

# Task Allocation Strategy Based on Variances in Bids for Large-Scale Multi-Agent Systems

Toshiharu Sugawara

► **To cite this version:**

Toshiharu Sugawara. Task Allocation Strategy Based on Variances in Bids for Large-Scale Multi-Agent Systems. Harris Papadopoulos; Andreas S. Andreou; Lazaros Iliadis; Ilias Maglogiannis. 9th Artificial Intelligence Applications and Innovations (AIAI), Sep 2013, Paphos, Greece. Springer, IFIP Advances in Information and Communication Technology, AICT-412, pp.110-120, 2013, Artificial Intelligence Applications and Innovations. <10.1007/978-3-642-41142-7\_12>. <hal-01459603>

**HAL Id: hal-01459603**

**<https://hal.inria.fr/hal-01459603>**

Submitted on 7 Feb 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Task Allocation Strategy Based on Variances in Bids for Large-Scale Multi-Agent Systems

Toshiharu Sugawara

Department of Computer Science and Engineering  
Waseda University, Tokyo 1698555, Japan

**Abstract.** We propose a decentralized task allocation strategy by estimating the states of task loads in market-like negotiations based on an announcement-bid-award mechanism, such as contract net protocol (CNP), for an environment of large-scale multi-agent systems (LSMAS). CNP and their extensions are widely used in actual systems, but their characteristics in busy LSMAS are not well understood and thus we cannot use them lightly in larger application systems. We propose an award strategy in this paper that allows multiple bids by contractors but reduces the chances of simultaneous multiple awards to low-performance agents because this significantly degrades performance. We experimentally found that it could considerably improve overall efficiency.

## 1 Introduction

Recent technology in the Internet and ambient intelligence has enabled services that are incorporated into daily activities. These services consist of a number of tasks each of which has required capabilities and data, and should be done in agents meeting these requirements. Of course, vast numbers of services are required everywhere in the environment; thus, they are provided in a timely manner by a large number of cooperative agents on computers and sensors through mutual negotiations to appropriately allocate associated tasks among agents. It is, however, unrealistic for agents to know all the capabilities and data other agents have in open environments. Well-known and simple methods of allocating tasks without this information are market-like negotiation protocols, like the contract net protocol (CNP) [7].

However, its characteristics in a large-scale multi-agent system (LSMAS) is poorly understood. Understanding the efficiency of CNP in LSMAS is not easy because if many manager agents simultaneously try to allocate tasks, interference among agents occurs. For example, multiple simultaneous bids are not allowed in a naive CNP. However, this restriction may considerably degrade the efficiency in a busy LSMAS, because bid/allocation processes occur almost sequentially. We can allow multiple bids to enhance concurrency and actually in [7] it is also discussed the possibility of multiple bids. However, allowing multiple bids will result in simultaneous multiple awards begin given to contractors. We can consider two solutions to this situation. First, extra awarded tasks are canceled with certain penalties (e.g. [5]). However, we believe that this makes the task allocation protocol so complicated that it may degrade the overall efficiency of a busy LSMAS. The second approach is to execute all awarded tasks in the

contracted agents. This is simpler, but awardees cannot complete the tasks within the contracted times, resulting in significant delays in executing tasks in busy environments.

There have been a number of studies on applying it to LSMASs by assuming multiple initiations and bids in CNP [1, 6, 8]. For example, in [6] the authors addressed the issue of the eager-bidder problem occurring in a LSMAS, where a number of tasks were announced concurrently so that a CNP with certain levels of commitment did not work well. They also proposed a CNP extension based on statistical risk management. However, the types of resources and tasks considered in these papers were quite different, i.e., they focused on exclusive resources such as airplane seats. We focus on divisible resources such as CPUs or network bandwidth, so any number of tasks can be accepted simultaneously but with reduced quality. In [2] it was also discussed the extension of CNP to allow multiple initiations and bids based on colored Petri nets. There have been other extensions that have introduced brokers [4] and personal agents [3]. However, these studies have focused on reducing the communication overhead in LSMAS. In [8] it was proposed an award strategy in which two different strategies were alternatively selected by statistically analysing bid values to improve overall efficiency. However, as their method could only be applied to a set of tasks whose costs were in a certain limited range, its applicability was severely restricted.

We adopt a simpler approach for a busy LSMAS in which all awarded tasks are executed in the awardees. Instead, we reduce the probability of multiple awards given to lower-performance agents, by extending the method in [8], because multiple awards to these types of agents are the major reason for degraded performance. The negative effect of multiple awards on a system's efficiency is strongly affected by the busyness or task loads of LSMAS. If the environment has a low load, hardly any multiple awards occurs; thus, managers can award tasks to the best bidders without having to make careful selections. However, the chance of multiple awards also increases according to the increase in task loads and thus the simple strategy of making awards in which a manager selects the best bidder becomes inappropriate. Lower-performing agents may be identified as the best bidders by multiple managers, particularly in busier situations, and thus be awarded multiple tasks simultaneously; this will considerably reduce performance. We tried to reduce the probability of multiple awards in our approach when the chance of bids going to lower performing agents had increased. Managers have to select the awardees by carefully understanding the states of their local environment to achieve this.

This paper is organized as follows. Section 2 describes the model and raises issues addressed in this paper. We then compare a naive award strategy with a probabilistic award in which non-best agents are selected to allocate tasks. We then propose a method with which managers decide the award strategies based on the statistical analysis only involving light computation. The experimental results revealed that the proposed method could perform well in a busy environment.

## 2 Problem Description

### 2.1 Overview and Issues

We define a simple extension of CNP in which simultaneous multiple initiations by different managers and multiple bids to different managers are allowed. First, let  $\mathcal{A} = \{1, \dots, n\}$  be a set of agents and  $t$  be a task.  $\mathcal{A}$  is the disjoint union of  $\mathcal{M} = \{m_1, \dots, m_{N_m}\}$ , which is the set of managers who allocate tasks, and  $\mathcal{C} = \{c_1, \dots, c_{N_c}\}$ , which is the set of contractors who execute the allocated tasks. Contracts have different capabilities which affect the time required to complete the assigned tasks. Let us assume that  $|\mathcal{A}|$  is large (on the order of thousands) and the agents are widely distributed, like servers on the Internet. This also means that communications between agents incur some delay. For manager  $m \in \mathcal{M}$ , we define the concept of the *scope*,  $S_m$ , which is the set of agents that  $m$  knows, and  $m$  can announce tasks to all or a part of  $S_m$ .

We have to consider a number of issues to avoid inefficiency due to multiple awards. First, task load affects the possibility of multiple awards. When an agent that has outstanding bids receives another task announcement, it will send its bid (multiple bids) and thus may receive multiple award messages. These multiple bids occur at quite a low rate if they are not busy. Even if they occur, multiple tasks are likely to be allocated to relatively efficient agents since efficient contractors are identified as the best bidders by managers, so no significant degradation to performance will occur. However, as the environment becomes busier, the chance of multiple awards being made to low-performing agents increases and overall performance decreases. Therefore, we propose an award strategy that selects awardees to reduce the probability of multiple awards being made to lower-performing agents accomplished by identifying their task load.

Second, longer communication delay (or latency) also increases the chances of multiple awards. However, we will not discuss reduced latency, because this strongly depends on the environmental settings and agents cannot directly handle it. Instead, we focus on the managers' decisions about awardees by identifying situations in which multiple awards to low-performing agents are likely to occur, regardless of high or low latency environments.

Agents have a number of decision-making strategies that may affect overall performance besides the award strategy. First, manager  $m \in \mathcal{M}$  having a task announces a call-for-bid message to contractors that are selected from its scope,  $S_m$ , on the basis of an *announcement strategy*. Then, agents decide whether they should act on the received call-for-bid messages or not using a *bid strategy*. Here, we assume simple announcement and bid strategies, i.e., an announcement strategy is where  $m$  selects all or  $N$  agents from  $S_m$  randomly, where  $N$  is a positive integer, and the bid strategy is where an agent always bids for call-for-bid messages if it can execute associated tasks. These strategies also play important roles in improving overall efficiency, such as restricting the numbers of multiple bids and announcement messages. However, we particularly focused on award strategies since we were concerned about the negotiations for task allocations to prevent multiple awards being made by appropriately selecting the awardees according to the received bid messages.

Because naive CNP involves a timeout basis to receive response messages from others, it is inefficient in large and busy environments. Thus, we made use of *regret*

messages, which were sent in the award phase to contractors who were not awarded the contract, to prevent long waits for bids and award messages (as in [9]).

## 2.2 Model of Agents and Tasks

For task  $t$ , let  $r_t = \{r_t^1, \dots, r_t^d\}$  be the set of required resources (or functions) to execute this, where  $r_t^k \geq 0$  is integer. Agent  $i$  is denoted by tuple,  $(\alpha_i, loc_i, S_i, Q_i)$ , where  $\alpha_i = (a_i^1, \dots, a_i^d)$  is the agent's capabilities, and  $a_i^h$  corresponds to the  $h$ -th resource and  $a_i^h \geq 0$ ;  $a_i^h = 0$  indicates agent  $i$  does not have the  $h$ -th resource. Element,  $loc_i$  is the location of  $i$ , and  $Q_i$  is a finite queue where the tasks allocated to  $i$  are temporarily stored. The set,  $S_i (\subseteq \mathcal{A})$ , is  $i$ 's scope.

We assume a discrete time for any time descriptions. A unit of time is called a *tick*. The execution time of  $t$  by  $i$  is:

$$\gamma_i(t) = \max_{1 \leq h \leq d} \lceil r_t^h / a_i^h \rceil, \quad (1)$$

where  $\lceil x \rceil$  denotes the ceiling function. The *metric* between the agents,  $\delta(i, j)$ , is based on their locations,  $loc_i$  and  $loc_j$ . It is only used to calculate communication delay; the time required for message transfer between agents  $i$  and  $j$  is  $\lceil \delta(i, j) / D \rceil$ , where  $D$  is the *speed factor* for passing messages. Parameter  $L (\geq 0)$  is called the *task load*, meaning that  $L$  tasks on average are generated according to a Poisson distribution every tick. These are then randomly assigned to different managers.

## 2.3 Task Assignment

When manager  $m$  receives a task,  $\tilde{t}$ , it immediately initiates the CNP modified for LSMAS: It first sends announcement messages to the contractors selected from its scope in accordance with the announcement strategy. Each of these contractors sends back a bid message with a certain *bid value*. The bid values might include parameters such as the price for executing the task, the quality of the result, or a combination of these values. We assume that their bid values contain the estimated times for completing  $\tilde{t}$ , because we are concerned with the efficiency of processing in LSMAS. This time is calculated as follows in contractor  $c$ :

$$\gamma_c(\tilde{t}) + \sum_{t \in Q_c} \gamma_c(t) + \beta, \quad (2)$$

where  $\beta$  is the execution time required for the task currently being executed. For multiple bids,  $c$  might have a number of outstanding bids. These bids are not considered to calculate the estimated required time because it is uncertain whether they will be accepted<sup>1</sup>. Then,  $m$  selects an awardee, on the basis of the *award strategy* and sends it an award message with the announced task,  $\tilde{t}$ .

When contractor  $c$  is awarded a task, it immediately executes it if it has no other tasks. If  $c$  is already executing a task, the new task is stored in  $Q_c$ , and the tasks in  $Q_c$  are executed in turn.

<sup>1</sup> Actually, we also examined a case in which agents considered the expected values of execution times for outstanding bids; however no significant differences were observed.

## 2.4 Performance Measures

We assumed that manager agents could observe the *completion time* for task  $t$ , which is the elapsed time from the time the award message is sent to the time the message indicating that  $t$  has been completed is received. The completion time thus includes the communication time in both directions, the queue time, and the execution time. We evaluated overall performance using the average completion time observed in all managers; this is denoted by  $\varphi$ . A shorter average completion time is preferable.

Since the queue length of agent  $i$  is bounded, some allocated subtasks might be *dropped*. The numbers of dropped tasks are also another measure of performance. However, no dropped tasks occur unless the systems are overloaded so they can only be used in limited situations. Thus, we use them to identify upper limit of the performance for the entire system.

## 3 Awards to non-best bidders and its features

It is plausible for a manager to select the best bidder in the award phase. This strategy is called the *best awardee strategy* (BAS). However, multiple awards are likely to occur with the increase in task loads. A simple award strategy for alleviating multiple awards is to allocate some tasks to non-best contractors by introducing randomness to some degree in the award phase [8].

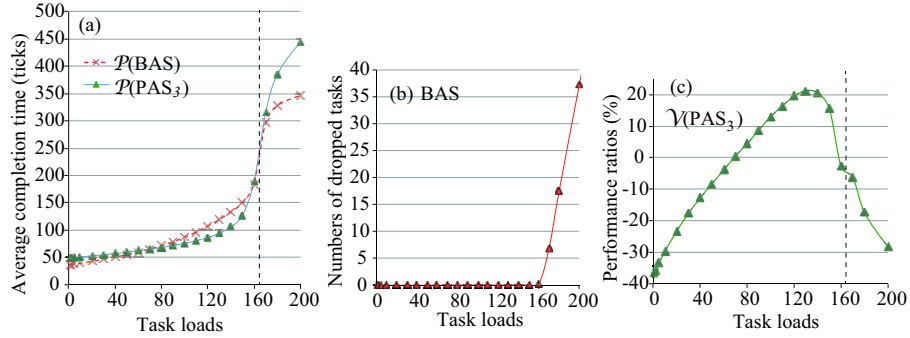
Let  $\{c_1, \dots, c_p\}$  be contractors that bid on the announced task. The estimated completion time in the bid message from  $c_i$  is denoted by  $b_{c_i}$ . The *probabilistic awardee selection strategy with the fluctuation factor*,  $f$  (which is denoted by  $PAS_f$ ) is the award strategy in which the manager selects the awardee according to the following probability:

$$\Pr(c_i) = \frac{1/(b_{c_i})^f}{\sum_{j=1}^p 1/(b_{c_j})^f}, \quad (3)$$

where non-negative integer  $f$  is the *fluctuation factor*. A smaller value for  $f$  means less randomness when awardees are selected.

We experimentally investigated the characteristics of BAS and  $PAS_f$  with various task loads. We set  $|\mathcal{C}| = 500$  and  $|\mathcal{M}| = 10,000$ . Only one type of task,  $t_{3000}$ , whose required resource  $r_{t_{3000}}$  was  $\{3000\}$  (so  $d = 1$ ) was used in this experiment (Exp. 1). For any  $c_i \in \mathcal{C}$ , a different capability was assigned to  $c_i$  so that the values of  $3000/a_{c_i}^1$  would be *uniformly distributed* over the range of 25–120; thus, the values of  $a_{c_i}^1$  ranged from 25 – 120. We assumed that the manager agents could not do the tasks themselves forcing them to assign the tasks to agents who could. The agents were randomly placed on a  $150 \times 150$  grid with a torus topology to which a Euclidean distance was introduced as its metric. Manager  $m$ 's scope,  $S_m$ , was the set of contractor agents whose distance from  $m$  was less than 10.0 because it was implausible to allocate tasks to distant agents. We also set  $f = 3$  and  $D = 2.5$ . Thus, tasks maximally took five ticks to send messages to known contractors. We defined these values by assuming that a tick was around 1 ms. The number of announcement messages,  $N$ , and the queue length,  $|Q_i|$ , were set to 20.

Figure 1 (a) plots the average completion times varying according to the values for task load,  $L$ . It indicates that average performance with BAS,  $\varphi(\text{BAS})$ , is higher than



**Fig. 1.** Results of preliminary experiment (Exp. 1).

that with  $PAS_f$ ,  $\varphi(PAS_f)$ , when task load was low or quite high but  $PAS_f$  outperformed BAS, otherwise. Note that the vertical dotted line when  $L = 165$  indicates the limit for the capability of the entire system. Actually, Fig. 1 (b), showing the numbers of dropped tasks per tick with BAS indicates that dropped tasks rapidly increased if  $L \geq 165$ .<sup>2</sup>

To clearly see the differences between their performance, we define the *performance ratio w.r.t. BAS* as:

$$\mathcal{V}(str) = \frac{\varphi(\text{BAS}) - \varphi(str)}{\varphi(\text{BAS})} \times 100, \quad (4)$$

where  $\varphi(str)$  indicates the average completion times with the award strategy specified by the variable,  $str$ . Fig. 1 (c), expressing the relationship between  $\mathcal{V}(PAS_f)$  and the task load, indicates that when  $L$  was low ( $L \leq 70$ ),  $\mathcal{V}(PAS_f)$  was minus and performance with  $PAS_f$  was maximally 40% lower than that with BAS. This tendency was also observed when the system was beyond the upper limit of all agents' capabilities ( $L \geq 160$ ). However, when it was moderately busy and near (but less than) the upper limit for the entire performance ( $80 \leq L \leq 160$ ), performance with  $PAS_f$  was maximally 20% higher than that with BAS. We believe that this is an important characteristic: when  $L$  is low, any simple strategy can result in acceptable performance and when it is beyond the upper limit for the system, no strategy can do all tasks within acceptable time limits. The system must exert its full capabilities, especially near its upper limit.

The variations in performance in Fig. 1 (c) were caused by the combination of the probabilities of multiple awards and awards to non-best bidders with  $PAS_f$ . When  $L$  was low, multiple awards rarely occurred. However, as non-best bidders were awarded with  $PAS_f$ ,  $\mathcal{V}(PAS_f)$  became minus. All contractors' queues, on the other hand, were almost full if  $L$  was beyond the system's limit. Thus, it was better to assign tasks to the best bidders, because (1) lower-performing agents were also so busy that there were no benefits by allowing some tasks to be shifted to these agents, and (2) there were so many tasks that the awards to non-best bidders resulted in other multiple bids in low-performing agents. In contrast, when  $L$  was moderate and less than the upper limit, busy

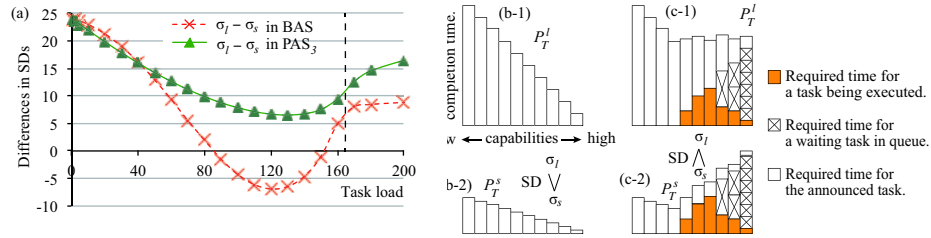
<sup>2</sup> We plotted the numbers of dropped tasks only with BAS in Fig. 1 (b), but they were almost identical to those with  $PAS_f$ .

and unbusy agents co-existed, i.e., relatively high performing agents had a number of tasks in their queues but low performing agents had few assigned tasks. Thus, managers might assign tasks to low-performing agents, resulting in making simultaneous multiple awards to them. This degraded overall performance. However, managers with  $PAS_f$  probabilistically awarded these agents to avoid multiple awards, to some degree.

## 4 Proposed Method

As  $PAS_f$  could attain approximately 20% better performance than BAS in situations near the upper limit of the system, this improvement was not small. We want to emphasize that the system must exert its potential capabilities in busy situations, especially near its upper limit of task load, but achieving this is not easy. However,  $PAS_f$  also drastically worsens performance in other situations. Thus, if managers could select an appropriate award strategy by estimating the current degrees of task loads in their local regions, they could take advantage of  $PAS_f$ .

The variance and standard deviation (SD) of bid values from local contractors can provide the information to estimate task loads in local regions. We introduced *phantom tasks* to achieve this, which are announced by managers even though no contractors are awarded to them. Phantom tasks were proposed in [8], but their use was quite restricted. We used it more actively in our proposed strategy; when a manager announces a task,  $t$ , it also announces two phantom tasks. The first one requires much smaller resources, say  $1/6$ , than those of  $t$  and this is denoted by  $p_t^s$ . The second requires identical to or slightly more resources than those of  $t$  and this is denoted by  $p_t^l$ . Then, the manager calculates the SDs of bid values for  $p_t^s$  and  $p_t^l$  received from local agents. These SDs for  $p_t^s$  and  $p_t^l$  correspond to  $\sigma_s$  and  $\sigma_l$ .



**Fig. 2.** Bid values in unbusy and moderately busy states.

We then investigated changes in the average values of  $\sigma_l - \sigma_s$  with variable task loads in an environment identical to that in Exp. 1. Since the task was  $t_{3000}$ , the resources required for  $p_{t_{3000}}^l$  and  $p_{t_{3000}}^s$  were set to  $\{3001\}$  for the former and  $\{500\}$  for the latter. The results are plotted in Fig. 2 (a).

This graph shows that  $\sigma_l > \sigma_s$  in BAS when  $L$  is low (or beyond the limit), but  $\sigma_s > \sigma_l$ , otherwise. We can explain this phenomenon as follows. When  $L$  is low, the



distribution of bid values for  $p_t^l$  and  $p_t^s$  (see Figs. 2 (b-1) and (b-2), which illustrate the distributions of bid values) directly reflects the capabilities of contractors. Thus,  $\sigma_l$ , the SD of bid values for  $p_t^l$ , is larger than  $\sigma_s$ , which is the SD of bid values for  $p_t^s$ . However, in moderately busy situations ( $L$  is between 80 and 160) there are chances for lower-performing agents to be awarded, because only higher performing agents have a number of tasks being queued and executed (Fig. 2 (c-1)). Consequently,  $\sigma_s > \sigma_l$  held in this situation (cf. Figs. 2 (c-1) and (c-2)). When the system was over the upper limit, all contractors already had a number of tasks waiting in their queues, and their estimated completion times of the tasks at the end of the queues were quite similar. Thus, the estimated completion times for the new announced tasks again reflected the performance of contractors. In contrast, managers with  $PAS_f$  allocated tasks with some fluctuations; thus,  $\sigma_l > \sigma_s$  was always positive as seen in Fig. 2(a).

From this discussion, we propose a novel award strategy in which BAS or  $PAS_f$  is selected using the phantom tasks associated with the requesting task. For task  $t$ , we generate phantom tasks,  $p_t^l$  and  $p_t^s$ , whose required resources are  $r_{p_t^l} = \{r_t^1 + \alpha, \dots, r_t^d + \alpha\}$  and  $r_{p_t^s} = \{r_t^1 / \beta, \dots, r_t^d / \beta\}$ , where  $\alpha$  is a small integer equal to or near 0 (in the experiments that follow, we set  $(\alpha, \beta) = (1, 6)$ ). Then, managers announce  $p_t^l$  and  $p_t^s$  as well as  $t$ . Because managers adopt an announcement strategy in which the announcement messages are sent to  $N$  agents randomly selected for individual scopes, these announcement messages are not sent to the same set of agents. However, we assumed that the task load states could be estimated by a survey of random sampling.<sup>3</sup> Then, contractors send back the bid messages for the received tasks. Because they cannot distinguish normal from phantom tasks, they adopt the same bid strategy in their decisions. Managers then calculate the SDs,  $\sigma_l$  and  $\sigma_s$ , and select BAS if  $\sigma_l - \sigma_s > 0$  and  $PAS_f$  if  $\sigma_l - \sigma_s \leq 0$ , as the award strategy that is used to determine the awardee for  $t$ . The proposed method of selecting the award strategy is called *award selection according to bid variances* (ASBV).

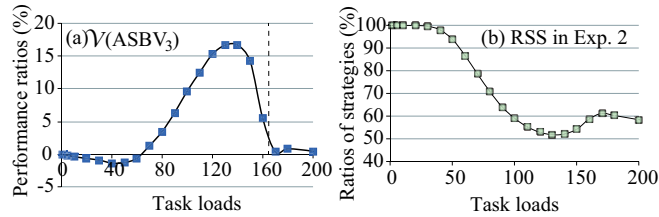
## 5 Experimental Evaluation and Remarks

We experimentally evaluated ASBV. The experimental setting for this experiment (Exp. 2) was identical to that of Exp. 1. The performance ratios w.r.t. the BAS of ASBV,  $\varphi(ASBV)$ , are plotted in Fig. 3 (a).

This graph indicates that ASBV outperformed the selected strategy ratios with BAS when  $L \geq 70$  and those with  $PAS_f$  near the upper limit for the system (around  $L = 150$  to 165). As was previously mentioned, this feature of ASBV is important for all agents in LSMAS to exert all their capabilities.

We also examined the *ratios of selected strategies* (RSS), i.e., the number of managers that selected BAS to that of managers that selected  $PAS_f$ . The results are plotted in Fig. 3 (b), which indicates that when task load was low, managers only selected BAS. Then, they increased the chance of selecting  $PAS_f$  according to the increase in the task

<sup>3</sup> Of course, we allowed multiple bids, where managers could send the same set of agents for these three tasks. The proposed method could slightly reduce the overhead for processing the messages for phantom tasks.



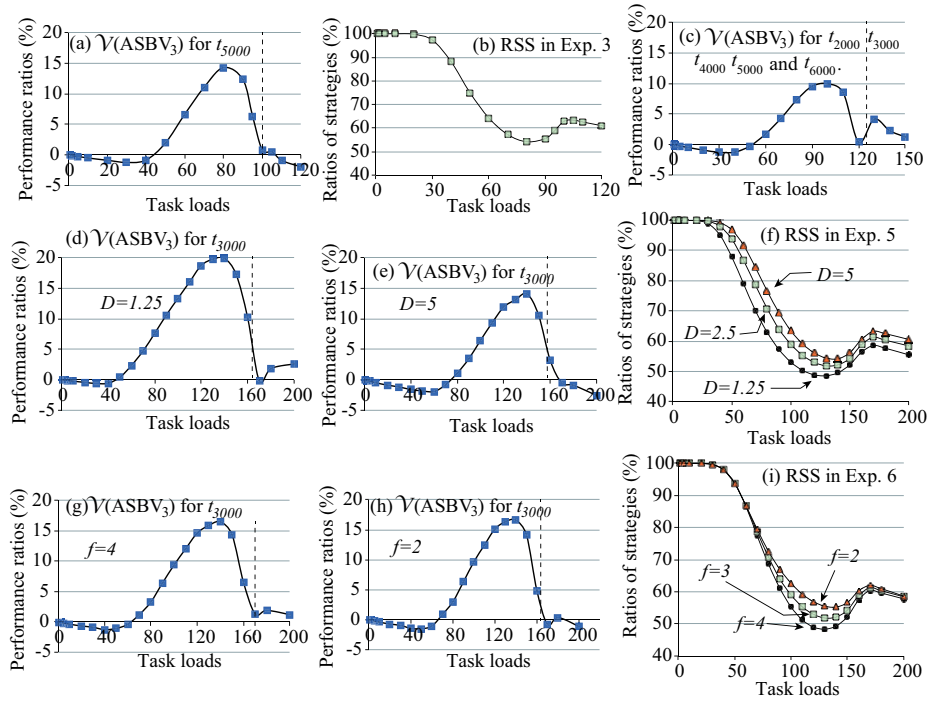
**Fig. 3.** Performance and selected strategy ratios in Exp. 2.

load. However, the selections of BAS and  $PAS_f$  in managers balanced at certain ratios to keep the values of  $\sigma_l - \sigma_s$  positive. We think that this balanced selection by local autonomous decisions by managers could improve overall performance near the upper limit for the system (around  $L = 160$ ). Fig. 3 (a) indicates that ASBV performed slightly worse when  $L$  was low although all managers adopted BAS: The manager selected the awardee in ASBV after all bids for  $t$ ,  $p_t^l$ , and  $p_t^s$  had arrived. Therefore, it took slightly longer than that in BAS and  $PAS_f$ .

We investigated influences with resources required by tasks, the fluctuation factor in  $PAS_f$ , and communication delay on overall performance with ASBV. We conducted a number of experiments to do this. The generated tasks were  $t_{5000}$  (so their required resources were  $r_{t_{5000}} = \{5000\}$ ). The resources required by tasks are denoted by subscripts, such as  $t_{5000}$ , after this) in Exp. 3 whose results are plotted in Figs. 4 (a) and (b). Five different tasks,  $t_{2000}$ ,  $t_{3000}$ ,  $t_{4000}$ ,  $t_{5000}$ , and  $t_{6000}$  were randomly generated in Exp. 4 whose results are plotted in Fig. 4 (c). The generated tasks in Exps. 5 and 6 are identical to those in Exp. 2, but the speed factor for message passing,  $D$ , was set to 1.25 and 5 in Exp. 5. Further, fluctuation factor  $f$  was set to two and four in Exp. 6. The results for Exp 5 are plotted in Figs. 4 (d) (e) and (f) and those for Exp. 6 are plotted in Figs. 4 (g) (h) and (i).

Figures 4 (a) and (c) indicate that ASBV could also perform better near (but less than) the upper limits of the entire capabilities for tasks that required different resources and for mixtures of different tasks. Note that the upper limits for task loads are approximately 100 and 125 in Exps. 3 and 4. The curves for improvement ratios in these figures are similar to that in Exp. 2. Figure 4 (b) has the values of RSS in Exp. 3 and these are also quite similar to that in Exp. 2. Note that we omitted the figure for RSS in Exp. 4 because it was almost identical to that in Exp. 3.

The length of communication delay affected improvements to the proposed method, because it also affected the chances of simultaneous multiple awards being made. When it was long ( $D = 1.25$ ), the improvement ratios increased more than those in Exp. 2 (cf. Figs. 4 (d) and 3 (a)). Here, as multiple awards occurred more frequently, performance with PAS was degraded more significantly. However, as managers with ASBV selected non-best bidders, improvements with ASBV increased. Of course, when communication delay was short, the improvement ratios decreased as we can see from Fig. 4 (e). Figure 4 (f) also confirms this phenomenon. Because the chance of multiple awards always increased in BAS if communication delay was longer,  $\sigma_l - \sigma_s$  lowered. Therefore,



**Fig. 4.** Performance ratios and RSS in Exps. 3 to 7.

the managers with ASBV adopted  $PAS_f$  more as award strategy to reduce the chance of multiple awards being made. These experimental results suggest that the proposed strategy is more useful in LSMAS that are widely distributed where their communication delay is significant.

The effect of fluctuation factors also yielded interesting features. If we compare the graphs in Figs. 3 (a), 4 (g), and (h), improvements in their performance when  $L$  was between 80 and 165 were almost similar. However, when  $L$  was beyond the limit of capabilities, performance was slightly lower when fluctuation factor  $f$  was two (therefore, the degree of randomness was high). Only a small amount of randomness was appropriate in the latter situations but condition  $f = 2$  provided too much randomness to awardee decisions even if RSS was larger. However, when  $L$  was between 80 and 165, the proposed strategy could control the values of RSS as shown in Fig. 4 (h); actually, managers with ASBV could adaptively change the values of RSS according to the degrees of randomness provided by  $PAS_f$  ( $f = 2, 3$ , or 4) only when randomness was necessary. Note that the values of RSS always differed according to the values of speed factor in Fig. 4 (f) since communication delay always existed regardless of the task loads. These experiments indicate that ASBV adaptively introduced a degree of randomness, since its selection of award strategy was based on the observed real-time data.

## 6 Conclusion

We proposed an award strategy called award selection according to bid variances, for CNP-like negotiation protocols, in which two award strategies were alternatively selected by estimating local task loads around individual managers based on the statistical analysis of received bid messages for phantom tasks. We then experimentally evaluated and investigated the proposed strategy in various environments. The results indicated that it could outperform naive and probabilistic methods, especially near the upper limits of capabilities for the entire system. This is quite an important characteristic of actual applications as it can exert potential capabilities when really required. It also performed better in wide-area distributed systems in which communication delay was not short. We plan to conduct theoretical analysis to understand the phenomena described in this paper.

**Acknowledgement** This work was in part supported by JSPS KAKENHI Grant Numbers 23650075 and 22300056.

## References

1. Akinine, S., Pinson, S., Shakun, M.F.: An extended multi-agent negotiation protocol. *Autonomous Agents and Multi-Agent Systems* 8(1), 5–45 (2004).
2. Billington, J., Gupta, A., Gallasch, G.: Modelling and Analysing the Contract Net Protocol - Extension Using Coloured Petri Nets. In: *Formal Techniques for Networked and Distributed Systems—FORTE 2008*, LNCS 5048, pp. 169–184. Springer (2008)
3. Fan, G., Huang, H., Jin, S.: An Extended Contract Net Protocol Based on the Personal Assistant. *ISECS International Colloquium on Computing, Communication, Control and Management*, 603–607 (2008)
4. Kinnebrew, J.S., Biswas, G.: Efficient allocation of hierarchically-decomposable tasks in a sensor web contract net. In: *Proc. of IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology*. vol. 2, pp. 225–232 (2009)
5. Sandholm, T., Lesser, V.: Issues in automated negotiation and electronic commerce: Extending the contract net framework. In: Lesser, V. (ed.) *Proc. of the 1st Int. Conf. on Multi-Agent Systems (ICMAS'95)*. pp. 328–335 (1995)
6. Schillo, M., Kray, C., Fischer, K.: The Eager Bidder Problem: A Fundamental Problem of DAI and Selected Solutions. In: *Proc. of AAMAS2002*. pp. 599 – 606 (2002)
7. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers* C-29(12), 1104–1113 (1980)
8. Sugawara, T., Fukuda, K., Hirotsu, T., Kurihara, S.: Effect of alternative distributed task allocation strategy based on local observations in contract net protocol. In: *Principles and Practice of Multi-Agent Systems*, LNCS 7057, pp. 90–104 (2012)
9. Xu, L., Weigand, H.: The Evolution of the Contract Net Protocol. In: *Proc. of WAIM 2001*. vol. LNCS 2118, pp. 257–264 (2001)