

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Characteristics of Contract-Based Task Allocation by a Large Number of Self-Interested Agents

Toshiharu Sugawara¹, Toshio Hirotsu²,
Satoshi Kurihara³ and Kensuke Fukuda⁴

¹Waseda University,

²Toyohashi University of Technology,

³Osaka University,

⁴National Institute of Informatics,

Japan

1. Introduction

Task and resource allocation is a key technology in Internet applications, such as grid computing and agent grids, for appropriately allocating tasks such as video and music downloads, scientific calculations, and language services (Cao et al., 2005; Chunlin & Layuan, 2006). In these applications, tasks should be allocated to and done in appropriate agents. Thus negotiation protocol for task allocation is one of the important issues in multi-agent systems research. In particular, the contract net protocol (CNP) and its extensions have been widely used in certain applications because of their simplicity and superior performance (Sandholm, 1993; Smith, 1980; Weyns et al., 2006). Agents in CNP play one of two roles: as *managers* that are responsible for allocating tasks and monitoring processes, or as *contractors* that are responsible for executing the allocated tasks. A manager agent announces a task to contractor agents, which bid for the task with certain promised values (such as cost, duration, and payment). The manager then awards the task to the contractor (called an *awardee*) that bid with the best tender (before the deadline) and allocates the task to it. However, the naive CNP is usually assumed to be applied to allocating tasks in small-scale, non-busy multi-agent systems (MASs).

Interference among agents is always observed in this kind of negotiation protocol. Consider many agents having to allocate tasks to other efficient contractors. In basic CNP, a contractor agent that receives task announcements bids for the tasks one by one. When many tasks are announced by many managers, however, they have to wait a long time to receive a sufficient number of bids. In the original conception of CNP (Smith, 1980), multiple bids were proposed to concurrently handle many announcements. If a contractor is awarded multiple tasks simultaneously, however, it may not be able to provide its promised quality or performance. In fact, efficient contractor agents are selected as awardees by many manager agents, leading to a concentration of tasks. In addition, when a large number of

Source: Evolutionary Computation, Book edited by: Wellington Pinheiro dos Santos,
ISBN 978-953-307-008-7, pp. 572, October 2009, I-Tech, Vienna, Austria

agents in a massively multi-agent systems (MMAS) interact with each other, the protocol generates an excessive number of messages, so that all agents become busy with (1) reading and analyzing many received messages (all agents), (2) deciding whether to bid for announced tasks (contractors), (3) calculating bid values (contractors), and (4) selecting awardees (managers). This phenomenon degrades the overall performance of a MAS. We believe that this kind of situation will often appear in Internet applications as mentioned before.

A simple solution to this problem is to implement manager-side control by restricting announcements to certain selected agents so as to reduce the number of messages and simplify the award process. In this article, we call this approach *restricted CNP*. We can easily expect, however, that a strong restriction may also degrade the task performance, because tasks are not announced to idle or high-ability agents. It is unclear whether the overall MAS performance will ultimately get better or worse if tasks are more widely announced, especially in an MMAS environment in which more than 1000 agents interact with each other. Restricting the audience for task announcement to improve performance, especially to avoid message congestion, has been proposed for small-scale MASs in a number of papers (Sandholm, 1993; Parunak, 1987; Schillo et al., 2002). To our knowledge, however, there has been little research on the efficiency and effectiveness of CNP (or more generally, negotiation protocols including CNP) when it is applied to a MMAS.

The goal of our research is to understand the behavior of CNP in an MMAS in order to develop efficient large-scale negotiation protocols. As a first step toward this purpose, we have investigated the performance features of CNP in an MMAS, especially the overall efficiency and the reliability of promised bid values, when tasks are allocated by CNP with a variety of manager-side controls, by using a multi-agent simulation environment that we previously developed (Sugawara et al., 2006). Manager-side controls can affect both the announcement and award phases. For this purpose, we previously reported that the relationship between overall performance and announcement control (Sugawara et al., 2007). The aim of this chapter is to discuss performance characteristics, that is, how overall performance of MMAS changes according to the manager-side control in the awarding phase.

On the other hand, we assume that all bid values from contactors reflect their states and abilities. Of course, we can also consider a number of contractor-side controls, and effective task allocation methods can be achieved through coordination of managers and contractors. The MAS performance also depends on the integrity of bid values and prior knowledge about other agents (Falcone et al., 2004). It is important, however, to clearly separate the effects of these controls; hence, this chapter is dedicated to showing that manager-side controls strongly affect the overall efficiency and the reliability of promised values.

This chapter is organized as follows. First, the restricted CNP model used in our simulation is presented. Then, we briefly discuss our experiments to investigate the performance of an MMAS when all agents use a simple version of restricted CNP, and we analyze the results to understand how the overall efficiency changes. Using these results as a baseline, we then introduce some variations of manager-side controls into restricted CNP for the award processes, and we show how the performance characteristics vary through our simulation. Finally, we discuss the meaning of our experimental results, related research, and applications.

2. Simulation of restricted CNP

2.1 Model for restricted CNP

First, we formally explain a simple version of restricted CNP in order to clarify the design of our simulation.

Let $A = \{a_1, \dots, a_n\}$ be a set of agents, $M = \{m_j\} (\subset A)$ be a set of managers that will allocate tasks, and $C = \{c_k\} (\subset A)$ be a set of contractors that can execute allocated tasks if a contract is awarded. To simplify the experimental setting below, we assume that M and C are disjoint and $A = M \cup C$.

When manager m_j has task T , it will allocate T to another agent according to CNP as follows. First, m_j announces task T to all contractors in C (i.e., the announcement phase). For brevity, we assume that all contractor agents can execute T . A contractor receiving this announcement must decide whether to bid for this task. If it decides to bid, it has to send m_j a bid message with a certain value called the *bid value* (i.e., the bidding phase). Although bid values in general might include parameters such as the price for executing T , the quality of the result, and a combination of these values, timely responses are always a great concern in interactive services and real-time applications. Thus, in this chapter, we assume that all agents are rationally self-interested on the basis of efficiency, and their bid values are assumed to reflect their efficiency and workload, giving a promised time for completing T . How this time is calculated will be describe in the next section. Finally, m_j selects one awardee, which is usually the contractor that bid with the best value, and sends an award message to that contractor to allocate the announced task (i.e., the award phase).

As the basic CNP mentioned above is only adopted for a small-scale MAS in a non-busy environment, we modify it to adapt busier, MMAS environments. First, as in (Smith, 1980), we assume that contractors are allowed to submit multiple bids concurrently to handle task announcements from many managers, in order to apply this approach to busy, large-scale Internet and ubiquitous-computing applications. Second, unlike in the original CNP, two new CNP messages, *regret* and *no-bid* messages, are introduced (for example, (Sandholm, 1993; Xu & Weigand, 2001)). Regret messages are sent in the award phase to contractors that have not been awarded the contract, while no-bid messages¹ are sent to managers when contractors decide not to bid on an announced task. Using these messages avoids long waits for bid and award messages.

Next, we define restricted CNP. First, we assume that $|A|$ is large (on the order of thousands), so $|M|$ and $|C|$ are also large, and that the agents are distributed widely, like servers and customer agents on the Internet. For $m_j \in M$, let K_{m_j} be a set of contractors known to m_j ; manager m_j can only announce a task to contractors in K_{m_j} . Set K_{m_j} is called *scope* in this article. Restricted CNP is thus defined as CNP where (1) multiple bids and regret and no-bid messages are allowed, and (2) any manager m_j is restricted to announcements to certain contractors in K_{m_j} according to a certain policy, called the *announcement policy*. Hereafter, the set of contractors selected according to the announcement policy is called the *audience*. We believe that an appropriate announcement policy can reduce the total number of messages and the cost of announcing bidding and awarding decisions, thus improving overall performance. In our experiments, we introduce

¹ A no-bid message might correspond to a busy response in (Smith, 1980).

various announcement policies and examine how the performance of an MMAS varied under these policies.

2.2 Simulation model

In this section, we describe the simulation settings in our experiments. First, we set $|C|=500$ and $|M|=10000$ and assume that all contractors in C are eligible to bid for any announced task². The agents are randomly placed on the points of the grid space that is expressed by a 150×150 coordinates with a torus topology. Then, the Manhattan distance $dist(a_i, a_j)$ between agents a_i and a_j is defined on this space. According to this distance, we can set the communication cost (or delay) of messages from a_i to a_j in our simulation. This cost is denoted by $cost(a_i, a_j)$. In this simulation, the communication cost ranges over $[1,14]$ (in *ticks*, the unit of time in the simulation), in proportion to the distance between the source and target agents. The elements of scope K_{m_j} for $\forall m_j \in M$ are also defined according to this distance: they consist of the nearest 50 or more contractors to m_j . More precisely, for an integer $n > 0$, let $K_{m_j}(n) = \{c \in C \mid dist(m_j, c) \leq n\}$. Then, it follows that $K_{m_j}(n) \subset K_{m_j}(n+1)$. K_{m_j} is defined as the smallest $K_{m_j}(n)$ such that $|K_{m_j}(n)| \geq 50$. K_{m_j} keeps a static value after it is initially calculated.

With every tick, tl tasks are generated by the simulation environment and randomly assigned to tl different managers, where tl is a positive integer. Parameter tl is called the *task load* and denotes tl tasks per tick, or simply tl T/t. A manager assigned a task immediately initiates restricted CNP to allocate the task to an appropriate contractor. We can naturally extend the task load for positive non-integer numbers by probabilistic task generation. For example, $tl=9.2$ means that the environment generates 9 tasks (with probability 0.8) or 10 tasks (with probability 0.2) every tick.

For task T and agent a_i , we introduce two parameters: the associated cost of T , $cost(T)$, expressing the cost to complete T ; and the ability of a_i , $Ab(a_i)$, expressing the processing speed of a_i . For convenience, we adjust these parameters so that contractor c_i can complete an allocated task, T , in $cost(T)/Ab(c_i)$ ticks. Since our experiments are preliminary and designed simply to understand the performance features of restricted CNP in an MMAS, we assume that all tasks have the same cost, 2500. Instead, we assigned different abilities to individual contractors; the abilities of the contractors are initially assigned so that the values of $cost(T)/Ab(c_i)$ (where $i=1, \dots, 500$) are *uniformly distributed* over the range $[20,100]$. This means that the values of $Ab(c_i)$ range from 25 to 125.

When contractor c_i is awarded a task, it immediately executes it if it has no other tasks. If c_i is already executing another task, the new task is stored in the queue of c_i , which can hold up to 20 tasks. The tasks in the queue are then executed in turn. Tasks that cannot be stored because of a full queue are dropped. This situation is called *over-allocation*.

The bid value reflecting the state of contractor c_i is the expected response time, calculated as follows. Suppose that s tasks are queued in c_i . Then, its bid value is $s \times (2500/Ab(c_i)) + \alpha$, where α is the required time to complete the current task, so smaller bid values are better. In multiple bidding, c_i might have a number of uncertain bids whose results have not yet

² We assume that the agents run on the Internet, where contractor agents provide services requested by manager agents, which correspond to user-side computers.

been received. These bids are not considered, however, because it is uncertain whether they will be awarded. This means that contractors always submit bids when a task announcement arrives. Note that we can introduce a bidding strategy to all contractors in order to avoid over-allocation, by disallowing multiple bids when a contractor is busy, but we do not consider this kind of contractor-side control or strategy here. The use of other bidding strategies is discussed later.

The *completion time* of each task is the elapsed time observed by the manager from the time of sending an award message with the allocated task to the time of receiving a message indicating task completion. The completion time thus includes the communication time in both directions, and the queue time and execution time of the contractor³. We define the *overall efficiency of a MAS* as the average completion time observed for all managers and the *reliability* as the expected value of the differences between the completion times and the promised response times.

tl (T/t)	Condition of MAS
0.1	The task load is extremely low; multiple bids are rare.
0.5--1	The MAS is not busy.
3--6	The MAS is moderately busy; no tasks are dropped.
9--10	The task load is near the limit of the MAS performance; some tasks may be dropped.
11	The MAS is extremely busy, beyond its theoretical performance limit; many tasks are dropped.

Table 1. MAS conditions for various task load (tl) values.

The simulation data reported in this chapter are the mean values from three independent experiments using different random seeds. The theoretical cumulative ability of all contractors in the three MASs in the three experiments ranges from 9.7 to 10.2 T/t, with an average value of 9.9 T/t. We set parameter tl as 0.1, 0.5--1, 3--6, 9--10, or 11; the conditions of the MAS in each case are listed in Table 1.

3. Random selection in announcement --- overview

We introduce the announcement policy under which any manager, m_j , announces tasks to only n contractors randomly selected from K_{m_j} to reduce the number of messages in CNP. This announcement policy is called the *random selection policy* and is denoted as RSP(n), where n is a positive integer called the *announcement number* indicating the number of announcement. This policy requires neither prior knowledge nor learning about the contractors, but some tasks are not announced to highly capable contractors.

³ Because our experiments are preliminary and our goal is to understand the effects of having many contractors and managers, the costs of processing announcement messages and selecting a bid from among bid messages are not included in the completion time. Of course, these costs should be included in our future work in order to identify their effects on performance.

We previously examined in (Sugawara et al., 2007) how overall efficiency varies with announcement number, n , and the task load, tl . The experimental data are shown in Fig. 1. The results can be summarized as follows:

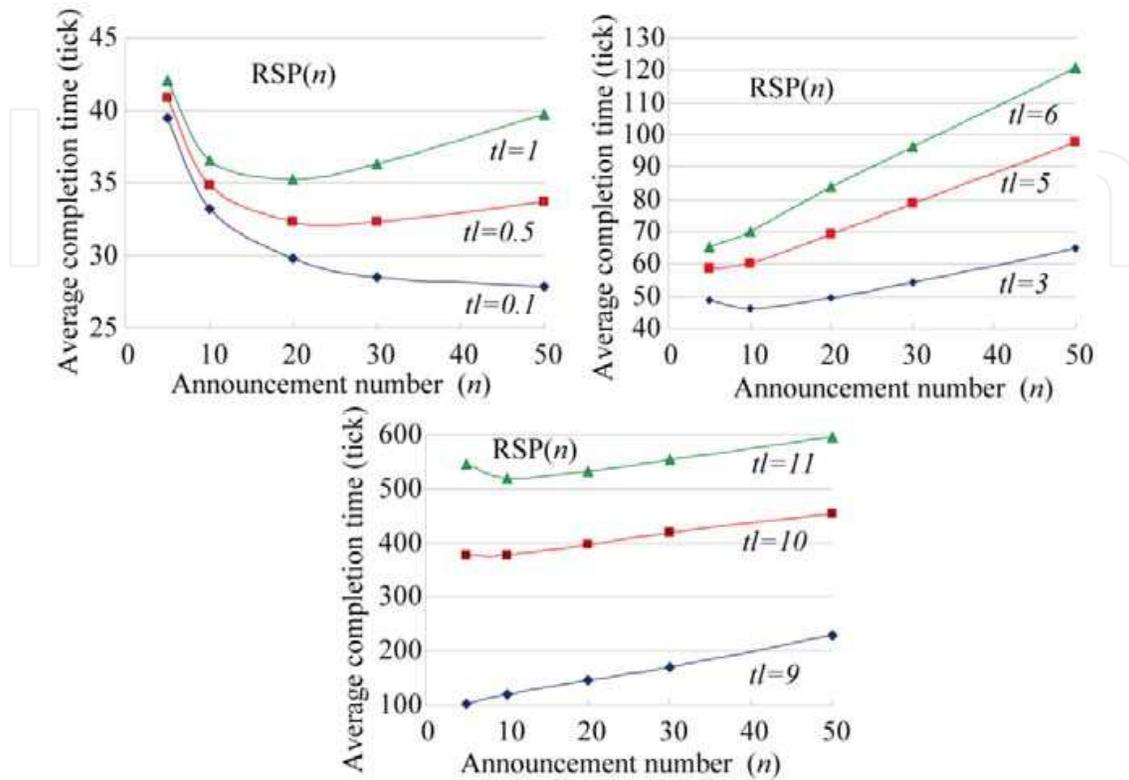


Fig. 1. Average completion time under RSP(n).

- Our notion that smaller n resulted in inefficiency in the entire MMAS because tasks were not announced to highly capable agents only applies when the task load was extremely low. Because managers send many task announcement messages under RSP(n) for larger n , we predicted task concentration in a few good contractors in busier environments, thus making the MMAS inefficient. However, our results clearly indicated that this phenomenon could be observed much earlier (i.e., even when task load was low, such as the case $tl=0.5$. See Fig. 1) than we expected.
- We conducted an experiment to find when concentration occurred. However, we found that the numbers of tasks awarded to (so executed in) individual agents in a busy case ($tl=10$) were almost equal under RSP(5) and RSP(50). This indicated that no concentration occurred in terms of the total numbers awarded against our prediction.
- We then additionally investigated the average difference between bid values (i.e., promised times) for contractor c_i and the actual queue and execution times, and the standard deviation of these differences, when $tl=10$. Note that this average difference is denoted by d_{c_i} and the standard deviation by sd_{c_i} . The results suggested that although the numbers of tasks awarded to each contractor under RSP(5) and RSP(50) eventually became almost the same, the excess tasks were only concentrated on a few contractors, and these busy contractors changed over time under RSP(50). We could observe another explicit tendency that more tasks were awarded simultaneously to contractors with less ability under RSP(50).

We also investigated what effect the learning method had by in which managers estimated which contractor was more efficient (so that completion time was expected to be shorter). We found, however, that this learning could only improve overall efficiency when task load was extremely low, which corresponds to the situation where sequential conventional CNP is used. See our previous paper (Sugawara et al., 2007) for a more detailed discussion.

4. Probabilistic fluctuation in award phase

4.1 Effect of small fluctuation

Our previous experiments showed that overall efficiency becomes worse with concentration on a few contractors caused by simultaneous and excess multiple awards. One method of avoiding such concentration is to introduce some degree of fluctuation in the award phase. After a task announcement, manager m_j receives bids from a number of contractors, $\{c_1, \dots, c_p\}$. We denote the bid value from contractor c_i as $b(c_i)$. Manager m_j then selects an awardee, c_i , according to the following probabilistic distribution:

$$Pr(c_i) = \frac{1/b(c_i)^k}{\sum_{l=1}^p 1/b(c_l)^k} \tag{1}$$

Note that smaller bid values are better. This award strategy is called *probabilistic award selection* and denoted as PAS_k . The larger k is, the smaller the degree of fluctuation is. To examine the effect of fluctuation as a preliminary experiment, k is set here to 2.

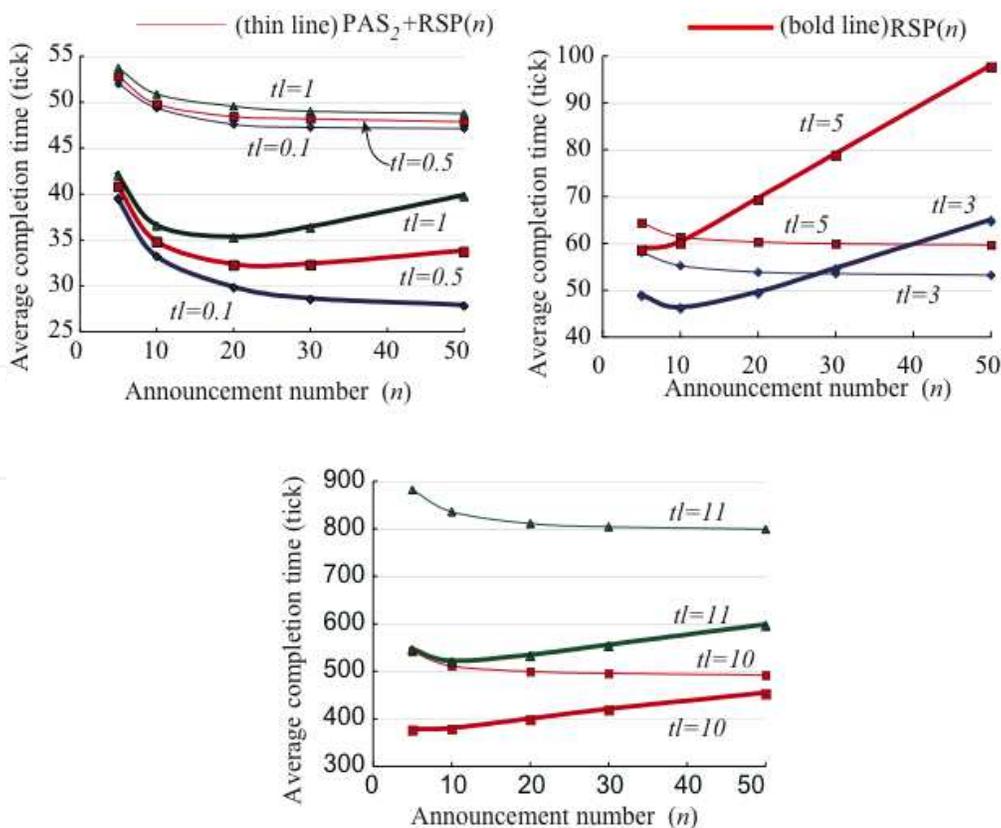


Fig. 2. Average completion times under $PAS_2+RSP(n)$.

The graphs in Fig. 2 compare the average completion times when managers only adopt $RSP(n)$ with these times when they adopt $RSP(n)$ as the announcement policy and PAS_2 as the award strategy, giving a combination denoted as $PAS_2+RSP(n)$. These graphs show that policy $RSP(n)$ only leads to better overall efficiency (than policy $PAS_2+RSP(n)$) when the task load is low ($tl < 3$) or extremely high ($tl > 10$, beyond the theoretical limit of the capability of all contractors). Conversely, we can observe that the overall efficiency under $PAS_2+RSP(n)$ is considerably improved when $3 \leq tl \leq 9$.

The graphs in Fig. 2 also indicate that (1) under $RSP(n)$, if the announcement number n is larger, the overall efficiency worsens except in situations where the task load is low, (2) but under $PAS_2+RSP(n)$, the overall efficiency slightly improves if n is larger. This suggests that a small fluctuation can ease the concentration to a few contractors, so the effect of larger announcement numbers appear as we can expected and actually $PAS_2+RSP(n)$ outperforms $RSP(n)$ when the system is moderately busy.

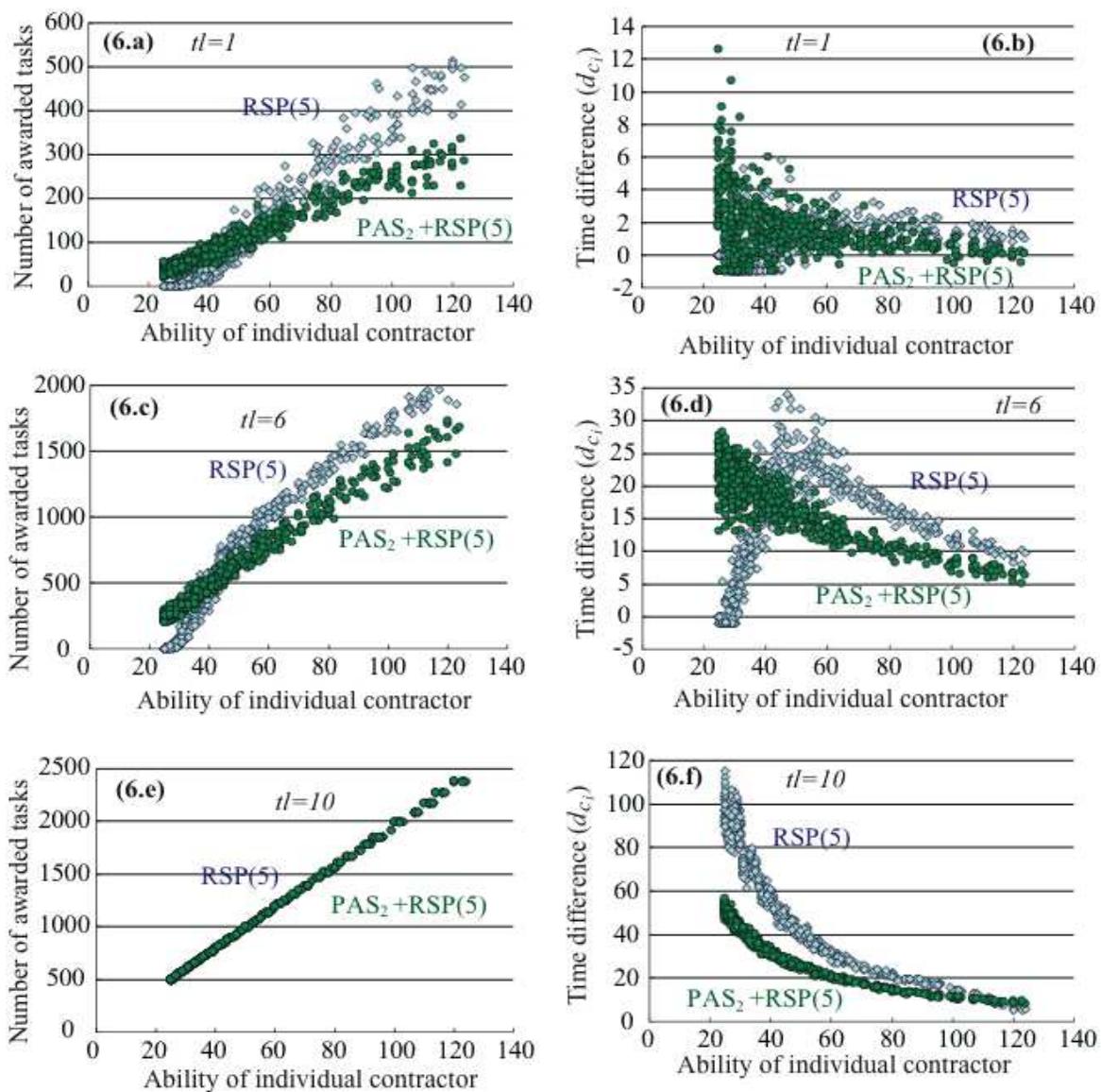


Fig. 3. Comparison of $PAS_2+RSP(n)$ and $RSP(n)$: Number of awarded tasks and differences between promised and actual completion times.

4.2 Distributions of awarded tasks and reliability

By comparing Fig. 2 with Fig. 1, we can observe that the overall efficiency under $PAS_2+RSP(n)$ is considerably improved when $3 \leq tl \leq 9$ but considerably degraded when $tl \leq 1$ or $tl \geq 10$. Figure 2 partly explains this situation. When the task load is low, awarding tasks to high-ability contractors can lead to better performance. Graphs (6.a) and (6.b) in Fig. 3 indicate that more tasks are awarded to low-ability contractors because of fluctuation, but this is unnecessary because not all contractors are busy; this results in poor efficiency. When the MAS is moderately busy, low-ability contractors are awarded few tasks under $RSP(5)$. This does not utilize the full capability of the entire MAS. Graphs (6.c) and (6.d) in Fig. 3 show, however, that low-ability contractors are appropriately awarded as a result of fluctuation. In an extremely busy situation (graphs (6.e) and (6.f) in Fig. 3), all contractors are already very busy. The fluctuation results in some tasks being awarded to contractors whose bids are not the best values, meaning that the awardees might be occupied. In addition, graph (6.e) clearly indicates that fluctuation eventually does not affect the distribution of awarded tasks. These results suggest that applying $PAS_2+RSP(5)$ shifted some tasks to inappropriate contractors with inadequate timing.

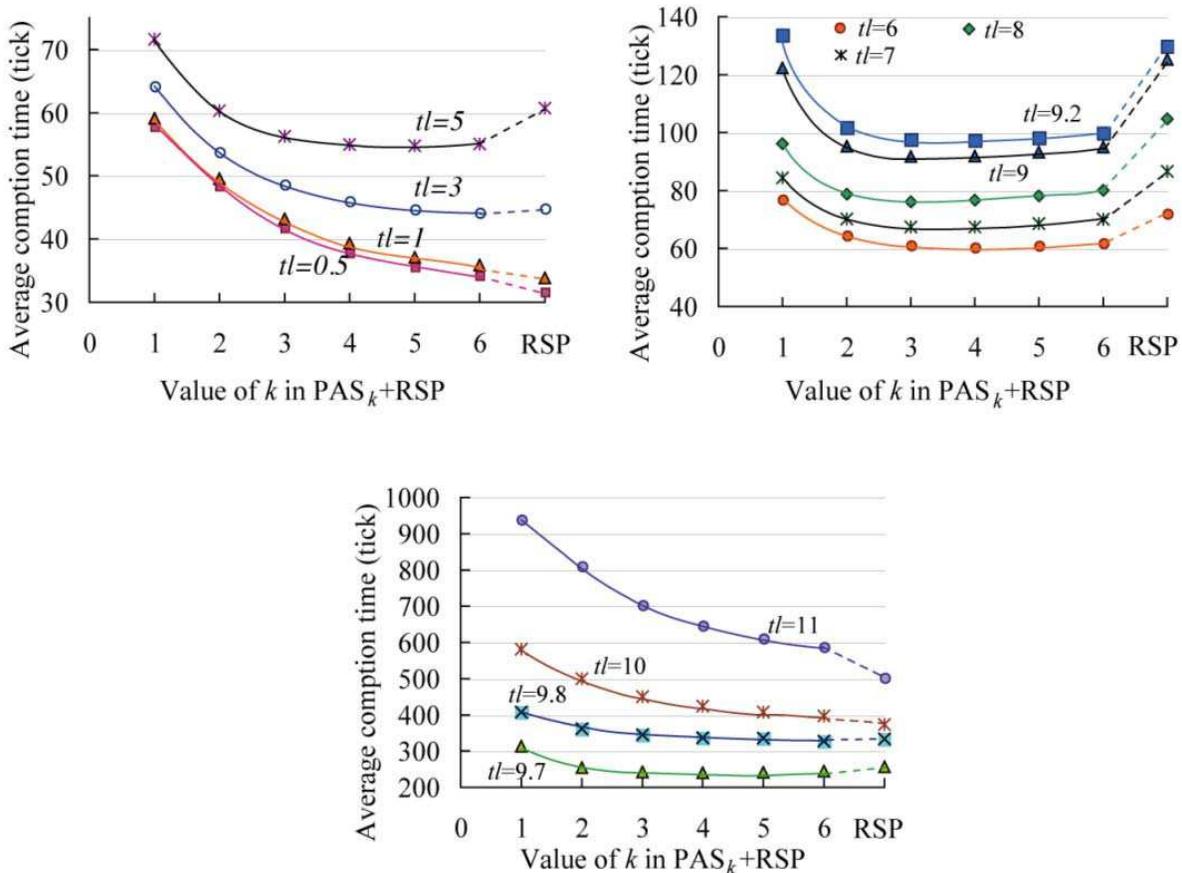


Fig. 4. Variations in completion times under $PAS_k+RSP(20)$.

5. Effect of degree of fluctuations --- detailed analysis

5.1 Effect on overall efficiency

To understand the effect of fluctuation more clearly, we varied the value of fluctuation factor k in Eq. (1), where smaller k increases the degree of fluctuation. The results of this

experiment are plotted in Figs. 4 (a) to (c), where k ranges from 1 to 6 and the announcement number, n is 20⁴. Note that “RSP” in x-axiom means the RSP policy.

These figures illustrate that policy, RSP, only results in better performance than PAS_k+RSP when task load is less than one (non-busy) or more than 10 (extremely busy, i.e., over the theoretical limits of the entire MAS). In other situations where $2 \leq tl < 10$, some degree of fluctuation can express much better performance, but a k value leading to the best performance depends on the task load. When tl is close to one, the larger k is better, when tl is greater than one, the value of k that expresses the best performance gradually approaches 3. However, after k is larger than nine, the best k value swiftly approaches 6, again. This analysis suggests that a flexible policy of selecting awards sensitive to task load is required to achieve overall efficiency in MMAS.

5.2 Effect on reliability of bids and number of dropped tasks

Another important performance measure for MMAS is the reliability of bid values; in our experiment, we define the reliability as the average difference, i.e., the expected values of $\{d_{c_i}\}_{c_i \in C}$ and the standard deviation of these differences. Of course, smaller differences indicate greater reliability. The graphs of observed differences are in Fig. 5. We also measured the standard deviation, but the shapes of the graphs are quite similar to the ones in Fig. 5. Thus, they have been omitted from this chapter. Instead, some example values are listed in Table 2, where the columns marked $k=i$ and RSP correspond to policies, $PAS_i+RSP(10)$ and $RSP(10)$, respectively. These graphs and the table clearly indicate that even when tl is low, the values of differences and the standard deviation are larger under RSP than those under PAS_k+RSP .

Reliability decreases when the task load increases but the graphs in Fig. 5 illustrate that introducing some fluctuation also makes the system more reliable, especially, $k=2$ or 3 generally makes it more reliable in all task loads. Conversely, policy RSP yields low reliability except when task load is extremely low. If we want to simultaneously pursue overall efficiency and reliability, PAS_2+RSP or PAS_3+RSP is reasonable for any task load.

Furthermore, if we compare graphs (a-2) and (b-2) in Fig. 5, smaller task number n results in much smaller differences. However, too few n makes the system inefficient as explained in the previous subsection. This is another trade-off between efficiency and reliability in MMASs.

To fully understand system reliability, we have to pay attention to the number of dropped tasks because of queues overflowing at contractors. The ratios of the numbers of dropped tasks to those of generated tasks are listed in Table 3 (a) and (b). Note that no dropped tasks were observed when $tl \leq 9.5$. These tables show that there is not much difference between RSP and PAS_k+RSP , in terms of the number of dropped tasks. This suggests that some appropriate degree of fluctuation can lead to properly efficient MMASs.

⁴ We have only shown graphs when $n=20$, but other graphs when $n=10$ to 50 have almost the same shapes.

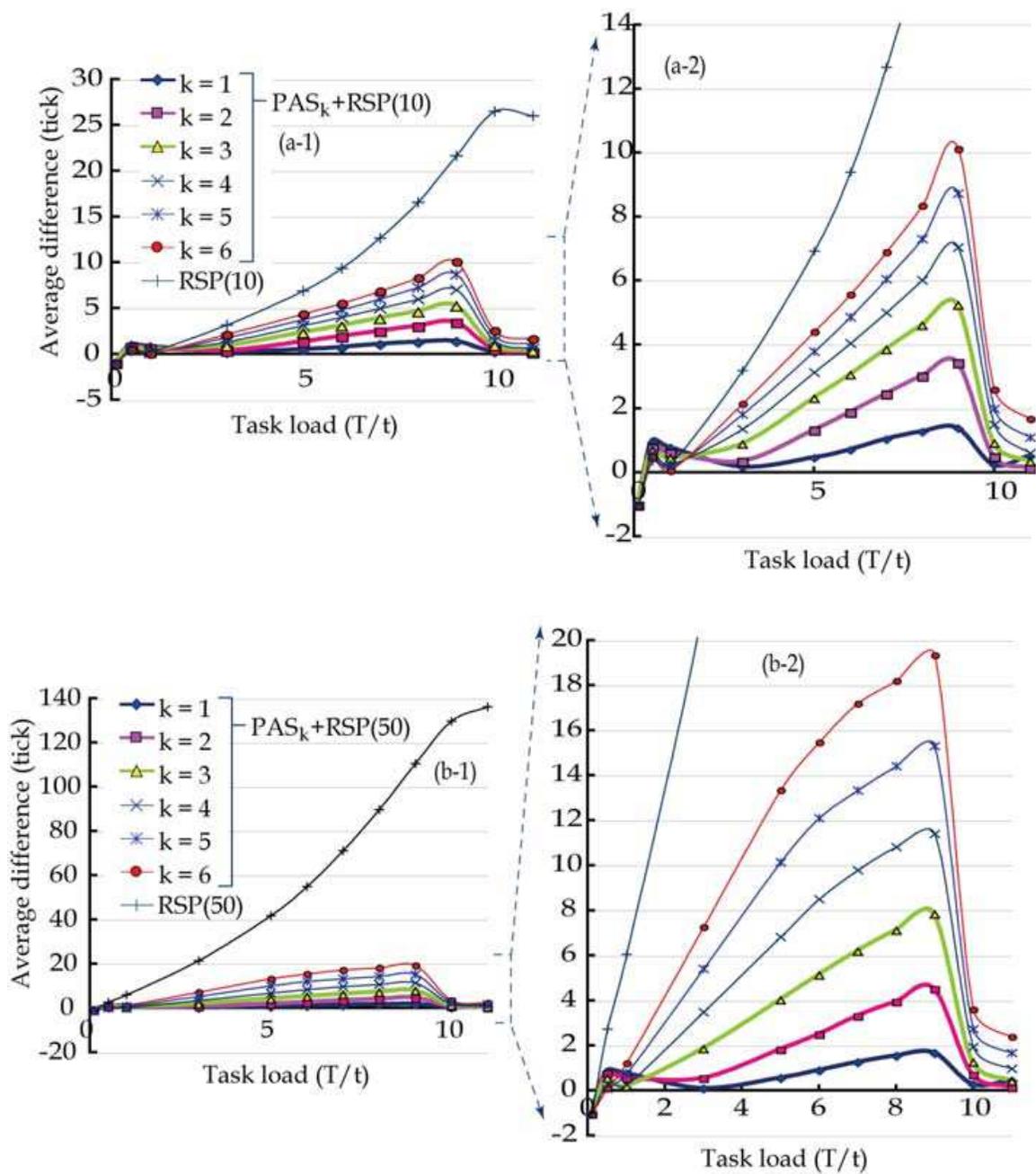


Fig. 5. Average differences between bid values and observed completion times under $PAS_k+RSP(n)$.

Policy	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	RSP
$n = 5$	5.13	4.70	4.71	5.16	5.41	5.90	6.23
$n = 9$	20.92	22.98	25.04	27.06	28.88	30.16	39.75
$n = 10$	20.03	20.50	21.25	22.21	22.98	23.84	47.11

Table 2. Standard deviation of differences between bid values and completion times under $PAS_k+RSP(10)$.

(a) Ratio (%) of dropped tasks when $n = 10$.

Fluctuation factor (k)	$tl = 9.7$	$tl = 9.8$	$tl = 10$	$tl = 11$
$k = 1$	0.02	0.30	2.03	11.11
$k = 2$	0.02	0.33	2.16	11.16
$k = 3$	0.04	0.48	2.34	11.17
$k = 4$	0.08	0.54	2.42	11.18
$k = 5$	0.09	0.64	2.47	11.18
$k = 6$	0.12	0.63	2.49	11.19
RSP	0.21	0.86	2.60	11.18

(b) Ratio (%) of dropped tasks when $n = 50$.

Fluctuation factor (k)	$tl = 9.7$	$tl = 9.8$	$tl = 10$	$tl = 11$
$k = 1$	0.01	0.24	1.98	11.11
$k = 2$	0.02	0.38	2.22	11.18
$k = 3$	0.06	0.52	2.38	11.18
$k = 4$	0.09	0.60	2.46	11.19
$k = 5$	0.12	0.65	2.50	11.18
$k = 6$	0.14	0.69	2.53	11.18
RSP	0.33	1.03	2.61	11.19

Table 3. Ratio of Observed Dropped Tasks in Policies RSP(n) and PAS $_k$ +RSP(n) when task load tl is high.

6. Pursuit of reliability

It is quite natural that, if we pursue the reliability of a system, manager agents will only announce tasks to a limited number of contractors based on past reliable completions. Taking this into consideration, we introduced an announcement policy, *more reliable contractor selection policy* (MRSP) as follows: manager m retains difference d_{c_i} ($c_i \in K_{m_j}$), and the manager agent only announces the task to the most reliable (i.e., the smallest d_{c_i}) n contractors.

However, MRSP cannot provide good reliability. More seriously, MRSP can neither improve the reliability nor the efficiency of the system despite many more dropped tasks. These data are listed in Table 4. More drops occur when announcement number n is larger; these are the reverse of characteristics of RSP that appeared in Table 3. More discussion is provided in Section 7.3.

Task load	$tl = 9.2$	$tl = 9.5$	$tl = 9.8$	$tl = 10$	$tl = 11$
$n = 10$	1.00	3.94	7.35	10.35	21.18
$n = 50$	0.00	0.00	1.32	2.76	12.93

Table 4. Ratio (%) of Observed Dropped Tasks under MRSP(n).

7. Discussion

7.1 Announcement restriction

Our first experiment suggests that a small announcement number n is better in an MMAS. We can easily expect that if the task load is high, many tasks will be awarded to high-ability contractors for a large n , thus lowering the performance. Our experiments also show, however, that this phenomenon appears even if the task load is not that high. From our experiments, applying RSP(n) with larger n gives better results only when $tl=0.1$. Since our simulation does not include the cost of message analysis, the overall efficiency of an actual MAS must be much worse than that obtained experimentally here. Conversely, when $tl=0.1$, using a small audience has an adverse effect.

7.2 Capriciousness -- efficiency or reliability

Our experiments expressly provide data that fluctuations in award selection will strongly affect overall efficiency and reliability; a little capriciousness in a manager's award selection will significantly improve them. This suggests that appropriate control of fluctuation is required to practically use CNP in large-scale MASs. The control of announcement number (random audience restriction) by manager agents is also required for this usage but its effect is relatively smaller.

If we carefully look at the experimental data, the required control must be slightly different depending on whether we give weight to efficiency or reliability. If reliability is more important, constant fluctuation in the award strategy with fewer announcements such as PAS₂+RSP(10) provides better results. However, if we attach a high value to efficiency, more tailored control must be investigated; i.e., (1) no fluctuation with a smaller announcement number is better when task load is extremely low or exceeds the entire processing capability of the MMAS, and (2) in the other case, the controls of fluctuation factors ranging from 2 to 6 with a relatively larger announcement number should be flexibly introduced in individual manager agents.

As described in Section 1, we have assumed that all agents are rationally self-interested on the basis of efficiency. However, the rational decisions in awarding lead to the concentrations. Our experiments suggest that sometimes selecting the second or third best contractors in awarding can provide more preferable performance to individual manager agents as well as the whole MMAS. It is, however, necessary that the degree of fluctuation should also be controlled according to the state, e.g., the task loads of the MMAS for better performance.

7.3 Effect of learning

In Section 6, we stated that MRSP leads to poor efficiency and reliability as well as more dropped tasks. This feature is quite similar to the strategy where all manager agents learn which contractors are more efficient and determine the audience for tasks based on these learning results (Sugawara et al., 2007). We believe that learning is too slow for managers to adapt to the variations in task allocations from the environment and this ironically makes the system more inflexible compared to random audience selection. Note that randomness is not uniform; although, in our experiment, tasks were generated in a constant rate and assigned to randomly but not uniformly selected managers, so slight, transient and continual variations in task assignment may always occur. However, we believe that this slight variation cannot be ignored in MMAS because local and small-scale concentrations

/biases in tasks reduce the system's efficiency and reliability. Although learning can identify high-performance contractors and other long-term variations, it cannot adapt easy-to-move concentrations derived by continual slight variations. Our experiments suggest that a small fluctuation in award selection leads to better results for this phenomenon.

Despite this, we believe that this kind of learning will be necessary for real applications from another viewpoint. For example, if there is a malicious contractor that usually bids with unreliable values, manager must exclude it from their scopes for the stable and fair task allocations.

7.4 Applications

We believe that the importance of this kind of research will become clear with the increase in large-scale interactions between agents. Consider the recent growth in the volume of e-commerce transactions on the Internet for buying goods, such as software, music, and video. These transactions usually consist of coordinated tasks including interactions with a variety of agents, which are in charge of, for example, customer authentication and management, stock management, shipping control, and payment processing. Suppose that a customer wants to download a music album from one of many music servers deployed worldwide. She (or her agent) has to determine the server to download from. Distant servers are probably inappropriate because of high latency, while nearby servers might be busy. Similar situations also occur when purchasing a good: if local sellers on the Internet are busy and the goods are transported on a FIFO basis, customers may have to wait for a long handling time and often back-ordering.

Reliability of bid values, which corresponds to the estimated completion time of required services, is also strongly required in a number of applications. Again, consider the example of downloading music. Customers do not always require immediate downloading. Suppose that a customer wants to download a music album, but the server is currently busy. She wants to listen to it tomorrow morning, so her agent only has to finish downloading it by then. Even if this album is very popular, the agent can schedule the download via a contract with an appropriate music server (i.e., a contractor in our experiment) by using the promised completion time and standard deviation. Our experiment suggests that this server should be selected probabilistically. These kinds of situations may also occur in other applications such as PC-grid computing where many tasks derived by decomposing a large computing problem should effectively be processed by distributed PC resources.

7.5 Related research

There is much existing research on improving the performance and functionality of CNP. In a city traffic scheduling system called TRACONET (Sandholm, 1993), transportation tasks are allocated by CNP, with bidding decisions based on the marginal cost. This approach to CNP has been further extended (Sandholm & Lesser, 1995) by introducing the concept of *levels of commitment*, making a commitment breakable with some penalty if a more valuable contract is announced later.

The necessity of message congestion management has also been demonstrated (Sandholm & Lesser, 1995). We believe that the first work discussing message control is (Parunak, 1987), which states that audience restriction can improve the overall efficiency. In (Parunak, 1987), this restriction is implemented by maintaining a bidding history. Because past bidding activities are known, agents lacking a specific ability cannot submit bids for tasks requiring

this ability. Another work (Gu & Ishida, 1996) analyzes how contractors' utilities change according to the ratio of contractors and managers. All this research assumes, however, that agents are not so busy that interference among agents is insignificant. Our experiments indicate that CNP in an MMAS exhibits quite different characteristics from CNP in a small-scale MAS.

More recently, (Schillo et al., 2002) discusses the issue of the eager-bidder problem occurring a large-scale MAS, where a number of tasks are announced concurrently so that CNP with levels of commitment does not work well. These authors propose another CNP extension, based on statistical risk management. The number of agents in the MAS considered in their paper is about 100, however, which is much less than in our model. More importantly, the types of resources and tasks considered are quite different: specifically, the resources are exclusive, such as airplane seats, so they should be allocated selectively, in order to gain more income. In our case, the resource is CPU or network bandwidth, which can accept a certain number of tasks simultaneously but with reduced quality. As a result, the many agents and tasks in our experiments incur floating uncertainty, which affects learning, statistical estimation, and rational decision-making.

8. Conclusion

In this chapter, we have described our experimental investigation of the performance features, especially the overall efficiency and the reliability of bid values, in an MMAS when the contract net protocol (CNP) is used with a number of manager-side controls. Our results suggest that applying a small amount of fluctuation can improve both the efficiency and the reliability. Thus, we also investigated how performance and reliability change according to the degree of probabilistic fluctuations. This experimental result indicates that there is the appropriate degree of fluctuation that can maximize the overall performance as well as that can improve the reliability of bids.

This chapter is dedicated solely to manager-side controls. It is clear that some contractor-side controls, such as strategies related to choosing to bid and calculating bid values, also affect performance. We must emphasize, however, that only manager-side controls can improve the reliability of promised values in busy situations, as well as the overall efficiency of an MMAS. For example, contractors can submit more credible bid values by taking into account the other submitted bids, whose results are uncertain. We also examined this case but found that the resulting overall efficiency was essentially the same as that obtained in the experiments described in this chapter. The effect of dynamic interference is rather strong, giving credible bids less meaning. Coordination among contractors and managers is probably required to address this issue, and this is our research direction.

9. Acknowledgement

This research was supported in part by Grant of Kayamori Foundation of Informational Science (2009-2010) and by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research, 20240015, 2008-2009.

10. References

Cao, J.; Spooner, D. P.; Jarvis, S. A. & Nudd, G. R. (2005). Grid load balancing using intelligent agents. *Future Gener. Comput. Syst.*, 21(1):135-149.

- Chunlin, L. & Layuan, L. (2006). Multieconomic agent interaction for optimizing the aggregate utility of grid users in computational grid. *Applied Intelligence*, Vol. 25, pp. 147–158.
- Smith, R. G. (1980). The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Transactions on Computers*, Vol. C-29, No. 12, pp. 1104–1113.
- Sandholm, T. (1993). An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations, *Proceedings of AAAI 93*, pp. 256–262.
- Weyns, D.; Bouck'e, N. & Holvoet, T. (2006). Gradient Field-Based Task Assignment in an AGV Transportation System, *Proc. of 5th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS2006)*, pp. 842 – 849.
- Parunak, H. V. D. (1987). Manufacturing experience with the contract net, *Distributed Artificial Intelligence*, (M. Huhns, Ed.), pp. 285–310, Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA.
- Schillo, M.; Kray, C. & Fischer, K. (2002). The Eager Bidder Problem: A Fundamental Problem of DAI and Selected Solutions, *Proc. of First Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS2002)*, pp. 599 – 606.
- Sugawara, T.; Kurihara, S.; Hirotsu, T.; Fukuda, K.; Sato, S. & Akashi, O. (2006). Total Performance by Local Agent Selection Strategies in Multi-Agent Systems, *Proc. of 5th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS2006)*, pp. 601–608.
- Sugawara, T.; Hirotsu, T.; Kurihara, S. & Fukuda, K. (2007). Performance Variation Due to Interference Among a Large Number of Self-Interested Agents. *Proceedings of 2007 IEEE Congress on Evolutionary Computation*, pp. 766-773.
- Falcone, R.; Pezzulo, G.; Castelfranchi, C. & Calvi, G. (2004). Why a cognitive trustier performs better: Simulating trust based Contract Nets, *Proc. of 3rd Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS2004)*, pp. 1394–1395.
- Xu, L. & Weigand, H. (2001). The Evolution of the Contract Net Protocol, *Proceedings of WAIM 2001*, LNCS 2118, pp. 257–264.
- Sandholm, T. & Lesser, V. (1995). Issues in automated negotiation and electronic commerce: Extending the contract net framework, *Proc. of First Int. Conf. on Multi-Agent Systems (ICMAS'95)*, The MIT Press: Cambridge, MA, USA, pp. 328–335.
- Gu, C. & Ishida, T. (1996). Analyzing the social behavior of contract net protocol, *Proc. of 7th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW 96)*, LNAI 1038, Springer-Verlag, pp. 116–127.



Evolutionary Computation

Edited by Wellington Pinheiro dos Santos

ISBN 978-953-307-008-7

Hard cover, 572 pages

Publisher InTech

Published online 01, October, 2009

Published in print edition October, 2009

This book presents several recent advances on Evolutionary Computation, specially evolution-based optimization methods and hybrid algorithms for several applications, from optimization and learning to pattern recognition and bioinformatics. This book also presents new algorithms based on several analogies and metafores, where one of them is based on philosophy, specifically on the philosophy of praxis and dialectics. In this book it is also presented interesting applications on bioinformatics, specially the use of particle swarms to discover gene expression patterns in DNA microarrays. Therefore, this book features representative work on the field of evolutionary computation and applied sciences. The intended audience is graduate, undergraduate, researchers, and anyone who wishes to become familiar with the latest research work on this field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Toshiharu Sugawara, Toshio Hirotsu, Satoshi Kurihara and Kensuke Fukuda (2009). Characteristics of Contract-Based Task Allocation by a Large Number of Self-Interested Agents, Evolutionary Computation, Wellington Pinheiro dos Santos (Ed.), ISBN: 978-953-307-008-7, InTech, Available from:
<http://www.intechopen.com/books/evolutionary-computation/characteristics-of-contract-based-task-allocation-by-a-large-number-of-self-interested-agents>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen