Reducing Communication Load on Contract Net by Case-Based Reasoning — Extension with Directed Contract and Forgetting —

Takuya Ohko
Anzai Laboratory. Department of Computer Science. Keio University.
3-14-1 Hiyoshi. Kohoku-ku. Yokohama 223, Japan
E-mail: ohko@aa.cs.keio.ac.jp

Kazuo Hiraki
PRESTO / Electrotechnical Laboratory. MITI.
1-1-4. Umezono. Tsukuba 305. Japan
E-mail: khiraki@etl.go.jp

Yuichiro Anzai
Anzai Laboratory. Department of Computer Science. Keio University,
3-14-1 Hiyoshi. Kohoku-ku. Yokohama 223. Japan
E-mail: anzai@aa.cs.keio.ac.jp

Abstract

This paper describes the communication load reduction on the task negotiation with Contract Net Protocol for multiple autonomous mobile robots. We have developed LEMMING. a task negotiation system with low communication load for multiple autonomous mobile robots. For controlling multiple robots, Contract Net Protocol(CNP) is useful. but the broadcast of the Task Announcement messages on CNP tends to consume much communication load. In order to overcome this problem LEMMING learns proper addressees for the Task Announcement messages with Case-Based Reasoning(CBR) so as to suppress the broadcast. The learning method is called Addressee Learning. In this paper. we extend LEMMING with 'directed contract' to reduce the communication load more effectively. Moreover. we extend LEMMING with 'forgetting' to restrict the number of cases. since it is impossible to have enough memory to keep all the cases. The efficiency of LEMMING is evaluated in a simulated multi-robot environment to show that these extensions are effective for LEMMING.

Introduction

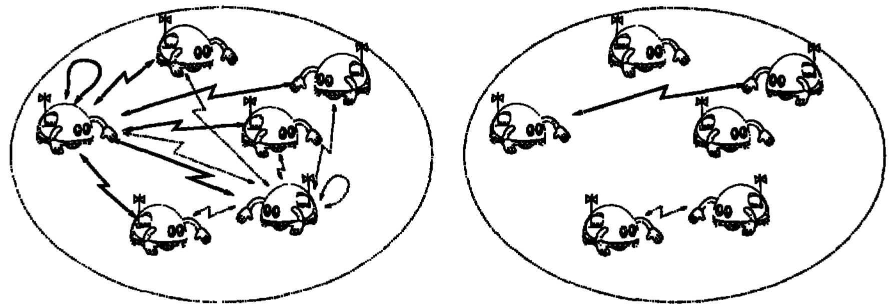
This paper extends LEMMING(Ohko. Hiraki. & Anzai 1993; 1995) with directed contract and forgetting. LEMMING is a task negotiation system with low communication load. LEMMING learns to reduce communication load for task negotiation using Contract Net Protocol(CNP)(Smith 1980) with Case-Based Reasoning(CBR)(Kolodner 1993).

It is difficult for autonomous mobile robots to use high-speed and wide-band communication lines. compared with desktop-computers and software-agents. And the ability of the robots tend to change dynamically by many facts like their location. their circumstances. and their damage. Thus. the adaptive reduction of communication load is an important issue for cooperation among multiple autonomous mobile robots. and it is worth spending much computation cost.

In general. messages handled by CNP include Task Announcement messages. Bid messages. Award messages. and Report messages. It is observed that a high proportion of the communication load comes from the Task Announcement broadcasting. To reduce this load. Smith proposes the *focused addressing*(Smith 1980). But the knowledge for the focused addressing must be provided by human experts in advance. Thus. we proposes a technique to let robots acquire such knowledge by CBR.

In previous approaches. like CNET(Smith 1980). the information gathered from a contract is not fully utilized and is disposed after each negotiation. Thus. this paper presents an attempt to extract useful knowledge from the messages in order to reduce the total number of message exchanges. CBR is employed to infer the most suitable robot to carry out the task specified in a particular Task Announcement message. This technique of learning with CBR is called *Addressee Learning*. LEMMING. residing in each robot. is developed to facilitate the task negotiation with CNP and Addressee Learning. LEMMING learns to reduce the communication load as shown in Figure 1.

In this paper. we extend LEMMING with the *directed*



(a) Before Learning (b) After Learning

Figure 1: Addressee Learning

contract(Smith 1980) to reduce the messages more effectively. The directed contract is the technique to omit task announcement and bidding processes on a task negotiation.

Moreover, we extend LEMMING with *forgetting* to restrict the number of cases. The capacity of CBR partly depends on the efficiency of the similarity function and the number of the cases(Wess & Globig 1993). But it is impossible to have enough memory to keep all the cases, and it takes much computation load according to the number of the cases.

The performance of LEMMING is evaluated in a simulated multi-robot environment, where requests on robots consist of serving various dishes. Each robot has different dish. Thus, on receiving a request for service, a robot is required to negotiate a task with its peer robots if it does not have the requested dish. In this paper, communication load for a message is assumed as the number of the robots addressed by the message. The focus of this evaluation is on the reduction of the communication load.

The definition of communication load is following:

$$ComLoad(M) = \begin{cases} NumAddr(M) & \text{(multicast or point-to-point)} \\ AllAddr & \text{(broadcast)} \end{cases} \quad (1)$$

M is the message to calculate communication load. $NumAddr(M)$ is the number of the robots addressed by M. $AllAddr$ is the number of the robots in the environment.

LEMMING

LEMMING is a task negotiation system with the low communication load for multiple autonomous mobile robots. LEMMING negotiates tasks with CNP. LEMMING reduces the communication load adaptively by Addressee Learning, a learning method to reason proper addressees for Task Announcement messages.

Architecture

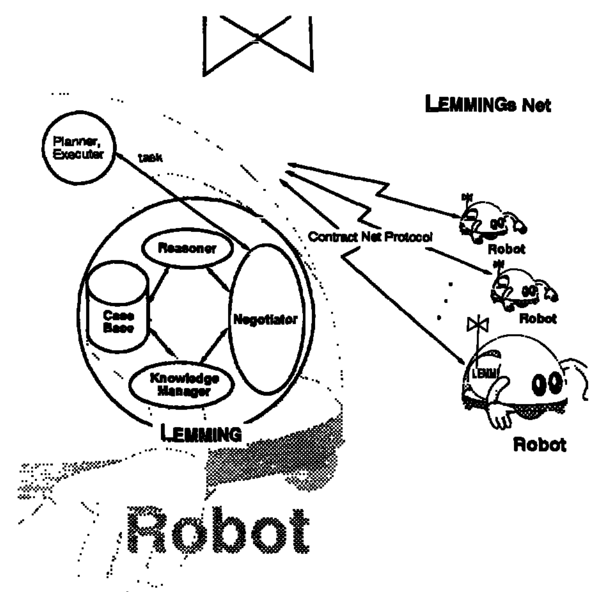


Figure 2: Architecture of LEMMING

Figure 2 shows the architecture of LEMMING. Each LEMMING controls a robot for negotiating tasks with CNP. Such robots with LEMMING composes LEMMINGs Net. LEMMING consists of Negotiator, Reasoner, Knowledge Manager, and Case Base (Figure 2). Negotiator negotiates tasks with CNP. Reasoner reasons proper addressees for Task Announcement messages with Addressee Learning. Knowledge Manager manages cases and status of the negotiation process in Case Base.

Contract Net Protocol

Contract Net Protocol(CNP)(Smith 1980) is a useful method for the task negotiation among multiple robots. CNP consists of Task Announcement message, Bid message, Award message, and Report message.

When a task distributer called *manager* wants to allocate a task, the manager generates and broadcasts Task Announcement message for the task. The robot receiving the message evaluates it, and if the robot can execute the task, the robot evaluates the performance for the task execution, and returns Bid message with the performance. The manager selects the bid with the best performance, and appoints the sender of the bid as a *contractor*. The manager sends Award message to the contractor to execute the task. On finishing the task, the contractor returns Report message to the manager.

Case-Based Reasoning

Case-Based Reasoning(CBR)(Kolodner 1993) is a learning method to solve a problem with past cases. A case consists of a problem and its solution. When a reasoning system is input a new problem, it searches the cases which include the similar problems to the new problem, and makes an answer of the cases. The problem and the answer are gathered and stored as a case to the Case Base incrementally. A case can be expressed with raw informations, so it can be given, read, and corrected by human.

Addressee Learning

Addressee Learning is a learning method to reason proper addressees for Task Announcement messages with CBR. On a general task negotiation system with CNP, a manager broadcasts Task Announcement message. But the broadcast tends to consume much communication load. Thus, LEMMING suppresses the broadcast with Addressee Learning. Moreover, Addressee Learning can deal with immigration, emigration, and capacity change of the robots.

Directed Contract

We can reduce more communication load by extending LEMMING with the *directed contract*(Smith 1980). The directed contract is the technique to omit task announcement and bidding processes and send Award message to a contractor directly. The contractor is reasoned by Addressee Learning. When the reasoned addressee is determined that it seems to be the best robot for the task, the robot of the addressee becomes the contractor. If the contractor could not execute the task, it should return Nack message so as to let the manager do task announcement.

Forgetting

Moreover, we extend LEMMING with *forgetting* to restrict the number of cases.

The capacity of CBR partly depends on the efficiency of the similarity function and the number of the cases(Wess & Globig 1993). But it is impossible to have enough memory to keep all the cases, and it takes much computation cost according to the number of the cases. Then, it is important to generalize and forget unnecessary cases. Thus, LEMMING forgets the cases to restrict the memory. But it is yet future work to discover proper standard to decide which cases are important. In this paper, we define the maximum number of the cases as $MaxCaseNum$, and if the number of the cases gets over the $MaxCaseNum$, the oldest cases should be forgotten.

And it is also important to avoid to forget the cases whose tasks are on the negotiation, so each case should have the value which shows the status of the negotiation process.

Data Description

Task Description: Task T is a list of attribute-value pairs described as follows:

$$T \;:\; (A_1V_1, A_2V_2, \ldots, A_{m-1}V_{m-1}, \quad \texttt{taskid}\ :taskid) \tag{2}$$

Where $A_k(k = 1 \ldots m-1)$ is the kth attribute, and $V_k(k = 1 \ldots m-1)$ is the value corresponding to A_k. The pair of `taskid:` and *taskid* is an identifier of the task. The identifier should be added when the task is generated.

Message Description: Figure 3 shows the descriptions of the messages for the task negotiation. Task Announcement message, Bid message, Award message, and Report message are used for each contract.

Task Announcement message: (TYPE:ANNOUNCE, CONTRACT:*contract*, FROM:*from*, TO:*to*, TASK:*task*, BID-SPECIFICATION:*bid-specification*, ELIGIBILITY-SPECIFICATION:*eligibility-specification*, EXPIRATION-TIME:*expiration-time*)

Bid message:
(TYPE:BID, CONTRACT:*contract*, FROM:*from*, TO:*to*, ESTIMATED-PERFORMANCE:*estimated-performance*)

Award message: (TYPE:AWARD, CONTRACT:*contract*, FROM:*from*, TO:*to*, TASK:*task*, REPORT-SPECIFICATION:*report-specification*, RESULT-SPECIFICATION:*result-specification*)

Nack message: (TYPE:NACK, CONTRACT:*contract*, FROM:*from*, TO:*to*, RESULT:*result*)

Report message: (TYPE:REPORT, CONTRACT:*contract*, FROM:*from*, TO:*to*, RESULT:*result*, ACTUAL-PERFORMANCE:*actual-performance*)

Figure 3: Message Description

contract is an identifier of the negotiation corresponded to by the message. *from* is the sender of the message. *to* is the addressee of the message. *task* is a task to be executed. *bid-specification* is the formula to estimate the performance for the task. *eligibility-specification* is a condition whether the receiver can return a bid or not. *expiration-time* is a time the manager waits for the bids. *estimated-performance* is a scalar value of the *estimated* performance for the task. *report-specification* is similar to the *bid-specification*.

but it should calculate *actual* performance. *result-specification* is the formula for calculating result. *result* is the result of the task execution. *actual-performance* is a scalar value which represents an actual performance for the task execution of the contractor at that time.

Case: A case is described in Figure 4.

case: (CASEID:*case-id*, TASK:*task*, ANNOUNCE:*announce-list*, BID:*bid-list*, AWARD:*award-list*, NACK:*nack-list*, REPORT:*report-list*, GENTIME:*gentime*, RCOUNT:*RCount*, STATUS:*status*)

Figure 4: Description of Case

case-id is the identifier of the case. *task* is a list of attribute-value pairs, and is the task of the case. *announce-list*, *bid-list*, *award-list*, *nack-list*, *and report-list* are lists of Task Announcement messages, Bid messages, Award messages, Nack messages, and Report messages dealt with the case respectively. *RCount* is how many counts the Reasoner called on the negotiation. *status* is the status of the negotiation process. The values of *status* are **announcing**, **executing**, or **finish**.

Learning Process

Figure 5 shows the task negotiation process of LEMMING. This subsection focuses on the process of Addressee Learning in the task negotiation process.

Negotiator When a new task is input to LEMMING, Negotiator starts to negotiate the task with CNP. At first, Negotiator calls Reasoner to decide which robots are proper for the task. If the directed contract is enabled and the best-selected robot seems reliable(the value of the suitability, described later, of the robot is good), the Negotiator determines the robot as the contractor, and sends Award message to the contractor directly. Then if Nack message is returned since the contractor could not execute the task, the Negotiator calls Reasoner again and commits a task announcement.

Else, the Negotiator sends Task Announcement message to the selected robots, waits for the Bid messages, and sends Award message to the contractor which is a robot with the best performance for the task.

If the Reasoner can not reason such robots or the Negotiator could not get any bid at the first announcement, the Negotiator broadcasts the Task Announcement message.

The other processes like bidding and reporting are almost the same as the ordinary contract net systems.

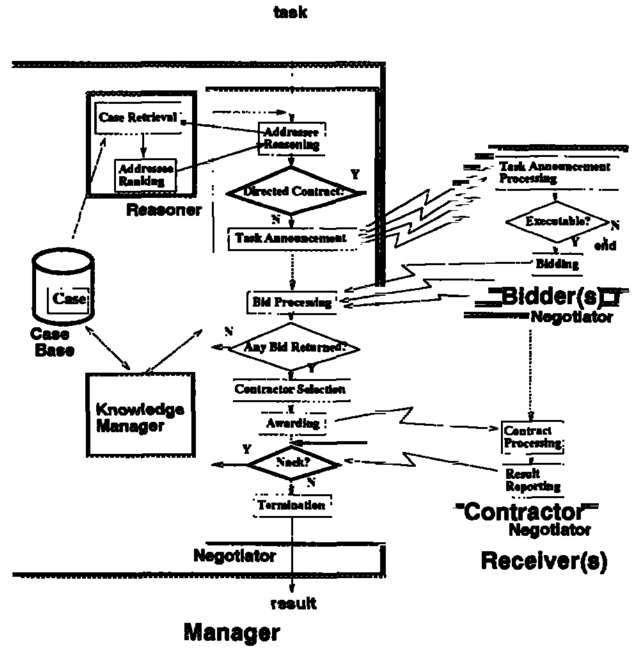


Figure 5: Process of LEMMING

Reasoner Reasoner searches such cases that include similar tasks to the input task. The function to evaluate the similarity is following:

$$Similarity(T_1, T_2) = \sum_{i=1}^{m}\sum_{j=1}^{n} Dist(A_{1i}, V_{1i}, A_{2j}, V_{2j}) \quad (3)$$

$$T_1 : (A_{11}V_{11}, \cdots, A_{1m}V_{1m}) \quad (4)$$

$$T_2 : (A_{21}V_{21}, \cdots, A_{2n}V_{2n}) \quad (5)$$

$$Dist(A_{1i}, V_{1i}, A_{2j}, V_{2j}) = W(A_{1i}, A_{2j})Equal(V_{1i}, V_{2j}) \quad (6)$$

$$W(A_{1i}, A_{2j}) = \begin{cases} 1 & \text{if } A_{1i} = A_{2j} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$Equal(X_1, X_2) = \begin{cases} 1 & \text{if } X_1 = X_2 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Where T_1 and T_2 are the tasks to evaluate similarity. According to the similarities, the Reasoner sorts the cases, and selects certain similar cases. And the Reasoner calculates "*suitabilities*" of the robots from the similar cases:

$$suit(R) = \max_{i,j} perf(m_R(Sc_i, j)) \quad (9)$$

$suit(R)$ is the suitability of robot R. Sc_i is a ith similar case. $m_R(Sc_i, j)$ is a jth Bid/Report message from robot R in Sc_i. $perf(m_R(Sc_i, j))$ is a performance value of $m_R(Sc_i, j)$. The suitability is a maximum performance of robot R extracted from the Bid and Report messages in the similar cases. Then the Reasoner picks up some best robots according to the suitabilities, and returns a list of the robot-suitability pairs to the Negotiator.

Knowledge Manager Knowledge Manager makes cases. Reasoner refers the cases for Addressee Learning. And Negotiator also refers the cases to know the status of the negotiation processes. Thus, we can suppose that the cases are the workareas of past negotiations. The LEMMING does not delete the workareas but uses them for Addressee Learning.

The Knowledge Manager also commits forgetting old cases. When the Knowledge Manager makes a case and the number of the cases gets over *MaxCaseNum*, Knowledge Manager deletes the oldest case according to the value of *gentime*. To avoid to forget the cases whose tasks are on the negotiations, the values of the *status* of the forgotten cases should be `finish`.

Simulated Environment

This section shows "Dinner Environment" (Figure 6), a simulated environment for evaluating LEMMING.

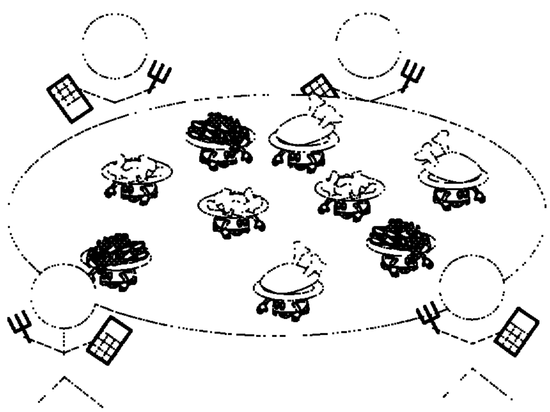


Figure 6: Dinner Environment

Mobile robots carry dishes, and they know what they carry. The positions of the guests are situated, and each robot knows which guest is where.

A guest selects a dish, and inputs a task sequence with a handy remote controller to a randomly selected robot. The robot negotiates the task to the robot which is the fastest to carry the dish to the guest.

The task is expressed as a list of attribute-value pairs. For example, a task:"Serve Chicken to Guest G0" is described as `(command:serve dish:Chicken to:G0 taskid:T0-1)`.

The definition of performance is following:

$$Performance(R, t, T) = \begin{cases} \max(0, C - (EndTime(T) - t)) & (R \text{ can execute } T) \\ -1 & (R \text{ can not execute } T) \end{cases} \quad (10)$$

T is the task. R is the robot which evaluate the performance of T. t is the time when T is allocated to R. $EndTime(T)$ is the time when R finished T. C is a constant which is enough large to $(EndTime(T) - t)$. We define that larger performance is better, and that (-1) means that R cannot execute T.

Evaluation

There are nine mobile-robots (`R0, R1, R2, R3, R4, R5, R6, R7, R8`) and four human guests (`G0, G1, G2, G3`) in the environment. The speed of the robots is the same. Each robot serves dish. There are three kinds of dishes (`Chicken, Salad, Pasta`). The robots know what they carry, but do not know the capacity of the other robots at first.

The values of the task are selected randomly, but the value of the taskid is generated as to be distinguishable from other values of taskid. Then, the number of the combination of the values except that of taskid is $1(\texttt{command :}) \times 3(\texttt{dish :}) \times 4(\texttt{to :}) = 12$ patterns. The task is commanded to a randomly selected robot.

Each robot carries different dishes : robot `R0`, `R1` and `R2` carry `Chicken`, `R3` `R4` and `R5` carry `Salad`, and `R6` `R7` and `R8` carry `Pasta`(Table 1). Thus, on receiving a request for service, a robot is required to negotiate a task with its peer robots.

Table 1: Relation of Robots-Dishes

Robot	Dish
R0,R1,R2	Chicken
R3,R4,R5	Salad
R6,R7,R8	Pasta

This evaluation focuses on the reduction of the communication load and the waiting time for the dish. Thus, we extract the communication load for all the messages and the waiting time per task by each robot in each simulation-loop; and show their average as the result of the evaluation. The communication load of each message is defined as the formula (1); and the waiting time is defined as $(EndTime(T) - t)$, a part of the formula (10). A simulation is finished when each robot has negotiated 40 tasks. The number of the simulation-loop is 50. The unit of waiting time is *clock*. It takes about 10 clocks for the robots to move from side to side on the table. The communication medium is properly multiplied so that messages should not be conflicted nor delayed.

As comparison systems with LEMMING, *broadcast-based contract net system* is evaluated. The broadcast-based contract net system always broadcasts task announcement messages. And the LEMMING with directed contract, and the LEMMINGs of which the number of the cases are restricted are also evaluated.

Directed Contract

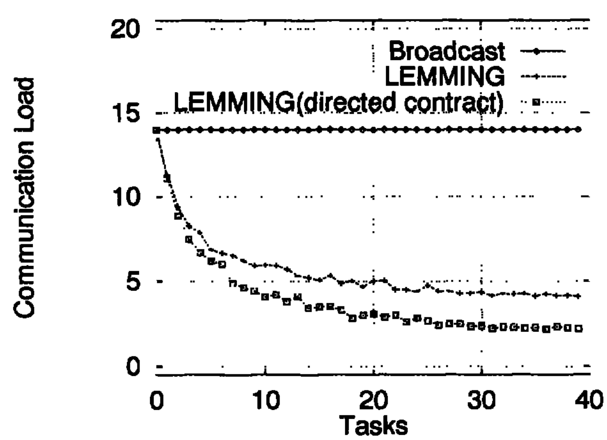


Figure 7: Communication Load(directed contract)

Table 2: Total Communication Load(After Learning)

	Ann.	Bid	Award	Nack	Rep.	total
Broadcast	9	3	1	0	1	14
LEMMING	1	1	1	0	1	4
LEMMING(d.c.)	0	0	1	0	1	2

Figure 7 shows the average total communication load. This is plotted against the number of the experienced tasks with broadcast-based contract net system, LEMMING, and LEMMING with directed contract. Table 2 shows the total communication load of each system after learning. The after learning is such a condition that LEMMING has become to be able to reason proper addressees always successfully.

Thus, Figure 7 and Table 2 suggest that LEMMING can reduce the communication load for the task negotiation, and the extension of the directed contract can reduce more communication load.

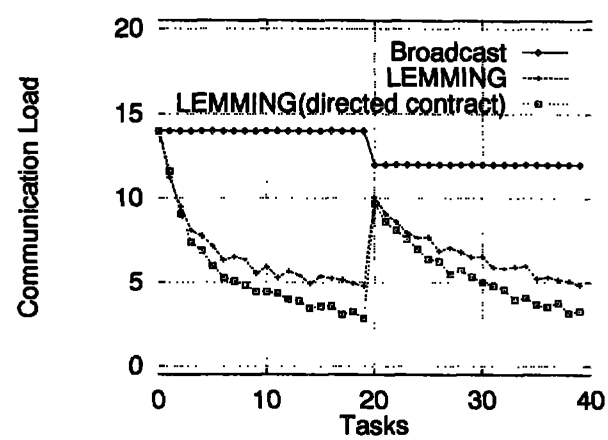


Figure 8: Communication Load(directed contract,empty)

Figure 8 also shows the average total communication load, but the dishes of R1, R2, R4, R5, R7, and R8 become empty when each robot negotiates 20 tasks. This is plotted against the number of the experienced tasks with broadcast-based contract net system, LEMMING, and LEMMING with directed contract.

Figure 8 shows the adaptability of LEMMING. This figure suggests that LEMMING keeps its adaptability as if the directed contract is extended.

Forgetting

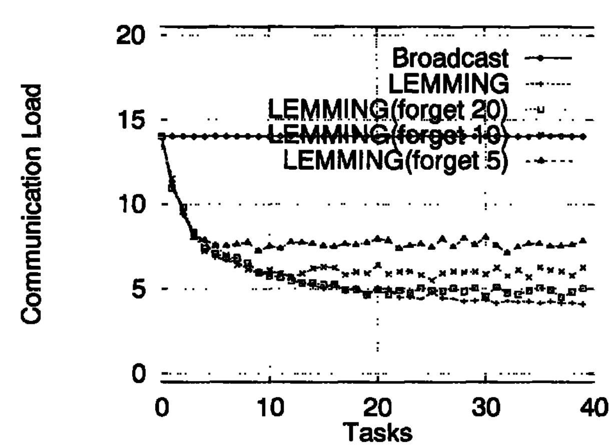


Figure 9: Communication Load(forget)

Figure 9 shows the average total communication load. This is plotted against the number of the experienced tasks with broadcast-based contract net system, LEMMING, and LEMMINGs with forgetting. Table 3 shows the parameter of the forgetting.

Table 3: Parameter of Forgetting

	$MaxCaseNum$
(null)	100
forget 20	20
forget 10	10
forget 5	5

Thus, Figure 9 suggests that LEMMING can reduce the communication load even if the number of the cases is restricted, and that LEMMING can reduce the load the more efficiently if LEMMING has the more number of the cases. The curves of average communication load is even after such the number of the experienced tasks that is the same as $MaxCaseNum$. Consequently, the more $MaxCaseNum$ is better. However, the size of the case LEMMING deals with is unfixed, it is difficult to decide $MaxCaseNum$ in advance even if the size of the memory is known. It may be important to change $MaxCaseNum$ according to the average size of the cases.

Combination of Directed Contract and Forgetting

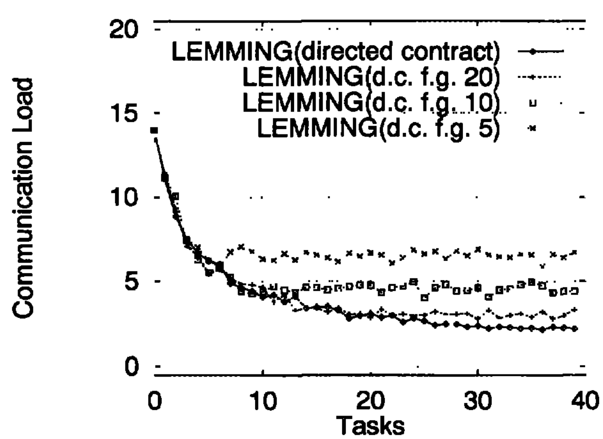


Figure 10: Communication Load(directed contract,forget)

Figure 10 shows the average total communication load. This is plotted against the number of the experienced tasks with broadcast-based contract net system, LEMMING, and LEMMINGs with directed contract and forgetting. Table 4 shows the parameter of the forgetting.

Table 4: Parameter of Forgetting(with Directed Contract)

	$MaxCaseNum$
directed contract	100
d.c. f.g. 20	20
d.c. f.g. 10	10
d.c. f.g. 5	5

Thus, Figure 10 suggests that LEMMING works well with both directed contract and forgetting. The curves of average communication load is also even after such the number of the experienced tasks that is the same as $MaxCaseNum$ like Figure 9.

Efficiency of Task Execution

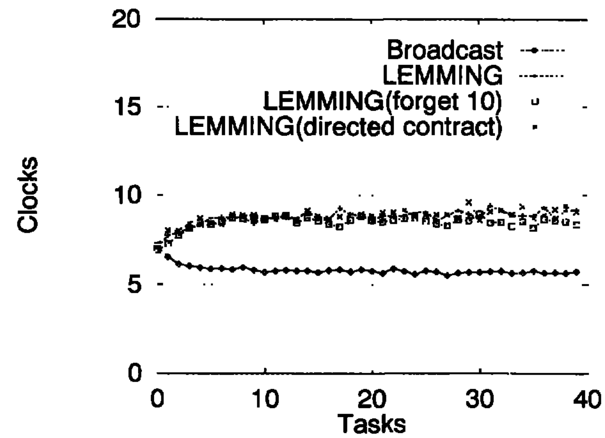


Figure 11: Efficiency of Task Execution

Figure 11 shows the average efficiency of the task execution, average waiting time for the dish. This is plotted against the number of the experienced tasks with broadcast-based contract net system, LEMMING, LEMMING whose $MaxCaseNum is 10$, and LEMMING with directed contract.

Thus, Figure 11 suggests that Addressee Learning makes the efficiency of the task execution worse. Although the broadcast-based contract net system always explores the best robot at each the negotiation, LEMMING does not. The directed contract does not relate the efficiency but the forgetting does, since the forgetting sometimes causes broadcast and is forced to explore the best robot.

To improve the efficiency of the task execution, LEMMING have been extended with Message Interception(Ohko, Hiraki, & Anzai 1996). Message Interception is the technique to allow a robot return Bid message for the successfully received Task Announcement messages even if the messages are not addressed to the robot. Message Interception improves the efficiency of the task execution with a few additional communication load.

Related Work

LEMMING can not only reduce communication load but also make human-robot interface simple and friendly(Ohko & Anzai 1995); since the LEMMING negotiates tasks adaptively instead of human. and it is less need to send human information about the situation.

Ramamritham and Stankovic apply focused addressing on multiprocessor systems(Ramamritham & Stankovic 1984). But their method needs computation time for each task. Then it is difficult for their method to apply for heterogeneous robots in dynamically changing environments. Shaw and Whinston use Classifier System and GA(Shaw & Whinston 1989). But the expression of the knowledge for focused addressing on their method is difficult to be read by human.

Conclusion

This paper described the communication load reduction on task negotiation with Contract Net Protocol(CNP) for multiple autonomous mobile robots.

We have been developing LEMMING. a task negotiation system with the low communication load for multiple autonomous mobile robots. LEMMING learns proper addressees for the Task Announcement messages on CNP with Case-Based Reasoning(CBR) so as to suppress the broadcast. The learning method is called Addressee Learning. The performance of LEMMING was evaluated in a simulated multi-robot environment. The evaluation shows that LEMMING can reduce the communication load adaptively, that the directed contract improves the efficiency of the communication load reduction. and that the forgetting works on LEMMING.

Acknowledgments

This research is supported by JSPS Research Fellowships for Young Scientists.

References

Kolodner. J. L. 1993. *Case-Based Reasoning.* Morgan Kaufmann.

Ohko. T.. and Anzai. Y. 1995. A user friendly interface for task assignment on multiple robots by contract net negotiation with case-based reasoning. In *Abridged Proc. of the 6th International Conference on Human-Computer Interaction(HCI'95).* 12.

Ohko. T.; Hiraki. K.; and Anzai. Y. 1993. LEMMING: A learning system for multi-robot environments. In *Proc. of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems. Yokohama. Japan(IROS'93).* volume 2. 1141–1146.

Ohko. T.; Hiraki, K.; and Anzai, Y. 1995. Learning to reduce communication cost on task negotiation among multiple robots. In *Working notes of the IJCAI-95 Workshop on Adaptation and Learning in Multiagent Systems.* 65–70.

Ohko. T.; Hiraki. K.; and Anzai, Y. 1996. Addressee learning and message interception for communication load reduction in multiple robot environments. In *Working notes of the ECAI-96 Workshop on Learning in Distributed Artificial Intelligence Systems.*

Ramamritham, K., and Stankovic. J. A. 1984. Dynamic task scheduling in hard real-time distributed systems. *IEEE Software* 65–75.

Shaw. M. J.. and Whinston. A. B. 1989. Learning and adaptation in distributed artificial intelligence systems. In Gasser, L.. and Huhns. M. N.. eds.. *Distributed Artificial Intelligence Volume II.* 119–137. Morgan Kaufmann.

Smith. R. G. 1980. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers* C-29(12):357–366.

Smyth. B., and Keane, M. T. 1995. Remembering to forget a competence-preserving case deletion policy for case-based reasoning. In *Proc. of IJCAI-95*, 377–382.

Wess. S., and Globig. C. 1993. Case-based and symbolic classification — a case study —. In Wess. S.; Althoff. K.-D.; and Richer. M. M., eds.. *Topics in Case-Based Reasoning(EWCBR-93)*, 76–91.