A Study on Real-Time Scheduling for Holonic Manufacturing Systems – Determination of Utility Values Based on Multi-agent Reinforcement Learning

Koji Iwamura, Norihisa Mayumi, Yoshitaka Tanimizu, and Nobuhiro Sugimura

Osaka Prfecture University, Gradutae School of Engineering, 1-1 Gakuen-cho, Nakaku, Sakai, Osaka, 599-8531, Japan
{iwamura,tanimizu,sugimura}@me.osakafu-u.ac.jp

Abstract. This paper deals with a real-time scheduling method for holonic manufacturing systems (HMS). In the previous paper, a real-time scheduling method based on utility values has been proposed and applied to the HMS. In the proposed method, all the job holons and the resource holons firstly evaluate the utility values for the cases where the holon selects the individual candidate holons for the next machining operations. The coordination holon secondly determine a suitable combination of the resource holons and the job holons which carry out the next machining operations, based on the utility values. Multi-agent reinforcement learning is newly proposed and implemented to the job holons and the resource holons, in order to improve their capabilities for evaluating the utility values of the candidate holons. The individual job holons and resource holons evaluate the suitable utility values according to the status of the HMS, by applying the proposed learning method.

Keywords: Holonic Manufacturing Systems, Real-time Scheduling, Multi-agent Reinforcement Learning, Coordination.

1 Introduction

Recently, automation of manufacturing systems in batch productions has been much developed aimed at realizing flexible small volume batch productions. The control structures of the manufacturing systems developed, such as FMS (Flexible Manufacturing System) and FMC (Flexible Manufacturing Cell), are generally hierarchical. The hierarchical control structure is suitable for economical and efficient batch productions in steady state, but not adaptable to very small batch productions with dynamic changes in the volumes and the varieties of the products.

Computer systems and manufacturing cell controllers have recently made much progress, and individual computers and controllers are now able to share the decision making capabilities in the manufacturing systems. The network architectures are widely utilized for the information exchange in the design and the manufacturing.

V. Mařík, T. Strasser, and A. Zoitl (Eds.): HoloMAS 2009, LNAI 5696, pp. 135–144, 2009.

New distributed architectures of manufacturing systems are therefore proposed to realize more flexible control structures of the manufacturing systems, which are adaptable to the dynamic changes in the volume and the variety of the products and also the unforeseen disruptions, such as malfunction of manufacturing equipment and interruption by high priority jobs. They are so called as ADMS (Autonomous Distributed Manufacturing Systems) [1], BMS (Biological Manufacturing Systems) [2], and HMS (Holonic Manufacturing Systems) [3] [4] [5] [6].

Distributed scheduling methods were proposed and applied to the real-time production scheduling problems of the HMS, in the previous research [4]. The proposed method was adaptable to the dynamic changes and the unforeseen disruptions, and it was suitable for the improvement of the objective functions of the whole HMS such as total make span. However, there were still remaining scheduling problems from the viewpoint for the improvement of the objective functions of the individual components of the HMS.

Therefore, a real-time scheduling method based on the utility values have been proposed and applied to the HMS, in order to improve the objective function values of the individual components of the HMS [5]. The holons in the HMS are divided into three classes based on their roles in the manufacturing processes and the scheduling processes.

(a) Resource holons: They transform the job holons in the manufacturing process. In the scheduling process, they evaluate the utility values for the candidate job holons which are processed by the resource holons in the next time period.
(b) Job holons: They are transformed by the resource holons from the blank materials to the final products in the manufacturing process. In the scheduling process, they evaluate the utility values for the candidate resource holons which carry out the machining operations in the next time period.
(c) Coordination holon: It selects a most suitable combination of the resource holons and the job holons for the machining operations in the next time period, based on the utility values sent from the resource holons and the job holons.

Multi-agent reinforcement learning is newly proposed and implemented to the job holons and resource holons, in order to improve their coordination processes. A reinforcement learning method was proposed and applied to centralized scheduling problems for semiconductor manufacturing processes [7]. In the present research, a reinforcement learning is applied to the agent-based distributed scheduling processes for the manufacturing processes of machine products.

2 Real-Time Scheduling Method for HMS [5]

2.1 Information for Real-Time Scheduling

It is assumed here that the individual job holons have the following technological information.

M_{ik} : k-th machining operation of the job holon i. (i = 1,..,α), (k = 1,..,β).

AC_{ik} : Required machining accuracy of machining operation M_{ik}. It is assumed that the machining accuracy is represented by the levels of accuracy indicated by 1, 2, and 3, which mean rough, medium high, and high accuracy, individually.

R_{ikm} : m-th candidate of resource holon, which can carry out the machining operation M_{ik}. ($m = 1,...,\gamma$).

W_i : Waiting time until the job holon i becomes idle if it is under machining status.

The individual resource holons have the following technological information.

T_{ikm} : Machining time in the case where the resource holon R_{ikm} carries out the machining operation M_{ik}.

MAC_{ikm} : Machining accuracy in the case where the resource holon R_{ikm} carries out the machining operation M_{ik}. MAC_{ikm} is also represented by the levels of 1, 2 and 3.

MCO_{ikm} : Machining cost in the case where the resource holon R_{ikm} carries out the machining operation M_{ik}.

W_{ikm} : Waiting time until resource holon R_{ikm} becomes idle if it is under machining status.

2.2 Real-Time Scheduling Process Based on Utility Values

A real-time scheduling process based on the utility values have been proposed, in the previous research [5], to select a suitable combination of the job holons and the resource holons which carries out the machining operation in the next time period.

At the time t, all the 'idling' holons have to select their machining schedules in the next time period. The following procedure is proposed for the individual holons to select their machining schedules.

(1) Retrieval of status data

The individual 'idling' holons firstly get the status data from the other holons which are 'operating' or 'idling'. The 'idling' holons can start the machining operation in the next time period.

(2) Selection of candidate holons

The individual 'idling' holons select all the candidate holons for the machining operations in the next time period. For instances, the job holon i selects the resource holons which can carry out the next machining operation M_{ik}. On the other hand, the resource holon j select all the candidate job holons which can be machined by the resource holon j.

(3) Determination of utility values

The individual 'idling' holons determine the utility values for the individual candidates selected in the second step. For instances, the job holon determines the utility values, based on its own decision criteria for all the candidate resource holons which can carry out the next machining operation. The utility values are given as follows.

$JUV_i(j)$ ($0 \leq JUV_i(j) \leq 1$): Utility value of the candidate resource j for the job holon i.
$RUV_j(i)$ ($0 \leq RUV_j(i) \leq 1$): Utility value of the candidate job i for the resource holon j.

(4) Coordination

All the 'idling' holons send the selected candidates and the utility values of the candidates to the coordination holon. The coordination holon determine a suitable combination of the job holons and the resource holons which carry out the machining operations in the next time period, based on the utility values. The decision criteria of the coordination holon is to maximize the total sum of the utility values of all the holons.

Table 1. Objective functions of holons

Objective functions	Objective function values
Efficiency of resource holon	Σ Machining time / Total time
Machining accuracy of resource holon	Σ(Machining accuracy of resources – Required machining accuracy of jobs)
Flow-time of job holon	Σ(Machining time + Waiting time)
Machining cost of job holon	Σ(Machining cost of resources)

2.3 Evaluation of Utility Values

The utility values are evaluated based on the decision criteria of the individual holons, and various decision criteria are considered for the holons. Therefore, it is assumed that the individual holons have one of the objective functions shown in Table 1 for evaluating the utility values.

The following procedures are provided for the resource holons to evaluate the utility values. Let us consider a resource holon j at a time t. It is assumed that $TT_{j\cdot t}$, $ME_{j\cdot t}$, and $MA_{j\cdot t}$ show the total time after the resource holon j starts its operations, the efficiency, and the evaluated value of machining accuracy of the resource holon j, respectively. If the resource holon j selects a candidate job holon i for carrying out the machining operation M_{ik}, the efficiency and the evaluated value of the machining accuracy are estimated by the following equations.

$$ME_{j\cdot t+1}(i) = (ME_{j\cdot t} \cdot TT_{j\cdot t} + T_{ikj}) / (TT_{j\cdot t} + T_{ikj} + W_i) . \quad (1)$$

$$MA_{j\cdot t+1}(i) = MA_{j\cdot t} + (MAC_{ikj} - AC_{ik}) . \quad (2)$$

where, the resource holon j can carry out the machining operation M_{ik} of job holon i ($j = R_{ikm}$).

As regards the job holons, the following equations are applied to evaluate the flow-time and the machining costs, for the case where a job holon i selects a candidate resource holon j (= R_{ikm}) for carrying out the machining operation M_{ik}. It is assumed that $JT_{i\cdot t}$ and $JC_{i\cdot t}$ give the total time after the job holon i is inputted to the HMS and the machining cost, respectively.

$$JT_{i\cdot t+1}(j) = JT_{i\cdot t} + T_{ikj} + W_{ikj} . \quad (3)$$

$$JC_{i\cdot t+1}(j) = JC_{i\cdot t} + MCO_{ikj} . \quad (4)$$

The objective functions mentioned above have different units. Some of them shall be maximized and others shall be minimized. Therefore, the utility values are normalized from 0 to 1.

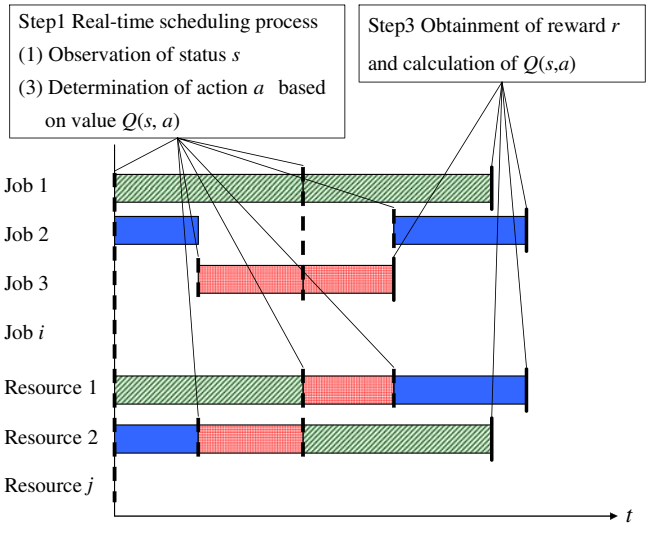

Fig. 1. Application of multi-agent reinforcement learning

3 Application of Multi-agent Reinforcement Learning

A multi-agent reinforcement learning is newly proposed and implemented to the job holons and resource holons, in the present research, in order to improve their coordination processes. In the reinforcement learning method [8], an agent must be able to sense the status of the environment to some extent and must be able to take actions that affect the status. The agent also must have a goal or goals relating to the status of the environment.

Figure 1 summarizes the multi-agent reinforcement learning procedure proposed here. The individual job holons and resource holons carry out the following four steps to obtain their suitable decision criteria for evaluation of the utility values by applying the multi-agent reinforcement learning.

Step1. The individual job holos and resource holons carry out the real-time scheduling process described in section 2.2, when their previous machining operations are finished. The real-time scheduling process (1) and (3) are modified as following for implementation of the multi-agent reinforcement learning.

(1) Retrieval of status data

The individual 'idling' holons get the status data from the other holons which are 'operating' or 'idling', and observe the status s of the manufacturing systems.

(3) Determination of utility values

The individual job holons and resource holons execute the action a based on the value $Q(s, a)$, to evaluate the utility values for all the candidate machining operations in the next time period.

Where, s and a represent the status and the actions in the reinforcement learning method, respectively.

Step2. The real-time scheduling process are repeated until all the machining operations of the job holons are finished by the resource holons in the HMS.
Step3. The individual job holons and resource holons obtain the reward r based on their own objective function values, and calculate the value $Q(s, a)$.
Step4. Step1 to Step3 are repeated for the new job holons to be manufactured in the manufacturing systems, in order to converge the value $Q(s, a)$ of the individual job holons and resource holons.

In these steps, the status s, the action a and the reward r are given as follows.
(1) Status s
The status s observed by the job holons and the resource holons is represented by the following equation, in the present research.

$$s = (s_1, s_2, s_3, s_4) . \tag{5}$$

where, s_p (p =1, 2, 3, 4) are the number of 'idling' holons, that have the objective functions of efficiency, machining accuracy, flow-time, and machining cost, respectively. This means that the learning process of the individual holons are carried out based on the numbers and the types of the 'idling' holons.
(2) Action a
The individual job holons and resource holons select the parameter n (= 1/5, 1/3, 1, 3, or 5) in the following equation to evaluate the utility values.

$$UV' = (UV)^n . \tag{6}$$

where, UV is the utility value calculated by the individual job holons and resource holons described in section 2.3. UV' is the modified utility value by applying the action a based on the status s.
ε-greedy method [8] is applied for the individual job holons and resource holons to determine the action a.
(3) Reward r
The individual job holons and resource holons obtain the reward r based on their own objection values. Three different methods are considered to calculate the reward r.
Type 1. Reward calculated by the objective function values of individual holons
The individual job holons and resource holons obtain the reward r_h given by following equations, based on their own objective functions.
(a) For the case that the objective function is efficiency

$$r_h = (a_h - b_h) / b_h . \tag{7}$$

(b) For the case that the objective function is either machining accuracy, flow time or machining cost

$$r_h = (b_h - a_h) / b_h . \tag{8}$$

where, a_h and b_h are the objective function values obtained by applying the proposed method with the reinforcement learning, and ones obtained without the reinforcement learning.
Type 2. Reward calculated by the objective function values of holons which have same objective function

The individual job holons and resource holons obtain the reward r_p given by following equations.

$$r_p = \sum_{h=1}^{\tau} r_h / \tau . \tag{9}$$

where, p and τ are the types of objective functions and the total number of holons with p-th type of objective functions, respectively. r_h is calculated by Eq. (7) and (8) based on the types of objective functions.

Type 3. Reward calculated by the objective function values of all holons

The individual job holons and resource holons obtain the reward r_q given by following equations.

$$r_q = (1/4) \sum_{p=1}^{4} r_p . \tag{10}$$

where, r_p is calculated by Eq. (9).

The value $Q(s, a)$ is determined by applying the monte carlo method [8]. The individual job holons and resource holons save the n rules (s_t, a_t) $(t = 0, 1, \ldots, n\text{-}1)$ between the time when they obtain the reward r and the time when they obtain the new reward r. The rule (s, a) means the set of status s and action a. The value $Q(s, a)$ is calculated by the following equations.

$$SumReward(s_t, a_t) \leftarrow SumReward(s_t, a_t) + r . \tag{11}$$

$$Q(s, a) \leftarrow SumReward(s, a) / RewardCount . \tag{12}$$

where, $SumReward(s, a)$ is the cumulative rewards in the case where the action a is applied in the status s. $RewardCount$ is the total number in the case where the rule (s, a) get the reward r.

4 Case Study

Some case studies have been carried out to verify the effectiveness of the proposed methods. The HMS model considered in the case studies has 10 resource holons. The individual resource holons have the different objective functions and the different machining capacities, such as the machining time T_{ikm}, the machining accuracy MAC_{ikm}, and the machining cost MCO_{ikm}.

As regards the job holons, 3 cases are considered in the case study, which have 16 job holons, 20 job holons and 30 job holons. The individual job holons have the different objective functions and the machining sequences. It is assumed that the same job holons are inputted to the HMS after the resource holons finish all the manufacturing processes. 12 cases are considered, in the case study, by changing the machining capacities of the resource holons.

ε is set to 0.2 for the ε-greedy method.

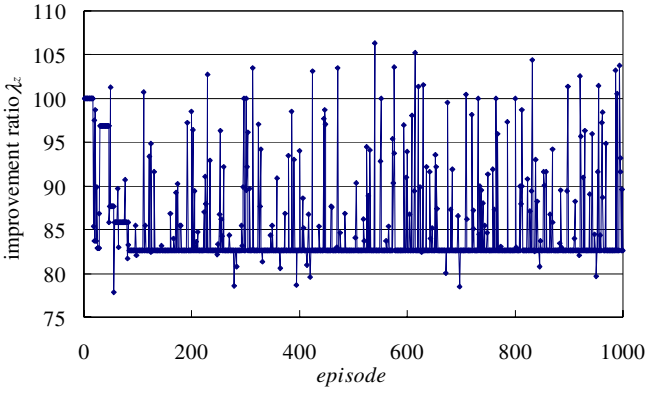

Fig. 2. Improvement ratio

Figure 2 shows the best result for the case where the reward is calculated by using Type3 described in section 3. In the figure, the horizontal and vertical axes show the *episode* and the improvement ratio λ_z, respectively. The *episode* means here the number of repetitions of all the manufacturing processes of the inputted jobs. The improvement ratio λ_z means the ratio between the objective function values of all the holons obtained by the proposed method and the ones by the conventional method in the case z. λ_z is calculated by following equation.

$$\lambda_z = \sum_{h=1}^{\nu} \mu_h / \upsilon . \tag{13}$$

where, μ_h and υ are the improvement ratio of the objective function values of the holon h and the total number of holons, respectively. The μ_h is calculated by the following equation based on the type of the objective functions.

(a) For the case that the objective function is efficiency

$$\mu_h = b_h / a_h . \tag{14}$$

(b) For the case that the objective function is either machining accuracy, flow time or machining cost

$$\mu_h = a_h / b_h . \tag{15}$$

where, a_h and b_h are the objective function values of the individual holons h obtained by the proposed method and the previous conventional method. As shown in the figure, the improvement ratio λ_z is converged until the episode reaches to 100.

Figure 3 shows the average improvement ratio $\lambda average_z$ of the best case and the worst case. Following equation gives the $\lambda average_z$ which means the average of improvement ratio λ_z until the *episode* reaches to ω in the case z.

$$\lambda average_z = \sum_{episode=1}^{\omega} \lambda_{episode} / \omega . \tag{16}$$

where, $\lambda_{episode}$ is the improvement ratio λ_z at the *episode*.

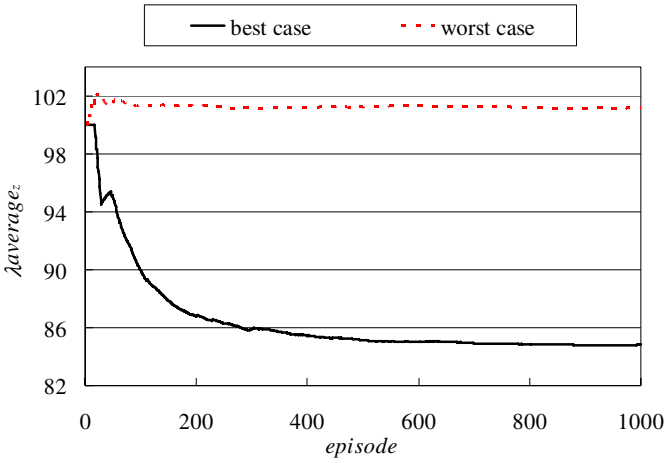

Fig. 3. Average improvement ratio

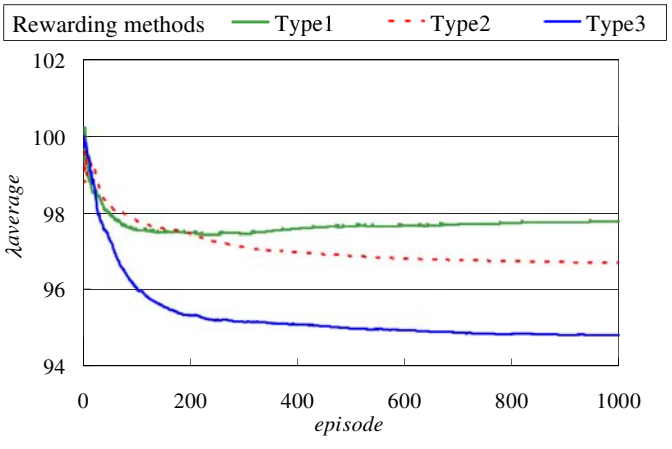

Fig. 4. Comparison of rewarding methods

Figure 4 shows the comparison of the cases using Type1, Type2 and Type3 rewarding methods described in section 3, from the view point of $\lambda averge$. $\lambda averge$ means the average of $\lambda average_z$ in the all 12 cases. $\lambda averge$ is calculated by the following equation.

$$\lambda average = \sum_{z=1}^{12} \lambda average_z / 12. \tag{17}$$

As shown in Fig. 4, all cases are effective to improve the objective function values in comparison with previous method without reinforcement learning. It means that the individual job holons and resource holons obtain the suitable decision criteria for evaluation of utility values. However, as shown in the figure, the value $Q(s, a)$ does

not converge in the case using Type1 where the reward is calculated by the objective function values of individual holons. The individual job holons and resource holons most improve their objective function values in the case using Type3 where the reward is calculated by the objective function values of all holons.

5 Conclusions

New systematic methods are proposed here to improve the coordination process among the job holons and the resource holons based on the multi-agent reinforcement learning. The following remarks are concluded.

(1) The real-time scheduling process are modified for implementation of multi-agent reinforcement learning in order to obtain the suitable decision criteria for evaluation of utility values.
(2) The status, the action and the reward are defined for the individual job holons and the resource holons to evaluate the suitable utility values based on the status of the HMS.
(3) Some case studies of the real-time scheduling have been carried out to verify the effectiveness of the proposed methods in comparison with the previous method. It was shown, through case studies, that the proposed methods are effective to improve the objective function values of the individual holons. The objective function values of individual holons are improved most effectively in the case where the reward is calculated by the objective function values of all holons.

References

1. Moriwaki, T., Sugimura, N.: Object-oriented modeling of autonomous distributed manufacturing system and its application to real-time scheduling. In: Proc. of the ICOOMS 1992, pp. 207–212 (1992)
2. Ueda, K.: An approach to bionic manufacturing systems based on DNA-type information. In: Proc. of the ICOOMS 1992, pp. 303–308 (1992)
3. Hendrik, B., Jo, W., Paul, V., Luc, B., Patrick, P.: Reference architecture for holonic manufacturing systems: PROSA. Computers in Industry 37, 255–274 (1998)
4. Sugimura, N., Tanimizu, Y., Iwamura, K.: A Study on real-time scheduling for holonic manufacturing system. CIRP journal of manufacturing systems 33(5), 467–475 (2004)
5. Iwamura, K., Okubo, N., Tanimizu, Y., Sugimura, N.: Real-time scheduling for holonic manufacturing systems based on estimation of future status. International journal of production research 44(18-19), 3657–3675 (2006)
6. Iwamura, K., Nakano, A., Tanimizu, Y., Sugimura, N.: A study on real-time scheduling for holonic manufacturing systems – simulation for estimation of future status by individual holons. In: Mařík, V., Vyatkin, V., Colombo, A.W. (eds.) HoloMAS 2007. LNCS (LNAI), vol. 4659, pp. 205–214. Springer, Heidelberg (2007)
7. Fujii, N., Takasu, R., Kobayashi, M., Ueda, K.: Reinforcement learning based product dispatching scheduling in a semiconductor manufacturing system. In: Proc. of the 38th CIRP International seminar on manufacturing systems, CD-ROM (2005)
8. Sutton, R., Barto, A.: Reinforcement learning: An introduction. MIT Press, Cambridge (1998)