven simulations using the dataset from Facebook[5] [24]. The data we obtained consist of ten users' ego networks which partially overlap with each other. Note that the network model is an undirected weighted graph, where the edge weight (i.e., tie strength) is determined by 42 features extracted from their profiles (e.g., eduction, hobbies, etc.) and the number of common friends.

As for the nodes in ego networks, we collect the profiles of 10 popular mobile devices, including Apple iPhone 5, Nokia Lumia 920, Samsung N7100, etc., and assume each node is a random type among them. To facilitate the simulations, three major types of resources are of our interests, i.e., CPU, RAM and ROM. The amount of idle resources at each user is uniformly distributed within [10%, 70%]. Besides, each user will randomly impose a power constraint within [500, 1000]mW unless it is specified. Each node's resource demand follows the uniform distribution within [0.5, 3] units of CPU resources, [100, 800]MB RAM and [1, 8]GB ROM.

A. Social Fairness and Efficiency

It is worth studying the fairness/efficiency of our DRF-based sharing algorithm and the influence of ϵ-DRF on sharing.

[5]Traces from Twitter and Google+ are also studied under the proposed scheme, but we omit them to avoid the repeated discussions, since most of the following results apply to Twitter and Google+ as well.

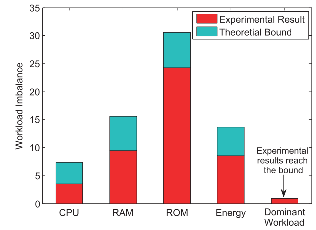


Fig. 2. Illustration of workload imbalance for different resources.

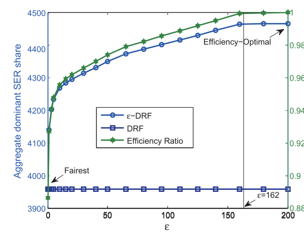


Fig. 3. Illustration of ϵ-DRF in social networks.

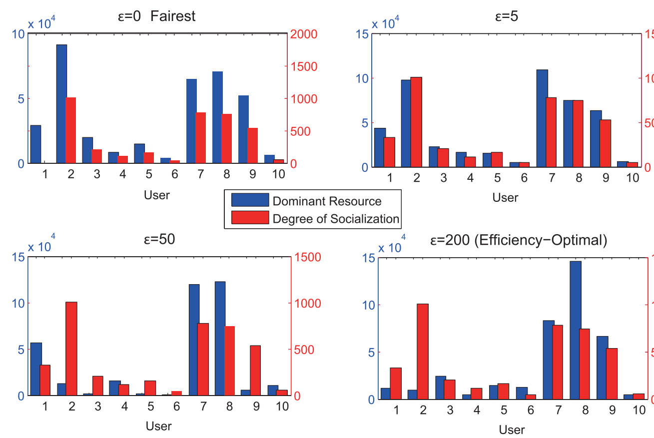


Fig. 4. Influence of ϵ on social fairness performance.

To demonstrate the efficiency of bottleneck resource, the multi-type workload imbalance across all nodes is studied in Fig. 2. For several specific types of resources, experimental results of workload imbalance and the corresponding theoretical bounds are marked, validating Proposition A.3. Besides, we observe that the dominant workload imbalance reaches the minimum ($\theta = 1$), which is consistent with Proposition A.2.

In Fig. 3, we demonstrate how the efficiency of ϵ-DRF varies with ϵ and compare it with the traditional DRF. When $\epsilon = 0$, DRF is identical to ϵ-DRF in yielding the least efficient result. As ϵ increases, ϵ-DRF gradually improves the aggregate user SER. Furthermore, when ϵ reaches $\max_{v,i} \pi(1, \phi_{vi}) = 162$, the efficiency-optimal result is obtained, and the aggregate user SER is then stabilized at the maximum. Note that in our simulations, even traditional DRF is also relatively efficient (above 89% of the optimal result), which might be occasional due to the sparsity of social relationship matrix.

On the other hand, the increase of ϵ does harm to the social fairness, as shown in Fig. 4. When $\epsilon = 0$, i.e., the exact DRF, the distribution of resource allocation among

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

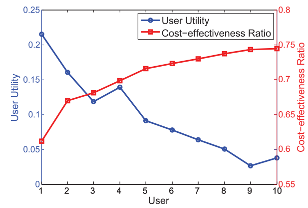


Fig. 5. User utility versus cost-effectiveness ratio.

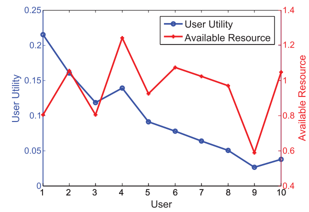


Fig. 6. User utility versus available resource.

users corresponds to their degrees of socialization,[6] which is the fairest result in social networks. However, when ϵ is raised, such social fairness is not maintained anymore since the resource allocation patterns gradually deviate from their degrees of socialization. In such cases, even though the majority of users do obtain more resources, the increase is not matched with their degrees of socialization, which may discourage their intention of socialization. This partially validates Propositions B.3-B.5. Hence, ϵ-DRF can improve the efficiency at the cost of a certain degree of fairness.

B. Incentive for Resource Sharing Among Users

We go further to investigate the practical and efficient VCG game-based incentive mechanism for resource sharing, where both user social relationship and participation incentive are involved. Specifically, we explore the economic potential of sharing incentive from two perspectives: users' *cost-effectiveness ratio* and *available resource*. Here available resource denoted by b_i is introduced to refer to the maximum resources that users are willing to share. Actually, it equals to the product of user unit social relationships $\frac{\phi_{vi}}{\sum_{v\in V_i}\phi_{vi}}$ and total idle resources $\sum_{k\in M} c_i^{(k)}$. We consider there are ten selected users among winning set Γ'_v. Each of them are in ascending cost-effectiveness ratio s_i.

Figure 5 illustrates the influence of cost-effectiveness ratio s_i on user utility. It is obvious that user utility is inversely proportional to s_i. According to the definition of s_i (i.e., $s_i = \alpha_i/p_i$), the influence on utility mainly depends on reputation p_i and true cost α_i. On the one hand, as mentioned in VCG mechanism, users with high reputation will be given more payment than those with low reputation. Thus user utility increases as p_i increases. On the other hand, user utility is defined as the difference between payment and cost. That is, use utility decreases as α_i increases.

We further demonstrate how user utility changes with respect to available resource in Fig. 6. Similar to Fig. 5, the overall trend in user utility is decreasing, mainly determined by cost-effectiveness ratio s_i. Taking these two curves together, we observe that available resource always leads to small fluctuations in user utility. The more available resource users have, the higher utility they will get, e.g., users 4 (the most available resource) and 9 (the least available resource). Intuitively, as for two users with the same s_i, the more resource one is willing to share, the more likely he/she is to be selected into winning set, and the higher utility he/she will get.

Thus we conclude that our proposed mechanism is useful to encourage four types of users to share their idle resources: (1) users with low bid, (2) users with high reputation, (3) users with considerable idle resources, and (4) users with strong social relationship. As for those users belonging to one or more types, they will have a high possibility of being selected.

[6]Here, a certain user's "degree of socialization" is determined by the total number of friends and the tie strength between the user and his/her friends.

VIII. Conclusion

We provide a novel VCG game-based incentive mechanism for multi-resource sharing to enable social crowdsourcing service, where the conflict between users' social connections and participation incentives is involved. To achieve the social fairness-efficiency tradeoff, DRF-based multi-resource sharing is further applied into the proposed mechanism. Numerical results validate the theoretical analysis about social fairness and efficiency of our sharing incentive mechanism.

In this paper, we focus on the issue of offline resource sharing, where any user takes on the predetermined roles of sharing or requesting resource over each time period. In the future work, we will consider the online sharing scenario. It is interesting to extend our system model to incorporate changing user role and dynamic joining/leaving system. In addition, it would be worthwhile to further study the effect of differences of user reputation or owned resources, e.g., a superuser with abundant resources and low bid joins system. We believe that system performance such as sharing efficiency and user experience is also related to the prediction accuracy of social tie among users, which can be enabled by Context-Aware and Big-Data techniques.

Appendix A
Proof of Theorem 1

Proof: We first prove the necessity. We consider the contradiction and assume that there exists a certain node v that does not have any bottleneck resources at some user $i \in \Gamma_v$, which corresponds to the following two possible cases.

Case 1: $\sum_{v\in V_i} \alpha_{iv}^{(k)} < 1$ $(\forall k \in M)$. If we further increase x_{iv} by $\min_{k\in M} \frac{c_i^{(k)}}{r_v^{(k)}}(1-\sum_{v\in V_i}\alpha_{iv}^{(k)})$, i.e., user U_i undertakes more of v's jobs, v's dominant SER share $\mu_{i,v}x_{iv}$ will also be raised without leading to the reduction of any other node's dominant SER share at user U_i.

Case 2: $\sum_{v\in V_i} \alpha_{iv}^{(k)} = 1$ for some resource $k \in M$ but $x_{iv'}\mu_{i,v'} > x_{iv}\mu_{i,v}$ for a certain node $v' \neq v$ who demands resource k at user U_i. In this case, we can decrease the allocated tasks of v' by an arbitrary small number ϵ and get new available resources amounting to $\vec{\sigma} = (\epsilon r_{v'}^{(1)}, \cdots, \epsilon r_{v'}^{(m)})$. Hence, the number of the allocated jobs as well as dominant SER share of v' can be increased using the above resource

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

vector $\vec{\sigma}$ without decreasing the dominant SER share of any other node v'' whose $x_{iv''}\mu_{i,v''} \le x_{iv}\mu_{i,v}$ at user U_i.

Thus, the necessary condition of DRF in social networks has been proved since both cases contradict the definition of DRF in social networks. The sufficiency is easy to prove using the definition of DRF, and we omit it for the brevity here. We have completed this proof of Theorem 1. □

Appendix B
Proof of Theorem 2

Proof: For a positive constant β, if user U_i allocates $\Delta_{iv} = \frac{\beta}{\mu_{i,v}}$ tasks to node v ($\forall v \in S_i$), all nodes in S_i will increase their dominant SER share at user U_i by β. To maintain the feasibility, we have that $\sum_{v\in S_i}\frac{\beta}{\mu_{i,v}}r_v^{(k)} \le L_i^{(k)},\ \forall k \in M$, which means the *maximum* value of β is $\beta = \min_{k\in M}\left(\frac{L_i^{(k)}}{\sum_{v'\in S_i}\frac{r_{v'}^{(k)}}{\mu_{i,v'}}}\right)$, and this explains the standing point of (5). If β reaches the maximum, one of the m resources will become saturated, and the resource type is exactly as (6) shows, which is also the reason why (6) holds. Therefore, as long as any node v in S_i is allocated Δ_{iv} tasks as (5) demonstrates, we can draw the following two conclusions:

(1) Resource $\widehat{k}$ is saturated (i.e., $\sum_{v\in V_i}\alpha_{iv}^{(\widehat{k})} = 1$).

(2) All nodes in S_i who demand resource $\widehat{k}$ will get the same dominant SER share at user U_i. It further implies that they also have equal or greater dominant SER share at U_i when compared with all other nodes in S_i that demand resource $\widehat{k}$, considering that some of such nodes in S_i may have already been excluded from S_i in previous loops.

The two conclusions above indicate that nodes in S_i (who also demand resource $\widehat{k}$) will have a bottleneck resource $\widehat{k}$. If Algorithm 1 continues so that $S_i = \emptyset$, all nodes in V_i will have a bottleneck resource at user U_i. Finally, when the algorithm is done for all users, any node $v \in V$ will have a bottleneck resource at each user $i \in \Gamma_v$. According to Theorem 1, we can get Algorithm 1 achieves the DRF in the proposed model. □

Appendix C
Proof of Proposition A.1

Proof: • **Pareto Efficiency**: By Theorems 1 and 2, under the allocation of Algorithm 1, any user U_i has a saturated resource $\widehat{k}$ at node $v \in V_i$. If we hope to increase U_i's dominant SER share, the allocations of resource $\widehat{k}$ must also be increased proportionally. There are two cases:

Case 1: Resource $\widehat{k}$ is exclusively demanded by U_i at node v. In this case, it would be impossible to provide more resource $\widehat{k}$ and more dominant SER share for U_i.

Case 2: There are other users who also demand resource $\widehat{k}$ at v (we denote them by the set D). If we allocate more resource $\widehat{k}$ to U_i, at least one user U_j in the set D will receive less resource $\widehat{k}$. It means the increase of U_i's dominant SER share will cause the reduction of U_j's dominant SER share.

Hence, the allocation of Algorithm 1 is Pareto Efficient.

• **Sharing Incentive**: If any resource $k \in M$ at node $v \in V$ is allocated to users in Γ_v in an equal-SER-share manner, it follows that $\alpha_{iv}^{(k)}\pi(1,\phi_{vi}) = \alpha_{jv}^{(k)}\pi(1,\phi_{vj}),\ \forall k \in M,\ \forall i,j \in \Gamma_v$. By transforming the form, summing over all users in Γ_v, and using $\sum_{j\in\Gamma_v}\alpha_{jv}^{(k)} \le 1$, we obtain $\alpha_{iv}^{(k)} \le \frac{1}{\sum_{j\in\Gamma_v}\frac{\pi(1,\phi_{vi})}{\pi(1,\phi_{vj})}}$. To validate the Sharing Incentive of our scheme, we only need to prove that for any user U_j, the *real share* of his/her dominant resources at any node $v \in V_j$ is more than $\frac{1}{\sum_{l\in\Gamma_v}\frac{\pi(1,\phi_{vj})}{\pi(1,\phi_{vl})}}$. For a given node v, let $\widehat{k}$ be the first saturated resource at v. Without loss of generality, we assume that U_i is allocated the maximum SER share of resource $\widehat{k}$. Thus, we have $\alpha_{i,v}^{(\widehat{k})}\pi(1,\phi_{vi}) \ge \alpha_{j,v}^{(\widehat{k})}\pi(1,\phi_{vi}),\ \forall j \in \Gamma_v$. Since $\sum_{j\in\Gamma_v}\alpha_{j,v}^{(\widehat{k})} = 1$, we further derive that $\sum_{j\in\Gamma_v}\frac{\pi(1,\phi_{vi})}{\pi(1,\phi_{vj})}\alpha_{i,v}^{(\widehat{k})} \ge 1$. Suppose $\alpha_{j,v}$ is the real share of U_j's ($\forall j \in \Gamma_v$) dominant resource at node v, and we have $\alpha_{i,v} \ge \alpha_{i,v}^{(\widehat{k})} \ge \frac{1}{\sum_{l\in\Gamma_v}\frac{\pi(1,\phi_{vi})}{\pi(1,\phi_{vl})}}$. According to Algorithm 1, for any user U_j in Γ_v, we have $\mu_{j,v} \ge \mu_{i,v}$, which means that $\alpha_{j,v}\pi(1,\phi_{vj}) \ge \alpha_{i,v}\pi(1,\phi_{vi}) \ge \frac{\pi(1,\phi_{vi})}{\sum_{l\in\Gamma_v}\frac{\pi(1,\phi_{vi})}{\pi(1,\phi_{vl})}}$. Thus, by an equivalent transformation of the above inequality, we conclude that $\alpha_{j,v} \ge \frac{1}{\sum_{l\in\Gamma_v}\frac{\pi(1,\phi_{vj})}{\pi(1,\phi_{vl})}}$ will be satisfied for $j \in \Gamma_v$. Therefore, Sharing Incentive has been proved.

• **Envy-Freeness**: For a given user U_i who envies another U_j at node v, user U_j must demand more types of resources than U_i and have strictly higher SER shares of every resource that U_i requires. According to Algorithm 1, the allocation to U_j is done no later than U_i, and thus $x_{iv}\mu_{i,v} \ge x_{jv}\mu_{j,v}$, which means that the share of U_i's dominant resource at node v is equal or greater than that of U_j. Hence, U_i will be allocated a larger amount of at least one type of resources than U_j, violating the previous assumption.

• **Strategy-Proofness:** Suppose a given user U_i untruthfully submits his/her resource demand as $\vec{r}_i \cdot \vec{H}$, where $\vec{H} = (H^{(1)},\cdots,H^{(M)})$ is the "untruthful gain" for multi-type resources. Since Algorithm 1 allocates computing resources on a job scale, all the resources that U_i demands must be raised proportionally so that U_i can *actually* fulfill more jobs. In other words, if we can prove that the allocation of the resource with the lowest "untruthful gain" cannot be increased, U_i will lose the incentive to cheat and the Strategy-Proofness is maintained. Specifically, we denote $\widehat{H} = \min_{k\in M}H^{(k)}$ as the lowest "untruthful gain" and $\bar{k} = \arg\min_{k\in M}H^{(k)}$ as the resource with this gain. Under the untruthful demand $\vec{r}_i\cdot\vec{H}$, the dominant SER share that U_i will gain in each "repeat-until" loop is $\bar{\beta} = \min_{k\in M}\left(\frac{L_v^{(k)}}{\sum_{j\in S_v\setminus\{i\}}\frac{r_j^{(k)}}{\mu_{j,v}}+\frac{H^{(k)}r_i^{(k)}}{H'\mu_{i,v}}}\right)$, where H' is the "untruthful gain" of U_i's dominant resource at node v.

According to the proof of Theorem 2, U_i will be allocated $\bar{\Delta}_{iv} = \frac{\bar{\beta}}{H'\mu_{i,v}}$ jobs in each loop, so U_i will receive resource $\bar{k}$ amounting to $\bar{\Delta}_{iv}\widehat{H}r_i^{(\bar{k})} = \frac{r_i^{(\bar{k})}}{\mu_{i,v}}\min_{k\in M}\left(\frac{L_v^{(k)}}{\sum_{j\in S_v\setminus\{i\}}\frac{H'r_j^{(k)}}{\widehat{H}\mu_{j,v}}+\frac{H^{(k)}r_i^{(k)}}{\widehat{H}\mu_{i,v}}}\right)$. Since $\frac{H'}{\widehat{H}} \ge 1$ and

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

$\frac{H^k}{\widehat{H}} \geq 1$, we have $\sum_{j \in S_\upsilon \setminus \{i\}} \frac{H' r_j^{(k)}}{\widehat{H} \mu_{j,\upsilon}} + \frac{H^{(k)} r_i^{(k)}}{\widehat{H} \mu_{i,\upsilon}} \geq \sum_{j \in S_\upsilon} \frac{r_j^{(k)}}{\mu_{j,\upsilon}}$. This naturally follows that $\bar{\Delta}_{i\upsilon} \widehat{H} r_i^{(\bar{k})} \geq \Delta_{i\upsilon} r_i^{(\bar{k})}$, where $\Delta_{i\upsilon}$ is the number of allocated jobs for U_i at node υ in each "repeat-until" loop under the truthful submission. We can conclude the untruthful submission of resource demands will not increase the *real* allocation of resource $\bar{k}$ and the Strategy-Proofness has been proved. □

Appendix D
Proof of Proposition A.3

Proof: For a given node $\upsilon \in V$, its workload in terms of resource k (we will refer it as workload k in the following) is $\tau_\upsilon^{(k)} = \sum_{i \in \Gamma_\upsilon} \frac{r_i^{(k)} x_{i\upsilon}}{c_\upsilon^{(k)}}$. Suppose $\widehat{k}$ is the dominant workload type at node υ. According to the inequality, we can derive $\frac{\tau_\upsilon^{(\widehat{k})}}{\tau_\upsilon^{(k)}} = \frac{\sum_{i \in \Gamma_\upsilon} \frac{r_i^{(\widehat{k})} x_{i\upsilon}}{c_\upsilon^{(\widehat{k})}}}{\sum_{i \in \Gamma_\upsilon} \frac{r_i^{(k)} x_{i\upsilon}}{c_\upsilon^{(k)}}} \leq \sum_{i \in \Gamma_\upsilon} \frac{r_i^{(\widehat{k})} c_\upsilon^{(k)}}{c_\upsilon^{(\widehat{k})} r_i^{(k)}} \leq \sum_{i \in \Gamma_\upsilon} \frac{A_{i\upsilon} c_\upsilon^{(k)}}{r_i^{(k)}}$. Proposition A.2 shows that $\tau_\upsilon^{(\widehat{k})} = 1$. Accordingly, we have $\tau_\upsilon^{(k)} \geq \frac{1}{\sum_{i \in \Gamma_\upsilon} \frac{A_{i\upsilon} c_\upsilon^{(k)}}{r_i^{(k)}}}$, $\forall \upsilon \in V$, which implies that for any $\upsilon \in V$, its workload k is lower-bounded (or the utilization of resource k at υ is lower-bounded). Since a certain type of workloads can achieve at most 100% at any node, the workload imbalance among all nodes in terms of resource k is $\theta^{(k)} = \max_{\upsilon, l \in V} \frac{\tau_\upsilon^{(k)}}{\tau_l^{(k)}} \leq \max_{\upsilon \in V} \sum_{i \in \Gamma_\upsilon} \frac{A_{i\upsilon} c_\upsilon^{(k)}}{r_i^{(k)}}$. The above bound can be reached in some cases. □

Appendix E
Proof of Proposition A.4

Proof: By Theorems 1 and 2, each user $U_i, i \in \mathcal{N}$ has at least one saturated resource, and we assume that one of such resources is $\widehat{k}$. This implies $\sum_{\upsilon \in V_i} \frac{x_{i\upsilon} r_\upsilon^{(\widehat{k})}}{c_i^{(\widehat{k})}} = 1$. Since the dominant SER share corresponds to the maximum share of all resources and $\pi(x, y)$ is linear of x, we obtain $\sum_{\upsilon \in V_i} \mu_{i,\upsilon} x_{i\upsilon} \geq \sum_{\upsilon \in V_i} \frac{x_{i\upsilon} r_\upsilon^{(\widehat{k})}}{c_i^{(\widehat{k})}} \pi(1, \phi_{\upsilon i}) \geq \min_{\upsilon \in V_i} \pi(1, \phi_{\upsilon i})$. Meanwhile, for the optimal allocation X^* at user U_i, we have $\sum_{\upsilon \in V_i} \mu_{i,\upsilon} x_{i\upsilon}^* \leq m \max_{\upsilon \in V_i} \pi(1, \phi_{\upsilon i})$. Therefore, at user U_i, the social welfare of the optimal result yielded by Algorithm 1 is at least $\frac{\min_{\upsilon \in V_i} \pi(1,\phi_{\upsilon i})}{m \max_{\upsilon \in V_i} \pi(1,\phi_{\upsilon i})}$. The social welfare of the efficiency-optimal allocation obtained by Algorithm 1 is lower-bounded by $\frac{B}{m}$. □

Appendix F
Proof of Proposition C.1

Proof: Intuitively, users have no incentive to submit a shorter sharing time since it will not contribute to the utility and may cause extra cost like battery consumption. Thus we only need to consider the case where a user U_i submits the cost $\hat{\alpha}_i$ and over submits his/her promised sharing time calculated with the promised number of tasks $\hat{z}_i$. The true cost is α_i and the true number of tasks he/she finish within the time limit is z_i.

First consider U_i does not over submit the sharing time, then the true utility is

$$
\begin{aligned}
f_i(\hat{\alpha}_i) &= p_i \left\{ \left[\min_{y_l \leq z_l} \sum_{l \in \Gamma'_\upsilon / i} y_l \hat{s}_l \right] - \left[\sum_{l \in \Gamma'_\upsilon / i} y_l^* \hat{s}_l \right] \right\} - y_i^* \alpha_i \\
&= p_i \left\{ \left[\min_{y_l \leq z_l} \sum_{l \in \Gamma'_\upsilon / i} y_l \hat{s}_l \right] - \left[\sum_{l \in \Gamma'_\upsilon} y_l^* \hat{s}_l \right] + y_i^* \left(\hat{s}_i - s_i \right) \right\}.
\end{aligned} \tag{12}
$$

In order to maximize the utility, we should maximize the second and third part in the brace of (12), since the first part has nothing to do with U_i. The second part is the gain or loss that U_i misreports the cost and the third part is the effect on the allocation and total cost by the misreport. The gain or loss in the second part is cancelled out by the third part and then the problem is to maximize $-\left[\sum_{l \in \Gamma'_\upsilon / i} y_l^* \hat{s}_l \right] - y_i^* s_i$. Thus U_i will not increase the utility by misreporting the cost, and submitting the true cost is a weakly dominant strategy.

If U_i over submits his/her sharing time, we have $y_i' \leq z_i \leq \hat{z}_i$, and the utility becomes

$$
\begin{aligned}
& f_i(\hat{\alpha}_i, \hat{z}_i) \\
&= p_i \left\{ \left[\min_{y_l \leq z_l} \sum_{l \in \Gamma'_\upsilon / i} y_l \hat{s}_l \right] - \left[\sum_{l \in \Gamma'_\upsilon, y_i \leq y_i'} y_l^* \hat{s}_l \right] + y_i^* \left(\hat{s}_i - s_i \right) \right\} - \kappa \gamma \\
&\leq p_i \left\{ \left[\min_{y_l \leq z_l} \sum_{l \in \Gamma'_\upsilon / i} y_l \hat{s}_l \right] - \left[\sum_{l \in \Gamma'_\upsilon, y_i \leq z_i} y_l^* \hat{s}_l \right] + y_i^* \left(\hat{s}_i - s_i \right) \right\}.
\end{aligned}
$$

So if U_i over submits the sharing time and fails to finish the tasks, he/she will surely get a lower utility. Thus truly submit the sharing time is also a weakly dominant strategy.

Above all, we conclude that the mechanism is truthful. □

Appendix G
Proof of Proposition C.2

Proof: If user U_i does't receive any task allocated by the platform, i.e., $y_i = 0$, then his/her utility $f_i = 0$. Therefore, we only need to consider the case where U_i is included into winning set. Derived by (12) when $\hat{s}_i = s_i$, the utility of U_i is $f_i(\alpha_i, z_i) = p_i \left\{ \left[\min_{y_l \leq z_i} \sum_{l \in \Gamma'_\upsilon / i} y_l s_l \right] - \left[\sum_{l \in \Gamma'_\upsilon} y_l^* s_l \right] \right\}$ As the first part in the brace is over a larger set than the second part, we obtain $\min_{y_l \leq z_i} \sum_{l \in \Gamma'_\upsilon / i} y_l s_l \geq \sum_{l \in \Gamma'_\upsilon} y_l^* s_l$. Hence, we can easily get $f_i(\alpha_i, z_i) \geq 0$ and then the proof is completed. □

Appendix H
Proof of Proposition C.3

Proof: Suppose user U_i only knows the interval of the sharing time he/she can achieve, thus the total tasks he/she can finish within the time limitation is in an interval as well, which is assumed to be $[z_i^{(1)}, z_i^{(2)}]$. Assume that z_i obeys uniform

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

distribution, and the corresponding probability distribution function is $pdf(i) = \frac{1}{z_i^{(2)} - z_i^{(1)}}$. The expected utility of U_i is

$$
\begin{aligned}
&E(f_i(\alpha_i, z_i)) \\
&= E\left\{ p_i \left[\min_{y_l \le z_i} \sum_{l \in \Gamma'_v / i} y_l s_l \right] - p_i \left[\sum_{l \in \Gamma'_v / i} y_l^* s_l \right] \right\} - \kappa \gamma \\
&= p_i \left[\min_{y_l \le z_i} \sum_{l \in \Gamma'_v / i} y_l s_l - \int_{z_i^{(1)}}^{\hat{z}_i} pdf(i) \left[\sum_{l \in \Gamma'_v, y_i \le y_i'} y_l^* s_l \right] dz_i \right] \\
&\quad - p_i \left[\int_{\hat{z}_i}^{z_i^{(2)}} pdf(i) \left[\sum_{l \in \Gamma'_v, y_i \le \hat{z}_i} y_l^* s_l \right] dz_i \right] - \gamma\, P(z_i < \hat{z}_i).
\end{aligned}
$$

The safest choice is $\hat{z}_i = z_i^{(1)}$. If U_i increases z_i by submitting a higher sharing time, the gain of expected utility is $\Delta E(f_i(\alpha_i, z_i))$.Thus U_i can choose the sharing time to make $\Delta E(f_i(\alpha_i, z_i))$ positive. But even if U_i submits a longer sharing time and can not fulfill all tasks, the scheme can guarantee U_i will work as long as he/she can. The utility is

$$
f_i(\hat{\alpha}_i, \hat{z}_i) = p_i \left\{ \left[\min_{y_l \le z_l} \sum_{l \in \Gamma'_v / i} y_l \hat{s}_l \right] - \left[\sum_{l \in \Gamma'_v, y_i \le y'} y_l^* \hat{s}_l \right] \right\} - \gamma\,.
$$

Thus U_i can only increase the utility by making ${y_i}'$ as large as possible, i.e., working as long as he/she can. Based on how v wants the tasks to be fulfilled, we choose the value of $P(z_i < \hat{z}_i)$ and set $\gamma = p_i \left[\sum_{l \in \Gamma'_v, y_i \le z_i^{(1)}} y_l^* s_l \right] / P(z_i < \hat{z}_i)$. Hence $\Delta E(f_i(\alpha_i, z_i))$ is below zero and users will lose the incentive to submit a longer sharing time. In view of the uncertainty of sharing period, we have completed the proof. □

REFERENCES

[1] "Cisco global cloud index: Forecast and methodology, 2014–2019," Cisco Inc., San Jose, CA, USA, White Paper, Oct. 2015.

[2] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1732–1744, Jun. 2016.

[3] T. Xi, W. Wang, E. C.-H. Ngai, Z. Song, Y. Tian, and X. Gong, "Energy-efficient collaborative localization for participatory sensing system," in *Proc. IEEE GLOBECOM*, Dec. 2015, pp. 1–6.

[4] T. Ernvall, S. El Rouayheb, C. Hollanti, and H. V. Poor, "Capacity and security of heterogeneous distributed storage systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 12, pp. 2701–2709, Dec. 2013.

[5] *Socialbakers: Social Media Marketing, Analytics & Performance Tools*, accessed on Mar. 18, 2017. [Online]. Available: http://www.socialbakers.com/

[6] P.-Y. Chen, S.-M. Cheng, P.-S. Ting, C.-W. Lien, and F.-J. Chu, "When crowdsourcing meets mobile sensing: A social network perspective," *IEEE Commun. Mag.*, vol. 53, no. 10, pp. 157–163, Oct. 2015.

[7] T. Luo, S. S. Kanhere, and H.-P. Tan, "SEW-ing a simple endorsement Web to incentivize trustworthy participatory sensing," in *Proc. IEEE SECON*, Jun. 2014, pp. 636–644.

[8] D. Lee, J. Kim, H. Lee, and K. Jung, "Reliable multiple-choice iterative algorithm for crowdsourcing systems," in *Proc. ACM SIGMETRICS*, Jun. 2015, pp. 205–216.

[9] X. Ma, J. Liu, and H. Jiang, "Resource allocation for heterogeneous applications with device-to-device communication underlaying cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 15–26, Jan. 2016.

[10] M. Tang, L. Gao, H. Pang, J. Huang, and L. Sun, "A multi-dimensional auction mechanism for mobile crowdsourced video streaming," in *Proc. IEEE WiOpt*, May 2016, pp. 1–8.

[11] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand, "Motivating smartphone collaboration in data acquisition and distributed computing," *IEEE Trans. Mobile Comput.*, vol. 13, no. 10, pp. 2320–2333, Oct. 2014.

[12] D. R. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Proc. NIPS*, Dec. 2011, pp. 1953–1961.

[13] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models and Applications*. Cambridge, U.K.: Cambridge Univ. Press, Jan. 2012.

[14] Y. Zhang and M. van der Schaar, "Incentive provision and job allocation in social cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 607–617, Sep. 2013.

[15] D. Zarchy, D. Hay, and M. Schapira, "Capturing resource tradeoffs in fair multi-resource allocation," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 1062–1070.

[16] T. Bonald and J. Roberts, "Multi-resource fairness: Objectives, algorithms and performance," in *Proc. ACM SIGMETRICS*, Jun. 2015, pp. 31–42.

[17] D. C. Parkes, A. D. Procaccia, and N. Shah, "Beyond dominant resource fairness: Extensions, limitations, and indivisibilities," *ACM Trans. Econ. Comput.*, vol. 3, no. 1, pp. 3:1–3:22, Mar. 2015.

[18] J. Kwak, Y. Kim, J. Lee, and S. Chong, "DREAM: Dynamic resource and task allocation for energy minimization in mobile cloud systems," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 12, pp. 2510–2523, Dec. 2015.

[19] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types," in *Proc. ACM USENIX NSDI*, Mar. 2011, pp. 1–14.

[20] A. Ghodsi, V. Sekar, M. Zaharia, and I. Stoica, "Multi-resource fair queueing for packet processing," in *Proc. ACM SIGCOMM*, Aug. 2012, pp. 1–12.

[21] C. Joe-Wong, S. Sen, T. Lan, and M. Chiang, "Multiresource allocation: Fairness-efficiency tradeoffs in a unifying framework," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1785–1798, Dec. 2013.

[22] D. Dolev, D. G. Feitelson, J. Y. Halpern, R. Kupferman, and N. Linial, "No justified complaints: On fair sharing of multiple resources," in *Proc. ACM ITCS*, Jan. 2012, pp. 68–75.

[23] J. McAuley and J. Leskovec, "Discovering social circles in ego networks," *ACM Trans. Knowl. Discovery Data*, vol. 8, no. 1, Feb. 2014, Art. no. 4.

[24] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

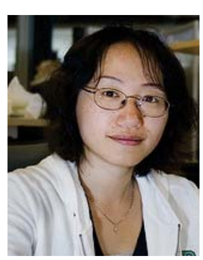
[25] *Open Garden, Connecting the Next Billion Mobile Devices*, accessed on Mar. 18, 2017. [Online]. Available: https://www.opengarden.com

[26] *Waze*, accessed on Mar. 18, 2017. [Online]. Available: https://www.waze.com/zh/

[27] A. Schrijver, *Theory Linear Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.

[28] J. Huang, Z. Han, M. Chiang, and H. V. Poor, "Auction-based resource allocation for cooperative communications," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 7, pp. 1226–1237, Sep. 2008.

[29] S. Yang and B. Hajek, "VCG-kelly mechanisms for allocation of divisible goods: Adapting VCG mechanisms to one-dimensional signals," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1237–1243, Aug. 2007.



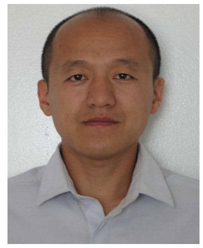
Xiaoying Gan received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006. From 2009 to 2010, she was a Visiting Researcher with the California Institute for Telecommunications and Information, University of California at San Diego, San Diego, CA, USA. She is currently an Associate Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. Her current research interests include network economics, social aware networks, heterogeneous cellular networks, multiuser multi-channel access, and dynamic resource management.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.

Yuqing Li received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2014. She is currently pursuing the Ph.D. degree in electronic engineering with Shanghai Jiao Tong University, Shanghai, China. Her current research interests include network economics, social aware networks, heterogeneous cellular networks, and network security.

Weiwei Wang received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2015. She is currently pursuing the Ph.D. degree in electronic engineering with Shanghai Jiao Tong University, Shanghai, China. Her current research interests include network tomography and compressed sensing.

Luoyi Fu received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, China, in 2009. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, with Prof. X. Wang. Her research of interests is information propagation and privacy protection in evolving social networks.



Xinbing Wang received the B.S. degree (with honors) from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001, and the Ph.D. degree (major from the Department of Electrical and Computer Engineering and minor from the Department of Mathematics) from the North Carolina State University, Raleigh, NC, USA, in 2006. He is currently a Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. He has been an Associate Editor of the IEEE TRANSACTIONS ON NETWORKING/ACM Transactions on Networking and the IEEE TRANSACTIONS ON MOBILE COMPUTING, and a member of the technical program committees of several conferences, including the ACM MobiCom 2012, the ACM MobiHoc 2012 and 2013, and the IEEE INFOCOM from 2009 to 2014.

Authorized licensed use limited to: Hong Kong University of Science and Technology. Downloaded on June 16,2020 at 02:32:48 UTC from IEEE Xplore. Restrictions apply.