Strategic Social Team Crowdsourcing: Forming a Team of Truthful Workers for Crowdsourcing in Social Networks

Wanyuan Wang, Zhanpeng He, Peng Shi, Weiwei Wu, Yichuan Jiang, *Senior Member, IEEE*, Bo An, Zhifeng Hao, and Bing Chen

Abstract—With the increasing complexity of tasks that are crowdsourced, requesters need to form teams of professional workers that can satisfy complex task skill requirements. Team crowdsourcing in social networks (SNs) provides a promising solution for complex task crowdsourcing, where the requester hires a team of professional workers that are also socially connected can work together collaboratively. Previous social team formation approaches have mainly focused on the algorithmic aspect for social welfare maximization; however, within the traditional objective of maximizing social welfare alone, selfish workers can manipulate the crowdsourcing market by behaving untruthfully. This dishonest behavior discourages other workers from participating and is unprofitable for the requester. To address this strategic social team crowdsourcing problem, truthful mechanisms are developed to guarantee that a worker's utility is optimized when he behaves honestly. This problem is proved to NP-hard, and two efficient mechanisms are proposed to optimize social welfare while reducing time complexity for different scale applications. For small-scale applications where the task requires a small number of skills, a binary tree network is first extracted from the social network, and a dynamic programming-based optimal team is formed in the binary tree. For large-scale applications where the task requires a large number of skills, a team is formed greedily based on the workers' social structure, skill, and working cost. For both mechanisms, the threshold payment rule, which pays each worker his marginal value for task completion, is proposed to elicit truthfulness. Finally, the experimental results of a real-world dataset show that compared to the benchmark exponential VCG truthful mechanism, the proposed small-scale-oriented mechanism can reduce computation time while producing nearly the same social welfare results. Furthermore, compared to other state-of-the-art polynomial heuristics, the proposed large-scale-oriented mechanism can achieve truthfulness while generating better social welfare outcomes.

Index Terms—Mechanism design, crowdsourcing, team formation, social networks

1 Introduction

Crowdsourcing has become a very popular business paradigm for the requesters to hire the professional or inexpensive workers through the online labor markets [1], [2]. Traditional crowdsourcing applications mainly focus on crowdsourcing simple tasks, each can be completed by a single worker, such as image annotation/data labeling [3], [4] and consumer survey/product review [5]. However, with the increasing complexity of tasks that are crowdsourced, the requester needs to hire a group of professional and social connected workers. These workers can work together as a team to satisfy task's complex skill requirements [6], [7], [8].

The mobile apps-based crowdsensing (MCS) is a typical motivating domain of social team crowdsourcing. MCS allows mobile users to utilize human-carried devices (e.g., smartphones) to sense environment information (e.g., air quality or noise level) of multiple interested regions [9], [10], [11], [12]. However, in the case that the users cannot directly reach out to the requester since the global infrastructure-based wireless networks (e.g., Wi-Fi) are unavailable or data transportation is costly, the MCS project might fail [13]. Within social team crowdsourcing, the requester recruits a team of users that can not only collect the environment information of the interested regions but also form a connected route such that any participant user can transport the sensed data to the requester through the local short-range device-to-device wireless communications [14], [15]. Another typical social team crowdsourcing domain is the web-based complex task (e.g., software project development) crowdsourcing, where the requester wishes to recruit a team of engineers that not only can provide the necessary skills (including requirement analysis, architecture design, implementation, testing, deployment, and maintenance), but also can collaborate with each other effectively for successfully completing these inter-dependent subtasks [16], [17].

The social team crowdsourcing application can be implemented through a reverse auction model where the requester announces a task's skill requirements through a business

• *W. Wang, Z. He, P. Shi, and Y. Jiang are with the School of Mathematics and Big Data, Foshan University, Foshan 528000, China, and the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China. E-mail: {wywang, zp_he, pengshi, yjiang}@seu.edu.cn.*
• *W. Wu is with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China. E-mail: weiweiwu@seu.edu.cn.*
• *B. An is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore. E-mail: boan@ntu.edu.sg.*
• *Z. Hao is with the School of Mathematics and Big Data, Foshan University, Foshan 528000, China. E-mail: zfhao@fosu.edu.cn.*
• *B. Chen is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China. E-mail: cb_china@nuaa.edu.cn.*

Manuscript received 26 Aug. 2016; revised 27 June 2018; accepted 24 July 2018. Date of publication 31 July 2018; date of current version 2 May 2019. (Corresponding author: Yichuan Jiang.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2018.2860978

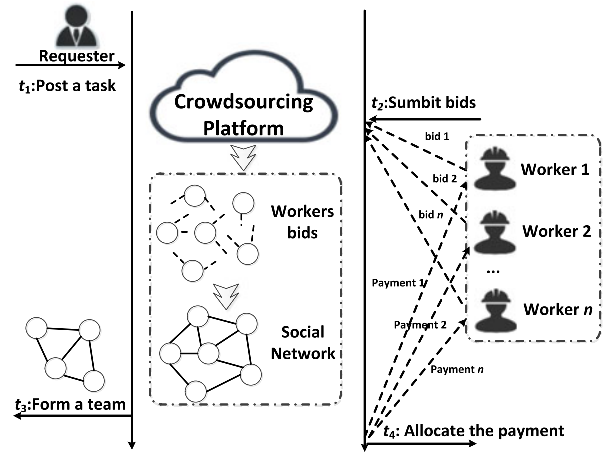


Fig. 1. The paradigm of social team crowdsourcing.

crowdsourcing platform such as mobile apps-based platform Placemeter (www.placemeter.com) or web-based platform Guru (www.guru.com). Each worker bids to sell skill services that are associated with skill provision costs. Based on these workers' bids, the crowdsourcing platform then can construct workers' social network (SN) according to their social connections. These social connections might represent the local available device-to-device communications in MCS or task collaboration experiences in web-based crowdsourcing. The advantage of using such SNs is that these workers who can communicate effortlessly are expected to work efficiently as a team [18], [19], which has recently been exploited to analyze the collaboration among crowd workers [20]. Finally, the requester forms a professional and collaborative team of workers in the SN with the aim of social welfare maximization. Fig. 1 depicts the framework of a social team crowdsourcing campaign.

Intuitively, previous social team formation approaches [21], [22], [23], [24], [25], [26], [27] that focused on forming a collaborative team in social networks can be extended to addressing this social team crowdsourcing problem. However, these existing approaches mainly focus on the algorithmic aspect of social welfare maximization. Such a social welfare maximization objective will encourage strategic workers to lie about their real private information, e.g., reporting a higher skill provision cost to maximize their own utility. This untruthful behavior will severely reduce the requester's utility and discourage other workers from participating for fear of market manipulation. A crowdsourcing platform without adequate worker participation is unprofitable for requesters and will fail to operate [28]. It is essential to design truthful social team crowdsourcing mechanisms where each worker's optimal bidding strategy is to declare his private information truthfully [29].

Unfortunately, as noted in [30], the well-known VCG truthful mechanism, which aims to form an optimal team with maximum social welfare, is not applicable. This is attributable to the fact that the proposed social team crowdsourcing problem is NP-hard, and forming an optimal team requires exponential $O(n^k)$ time complexity, where k is the number of required task skills and n is the number of workers in a SN. On the other hand, directly extending existing polynomial social team formation approaches [21], [22], [23], [24], [25], [26], [27] is infeasible since they are not monotone and cannot guarantee truthfulness. Against this background, to reduce time complexity while optimizing social welfare, we propose two efficient truthful mechanisms for small- and large-scale applications, respectively. For small-scale applications where k is small, we develop a fixed-parameter $O(n^2 4^k)$ mechanism. This mechanism works by first transforming workers' SN into a binary tree network, and implementing the optimal dynamic programming mechanism in the transformed tree network. For large-scale applications where k is large, we develop a polynomial $O(k^3 n^2)$ mechanism that selects team workers greedily based on these workers' social structure, skill and working cost. For both mechanisms proposed, the threshold payment rule, which pays each worker the marginal value for task completion, is used to elicit truthfulness.

The main contributions of this paper are summarized as follows: 1) we propose the social team crowdsourcing campaign as a mechanism design problem where the crowdsourcing platform needs to design computationally efficient team formation mechanisms that not only form teams with the maximal social welfare but that also ensure that truthfully reporting is a dominant strategy. 2) We develop a parameterized exponential $O(n^2 4^k)$ time mechanism and polynomial $O(k^3 n^2)$ time mechanism for small- and large-scale applications, respectively. The theoretical results show that both proposed mechanisms achieve truthfulness. 3) We apply the proposed mechanisms to a real-world dataset collected from the crowdsourcing platform Guru. The experimental results show that compared to the benchmark exponential VCG truthful mechanism [29], the proposed small-scale-oriented mechanism can reduce computation time while achieving nearly the same social welfare outcomes. Furthermore, compared to other state-of-the-art polynomial social team crowdsourcing heuristics [21], the proposed large-scale-oriented mechanism can achieve truthfulness while generating better social welfare outcomes.

The remainder of this paper is organized as follows. In Section 2, we provide a brief review of related subjects on crowdsourcing, truthful mechanisms and team formation in a SN. In Section 3, we formulate the social team crowdsourcing problem. We propose the parameterized exponential and polynomial time mechanisms in Sections 4 and 5, respectively. In Section 6, we conduct a set of experiments to evaluate our proposed mechanism's performance on social welfare, requester utility and worker utility on a real-world dataset. Finally, we conclude our paper and discuss future work in Section 7.

2 Related Work

2.1 Crowdsourcing for Task Allocation

Crowdsourcing is a very useful paradigm to help requesters access powerful human resources to complete tasks that are difficult for computers [1], [2]. In traditional crowdsourcing applications, workers can work at different skill levels, and requester allocates tasks to expert workers with high-level skills, e.g., improving the labeling accuracy and completing as many labeling tasks as possible in web-based crowdsourcing [3], [4], [5], providing high quality of information and acquiring these data as fast as possible in mobile-apps based crowdsensing [6], [9]. These typical crowdsourcing applications mainly focus on solving simple tasks that each worker can complete independently. However, with the increasing complexity of tasks that are crowdsourced, recent studies have begun to address the need for more complex tasks that must be solved by teams [7], [8], [31], [32]. A popular method for solving complex tasks involves dividing such tasks into flows of simple subtasks, allocating these sub-tasks to team workers and finally combining the partial results of sub-tasks to achieve the final outcome [33], [34]. Complementary to these network-ignorance studies, within the SN environment, we aim to hire teams of workers. These workers are not only professional but also socially connected such that they can collaborate with one another effectively, thereby improving task quality further. Recently, Chen et al. [10] realizes that workers not only can achieve the extinct rewards from the requester but also enjoys an intrinsic benefit from network effects. But in their study, the "network" concept is the global crowdsourcing environment. However, in our paper, the network means the local collaboration efficiency among workers, which can not only help the worker achieve the intrinsic benefit but also help the requester recruit social collaborative teams.

2.2 Mechanism Design for Crowdsourcing

The key to making a crowdsourcing market successful is incentivizing workers to participate [28]. An auction-based incentive mechanism for crowdsourcing can be generalized as follows: requester first posts the task to crowd workers, workers submit their bids including skills and skill provision costs, and finally requester selects adequate workers and pays these workers properly based on their bids [5], [6]. However, workers are always strategic by submitting untruthful private information to maximize their own utility. To guarantee truthfulness, Nisan [29] first proposes the VCG mechanism by allocating tasks to workers optimally and offering workers critical payment to elicit truthfulness. The main idea behind the critical payment scheme is that each winner worker is paid to his marginal skill contribution value for task completion. Due to the intractability of computing the optimal allocation of tasks to workers, polynomial time schemes with monotone characteristics are more desirable [30]. On the other hand, Han et al. [11] propose a posted price-based incentive mechanism, where requester first announces the predetermined take-it-or-leave-it price to workers, workers choose to join the campaign based on their own willingness. From the perspective of the requester, this price should be optimally devised to incentive workers to join while spending the minimum budget. These traditional incentive mechanisms [5], [6], [11], [28], [29], [31] mainly focus on crowdsourcing simple and independent tasks without the need of worker collaboration. By contrast, this paper is mainly interested in addressing social team crowdsourcing problems where the requester aims to form a team of professional and collaborative workers in SNs. Recently, the research [35] also studied team formation mechanism. However, they aimed to elicit each individual's truthful preferences over his teammates, and thus had a different objective.

2.3 Team Formation in Social Networks

Lappas et al. [21] were the first to formulate the problem of team formation in social networks. They model each social network as a weighted and undirected graph $G = <V, E, W>$ where vertices V represent individuals, edges E represent social connections among individuals and weights W on the edges represent individuals' coordination costs. Given a social network G and a task J, Lappas et al. [21] aim to build a team of individuals $V^* \subseteq V$ such that V^* not only meets all skill requirements of J but also incurs the lowest team coordination costs. What followed [21] are other social team formation variants with different constraints and objectives, such as online dynamic settings where tasks arrive dynamically [22]; load balancing objective where workers' assigned tasks are proportional to their capacity [23] and budget minimization schemes where workers must be paid for their skill service provisions [24], [25], [26]. Rangapuram et al. [27] propose a generalized social team formation problem where team size, team budgets and team coordination costs are considered simultaneously. Considering that socially close individuals can improve task quality levels, Voice et al. [14], Bistaffa et al. [15] and Wang et al. [36] are interested in forming a team of social connected individuals for task completion. All of the above studies assume that individuals are honest and can present their private information truthfully. However, since workers are selfish, within traditional social team formation mechanisms, each worker must expend considerable efforts computing the most beneficial strategy based on beliefs of the strategies of other bidders. This overhead and the impossibility of computing the best strategy discourages workers from participating and is unprofitable for the requester [28], [37]. Therefore, a truthful mechanism where each worker's best interest is to report his private information truthfully is essential for social team crowdsourcing.

3 Problem Description

3.1 Auction-Based Social Team Crowdsourcing

We model social team crowdsourcing campaign between a requester and workers as a three-stage reverse auction framework [9], [31], [38].

- In the first stage, the requester announces a task $T = <V_T, O_T>$, where V_T is a predefined profit that will be yielded to the requester upon the completion of T, and $O_T = \{s_1, s_2, \dots, s_k\}$ represents the skill requirements of task T, such as the interested regions he covers. Let $k = |O_T|$ denote the task size, i.e., the number of skills required by T.
- In the second stage, each worker a_i submits a bid $B_i = (R_i', c_i')$ where $R_i' \in O_T$ represents a subset of skills one can provide, and the working cost c_i', represents the reserved price a_i wants to charge for his skill provisions, e.g., a worker charges \$20/hr offering his skilled service of sensing environmental information. Let $R_i \in O_T$ and c_i denote a_i's real skill set and cost, and $R_i'(\neq R_i)$ and $c_i'(\neq c_i)$ denote a_i's wrong skill set and cost. Each worker a_i has its own discretion to determine personal information R_i' and c_i', which is guaranteed to be the same as truthful personal information R_i and c_i in our mechanism. During this bidding stage, we have a couple of practical settings: 1) each worker submits his bids independently [39] and 2) each worker's working cost is fixed and independent on which and how many skills he actually provides for task completion. This cost-skill independent assumption is widely used in MCS [6], [9]1 and web-based software crowdsourcing [21], [23]2.
- In the third stage, upon collecting the bids, the requester first determines the workers' social network

1. Once a worker participates in the MCS, the cost (e.g., the smartphone battery power consumption per unit time) is determined and is independent on the interested regions the worker actually accesses.

2. Once a worker is recruited to develop software, the requester should pay the worker at least the payment the worker charges no matter which skills are actually used for software development.

$SN = <A, E>$ where $A = \{a_1, a_2, \ldots, a_n\}$ represents the collection of workers and $\forall(a_i, a_j) \in E$ represents the existence of a collaborative relationship between workers a_i and a_j, such that they can communicate with one another effectively, e.g., workers a_i and a_j are geographically nearby between which the local device-to-device communications are available. In such a *SN*, each worker $a_i \in A$ can be defined by a 3-tuple (R'_i, c'_i, N_i) where R'_i denotes a_i's skills, c'_i denotes a_i's working cost, and N_i denotes neighbors of worker a_i, i.e., $N_i = \{a_j | (a_i, a_j) \in E\}$.

A mechanism $M = (TF, Pay)$ consists of a social team formation function *TF* and a payment function *Pay*. The social team formation function $TF = \{x_1, x_2, \ldots, x_n\}$ indicates whether a worker a_i is selected as a winner ($x_i = 1$ or not $x_i = 0$). Let $G = \{a_i | x_i = 1\}$ denote the winning team and $C(G) = \sum_{a_i \in G} c'_i$ denote the team cost incurred by team members in *G*. To complete the task *T* successfully, the team *G* must satisfy the following two conditions. 1) *Professional*: each required skill must be satisfied by at least one team member, i.e., $\forall s_j \in O_T, \exists a_i \in G,\ s.t.\ s_j \in R'_i$; and 2) *Collaborative*: the sub-graph induced by team members in *G* must be connected. We refer to such a professional and collaborative team a *feasible* team. The payment function $Pay = \{p_1, p_2, \ldots, p_n\}$ determines the reward paid to each team member upon the completion of *T*. The utility u_i of each winner $a_i \in G$ then is the difference between the payment p_i and its true cost c_i, i.e., $u_i = p_i - c_i$, which also equals $p_i - c'_i$ under truthful bidding. Otherwise, when worker a_i is not selected as a winner, then $u_i = 0$. The requester's utility is the task's profit minus the payment given to team members, i.e., $u_T = V_T - \sum_{a_i \in G} p_i$.

The social welfare W_T of the crowdsourcing platform is the sum of the requester's utility and workers' aggregate utility $\sum_{a_i \in G} u_i = \sum_{a_i \in G} (p_i - c'_i)$, which equals the task profit minus aggregate bidding costs of the winning team *G*, i.e., $W_T = V_T - \sum_{a_i \in G} p_i + \sum_{a_i \in G} (p_i - c'_i) = V_T - \sum_{a_i \in G} c'_i$. In this paper, we primarily consider social welfare maximization objective. As the crowdsourcing platform is a social service-oriented application and it is natural to design mechanisms that maximize social welfare. It should also be noted that in the proposed mechanism, we also guarantee that both the requester and worker can achieve positive utilities.

Now the social team crowdsourcing problem can be formulated as follows.

Definition 1 (Social Team Crowdsourcing Problem (STCP)). *Given a task* $T = <V_T, O_T>$ *and workers' bids* $B = \{B_1, \ldots, B_n\}$ *associated with the workers' social network* $SN = <A, E>$*, the crowdsourcing system is designed to form the optimal feasible team that produces maximal social welfare for* T, *i.e.,*

$$Maximize\, W_T. \tag{1}$$

Subject to:

$$y_{ij} = x_i, \forall s_j \in R'_i; \sum_{a_i \in A} y_{ij} \geq 1, \forall s_j \in O_T \tag{2}$$

$$x_i + x_j - 1 \leq z_{ij}, z_{ij} \leq x_i, z_{ij} \leq x_j, \forall (a_i, a_j) \in E \tag{3}$$

$$\forall r \in [1, n], w_r \leq x_r, \forall i > r, w_i \leq 1 - x_r, \sum_{a_i \in A} w_i = 1 \tag{4}$$

$$\sum_{(a_u, a_v) \in \sigma(H)} z_{uv} + f(H, A) \geq x_i, \forall \{a_r\} \subseteq H \subseteq A, a_i \in A \tag{5}$$

$$x_i, y_{ij}, z_{ij}, w_i \in \{0, 1\}. \tag{6}$$

Constraint (2) is designed to build a professional team, where variable $x_i \in \{0, 1\}$ is the decision variable determining whether worker a_i is selected as a team member ($x_i = 1$) or not ($x_i = 0$); $y_{ij} \in \{0, 1\}$ represents whether worker a_i can provide skill s_j ($y_{ij} = 1$) or not ($y_{ij} = 0$). Constraints (3)-(5) are designed to build a collaborative team such that the selected team members form a connected sub-graph, which is inspired by recent graph theory technique [40], where an integer program (IP) technique (i.e., there must be a flow from a selected node to any selected nodes via edges in the selected subgraph) is proposed to formulate a connected subgraph. In constraint (3), $z_{ij} = 1$ indicates that the connection (a_i, a_j) is selected when both endpoint workers a_i and a_j are selected. In constraint (4), w_i denotes whether a_i is the selected worker with the smallest index ($w_i = 1$) or not ($w_i = 0$). For an arbitrarily selected worker a_r such that $w_r = 1$ (which is called the root worker), when there is a path from a_r to all of the other selected workers $\{a_j | x_j = 1, j \neq r\}$, these selected workers form a connected sub-graph. Without loss of generality, we choose the selected worker with the smallest index as the root worker, i.e., a_r. Constraint (5) indicates that worker a_i can be selected when for each worker set *H* containing the root worker a_r, either all selected workers $\{a_i | x_i = 1\}$ are in *H* (i.e., $f(H, A) = 1$) or at least one connection $(a_u, a_v) \in \delta(H)$ exists among the selected workers. Function $f(H, A)$ is defined as follows: if $\sum_{a_i \in H} x_i = \sum_{a_i \in A} x_i$, then $f(H, A) = 1$; otherwise, $f(H, A) = 0$. Function $\sigma(H)$ denotes the set of connections crossing the cut $(H, A \backslash H)$, i.e., $\sigma(H) = \{(a_i, a_j) \in E | a_i \in H, a_j \in A \backslash H\}$.

Lemma 1. *The social team crowdsourcing problem (STCP) defined in Definition 1 is NP-hard and has no polynomial algorithm with a constant approximation factor unless* $P = NP$.

For space limitation, all omitted proofs can be seen in Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2018.2860978.

3.2 Strategic Workers and the Purpose of Truthful Mechanisms

In practical crowdsourcing markets, each worker strategically maximizes his own utility. Such an objective to maximize social welfare alone encourages strategic workers to lie about their real valuations, e.g., by declaring a higher working cost value in hope of securing higher payment. As a result, in competitive crowdsourcing markets, each worker will exert considerable efforts to compute his best bidding strategy based on his knowledge of other workers' strategies. This untruthful behavior and overhead for computing the best strategy discourages workers from participating and reduces social welfare [28], [41]. Therefore, it is necessary to design truthful mechanisms to guarantee that each worker can maximize his utility by reporting his private information truthfully.

For each worker a_i, there are two types of private information he can manipulate: the available skills R_i he can provide and the working cost c_i of participation. In other words, within the STCP, each worker a_i has the incentive to misreport skill and working cost information, i.e., for a_i's bid $B_i' = <R_i', c_i'>$, R_i' and c_i' can differ from the real values R_i and c_i. We now present truthfulness definition applied for a social team crowdsourcing mechanism.

Definition 2 (Truthfulness). *A mechanism $M = (TF, Pay)$ is truthful for STCP if for each worker a_i with the true bid $B_i = <R_i, c_i>$ and for any other bid $B_i' = <R_i', c_i'>$, $R_i \neq R_i'$ or $c_i \neq c_i'$, we have $u_i(B_i, B_{-i}) \geq u_i(B_i', B_{-i})$, where B_{-i} is other workers' bids.*

Moreover, the mechanism *M* should also satisfy the property of individual rationality, ensuring each worker receives non-negative utility when he behaves honestly.

Definition 3 (Individual Rational). *An individual worker receives the non-negative utility when he behaves truthfully, i.e., $u_i = p_i - c_i \geq 0$.*

In addition to satisfying truthful and individual rational properties, the mechanism *M* should also be computationally efficient, i.e., the social team formation function *TF* and payment function *Pay* should be computationally feasible. Consequently, the typical VCG truthful mechanism [29], which aims to form the optimal team, is not applicable as STCP is NP-hard. Actually, the state-of-the-art team formation technique [15] is infeasible on the small-scale social team crowdsourcing instances where $k = 5$ and $n = 100$, because its exponential $O(n^k)$ computations to return the optimal team. Against this background, to reduce time complexity while guaranteeing truthfulness and optimizing social welfare, we develop two computationally efficient mechanisms for small- (i.e., task size *k* is small) and large-scale (i.e., task size *k* is large) applications in Sections 4 and 5, respectively.

4 Towards A Small-Scale Social Team Crowdsourcing Mechanism

The key principle of the small-scale-oriented mechanism is that given a social network *SN*, rather than consuming expensive $O(n^k)$ time to form an optimal team in *SN*, we first extract a tree network from the *SN* (Sections 4.1 and 4.2), and propose an optimal fixed-parameter $O(n^2 4^k)$ time mechanism in the extracted tree (Section 4.3).

4.1 Tree Network Extraction

This tree extraction principle is inspired by large-scale social network analysis researches [42]. Given a social network $SN = <A, E>$, we try to extract a tree network Γ that preserves as much social connection information as that in the original network *SN*. Intuitively, we posit that when two workers in an original network *SN* are socially close, they should be as socially close as possible in the extracted tree network Γ. Based on the related definition used in sociology [43], we first define the social closeness of a worker and a network as follows.

Definition 4 (Worker and Network Closeness). *Given a social network $SN = <A, E>$, the closeness $CL(a_i, SN)$ of a worker a_i is defined as how close a_i connects with other workers $a_j \neq a_i$, i.e., $CL(a_i, SN) = \sum_{a_j \neq a_i} 1/d(a_i, a_j, SN)$, where $d(a_i, a_j, SN)$ is the shortest social distance between a_i and a_j in SN. When a_i and a_j are disjointed, $1/d(a_i, a_j, SN) = 0$. Similarly, network closeness $CL(SN)$ can be defined as how social closely these workers A connect with one another, i.e., $CL(SN) = \sum_{a_i \in A} CL(a_i, SN) = \sum_{a_i \in A} \sum_{a_j \neq a_i} 1/d(a_i, a_j, SN)$.*

The larger the value of $CL(SN)$, these workers A are more socially close in *SN*. Next, we define the tree extraction problem as an optimization problem by minimizing the difference between *SN* and extracted tree Γ on network closeness.

Definition 5 (Tree Extraction Problem). *Given a social network $SN = <A, E>$, we would like to extract a tree Γ from SN such that Γ captures as much social connection information as possible in SN,, i.e., minimize $|CL(SN) - CL(\Gamma)|$.*

Deleting a connection in *SN* will decrease network closeness, we can conclude that $CL(SN) \geq CL(\Gamma)$. Now the tree extraction problem is equivalent to extracting the optimal tree Γ^* from the *SN* with maximal network closeness, i.e., $\Gamma^* = \mathrm{argmax}_\Gamma CL(\Gamma)$. This problem of maximizing tree network closeness is a new variant of the NP-hard network design problem [44]. This problem has a different objective function from ours and thus such approaches are not applicable. To this end, we present a polynomial approximation algorithm for network closeness maximization, shown in Algorithm 1. In Steps 2-5 of Algorithm 1, we first locate each worker $a_i \in A$ as the root worker and for such a root worker a_i, we compute its worker closeness $CL(a_i)$ (Step 3). The worker a_r with the highest level of closeness becomes the root of the extracted tree Γ (Step 6). In Steps 7-11, the extracted tree Γ is constructed iteratively: first, workers with one unit of social distance from a_r are added as first level children *Children1* of the tree root a_r, i.e., $\forall a_i \in Children1 : d(a_r, a_j) = 1$. Next, workers with two units of social distance from a_r are denoted as second level children *Children2*, and for each worker $a_i \in Children2$, his parent is the previous worker of a_i along the shortest path between a_r and a_i, i.e., when the shortest path between a_r and a_i is $Path = \{a_r, a_p, \ldots, a_q, a_i\}$, then a_i's parent worker is a_q. This procedure of adding tree nodes continues until all nodes are visited.

Algorithm 1. Tree Extraction

```
Input: Social Network $SN = <A, E>$;
Output: Tree $\Gamma$.
1. Set $a_r = \emptyset$, $d = 1, \max = 0$;
2. For each $a_i \in A$
3.     Set $CL(a_i) = \Sigma_{ai \neq aj} 1/d(a_i, a_j)$;
4.     If $CL(a_i) >$ max, then $\max = CL(a_i)$ and $a_r = a_i$;
5. End for
6. $\Gamma = \Gamma \cup \{a_r\}$;
7. While $A \neq \emptyset$
8.     For each $a_i \in A : d(a_r, a_j) = d$
9.         Set $A = A \setminus \{a_i\}$, $\Gamma = \Gamma \cup \{a_i\}$ and $d = d + 1$.
10.    End for
11. End while
```

We provide the approximation ratio of Algorithm 1 on maximizing tree network closeness.

Lemma 2. *Let $CL(\Gamma)$ and $CL(\Gamma^*)$ be the tree network closeness returned by Algorithm 1 and the optimization, respectively. Then, the approximation ratio of Algorithm 1 on network closeness is $CL(\Gamma)/CL(\Gamma^*) \geq 1/(D+1)$, where D is the diameter of the tree network Γ, i.e., $D = \max_{a_i, a_j} d(a_i, a_j, \Gamma)$.*

An illustrative example of transforming a social network to a tree network is shown in Figs. 2a and 2b. After extracting the tree network Γ, we present the dynamic programming-based

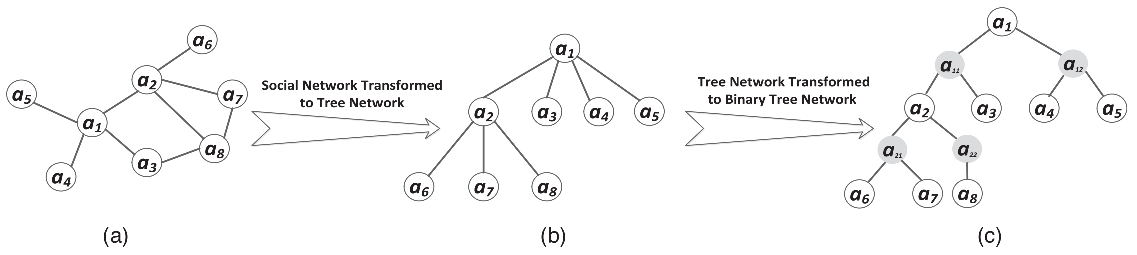

Social Network Transformed to Tree Network
Tree Network Transformed to Binary Tree Network
(a)
(b)
(c)

Fig. 2. (a) The social network. (b) The tree network with the largest network closeness transformed from (a). (c) The binary tree network transformed from the tree network in (b) where gray workers are newly added auxiliary workers.

optimal team formation approach for Γ. The key principle behind the dynamic programming approach is to consider the following two cases for the root worker a_r and recursively for all other workers down a_r. In the first case, we do not select root a_r as a team member. We thus need to find a team of workers from one of a_r's children subtrees, e.g., Γ_{ax}, to satisfy task T's skills O_T where a_x is one of a_r's children. Throughout this paper, we refer to Γ_{ai} as the subtree consisting of the root worker a_i and workers down a_i in Γ, e.g., in Fig. 2b, $\Gamma_{a2} = \{a_2, a_6, a_7, a_8\}$. In the second case, we select a_r as a winner. We need to partition the remaining unsatisfied skills $O_T \setminus R_r$ optimally among a_r's children subtrees. However, computing all possible partitions requires $O(l^k)$ computation, where l is the number of a_r's children, making this dynamic programming approach intractable when l is large. To reduce such partition complexity, in the next section, we attempt to transform tree Γ into an equivalent binary tree Γ^β.

4.2 Binary Tree Transformation

We transform the original tree Γ to a binary tree Γ^β as follows: start from the root worker a_r of tree Γ and assume that a_r has $l > 2$ children workers, i.e., $Children(a_r) = \{a_1, a_2, \ldots, a_l\}$. We can then replace a_r and a_r's children by a binary tree of depth $\lfloor \log_2 l \rfloor + 1$ where the root worker is still a_r and leaf workers are $\{a_1, a_2, \ldots, a_l\}$ and $\lfloor x \rfloor = \max\{t \in \mathbb{Z} | t \le x\}$. The newly added auxiliary internal workers between a_r and $\{a_1, a_2, \ldots, a_l\}$ in Γ^β neither possess any skill nor incur any working cost. Moreover, once their parent workers are selected, they will be selected as well. This transformation repeats recursively for each a_r's child $a_j \in \{a_1, a_2, \ldots, a_l\}$ and for its children's children if any. An illustrative example of transforming an arbitrary tree to a binary tree is shown in Figs. 2b and 2c. The binary tree transformation has two desirable properties, shown as follows.

Lemma 3. *For the original tree Γ with n workers, the number of workers in the transformed binary tree Γ^β is at most 3n.*

Lemma 4. *The optimal solution of the STCP in binary tree Γ^β is equivalent to the optimal solution of the STCP in tree Γ.*

This is attributable to the fact that auxiliary workers in newly added Γ^β do not offer any skills, and once their parent worker is selected, they will be selected as well without incurring any working cost.

4.3 Optimal Truthful Mechanism in Binary Trees

In this section, we propose an optimal truthful mechanism $M = <TF, Pay>$ in the binary tree Γ^β where the social team formation function TF is the optimal dynamic programming-based algorithm and the payment function is the VCG type payment function.

For each worker $a_i \in \Gamma^\beta$, let $S(a_i, 1, U)$ be the optimal team formed to satisfy skills U in the subtree Γ^β_{ai}, where a_i is selected as a winner. Let $W(a_i, 1, U)$ be the welfare of the team $S(a_i, 1, U)$. Similarly, let $W(a_i, 0, U)$ be the welfare of the optimal team $S(a_i, 0, U)$ formed in Γ^β_{ai} without selecting a_i. Let $l(a_i)$ and $r(a_i)$ be a_i's left and right child, respectively. We implement the following dynamic programming recurrence for each worker a_i:

$$W(a_i, 1, U) = \max \begin{cases} W(r(a_i), 1, U \backslash R_i) - c'_i, \\ W(l(a_i), 1, U \backslash R_i) - c'_i, \\ \max_{U' \in U \backslash R_i} [W(r(a_i), 1, U') + W(l(a_i), 1, U \backslash \{R_i \cup U'\}) - V_T - c'_i]. \end{cases} \tag{7}$$

and

$$W(a_i, 0, U) = \max\{W(r(a_i), 0, U), W(r(a_i), 1, U), W(l(a_i), 0, U), W(l(a_i), 1, U)\}. \tag{8}$$

Formula (7) corresponds to the case of selecting worker a_i as a winner. Once a_i is selected, to form a connected team, three scenarios need to be considered:

- Only the right child $r(a_i)$ is selected as the winner and remaining unsatisfied skills $U \backslash R_i$ are assigned to a_i's right subtree $\Gamma^\beta_{r(a_i)}$. This case corresponds to the first term in Formula (7).
- Only the left child $l(a_i)$ is selected as the winner and remaining unsatisfied skills $U \backslash R_i$ are assigned to a_i's left subtree $\Gamma^\beta_{l(a_i)}$. This case corresponds to the second term in Formula (7).
- Both of the right child $r(a_i)$ and left child $l(a_i)$ are selected as winners and unsatisfied skills $U \backslash R_i$ are partitioned optimally among a_i's right subtree $\Gamma^\beta_{r(a_i)}$ and left subtree $\Gamma^\beta_{l(a_i)}$. There is only one task to be satisfied and a copy of the task profit V_T should be subtracted from the aggregate welfare of a_i's right and left subtrees. This case corresponds to the third term in Formula (7).

Formula (8) determines the outcome of not selecting a_i as the winner, where unsatisfied skills U must be satisfied by a_i's right children's subtree $\Gamma^\beta_{r(a_i)}$ or left children's subtree $\Gamma^\beta_{l(a_i)}$. For this dynamic programming approach, we have the following initial conditions: $W(\emptyset, 0, \emptyset) = V_T, W(a_i, 1, \emptyset) = V_T - c'_i$ and $\forall U \neq \emptyset$, $W(\emptyset, 0, U) = 0$. Finally, the optimal team formed in Γ^β with maximum social welfare is returned by the function of $max\{W(a_r, 0, O_T), W(a_r, 1, O_T)\}$.

Let $S(\Gamma^\beta)$ be the optimal team formed in the binary tree Γ^β and $W(\Gamma^\beta)$ be the team $S(\Gamma^\beta)$'s welfare returned using the dynamic programming (DP) approach. During team formation, we further define a skill allocation $\Phi(S(\Gamma^\beta)) = \{\varphi(a_1), \ldots, \varphi(|S(\Gamma^\beta)|)\}$ that is a match between team members and skills, where $\varphi(a_i) = R'_i \cap O_T$ indicates the skills provided by team member $a_i \in S(\Gamma^\beta)$.

We adopt the VCG type payment rule [29] to define the payment p_i of each winner worker $a_i \in S(\Gamma^\beta)$ as:

$$p_i = (W(\Gamma^\beta) + c'_i) - \max\{W(\Gamma^\beta_{r(a_i)}), W(\Gamma^\beta_{l(a_i)}), W(\Gamma^\beta \backslash \Gamma^\beta_{a_i})\} \\ - (1 - \rho_i(\varphi(a_i), R_i))V_T. \tag{9}$$

The value $W(\Gamma^\beta_{r(a_i)}) = V_T - \sum_{a_j \in S(\Gamma^\beta_{r(a_i)})} c'_j$ is the welfare of the team $S(\Gamma^\beta_{r(a_i)})$, which is the optimal team of a'_is right subtree $\Gamma^\beta_{r(a_i)}$. Similarly, $W(\Gamma^\beta_{l(a_i)}) = V_T - \sum_{a_j \in S(\Gamma^\beta_{l(a_i)})} c'_j$ value is the optimal welfare of a'_is left subtree $\Gamma^\beta_{l(a_i)}$. The $W(\Gamma^\beta \backslash \Gamma^\beta_{a_i}) = V_T - \sum_{a_j \in S(\Gamma^\beta \backslash \Gamma^\beta_{a_i})} c'_j$ value is the optimal welfare of the subtree $\Gamma^\beta \backslash \Gamma^\beta_{a_i}$, which is generated by removing subtree $\Gamma^\beta_{a_i}$ from the binary tree Γ^β. For example, in Fig. 2c, $\Gamma^\beta \backslash \Gamma^\beta_{a_2} = \{a_1, a_{11}, a_3, a_{12}, a_4, a_5\}$. Finally, the value $W(\Gamma^\beta) + c'_i$ is the optimal welfare returned from the binary tree Γ^β without considering worker a'_is working cost c'_i. The Boolean function $\rho_i(\varphi(a_i), R_i)$ indicating whether the team member can provide skills successfully, which is defined as

$$\rho_i(\varphi(a_i), R_i) = \begin{cases} 1, & \varphi(a_i) \subseteq R_i, \\ 0, & otherwise. \end{cases} \tag{10}$$

For the agent that cannot provide the skills he reports (i.e., $\rho_i(\varphi(a_\text{i}), R_i) = 0$), he will be punished to being paid additional V_T payment to the requester.

Finally, a formal description of the optimal truthful mechanism for the small-scale social team crowdsourcing application is shown in Algorithm 2. In Algorithm 2, after forming the team $S(\Gamma^\beta)$, in Steps 4-6, we require each team member $a_i \in S(\Gamma^\beta)$ (according to their order of joining the team), to provide their available skills $\varphi(a_\text{i}) = R'_i \cap O_T$. In the following, we present desirable properties (i.e., computationally efficient, individual rational and truthfulness) of the proposed small-scale-oriented social team crowdsourcing mechanism.

Lemma 5. *For a STCP with n workers and a task that requires k types of skills, the running time of Algorithm 2 is $O(n^2 4^k)$.*

Theorem 1. *The proposed small-scale-oriented mechanism (i.e., Algorithm 2) is truthful and individual rational.*

Algorithm 2. Optimal Truthful Mechanism for Small-Scale Social Team Crowdsourcing

```
Input: Social Network $SN = <A, E>$, Bids $B'_i = <R'_i, c'_i>$, Task $T = <V_T, O_T>$;
Output: Team $S(\Gamma^\beta)$, Payment $Pay = (p_1, p_2, \ldots, p_n)$.
1. $\Gamma \leftarrow$ Tree Extraction $(SN)$;
2. $\Gamma^\beta \leftarrow$ Binary Tree Transformation $(\Gamma)$;
3. $(S(\Gamma^\beta), W(\Gamma^\beta)) \leftarrow$ Dynamic Programming-DP $(\Gamma^\beta)$;
4. For $1 \le i \le |S(\Gamma^\beta)|$
5.     $\varphi(a_\text{i}) = R'_i \cap O_T$, $O_T = O_T \backslash R'_i$;
6. End for
7. Set $p_i = 0$, $\forall a_i \in A$;
8. For each $a_i \in S(\Gamma^\beta)$
9.     $W(\Gamma^\beta \backslash \{a_i\}) \leftarrow max\{W(\Gamma^\beta_{r(a_i)}), W(\Gamma^\beta_{l(a_i)}), W(\Gamma^\beta \backslash \Gamma^\beta_{a_i})\}$;
10.    $p_i = ((W(\Gamma^\beta) + c'_i) - W(\Gamma^\beta \backslash \{a_i\})) - (1 - \rho_i(\varphi(a_\text{i}), R_i))V_T$.
11. End for
```

Proof. *Truthful:* we first prove that for each worker $a_i \in A$, reporting his truthful skills R_i is the dominated strategy for any proposed cost c'_i *(Sub-theorem 1)*. Next we prove that for each worker $a_i \in A$, reporting his truthful cost c_i is the dominated strategy *(Sub-theorem 2)*.

For Sub-theorem 1. we only need to show that *i)* worker a_i declaring his true skills R_i wins and if he under-repots his skills $R_i{}' \subset R_i$ might lose without receiving any payment. This is because under-reporting skills does not change the workers' network SN as well as the transformed binary tree Γ^β. Under-reporting skills can make a_i excluded in Algorithm 2 or make Algorithm 2 hire more team members, both are not beneficial for a_i. And *ii)* worker a_i declaring his true skills R_i wins and if he over-reports his skills $R_i{}' \supset R_i$ still wins but receives a negative payment. This is because over-reporting skills does not change the structure of the binary tree Γ^β and leads to form a smaller size team by requiring a_i to provide his over-reported skills $R'_i \backslash R_i$. Once a_i is detected by over-reporting skills (i.e., $\rho_i(\varphi(a_\text{i}), R_i) = 0$), he will be penalized to provide V_T) payment to the requester, which will yield a negative utility for a_i. Similarly, the truthful skill reporting property also holds for the truthful worker that loses the team formation.

For Sub-theorem 2. We first consider the case in which the worker wins in social team formation by reporting his true cost c_i and skills R_i. Let $u_i \ge 0$ be his utility when reporting his true information c_i and R_i. Suppose that a_i proposes a different cost $c'_i \ne c_i$, and achieves the utility u'_i. We divide this case into two scenarios:

1. Worker a_i under-reports his cost $c'_i < c_i$. Worker a_i must still win, and the structure of tree Γ^β does not change. Worker a'_is utility becomes $u'_i = p'_i - c_i = (W(\Gamma^\beta) + c'_i) - W(\Gamma^\beta \backslash \{a_i\}) - c_i = V_T - \Sigma_{a_j \in S(\Gamma^\beta) \backslash \{a_i\}} c'_j - W(\Gamma^\beta \backslash \{a_i\}) - c_i = u_i$.
2. Worker a_i over-reports his cost $c'_i > c_i$. We further distinguish two scenarios:
   a. Worker a_i still wins, and his utility becomes $u'_i = p'_i - c_i = (W(\Gamma^\beta) + c'_i) - W(\Gamma^\beta \backslash \{a_i\}) - c_i = V_T - \Sigma_{a_j \in S(\Gamma^\beta) \backslash \{a_i\}} c'_j - W(\Gamma^\beta \backslash \{a_i\}) - c_i = u_i$.
   b. Worker a_i loses with utility $u'_i = 0 \le u_i$.

We next consider the case in which the worker loses in social team formation by reporting true cost c_i and skills R_i. Let $u_i = 0$ be his utility when reporting his true skills R_i and cost c_i. Suppose that the a_i proposes a different cost $c'_i \ne c_i$, and achieves the utility u'_i. We also divide this case into two scenarios

1. Worker a_i under-reports his cost $c'_i < c_i$. We also further distinguish two scenarios:
   a. Worker a_i wins and his utility becomes

$$\begin{aligned} u_i{}' &= p_i{}' - c_i \\ &= W(\Gamma^{\beta'}) + c_i{}' - W(\Gamma^{\beta'} \backslash \{a_i\}) - c_i \\ &= (W(\Gamma^{\beta'}) + c_i{}') - W(\Gamma^\beta) - c_i \end{aligned} \tag{11}$$

$$\le V_T - \Sigma_{a_j \in S(\Gamma^{\beta'}) \backslash \{a_i\}} c_j{}' - c_i - W(\Gamma^\beta) \le 0 \tag{12}$$

$S'(\Gamma^{\beta'})$ is the team returned by Algorithm 2 when a_i reports cost c_i', and $W(\Gamma^{\beta'})$ is the social welfare of team $S'(\Gamma^{\beta'})$. The Eq. (11) holds because $W(\Gamma^{\beta'}\backslash\{a_i\}) = W(\Gamma^{\beta})$, where Algorithm 2 forms the optimal team Γ^{β} without involving a_i. The In eq. (12) holds because when a_i reports true cost c_i, the social welfare $W(\Gamma^{\beta})$ of team $S(\Gamma^{\beta})$ is larger the team $S(\Gamma^{\beta'})$. This is because when a_i is truthful, Algorithm 2 always forms the team with the maximal social welfare.

b. Worker a_i still loses with utility $u_i' = 0 = u_i$.

2. Worker a_i over-reports his cost $c_i' > c_i$, where a_i must still lose with utility $u_i' = 0 = u_i$.

In summary, we can conclude that reporting true skills R_i and cost c_i is each worker a_i's dominated strategy of maximizing his utility.

Individual Rational. According to Formula (9), for each team member $a_i \in S(\Gamma^{\beta})$, his utility is $u_i = p_i - c_i = W(\Gamma^{\beta}) - \max\{W(\Gamma^{\beta}_{r(a_i)}), W(\Gamma^{\beta}_{l(a_i)}), W(\Gamma^{\beta}\backslash\Gamma^{\beta}_{a_i})\} \geq 0$. This is because the DP-based social team formation can always form a more beneficial team with larger welfare $W(\Gamma^{\beta})$ in the global tree Γ^{β} than the welfare of the team formed in the left subtree $\Gamma^{\beta}_{r(a_i)}$, right subtree $\Gamma^{\beta}_{l(a_i)}$, or the remaining subtree $\Gamma^{\beta}\backslash\Gamma^{\beta}_{a_i}$. Thus, Algorithm 2 satisfies individual rational. □

5 Towards a Large-Scale Social Team Crowdsourcing Mechanism

Although forming an optimal team in a binary tree is efficient in terms of maximizing social welfare, this is still computationally expensive for large-scale applications for which the task size *k* is large. Therefore, in this section, we present a polynomial $O(k^3n^2)$ time mechanism to address the large-scale social team crowdsourcing applications. The polynomial mechanism includes the greedy monotonous social team formation algorithm (Section 5.1) and the threshold payment algorithm (Section 5.2).

5.1 Greedy Social Team Formation

STCP is a networked variant of a *setcover* problem. The greedy principle is a natural fit, as it guarantees monotonicity when workers are sorted in increasing order of their marginal contribution-per-cost value.

Definition 6 (Worker Marginal Contribution and Marginal Contribution-per-Cost). *Given a skill set U, the marginal contribution of a worker a_i with respect to U is defined as $\pi(a_i, U) = |U \cap R_i|$, i.e., the number of skills a_i can provide to satisfy U and a_i's marginal contribution-per-cost $\varepsilon(a_i, U)$ is defined as the ratio between a_i's marginal contribution and working cost, i.e., $\varepsilon(a_i, U) = \pi(a_i, U)/c_i$.*

Before presenting the greedy social team formation algorithm, we first provide another two definitions that are useful to design the greedy algorithm.

Definition 7 (Accessible Workers and Skills of a Team). *Given a worker team G and a skill set U, the accessible workers AA(G) of team G are defined as team members' neighbor workers, i.e., $AA(G) = \{a_j | a_i \in G, a_j \in N_i\}$, and the accessible skills AS(G, U) are defined as G's accessible workers' skills that can satisfy U, i.e., $AS(G, U) = \{s_x | a_i \in AA(G), s_x \in R_i \cap U\}$.*

Definition 8 (Complementary Workers of a Team). *Given a worker team G and a skill U, a worker a_i is complementary to this team G when and only when he can contribute skills to U, i.e., $\pi(a_i, U) > 0$ and its neighbors can provide newly accessible skills to G, i.e., $AS(\{a_i\}, U)\backslash AS(G, U) \neq \emptyset$. We use the function $\delta(a_i, U, G)$ to denote whether a_i is complementary to team G with respect to U : $\delta(a_i, U, G) = 1$, when and only when $\pi(a_i, U) > 0$ and $AS(\{a_i\}, U)\backslash AS(G, U) \neq \emptyset$; otherwise, $\delta(a_i, U, G) = 0$.*

In the Appendix, Section A5 available in the online supplemental material, we use an example to illustrate Definitions 6, 7 and 8 clearly.

Algorithm 3. Greedy social Team Formation

```
Input: Social Network SN = $<A, E>$, Bids $B_i' = <R_i', c_i'>$, Task $T = <V_T, O_T>$;
Output: Team $Q$, skill-allocation $\Phi$, and team root $a_r$.
1. Create Queue($Q$);
2. Initialize $a_r = \mathrm{argmax}_{a_i \in A}\varepsilon(a_i, O_T) \times \delta(a_i, O_T, \emptyset)$;
3. Insert Queue $(Q, a_r)$ and set $O_T = O_T\backslash R_{a_r}'$;
4. While $O_T \neq \emptyset$, do
5.     If $AS(Q, O_T) = O_T$, then set $I = AA(Q)$;
6.     Set $I = \{a_x | a_x \in AA(Q) : \delta(a_x, O_T, Q) = 1|\}$;
7.     Select $a^* = \mathrm{argmax}_{a_x \in I}\varepsilon(a_x, O_T)$;
8.     Insert Queue $(Q_T, a^*)$, $\varphi(a^*) = R_{a^*}' \cap O_T$, $O_T = O_T\backslash R_{a^*}'$.
9. End while
```

Now we present the greedy social team formation algorithm, i.e., Algorithm 3. The key principle of this greedy algorithm is that team members are selected according to a worker's social structure and marginal contribution-per-cost value. To form a collaborative team, we only need to identify a tree that connects a team of workers. To this end, in Step 2 of Algorithm 3, we first determine the worker a_r with the largest marginal contribution-per-cost value as the team root. After determining the root worker a_r, we build a tree structure team from its social contextual workers round by round (Steps 4-9). Before building this team, we create a queue structure Q to store team members (Step 3). The queue structure is used to keep track of the round in which each team member joins. For each round, we select the optimal accessible and complementary worker $a^* = \mathrm{argmax}_{ax \in I}\varepsilon(a_x, O_T)$ with the largest positive marginal contribution-per-cost value as a new team member (Step 7). The set I is current team Q's accessible and complementary workers, which can be represented by $I = AA(Q)$ if current team Q can access enough skills to satisfy O_T (Step 5), otherwise, $I = \{a_x | a_x \in AA(Q) : \delta(a_x, O_T, Q) = 1|\}$ (Step 6). Worker a^* must be accessible to the team Q because the participation of an accessible worker ensures that the formed team is connected, facilitating collaborative team building. Worker a^* must also be complementary to the team Q because complementary worker participation grants team Q access to additional skills for professional team building. In Step 8, the selected agent a^* joins team Q, and is required to provide the available skills $\varphi(a^*) = R_{a^*}' \cap O_T$. We select this kind of accessible and complementary workers round by round until the team formed is professional enough such that task skills are satisfied (Step 4). Compared to Algorithm 1, the necessity of the additional operations of Steps 4-9 is that Algorithm 3 finds the local optimal team member from current team's neighbor workers. This local social team formation process might fall into forming the local incomplete team whose team members are not professional to satisfy task skills and these team members cannot find any neighbor worker that can provide the lacking skills. On the other hand, Algorithm 1 is a global algorithm that forms a professional team from the whole tree network. Thus, Algorithm 1 does not need such additional operations of Steps 4-9 of Algorithm 3.

In the following, we present the complexity, monotonicity (Lemma 6) and approximation ratio (Theorem 7) results of Algorithm 3.

Lemma 6. *Algorithm 3 is monotone with polynomial-time $O(k^2n^2)$ complexity.*

Lemma 7. *For any social team crowdsourcing problem (STCP), where each worker a_i's cost $c_i \in [c_{min}, c_{max}]$, c_{min} (resp. c_{max}) is the minimum (resp. maximum) cost that a worker can bid. Then the team cost of Algorithm 3 is at most a factor of $O(c_{max}/c_{min})$ larger than the optimal team cost.*

5.2 Threshold Payment for the Greedy Social Team Formation Algorithm

Algorithm 4 presents the threshold payment rule for the greedy social team formation algorithm (i.e., Algorithm 3). Let Q be the winning team returned by Algorithm 3. Next, we compute the payment for each team member $a_i = Q(i)$ in their order of joining team Q, where a_i denotes the team member that joins Q in the ith round (Step 2). The threshold payment p_{a_i} paid to a_i is the maximal cost that a_i can bid and still be selected by Algorithm 3. To achieve this value, we extend Singer's [45] threshold payment rule, where the underlying SN is ignorant.

We first briefly describe Singer's payment rule as follows: they first use the marginal contribution-per-cost sorting rule to form an alternative winner team Q' without a_i's participation and then find the maximal cost that worker a_i can declare in order to be selected in the jth $(1 \leq j \leq |Q'|)$ round in team Q'. There are two main advantages of our proposed payment rule over Singer's within the SN setting. 1) Compared to Singer's payment rule of building an alternative team Q' without a_i's participation, our proposed payment rule forms an alternative team Q' by assuming a_i bids a very large cost value MAXVALUE (Step 4). This technique is used to keep the underlying network structure unchanged, which is essential for truthful social team formation mechanism design. 2) Compared to Singer's rule of considering all possible winning rounds $[1, |Q'|]$ to compute a_i's payment, our proposed rule makes use of the underlying local network structure and narrows the range of possible winning rounds among $[i, |Q'|]$, where i is the winning round of a_i in team Q. This advantage makes the proposed payment rule not only guarantee truthfulness but also reduce time complexity.

Next, we present how our payment rule computes payment p_{a_i} for a_i in detail (Steps 5-13). Suppose that a_i attempts to win in the jth round $(i \leq j \leq |Q'|)$ in Q', and the maximal cost he can bid is $c^*_{a_i,j} = \pi(a_i, U_j')/\varepsilon(a_j', U_j')$, where U_j' is unsatisfied skills after the first $j-1$ workers have joined team Q' (Step 7). Namely, $U_j' = O_T \backslash \{R_{a'x} | a_x' \in Q'(j)\}$, where $Q_j' = \{Q'(x) | 1 \leq x \leq j-1\}$ denotes the first $j-1$ team members who join Q' (Step 6). As the marginal contribution $\pi(a_i, U_j')$ and marginal contribution-per-cost $\varepsilon(a_j', U_j')$ both monotonically decrease with j, the value $c^*_{a_i,j}$ might present arbitrary behaviors with respect to j. Thus, it is reasonable to take the maximum of this value $c^*_{a_i,j}$ as the maximal cost that a_i can declare to win among the rounds $[i, |Q'|]$ (Steps 8-9). In Step 12, if team member a_i can provide the reported skills successfully, i.e., $\rho_i(\varphi(a_i), R_i) = 1$, he will receive the *max* payment; otherwise, he will receive a considerable negative payment $max - V_T$.

In the Appendix, Section A8 available in the online supplemental material, we also use an example to illustrate how Algorithm 4 works. In the following, we present the threshold payment property of Algorithm 4.

Algorithm 4. Threshold Payment

```
Input: Social Network SN = $<A, E>$, Bids $B_i' = <R_i', c_i'>$, Task $T = <V_T, O_T>$, winner team $Q$, and skill allocation $\Phi$
Output: Payment $Pay = (p_1, p_2, \ldots, p_n)$.
 1. Initialize $Pay = \{0, \ldots, 0\}$;
 2. For $i = 1$ to $|Q|$
 3.    Set $max = 0$ and $a_i = Q(i)$; /*return the ith element of Q*/
 4.    Set $Q' = \mathbf{GTF}(c_i = \mathrm{MAXVALUE}, B_{-i}, T)$;
 5.    For $j = i$ to $|Q'|$
 6.       Set $a_j' = Q'(j)$ and $Q_j' = \{Q'(x) | 1 \leq x \leq j-1\}$;
 7.       Set $U_j' = O_T \backslash \{R_{a'x} | a_x' \in Q_j'\}$;
 8.       If $\delta(a_i, U_j', Q_j') = 1$ && $\pi(a_i, U_j')/\varepsilon(a_j', U_j') > max$, then
 9.          $max = \pi(a_i, U_j')/\varepsilon(a_j', U_j')$;
10.       End if
11.    End for
12.    $p_i = max - (1 - \rho_i(\varphi(a_i), R_i)) \cdot V_T$.
13. End For
```

Lemma 8. *Algorithm 4 returns the threshold payment for each winner team member.*

Proof. For team Q returned by Algorithm 3, let i $(1 \leq i \leq |Q|)$ be the winning round of team member $a_i \in Q$. Now assume that a_i submits a very large cost value MAXVALUE, an alternative team Q' will be returned by Algorithm 3. Let $l \in [i, |Q'|]$ be the index at which $\pi(a_i, U_l')/\varepsilon(a_l', U_l')$ is maximal, where a_l' is the lth team member in team Q'. As Algorithm 4 shows, this value $\pi(a_i, U_l')/\varepsilon(a_l', U_l')$ is the payment p_{a_i} for a_i. Next, we prove that p_{a_i} is the maximal cost a_i can bid to secure a place in Q'.

On one hand, bidding a cost below p_{a_i} will ensure that a_i wins. This is because in the ith round, worker a_i must be accessible and complementary to team Q. Therefore, as long as a_i declares a smaller cost than p_{a_i}, it will definitely be selected as a winner at a certain round among $[i, |Q'|]$ in Q'. Next, we show that declaring a larger cost $c_{a_i}' > p_{a_i}$ will prevent a_i from securing a place in Q'.

First, we note that declaring a cost larger than p_{a_i} will make a_i lose at any round among $[i, |Q'|]$ in Q'. This is true because p_{a_i} is the maximal cost that a_i can declare and win in a round among $[i, |Q'|]$. Now we prove that declaring a higher cost $c_{a_i}' > p_{a_i}$ also prevents a_i from winning among rounds of $[1, i-1]$ in Q'. Proof by contradiction, assume that a_i declares a higher cost $c_{a_i}' > p_{a_i}$ and wins at a round $j \in [1, i-1]$ (*Assumption 1*). Let a_j' be the jth team member in Q'. Then, we can show that a_j' is exactly the jth team member in Q because the network structure remains unchanged in Q'. Now based on *Assumption 1*, $c_{a_i}' \leq \pi(a_i, U_j')/\varepsilon(a_j', U_j')$. Recall that in team Q, the winner a_i joins Q after the jth round for two reasons.

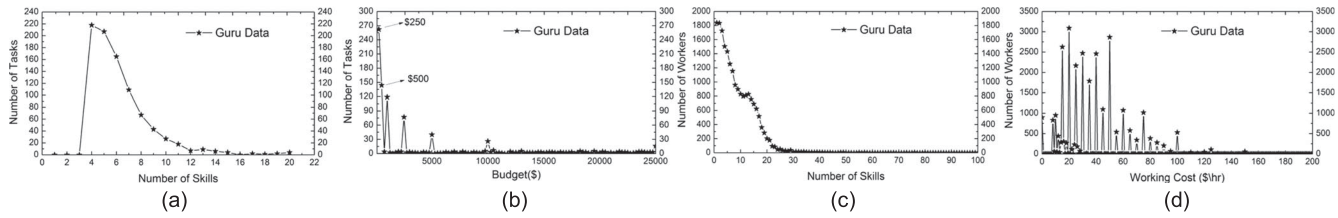


Fig. 3. The Guru's data information, where (a) task skill requirement, (b) task budget, (c) worker skill possession, and (d) workers' salary.

- *Reason 1*: a_i is accessible and complementary during the *j*th round, but it has a smaller marginal contribution-per-cost than the *j*th team member a'_j under truthful bidding, i.e., $c_{a_i} \geq \pi(a_i, U'_j)/\varepsilon(a'_j, U'_j)$. For this scenario, we can derive the following contradiction: $c_{a_i} \geq \pi(a_i, U'_j)/\varepsilon(a'_j, U'_j) \geq c'_{a_i} > p_{a_i} \geq c_{a_i}$. The last inequality holds because payment p_{a_i} is larger than winner a'_is true cost c_{a_i}. Therefore, in this case, a larger cost than p_{a_i} will cause a_i to lose in any round among $[1,i-1]$.
- *Reason 2*: a_i has a larger marginal contribution-per-cost than that of a'_j, but a_i is not accessible or complementary at the *j*th round in *Q*. For this scenario, declaring a higher cost than p_{a_i} will not make a_i become accessible and complementary to team Q' before the *j*th round, as the network structure and worker's skills are not changed. In this case, a_i will never win in any round among $[1,i-1]$ neither.

Now we have that p_{a_i} is the maximum cost (i.e., threshold payment) a_i can bid to still be selected by Algorithm 3. □

Finally, we present the individual rational, computationally efficient and truthfulness properties of the large-scale-oriented social team crowdsourcing mechanism.

Theorem 2. *The large-scale-oriented social team crowdsourcing mechanism that consists of the greedy social team formation algorithm (i.e., Algorithm 3) and threshold-based payment rule (i.e., Algorithm 4) is truthful, individual rational and computationally efficient with $O(k^3n^2)$ time complexity.*

Proof. *Truthful*: we first prove that for each worker $a_i \in A$, reporting his truthful skills R_i is the dominated strategy for any proposed cost c_i *(Sub-theorem 1)*. Next we prove that for each worker a_i, reporting his truthful cost c_i is the dominated strategy *(Sub-theorem 2)*.

For Sub-theorem 1: similar to Theorem 1, we have that by penalizing the over-reporting behavior, workers always report their skills truthfully for any proposed cost with the aim of maximizing their utility.

For Sub-theorem 2: For such a single parameter mechanism design problem where each worker has one type of private value, the mechanism $M = (TF, Pay)$ is truthful if and only if the social team formation function *TF* is monotone and the payment function *Pay* is the threshold payment [46]. Derived from Lemmas 6 and 8, the greedy social team formation mechanism (i.e., Algorithm 3) is monotone and the payment rule (i.e., Algorithm 4) is based on the threshold payment. Therefore, the large-scale-oriented social team crowdsourcing mechanism is truthful. *Individual Rational*: This is true because Algorithm 4 returns the maximal working cost the winner can bid and still wins (including the case where a_i bids c_i). Thus, the winner can always achieve the non-negative utility $u_i = p_i - c_i \geq 0$. *Computationally Efficient*: Algorithm 4 computes the payment for each team member in winner team Q (where $|Q| \leq k$), and for each winner, it needs to evoke Algorithm 3 once, which requires the execution of $O(k^2n^2)$ computations. Therefore, the running time of Algorithm 4 is $O(k^3n^2)$. □

6 Experimental Validation and Analyses

6.1 Experimental Setting

6.1.1 Dataset

To validate the proposed mechanisms under realistic settings, we use a dataset from the popular web-based crowdsourcing platform Guru, where social team crowdsourcing is needed for complex task completion. We collect 887 tasks and 28,808 workers in the IT field to analyze tasks and workers' features. For each task, we mainly record two types of information: 1) skills it requires and 2) budget used for task completion. For each worker, we record two types of information: 1) the skills he offers and 2) the salary (dollars/hour) he requires (i.e., working cost). The task and worker data are described in Fig. 3, from which we make the following observations: *i*) Fig. 3a records task skill requirement information, from which we find that roughly 98% of task sizes are not larger than 12 and almost all task sizes are not larger than 20. *ii*) Fig. 3b records task budget information, from which we observe that almost 90% of task budgets are distributed over a range of 250, 500, 750 and 1,000. *iii*) Fig. 3c records worker skill information, from which we can conclude that almost all of the workers possess less than 30 type of skills. *iv*) Fig. 3d presents worker salary requirement information, from which we can determine that workers' salary values are randomly distributed.

6.1.2 Social Network Setting

A social network consists of workers and their connections. In the following, we construct these two components, respectively. *Workers*: we first extract the top 40 skills that are most frequently required by the collected tasks, and then we collect 928 workers listed on Guru that can offer at least one of these top 40 skills. *Workers' Social Connection*. we connect these workers by three typical social network structures (Random (Random), Small-World (SMW) and Scale-Free (SF)) and each with a network degree of 8 ([47] provides a detailed description on how to construct these networks).

6.1.3 Comparison Methods and Performance Metrics

We compare the proposed social team crowdsourcing mechanisms, i.e., an optimal mechanism in a tree network *OPT-Tree* and a greedy mechanism *Greedy* with the other two mechanisms:

- The benchmark VCG-based optimal mechanism (*OPT*). This mechanism first adopts the exponential

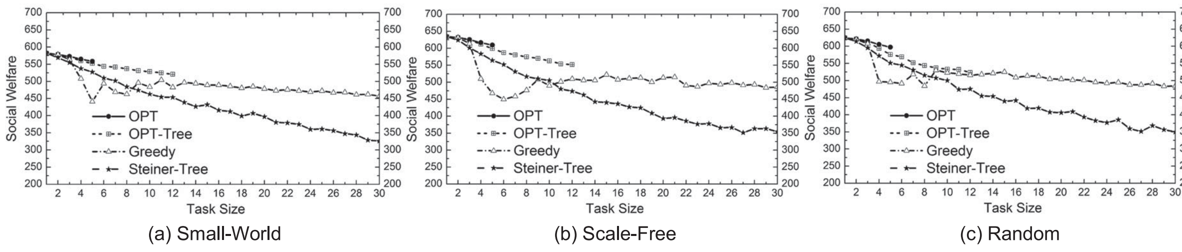


Fig. 4. The social welfare of OPT, OPT-Tree, Greedy, and Steiner-Tree in different networks.

coalition formation approach to build an optimal feasible team in the social network [15], and adopts the VCG-based threshold payment mechanism [29] to elicit worker truthfulness.

- The Steiner Tree-based mechanism (*Steiner-Tree*). This mechanism first forms a professional team based on worker marginal contribution-per-cost values without considering the underlying workers' network. It then exploits a Steiner tree-based algorithm to form a collaborative team [21]. Finally, the binary search-based threshold payment technique [48] is used to elicit workers' truthfulness.

We compare these mechanisms on the following three performance metrics: 1) the utility of the task requester $u_T = V_T - \Sigma_{a_i \in G} p_i$, where p_i is the payment paid to each worker and G is the winning team; 2) the utility of each worker $u_i = p_i - c_i$ and 3) the social welfare of the crowdsourcing system $W_T = V_T - \Sigma_{a_i \in G} c_i$. All results are recorded by averaging over 100 instances, over which the standard deviation ranges from 15% to 40%. For each instance, a task T is crowdsourced with a profit V_T randomly drawn from a range of 250, 500, 750, and 1,000.

6.2 Experimental Results

6.2.1 Social Welfare Evaluation

Fig. 4 shows the social welfare of the OPT, OPT-Tree, Greedy and Steiner-Tree for different networks. From Fig. 4, we have the following observations.

1) In all of the networks, when the task size increases, the social welfare of these mechanisms deceases as well. This observation is intuitive because in the case that more skills to be satisfied, more workers must be hired, and less social welfare will result. The OPT-Tree performs similarly to the OPT. This indicates that although the OPT-Tree forms an optimal team in the extracted tree by disregarding some connection information, it effectively maximizes social welfare while reducing computation time. The Greedy mechanism (i.e., Algorithm 3) performs worse than the Steiner-Tree system when the task size increases from 1 to 8. In the Greedy mechanism, the formation of a professional team is highly dependent on team members' accessibility to skillful neighbor workers. When the task size is small, the Greedy mechanism might fail to form a professional team because the skills required by the task do not match the skills owned by current accessible neighbor workers. The failure of forming a professional team lead the failure of task completion and zero social welfare achieved. However, as task size increases (i.e., $k \geq 8$), social welfare of the Greedy performs much better than that of Steiner-Tree. This occurs because *i*) in the Greedy mechanism, as a requester requires more skills, more complementary workers can be found. This advantage increases the probability of building a professional team, thus increasing social welfare. *ii*) The Steiner-Tree mechanism involves searching for a large number of additional workers for team connecting, incurring higher working costs and thereby reducing social welfare.

2) The Greedy performs worse when applied to an SMW network than when applied to Random and SF networks. This can be explained from a network property perspective. Table 1 presents three typical network properties: network diameter (*Dia*), average shortest path length (*Aspl*) and clustering coefficient (*Clu*) ([47] provides a detailed illustration of these properties). From Table 1, we can conclude that the Random (or SF) network presents smaller *Dia* and *Aspl* values than those of the SMW. These smaller *Dia* and *Aspl* values indicate that the workers are more socially close. Therefore, owing to its greedy nature, the Greedy system is more likely to form a professional team in a Random (or SF) than in an SMW.

3) Although OPT can always form the optimal team with the maximum social welfare, its exponential complexity limits its use in tiny-scale applications (i.e., $k \leq 5$). Here, a mechanism is regarded as computationally infeasible when its running time exceeds 500 seconds. However, the OPT-Tree can extend its application scope to small-scale applications (i.e., the typical application in the practice shown in Section 6.1 available in the online supplemental material). Moreover, the Greedy and Steiner-Tree systems work well in any scale application, exhibiting their advantages in terms of scalability.

6.2.2 Requester Utility Evaluation

Fig. 5 shows the requester utility generated from the OPT, OPT-Tree, Greedy and Steiner-Tree systems, from which we have the following observations. 1) In all of the

TABLE 1
The Properties of Networks

Property / Network	Diameter L	Average Shortest Path length $ASPL$	Clustering Coefficient C
Small-World	6.5	4.0	0.28
Scale-Free	5.0	3.2	0.03
Random	6.0	3.5	0.01

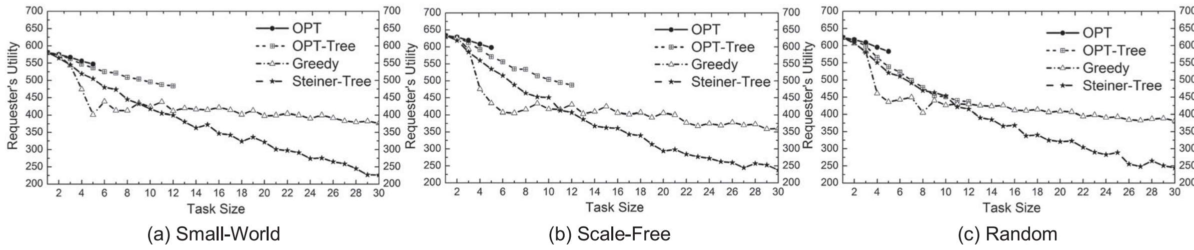


Fig. 5. The requester's utility returned by OPT, OPT-Tree, Greedy, and Steiner-Tree in different networks.

networks, the requester's utility decreases with task size. This is because as more skills need to be satisfied, more workers must be hired and less utility will result. 2) The OPT generates the most utility for the requester, which is followed by the OPT-Tree, Greedy and Steiner-Tree. This observation is in accordance with the social welfare results shown in Fig. 4. 3) Although the differences are slight, the requester utility gap between the OPT-tree and Greedy is larger than the social welfare gap between these two mechanisms. This might be because in the Greedy system, a requester might hire a much more expensive worker team (as shown in Fig. 6 on worker utility) than in the OPT-Tree system, and in turn a requester in the Greedy system must pay more than a requester of the OPT-Tree system.

6.2.3 *Worker Utility Evaluation*

Fig. 6 shows workers' utility distribution collected from the OPT-Tree, Greedy in SM network (the results are similar for Random and Scale-Free networks and due to space limitations, we omit these network results), from which we observe that: 1) each worker achieves a non-negative utility, satisfying individual rationality requirements. 2) A worker in the Greedy system can achieve a much higher degree of utility (i.e., the maximal utility is \$270) than a worker in the OPT-Tree system (i.e., the maximal utility is \$85). This occurs because the Greedy system forms teams locally and might choose an expensive worker to join a team. Thus, the Greedy system pays much more while the OPT-Tree system forms teams globally, thus building cheaper teams without too many expensive workers.

6.2.4 *The Advantage of Truthful Behavior*

Fig. 7 shows the utility difference between of truthful and untruthful behaviors under large-scale (Fig. 7a) and small-scale (Fig. 7b) social team formation mechanisms. We first compute the utility of each winner worker that reports his true skill and cost values (i.e., Truthful worker), next we compute each winner worker's utility of manipulating his skill or cost value (i.e., Untruthful worker). From Fig. 7, we can find that the worker who behaves untruthfully cannot achieve a larger utility than that achieved by behaving truthfully. For example, in Fig. 7a, workers $\{a_1, a_2, a_3, a_4, a_5\}$ are the selected team members when they behave truthfully. Truthful workers a_1 and a_2 achieve the positive payment, while achieving the negative payment by behaving untruthfully. This is because a_1 and a_2 over-reporting their skills, which are detected and penalized by the requester. Compared to the larger utility achieved of truthful reporting skills, worker a_3 achieves a smaller utility by under-reporting his skills. On the other hand, when worker a_5 over-reports his cost, he will be not selected as a team member, resulting in zero utility. In summary, we can determine that each worker can always achieve the maximal utility by behaving truthfully.

7 Conclusions and Future Work

This paper addresses social team crowdsourcing problems, where crowdsourcing platforms are designed to hire teams of socially tight-knit workers for the requester's complex task completion. We study this NP-hard problem from a mechanism design perspective with the aim of guaranteeing that each worker's utility is optimized by behaving truthfully. To meet the truthfulness goal over tolerable running periods, we propose two efficient OPT-Tree and Greedy mechanisms for small- (i.e., task size $k \leq 12$) and large-scale (i.e., $k > 12$) applications, respectively. The experimental results determine that the OPT-Tree mechanism performs similarly to the optimal benchmark exponential VCG type truthful mechanism in terms of social welfare while largely reducing computation time. Moreover, the Greedy mechanism not only achieves better social welfare outcomes than existing large-scale-oriented social team crowdsourcing heuristics but also satisfies the truthfulness property. These tools can help crowdsourcing platforms determine which mechanisms are suitable for which type applications.

There are two interesting directions for future research. 1) In some crowdsourcing applications, each worker's cost depends on which and how many skills he provides. Then each worker can manipulate the crowdsourcing market by under-reporting his skills for marginal contribution-per-cost value, which is immune to the proposed marginal contribution-per-cost -based greedy mechanism. Thus a novel truthful mechanism, where the payment is independent on the declared types of the workers is essential to address this cost-skill interdependent setting. One feasible method is first

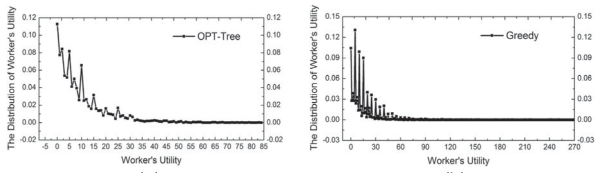


Fig. 6. The worker's utility distribution returned by (a) OPT-Tree and (b) Greedy.

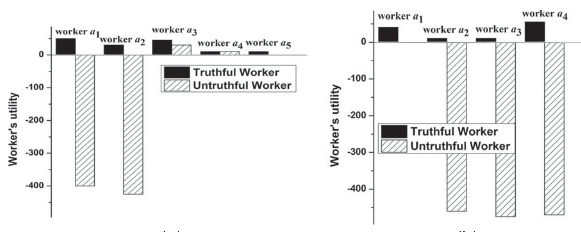


Fig. 7. The advantage of truthfulness on maximizing utility under (a) large-scale and (b) small-scale social team formation problems.

to partition the large size task into l small size sub-tasks and cluster the social networks into l disjoint sub-graphs, and utilize the proposed optimal algorithm (i.e., Algorithm 2) to build teams from each sub-graph to satisfy each subtask. 2) For the collusion scenario where the workers can form a group to manipulate the social team crowdsourcing mechanisms (e.g., two workers collude to report the non-existing social connections), the idea of group-strategyproof in previous study [49] can be introduced to prevent collusion manipulation.

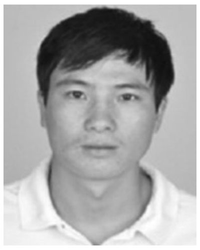
ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (61472079, 61170164, 61807008 and 61806053, 61472089), the Natural Science Foundation of Jiangsu Province of China (BK20171363), the Joint Fund of the National Natural Science Foundation of China and Guangdong Province (U1501254), the Science and Technology Planning Project of Guangdong Province (2015B010131015 and 2015B010108006), the Natural Science Foundation of Guangdong Province (2014A030308008), and Guangdong Regular University International and Hong Kong, Macao and Taiwan Cooperative Innovation Platform and International Cooperation Major Projects (2015KGJHZ023). An extended abstract of this paper was presented at the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-16) [50].

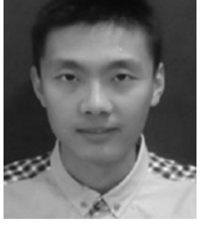
REFERENCES

[1] D. C. Brabham, "Crowdsourcing as a model for problem solving: An introduction and cases," *Convergence: Int. J. Res. Into New Media Technol.*, vol. 14, no. 1, pp. 75–90, 2008.
[2] C. C. Yang, J. Yen, and J. Liu, "Social intelligence and technology," *IEEE Intell. Syst.*, vol. 29, no. 2, pp. 5–8, Mar./Apr. 2014.
[3] Q. Zhang, Y. Wen, X. Tian, X. Gan, and X. Wang, "Incentivize crowd labeling under budget constraint," in *Proc. 34th IEEE Conf. Comput. Commun.*, 2015, pp. 2812–2820.
[4] S. Goto, D. Lin, and T. Ishida, "Crowdsourcing for evaluating machine translation quality," in *Proc. 9th Int. Conf. Language Resources Eval.*, 2014, pp. 3456–3463.
[5] G. Goel, A. Nikzard, and A. Singla, "Mechanism design for crowdsourcing markets with heterogeneous tasks," in *Proc. 2nd AAAI Conf. Hum. Comput. Crowdsourcing*, 2014, pp. 77–86.
[6] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 173–184.
[7] A. Singla, E. Horvitz, P. Kohli, and A. Krause, "Learning to hire teams," in *Proc. 3rd AAAI Conf. Hum. Comput. Crowdsourcing*, 2015, pp. 34–35.
[8] B. Golshan, T. Lappas, and E. Terzi, "Profit-maximizing cluster hires," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 1196–1205.
[9] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 647–661, Apr. 2016.
[10] Y. Chen, B. Li, and Q. Zhang, "Incentivizing crowdsourcing systems with network effects," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1944–1952.
[11] K. Han, H. Huang, and J. Luo, "Posted pricing for robust crowdsensing," in *Proc. 17th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2016, pp. 261–270.
[12] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, "iCrowd: Near-optimal task allocation for piggyback crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 2010–2022, Aug. 2016.
[13] Y. Han, T. Luo, D. Li, and H. Wu, "Competition-based participant recruitment for delay-sensitive crowdsourcing applications in D2D networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 12, pp. 2987–2999, Dec. 2016.
[14] T. Voice, S. D. Ramchurn, and N. R. Jennings, "On coalition formation with sparse synergies," in *Proc. 11th Int. Conf. Auton. Agents Multiagent Syst.*, 2012, pp. 223–230.
[15] F. Bistaffa, A. Farinelli, J. Cerquides, J. Rodríguez-Aguilar, and S. D. Ramchurn, "Anytime coalition structure generation on synergy graphs," in *Proc. 13th Int. Conf. Auton. Agents Multiagent Syst.*, 2014, pp. 13–20.
[16] T. Wolf, A. Schröter, D. Damian, L. D. Panjer, and T. H. D. Nguyen, "Mining task-based social networks to explore collaboration in software teams,". *IEEE Softw.*, vol. 26, no. 1, pp. 58–66, Jan./Feb. 2009.
[17] J. Chamberlain, "Groupsourcing: Distributed problem solving using social networks," in *Proc. 2nd AAAI Conf. Hum. Comput. Crowdsourcing*, 2014, pp. 22–29.
[18] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *IEEE Trans. Comput.*, vol. 63, no. 7, pp. 1682–1695, Jul. 2014.
[19] I. Lykourentzou, A. Antoniou, Y. Naudet, and S. P. Dow, "Personality matters: Balancing for personality types leads to better outcomes for crowd teams," in *Proc. 19th ACM Conf. Comput.-Supported Cooperative Work Soc. Comput.*, 2016, pp. 259–272.
[20] M. L. Gray, S. Suri, S. S. Ali, and D. Kulkarni, "The crowd is a collaborative network," in *Proc. 19th ACM Conf. Comput.-Supported Cooperative Work Social Comput.*, 2016, pp. 134–147.
[21] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 467–475.
[22] A. Anagnostopoulos, L. Becchetti, and C. Castillo, "Online team formation in social networks," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 839–847.
[23] S. Datta, A. Majumder, and K. V. M. Naidu, "Capacitated team formation problem on social networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1005–1013.
[24] M. Kargar and A. An, "Discovering top-k teams of experts with/without a leader in social networks," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 985–994.
[25] M. Kargar, M. Zihayat, and A. An, "Finding affordable and collaborative teams from a network of experts," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 587–595.
[26] S. Tang, "Profit-driven team grouping in social networks," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, USA, February 4–9, 2017, pp. 45–51.
[27] S. Rangapuram, T. Buhler, and M. Hein, "Towards realistic team formation in social networks based on densest subgraphs," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1077–1087.
[28] A. Ghosh, "Game theory and incentives in human computation systems," *Handbook of Human Computation.* New York, NY, USA: Springer, 2013, pp. 725–742.
[29] N. Nisan, "Algorithmic Mechanism Design," in *Proc. 31st Ann. ACM Symp. Theory Comput.*, 1999, pp. 129–140.
[30] S. Dobzinski and N. Nisan, "Limitations of VCG-based mechanisms," in *Proc. 39th Annu. ACM Symp. Theory Comput.*, 2007, pp. 338–344.
[31] Q. Liu, T. Luo, R. Tang, and S. Bressan, "An efficient and truthful pricing mechanism for team formation in crowdsourcing markets," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 567–572.
[32] Z. Pan, H. Yu, C. Miao, and C. Leung, "Efficient collaborative crowdsourcing," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 4248–4249.
[33] H. Jiang and S. Matsubara, "Efficient task decomposition in crowdsourcing," in *Proc. 17th Int. Conf. Principles Practice Multi-Agent Syst.*, 2014, pp. 65–73.
[34] L. Tran-Thanh, T. D. Huynh, A. Rosenfeld, S. Ramchurn, and N. R. Jennings, "BudgetFix: Budget limited crowdsourcing for interdependent task allocation with quality guarantees," in *Proc. 13th Int. Conf. Auton. Agents Multiagent Syst.*, 2014, pp. 477–484.
[35] M. Wright and Y. Vorobeychik, "Mechanism design for team formation," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1050–1056.
[36] W. Wang, J. Jiang, B. An, Y. Jiang, and B. Chen, "Towards efficient team formation for crowdsourcing in non-cooperative social networks," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4208–4222, Dec. 2017.
[37] Y. Jiang and J. C. Jiang, "Understanding social networks from a multiagent perspective," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2743–2759, Oct. 2014.

[38] K. Zhu and E. Hossain, "Virtualization of 5G cellular networks as a hierarchical combinatorial auction," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2640–2654, Oct. 2016.
[39] M. Daltayanni, L. de Alfaro, and P. Papadimitriou, "WorkerRank: Using employer implicit judgements to infer worker reputation," in *Proc. 8th ACM Int. Conf. Web Search and Data Mining*, 2015, pp. 263–272.
[40] Z. Wang, Y. Yin, and B. An, "Computing optimal monitoring strategy for detecting terrorist plots," in *Proc. 30th AAAI Conf. Artif. Intell*, 2016, pp. 637–643.
[41] S. Jagabathula, L. Subramanian, and A. Venkataraman, "Reputation-based worker filtering in crowdsourcing," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2492–2500.
[42] T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila, "Finding effectors in social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 1059–1068.
[43] C. Dangalchev, "Residual closeness in networks," *Physica A: Statist. Mech. Appl.*, vol. 365, no. 2, pp. 556–564, 2006.
[44] D. S. Johnson, J. K. Lenstra, and A. H. G. Rinnooy Kan, "The complexity of the network design problem," *Netw.*, vol. 8, no. 4, pp. 279–285, 1978.
[45] Y. Singer, "Budget feasible mechanisms," in *Proc. 51th Annu. IEEE Symp. Found. Comput. Sci.*, 2010, pp. 765–774.
[46] A. Archer and É. Tardos, "Truthful mechanisms for one-parameter agents," in *Proc. 42nd IEEE Symp. Found. Comput. Sci.*, 2001, pp. 482–491.
[47] F. Bergenti, E. Franchi, and A. Poggi, "Selected models for agent-based simulation of social networks," in *Proc. 3rd Int. Conf. Social Networks and Multiagent Syst. Symp.*, 2011, pp. 27–32.
[48] L. Mashayekhy, M. M. Nejad, and D. Grosu, "A PTAS mechanism for provisioning and allocation of heterogeneous cloud resources, "*IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 9, pp. 2386–2399, Sep. 2015.
[49] F. Brandt, "Group-strategyproof irresolute social choice functions," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 79–84.
[50] W. Wang, Z. He, P. Shi, W. Wu, and Y. Jiang, "Truthful team formation for crowdsourcing in social networks," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2016, pp. 1327–1328.


Wanyuan Wang received the PhD degree in computer science from Southeast University, China, in 2016. He is an assistant professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. He has published several articles in refereed journals and conference proceedings, such as the *IEEE Transactions*, AAAI, and AAMAS. He won the Best Student Paper Award from ICTAI14. His main research interests include artificial intelligence, multiagent systems, game theory, and crowdsourcing.

Zhanpeng He received the BE degree in computer science and engineering from Southeast University, Nanjing, China, in 2015. He is currently working toward the ME degree in the School of Computer Science and Engineering, Southeast University. His main research interests include social networks, multiagent systems, and crowdsourcing.


Peng Shi received the BE degree in computer science and engineering from Southeast University, Nanjing, China, in 2015. He is currently working toward the ME degree in the School of Computer Science and Engineering, Southeast University. His main research interests include social networks, multiagent systems, and crowdsourcing.

Weiwei Wu received the PhD degree in computer science from both the City University of Hong Kong and the University of Science and Technology, China, in 2011. He is currently an associate professor at Southeast University, China. His main research interests include optimizations for networks, game theory, and mechanism design. He has published several articles in refereed journals and conference proceedings such as *IEEE Transactions on Mobile Computing*, the *IEEE Journal on Selected Areas in Communications*, *Theoretical Computer Science*, and INFOCOM.

Yichuan Jiang (SM'13) received the PhD degree in computer science from Fudan University, Shanghai, China, in 2005. He is a professor with the School of Computer Science and Engineering, Southeast University, Nanjing, China. His main research interests include multiagent systems, social computing, and social networks. He has published more than 90 scientific articles in refereed journals and conference proceedings, such as the *IEEE Transactions on Parallel and Distributed Systems*, the *IEEE Journal on Selected Areas in Communications*, the *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, the *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, the *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, the *IEEE Transactions on Cybernetics*, the *ACM Transactions on Autonomous and Adaptive Systems*, the *Journal of Autonomous Agents and Multi-Agent Systems*, the *Journal of Parallel and Distributed Computing*, IJCAI, AAAI, and AAMAS. He won the Best Paper Award from PRIMA06 and Best Student Paper Awards twice from ICTAI13 and ICTAI14. He is a senior member of the IEEE.

Bo An received the PhD degree in computer science from the University of Massachusetts, Amherst. He is an associate professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His current research interests include artificial intelligence, multiagent systems, game theory, and optimization. He has published more than 70 referred papers at AAMAS, IJCAI, AAAI, ICAPS, KDD, JAAMAS, AIJ and ACM/IEEE Transactions. He was the recipient of the 2010 IFAAMAS Victor Lesser Distinguished Dissertation Award, an Operational Excellence Award from the Commander, First Coast Guard District of the United States, the Best Innovative Application Paper Award at AAMAS'12, the 2012 INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice, and the Innovative Application Award at IAAI'16. He was invited to give the Early Career Spotlight talk at IJCAI'17. He led the team HogRider which won the 2017 Microsoft Collaborative AI Challenge. He was named to *IEEE Intelligent Systems'* "AI's 10 to Watch" list for 2018. He is a member of the editorial board of *JAIR* and the associate editor of *JAAMAS*. He was elected to the board of directors of IFAAMAS.

Zhifeng Hao received the PhD degree in applied mathematics from Nanjing University, Nanjing, China, in 1995. He is now a professor with the School of Mathematics and Big Data, Foshan University, China. His research interests include artificial intelligence and big data. He has published several scientific articles in refereed journals and conference proceedings, such as the *IEEE Transactions on Neural Networks*, and the *IEEE Transactions on Knowledge and Data Engineering*.

Bing Chen received his BS and MS degrees in computer engineering from the Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China, in 1992 and 1995, respectively. He received the PhD degree at the College of Information Science and Technology from NUAA, in 2008. He has worked for NUAA since 1998. He is currently a professor at the School of Computer Science and Technology, NUAA. His main research interests include computer networks, wireless communications, and mobile computing