Trade of a Problem-solving Task

Shigeo Matsubara
NTT Communication Science Laboratories
NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun,
Kyoto 619-0237 Japan
matsubara@cslab.kecl.ntt.co.jp

ABSTRACT

This paper focuses on a task allocation problem, particularly in cases where the task is to find a solution to a search problem or a constraint satisfaction problem. If the search problem is difficult to solve, a contractor may fail to find a solution. Here, the more computational resources, such as the CPU time, the contractor invests in solving the search problem, the more likely a solution will be found. This brings about a new problem in which a contractee has to find an appropriate quality level in task achievement as well as to efficiently allocate a task among contractors. For example, if the contractee asks the contractor to find a solution with certainty, the payment from the contractee to the contractor may exceed the contractee's benefit from obtaining a solution, which discourages the contractee from trading a task. However, it is difficult to solve this problem because the contractee cannot ascertain the contractor's problem-solving ability such as the amount of available resources and knowledge (e.g. algorithms, heuristics) nor monitor how many resources are actually invested in solving the allocated task. To solve this problem, we propose a task allocation mechanism that is able to choose an appropriate quality level in task achievement and prove that this mechanism guarantees that each contractor reveals its true information. Moreover, we show that our mechanism can increase the contractee's utility compared with a simple auction mechanism by using computer simulations.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*; K.4.4 [**Computers and Society**]: Electronic Commerce

General Terms

Design, Economics

Keywords

task allocation, auction, contract theory, multiagent systems

1. INTRODUCTION

The development of computer networks has enabled us to solve problems by using distributed computational resources across the networks. If we can easily borrow as many resources from other computers as we need at any time, we do not have to maintain extra resources by ourselves. Moreover, it may enable us to solve large-scale problems that have been difficult to solve so far.

One of the difficulties of dealing with the trade of computing power is that available resources change every moment. This problem can be solved by incorporating software agents. However, other difficulties exist, where although a lot of resources are available on the networks, the total amount of resources is limited and different resources are managed by different parties/organizations. Therefore, we adopt a mechanism design approach, which has recently been given sufficient attention by multiagent researchers [13, 2, 15, 9, 16]. In particular, we examine the interaction between computing solutions and negotiation. This direction in multiagent research can be found in [5].

This paper focuses on the following scenario: a contractee gives its problem to a contractor, then the contractor solves the problem by using the contractor's computational resources, after which the contractor returns the obtained solution to the contractee. In this situation, the contractee has to evaluate the contractors' problem-solving ability to choose an appropriate contractor.

Suppose that a task to be traded is not a simple arithmetic computation, but a search problem. In this case, a simple auction mechanism does not work well due to the nature of the problem. If the search problem is difficult to solve, a contractor may fail to find a solution. Here, the more computational resources, such as the CPU time, the contractor invests in solving the search problem, the more likely a solution will be found.

This brings about a new problem in which a contractee has to find an appropriate quality level in task achievement in addition to efficiently allocating the task among contractors. That is, if the contractee asks the contractor to find a solution with certainty, it may happen that the payment from the contractee to the contractor exceeds the contractee's benefit from obtaining a solution, which discourages the contractee from trading tasks. On the other hand, if the contractee requires a too-low-quality task achievement in order to reduce the payment, the probability of finding a solution becomes very small, which results in lowering the contractee's utility.

However, solving this problem is difficult because (1) the contractee cannot ascertain the contractor's problem-solving ability, such as the amount of available resources (e.g. the CPU time, memories) and knowledge (e.g. algorithms, heuristics) or, (2) the contractee cannot monitor the amount of resources that are actually invested in solving the allocated task. The former makes it difficult to efficiently allocate the task among contractors. The latter makes it difficult to induce each contractor's appropriate behavior; that is, even if the contractee can find an appropriate quality level in task achievement, the contractor may provide services of a different quality if it increases the contractor's utility.

To solve this problem, we have developed a new mechanism in which a contractee first announces a task, then contractors submit their performance profiles. Here, a performance profile represents the relation between a quantity of invested resources and the probability of finding a solution. The contractee then determines a winner and calculates a set of rewards based on the declared performance profiles. We prove that this mechanism can induce contractors' true declarations and experimentally show that our mechanism can increase the contractee's utility compared with a simple auction mechanism. Moreover, we propose an extended mechanism for multiple-task cases.

Related studies are being done on contract theory, where an agent selection problem in a principal-agent model has already been discussed [7, 4, 14]. However, their models do not deal with a situation in which the tasks may not be completed and the contractee has to determine an appropriate quality level in task achievement. Another related study is conducted in the field of computer science, where the researchers have tried to incorporate the concept of execution uncertainty in distributed computing studies into mechanism design [10]. They dealt with situations where two types of unknown information exist: the cost to achieve a task and the probability of failure. However, they limited their discussion to the case where a contractor incurs a fixed cost that does not depend on effort.

The remainder of this paper is organized as follows. First, we describe the formal model, then we propose a new mechanism for a single-task allocation problem. Next, we discuss multiple-task cases. Following that, we discuss the limitations of our mechanism, and finally, we give our concluding remarks.

2. MODEL

In this section we present a formal model to enable rigorous discussion. In a trading environment, there exist a contractee and multiple contractors. The contractee has a search problem or a constraint satisfaction problem to be solved. The contractor who is allocated a problem-solving task attempts to find a solution to the problem. If the contractor does find a solution, the contractor reports the solution to the contractee who subsequently pays a reward to the contractor.

The cost for solving the allocated problem k_i depends on contractor i's capabilities, which include the available computational resources (e.g., the CPU speed, memory size) and knowledge (e.g., algorithms, heuristics). We call these capabilities the contractor's technology.

Assumption 1. Contractor i's technology to solve the allocated problem k_i is characterized by $\alpha_i(k_i)$.

For simplicity, we designate the technology $\alpha_i(k_i)$ as α_i.

Contractor i assigns its resources to solving the allocated problem k_i. Assigning more computational resources, e.g., spending more CPU time, increases the probability of finding a solution, although contractor i incurs more costs.

Assumption 2. The amount of assigned resources by contractor i is called the contractor's effort, which is denoted by e_i. We assume that contractor i's loss caused by its effort e_i is equal to e_i.

Assumption 3. The result of a task achievement takes one of two states: success or failure.

Assumption 4. The contractee can verify the validity of the reported solution at a negligible cost.

This means that the problem to be traded is not an optimization problem but a kind of a satisfaction problem. This assumption excludes the possibility that the contractor makes up a false solution without making any effort and reports it to the contractee.

By its very nature, it is difficult to estimate in advance the cost of finding a solution to a search problem in each case. Therefore, we introduce the following probability function.

Assumption 5. The result of contractor i achieving a task is determined probabilistically based on contractor i's technology and effort. Let $p(e_i; \alpha_i)$ denote the probability of successfully finding a solution when contractor i's technology is α_i and its effort is equal to e_i.

We call this probability function contractor i's performance profile. This probability can be viewed as the quality of the problem-solving services. We believe that contractors can obtain this statistical information, and this claim can be supported by referring to some research results on search methods [8, 3]. The researchers have attempted to specify how problem structures, e.g., the density of the constraints, affect the search cost [1].

Here, we introduce an order relation in terms of the value of technology α_i.

Assumption 6. If $p(e; \alpha_i) > p(e; \alpha_j)$, we say that the value of the technology held by contractor i is greater than that of contractor j.

The better the contractor's technology becomes, the more efficient its problem-solving is. That is, a contractor who has a higher technology value can provide the same level of problem-solving services at a lower cost than other contractors who have lower technology values. Note that given a task specification, the value of technology α_i is uniquely defined for each contractor i, while the value of effort e_i depends on contractor i's decision.

[1]It may seem difficult to obtain a performance profile by observing the input-output relation because it is a two-valued input function. However, from a contractor's viewpoint, the performance profile is a single-value input function. This is because the technology value is introduced to order contractors' performance profiles and it is sufficient for a contractor to know the relation between the invested computational resources and the probability of succeeding the problem solving when he/she tries to obtain the performance profile.

Assumption 7. The probability for success in solving a problem by contractor i, $p(e_i;\alpha_i)$, is an increasing concave function of contractor i's effort e_i and an increasing concave function of contractor i's technology α_i. Additionally, $p(e;\alpha_i)$ and $p(e;\alpha_j)$ do not intersect with each other if $\alpha_i \neq \alpha_j$.

This concavity assumption means that if the probability of success in solving a problem is already high, it becomes difficult to increase the probability by investing additional resources. We need the concavity assumption to make the analysis tractable. Our future work will include relaxing the concavity assumption, although it is quite a difficult challenge.

In a real-world situation, the no-intersect assumption might be violated. This problem will be addressed in Section 5.

Assumption 8. The contractee cannot observe contractors' performance profiles, including their technology values and their efforts.

This is because contractors' computational resources are maintained by the contractors.

Assumption 9. The contractee pays a reward w_i^H to contractor i if contractor i succeeds in solving a problem and pays a reward w_i^L if its problem solving fails. If contractor i is not allocated any tasks, $w_i^H = w_i^L = 0$.

Contractor i's expected utility, $U_i(e_i)$, is defined as follows.

Definition 1.

$$U_i(e_i) = p(e_i;\alpha_i)w_i^H + (1 - p(e_i;\alpha_i))w_i^L - e_i$$

The contractor is risk-neutral.

On the other hand, the contractee's utility is defined as follows.

Definition 2.

$$p(e_i;\alpha_i)v - (p(e_i;\alpha_i)w_i^H + (1 - p(e_i;\alpha_i))w_i^L),$$

where v represents the valuation value of obtaining a solution.

In this expression, the first term represents the increase in the contractee's profit caused by obtaining a solution, and the second and third terms represent the payments to the contractor. The contractee is risk-neutral.

Finally, we assume the following.

Assumption 10. Contractors' collusion does not exist.

3. MECHANISM

In this section, we propose a new mechanism that determines the allocation of the task and contract (a set of rewards). This section examines the single-task case, and the next section discusses multiple-task cases. Although discussions in Sections 3.3 and 3.4 are borrowed from existing contract theory, the other results are from our original work.

3.1 Failure of a simple auction mechanism

We first explain why a simple auction mechanism does not work well for cases in which the task to be allocated is to find a solution to a search problem. If the contractee asks the contractor to find a solution with certainty, this task allocation problem is equivalent to an ordinary task allocation problem that can be solved by using an auction. However, if the problem to be traded is difficult to solve, the payment from the contractee to the contractor may exceed the contractee's benefit from obtaining a solution, which discourages the contractee from trading its problem-solving task.

Once the contractee gives up trying to obtain a solution with certainty, the contractee has to find an appropriate quality level in the attempt to solve the problem, which can be described as follows. If the contractor's marginal cost caused by making an effort of e_i is lower than the contractee's marginal profit by increasing the probability of success in solving the problem, the contractee has a chance to increase its profit by paying compensation to the contractor and inducing the contractor's effort of e_i. Namely, the contractor should require a higher level of quality in solving the problem. However, the relation between the contractor's marginal cost and the increase of the probability of success in solving the problem is the contractor's private information. Thus, it is difficult for the contractee to find an appropriate quality level in problem solving.

Even if the contractee knows the appropriate level of quality required for solving the problem, a simple auction mechanism is not sufficiently effective. The reason is as follows. Suppose there are two contractors: an efficient contractor and an inefficient contractor. In addition, suppose that the contractee employs a Vickrey auction and asks the two contractors to bid their costs to provide the problem-solving services at a predetermined level of the quality.

Here, if the inefficient contractor declares its true cost, the inefficient contractor loses the auction because its cost is higher than that of the efficient contractor. If the inefficient contractor understates its cost and wins the auction, he/she suffers a loss. However, the inefficient contractor can win the auction without suffering a loss by providing a lower level of quality in problem solving than the predetermined level, which reduces the cost, thus the inefficient contractor still makes a profit. This is possible because the contractee cannot distinguish between the following two situations: the contractor shirks and fails to find a solution or does not shirk and fails to find a solution. In this case, the efficient contractor also understates its cost to win the auction. Thus, the two contractors understate their costs and the winner provides quite a low-quality problem-solving service than the predetermined level, which results in reducing the contractee's utility.

3.2 Mechanism for determining an allocation and a contract

The fact that there are two types of unknown information, namely, technology and effort, makes it difficult to obtain a contract that maximizes the contractee's utility. Therefore, as an approximation method, we develop a mechanism that first obtains an allocation that maximizes social surplus in terms of declared performance profiles and then calculates contracts based on the allocation obtained in the first step.

The procedure of our mechanism is as follows.

1. The contractee announces a task.
2. Each contractor reports its performance profile (which may or may not be true) to the contractee. Any reported information about other contractors remains undisclosed to the contractor.
3. The contractee finds the contractor i who reports the highest technology value.
4. The contractee calculates a contract (w_i^H, w_i^L) and offers it to contractor i.
5. Contractor i decides whether to accept the contract (w_i^H, w_i^L).
6. If contractor i rejects the contract (w_i^H, w_i^L), the task is not allocated to any other contractor.

The calculation method of the contract (w_i^H, w_i^L) in step 4 is described below.

3.3 Behavior of a contractor

In this subsection, we examine what type of contract should be offered to a contractor to induce it to select an effort level of e. To induce contractor i to select an effort level of e, the following incentive compatibility constraint must hold.

$$\begin{aligned} & p(e;\alpha_i)w_i^H + (1-p(e;\alpha_i))w_i^L - e \\ \geq \ & p(e_i;\alpha_i)w_i^H + (1-p(e_i;\alpha_i))w_i^L \\ & - e_i \quad (where\ e_i \neq e) \end{aligned}$$

This constraint means that the utility obtained by selecting an effort level of e must be greater than or equal to that obtained by selecting another effort level.

Regarding the incentive compatibility constraint, contract theory tells us that if both the monotone likelihood ratio condition (MLRC) and the convexity of distribution function condition (CDFC) hold, we can apply the first-order approach [12]. MLRC means that the greater the probability of success in solving a problem, the greater the likelihood that a higher effort level is selected. CDFC intuitively means that the contractor's marginal profit for an additional effort probabilistically decreases.

Because this paper assumes that the number of possible results of achieving a task is two, namely, success or failure, and the probability $p(e_i;\alpha_i)$ is an increasing concave function of e_i, we can conclude that the two conditions of MLRC and CDFC hold [12]. Therefore, we can apply the first-order approach. The first-order approach means that the first-order partial derivative of contractor i's utility with respect to e_i is equal to 0. That is, the following expression must hold.

$$p'(\alpha_i, e_i)(w_i^H - w_i^L) = 1 \tag{1}$$

Next, we examine the participation constraint. The participation constraint means that no contractor suffers any loss by signing a contract. Note that we discuss this constraint in terms of the expected utility. If the participation constraint does not hold, contractors are not willing to sign a contract. Here, we assume that if a contractor is not allocated any tasks, its utility is equal to 0. Therefore, the participation constraint is represented as follows.

$$p(e_i;\alpha_i)w_i^H + (1-p(e_i;\alpha_i))w_i^L - e_i \geq 0 \tag{2}$$

From the contractee's perspective, it is sufficient to give the contractor the minimum amount of reward that the contractor is willing to enter in the contract. Therefore, we deal with the participation constraint as follows.

$$p(e_i;\alpha_i)w_i^H + (1-p(e_i;\alpha_i))w_i^L - e_i = 0 \tag{3}$$

From conditions (1) and (3), the amounts of rewards are calculated as follows.

$$w_i^H = e_i + \frac{1-p(e_i;\alpha_i)}{p'(e_i;\alpha_i)} \tag{4}$$

$$w_i^L = e_i - \frac{p(e_i;\alpha_i)}{p'(e_i;\alpha_i)} \tag{5}$$

The reward for failure may become less than zero, although the expected utility of the contractor never becomes negative.

Note that the method of selling the store, namely, selling the problem and the right to take all the profits from obtaining a solution to contractors at a flat fee, cannot be applied because a solution itself does not benefit the contractors [11].

3.4 Behavior of a contractee

In the previous subsection, we obtained a contract (w_i^H, w_i^L) that induces contractor i to select an effort level of e. Based on this result, here we examine what effort level needs to be set to maximize the contractee's utility.

The objective function of the contractee is given as follows.

$$\max_{e_i}(p(e_i;\alpha_i)v - (p(e_i;\alpha_i)w_i^H + (1-p(e_i;\alpha_i))w_i^L))$$

By substituting expressions (4) and (5) for the above expression, the best contract for the contractee is obtained by calculating an e_i that satisfies the following expression.

$$p'(e_i;\alpha_i) = 1/v$$

Here, let e_i^* denote the e_i that satisfies this expression. We will use e_i^* in the next subsection.

3.5 Properties of the mechanism

In the discussion in the previous two subsections, we assumed that the declared performance profile is used for calculating the reward amounts. However, contractor i may report a false performance profile, actually, contractor i can obtain additional profit by understating the value of α_i. If contractors declare false values and there is no equilibrium, we cannot predict which allocation is realized and how much profit the contractee makes. Therefore, to induce a contractor's truth declaration, we use the performance profile of the second highest declared value, α_j $(j \neq i)$, instead of the performance profile of the highest declared value of α_i. We designate the declared value of α_i as $\tilde{\alpha}_i$ and the second highest declared value of α_j $(j \neq i)$ as $\tilde{\alpha}^{(2)}$.

We insert the following step into the procedure of the mechanism proposed above between steps 3 and 4.

3.5 Set the value of α_i to $\tilde{\alpha}^{(2)}$.

By using the second highest declared value, we give up the notion of maximizing the contractee's utility. However, we believe that calculating a contract based on a socially efficient allocation in terms of technology is an appropriate method of approximation. This can be supported by the experimental results in Section 3.6.

In this case, from expressions (1), (4), (5), contractor i selects an effort level of e_i that satisfies the following expression.

$$p'(e_i;\alpha_i)\frac{1}{p'(e_i^*;\tilde{\alpha}^{(2)})} = 1 \tag{6}$$

Therefore, it is no longer guaranteed that contractor i selects e_i^*. However, the participation constraint still holds.

PROPOSITION 1. *Even if we calculate a contract by using the performance profile of $\tilde{\alpha}^{(2)}$ as the performance profile of α_i, the participation constraint of contractor i still holds if contractor i does not overstate its performance profile of α_i.*

PROOF. Let e_i^{**} denote the value of e_i that satisfies expression (6). Here, $U_i(e_i^{**}) \geq U_i(e_i^*)$ holds. If contractor i selects an effort level of e_i^*, expected utility, $U_i(e_i^*)$, is calculated as follows.

$$\begin{aligned}
&p(e_i^*;\alpha_i)w_i^H + (1-p(e_i^*;\alpha_i))w_i^L - e_i^* \\
&= p(e_i^*;\alpha_i)(e_i^* + \frac{1-p(e_i^*;\tilde{\alpha}^{(2)})}{p'(e_i^*;\tilde{\alpha}^{(2)})}) \\
&\quad + (1-p(e_i^*;\alpha_i))(e_i^* - \frac{p(e_i^*;\tilde{\alpha}^{(2)})}{p'(e_i^*;\tilde{\alpha}^{(2)})}) - e_i^* \\
&= \frac{1}{p'(e_i^*;\tilde{\alpha}^{(2)})}(p(e_i^*;\alpha_i) - p(e_i^*;\tilde{\alpha}^{(2)}))
\end{aligned}$$

Because $\alpha_i \geq \tilde{\alpha}^{(2)}$ holds, $p(e_i^*;\alpha_i) > p(e_i^*;\tilde{\alpha}^{(2)})$ holds. Therefore, we find that $U_i(e_i^*) > 0$, and thus the participation constraint of contractor i is satisfied. □

PROPOSITION 2. *In the proposed mechanism, it is best for contractor i to report its true performance profile of α_i.*

PROOF. First, we examine the case where contractor i wins the auction if it declares the true performance profile. Contractor i cannot manipulate the performance profile of $\tilde{\alpha}^{(2)}$ because it is a performance profile reported by another contractor. Therefore, even if contractor i understates the performance profile of α_i, as long as it is the winner of the auction, the performance profile of $\tilde{\alpha}^{(2)}$ does not change. Consequently, the expected utility of contractor i does not change. If contractor i overstates the performance profile of α_i, it cannot obtain any additional utility because the performance profile of $\tilde{\alpha}^{(2)}$ does not change.

Next, we examine the case where contractor i loses the auction if it declares the true performance profile. If the contractor overstates the performance profile of α_i and becomes the winner of the auction, $\alpha_i < \tilde{\alpha}^{(2)}$ holds.

Let e_i^{**} denote the value of e_i that satisfies expression (6). Contractor i's expected utility U_i takes the maximum value at an effort level of e_i^{**}. $U_i(e_i^{**})$ is calculated as follows.

$$\begin{aligned}
&p(e_i^{**};\alpha_i)w_i^H + (1-p(e_i^{**};\alpha_i))w_i^L - e_i^{**} \\
&= e_i^* - e_i^{**} + \frac{1}{p'(e_i^*;\tilde{\alpha}^{(2)})}(p(e_i^{**};\alpha_i) - p(e_i^*;\tilde{\alpha}^{(2)}))
\end{aligned}$$

First, we examine the case where $e_i^{**} < e_i^*$. From the assumption on the probability $p(e_i;\alpha_i)$, the following inequalities hold.

$$p'(e_i^*;\tilde{\alpha}^{(2)}) < \frac{p(e_i^*;\tilde{\alpha}^{(2)}) - p(e_i^{**};\tilde{\alpha}^{(2)})}{e_i^* - e_i^{**}} < p'(e_i^{**};\tilde{\alpha}^{(2)})$$

By transforming this expression, the following inequality is obtained.

$$\begin{aligned}
e_i^* - e_i^{**} &< \frac{p(e_i^*;\tilde{\alpha}^{(2)}) - p(e_i^{**};\tilde{\alpha}^{(2)})}{p'(e_i^*;\tilde{\alpha}^{(2)})} \\
&< \frac{p(e_i^*;\tilde{\alpha}^{(2)}) - p(e_i^{**};\alpha_i)}{p'(e_i^*;\tilde{\alpha}^{(2)})}
\end{aligned}$$

By substituting this inequality for the expression of the expected utility, we can obtain the following inequality.

$$U_i(e_i^{**}) \quad < \quad e_i^* - e_i^{**} + e_i^{**} - e_i^* = 0$$

Thus, the expected utility of $U_i(e_i^{**})$ becomes less than zero.

Second, we examine the case where $e_i^{**} > e_i^*$. From the assumption on the probability $p(e_i;\alpha_i)$, the following inequalities hold.

$$p'(e_i^{**};\tilde{\alpha}^{(2)}) < \frac{p(e_i^{**};\tilde{\alpha}^{(2)}) - p(e_i^*;\tilde{\alpha}^{(2)})}{e_i^{**} - e_i^*} < p'(e_i^*;\tilde{\alpha}^{(2)})$$

By transforming this expression, the following inequality is obtained.

$$\begin{aligned}
e_i^{**} - e_i^* &> \frac{p(e_i^{**};\tilde{\alpha}^{(2)}) - p(e_i^*;\tilde{\alpha}^{(2)})}{p'(e_i^*;\tilde{\alpha}^{(2)})} \\
&> \frac{p(e_i^{**};\alpha_i) - p(e_i^*;\tilde{\alpha}^{(2)})}{p'(e_i^*;\tilde{\alpha}^{(2)})}
\end{aligned}$$

By substituting this inequality for the expression of the expected utility, we can obtain the following inequality.

$$U_i(e_i^{**}) \quad < \quad e_i^* - e_i^{**} + e_i^{**} - e_i^* = 0$$

Thus, the expected utility of $U_i(e_i^{**})$ becomes less than zero.

Lastly, we examine the case where $e_i^{**} = e_i^*$. In this case, the expected utility of $U_i(e_i^{**})$ becomes less than zero because $p(e_i^{**};\alpha_i) < p(e_i^*;\tilde{\alpha}^{(2)})$.

That is, contractor i cannot obtain positive utility by overstating its performance profile of α_i. □

This is a desirable property of the mechanism because contractors do not have to consider what performance profiles they should submit and the existence of an equilibrium in this context means that the contractee is guaranteed to obtain an efficient task allocation.

3.6 Evaluations

We evaluated to what extent the proposed mechanism can increase the contractee's utility. We compared our mechanism with two other mechanisms: ideal and fixed-quality cases. In the mechanism labeled ideal, the contractee calculates the contract under the assumption that the contractee knows the true performance profile of each contractor. On the other hand, in a mechanism labeled fixed-quality, under the assumption that a contractee can monitor how many resources the contractor assigns for solving the allocated problem, a contractee first determines the problem-solving quality level, then determines a task allocation by using a Vickrey auction but does not change the problem-solving quality level based on the auction result. In this auction, the lowest bidder wins the auction and receives an amount of money equal to the second lowest bid if the problem solving succeeds and receives nothing if the problem solving fails.

We assume that each contractor's performance profile takes the following form.

$$p(e_i;\alpha_i) = \sqrt{\alpha_i e_i}$$

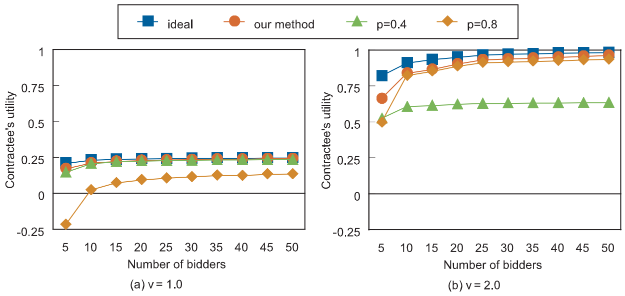

Figure 1: Experimental results (easy task)

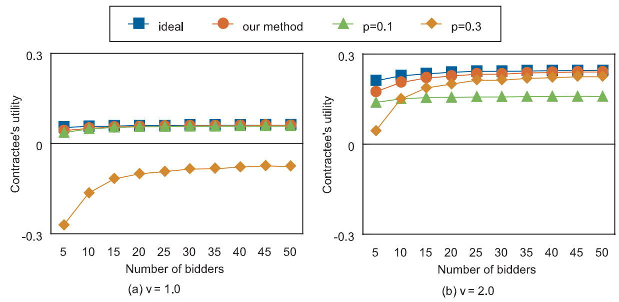

Figure 2: Experimental results (difficult task)

We examine two cases: an easy task and a difficult task. In an easy task, the value of α_i of each contractor is drawn from a uniform distribution over $[0.0, 1.0]$, while in a difficult task, the α_i value of each contractor is drawn from a uniform distribution over $[0, 0, 0.25]$. The contractee's valuation value for obtaining a solution, v, is set to 1.0 and 2.0. In addition, we set the problem-solving quality level p to 0.4 and 0.8 for an easy task and 0.1 and 0.3 for a difficult task in the fixed-quality mechanism. In each case, we generated 100 instances and calculated the average values of the contractee's utility.

Figures 1 and 2 show the results of the contractee's utility in an easy task and a difficult task, respectively. The x-axis represents the number of bidders (contractors), while the y-axis represents the contractee's utility.

These figures elucidate the following.

- As the number of bidders increases, the performance of the proposed mechanism becomes closer to that of the ideal mechanism.
- The performance of the fixed-quality of 0.4 is better than that of the fixed-quality of 0.8 in Figure 1(a), while the performance of the fixed-quality of 0.4 is worse than that of the fixed-quality of 0.8 in Figure 1(b). That is, an appropriate problem-solving quality level for one problem is not appropriate for another. Moreover, as Figure 1(a) shows, the contractee suffers a loss if the contractee uses the fixed-quality mechanism and fails to choose an appropriate problem-solving quality level. The same assertion holds in Figure 2.
- The proposed mechanism can increase the contractee's utility more effectively than the fixed-quality mechanism. If the contractee can find an appropriate problem-solving quality level, the fixed-quality mechanism seems to perform as well as our mechanism. However, note that the fixed-quality mechanism assumes that the contractee can monitor the contractor's behavior. This assumption is often difficult to satisfy.

Although this evaluation is a preliminary one, the obtained results show that our mechanism is a promising way to increase the contractee's utility.

4. MULTIPLE-TASK CASES

So far, we have restricted our discussions to the single-task case. However, a contractee's problem may be divided into several subproblems. In this case, the contractee may be able to increase its utility by allocating these subproblems to multiple contractors. In this paper, we assume the following.

Assumption 11. Each subproblem is identical to others in terms of the difficulty of problem solving, and can be solved independently of others; namely, a contractor does not need to communicate with other contractors to solve the allocated problem.

Moreover, we assume the following.

Assumption 12. Marginal cost with respect to an additional unit increases for each contractor.

Our mechanism's procedure is as follows. Here, we assume that m units exist.

1. The contractee announces m tasks.
2. Each contractor reports its performance profiles (which may or may not be true) to the contractee. More specifically, each contractor is allowed to report its performance profile for the first task, that for the second task, etc. Any reported information of other contractors remains undisclosed to the contractor.
3. The contractee finds the m contractors whose performance profile of α_i is from the highest to the m-th highest. Here, we allow the contractee to allocate more than one task to the same contractor.
4. For each winner, if the winner i who has technology values of $\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{im_i}$ wins m_i units, the contractee finds the m_i highest rejected bids, $\tilde{\alpha}_1^{(2)}, \tilde{\alpha}_2^{(2)}, \cdots, \tilde{\alpha}_{m_i}^{(2)}$. Set the value of α_{i1} to $\tilde{\alpha}_1^{(2)}$, α_{i2} to $\tilde{\alpha}_2^{(2)}$, $\cdots$, and α_{im_i} to $\tilde{\alpha}_{m_i}^{(2)}$.
5. The contractee calculates contracts (w_i^H, w_i^L) and offers them to contractor i.
6. Contractor i decides whether to accept the contract (w_i^H, w_i^L).
7. If contractor i rejects the contract (w_i^H, w_i^L), the task is not allocated to any contractor.

The calculation method of the contract (w_i^H, w_i^L) is the same in the previous section.

In this case, the participation constraint holds.

PROPOSITION 3. *Even if we calculate a contract by using the performance profile of* $\tilde{\alpha}_l^{(2)}$ *as the performance profile of* α_{il}*, the participation constraint of contractor* i *still holds.*

Because this proposition can be proved in a similar way to the proof of Proposition 1, we omit the proof.

PROPOSITION 4. *In the proposed mechanism, it is best for contractor i to report its true performance profiles of $\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{im_i}$, if the marginal cost with respect to an additional unit increases for each contractor.*

To prove this proposition, we use the following three lemmas.

LEMMA 5. *In the proposed mechanism, contractor i cannot obtain an additional unit of utility by understating its performance profile of α_{il} if the allocation does not change.*

Because this lemma can be proved in a similar way to the proof of Proposition 2, we omit the proof.

LEMMA 6. *In the proposed mechanism, contractor i cannot obtain an additional unit of utility by overstating its performance profile of α_{il}.*

Because this lemma can be proved in a similar way to the proof of Proposition 2, we omit the proof.

Lastly, we examine the possibility of a demand-reduction lie [1]. The following is an example of a demand-reduction lie. Suppose that there are two tasks and two contractors 1 and 2 demanding one task and two tasks, respectively. If the rewards are determined based on each contractor's declaration, there might be the following situation. If contractor 2 insists on winning two tasks, its reward is reduced due to the competition between contractors 1 and 2, whereas if contractor 2 reduces its demand from two tasks to one task, contractor 2 wins one task without competition, which leads to increasing its utility. That is, truth telling is no longer an equilibrium strategy.

This problem can be solved if we calculate rewards based on the m_i highest rejected bids instead of the highest rejected bid. This is because contractor i cannot manipulate the m_i highest rejected bids. A formal explanation is given below.

LEMMA 7. *In the proposed mechanism, contractor i cannot obtain an additional unit of utility by reducing demand.*

PROOF. Suppose that contractor 1 has technology values of α_{11} and α_{12}, contractor 2 has a technology value of α_{21}, contractor 3 has a technology value of α_{31}, and $\alpha_{11} > \alpha_{12} > \alpha_{21} > \alpha_{31}$ hold. In addition, suppose that two units exist. If contractor 1 reports the true values of its technology, contractor 1 wins two units. The contracts are calculated based on α_{21} and α_{31}. On the other hand, if contractor 1 reports only α_{11}, contractor 1 wins one unit. The contract in this case is calculated based on α_{31}. In the former case, contractor 1 can choose which contracts, a contract based on α_{21} or a contract based on α_{31}, are applied to the first unit. The remaining contract is applied to the second unit. Thus, contractor 1's utility is greater than or equal to the sum of utility from the contract based on α_{31} for the first unit and utility from the contract based on α_{21} for the second unit. It is obvious that the amount of this utility is greater than that in the demand-reduction case. Other cases can be discussed in a similar way. □

From the above three lemmas, we can conclude that Proposition 4 holds.

5. DISCUSSIONS

In this section, we describe limitations of our framework and the proposed mechanism. First, under our framework a contractee gives its problem itself to contractors. However, computing power trading can be implemented in various ways. For example, a contractor creates an account for the contractee on the contractor's computer, enabling the contractee to obtain access to the contractor's computer and execute codes on it. Another method of implementation is one in which a contractee writes an executable code, then a contractor executes the code on the contractor's computer.

However, the former case increases the management cost of user accounts. Moreover, if executing the codes needs proprietary software, it may cause some trouble with software licenses. In addition, in the latter case, the contractor may not be able to know in advance how many resources are required to execute the code. Furthermore, executing the code written by other parties may cause the contractor trouble in a similar way as a computer virus does.

Therefore, this paper focused on the following scenario in which a contractee gives its problem to a contractor, after which the contractor solves the problem and returns the obtained solution to the contractee. This framework, however, may have a problem of the contractee's trade secrets leaking out.

However, there are some fields, such as scientific research, where such a problem is not serious. Instead of revealing the contractee's problem to the contractor, our framework enables the contractee to choose an appropriate problem solver, including algorithms/heuristics, by placing contractors in a competitive situation. This enables the contractee to avoid worrying which software products to buy.

Next, as mentioned above, the problem must be one for which the contractee can verify the validity of the reported solution at a negligible cost. This assumption is made to exclude the possibility that the contractor invents a false solution without making any effort and reports it to the contractee. Thus, the proposed mechanism cannot deal with an optimization problem. However, we may relax this constraint by employing the following method. The contractee allocates the same problem to two contractors and compares the reported solutions to each other. This prevents the contractors from reporting a false solution, although the contractee's utility decreases because it must make two payments. Our future work will include a detailed analysis of this method.

We have already mentioned the availability of the performance profile in Section 2. In our framework, the more accurate the performance profiles a contractor has, the more its utility increases. Therefore, the contractor has an incentive to obtain accurate performance profiles.

One of the assumptions made about contractors' performance profiles is concavity. If this assumption does not hold, obtaining an appropriate contract becomes considerably difficult. Examining whether the concavity assumption holds in the real problems will be included in our future work.

Another assumption made about contractors' performance profiles is that different contractors' performance profiles do not intersect with each other. However, this might be violated in a real-world situation. A search algorithm could be more speculative at the beginning, trying some pre-processing steps that gives it a relatively high chance of succeeding early but do not help the rest of the search if no solution is found in this time. This would yield a curve that would eventually sink below that of a more conservative and systematic algorithm.

If this assumption does not hold, we can no longer determine a sole winner. If we try to choose the most efficient one among several candidates based on the declared performance profiles, it is no longer guaranteed that each contractor reveals its true information, which results in a failure of the mechanism. A solution is to choose one at random, although it may reduce the contractee's utility.

Finally, the reward for failure may become less than zero, although the expected utility of a contractor never becomes less than zero. In the Internet environment, if the contractor is forced to receive negative reward, namely, pay some penalty, the contractor may disappear without paying any penalty [6]. In such cases, we have to devise some exchange mechanisms, one of which is to pool rewards at a trusted third party.

6. CONCLUDING REMARKS

This paper developed a task allocation mechanism for cases in which the task is to find a solution to a search problem or in a constraint satisfaction problem. In this situation, a contractee has to find an appropriate level of quality for task achievement as well as to efficiently allocate a task among contractors. However, it is difficult to solve this problem because the contractee cannot ascertain the contractor's problem-solving ability, such as the amount of available resources and knowledge, nor monitor how many resources are actually assigned to the allocated task.

To solve this problem, we have developed a new mechanism that is able to choose an appropriate quality level for task achievement. More specifically, the contractee first announces a task, then contractors submit their performance profiles that show the relation between the amount of invested resources and the probability of finding a solution. The contractee then determines a winner and calculates a set of rewards based on the declared performance profiles. By analyzing the mechanism through game theory, we showed that this mechanism guarantees that each contractor reveals its true information, which contributes to stabilizing the whole system. Experimental results showed that our mechanism can increase the contractee's utility compared to a simple auction mechanism. Moreover, we proposed an extended mechanism for multiple-task cases. Although this paper focused on contracting out a search or CSP problem, the problem considered here is a significant generalization of a mechanism design problem and it can be applied to a range of different tasks.

Finally, we assumed that a contractor does not need any communication with other contractors to solve the allocated problem in multi-task cases. However, the contractee may need to coordinate multiple contractors to solve a large-scale problem. In such a case, the result of a task achievement by a contractor may depend on the result of a task achievement by another contractor. Developing mechanisms for such cases is one of our future projects.

7. ACKNOWLEDGMENTS

The author wishes to thank anonymous reviewers for their comments, and Hal Varian, Makoto Yokoo and Yuko Sakurai for their helpful discussions.

8. REFERENCES

[1] L. M. Ausubel and P. Cramton. Demand reduction and inefficiency in multi-unit auctions. Working Paper, 1998.

[2] Y. Fujishima, D. McAdams, and Y. Shoham. Speeding up ascending-bid auctions. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 554–559, 1999.

[3] T. Hogg and C. P. Williams. The hardest constraint problems: A double phase transition. *Artificial Intelligence Journal*, 69:359–377, 1994.

[4] J.-J. Laffont and J. Tirole. Auctioning incentive contracts. *Journal of Political Economy*, 95(5):921–937, 1987.

[5] K. Larson and T. Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence Journal*, 132(2):183–217, 2001.

[6] S. Matsubara and M. Yokoo. Defection-free exchange mechanisms based on an entry fee imposition. *Artificial Intelligence Journal*, 142(2):265–286, 2002.

[7] R. P. McAfee and J. McMilan. Bidding for contracts: a principal-agent analysis. *Rand Journal of Economics*, 17(3):326–338, 1986.

[8] D. G. Mitchell, B. Selman, and H. J. Levesque. Hard and easy distributions of SAT problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 459–465, 1992.

[9] D. C. Parkes, J. Kalagnanam, and M. Eso. Achieving budget-balance with Vickrey-based payment schemes in exchanges. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 1161–1168, 2001.

[10] R. Porter, A. Ronen, Y. Shoham, and M. Tennenholtz. Mechanism design with execution uncertainty. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI-02)*, 2002.

[11] E. Rasmusen. *Games and Information: an introduction to game theory*. Blackwell Publishers, 3rd edition, 2001.

[12] B. Salanié. *The Economics of Contracts*. MIT Press, 1997.

[13] T. W. Sandholm. Limitation of the Vickrey auction in computational multiagent systems. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96)*, pages 299–306, 1996.

[14] D. E. M. Sappington. Incentives in principal-agent relationships. *Journal of Economic Perspectives*, 5(2):45–66, 1991.

[15] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behavior*, 35(1):271–303, 2001.

[16] M. Yokoo, Y. Sakurai, and S. Matsubara. Robust combinatorial auction protocol against false-name bids. *Artificial Intelligence Journal*, 130(2):167–181, 2001.