Inefficiency of Equilibria in Task Allocation by Crowdsourcing

Shigeo Matsubara
Department of Social Informatics, Kyoto University
matsubara@i.kyoto-u.ac.jp

Masanori Hatanaka
Department of Social Informatics, Kyoto University
hatanaka@ai.soc.i.kyoto-u.ac.jp

Huan Jiang
Department of Social Informatics, Kyoto University
jiang@ai.soc.i.kyoto-u.ac.jp

ABSTRACT

Crowdsourcing is a promising way for solving problems in a distributed manner, which outsources tasks to an undefined, generally large group of agents in the form of an open call. The important feature in crowdsourcing is that contractors choose tasks to solve, which is different from a procurement auction that a contractee chooses contractors. How efficient crowdsourcing is has not been sufficiently studied, although a lot of studies about auction mechanisms exist. To clarify the efficiency of crowdsourcing-style problem solving, we provide two analysis models: one is for the case that all the contractors have the same type and another is for the case that two types of contractors exist. Then, we examine the efficiency by comparing social surplus in an equilibrium allocation to social surplus in an optimal allocation. The theoretical analysis and the experimental results show how efficiency deteriorates in terms of the values of parameters such as the probability of success, an amount of reward, and the cost for completing the task.

1. INTRODUCTION

Crowdsourcing is an online, distributed problem solving. Howe defined crowdsourcing as the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call [7].

Notable examples of the model include InnoCentive and Amazon Mechanical Turk. InnoCentive is a global, online marketplace where organizations in need of innovation including companies, academic institutions, public sector, and non-profit organizations can utilize a global network of approximately 250,000 of the problem solvers from nearly 200 countries. Amazon Mechanical Turk is a crowdsourcing marketplace that enables computer programs to co-ordinate the use of human intelligence to perform tasks which computers are unable to do.

Crowdsourcing has recently received remarkable attention from the community of artificial intelligence research and multiagent systems research [3, 11, 5]. Although most of existing crowdsourcing models suppose to ask people to carry out the tasks, it is valuable to consider the model of crowdsourcing including human and computational agents. However, sufficient knowledge about its operation has not been accumulated. How difference in the properties of tasks affect contractors’ behaviors of selecting tasks and the efficiency in the problem solving is not clear. That is, this is a multiagent resource allocation problem.

The objective of this research is to identify what causes inefficiency, which helps to design an efficient allocation mechanism. This research deals with the similar protocol used in InnoCentive, which can be viewed as a standard protocol in crowdsourcing. Contractees announce their tasks, and then each of contractors chooses a task and reports the result if he/she completed the task. The contractee receiving reports from the contractors selects the best solution, and then pays reward to the contractor who reported the best solution. If the contractee is not satisfied with any reports, the contractee does not have to pay anything.

We compare social efficiency in an equilibrium allocation to that in an optimal allocation by setting up two models. One corresponds to the case where all the contractors have the same type in terms of skills and abilities and another corresponds to the case where two types of contractors are included. A critique may be posed that both of contractees and contractors are interested in their own utilities and they are not interested in social efficiency. However, if a problem solving method is socially inefficient, crowdsourcing will be taken over by other problem solving methods. Thus, the viewpoint of social efficiency is important.

The above mentioned protocol can be viewed as a contract-net protocol [12] in the field of distributed artificial intelligence. However, the situations are different because crowdsourcing assumes that a contractee asks an undefined, generally large group of contractors and contractors may fail to complete tasks.

Auction has been widely studied in the field of artificial intelligence research and multiagent systems research. Auction might be considered as an efficient method for this task allocation problem. However, an auction seems difficult to find an efficient task allocation because of the nature of crowdsourcing. In crowdsourcing the contractor is not asked to pay more than pre-specified payment at the beginning. Even if more than one contractor find solutions, the contractor is sufficient to pay reward only to the contractor.
2. RELATED WORKS

The study about the comparison between the optimal allocation and the allocation in equilibrium is called as the price of anarchy. A contractee pays the price of anarchy of crowdsourcing. The price of anarchy in other domains has been actively studied [8, 10]. Roughgarden discussed the selfish routing of agents in the network routing. The ratio between the cost of equilibrium flows and that of an optimal flow are evaluated in this field. However, to the best of our knowledge, the price of anarchy in crowdsourcing has not been discussed yet.

Zhao studied the caching mechanisms in order to guarantee the fault tolerance of data or services in the information network [13]. They discussed how to maintain the high probability of service providing, but did not consider strategic actions of agents.

Porter studied about resource allocation mechanisms in which they considered not only the execution cost of each agent, but also the success probability of tasks [9]. They examined the mechanism to maximize the social surplus, assuming that the individual rationality of agents should be held and a central designer aggregates information about agents and determines the resource allocation. In the crowdsourcing, however, contractors do not obey the task assignment by the center but autonomously select their tasks. Thus, the top-down assignment by contractees cannot be realized.

DiPalantino and Vojnovic considered the crowdsourcing as the all-pay auctions, and analyze it as a system of competitive contests [4]. They assumed that there are tasks with different rewards and agents having different skill levels and discussed the relation between reward and participation rate. However, they did not consider the case that many agents exist compared to the number of tasks. In other words, this paper examines the case that the number of agent types is one or two, where the skill levels do not sufficiently work for achieving the balanced allocation of contractors.

From the viewpoint of invention, stemming from the seminal work by Arrow [1], various models about allocation of resources for invention have been studied in the field of microeconomics. Barzel pointed out the possibility of overinvestment by the tragedy of commons [2]. Gilbert and Newbery claimed that the investment for invention have the role as a strategy for preventing new entry by dealing with the oligopoly models including a few contractors [6]. Our research, however, is different from theirs because we assume that large indefinite numbers of contractors exist and focus on the efficiency by contractors’ decision making processes in selecting tasks.

3. MODEL

In this section we present a formal model to enable rigorous discussion. In a crowdsourcing market, there exist a contractee, contractors, and a marketplace operator. The contractee has tasks \( t_j \) (\( j = 1, \cdots, l \)). This paper focuses on investigating contractors’ behaviors. Thus, assuming a single contractee is sufficient for discussions. Investigating multiple contractees’ strategic behaviors is included in our future works. If task \( t_j \) is successfully completed by the contractor, the contractee enjoys the utility of \( v_j \). Completing each task is independent from completing the other tasks. This paper assumes that \( v_j \) is announced as the amount of reward for completing task \( t_j \) before a deadline to report the result. This means that the amount of reward is given and no strategic manipulation on it is assumed in this paper. A market operator is assumed to be interested in maximizing social surplus.

On the other hand, contractors are characterized as type \( \theta_i \) (\( i = 1, \cdots, m \)). If a contractor of type \( \theta_i \) chooses task \( t_j \), he/she can successfully complete the task with a probability of \( \phi_{ij} \) and incurs the cost of \( c_{ij} \). It may happens that if a contractor invests more resources, which incurs a larger cost, the probability of success \( \phi_{ij} \) increases. However, this paper assumes that the amount of cost is a constant value of \( c_{ij} \) if the type of the contractor \( \theta_i \) and task \( t_j \) are given. Note that this cost is incurred by the contractor regardless of whether the task is completed or not. In addition, this paper assumes that a contractor of type \( \theta_i \) knows its probability of success \( \phi_{ij} \) and its cost of \( c_{ij} \). If contractor of type \( \theta_i \) choose task \( t_j \), his/her expected utility is defined as follows.

\[
 u = p_{ij}v_j - c_{ij}
\]

Here, \( p_{ij} \) represents the probability of becoming a winner among contractors choosing task \( t_j \). This expression means the difference between the expected reward and its cost. Individual rationality holds if the expected utility of each contractor is larger than zero or equal to zero.

The readers may have a question how the values of these parameters can be obtained, e.g., how the contractor can know the value of \( \phi_{ij} \). In this paper, we have tried to understand what outcome is obtained if a contractor is allowed to decide whether he/she participates in the market and which task to be tackled because we think these characteristics are essential to crowdsourcing. If a contractor can easily learn the distribution of the contractor’s type, it results in the outsourcing-style problem solving. Thus, we assume that a contractor can infers the value of \( \phi_{ij} \) based on the previous trials. The value of \( p_{ij} \) can be calculated if a contractor knows how many contractors participate in the market. The accurate number of contractors is difficult to know but even a roughly estimated number about the participants helps the contractors predict their payoff.

Social surplus is calculated as the sum of all participants’ utilities. This paper assumes that a contractee pays reward of \( v_j \) that is equal to his/her valuation of completing task \( t_j \). Thus, the contractee’s utility is equal to zero whether
4. SINGLE TYPE CONTRACTOR CASE

This section examines the case that all the contractors have the same type, that is, a single type of contractors’ case. Here, we examine two classes: sufficiently large number of contractors and small number of contractors. The former means that an additional contractor can be found if there is a room to obtain positive utility.

4.1 Sufficiently large number of contractors

We assume that completing a task is independent from completing the other tasks and a sufficiently large number of contractors exist. Thus, we can consider each task separately. First, we examine optimal contractor allocation. Because contractors have the same type, discussing only the number of contractors is sufficient. We do not have to consider which contractors should be included. If \( n_{ij} \) contractors choose task \( t_j \), social surplus about task \( t_j \) can be calculated as follows.

\[
v_j(1 - (1 - \phi_{ij})^{n_{ij}}) - n_{ij} c_{ij}
\]

The term of \( v_j \) is the contractor’s valuation of completing the task and the term of \( (1 - (1 - \phi_{ij})^{n_{ij}}) \) represents the probability that at least one contractor succeeds to complete the task. The second term represents the costs incurred by the contractors choosing task \( t_j \).

The optimal number of contractors \( n_{ij}^* \) can be obtained by differentiating the above equation with respect to \( n_{ij} \) and letting it zero, which is represented as follows.

\[
-v_j(1 - \phi_{ij})^{n_{ij}^*} \ln(1 - \phi_{ij}) - c_{ij} = 0
\]

Thus, we obtain the following expression.

\[
n_{ij}^* = \ln\left(\frac{-c_{ij}}{v_j \ln(1 - \phi_{ij})}\right) / \ln(1 - \phi_{ij})
\]

Next, we examine a Nash equilibrium by assuming that each contractor choose a task by itself. Each contractor keeps choosing a task as long as its expected utility is larger than zero or equal to zero. That is, if individual rationality condition holds, a contractor chooses the task. We assumed that contractors come to the market place randomly. However, this does not mean the first comer can choose a task without considering the subsequent contractors’ participation. If \( n_{ij} \) contractors choose task \( t_j \), each contractor’s expected utility can be calculated as follows.

\[
u = v_j(1 - (1 - \phi_{ij})^{n_{ij}})/n_{ij} - c_{ij}
\]

In this expression, the term of \((1 - (1 - \phi_{ij})^{n_{ij}})\) represents the probability that at least one contractor can successfully complete the task. Each contractor has the same probability of receiving the reward, because we assume that all the contractors have the same type. Therefore, \( v_j(1 - (1 - \phi_{ij})^{n_{ij}})/n_{ij} \) represents the contractor’s expected reward. The term of \( c_{ij} \) represents the cost incurred by each contractor.

As long as the expected utility is larger than zero or equal to zero, an additional contractor participates in doing task \( t_j \). Thus, the maximum number of contractors \( n_{ij}^* \) choosing task \( t_j \) is the largest number that satisfies the following inequality.

\[
v_j(1 - (1 - \phi_{ij})^{n_{ij}^*})/n_{ij}^* - c_{ij} \geq 0
\]

In the above expression if \( n_{ij}^* \) is a large number, \((1 - \phi_{ij})^{n_{ij}^*}\) approaches to zero, for example, if the probability of success \( \phi_{ij} = 0.5 \) and \( n_{ij} \) is 10, \((1 - \phi_{ij})^{n_{ij}^*}\) becomes 0.000077.

Here, the speeds that \((1 - \phi_{ij})^{n_{ij}^*}\) is approaching to zero and that \( n_{ij}^* \) is approaching to a large value should be examined. We assume that the contractor’s effort is sunk cost, i.e., an all-pay auction situation and the valuation of solving the task is a finite number. Thus, if the huge number of contractors participate in the market, the utility of each contractor becomes not zero but a negative value. Because we assume that contractors are rational, no other contractors will not participate in solving the task, if \( n_{ij}^* \) contractors have already participate in solving the task. Therefore, \( n_{ij}^* \) can be approximated to be \( v_j/c_{ij} \). This is the case that the cost incurred by each contractor is small compared to the amount of reward. In addition, social surplus is close to zero.

So far, we have not discussed which task a contractor chooses if more than one tasks exist. If \( v_j/c_{ij} \) becomes large enough for all the tasks, contractor’s expected utility approaches to zero, which means that a contractor is indifferent to a task selection.

From the above discussion, if \( v_j/c_{ij} \) is sufficiently large and a sufficiently large number of contractors exist, the efficiency ratio approaches to zero, that is, social surplus in an equilibrium allocation approaches to zero.
Table 1: The expected utilities of a contractor.

<table>
<thead>
<tr>
<th>$n_1$</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$n_1$</th>
<th>$t_1$</th>
<th>$t_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>0.66196</td>
<td>10</td>
<td>1.61384</td>
<td>0.82927</td>
</tr>
<tr>
<td>1</td>
<td>3.00000</td>
<td>0.67772</td>
<td>11</td>
<td>1.48819</td>
<td>0.84724</td>
</tr>
<tr>
<td>2</td>
<td>2.82000</td>
<td>0.69362</td>
<td>12</td>
<td>1.36733</td>
<td>0.85646</td>
</tr>
<tr>
<td>3</td>
<td>2.64720</td>
<td>0.70987</td>
<td>13</td>
<td>1.25103</td>
<td>0.88392</td>
</tr>
<tr>
<td>4</td>
<td>2.48127</td>
<td>0.72626</td>
<td>14</td>
<td>1.13912</td>
<td>0.90262</td>
</tr>
<tr>
<td>5</td>
<td>2.32192</td>
<td>0.74287</td>
<td>15</td>
<td>1.03138</td>
<td>0.92158</td>
</tr>
<tr>
<td>6</td>
<td>2.16883</td>
<td>0.75970</td>
<td>16</td>
<td>0.92766</td>
<td>0.94079</td>
</tr>
<tr>
<td>7</td>
<td>2.02714</td>
<td>0.77675</td>
<td>17</td>
<td>0.82770</td>
<td>0.96026</td>
</tr>
<tr>
<td>8</td>
<td>1.88038</td>
<td>0.79402</td>
<td>18</td>
<td>0.73153</td>
<td>0.98000</td>
</tr>
<tr>
<td>9</td>
<td>1.74450</td>
<td>0.81153</td>
<td>19</td>
<td>0.63882</td>
<td>1.00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>0.54946</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: The social surplus in Nash equilibrium.

<table>
<thead>
<tr>
<th>equilibrium solution</th>
<th>$n_1$</th>
<th>$n_2$</th>
<th>$n$</th>
<th>social surplus</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>4</td>
<td>20</td>
<td>46</td>
<td>18.605755</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>20</td>
<td>30</td>
<td>24.627385</td>
</tr>
</tbody>
</table>

4.2 Small number of contractors

It turns out that social surplus in a Nash equilibrium approaches to zero and efficient allocation cannot be attained. However, if a sufficiently large number of contractors do not exist, the expected utility does not become zero. This means that a task selection is important in maximizing its utility. Here, we assume that $n$ contractors exist and two tasks of $t_1$ and $t_2$ exist. The numbers of contractors who carry out task $t_1$ and $t_2$ are $n_1$ and $n_2$, respectively ($n_1 + n_2 = n$). At a Nash equilibrium the expected utility of a contractor carrying out task $t_1$ and the expected utility of a contractor carrying out task $t_2$ should be equal to each other. Otherwise, for example, if a contractor can obtain larger benefit by carrying out task $t_1$, a contractor exists who can increase his/her utility by changing from $t_2$ to $t_1$, that is, $n_1$ increases ($n_2$ decreases) until both tasks get to give the same utility.

$$v_1(1 - (1 - \phi_{11})^{n_1})/n_1 - c_{11} = v_2(1 - (1 - \phi_{12})^{n_2})/n_2 - c_{12}$$

Here, $n_1$ and $n_2$ are not continuous value. Thus, we define it as an equilibrium at $(n_1, n_2)$ if both of the following inequalities are satisfied.

$$v_1(1 - (1 - \phi_{11})^{n_1})/n_1 - c_{11} \geq v_2(1 - (1 - \phi_{12})^{n_2+1})/(n_2+1) - c_{12}$$

$$v_2(1 - (1 - \phi_{12})^{n_2})/n_2 - c_{12} \geq v_1(1 - (1 - \phi_{11})^{n_1+1})/(n_1+1) - c_{11}$$

We will illustrate the optimal task allocation does not coincide the allocation in a Nash equilibrium. Suppose that two tasks $t_1$ and $t_2$ exist, both of whose rewards $v_1$ and $v_2$ are equal to 100. We also assume that $\phi_{11} = 0.06$, $\phi_{12} = 0.02$, for the success probabilities, and $c_{11} = 3$, $c_{12} = 1$ for the costs.

Table 1 shows the expected utility of each $n_{11}$ for the number of contractors $n \equiv 20$, which tells that an equilibrium allocation is that $n_{11} = 16$ and $n_{12} = 4$. On the other hand, as shown in Table 2, an socially efficient allocation is $n_{11} = 9$ and $n_{12} = 11$.

If the properties of two tasks are different from each other, socially efficient allocation does not coincide to the allocation realized in a Nash equilibrium. Thus, we examine how the success probability, reward, cost affect social efficiency.

4.2.1 The probability of success

Proposition 1. If the success probability of completing task $t_1$ ($t_2$) increases, the number of choosing $t_1$ does not decrease in the optimal allocation, while it does not increase in an equilibrium.

Suppose that $\phi_{11}$ changes to $\phi_{11}'$ ($> \phi_{11}$). At the optimal allocation, the following expressions should be maximized.

$$\max\{v_1(1 - (1 - \phi_{11})^{n_{11}}) - n_{11}c_{11} + v_2(1 - (1 - \phi_{12})^{n_{12}}) - n_{12}c_{12}\}$$

$$\max\{v_1(1 - (1 - \phi_{11}')^{n_{11}}) - n_{11}'c_{11} + v_2(1 - (1 - \phi_{12})^{n_{12}}) - n_{12}c_{12}\}$$

Here, assume that $n_{11}'$ is larger than $n_{11}$. The increase of $1 - (1 - \phi_{11})^{n_{11}}$ is larger than that of $1 - (1 - \phi_{11}')^{n_{11}}$, while the decrease of $n_{11}c_{11}$ is equal to that of $n_{11}'c_{11}$ when the number of contractors carrying out task $t_1$ increases from $n_{11}$ to $n_{11}'$. The changes of terms related to task $t_2$ are also the same as each other because the values of the parameters are the same at expressions 1 and 2. Thus, if social surplus increases by increasing from $n_{11}$ to $n_{11}'$, for expression 2, social surplus also increases by doing the same for expression 1. This contradicts with the assumption that expression 1 is maximized at $(n_{11}, n_{12})$. Therefore, $n_{11}'$ is not larger than $n_{11}$.

On the other hand, the following equation is satisfied at an equilibrium.

$$v_1(1 - (1 - \phi_{11})^{n_{11}})/n_{11} - c_{11} = v_2(1 - (1 - \phi_{12})^{n_{12}})/n_{2} - c_{12}$$

If the success probability of completing task $t_1$ increases from $\phi_{11}$ to $\phi_{11}'$, the left-hand side increases. If a contractor change his mind to choose $t_2$ instead of $t_1$, the difference between the expected utility by choosing $t_1$ and the expected utility by choosing $t_2$ increases, which means it cannot reach a Nash equilibrium. Thus, $n_{11}'$ is not smaller than $n_{11}$.

Figure 1 shows how the efficiency ratio changes if the probability of success $\phi_{11}$ increases. The horizontal axis represents the probability of success $\phi_{11}$, while the vertical axis represents the efficiency ratio. Here, we set $v_1 = v_2 = 100$, and $c_{11} = c_{12} = 3$. The three lines in the figure corresponds to the cases of $\phi_{12} = 0.06, 0.12, 0.24$. This figure confirms that the above discussion is correct.

4.2.2 An amount of reward

The increase of $\phi_{11}$ affects the behaviors at the optimal allocation and a Nash equilibrium differently; that is, at the optimal allocation $n_{11}$ decreases, while at a Nash equilibrium $n_{11}$ increases. Thus, as $\phi_{11}$ increases, the efficiency ratio monotonically decreases.

On the other hand, as reward $v_1$ increases, the efficiency ratio does not change monotonically.

Given reward $v_1$, the optimal allocation $(n_{11}^{\text{opt}}, n_{12}^{\text{opt}})$ and an allocation in a Nash equilibrium $(n_{11}^{\text{Nash}}, n_{12}^{\text{Nash}})$ can be obtained. As the reward $v_1$ increases, $n_{11}^{\text{opt}}$ reaches $n$ and
Let \( n_{i}^{Nash} \) reach \( n \). If \( n_{i}^{opt} \) is smaller than \( n_{i}^{Nash} \), we can find the following sequence of \((n_{i1}, n_{i2}, n_{i1}^{Nash}, n_{i2}^{Nash})\), \((n_{i1}^{opt} + 1, n_{i2}^{opt} + 1, n_{i1}^{opt}, n_{i2}^{opt})\), \((n_{i1}^{Nash}, n_{i2}^{Nash})\). At a change from \((n_{i1}^{opt}, n_{i2}^{opt}, n_{i1}^{Nash}, n_{i2}^{Nash})\) to \((n_{i1} + 1, n_{i2}^{opt} + 1, n_{i1}^{Nash}, n_{i2}^{Nash})\), the gap between the contractors’ allocation in an optimal state and in a Nash equilibrium becomes large. This means the efficiency ratio becomes worse. On the other hand, at a change from \((n_{i1}^{opt} + 1, n_{i2}^{opt} + 1, n_{i1}^{Nash}, n_{i2}^{Nash})\) to \((n_{i1}^{Nash} - 1, n_{i2}^{Nash} - 1, n_{i1}^{Nash}, n_{i2}^{Nash})\), the gap between the contractors’ allocation in an optimal state and in a Nash equilibrium becomes small. This means the efficiency ratio is improved. Thus, the change of the efficiency ratio is not monotonic.

Figure 2 shows how the efficiency ratio changes if the reward increases in an example. The horizontal axis represents the reward \( t_1 \), while the vertical axis represents the efficiency ratio. Another reward \( v_2 \) is fixed to 100. The other parameters are also fixed except \( t_1 \). We let \( c_{11} = c_{12} = 3 \). The data are plotted for the three cases of \( \phi_{11} = 0.06, 0.12, 0.24 \). The difference between the rewards of \( t_1 \) and \( t_2 \) is not large, the efficiency ratio decreases first, then the efficiency ratio becomes 1.

Discontinuities in Figure 2 are due to the change of the contractors’ allocation in the equilibrium. The number of the contractors are set to 20. So, for example, the change of \( n_{11} \) from 10:10 to 11:9 is not so small to smoothen the profiles in the figure.

We will explain why the efficiency ratio has the trend of decrease, while the small fluctuations are caused by the change of the contractors’ allocation in the equilibrium. Figure 3 shows the contractors’ allocation in the case of \( \phi_{11} = 0.06 \). The horizontal axis represents the reward for task \( t_1 \). The vertical axis represents the ratio of contractors carrying out task \( t_1 \). Figure 3 (a) shows the optimal task allocation, while Figure 3 (b) shows the allocation in a Nash equilibrium.

When the difference between the amounts of rewards is not large, the number of contractors choosing task \( t_1 \) rapidly increases in the case of a Nash equilibrium as the reward of task \( t_1 \) increases. All contractors choose task \( t_1 \) at \( v_1 = 170 \). On the other hand, the number of contractors choosing task \( t_1 \) gradually increases in the case of an optimal allocation as the reward of task \( t_1 \) increases. The value of reward \( v_1 \) where all contractors choose task \( t_1 \) is 330. Thus, it is confirmed that the difference of these profiles in task allocations causes social inefficiency.

4.2.3 Cost for completing a task

This case can be discussed in the same manner as the case of changing the rewards. Figure 4 shows how social efficiency changes as the cost of \( t_1 \) changes. The horizontal axis represents the cost of \( t_1 \), while the vertical axis represents the efficiency ratio. Here, we set \( v_1 = v_2 = 1000 \), and \( c_{11} = 1 \). The three lines in the figure corresponds to the cases of \( \phi_{11}(= \phi_{12}) = 0.3, 0.5, 0.9 \). Discontinuities in Figure 4 are also due to the change of the contractors’ allocation in the equilibrium as explained in Figure 2. This figure confirms that the efficiency ratio decreases if the cost of completing a task increases. We can observe two break points in each line. These correspond to the points that the number of contractors choosing \( t_1 \) changes.

5. MULTIPLE TYPES CONTRACTOR CASE

In the previous section, we assume the case where all the contractors have the same type. From a practical standpoint, however, some contractors might be experts, and their success probability or cost might be different from general contractors. This section examines the case that multiple types of contractors’ case. Here, we examine the case of two types for the sake of simplicity. Types of contractors are denoted by \( \theta_1, \theta_2 \), the success probabilities of task \( t_j \) are denoted by \( \phi_{1j}, \phi_{2j} \), and the costs of task \( t_j \) are denoted by \( c_{1j}, c_{2j} \). When the number of type \( \theta_i \) contractors who carry out task \( t_j \) is equal to \( n_{ij} \), the expected surplus \( V_j \) of task \( t_j \) is expressed as follows.

\[
V_j = v_j (1 - \prod_{i=1}^{n} (1 - \phi_{ij})) - \sum_{i=1}^{n} n_{ij} c_{ij}
\]

As the above expression shows, expected utilities of contractors depend on the success probability of contractors of other types, and the number of participants of the task. We introduce the probability of becoming a winner, \( \alpha \), given the success probability of contractors of other types, and
the number of participants of the task. How to calculate \( \sigma \) is given below. By using \( \sigma \), the expected utility \( u_{1j} \) of the contractors who have type \( \theta_i \) is expressed as follows.

\[
u_{1j} = v_j \cdot \sigma(\phi_{1j}, \phi_{2j}, n_{1j}, n_{2j}) - c_{1j}
\]

Here, \( \sigma() \) is calculated as follows.

\[
\sigma(\phi_{1j}, \phi_{2j}, n_{1j}, n_{2j}) = \sum_{k=0}^{n_{1j}-1} \sum_{l=0}^{n_{2j}-1} \theta(\phi_{1j} \cdot n_{1j-1}C_k \cdot (\phi_{1j})^k \cdot (1 - \phi_{1j})^{(n_{1j}-1)-k})
\]

\[
\cdot n_{2j}C_l \cdot (\phi_{2j})^l \cdot (1 - \phi_{2j})^{n_{2j}-l})/(k + l + 1)
\]

The first term of the \( u_{1j}'s \) expression represents the expected reward for attaining task \( t_j \). \( \sigma() \) represents the winning probability of the contractors of type \( \theta_i \). If the number of contractors of type \( \theta_i \) who succeed the task is \( k + 1 \) and the number of contractors of type \( \theta_2 \) is \( l \), the probability of attaining the task calculates as follows.

\[
\phi_{1j} \cdot n_{1j-1}C_k \cdot (\phi_{1j})^k \cdot (1 - \phi_{1j})^{(n_{1j}-1)-k})
\]

\[
\cdot n_{2j}C_l \cdot (\phi_{2j})^l \cdot (1 - \phi_{2j})^{n_{2j}-l})/(k + l + 1)
\]

The winning probability of contractors of type \( \theta_i \) is calculated by multiplying the probability of attaining the task by \( 1/(k + l + 1) \) because we assume that the best contractor is chosen at random if many contractors succeed the task. In \( \sigma() \), the sum from \( (k, l) \) equals \((0, 0) \) to \((n_{1j} - 1, n_{2j}) \) of the winning probability is calculated. The similar calculation holds for contractors of type \( \theta_2 \).

Next, we illustrate the conditions of Nash equilibrium. Given the number of contractors of type \( \theta_i \) is \( n_i \), the conditions of Nash equilibrium where the allocation (\( n_{11}, n_{12}, n_{21}, n_{22} \)) satisfies \( n_{11} + n_{12} = n_1 \) and \( n_{21} + n_{22} = n_2 \) are defined as follows.

\[
v_1 \cdot \sigma(\phi_{11}, \phi_{21}, n_{11}, n_{21}) - c_{11} \geq 0
\]

\[
v_2 \cdot \sigma(\phi_{12}, \phi_{22}, n_{12} + 1, n_{22}) - c_{12} \geq 0
\]

and

\[
v_1 \cdot \sigma(\phi_{11}, \phi_{21}, n_{11} + 1, n_{21}) - c_{11} \geq 0
\]

\[
v_2 \cdot \sigma(\phi_{12}, \phi_{22}, n_{12}, n_{22}) - c_{12} \geq 0
\]

and

\[
v_1 \cdot \sigma(\phi_{21}, \phi_{11}, n_{21}, n_{11}) - c_{21} \geq 0
\]

\[
v_2 \cdot \sigma(\phi_{22}, \phi_{12}, n_{22} + 1, n_{12}) - c_{22} \geq 0
\]

\[
v_1 \cdot \sigma(\phi_{21}, \phi_{11}, n_{21} + 1, n_{11}) - c_{21} \geq 0
\]

It is computationally hard to calculate the expected utility, when \( n_i \) or \( n_j \) become large. In this section, we experimentally calculate Nash equilibrium about some concrete examples, and compare the social surplus in Nash equilibrium with the optimal one.

Suppose that two tasks \( t_1 \) and \( t_2 \) exist, \( v_1 = v_2 = 100 \), and the ability of contractors are given by \( \phi_{11} = 0.05, \phi_{12} = 0.02, \phi_{21} = 0.02, \phi_{22} = 0.04, c_{11} = 3, c_{12} = 1, c_{21} = 1, c_{22} = 3 \). In this example, the allocation in a Nash equilibrium becomes \( n_{11} = 10, n_{12} = 0, n_{21} = 3, n_{22} = 7 \), the social surplus is \( V_1 + V_2 = 14.50 \). On the other hand, the optimal allocation becomes \( n_{11} = 2, n_{12} = 8, n_{21} = 10, n_{22} = 0 \), and the social surplus is \( 17.18 \). Thus, we found that the allocation in a Nash equilibrium did not become the optimal allocation in this example.

We will analyze the case where experts of tasks exist. We assume that experts have the high success probability of specific tasks, and examine the case where contractors of type \( \theta_i \) are experts of task \( t_i \).

Figure 5 shows the case where the settings are \( v_1 = v_2 = 100, c_{11} = c_{22} = 3, c_{12} = c_{21} = 1 \). The data are plotted for
Figure 5: The success probability of experts and the social efficiency.

Table 3: Inefficiency of equilibrium.

<table>
<thead>
<tr>
<th>(φ_{11}, φ_{12}) = (0.075, 0.05)</th>
<th>n_{11}</th>
<th>n_{12}</th>
<th>n_{21}</th>
<th>n_{22}</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>OA</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(φ_{11}, φ_{12}) = (0.08, 0.05)</th>
<th>n_{11}</th>
<th>n_{12}</th>
<th>n_{21}</th>
<th>n_{22}</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0.88</td>
</tr>
<tr>
<td>OA</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(φ_{11}, φ_{12}) = (0.11, 0.05)</th>
<th>n_{11}</th>
<th>n_{12}</th>
<th>n_{21}</th>
<th>n_{22}</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>EA</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>1.00</td>
</tr>
<tr>
<td>OA</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>-</td>
</tr>
</tbody>
</table>

the three cases of φ_{12}(= φ_{21}) = 0.02, 0.05, 0.2. Discontinuities in Figure 5 are also due to the change of the contractors’ allocation in the equilibrium as explained in Figures 2 and 4.

As mentioned in previous section, when the success probability of tasks becomes high, the expected utility of contractors increases. Thus, in this situation, the number of allocation to experts increases. On the other hand, the expected surplus of the task with high success probability becomes enough large, even if the number of allocation is small. Therefore, in the optimal allocation, the number of allocation to experts is relatively small. As a result, inefficient allocations arise when the success probability of experts is enough high. We also found that these problems can arise regardless of the cost of contractor’s type.

Meanwhile, we discovered that the efficiency ratio decreases rapidly in specific range in the situation that the cost of experts is larger than the cost of amateurs, like c_{11} = c_{22} = 3, c_{12} = c_{21} = 1.

To understand this problem, we analyze this case in detail. Table 3 shows the allocation of the case, and two cases when φ_{11} is changed slightly. The term ER means the efficiency ratio, EA means the allocation in equilibrium, and OA means the optimal allocation. In the case where φ_{11} = 0.08, a reversal phenomenon arises. Indeed, although special tasks are assigned to the experts in Nash equilibrium, in the optimal allocation, they are assigned to the amateurs. This is a mechanical problem, so we need to propose new matching systems to solve it. Designing the reward systems of crowdsourcing are considered as promising approaches.

6. CONCLUDING REMARKS

Crowdsourcing is a distributed problem solving method. This paper presented models of contractors’ allocation in crowdsourcing and examined to what extent social surplus can be attained in an equilibrium allocation to understand the efficiency in crowdsourcing-style problem solving.

In the case that all contractors have the same type, we showed that social surplus realized in an equilibrium allocation becomes close to zero if a sufficiently large number of contractors exist. In addition, we analyzed how social efficiency in an equilibrium allocation changes as the parameters such as the probability of success, an amount of reward, the cost for completing the task change. The results showed that the efficiency linearly decreases as the difference between the probabilities of success between two tasks becomes large. Moreover, we clarified the difference in the speed of convergence to the state that all contractors choose the same task between in an equilibrium allocation and in optimal allocation causes inefficiency when the difference between the amounts of reward becomes large.

In the case that two types of contractors, expert and amateur, exist, we showed that the efficiency in an equilibrium allocation decreases compared to the efficiency in an optimal allocation as the probability of success for experts becomes large. In addition, we observed an interesting inversion phenomenon that each task is chosen by its experts in an optimal allocation, while each task is chosen by its amateurs in an equilibrium allocation.

Based on the obtained results, we are planning to design a mechanism that can avoid highly inefficient allocation. An example is to design how to impose a market operation cost on participants. A question is whether is better, imposing the equal fees on contractors or the proportional fees on contractees according to their rewards. The analysis done in this paper tells that imposing the equal fees is better if the difference between rewards is sufficiently large, while imposing the proportional fees is likely to be better if the difference between rewards is small, although in the latter case an efficiency realized in an equilibrium allocation is sensitive to subtle change of fees. The more detailed analysis is included in our future works.

7. ACKNOWLEDGMENTS

This research was partially supported by a Grant-in-Aid for Scientific Research (B) (22300052, 2010-2012) from Japan Society for the Promotion of Science (JSPS).

8. REFERENCES


