Adaptive Manager-side Control Policy in Contract Net Protocol for Massively Multi-Agent Systems

(Short Paper)

Toshiharu Sugawara[1] Satoshi Kurihara[2] Toshio Hirotsu[3] and Kensuke Fukuda[4]

[1]Dept. of Computer Science and Engineering
Waseda University
Tokyo 1698555, Japan
sugawara@waseda.jp

[3]Dept. of Information and Computer Sciences
Toyohashi University of Technology
Toyohashi, Aichi 441-8580, Japan
hirotsu@ics.tut.ac.jp

[2]Inst. of Scientific and Industrial Research
Osaka University
kurihara@ist.osaka-u.ac.jp

[4]National Institute of Informatics
Chiyoda, Tokyo 100-000, Japan
kensuke@nii.ac.jp

ABSTRACT

We describe a new adaptive manager-side control policy for the contract net protocol for a massively multi-agent system (MMAS). To improve overall performance of MMAS, tasks must be allocated to appropriate agents. From this viewpoint, a number of negotiation protocols were proposed in the MAS context, but most assume a small-scale, unbusy environment. We previously reported that, using contract net protocol (CNP), the overall efficiency could improve by an adequate control of degree of fluctuation in the awarding phase depending on the state of MMAS. In this paper, we propose the method to estimate these states from the bid values, which have hitherto not been used effectively. Then the manager-side policy flexibly and autonomously with some degree of fluctuation responsive to the estimated states is introduced. We also evaluate that our proposed CNP policy.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Experimentation, Performance

Keywords

Negotiation, Coordination, Multi-Agent Simulation

1. INTRODUCTION

Task allocation is an important issue for efficient and high-quality services in many applications of multi-agent systems (MAS). In particular, CNP and its extensions[5, 4, 8] have been widely used in certain applications because of the simplicity and good performance of CNP. However, most assume a small-scale, unbusy environment.

On the other hand, recent advances in many domains such as Internet services, sensor networks, pervasive computing and grid computing exhibit the need for massively MAS (MMAS), in which more than thousands of agents interact with one another. It is obvious that interference among agents is often observed in this kind of negotiation protocol if many managers have tasks to allocate to efficient contractors. In basic CNP, a contractor agent receives task announcements one by one. When many managers announce tasks, however, they have to wait a long time to receive a sufficient number of bids. In the original conception of CNP [5], the use of multiple bids was proposed for concurrently handling many announcements. If a contractor is awarded multiple tasks simultaneously, however, it may not be able to provide the quality or performance promised. In addition, a large number of tasks in a MMAS induce an excessive number of messages, which degrades the overall performance of an MAS.

In the preliminary experiments, we previously investigated the performance of a MMAS, especially the overall efficiency, when tasks were allocated by CNP with a variety of manager-side controls such as announcement restriction[6].This paper indicates the possibility that appropriate degree control of fluctuation in the award selection and appropriate control of announcement restriction in the announcement phase on the basis of the MMAS task load greatly improve overall performance. Note that “fluctuation” in the award selection means that managers do not always select the best contractor among bidding contractors.

Accordingly, we have now developed a flexible manager-side control policy, in the award phase, that improves overall system performance, using our previous results. First, we designed the control of fluctuation in award selection on the basis of the system’s task load and confirmed that this control policy could considerably improve overall performance. In actual open-system environment, however, it is almost impossible to acquire global information such as the task load of an entire MMAS. We will thus modify this control policy so that each manager can estimate the task loads of its local contractors in accordance with the received bid values with supplemental data, which were previously used only

Cite as: Adaptive Manager-side Control Policy in Contract Net Protocol for Massively Multi-Agent Systems (Short Paper), Toshiharu Sugawara Satoshi Kurihara Toshio Hirotsu and Kensuke Fukuda, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril,Portugal,pp.1433-1436.

to select awardees. We will compare the performance under this flexible control policy with those under others.

2. SIMULATION

2.1 Restricted CNP Model

Let $A = \{a_1, \ldots, a_n\}$ be a set of agents, $M = \{m_j\}(\subset A)$ be a set of managers that allocate tasks, and $C = \{c_k\}(\subset A)$ be a set of contractors that can execute allocated tasks if a contract is awarded. We assume that $M \cap C = \emptyset$ and $A = M \cup C$. Let us 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 in the Internet.

Next, the *restricted CNP*, which is modified for MMAS and used in our experiments, is defined as CNP in which (1) multiple bids and regret and no-bid messages are allowed and (2) manager m_j announces task T to contractors selected from S_{m_j} by its *announcement policy*, where S_{m_j} is the scope of m_j, the set of known contractors. *Regret messages* are sent in the award phase to contractors that have not been awarded the contract, while *no-bid messages* are sent to managers when contractors decide not to bid on an announced task. These messages avoids long waits for bid and award messages (e.g., [4, 9]). When manager m_j has task T, it allocates T to another agent in accordance with the restricted CNP. A contractor receiving this announcement sends m_j a bid message with a certain value called the *bid value*. Because timely responses are always of great concern in interactive services and real-time applications, we assume that all agents are rationally self-interested on the basis of efficiency and that their bid values are simply promised times for completing T. Finally, m_j selects a contractor, usually one that bid the best value, and sends an award message to the awardee allocating the announced task.

2.2 Simulation Model

We will briefly describe the simulation environment[6]. We set $|C| = 500$ and $|M| = 10000$ in our simulation model. (We assume that the contractor agents run on the Internet, providing services requested by manager agents, which correspond to clients.) The agents are randomly placed on the points of a 150 x 150 grid with a torus topology. Then, the Manhattan distance *dist*(a_i,a_j) between agents a_i and a_j is defined on this grid. We then set the communication cost (or delay) for messages from a_i to a_j. This cost is denoted by *cost*(a_i,a_j). The communication cost ranges between 1 and 14 (in *ticks*, the unit of time in the simulation), in proportion to the distance, *dist*(a_i,a_j). With every tick, tl tasks on average are generated, based on a Poisson distribution, in the simulation environment and randomly assigned to different tl managers, where tl is a positive number. Parameter tl is called the *task load* and denotes tl tasks per tick, or simply tl T/t.

For task T, we define the associated cost of T, *cost*(T). For convenience, we adjust this parameters so that contractor c_i can complete T in *cost*$(T)/e_{c_i}$ ticks, where e_{a_i} is an integer expressing the capability of a_i to process a given task. Since our experiments were designed simply to clarify the performance of restricted CNP in a MMAS, we assumed that all tasks would have the same cost, i.e., 2500. Instead, the abilities of contractors were initially assigned so that the values of *cost*$(T)/e_{c_i}$ (where $i = 1, \ldots, 500$) were *uniformly distributed* over the range $20 - 100$.

When contractor c_i is awarded a task, c_i immediately executes it if it has no other task. If c_i is already executing another task, the new task is stored in its queue whose size is maximally 20. The tasks in the queue are then executed in turn. Tasks that cannot be stored because of a full queue are dropped.

The bid value reflecting the state of contractor c_i is $|q_{c_i}| * (2500/e_{c_i}) + \alpha$, where α is the required time to complete the current task. In multiple bidding, c_i might have a number of uncertain bids for which results have not yet been received. These bids are not considered, however, because it is uncertain whether they will be awarded.

The *completion time* for each task is the elapsed time observed by the manager, from the time an award message with the allocated task was sent to the time a message indicating that the task has been completed is received. We define the *overall efficiency of a MAS* as the average completion time observed for all managers. The simulation data reported here are the mean values from three independent experiments using different random number seeds. The theoretical limit of processing capability, that is, the cumulative capability of all contractors of the MAS, in the three experiments ranged from 9.7 to 10.2 T/t, with an average value of 9.9 T/t.

3. FLUCTUATION IN AWARD SELECTION

3.1 Previous Results

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

We previously examined [6] how overall efficiency varies for n ranging from 5 to 50 and $0.1 \leq tl \leq 11$. We found that our expectation that a smaller n results in inefficiency in the MMAS because tasks may not be announced to capable agents applies only when the task load is extremely low. Because managers send more task announcement messages under RSP(n) for a larger n, we can predict task concentration in a few good contractors in busier environments, thus making the MMAS inefficient.

More importantly, we also tested a learning-based announcement restriction policy [6] in which each manager learns which contractors are more capable by observing completion times. While this restriction policy did not lead to better performance as expected, we did find that a small degree of fluctuation in the award phase considerably improved the overall performance.

After a task announcement, manager m_j receives bids from a number of contractors, $\{c_1, \ldots, c_p\}$. We denote the bid value from contractor c_i as $b(c_i)$; m_j selects an awardee, c_i, in accordance with the following probability:

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

Note that smaller bid values are better. This *probabilistic award selection* control in the award selection is denoted as PAS$_k$. The policy combined RCP(n) with PAS$_k$ is denoted by PAS$_k$+RSP(n). The larger the k, the smaller the degree of fluctuation; and PAS$_0$ and PAS$_\infty$ correspond to ‘random selection’ and ’no randomness’, respectively; so PAS$_\infty$+RSP(n) is identical to RSP(n). Variable k is called a *fluctuation factor* (FF), hereafter.

Figures 1 (a) to (c) show how overall performance varies under policies RSP and PAS$_k$+RSP; k ranges from 1 to 6 and the announcement number, n, is fixed at 20. Note that we show graphs for only $n = 20$ because the overall performance in this case is generally better than that in cases where fluctuation is introduced [6]. We set n to 20 for most of the experiments discussed in this paper, and the announcement number is often omitted if $n = 20$.

These figures illustrate that the RSP policy only results in better performance than PAS$_k$+RSP when the task load is less than 3

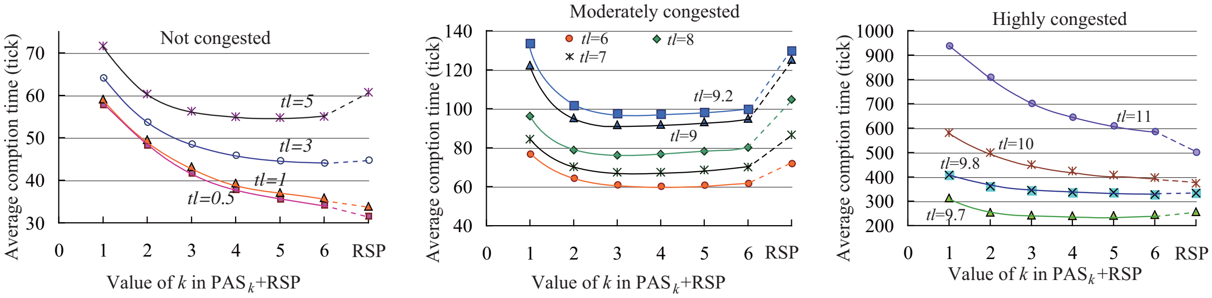

Figure 1: Completion times under PAS_k+RSP(20).

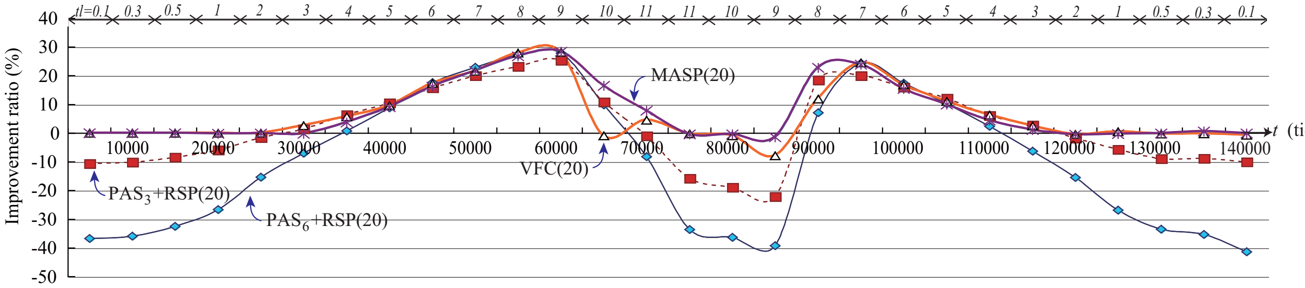

Figure 2: Ratio of completion times under PAS_k+RSP(20), VAS(20) and MASP(20).

(not so busy) or more than 10 (extremely busy). In other situations where $3 \le tl < 10$, some degree of fluctuation can result in much better performance, but the k value leading to the best performance depends on the task load. For example, when tl is close to three, a larger k is better, but when tl is greater than six, the value of k that expresses the best performance gradually approaches 3. However, if tl is larger than nine, the best k value swiftly approaches 6. This analysis suggests that the award selection policy must be sensitive to the task load of the MMAS.

3.2 Performance with Variable Task Load

Although Fig. 1 shows performance only when tl does not vary, the task load usually varies in real-world applications. We have now examined how the overall performance changes when the task load varies over time. The curves labeled "PAS_3+RSP(20)" and "PAS_6+RSP(20)" in Fig. 2 show the *improvement ratio* (%) with PAS_3+RSP and PAS_6+RSP with respect to RSP; that is,

$$\frac{\wp(\text{RSP}(n)) - \wp(\text{PAS}_k + \text{RSP}(n))}{\wp(\text{RSP}(n))} * 100,$$

where $\wp(p)$ is the overall performance under policy p. In this experiment, tl started at 0.1 and gradually increased to 11 for 5000 ticks and then returned to 0.1. The improvement ratios are plotted every 5000 ticks. The values of tl are also shown in the figure.

Figure 2 not only clarifies the results of our previous experiments described in Section 3.1 but also suggests that, in the awarding phase, selecting the policy flexibly on the basis of the task load can improve the overall efficiency by as much as 30%. To evaluate the flexible control, we introduce *variable fluctuation control* into policy PAS_k+RSP(n) in which FF is adaptively selected using the following *fluctuation control rule* (FCR):

$$\begin{array}{ll} k = \infty \text{ (i.e., RSP)} & \text{if } tl < 3 \text{ or } tl > 10, \\ k = 6 & \text{if } 3 \le tl \le 5 \text{ or } 9 < tl \le 10, \quad \text{(R1)} \\ k = 3 & \text{if } 5 < tl \le 9. \end{array}$$

This policy with FCR (R1), is called *variable awardee selection policy* and denoted by VAS(n). This FCR is induced from the experimental results shown in Fig. 1.

The overall performance under VAS(20) is also shown in Fig. 2. It indicates that, in general, VAS provides better performance than other policies using a fixed degree of fluctuation.

4. FLUCTUATION CONTROL BASED ON ESTIMATION

The major drawback of VAS is that it requires knowing the state of the system's task load, which is global information and usually unavailable in an open system like the Internet. To overcome this problem, we propose estimating the system's state from the bid values and supplemental information that is usually available from contractors.

With the proposed variable control, in the announcement phase, managers request the current queue length of each contractor as well as the bid values in their bid messages. This request is included in the bid specifications [5]. The managers can then estimate the task load (from their local viewpoints) using the queue lengths. First, suppose that manager m announces the task to n contractors, $c_1, \dots, c_n$, randomly selected from K_m. It then calculates

$$r = \sum_{1 \le i \le n} \frac{q(c_i)}{n}, \qquad (2)$$

where $q(c_i)$ denotes the queue length received from c_i. Ratio r is the average queue length of the contractors to which m made the announcement; it thus indicates how many tasks are simultaneously

Table 1: Ratios (%) of dropped tasks.

Time range	65000–70000	70000–75000	75000–80000
RSP	6.85	10.28	3.65
PAS_3+RSP	4.86	9.58	3.75
PAS_6+RSP	6.91	10.86	5.42
VAS(20)	6.48	10.70	2.93
MASP(20)	6.75	10.98	2.94

awarded. Then m select an awardee under VAS(n) but its FF is determined by the following FCR:

$$\begin{array}{ll} k = \infty & \text{if } r \leq 0.05 \text{ or } r > 2, \\ k = 6 & \text{if } 0.05 < r \leq 0.15 \text{ or } 1.2 < r \leq 2.0, \quad \text{(R2)} \\ k = 3 & \text{if } 0.15 < r \leq 1.2. \end{array}$$

This policy with task-load estimation is called *multiple-award-number-based award selection policy* and denoted by MASP(n). Note that FCR (R2) is derived by modifying FCR (R1) and these threshold values dividing the award policies are derived from our prior experiments in which we investigated the relationship between the average queue length and the task load, tl. For example, when $tl = 9$, the average queue length reported with bid messages is approximately 1.2.

The curve of the overall performance under MASP(20) are shown in Fig. 2. The performance under MASP(20) was slightly better than that under VAS(20). We believe that this originates from small variations in the task load due to randomness; randomness does not mean uniformity. Thus, while estimation based on data from local contractors may not be accurate, it can reflect the local variations of the task load in a timely manner; these small variations, which can occur anywhere, significantly affect performance. Given these results, we believe that MASP(20) is superior to VAS(20), because it is important to use all the capabilities of agents in MMAS when they are busy.

The better overall performance was not achieved at the expense of many dropped tasks. In our experiments, task drops were mainly observed only when t was in the range 65000 to 80000. The ratios between the observed numbers of dropped tasks and those of tasks generated in the simulation environment are shown in Table 1. Although the ratios under RSP_3+RSP(20) were slightly smaller and those under MASP(*) were slightly higher, no significant differences were found.

5. CONCLUSION

We have described a new flexible manager-side control policies for the contract net protocol that effectively uses the capabilities of all agents in an MMAS. The basic ideas of our control policy are the use of bid values from local contractors to estimate the local state of the MMAS and that managers in the CNP autonomously and adaptively changes (1) the degree of fluctuation in the award policies and (2) the number of announcements, from their local perspectives. We showed experimentally that a control policy responsive to the local task load has better performance than a naive CNP and a CNP with inflexible control policies, even though our policy does not use global information.

The results of our experiments show that flexible control of fluctuation in award selection policy, which is controlled by k in Eq. 1, strongly affects the overall efficiency of a MMAS; a little capriciousness by the manager when making an award would significantly improve the overall performance. However, this suggests that rational decisions in award selection do not always lead to the best results. For truly rational decision-making, agents must intentionally introduce fluctuated decision based on the system's states. The method proposed here provides one solution for this issue.

The communication bottlenecks in broadband network are shifting from the communication links to the server nodes, so the control of load balancing among servers in large-scale and worldwide systems is becoming critical. A more sophisticated control that fully utilizes the potential capability of systems is required for future network applications, and our research is aimed at to this requirement.

Finally, we must describe the importance of the system development methodology based on MAS simulation for large-scale applications. We obtained threshold numbers for switching the award-selection and announcement policies from simulations. One of the purposes of the simulations was to clarify the phenomenon, performance and operations of systems in an early state of development or when their actual testing and evaluation are impossible. The large-scale applications on the Internet are such systems; therefore, simulation-based performance tuning for MMAS should become more important. The importance of multi-agent-based simulation is now recognized and has been used for several applications such as design of pervasive computing applications [1], evaluation of high-performance cluster system [3], load-balancing in widely distributed systems [7] and explanation of phenomena occurring in markets [2]. Of course, we need further improvement in simulations so that they can accurately reflect real systems and, to this end, we should develop more reliable tools for simulating, for example, the Internet.

Acknowledgement: This research was supported in part by JST Grant 19500138.

6. REFERENCES

[1] T. Ishida, Y. Nakajima, Y. Murakami, and H. Nakanishi. Augmented Experiment: Participatory Design with Multiagent Simulation. In *IJCAI-07*, 2007.

[2] K. Izumi, T. Yamashita, and K. Kurumatani. Analysis of Learning Types in an Artificial Market. In *MABS2004*, LNCS 3415, pages 145–158, 2005.

[3] M. J. North and C. S. Hood. A Multi-agent Systems Model of High Performance Computing Cluster Users. In *MABS2004*, LNCS 3415, pages 99–113, 2005.

[4] T. Sandholm. An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. In *AAAI93*, pages 256–262, 1993.

[5] R. G. Smith. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Trans. on Comp.*, C-29(12):1104–1113, 1980.

[6] T. Sugawara, T. Hirotsu, S. Kurihara, and K. Fukuda. Performance Variation Due to Interference Among a Large Number of Self-Interested Agents. In *IEEE Cong. on Evolutionary Compututation*, pages 766–773, 2007.

[7] T. Sugawara, S. Kurihara, T. Hirotsu, K. Fukuda, S. Sato, and O. Akashi. Total Performance by Local Agent Selection Strategies in Multi-Agent Systems. In *AAMAS2006*, pages 601–608. ACM, May 2006.

[8] D. Weyns, N. Boucké, and T. Holvoet. Gradient Field-Based Task Assignment in an AGV Transportation System. In *AAMAS2006*, pages 842 – 849, 2006.

[9] L. Xu and H. Weigand. The Evolution of the Contract Net Protocol. In *Proceedings of WAIM 2001*, LNCS 2118, pages 257–264, 2001.