21

MULTI-AGENT BASED ROBUST SCHEDULING FOR AGILE MANUFACTURING

Toshiya Kaihara* and Susumu Fujii**

** Graduate School of Science and Technology, Kobe University*

*** Department of Computer and Systems Engineering, Kobe University*

{kaihara, fujii}@cs.kobe-u.ac.jp, JAPAN

Scheduling is the problem of allocating resources to alternate possible uses over designated period of time. Contract mechanisms use prices derived through distributed bidding protocols to determine an allocation. A robust manufacturing scheduling protocol based on multi-agent paradigm is proposed in this paper. We define all the manufacturing units, such as machines and jobs, as economic agents, which conduct strong robustness against practical manufacturing conditions. A contract mechanism with bidding protocol corresponding to market structure is proposed. We study the dynamism of the proposed scheduling protocol, and confirm its validity by several simulation experiments.

1. INTRODUCTION

There is a growing recognition that current manufacturing enterprises must be agile, that is, capable of operating profitably in a competitive environment of continuously changing customer demands. It is important to realise the total productivity, efficiency and flexibility in factory management under such an environment. Scheduling problem is one of the major issues on the effective manufacturing management in the agile environment. Distributed autonomous manufacturing control is recently introduced, and several distributed scheduling methodologies are proposed by several researchers (Sugimura, 1994; Kaihara, et al., 1997; Kaihara, et al., 1998; Rabelo, et al., 1998).

Recently the utilisation of multi-agent system in manufacturing application increases, such as robot assembly planning, multiple mobile robot control and so forth (Deneubourg, 1991). Multi-agent paradigm has several characteristics to overcome the current scheduling problems in the agile manufacturing environment (Ishida, 1995). The capacity of a single scheduling rule to achieve efficiently for any length of time will be in doubt - only autonomous and coordinated paradigm will succeed (Walsh, 1998). By a social goal we mean a goal that is not achievable by any single agent alone but is achievable by a group of agents. The key element that distinguishes social goals from other goals is that they require cooperation; social goals are not, in general, decomposable into separate subgoals that are achievable independently of the other agent's activities. In other words, any agent cannot simply proceed to perform its action without considering what the other agents are doing. The attainment of social goals appears to require a coordination of agent actions (Kaihara, 1996).

Solving scheduling problems with and for distributed computing systems presents particular challenges attributable to the decentralised nature of the computation. System modules represent independent entities with conflicting and competing scheduling requirements, who may possess localised information relevant to their utilities in such an environment. To recognise this independence, we treat the modules as agents, ascribing each of them autonomy to decide how to deploy resources under their control in service of their interests. It is assumed that the agents can communicate with messages in which they may convey some of their private information.

Our goal is to propose a decentralised universal scheduling concept which is robust against several environmental changes despite its simple architecture. We present a new distributed scheduling concept based on the Contract Net Protocol (CNP) (Smith, 1980), which is one of the negotiation protocols taking the metaphor of market behaviour. The task allocation is realised by a negotiation process between agents called manager that has tasks to be executed and agents called contractor that may be able to execute those tasks. These agents negotiate each other by exchanging mutual messages. In the negotiation, decision-making criteria are necessary for agents to select a contracting partner to send a message. Therefore, to decide of appropriate criteria is very important because the criteria affect the system performance (Ishida, 1996).

In this paper, after a brief explanation of CNP, the criteria on basis of utility in each agent are formalised for the decentralised manufacturing scheduling. We demonstrate the applicability of the CNP based scheduling concept by simulation experiments. Finally it is proved the proposed concept can provide several advantages on decentralised manufacturing scheduling.

2. COOPERATIVE SCHEDULING CONCEPT

2.1 Contract Net Protocol (CNP)

The Contract Net Protocol is based on multi-agent paradigm, which explored a distributed approach to problem-solving using a "negotiated" mutual selection process for task allocation. A CNP based problem-solver is a collection of nodes in manager and contractor roles. A top level task is allocated to a manager node, which generates subtasks and issuing task announcements for them to some subset of the potential contractor nodes (a process called task announcement). Contractors bid on tasks they desire and are qualified for.

The manager selects the highest rated bid, and allocates the task to that contractor, possibly monitoring the contractor's progress toward solution. When several contractors supply final reports of individual subtask results, the manager is responsible for integrating and supplying a global solution. The manager-contractor relation is recursive, and nodes simultaneously may be managers for some tasks and contractors for others.

2.2 Scheduling problem

Generally scheduling problem involves several criteria, and a solution that minimises optimality of all the criteria does not exist. It is required for scheduling algorithm to search a Pareto optimal solution. We treat two types of general criteria about scheduling in this paper as a basic study, f: lead time and g: throughput.

Notations

Let J_i denote job i (i=1,...,N), M_j machine j (j=1,...,L), K_i the number of operations in job i, O^j_{ik} operation i (i=1,...,N), j (j=1,...,L), k (k=1,...,K_i), TO^j_{ik} process time and $STO^l_{ik}O^l_{i'k'}(O^l_{ik} \neq O^l_{i'k'})$ set-up time between O^l_{ik} and $O^l_{i'k'}$. We introduce the following assumptions in our scheduling model:

- Operational order in job J_i is given and fixed.
- Machine M_j deals with one product at the same time.
- Process time TO^j_{ik} varies and depends on machine M_j.

Then the objective function in our scheduling problem is described as

$$\min\left(\left(\sum_{i=1,N} f_{Ji}\right)/N\right) \cap \max(g_{Mj'}) \tag{1}$$

where

f_{Ji} : lead time of job j_i

$g_{Mj'}$: throughput of machine $M_{j'}$ $\quad g_{Mj'} = \min(\forall g_{Mj(j=1,L)})$

Generally, these criteria, lead time and throughput, are in trade-off relationship. Shortening lead time requires small WIP (Work In Process) size, that causes small throughput in the production. Conventional scheduling methodologies apply heuristic rule based approach, but they can't handle such a trade-off relationship appropriately.

2.3 Machine agent

Machine agents try to process as many products as possible so as to maximise the individual throughput. Their utility function is defined as follows:

$$U^l_{machine} = \max \sum_{i=1,N}\sum_{k=1,Ki} Ocomp^l_{ik} \tag{2}$$

where

$Ocomp^l$: the number of completed operations in machine l

Machine agents adopt the following scheduling policy to satisfy the utility function described in (2);

$$Select^l_{machine} = \exists i.(\min(TO^l_{ik} + STO^l_{i'k'}O^l_{ik})) \tag{3}$$

where the operation $O^l_{i'k'}$ is followed by O_{ik} consecutively in machine l.

2.4 Job agent

Job agents try to proceed as fast as possible so as to minimise the individual lead time. Their utility function is defined as follows:

$$U^n_{job} = \min \sum_{k=1,Ki} TO_{nk} \tag{4}$$

where

TO_{nk}: process time for O_{nk} in job n

Job agents adopt the following strategy to satisfy their utility function defined in (4):

$$Select^n_{job} = \exists l.(\min TO^l_{nk}) \tag{5}$$

The job agents can't acquire set-up time, because they have no idea which job agent comes next with their local scope. Only the machine agents can hold the set-up information.

3. SCHEDULING PROTOCOL

3.1 Scheduling model

In most of conventional server-client scheduling models, process machine and job are normally defined as server and client, respectively. However, they can behave bilaterally in the metaphor of general market. In this paper we assume two types of scheduling model in terms of agent role in CNP shown in table 1.

Table 1 Scheduling model

	Manager	Contractor
Case1: Contract_Mac	Machine	Job
Case2: Contract_Job	Job	Machine

Needless to say, Case 1 and Case 2 correspond to PULL logic and PUSH logic in factory management, respectively. Therefore this classification is quite natural in manufacturing scheduling.

3.2 Scheduling protocol

We propose a new distributed scheduling concept based on the CNP. The task allocation is realised by a negotiation process between agents called manager. A principle feature is mutual selection mechanism between manager and contractor. In this section, we describe the proposed scheduling protocol in the case 1 (Manager: machine), as an example.

Step 1: Task announcement

After completed a process, machine l constructs task announcements t_l for possible processing service and distributes it by broadcasting to all jobs with requesting information. Bidding time, when the task validity expires, is also included in the information.

Step 2:

After job n receives task announcements, evaluates its own eligibility. If the task satisfies the eligibility, go to *Step 3*. If not, ignore the task. If it receives multiple tasks at the same time, select the most favourite task measured by equation (5).

Step 3: Bid

Job n send a bid with the requested data to machine l.

Step 4: Task allocation

When the bidding time expires, Machine l selects the most appropriate returned bid measured by (3) and allocates the task to that bidder by award notification, and go to *Step* 5. Fail messages are sent to all the other bidders, then go to *Step* 6.

Step 5: Process execution and report

Job n performs the task allocated to it and reports results produced from the performing task to the machine l.

Step 6:
Job waits other appropriate task announcements sent by machines.

4. EXPERIMENTAL RESULTS

4.1 Simulation model

A virtual primitive factory, which is installed the proposed scheduling protocol, has been constructed as a simulation model in order to analyse the scheduling dynamism of the protocol. In this paper we assume the factory has no internal disturbances, such as machine faults or higher priority lot, as a basic study.
Experimental parameters are defined as follows:

- *L*: the number of machines
- *Kind*: the number of Job types
- *Kn*: the number of operations in job *n*
- *SimTime*: simulation period
- *ProcTime*(*Mc*, *Vc*): process time distribution
- *SetUpTime*(*Mp*, *Vp*): set up time distribution
- *ArrivalTime*(*Ma*, *Va*): arrival time distribution
- *Lot*(*Mr*, *Vr*): lot size distribution
- *Bidding period*: BiddingPeriod

  where

  (M*: average, V*: Standard Deviation) in regular distribution

We prepared the following 3 types of conventional heuristics rule-based scheduling algorithms for the comparison in this experiment:

FIFO: first in fist out
SPT-A: shortest processing time
SPT-B: shortest (processing + set up) time

Two kinds of typical manufacturing conditions, "high-volume & low-variety" and "low-volume & high-variety" are examined as the simulation scenario.

4.2 Large lot size manufacturing

The proposed scheduling protocol is evaluated and compared with the conventional heuristic rule-based scheduling in *Kind* = 3, as an example of "high-volume & low-variety" manufacturing. Simulation results are shown in Figure 1, 2, 3.

As described in 2.2, lead time and throughput are two major criteria. At first, Figure 1 shows the relationship between set up time and lead time, the first criterion. It is obvious that the proposed methods show better performances than conventional approaches. Additionally, if we focus only on the proposed approaches, Case-2 is better than Case-1. Figure 2 indicates the same tendency in terms of throughput (= yields). We analysed the relationship between lead time and throughput in Figure 3. Two new parameters are introduced in Figure 3 for a simple analysis as follows:

$$\textit{Lead Time Rate}(i) = \textit{Lead Time}_i / \max(\textit{Lead Time}_{j(j \in P)}) \quad (6)$$

$$\textit{Throughput Rate}(i) = \min(\textit{Yield}_{j(j \in P)}) / \textit{Yield}_i \quad (7)$$

where

P: the set of all the examinations
i, j, p: an examination $i, j, p \in P$

Figure 3 shows the proposed methods have higher robustness compared with conventional scheduling algorithms in term of set up time influence. Especially Case-2, job plays manager, performs the best of all the methods.

In our agent definitions, machine agents try to increase their throughputs and job agents aim at shortening their lead time. Finally a scheduling solution is acquired as the result of their negotiations, and that means the scheduling dynamism is characterised by the mutual selection of the heterogeneous agents with different criteria.

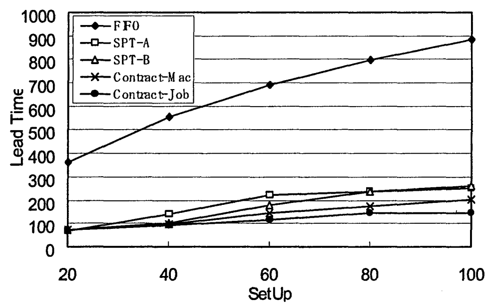

Figure 1 Large lot size manufacturing (Lead Time) L=3, $Kind$=3, Kn=1, $SimTime$=3600, $ProcTime$(50,10), $SetUpTime(Mp,Vp)$: Mp={20,40,60,80,100}, Vp={4,8,12,16,20}, $ArrrivalTime$=50, Lot=3, $BiddingPeriod$=0

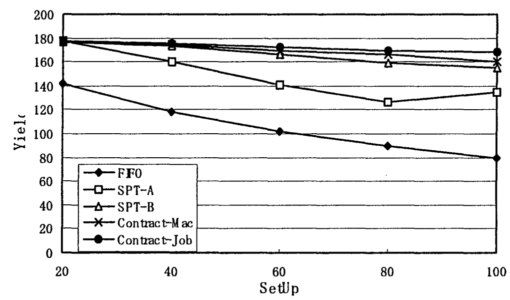

Figure 2 Large lot size manufacturing (Yield) L=3, $Kind$=3, Kn=1, $SimTime$=3600, $ProcTime$(50,10), $SetUpTime(Mp,Vp)$: Mp={20,40,60,80,100}, Vp={4,8,12,16,20}, $ArrrivalTime$=50, Lot=3, $BiddingPeriod$=0

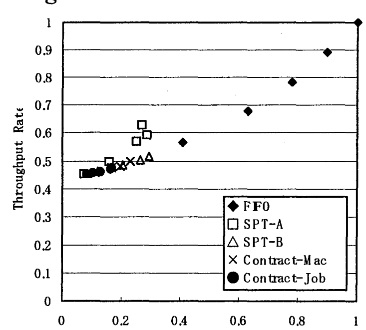

Figure 3 Large lot size manufacturing (Lead time-Throughput) L=3, $Kind$=3, Kn=1, $SimTime$=3600, $ProcTime$(50,10), $SetUpTime(Mp,Vp)$: Mp={20,40,60,80,100}, Vp={4,8,12,16,20}, $ArrrivalTime$=50, Lot=3, $BiddingPeriod$=0

The results shown in Figure 1, 2, 3 indicate that the mutual selection mechanism amongst the heterogeneous agents plays an important role in the scheduling robustness against set up time. Conventional heuristic rule-based approach can't handle with multi-criteria scheduling demands. It is obvious that our approach is effective in terms of the flexibility against the multi-criteria.

By the comparison between Case-1 and Case-2, it is obvious that the careful construction of the decision process is also important even in the proposed approach as well as the conventional ones.

4.3 Small lot size manufacturing

The performance of the proposed scheduling protocol is compared with the conventional approach in *Kind* = 30, as an example of "low-volume & high-variety" manufacturing. Simulation results are shown in Figure 4.

It is clear that the general tendency of the results is almost equivalent to the large lot size manufacturing described in 4.2. It has been confirmed that our approach performs well with robustness in general case. One obvious difference is that Case-2 performs much better than Case-1, compared with the large lot size manufacturing. That points out an important characteristic of the proposed approach.

Our approach is based on the mutual selection amongst the heterogeneous agents. However, first selection is carried out by Job agents and Machine agents as contractors in Case-1 and Case-2, respectively. Machine agent behaviour, shown in equation (3), is to minimise (process + set up) time for the maximum throughput, that is required especially in small lot size manufacturing. These consideration lead the fact, that is the contractor's willingness influences the final scheduling solution more than manager's decision. As the result, the negotiation process is conducted by the contractors more strongly than the managers in the proposed approach.

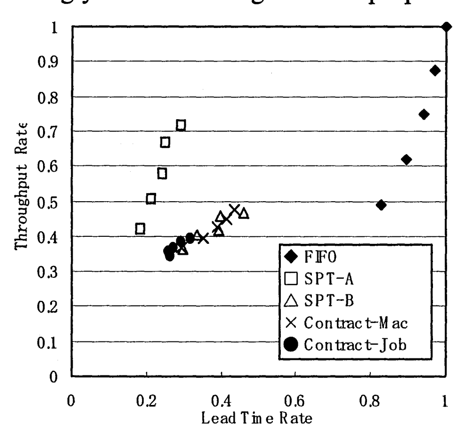

Figure 4 Small lot size manufacturing (Lead time-Throughput) *L*=3, *Kind*=30, *Kn*=1, *SimTime*=3600, *ProcTime*(50,10), *SetUpTime*(*Mp*, *Vp*): *Mp*={20,40,60,80,100}, *Vp*={4,8,12,16,20}, *ArrrivalTime*=50, *Lot*=3, *BiddingPeriod*=0

5. CONCLUSIONS

We have proposed a new scheduling concept that takes into consideration the special requirements of decentralised manufacturing environment. The highlights of the system are that it maintains the higher robustness under multi-utilities in trade-off relationship, such as lead time and throughput.

In this paper, we introduced multi-agent based negotiation protocol, CNP, into scheduling algorithm. After a brief explanation of CNP concept, the criteria on basis of utility in heterogeneous agents, named manager and contractor, were formalised for the decentralised manufacturing scheduling. We demonstrated the applicability of the CNP based scheduling concept by simulation experiments and clarified several important dynamism of the proposed scheduling protocol. Finally it has been proved the proposed concept can provide several advantages on decentralised & distributed manufacturing scheduling.

There are two obvious extensions. The first is to elaborate the negotiation protocol, possibly by exploiting some complexity with bidding period. The second extension is to analyse the robustness against dynamic disturbances in manufacturing system, such as machine failure.

6 REFERENCES

1. Deneubourg, J. (1991), The dynamics of collective sorting robot-like ants and ant-like robots, Proceedings of the First International Conference on Simulation of Adaptive Behavior, The MIT Press.
2. Ishida, T. (1995), Discussion on Agents, Journal of Japanese Society for Artificial Intelligence, Vol.10 No.5, pp.663-667.
3. Ishida, T., Y. Katagiri and K. Kuwabara (1996), Distributed Artificial Intelligence, Corona Publishing Co. Ltd.
4. Kaihara, T. (1996), A Study on Multi Agent Scheduling - An Analysis if the self-organization Phenomenon-, Journal of the University of Marketing and Distribution Sciences -Information, Economics & Management Science-, Vol.5, No.2, pp.113-120.
5. Kaihara, T. and S. Fujii (1997), A self-organization scheduling paradigm using coordinated autonomous agents, Rapid Product Development (N. Iwata, T. Kishinami and F. Kimura Eds.), Chapman & Hall, London, pp.489-498.
6. Kaihara, T. and S. Fujii (1998), An evolutionary scheduling paradigm using coordinated autonomous agents, Innovation, Globalization of Manufacturing in the Digital Communication Era of the 21st Century (G. Jacucci, G. J. Olling, K. Preiss and M. Wozny Eds.), Kluwer Academic Publishers, Boston, pp.553-563.
7. Rabelo, J. R., L. M. Camarinha-Matos and H. Afsarmanesh(1998), Mutigent perspectives to agile scheduling. BASYS 1998, pp. 51-66.
8. Sugimura, N. (1994), Autonomous distributed scheduling, Journal of the Society of Instrument and Control Engineers, Vol.33, No.7, pp.585-589.
9. Smith, R.G. (1980), The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver, IEEE Trans. Comput., Vol.C-29, No.12, pp.1104 - 1113.
10. Walsh, W. E., M. P. Wellman, P. R. Wurman, J. K. Mackie-Mason (1998), Some economics of market-based distributed scheduling, Proceedings of the eighteenth international conference on distributed computing systems.