Truthful Mechanism for Task Assignment in Crowdsourcing with Social Network

Haiyan Qin\(^1\), Yonglong Zhang\(^2\), and Bin Li\(^3\)

\(^1\)Guangling College, Yangzhou University, China
\(^2\)College of Information Engineering Yangzhou University, China
\(^3\)State Key Laboratory for Novel Software Technology Nanjing University, China

E-mail: 13092006508@163.com

Abstract. Macrotasks like document writing, product design, or web development, require expert skills and the collective contributions of the crowds. In fact, there are few studies give close attention to the workers with social network in crowdsourcing. Social networks provide an available platform for cooperation among social workers. In this paper, we discuss the complex task assignment in crowdsourcing where the workers are in a social network. A Truthful Mechanism for the task assignment in Crowdsourcing with Social Network (TMC-SN) is proposed to solve the challenges in this scenario. The problem is modeled as an auction, where the requesters are buyers, workers are sellers, and the crowdsourcing platform serves as an auctioneer. In order to find out the most appropriate team, TMC-SN measures Worker Fitness of the worker for the team from both Marginal Contribution and Team Cohesion. We demonstrate that TMC-SN has fine economic properties such as truthfulness, individual rationality, and budget balance. The evaluation results show that TMC-SN is advantageous in terms of Social Welfare and favorable for the workers.

1. Introduction

Crowdsourcing, coined by Jeff Howe in 2006, is defined as the method of outsourcing a specific task to a set of uncertain people instead of assigning it to designated employees. From then on, crowdsourcing is prevalent in many fields. In fact, the technique of crowdsourcing can be traced back to 1884 when Oxford English Dictionary depended on a crowd to catalog words[1]. Amazon Mechanical Turk is a functional crowdsourcing platform where employers release tasks such as translation and labelling etc and Internet workers like to complete tasks with small rewards. Despite micro payments, part-time or full-time crowds are willing to take different types of tasks which not only decreases overall expense of the requester but also helps to generate high-quality work.

It is worth noting that there are few studies that give close attention to the workers with social network in crowdsourcing. Conversely, the scientific collaboration network has been considered as a representative social network in previous studies, in which teamwork is an important issue in scientific and technological activities[2]. Obviously, some research projects are too complex to be performed by a single scientist. Therefore, scientific projects are necessarily conducted with the collaboration of many scientists who tend to form or have built up communities in accordance with their research topics and interests. Social networks provide an available platform for cooperation among social workers. In crowdsourcing, TopCoder is a representative crowdsourcing marketplace for software development which needs the cooperation of a set of workers to generate the project. Actually, once the social network of the workers is introduced into crowdsourcing, it is triple-win for the workers, task requesters and the platform. The workers would rather collaborate with those who are familiar with than those who need to be further run-in. Then, the task would be completed more outstandingly and efficiently. Task requesters are certainly glad to receive the finished task with higher-quality. Furthermore, the platform may hire less workers and then cut costs.
In this paper, we discuss the complex task assignment in crowdsourcing where the workers are connected. The problem is modeled as an auction, where the requesters are buyers, workers are sellers, and the crowdsourcing platform serves as an auctioneer. After the requester posts the task and the connected workers submit skills, the platform will make the assignment. In this scenario, there are several unsolved challenges. First of all, it has been proved that the complexity of task assignment in social networks remains NP-hard\cite{3}. Thus, presenting an efficient mechanism is the most important issue. After that, the workers in social networks are connected with complex relationship. In that way, the mechanism should figure out the relations of the connected workers and then find out a suitable team. Last but not least, the self-serving workers may attempt to misreport their ask to gain a higher utility. Therefore, the mechanism has to be truthful.

A Truthful Mechanism for the task assignment in Crowdsourcing with Social Network (TMC-SN) is proposed to solve the challenges mentioned above. In TMC-SN, an efficient assignment within polynomial time complexity is proposed. In order to find out the most appropriate team, TMC-SN measures Worker Fitness of the worker for the team from both Marginal Contribution and Team Cohesion. Conventional solution ignored the self-centred players who may misreport their value in order to gain higher utility. However, auction is introduced into TMC-SN as a valid method to forbid the players from lying. TMC-SN satisfies all the properties of auction, truthfulness, individual rationality, and budget balance.

The remainder of this paper is structured as follows. The detailed system model and some definitions are described in Section 2. We present the design of TMC-SN in detail in Section 3 and then analyse related properties, specifically the truthfulness. Our simulation results are presented in Section 4, followed by conclusions in Section 5.

2. Problem Formulation

2.1. System Model

Social Networks: A social network, \( SN = (A,E,D) \) is an undirected graph, where \( A = \{a_1,a_2,\ldots,a_m\} \) is the set of workers. \( \forall (a_i,a_j) \in E \) indicates that there is a direct social interaction between \( a_i \) and \( a_j \). Each pair of workers, \( (a_i,a_j) \in E \) is associated with a cost \( d(a_i,a_j) \in D \) incurred by the necessary communication and \( d(a_i,a_j) = d(a_j,a_i) \). In other words, \( d(a_i,a_j) \in D \) is the value of edge for \( (a_i,a_j) \in E \).

Requester: The requester posts her task \( R \) with bid \( b \) to the platform. \( b \) is the requester’s claimed value if the task is completed and \( \tilde{b} \) is the truthful value. We assume that there are \( \tau \) types skills \( S = \{s_1,s_2,\ldots,s_{\tau}\} \) the requesters need. \( S_R = \{s_{1R},s_{2R},\ldots,s_{\tau R}\} \) is a \( \tau \)-dimension vector, where \( s_{kR} = \{0,1\} \) means that \( k \)-th skill is required or not in \( R \). We assume that there is an uncovered skill vector \( S_R' \) as the members are added into the team and \( S_R' \) is the same as \( S_R \) at the beginning of process.

Worker: Each worker \( a_i \in A \) submits a tuple \( C_i = \{< s_{kR}^i,c_{kR}^i >|k=1,2,\ldots,\tau\} \) to the platform. Among \( C_i \), \( s_{kR}^i = \{0,1\} \) denotes whether \( a_i \) holds the \( k \)-th skill or not and \( c_{kR}^i \) is the reward that \( a_i \) expects to be paid by contributing \( s_{kR}^i \). \( c_{kR}^i \) is not necessary equal to \( c_k^i \), which is the true cost of performing the \( s_k \). \( C = \{C_1,C_2,\ldots,C_m\} \) is the bid vector of all the workers. \( N(a_i) \) is the set of agents who have the direct connection with \( a_i \).

The worker may contribute one or several skills to the task. Therefore, we import overall cost \( oC_i \) to deal with the gathering skills. \( H = \{h_1,h_2,\ldots,h_{\tau}\} \) represents the weight vector of \( S \), where \( h_k \) means the workload of the skill \( s_k \). If \( a_i \) has the skills that is required in \( S_R' \), the overall cost of \( a_i \) can be calculated as follows.

\[
oC_i = \frac{\sum_{k=1}^{\tau} s_{kR}^i h_k^i c_{kR}^i}{\sum_{k=1}^{\tau} h_k s_{kR}^i c_{kR}^i}
\]  

Platform: After gathering information about requesters and workers, the platform produces the outcome of the auction, including the chosen team members \( T \subseteq A \) and the payment vector \( P = \{p_1,p_2,\ldots,p_m\} \) of the workers in \( T \).

Therefore, the utility of the requester is the difference between the value and the payment:

\[
U_R = \begin{cases} 
\tilde{b} - \sum_{a_i \in T} p_i & \text{if } R \text{ is completed } \\
0 & \text{otherwise.}
\end{cases}
\]  

Similarly, the utility of winning worker \( a_i \in T \) is the reward from the auctioneer minus the cost of supplying the skills:

\[
U_i = \begin{cases} 
p_i - oC_i & \text{if } a_i \in T \\
0 & \text{otherwise.}
\end{cases}
\]  

Here, \( oC_i \) is the true overall cost corresponding to \( oC_i \).
2.2. Definition of Concepts
In this section, we introduce several economic properties we are desirable to achieve and some concepts we will use.

Definition 1. (Truthfulness). An auction is truthful shows that neither buyers nor sellers can gain a higher utility by deviating from their true value or cost. In our auction, it implies for $\forall a_i \in A$, if $oC_i = \tilde{C}_i$, $U_i$ is maximized; for $R$, if $b = \tilde{b}$, $U_R$ is maximized.

Truthfulness is the most crucial property in auction theory. Since exposing true bid or ask procures the highest utility, no rational buyer or seller would cheat any more. As a result, the trade can be free from market manipulation. In addition, irrespective of other’s behavior, players would simply apply their truth-telling strategy so that the system would reach an equilibrium. Myerson theorem is the most common theory used in auction which is shown as follows.

Theorem 1. (Myerson theorem) An auction mechanism is truthful if and only if [4]:

- Monotone allocation: If $a_i$ is allocated successfully by bidding $c_k$, he can also win by bidding $c_k' \leq c_k$.
- Critical value: If $a_i$ asks higher than the critical value, he will not win the auction.

Definition 2. (Individual Rationality). A winning seller is paid more than his cost and the winning buyer is charged less than her budget. In other words, if the trade is successful, for $\forall a_i \in T$, $p_i \geq oC_i$; for $R$, $\sum_{a_i \in T} p_i \leq b$.

Definition 3. (Budget Balance). Budget balance means that the budget from the requester is more than the total payment to the workers, i.e., $b - \sum_{a_i \in T} p_i \geq 0$.

Definition 4. (Social Welfare). Social Welfare is calculated as the aggregate utility of all the winning buyers, sellers and the auctioneer. It is also defined as the difference between the sum of winning buyers’ value and the sum of winning sellers’ costs.

3. TMC-SN Mechanism

3.1. Design Details
Many factors will influence the performance of a team. The requester prefers the workers who can make more contributions to the team. It is mentioned in Ref. [5] that group cohesion was positively related to overall performance. Therefore, considering both contribution and cohesion, we give the definitions as follows.

Definition 5. (Marginal Contribution). If $a_i$ is selected into the team $T$, $a_i$’s Marginal Contribution is the workload that $a_i$ can take to the uncovered skills in $S'_R$, i.e.,

$$\Delta_i = \sum_{s^k \in S'_R} h_k$$

(4)

Definition 6. (Team Cohesion). Team Cohesion is calculated as the communication cost with his neighbours if $a_i$ is selected to $T$, i.e.,

$$\Gamma_i = \sum_{a_j \in T'} d(a_i, a_j)$$

(5)

Definition 7. (Worker Fitness). Both Marginal Contribution and Team Cohesion are involved in the fitness of $a_i$. Then, the fitness of $a_i$, $\text{fitness}(a_i)$, is defined as

$$\text{fitness}(a_i) = \alpha \ast \text{nor}(\Delta_i) + (1 - \alpha) \ast \text{nor}\left(\frac{1}{\Gamma_i}\right)$$

(6)

Here, $\alpha \in [0, 1]$ is a tune parameter between Marginal Contribution and Team Cohesion and $\text{nor}(\cdot)$ is the normalization function.

The TMC-SN mechanism is shown in Algorithm 1. At the beginning of the mechanism, we search for the neighbours of the teammates and add them into $Neis$. In allocation rule, the worker with the smallest cost per fitness is selected as the candidate, i.e., the lowest $oC_i / \text{fitness}(a_i)$. It’s worth mentioning that the candidate workers are connected with each other. Only the neighbors of $a_i \in T$ will be added into the team. In other words, the winning workers form a connected subgraph at the end of the process. The payment rule follows the Myerson Theorem in order to make sure that the selected worker is paid the highest cost he can report or else he will not
be selected. We rebuild the team without the participation of $a_i$. It select $a_j$ from $\{ Neis' \setminus T \}$ with the smallest cost per fitness and $a_j$ still has the connection with the existing workers. Therefore,

\[
\frac{oC_i}{\text{fitness}(a_i)} \leq \frac{oC_j}{\text{fitness}(a_j)} \\
\frac{oC_i}{\text{fitness}(a_i)} \leq \frac{oC_j}{\text{fitness}(a_j)}*\text{fitness}(a_i) \tag{7}
\]

In the case that $\frac{oC_i}{\text{fitness}(a_i)} > \frac{oC_j}{\text{fitness}(a_j)}$, $a_j$ will be selected instead of $a_i$.

Repeat selecting $a_j$ until the budget is run out or the skills are covered. Throughout the process, the largest value $\{ \frac{oC_i}{\text{fitness}(a_i)}*\text{fitness}(a_i), p_i \}$ is set as the critical value of $a_i$. Moreover, if the payment for $a_i$ is less than remaining budget, $a_i$ will be the winner. During the process, $\Delta_i$ and $\Gamma_i$ are updated once a new worker is added into the team. In the end of the process, the trade is a success in the case that the skill requirements of $R$ are covered.

Algorithm 1 TMC-SN

Input: $SN, b, SR, C$
Output: $T, P$

1: $T \leftarrow \emptyset, S'_{R} \leftarrow S_{R}, p_1 \leftarrow 0$
2: while $A \neq \emptyset$ and $S'_{R}$ hasn’t been covered do
3: \hspace{1em} if $T \neq \emptyset$ then
4: \hspace{2em} $Neis \leftarrow T \cup \{ \cup_{a_i \in T} Nei(a_i) \}$
5: \hspace{2em} else
6: \hspace{2em} $Neis \leftarrow A$
7: \hspace{2em} end if
8: \hspace{1em} $a_i \leftarrow \text{argmin}_{a_i \in Neis} \left( \frac{oC_i}{\text{fitness}(a_i)} \right)$
9: \hspace{1em} $Neis' \leftarrow Neis \setminus \{ T \cup \{ a_i \} \}$
10: \hspace{1em} $K \leftarrow T$
11: \hspace{1em} while $b \geq p_i$ and $\Delta_j \neq 0$ do
12: \hspace{2em} $a_j \leftarrow \text{argmin}_{a_j \notin Neis'} \left( \frac{oC_j}{\text{fitness}(a_j)} \right)$
13: \hspace{2em} $p_i = \max\left\{ \frac{oC_j}{\text{fitness}(a_j)}*\text{fitness}(a_i), p_i \right\}$
14: \hspace{2em} $K \leftarrow K \cup \{ j \}$
15: \hspace{2em} end while
16: \hspace{1em} if $b \geq p_i$ then
17: \hspace{2em} $T \leftarrow T \cup \{ a_i \}, b \leftarrow b - p_i$
18: \hspace{2em} update the uncovered skills in $S'_{R}$
19: \hspace{1em} else
20: \hspace{2em} $A \leftarrow A \setminus \{ a_i \}$
21: \hspace{1em} end if
22: \hspace{1em} end while
23: if the skills of $S_{R}$ have not been covered then
24: \hspace{1em} $T \leftarrow \emptyset, p_1 \leftarrow 0$
25: \hspace{1em} end if

3.2. Analysis of Desirable Properties

Having given the detailed design of TMC-SN, we now prove that TMC-SN achieves truthfulness, individual rationality, and budget balance.

Theorem 2. TMC-SN achieves truthfulness.

Proof: The truthfulness of the requester is obvious. There are two possible reasons why $R$ can not be allocated. One is that the skills of $R$ cannot be covered by the remaining workers, and the other is that the budget is limited. If it’s the first reason, the requester can not be assigned anyway. If a requester with a limited budget wants to be successfully allocated, the requester must raise the bid until it is higher than the asks of the
For the workers, it is obvious that TMC-SN has a monotone allocation. If $a_i$ is selected by bidding $c_i^j$, he can also be selected by bidding $c_i^k$ with a smaller $\frac{aC_j}{\text{fitness}(a_i)}$. The payment to $a_i$ is decided by selecting a team without $a_i$’s participation. $a_j$ is selected from the set $Neis\{K \cup \{a_i\}\}$ with the minimum $\frac{aC_j}{\text{fitness}(a_i)}$. In the case that $\frac{aC_j}{\text{fitness}(a_i)} > \frac{aC_{\tilde{i}}}{\text{fitness}(a_i)}$, $a_j$ will be selected instead of $a_i$. The payment to $a_i$ is temporarily equal to $p_i' = \frac{aC_j}{\text{fitness}(a_i)} \cdot \text{fitness}(a_i)$. Here, $p_i'$ isn’t the highest price which $a_i$ can report because his competitors in $Neis\{K \cup \{a_i\}\}$ haven’t been considered entirely and the skill requirements of $a_i$ haven’t been covered. Therefore, the process of iteration continues until all the skills are covered or the budget is exhausted. Finally, the payment is the critical value, i.e., $p_i = \max\{\frac{aC_j}{\text{fitness}(a_i)} \cdot \text{fitness}(a_i), p_i\}$. Because of critical value, if $a_i$ asks higher than $p_i$, $a_i$ will be placed after the last selected worker, then $a_i$ will not be selected as the winner. From above, the critical value has been proved.

Adhering to Theorem 1, the truthfulness of workers is proved.

Theorem 3. TMC-SN achieves individual rationality.

Proof: For the requester, $b \geq p_i$ is checked every time before including a new teammate and the budget is updated after a successful trade. In other words, remaining budget is always higher than the payment to the new joined workers. Thus, $U_R = b - \sum_{a_i \in T} p_i \geq 0$ is satisfied. For the workers, we have $aC_i \leq \frac{aC_j}{\text{fitness}(a_i)} \cdot \text{fitness}(a_i) \leq \max\{\frac{aC_j}{\text{fitness}(a_i)} \cdot \text{fitness}(a_i), p_i\} = p_i$. Therefore, $U_i = p_i - aC_i = p_i - aC_i \geq 0$ is satisfied.

In general, the individual rationality of TMC-SN is achieved.

Theorem 4. TMC-SN achieves budget balance.

Proof: According to the payment rule, the remaining budget is checked whether enough for the payment of the new joined workers. Hence, we have $b - \sum_{a_i \in T} p_i \geq 0$. The budget balance of TMC-SN achieves.

4. Performance Evaluation

In this section, we simulate a crowdsourcing platform where workers are in a social network. In the experiment, the truthfulness of the TMC-SN is verified. The TMC-SN mechanism is compared with the complex task mechanism among workers are neglected. In that case, the Worker Fitness is only related to Marginal Contribution, i.e., $\alpha = 1$. From Figure 2(a), it can be observed that with the increase of workers, the number of transactions of non-SN is always higher than that of TMC-SN. This is because that TMC-SN considers the Team Cohesion of workers. The more stringent the trading are, the lower the transaction rate will be. Figure 2(b) indicates that the utility of requester increases as the augment of workers while the utility of workers decreases. It can be seen that the utility of requesters in non-SN is better than that in TMC-SN, but the utility of workers in non-SN is worse.
6 (a) Truthfulness for requesters. (b) Truthfulness for workers.

Figure 1: Truthfulness.

(a) Number of transactions. (b) Utility. (c) Social Welfare.

Figure 2: Comparison of TMC-SN and non-SN.

This demonstrates that the concern in Team Cohesion is profitable for workers but disadvantageous for requesters. Figure 2(c) shows that the Social Welfare of TMC-SN and non-SN is fluctuant. However, Social Welfare in TMC-SN performs better than non-SN as a whole. The concern in Team Cohesion is serviceable in teams of Social Welfare.

5. Conclusion
In this paper, we discuss the complex task assignment in crowdsourcing where the workers are in a social network. The TMC-SN mechanism is designed to solve the challenges in this scenario. In order to find out the most appropriate team, TMC-SN measures Worker Fitness of the worker for the team from both Marginal Contribution and Team Cohesion. The payment rule follows Myerson theorem. The theoretical analysis has proved that TMC-SN achieves truthfulness, individual rationality, and budget balance. In performance evaluation, the truthfulness of TMC-SN is further verified. We also compare the performance between TMC-SN and non-SN. The evaluation results show that TMC-SN is advantageous in terms of Social Welfare and favorable for the workers.

Acknowledgment
This work was funded in part by Open Project Foundation of Information Technology Research Base of CAAC (Grant No.CAAC-ITRB-201604); Natural Science Research Project of Guangling College of Yangzhou University (Grant No.ZKYB180020).

References
[1] Anand Inasu Chittilappilly, Lei Chen and Sihem Amer-Yahia, A Survey of General-Purpose Crowdsourcing Techniques, IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 9, pp. 2246-2266, April, 2016.
[2] Jordi Duch, Joshua S. Waitzman and Luis A. Nunes Amaral, Quantifying the Performance of Individual Players in a Team Activity, PLOS ONE, vol. 5, no. 6, p. e10937, June 2010.
[3] Mathijs de Weerdt, Yingqian Zhang and Tomas Klos, Distributed Task Allocation in Social Networks, in Proc. 6th AAMAS, Honolulu, May, 2007, pp. 500-507.
[4] R. Myerson, Optimal auction design, Mathematics of Operations Research, vol. 6, no. 1, pp. 58-73, February 1981.
[5] Heng-Li Yang and Jih-Hsin Tang, Team structure and team performance in IS development: a social network perspective, Information & Management, vol. 41, no. 3, pp. 335-349, 2004.