Truthful Team Formation for Crowdsourcing in Social Networks

(Extended Abstract)

Wanyuan Wang, Zhanpeng He, Peng Shi, Weiwei Wu and Yichuan Jiang*
School of Computer Science and Engineering, Southeast University, Nanjing, China,
{wywang.seu, zphe.seu}@gmail.com, {pengshi, weiweiwu, yjiang}@seu.edu.cn

ABSTRACT
We study complex task crowdsourcing by team formation in social networks (SNs), where the requester wishes to hire a group of socially close workers that can work together as a team. The workers are selfish that can manipulate the crowdsourcing system by providing unreal private information, which will discourage other workers from participation and is unprofitable for the requester. This paper develops two efficient truthful mechanisms for the small- and large-scale social team crowdsourcing applications, to guarantee each worker’s profit is optimized by behaving truthfully.

Keywords
Mechanism design, team crowdsourcing, social networks

1. INTRODUCTION
This paper studies team/group crowdsourcing, where the requester wishes to hire a group of workers that can work together as a team for the complex task completion [1]. Social networks (SNs) provide good opportunities to address the team crowdsourcing problem, where the social connections among workers are often good indicator of effective collaboration [3]. This social team crowdsourcing can be implemented by a reverse auction model, where the requester, first announces his task’s skill requirements and each worker then bids to sell his skill services associated with his working cost and social relationships. Based on the workers’ bids, the crowdsourcing system aims to form the optimal feasible team members and 2) Collaborative: the subgraph induced by the team members S must be connected. The payment function Pay = {p1, p2, ..., pk} determines the reward paid to each winner. The utility ui of each winner ai S then is ui = pi - ci, equals pi - ci under truthful bidding. The requester’s utility then is ut = vT - \sum ai\in S pi. The social welfare WT of the crowdsourcing system is the sum of the requester’s utility and the agents’ aggregate utility, i.e., WT = vT - \sum ai\in S pi + \sum ci\in S (pi - ci) = vT - \sum ci\in S. This paper considers maximizing social welfare.

Each agent is strategic for maximizing its own utility, such an objective of maximizing social welfare alone will encourage the strategic agents to lie about its working cost information. This paper designs truthful mechanisms to elicit the agents to report their working cost truthfully.

* Corresponding author

for the small-scale applications, which works by first transforming a social network to a binary tree network, and then a dynamic programming-based optimal mechanism is developed in the transformed tree, and a novel polynomial time mechanism for the large-scale applications, which works by selecting the team members greedily based on their social structure as well as on their skills and working cost.

2. PROBLEM DESCRIPTION
There is a task T=<vT, OT>, where vT is the profit of T and OT = {s1, s2, ..., sk} is T’s skill requirements. We model the team crowdsourcing paradigm as a reverse auction framework, where the requester announces its task skill requirements OT and then each worker ai submits its bid Bi=(ri, ci, Ni), consisting of the skills Ri = {s1, s2, ..., sk} it can provide, the working cost ci, representing the minimum reward ai wishes to be paid and its neighbors/collaborative partners Ni. After receiving these workers’ bids, the crowdsourcing platform can determine the workers’ social network SN=<A, E>, where A={a1, a2, ..., ak} denotes the collection of agents/workers and \forall (ai, aj) \in E represents the existence of a connection between ai and aj.

A mechanism M = (X, Pay) consists of a team formation function X and a payment function Pay. The team formation function X = {x1, x2, ..., xn} determines whether an agent ai is selected as a winner (xi=1) or not (xi=0). Let S = {ai|xi = 1} be the winner team. To complete task T successfully, the formed team S must satisfy 1) Professional: each skill sj \in OT must be satisfied by at least one team member and 2) Collaborative: the subgraph induced by the team members S must be connected. The payment function Pay = {p1, p2, ..., pk} determines the reward paid to each winner. The utility ui of each winner ai S then is ui = pi - ci, equals pi - ci under truthful bidding. The requester’s utility then is ut = vT - \sum ai\in S pi. The social welfare WT of the crowdsourcing system is the sum of the requester’s utility and the agents’ aggregate utility, i.e., WT = vT - \sum ai\in S pi + \sum ci\in S (pi - ci) = vT - \sum ci\in S. This paper considers maximizing social welfare.

Each agent is strategic for maximizing its own utility, such an objective of maximizing social welfare alone will encourage the strategic agents to lie about its working cost information. This paper designs truthful mechanisms to elicit the agents to report their working cost truthfully.

Regarding why workers cannot manipulate their skills and social relationships: Over-report the skills that a worker cannot provide might make task failure, which can be detected during task execution. Each social relationship depends on two workers and once a worker reports the nonexistent relationships, it can also be detected.
3. **TRUTHFUL SOCIAL TEAM CROWDSCORING MECHANISMS**

Towards Small-Scale Applications. The small-scale-oriented mechanism consists of the following three phases.

- **Tree Network Extraction.** We first extract a tree network Γ from the original network SN =< A, E > such that Γ preserves as much social connection information as that in SN. The proposed tree extraction algorithm is $\frac{1}{2}$-approximation on maximizing network closeness, where network closeness $CL(SN)$ is defined as how socially close these agents A connect with each other in network SN, i.e., $CL(SN)=\sum_{a_i \in A} \sum_{a_j \neq a_i, a_i a_j \in SN} \frac{1}{D}$ and $D$ is the diameter of the network SN.

- **Binary Tree Transformation.** We then transform the tree Γ to the binary tree $\Gamma_3$. We start from the root agent $a_0$ of Γ. Suppose that $a_0$ has I children $\{a_1, a_2, \ldots, a_I\}$ and we replace $a_0$ and $a_0$’s children with a binary tree of depth $[\log I]+1$, where the root agent still is $a_0$, and the leaf agents are $\{a_1, a_2, \ldots, a_I\}$. The newly-added auxiliary agents between $a_0$ and $\{a_1, a_2, \ldots, a_I\}$ in this binary tree neither have any skill nor require any working cost. Moreover, once their parent agent is selected as a winner, they will also be selected as winners. This transformation repeats recursively for all of the other non-leaf agents down $a_0$ and finally the binary tree $\Gamma_3$ is constructed.

- **Optimal Truthful Mechanism in Binary Tree.** For each agent $a_i$ in $\Gamma_3$, let $S(a_i, 1, U)$ be the optimal team formed to satisfy the skills $U \subseteq O_T$ in the subtree $\Gamma_{a_i}^3$, where $a_i$ is selected as a winner. Let $W(a_i, 1, U)$ be the welfare of $S(a_i, 1, U)$. Similarly, let $W(a_i, 0, U)$ be the welfare of the optimal team $S(a_i, 0, U)$ formed in $\Gamma_{a_i}^3$ without selecting $a_i$. Let $l(a_i)$ and $r(a_i)$ denote $a_i$’s left and right child. The following dynamic programming is then implemented recursively for each agent $a_i$.

\[
W(a_i, 1, U) = \max \left\{ \begin{array}{ll}
W(r(a_i), 1, U\setminus R_i) - \tilde{c}_i; \\
W(l(a_i), 1, U\setminus R_i) - \tilde{c}_i; \\
\max_{U' \subseteq U \setminus R_i} W(r(a_i), 1, U') + \\
W(l(a_i), 1, U\setminus U') - \tilde{c}_i - V_T.
\end{array} \right. 
\]

and

\[
W(a_i, 0, U) = \max \{ W(r(a_i), 1, U), W(r(a_i), 0, U), W(l(a_i), 1, U), W(l(a_i), 0, U) \}. 
\]

The initial conditions of this dynamic programming approach are: $W(\emptyset, 0, \emptyset) = V_T$ and $\forall a_i \in A, W(a_i, 1, \emptyset) = V_T - \tilde{c}_i$ and $\forall U \neq \emptyset, W(\emptyset, 0, U) = 0$. Finally, the optimal team formed in $\Gamma_3$ is returned from function $\max \{ W(a_i, 0, O_T), W(a_i, 1, O_T) \}$. Denoted by the optimal team and its welfare as $\beta_3$ and $W_\beta$, then the VCG-based threshold payment $p_i$ for each winner agent $a_i$ in $\beta_3$ is defined as:

\[
p_i = (W_\beta + \tilde{c}_i) - \max \{ W_{\beta\setminus r(a_i)}, W_{\beta\setminus l(a_i)}, W_{\beta\setminus r(a_i)} \}. 
\]

The value $W_{\beta\setminus r(a_i)} = V_T - \sum_{a_j \in S_{\beta\setminus r(a_i)}} \tilde{c}_j$ is the welfare of $S_{\beta\setminus r(a_i)}$, where $S_{\beta\setminus r(a_i)}$ is the optimal team returned from $a_i$’s right subtree $\Gamma_{r(a_i)}^3$. The other terms have the similar meanings.

Towards Large-Scale Applications. We present a polynomial time truthful mechanism for the large-scale applications, which includes the monotonously greedy team formation algorithm and the threshold payment algorithm.

- **Greedy Team Formation.** We first locate the agent $a_i$ that has the largest marginal contribution-per-cost value as team root, where agent $a_i$’s marginal contribution-per-cost value with respect to the skill set $U \subseteq O_T$ is $\epsilon(a_i, U) = |U \cap R_i|/\tilde{c}_i$. Then, we select the team’s best neighbor agent $a^\ast = \arg\max_{a_j \in T} \epsilon(a_j, O_T)$ with the largest marginal contribution-per-cost to join this team, where $I = \bigcup_{a_j \in Q} \{ a_j \in N_i : \epsilon(a_j, O_T) > 0 \}$. We proceed to select the desirable team neighbors round by round until the team is professional.

- **Threshold Payment.** We adapt the threshold payment technique of Singer [5] to achieve the threshold payment $p_i$ for each winner $a_i$ such that $p_i$ is the maximal value $a_i$ can bid and still be selected by the greedy team formation.

4. **EXPERIMENTS**

We collect 928 workers data from a popular crowdsourcing website Guru. These workers are interconnected by the scale-free network structure. We also collect the tasks on Guru and observe that most of the tasks require less than 30 kinds of skills. For each task $T$, we assume its profit $V_T$ is drawn from the range [300, 400] randomly. We compare the proposed mechanisms, i.e., optimal mechanism in a tree network OPT-Tree and the greedy mechanism Greedy with the benchmark optimal mechanism OPT on social welfare.

The right figure shows the social welfare of the these mechanisms. For the small-scale applications where task size $k \leq 5$, OPT-Tree performs very close to OPT. Greedy performs worse when task size grows up from 1 to 3. However, as task size ranges from 3 to 8, the social welfare of Greedy grows up. Interestingly, when task size becomes larger further, i.e., $\geq 8$, the social welfare of Greedy decreases again. Although OPT can always form the optimal team with the maximum social welfare, its exponential time complexity on task size limits itself to be applicable to the small-scale applications (i.e., $k \leq 5$) only, while Greedy scales well to various scale applications.

**Acknowledgments**

This work was supported by the National Natural Science Foundation of China (No. 61170164, No. 61472079 and No. 71201077), the Funds for Distinguished Young Scholars of the Natural Science Foundation of Jiangsu Province (No. BK2012020), and the Program for Distinguished Talents of Six Domains in Jiangsu Province (No. 2011-DZ023).

**REFERENCES**


