Work Scheduling by Use of Worker Model in Consideration of Learning by On-The-Job Training*

Toshitake TATENO** and Keiko SHIMIZU**
**Tokyo Metropolitan University,
6-6 Asahigaoka, Hino, Tokyo, Japan
E-mail: tateno-toshitake@c.metro-u.ac.jp

Abstract
This paper deals with a method of scheduling manual work in consideration of learning by on-the-job training (OJT). In skilled work such as maintenance of trains and airplanes, workers must learn many tasks by OJT. While the work processing time of novice workers is longer than that of experts, the time will be reduced with repeated OJT. Therefore, OJT is important for maintaining the skill level and the long-term work efficiency of an organization. In order to devise a schedule considering OJT, the scheduler must incorporate a management function of workers to trace dynamically changing work experience. In this paper, after the relationship between scheduling problems and worker management problems is defined, a simulation method, in which a worker model and an agent-based mechanism are utilized, is proposed to derive the optimal OJT strategy toward high long-term performance. Finally, we present some case studies showing the effectiveness of OJT planning based on the simulation.

Key words: Scheduling, OJT, Learning, Worker Model, Simulation, Agent System

1. Introduction
In work such as the maintenance of trains and airplanes, workers have been required to have technical skills for performing a wide range of work. However, because recent products are too complex to learn the required maintenance skills quickly and the employment duration of workers are not permanent, it is difficult to expect all workers to learn work uniformly. While these situations make management difficult, the difference among workers presents the possibility of promoting workers’ motivation and improving work efficiency. Therefore, the authors have proposed personalized job management using a worker model that represents the properties of worker skills. In this paper, the worker model is used for the problem of devising a schedule that reflects workers’ skill properties.

In general, processing time required by a worker to complete a manual task depends on the worker competency. Since the processing time in on-the-job training (OJT) for primary-level workers in particular is longer than the standard processing time, OJT generates much time delay that the scheduler cannot neglect. It also consumes the instructor time. If we pursue only the work efficiency during a short period, the best choice may be to eliminate OJT. However, OJT is very important for maintaining the skill level and work efficiency of an organization over a long period. There is a trade-off relation between the short-term evaluation and the long-term evaluation. The scheduling system should enable the planning of OJT in consideration of both the immediate due date and the future order plan.
The processing time changes dynamically with experience. Usually, it becomes progressively shorter than in the previous job because the worker learns the task. However, if the worker does not perform the job for a long time, the processing time may become long because the worker will forget the job. Particularly, in maintenance work, even though a worker has experience, he is often considered a novice worker after a long absence because of license issues. In addition, OJT in maintenance work cannot be carried out at any arbitrary time because it is performed in an actual job. If the work target does not exist, the worker cannot learn the job. Therefore, the future performance depends on who executes the job. Traditionally, field managers have allocated many kinds of jobs to group workers, expecting sufficient work performance to fulfill the deadline, and taking the future order plan into consideration. However, in recent situations including many kinds of jobs and various employment types, as written above, managers cannot sufficiently comprehend individual skills of workers. Computer support for the manager is clearly needed.

Figure 1 shows the problem discussed in this paper. The left side is the production scheduling problem. Most studies on the production scheduling problem involve the allocation of jobs to machine tools and workers so as to optimize evaluation factors, such as makespan minimization. Although there are a few studies on scheduling in consideration of worker learning, they concern theoretical approaches using simple models. On the other hand, the right side of Fig. 1 shows the worker management problem, for which the learning problem has long been treated. However, discussions on the expected individual or organization performance are limited. Moreover, differences among the experiences of individuals did not need to be considered since all workers start learning a job at the time in the mass production paradigm. Therefore, the job scheduling problem in consideration of individual work performance and learning is a new problem arising from the recent situations.

2. Problem definition and scope of this paper

The OJT considered in this paper is the primary-level OJT for workers who have little job experience. Since they must learn many jobs in the maintenance work of equipment and large machine products, a very long time is required for learning. Moreover, if workers have no opportunity of a certain job for a long time, they will forget how to perform the job. Therefore, the processing time required for each job is assumed to depend on the worker’s experience. The processing time becomes shorter than that in the previous job if the job is executed frequently. In contrast, the processing time becomes long if the worker has a long interruption time of that specific job.
2.1 Problem formulation

We assume \( n \) kinds of jobs \( J_j \) \((j = 1, 2, \ldots, n)\) in the \( i \)th job set \( S_i \) \((i = 1, 2, 3, \ldots, s)\) processed by \( m \) parallel machines representing workers. All jobs in job set \( S_i \) are different, and every job set has the same contents. The jobs in job set \( S_{i+1} \) are started after all jobs in job set \( S_i \) are completed. The problem is to minimize the makespan for all jobs of all job sets. For a given schedule \( q_i \) for job set \( S_i \) at the \( i \)th cycle, \( C_{i,j} = C_{i,j}(q_i) \) represents the completion time of job \( J_j \) in job set \( S_i \). Then, the makespan \( \sum_{j=1}^{S_i} C_{i,j} \) represents the makespan for all job sets. The important condition of this study is the processing time \( T_j \) for job \( J_j \) depends on the worker's experience and interval time because of the learning and forgetting effects of workers. Using LFE to denote this effect, the problem can be described as \( m/LFE/ \sum_{j=1}^{S_s} C_{i,j} \). We call it the long-term scheduling problem. When we focus on the first job set \( S_1 \), the scheduling problem becomes \( m/LFE/ C_{1,max} \). We call it the short-term scheduling problem. For a similar scheduling problem, there are some theoretical studies\(^{(1)-(4)}\) on the computational time for calculating the optimal sequence. However, in a practical situation, it is not so important to derive long-term optimal sequence strictly because a schedule often changes. In particular, we deal with processing by human workers. In such a case, the simulation-based method is a useful approach. This method does not aim at a perfect optimal solution, but it can derive a strategic scheduling rule by simulation under several different conditions in a practical computational time. We discuss the simulation-based approach to formulating a simple job allocation rule.

Figure 2 shows the scheduling and simulation process. The scheduler allocates given jobs to workers with referring to the worker model for calculating the processing time. By repeating the simulation, the job allocation rule can become sophisticated.

2.2 Previous work

The learning effects in a scheduling problem was first considered by Biskup.\(^{(1)}\) He formulated the single-machine scheduling problem with learning effects, and showed that the schedule is optimized with the shortest processing time (SPT) rule in polynomial time of computation. After that, different settings of the scheduling problem with learning effects
have been considered. Mosheiov\cite{2} considered multimachine settings. Mosheiov and Sidney\cite{3} considered the case of job-dependent learning curves. Wang\cite{4} considered deterioration effects in addition to learning effects. In those papers, the scheduling problem was discussed from theoretical points of view. However, our interest is a job allocation strategy rather than strictly the optimal job sequence. Furthermore, the problem setting in this paper is too complex to solve the optimal sequence in a practical computational time.

Recently, in order to cope with unexpected situations, such as machine breakdown, many agent-based approaches including holonic systems have been studied\cite{5}~\cite{8}. Since the agent-based approach allows easy implementation of complex systems, some papers include the worker schedule in job shop scheduling\cite{5}\cite{6}. However, there are few papers on workers’ learning effects.

The job allocation problem is similar to the workers’ shift scheduling problem\cite{9} from the viewpoint of scheduling considering individual data. However, previous studies do not include learning problems.

In management research, workers’ learning has been dealt with for a long time\cite{10}~\cite{12}. The learning effect model has been used to predict future performance. However, there are few studies on the trade-off problem between time consumed for training and performance after learning. Such discussions were not needed because all workers start learning a job at the same time in the traditional mass production system. Situations such as maintenance work, in which a worker handles many jobs, has not been considered.

### 3. Worker model

The worker model should be defined according to the application. In this scheduling problem, the processing time of workers is the target issue. Since the processing time depends on the worker’s experience, the worker model in this paper represents job experience data that allow the system to estimate the processing time required for that specific job next time.

The learning curve has been discussed for more than 6 decades since Wright’s paper\cite{10}. However, only a few decades have passed since the learning and forgetting model began to be actively discussed against the background of small-lot production for many kinds of products\cite{11}. Wright’s learning curve (WLC) equation is

\[
T_x = T_1 x^{-l}
\]

where \(T_x\) is the time taken to produce the \(x\)th unit, \(T_1\) is the time taken to produce the first unit, \(x\) is the cumulative production unit, and \(l\) is the learning slope (0<\(l\)<1).

While the WLC is a widely recognized model, some forgetting models based on power form equations are proposed as

\[
\hat{T}_x = \hat{T}_1 x^f
\]

where \(\hat{T}_x\) is the time taken to perform the \(x\)th job of lost experience on the forgetting curve, and \(x\) is the corresponding cumulative production unit. Because this equation is used for the nonproductive period, \(x\) actually means the number of potential jobs that would have been performed if interruption had not occurred. \(\hat{T}_1\) is the intercept of the forgetting curve, and \(f\) is the forgetting slope.

Jabor and Sikstrom\cite{12} arranged the forgetting model as three types of models: the learn-forget curve model (LFCM), the recency model (RC), and power integration diffusion (PID). We do not discuss the difference between them, and use the LFCM model\cite{11} to simulate the learning and forgetting of workers.
Figure 3 illustrates the relationship between learning and forgetting in LFCM. The vertical axis $T$ means processing time, and the horizontal axis $x$ means the number of units the worker has produced. $t$ is the learning period that is converted from corresponding units. In the LFCM, it is assumed that $n_i$ units are produced sequentially in the $i$th cycle and that interruption occurs. When a new run starts, there is a gap because the learning accumulated form producing $n_i$ units in the previous lots is not retained as a result of forgetting. We adapt LFCM to maintenance work or single production by regarding the lot size to be equal to 1 ($n_i = 1$), that is, a cycle with one job and one interruption. The $i$th cycle corresponds to the processing of job set $S_i$.

After one job in the $i$th cycle is executed, the learning curve is switched to the forgetting curve. At the interruption point, the processing time of the learning curve is equal to that of the forgetting curve. The forgetting slope $f_i$ is renewed every cycle. Therefore,

$$T_i^{-1} = \hat{T}_i n_i f_i$$

(3)

By using $n_i = 1$ and $u_i$, which is the virtual number of units of memory converted at the beginning of cycle $i$ from experience gained in the previous $i-1$ cycle, the following equation is formed.

$$T_i (u_i + 1)^{-1} = \hat{T}_i (u_i + 1) f_i$$

(4)

We assume $r_i$ units as the break time in which complete forgetting occurs.

$$T_i = \hat{T}_i (u_i + 1 + r_i) f_i$$

(5)

From Eqs. (4) and (5), $\hat{T}_i$ disappears, and the forgetting slope at the $i$th cycle can be expressed as

$$f_i = \frac{I \cdot \log(u_i + 1)}{\log(u_i + 1 + r_i) - \log(u_i + 1)}$$

(6)

Then, using $f_i$, the processing time of the next cycle can be calculated as

$$T_{i+1} = \hat{T}_i (u_i + 1 + s_i) f_i = T_i (u_i + 1)^{-1} (u_i + 1 + s_i) f_i$$

(7)
In order to derive \( s_j \), the number of loss units during break time \( d_j \), the conversion function is defined simply as

\[
s_j = \frac{d_j \cdot u_i}{t(u_i + 1)}
\]  

(8)

where \( t(u_i + 1) \) denotes the total time required to produce \( u_i + 1 \) units, \( d_j \) is the interruption time from the finishing time of \( i \)-th cycle to the starting time at the next \( i+1 \)-th cycle.

The virtual number of units of memory converted at the beginning of the next cycle is calculated, from Eqs. (1) and (7), as

\[
u_{i+1} = \left( \frac{T_i}{T_{i+1}} \right) ^{-1/1} = (u_i + 1)^{(1+f_i/1)}(u_i + 1 + s_j)^{-f_i/1}
\]  

(9)

Therefore, by keeping \( u_i \) and the newest job time for processing \( s_j \), the worker model enables a scheduler to compute the processing time of each worker. These variables are provided for every kind of job for each worker.

4. Scheduling algorithm

The agent-based dynamic scheduling method is applied. The combination searching method is also applicable, but it is inappropriate for this study. Scheduling in this paper means that all jobs are allocated to workers under certain rules when a set of jobs is given. A set of jobs corresponds to one target product job that is ordered for maintenance. Since the order usually has a due date, the estimation of job completion time is important. This is the meaning of the simulation in this study.

In this chapter, we explain the scheduling algorithm of agent-based job allocation and learning control. Then, the simulation using this algorithm is discussed in the next chapter.

4.1 Job allocation

Three kinds of agents are defined in this scheduling system: job agent, manager agent and worker agent.

The job agent represents and manages job information. Some jobs have order restrictions and priority. Order restriction means the relationship between two jobs, where the latter job must start after the former job is finished. Priority means that the prior job should be allocated earlier. We here do not consider the order restriction, but the priority value is used for risk balancing. The selection of the priority value is beyond the scope of this paper. However, we have confirmed that priority does not seriously affect the trend of OJT frequency.

The manager agent communicates with job agents and lists possible jobs in a prioritized order. The manager agent announces the job one by one in the order of ascending priority, and allocates the job to a worker agent with the shortest ECT (Estimated Completion Time) for the job.

The worker agent calculates the ECT of the job when it receives a job announcement from the manager agent. The ECT is calculated by the following formula that is the estimated completion time of the job that the worker has contracted at this moment (ECT\(_C\)) plus processing time (PT) minus education coefficient (EC).

\[
ECT = ECT_C + PT - EC
\]  

(10)
EC is an original coefficient proposed in this study. EC can be used to control the frequency of OJT in scheduling. In this paper, EC is defined as a constant value \( EC_c \) that is applied only if the virtual number of memory \( u_i \) is less than a certain number \( u_c \).

\[
EC = \begin{cases} 
EC_c & (u_i \leq u_c) \\
0 & (u_i > u_c)
\end{cases}
\] \hspace{1cm} (11)

The unit of EC is time. By adding bias against ECT, EC gives advantage to novice workers to have a chance to learn. When EC>0, novice workers more easily receive jobs. On the other hand, when EC<0, experts more easily receive jobs.

EC may be defined as a variable corresponding to \( u_i \). However, we do not deal with such a discussion here. As the first step, a constant value is used to demonstrate that EC can control the amount of OJT and can represent the OJT strategy.

4.2 Agent collaboration

Figure 4 shows the agent collaboration process. The contract net protocol(13) is applied to this procedure. The procedure is explained as the following steps that correspond to the numbers in Fig. 4.

**Step 1:** The manager agent requests job information from every job agent.
**Step 2:** Every job agent responds with the job information.
**Step 3:** The manager agent lists the possible jobs in the order of ascending priority.
**Step 4:** The manager agent announces the job information to each worker agent.
**Step 5:** Each worker agent calculates ECT using Formula (10) and bids the ECT.
**Step 6:** The manager agent contracts the job to a worker agent who bids the earliest ECT, and the manager agent requests other worker agents to cancel the job.
**Step 7:** The manager agent enters the job information into the scheduling chart and requests the job agent to change the work record.

The process from step 1 to step 7 is repeated until all jobs are allocated.

5. Simulation and learning control

5.1 Simulation

Simulation is used to schedule a long period of time with a prospective order plan and to evaluate the long-term performance of the plan.
Figure 5 shows the flow of the simulation process. First, initial conditions such as team organization and workers’ experience are set. Next, event plans for a long period are input, such as order entry and worker change. By trying many different cases of these events, the most appropriate learning strategy will be derived. The nested process is job scheduling. This process follows step 1 to step 7 described in §4.2. Whenever one job is dispatched, the worker model is updated. Finally, the simulation clock is put forward by one time unit. These processes are repeated until the assumed simulation time ends.

5.2 Learning control by EC

As shown in Formula (10), ECT is biased by EC. The EC can be used as a parameter to control the frequency of OJT chances. OJT is not so effective for short-term performance. However, if OJT is continuously repeated, the processing time of a worker will decrease. Therefore, the appropriate learning frequency may be adjusted to improve the long-term performance by using long-term simulation including prospective jobs and team member changes.

EC should be chosen by simulating many cases of prospective orders in consideration of the variance of the amount and time of order entry. The chosen EC represents the learning strategy toward the prospective orders.

6. Case study and discussion

6.1 Controlling OJT frequency by EC

The relationship between EC and total processing time for one job is evaluated using the simulation system. One hundred jobs are assumed as one job set and they are scheduled
to minimize the makespan. The jobs are all different and have different standard processing times. The average of the initial processing times is 60 hours. The worker’s initial experience for each job is set randomly between 0 and 4 times. The learning slope is set as \( l = 0.8 \), referring to the authors’ OJT experimental research data (14). These parameters and other specifications used in this simulation are listed in Table 1.

Table 1 Parameter specifications for the simulation

<table>
<thead>
<tr>
<th>Kind of jobs for one job set</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of jobs for one job set</td>
<td>100</td>
</tr>
<tr>
<td>Number of workers</td>
<td>6</td>
</tr>
<tr>
<td>Average processing time at first time (hour)</td>
<td>60</td>
</tr>
<tr>
<td>Learning slope</td>
<td>0.8</td>
</tr>
<tr>
<td>Average initial experience: ( u_i )</td>
<td>0 ~ 4</td>
</tr>
</tbody>
</table>

Figure 6 shows part of a Gantt chart for one job set in the case of EC=0. The thick lines show that these jobs are allocated as OJT. Although EC=0, some OJT jobs are allocated. Figure 7 shows the relationship between EC and the number of OJT jobs allocated in the scheduling. The number of OJT jobs increases according to EC. We can confirm that EC controls the frequency of OJT allocations.

### 6.2 Long-term performance with different EC

The most important point of this simulation is to analyze the change in the makespan over a long period of time. Then, the makespan for each job set is simulated using 100 job sets. Figure 8 shows the simulation results for different EC values while 100 job sets are processed successively without time lags.
In accordance with the repetition of jobs, the processing time changes every time because workers learn and forget. The trend of the change depends on not only the learning and forgetting slope but also many work conditions, such as the number of kinds of jobs, the number of workers, the time of order entry, and the value of EC.

In all cases of EC, makespan is reduced and balanced at a constant value. An interesting point is that the makespan for EC=0 to 90 is proportional to EC initially, but it finally balances at a constant value. On the other hand, the makespan for EC=-30 to -90 remains at each value which indicates imperfect learning. We can see that there is an optimal value of EC that gives the minimum makespan for 100 job sets.

Figure 9 shows the average number of OJT jobs and the average makespan for 100 jobs over different EC values. Although the average number of OJT jobs increases monotonically with EC, the average makespan has a minimum value. In this case, approximately EC=10 gives the best performance.

Figure 10 and Fig. 11 show the simulation results in case that the number of workers was increased to 8 persons. As shown in Fig. 10, in all cases, approximately the initial makespan is kept through 100 job sets. The reason is that learning cannot be accumulated since the learning chances are spread over 8 persons. In this situation, most OJT jobs becomes useless learning. Figure 11 shows that approximately EC=-30 gives the best performance.

Therefore, the best EC depends on the work conditions. We can see that the simulation is effective for optimizing the long-term performance including prospective orders and for planning the organization of the work field.
6.3 Learning strategy to adapt complex conditions

In the previous discussion, EC was assumed to be a constant value. Furthermore, EC can be set as a variable that adapts to complex conditions, such as an irregular order and unexpected member change among workers. However, that will require many calculation steps to optimize the EC plan. The development of an effective algorithm is a future work.

As another approach, it is thought that workers have individual learning strategies and negotiate the contract of jobs to fulfill their learning plan. This approach may prevent the spread of learning chances, as seen in the latter case described in § 6.2. This approach is consistent with the concept of the worker model. The worker model is naturally defined to express the individual characteristics of workers and to develop the workers' strong points. This is also an important point to study in a future work.

7. Conclusions

(1) A new scheduling problem was introduced, in which the processing time of manual work is assumed to be changed dynamically by OJT. The problem was broken down and formulated.

(2) In order to manage the processing time of each worker in scheduling, the learning and forgetting model was used as a worker model that represents worker's skill property.

(3) Agent-based dynamic scheduling method was applied to the problem. Agent behaviors and a simulation framework were proposed. In the job allocation, the parameter EC was defined in order to control the frequency of OJT jobs.

(4) The scheduling algorithm was implemented. Some case studies revealed that EC could control the frequency of OJT allocation and the simulation enabled us to derive the optimal strategy for long-term performance by adjusting EC according to prospective orders.
References


