



Research article

Single machine and group scheduling with random learning rates

Dingyu Wang^{1,2} and Chunming Ye^{1,*}

¹ Business School, University of Shanghai for Science and Technology, Shanghai 200093, China

² College of Finance and Mathematics, West Anhui University, Lu'an 237012, China

* **Correspondence:** Email: yechm@usst.edu.cn.

Abstract: This study mainly considers the scheduling problems with learning effects, where the learning rate is a random variable and obeys a uniform distribution. In the first part, we introduce a single machine model with location-based learning effects. We have given the theoretical proof of the optimal solution for the five objective functions. In the second part, we study the problem with group technology. Both intra-group and inter-group have location-based learning effects, and the learning rate of intra-group jobs follows a uniform distribution. We also give the optimal ranking method and proof for the two problems proposed.

Keywords: stochastic scheduling; group scheduling; learning effect; uniform distribution

Mathematics Subject Classification: 68M20, 90B36

1. Introduction

Research on single machine scheduling problems with location-based learning effects has long been the focus of scholars. Early pioneering research can be found in Biskup [1] and Moshiov [2]. In their research, they assumed that the processing time of the job was constant and a decreasing function of its position respectively. Moshiov [3] described the parallel machine case. Mosheiov and Sidney [4] introduced job-dependent learning effects and thought that the problem could be transformed into an allocation problem. Since then, many scholars have studied similar or improved models. There are many related achievements, and the research perspective is becoming more and more open. Mosheiov and Sidney [5] studied the case that all tasks have a common due date, and proved that the model is polynomial solvable. Bachman and Janiak [6] provided a method of proof using diagrams. Lee, Wu and Sung [7] discussed the two-criteria problem, and put forward the skills of searching for the optimal solution. The authors followed the methods of the non-increasing order of ω_j and the shortest processing time sequence achieves the optimal solution. Lee and Wu [8] solved a flow shop scheduling problem with two machines using a heuristic algorithm. Janiak and Rudek [9] discussed the complexity

results of a single machine scheduling problem that minimizes the number of delayed jobs. Zhao, Zhang and Tang [10] investigated the polynomial solutions of some single machine problems, parallel machine problems, and flow shop problems in environments with learning effects. Cheng, Sun and Yu [11] considered some permutation flow shop scheduling problems with a learning effect on no-idle dominant machines. Eren and Ertan [12] studied the problem of minimizing the total delay in the case of learning effects, and used the 0-1 integer programming model to solve this problem. Zhang and Yan [13], Zhang et al. [14], Liu and Pan [15] and Liu, Bao and Zheng [16] have all studied scheduling problems based on learning effects from the perspective of problem model innovation or improvement.

With the deepening of research, many scholars found that p is not always a constant and there will be different processing times in different environments, different machines or different workers. Therefore, some scholars put forward the stochastic scheduling problem. Pinedo and Rammouz [17], Frenk [18] and Zhang, Wu and Zhou [19] have done a lot of pioneering work. Based on research results of the above scholars, Zhang, Wu and Zhou [20] studied the single machine stochastic scheduling problem based on the learning effect of location, and studied the optimal scheduling strategy for stochastic scheduling problems with and without machine failures. Ji et al. [21] considered the parallel machine scheduling caused by job degradation and the learning effect of DeJong. They proved that the proposed problem is polynomial solvable, and provided a fully polynomial time approximation solution. A labor scheduling model with learning effect is proposed by Qin, Liu and Kuang [22]. By piecewise linearizing the curve, the mixed 0-1 nonlinear programming model (MNLP) was transformed into the mixed 0-1 linear programming model (MLP) for solution. Zhang, Wang and Bai [23] proposed a group scheduling model with both degradation and learning effects. Xu et al. [24] proposed a multi machine order scheduling problem with learning effect, and use simulated annealing and particle swarm optimization to obtain a near optimal solution. Wu and Wang [25] consider a single machine scheduling problem with learning effects based on processing time and truncation of job delivery times related to past sequences. Vile et al. [26], Souiss, Benmansour and Artiba [27], Liu et al. [28] and Liu et al. [29] applied scheduling models separately to emergency medical services, supply chain, manufacturing management and graph theory. Li [30] studied the processing time of the job as random and uses a job-based learning rate. At the same time, he provided a method to deal with problems by using the difference between EVPI and EVwPI. Toksari and Atalay [31] studied four problems of the coexistence of learning effect and homework refusal. In order to reduce production costs, Chen et al. [32] focused on multi-project scheduling and multi-skilled labor allocation. Shi et al. [33] applied a machine learning model to medical treatment, estimate service level and its probability distribution, and used various optimization models to solve scheduling programs. Wang et al. [34] improved and studied several existing problems.

Ham, Hitomi and Yoshida [35] first proposed the “group technology” (GT). According to type or characteristics, jobs are divided into different groups. Each group of jobs is produced by the same means and, once the jobs are put into production, they cannot be stopped. Lee and Wu [36] proposed a group scheduling learning model where the learning effect not only depends on the work location, but also on the group location. We have demonstrated that the problem is polynomial solvable under the proposed model. Yang and Chand [37] studied a single machine group scheduling problem with learning and forgetting effects to minimize the total completion time of tasks. Zhang and Yan [38] proposes a group scheduling model with deterioration and learning effects. Under the proposed model, the completion time and total completion time problems are polynomial optimal solvable.

Similarly, Ma et al. [39], Sun et al. [40] and Liu et al. [41] provided appropriate improvements to the model. Li and Zhao [42] considered the group scheduling problem on a single machine with multiple expiration window assignments. They also divided the homework into several groups to improve production efficiency and save resources. However, this work is only an improvement on some problems or solutions and has not achieved groundbreaking results. Liang [43] inquired into the model with deteriorating jobs under GT to minimize the weighted sum of manufacturing time and resource allocation costs. Wang et al. [44] considered the issue of maturity allocation and group technology at the same time. They determined the best sequence and the best deadline allocation of the group and intra-group jobs by minimizing the weighted sum of the absolute values of the lateness and deadline allocation costs. Wang and Ye [45] established a stochastic grouping scheduling model. Based on SDST and preventive maintenance, Jain [46] proposed a method based on a genetic algorithm to minimize the performance measurement of completion time.

The above literature provides assistance in establishing problem models and solving methods for classical sorting and random sorting with learning effects. In the learning effect, the learning factor is a very important quantity which often affects the actual processing time of the job. There are many factors that affect learning factors, which can be internal or external and often have randomness. In this case, the learning factor is no longer a constant—it is variable—and sometimes the probability density can be calculated. These aspects were not considered in previous works. Therefore, the discussion of the randomness of learning factors in this study has practical significance. Based on the above ideas, this study establishes a new stochastic scheduling model. In the model, workers participate in production, the processing of jobs has a position based learning effect and the learning index is random. The method to solve the problem is to use heuristic algorithms to find the optimal sequencing of the problems.

2. Scheduling problem with stochastic learning rate: single machine

n independent jobs are processed on one machine. The job can be processed at any time and cannot be stopped during processing.

First model is

$$1 \left| p_j^r = p_j r^{-a_j}, a_j \sim U(0, \lambda_j) \right| E[f(C_j)],$$

where $f(C_j)$ is function of processing time of job J_j .

First, we give some lemmas.

Lemma 2.1. [16] X is a random variable, continuous on the definition field, and $f_X(x)$ is density function, $Y = g(X)$, there is $E(Y) = E[g(X)] = \int_{-\infty}^{+\infty} g(x)f_X(x) dx$.

Lemma 2.2. [16] The meaning of X is the same as Lemma 2.1, function $g(x)$ is everywhere derivable and monotonous, $Y = g(X)$, then,

$$f_Y(y) = \begin{cases} f_X[h(y)] \cdot |h'(y)|, & \alpha < y < \beta, \\ 0, & \text{others,} \end{cases} \quad (2.1)$$

in which α, β is the minimum and maximum value of $g(-\infty), g(+\infty)$ respectively, $h(y) = g^{-1}(x)$.

Lemma 2.3. When $x > 1$, $y(x) = \frac{1}{\ln x} \left(1 - \frac{1}{x}\right)$ is monotonic and does not increase, $z(x) = 1 - \frac{1}{\ln x} \left(1 - \frac{1}{x}\right)$ is a monotone increasing function, and $z(x) > 0$.

It is easy to prove the conclusion by using the method of derivation and function limit.

Lemma 2.4. If the random variable $a \sim U(0, \lambda)$, $X = pr^{-a}$, where p is constant, then $f_X(x) = \frac{1}{\lambda} \cdot \frac{1}{x \ln r}$.

Proof. From (2.1), when $a \sim U(0, \lambda)$, we can get $X \in (pr^{-\lambda}, p)$, and

$$f_X(x) = f_X[h(a)] \cdot |h'(a)| = \frac{1}{\lambda} \cdot \left| \left(\frac{\ln p - \ln x}{\ln r} \right)' \right| = \frac{1}{\lambda} \cdot \frac{1}{x \ln r}. \quad (2.2)$$

□

Second, we give the symbols and their meanings in the theorems (see Table 1).

Table 1. The symbols and their meanings.

Symbol	Description
J	A collection of independent jobs
J_j	A job in J
r	Job position in the sequenc
a	Learning rate, $a > 0$
ω_i	Weight of job i
p_j	The p of J_j , $p_j \sim U(0, \lambda_j)$
S, S'	Job sequence
d_i	The due date of J_i
L_{\max}	Maximum delay
T_j	Delay of job J_j
U_j	Penalty of job j
$p_i^r(S)$	Random processing time when J_i in S is r -position
π_1, π_2	Jobs without J_i, J_j in the sequence
$E(\cdot)$	Mathematical expectation of random variable
t_0	Completion time in the sequence except for J_i, J_j
t_0'	Time required for the $(r - 1)$ -th job in the sequence to finish processing
$C_j(S)$	Time spent on completion of J_j in S
$\sum \omega_j C_j$	Weighted completion time when all jobs are processed

Theorem 2.1. For $1 \left| p_j^r = p_j r^{-a_j}, a_j \sim U(0, \lambda_j) \right| E(C_{\max})$, if p is consistent with the parameters λ , that is, for all J_i, J_j , if there is $p_i \leq p_j \Rightarrow \lambda_i \leq \lambda_j$, if the λ is large, arrange it first, we can obtain the optimal ranking of the problem.

Proof. (1) There is the first job in the exchanged jobs, the job in the first and second positions is exchanged.

According to hypothesis and Lemma 2.1, we have

$$E [C_2 (S)] = E (t_0) + E [p_1^1 (S)] + E [p_2^2 (S)] = E (t_0) + p_1 + \frac{p_2}{\ln (2^{\lambda_2})} \cdot \left(1 - \frac{1}{2^{\lambda_2}}\right), \quad (2.3)$$

$$E [C_1 (S')] = E (t_0) + E [p_2^1 (S')] + E [p_1^2 (S')] = E (t_0) + p_2 + \frac{p_1}{\ln (2^{\lambda_1})} \cdot \left(1 - \frac{1}{2^{\lambda_1}}\right). \quad (2.4)$$

Note that $p_1 \leq p_2 \Rightarrow \lambda_1 \leq \lambda_2, 2^{\lambda_2} > 2^{\lambda_1} > 1$, from Lemma 2.3, we can get

$$\frac{1}{\ln (2^{\lambda_2})} \cdot \left(1 - \frac{1}{2^{\lambda_2}}\right) < \frac{1}{\ln (2^{\lambda_1})} \cdot \left(1 - \frac{1}{2^{\lambda_1}}\right), 1 - \frac{1}{\ln (2^{\lambda_1})} \left(1 - \frac{1}{2^{\lambda_1}}\right) > 0. \quad (2.5)$$

From (2.3)–(2.5), it can be obtained that

$$\begin{aligned} & E [C_2 (S)] - E [C_1 (S')] \\ &= p_1 - p_2 + \frac{p_2}{\ln (2^{\lambda_2})} \cdot \left(1 - \frac{1}{2^{\lambda_2}}\right) - \frac{p_1}{\ln (2^{\lambda_1})} \cdot \left(1 - \frac{1}{2^{\lambda_1}}\right) \\ &< p_1 - p_2 + \frac{p_2}{\ln (2^{\lambda_1})} \cdot \left(1 - \frac{1}{2^{\lambda_1}}\right) - \frac{p_1}{\ln (2^{\lambda_1})} \cdot \left(1 - \frac{1}{2^{\lambda_1}}\right) \\ &= \left[1 - \frac{1}{\ln (2^{\lambda_1})} \cdot \left(1 - \frac{1}{2^{\lambda_1}}\right)\right] \cdot (p_1 - p_2) < 0. \end{aligned}$$

(2) When there is no first job in the exchanged jobs, that is $r \geq 2$. We compare $E [C_j (S)]$ of J_j in S with $E [C_i (S')]$ of J_i in S' .

From hypothesis and Lemma 2.1, we get

$$\begin{aligned} E [C_j (S)] &= E (t_0) + E [p_i^r (S)] + E [p_j^{r+1} (S)] \\ &= E (t_0) + E [p_i r^{-a_i}] + E [p_j (r+1)^{-a_j}] \\ &= E (t_0) + \frac{p_i}{\ln r \cdot \lambda_i} \left(1 - r^{-\lambda_i}\right) + \frac{p_j}{\ln (r+1) \cdot \lambda_j} \left[1 - (r+1)^{-\lambda_j}\right], \end{aligned} \quad (2.6)$$

$$\begin{aligned} E [C_i (S')] &= E (t_0) + E [p_j^r (S')] + E [p_i^{r+1} (S')] \\ &= E (t_0) + E [p_j r^{-a_i}] + E [p_i (r+1)^{-a_j}] \\ &= E (t_0) + \frac{p_j}{\ln r \cdot \lambda_i} \left(1 - r^{-\lambda_i}\right) + \frac{p_i}{\ln (r+1) \cdot \lambda_j} \left[1 - (r+1)^{-\lambda_j}\right]. \end{aligned} \quad (2.7)$$

Notice the hypothesis of the Theorem 2.1, for all J_i, J_j , if there is $p_i \leq p_j \Rightarrow \lambda_i \leq \lambda_j$, and when $r \geq 2, r^{\lambda_i} > 1$. From Lemmas 2.2, 2.3 and (2.6), (2.7), we get

$$\begin{aligned} & E [C_j (S)] - E [C_i (S')] \\ &= \left\{ \frac{1}{\ln (r^{\lambda_i})} \left(1 - \frac{1}{r^{\lambda_i}}\right) - \frac{1}{\ln [(r+1)^{\lambda_j}]} \left(1 - \frac{1}{(r+1)^{\lambda_j}}\right) \right\} \cdot (p_i - p_j) < 0. \end{aligned}$$

Proof complete. □

Theorem 2.2. For model 1 $\left| p_j^r = p_j r^{-a_j}, a_j \sim U(0, \lambda_j) \right| E(\sum \omega_j C_j)$, if p and its weight meet $\frac{p_i}{p_j} \leq \min\left\{1, \frac{\omega_i}{\omega_j}\right\}$, the optimal order is obtained by the larger $\lambda_j \omega_j$ first.

Proof. Suppose $p_i \leq p_j$, according to the previous assumptions, we have

$$E[C_i(S)] = E(t_0') + E[p_i^r] = E(t_0) + \frac{P_i}{\ln r \cdot \lambda_i} (1 - r^{-\lambda_i}), \quad (2.8)$$

$$\begin{aligned} E[C_j(S)] &= E(t_0') + E[p_i^r] + E[p_j^{r+1}] \\ &= E(t_0) + \frac{P_i}{\ln r \cdot \lambda_i} (1 - r^{-\lambda_i}) + \frac{P_j}{\ln(r+1) \cdot \lambda_j} \cdot [1 - (r+1)^{-\lambda_j}], \end{aligned} \quad (2.9)$$

$$E[C_j(S')] = E(t_0') + E[p_j^r] = E(t_0) + \frac{P_j}{\ln r \cdot \lambda_j} (1 - r^{-\lambda_j}), \quad (2.10)$$

$$\begin{aligned} E[C_i(S')] &= E(t_0') + E[p_j^r] + E[p_i^{r+1}] \\ &= E(t_0) + \frac{P_j}{\ln r \cdot \lambda_j} (1 - r^{-\lambda_j}) + \frac{P_i}{\ln(r+1) \cdot \lambda_i} \cdot [1 - (r+1)^{-\lambda_i}]. \end{aligned} \quad (2.11)$$

Notice $\frac{p_i}{p_j} \leq \min\left\{1, \frac{\omega_i}{\omega_j}\right\}$, $\omega_j \lambda_j \geq \omega_i \lambda_i$ and (2.8)–(2.11), we can get

$$\begin{aligned} &\omega_i E[C_i(S)] + \omega_j E[C_j(S)] - \omega_i E[C_i(S')] - \omega_j E[C_j(S')] \\ &= (\omega_i + \omega_j)(p_i - p_j) \cdot \frac{1}{\ln r \cdot \lambda_i} (1 - r^{-\lambda_i}) + (\omega_j p_j - \omega_i p_i) \cdot \frac{1}{\ln(r+1) \cdot \lambda_j} \cdot [1 - (r+1)^{-\lambda_j}] \\ &\leq (\omega_i + \omega_j)(p_i - p_j) \cdot \frac{1}{\ln r \cdot \lambda_i} (1 - r^{-\lambda_i}) + (\omega_j p_j - \omega_i p_i) \cdot \frac{1}{\ln r \cdot \lambda_i} (1 - r^{-\lambda_i}) \\ &= (\omega_j p_i - \omega_i p_j) \cdot \frac{1}{\ln r \cdot \lambda_i} (1 - r^{-\lambda_i}) \leq 0. \end{aligned}$$

Theorem 2.2 is proved. \square

Theorem 2.3. When $\lambda_i \leq \lambda_j \Rightarrow d_i \leq d_j$ for J_i, J_j , the EDD rule, that is non-decreasing of d_j is the optimal algorithm of problem 1 $\left| p_j^r = p_j r^a, p_j \sim U(0, \lambda_j) \right| E(L_{\max})$.

Proof. First, we consider $\lambda_i \leq \lambda_j$. The parameters of J_i, J_j in S' satisfy the relationship $\lambda_i \leq \lambda_j \Rightarrow d_i \leq d_j$, but they violate the EDD rule, that is, J_j is processed before J_i . In fact, J_i is processed before J_j in S . Then we prove that the sequence S' can be transformed into a non-decreasing of d_j sequence S and this transformation makes the maximum delay non-increasing. Further more, we suppose the last job in π_1 is on the $(r-1)$ -th position and the expected completion time of it is $E(t_0)$. From (2.8)–(2.11), we get the expected delays of J_i, J_j in sequence S are

$$E[L_i(S)] = E[C_i(S)] - d_i = E(t_0) + \frac{1}{2} \lambda_j r^a + \frac{1}{2} \lambda_i (r+1)^a - d_i, \quad (2.12)$$

$$E[L_j(S)] = E[C_j(S)] - d_j = E(t_0) + \frac{1}{2}\lambda_j r^a - d_j. \quad (2.13)$$

Similarly, the expected delays of J_i, J_j in sequence S' are

$$E[L_i(S')] = E[C_i(S')] - d_i = E(t_0) + \frac{1}{2}\lambda_i r^a - d_i, \quad (2.14)$$

$$E[L_j(S')] = E[C_j(S')] - d_j = E(t_0) + \frac{1}{2}\lambda_i r^a + \frac{1}{2}\lambda_j(r+1)^a - d_j. \quad (2.15)$$

Because $\lambda_i \leq \lambda_j, d_i \leq d_j$ and (2.12)–(2.15), so

$$E[L_i(S)] - E[L_j(S')] = \frac{1}{2}(\lambda_j - \lambda_i) \cdot [r^a - (r+1)^a] + (d_j - d_i) \geq 0, \quad (2.16)$$

$$E[L_j(S)] - E[L_i(S')] = \frac{1}{2}(\lambda_j - \lambda_i) \cdot r^a + \frac{1}{2}\lambda_i \cdot (r+1)^a > 0. \quad (2.17)$$

Then from (2.16) and (2.17), we have

$$\max\{E[L_i(S)], E[L_j(S)]\} \leq \max\{E[L_i(S')], E[L_j(S')]\}. \quad (2.18)$$

Thus, exchanging the position of J_i and J_j does not cause an increase in the maximum delay. After a limited number of similar processing, the optimal sequence can be transformed into the order of non-decreasing of due time d_j and the expected maximum delay will not increase. \square

Theorem 2.4. For $1 \left| p_j^r = p_j r^a, p_j \sim U[0, \lambda_j] \right| E(\sum T_j)$, when $\lambda_i \leq \lambda_j \Rightarrow d_i \leq d_j$, we have the optimal algorithm by non-decreasing order of d_j .

Proof. First we assume $d_i \leq d_j$. The necessary condition for the theorem proof is the same as that of Theorem 2.3. Now we show that exchanging the position of J_i and J_j does not increase the objective function value. Then we repeat the exchange of adjacent jobs to get the optimal sequence of problem.

We discuss it in two cases:

Case 1. $E[C_j(S')] \leq d_j$, in which $E[T_i(S)] \neq 0, E[T_j(S)] \neq 0$. If one of them is zero, the conclusion is obviously true.

From Theorem 2.3 and $d_i \leq d_j$, we have

$$\begin{aligned} & (E[T_i(S')] + E[T_j(S')]) - (E[T_i(S)] + E[T_j(S)]) \\ &= \max\{E[C_i(S')] - d_i, 0\} + \max\{E[C_j(S')] - d_j, 0\} \\ & \quad - \max\{E[C_i(S)] - d_i, 0\} - \max\{E[C_j(S)] - d_j, 0\} \\ &= E[C_i(S')] - E[C_i(S)] - (E[C_j(S)] - d_j) \geq 0. \end{aligned} \quad (2.19)$$

Case 2. $E[C_j(S')] > d_j$. Notice $d_i \leq d_j, \lambda_i \leq \lambda_j$, we have

$$\begin{aligned} & (E[T_i(S')] + E[T_j(S')]) - (E[T_i(S)] + E[T_j(S)]) \\ &= E[C_i(S')] + E[C_j(S')] - E[C_i(S)] - E[C_j(S)] \geq 0. \end{aligned} \quad (2.20)$$

From (2.19) and (2.20), we get $E[T_i(S)] + E[T_j(S)] \leq E[T_i(S')] + E[T_j(S')]$. \square

In the following discussion, we assume that $d_j = d, j = 1, 2, \dots, n$.

Theorem 2.5. For $1 \leq p_j^r = p_j r^a, p_j \sim U(0, \lambda_j), d_i = d \mid E(\sum \omega_j U_j)$, there are the following results:

(1) When $d - t > \min(\lambda_1, \lambda_2)$, according to the non-decreasing order of λ_j , we can get the smaller number of expected weighted tardy jobs;

(2) When $d - t < \lambda_1, d - t < \lambda_2, \omega_1 \leq \omega_2 \Rightarrow d - t \geq \frac{2(\lambda_2 - \lambda_1)}{(r+1)^a} \cdot \frac{\omega_1 + \omega_2}{\omega_1 - \omega_2} + \frac{\omega_1 \lambda_1 - \omega_2 \lambda_2}{\omega_1 - \omega_2}$, we can get the minimum expected value of the objective function according to the non-decreasing rules of job weight.

Proof. (1) First we assume the machine is idle at time t and then there are two jobs J_1, J_2 without processing. Suppose processing times of J_1, J_2 are p_1 and p_2 , J_1, J_2 are processed at the r -th position and $(r+1)$ -th position respectively. The weights of J_1, J_2 are ω_1 and ω_2 . When $d - t > \min(\lambda_1, \lambda_2)$, we should process the job for which λ_j is smaller first, so that we can guarantee the target value is smaller.

(2) In the following we assume $\omega_1 \leq \omega_2, d - t < \lambda_1$ and $d - t < \lambda_2$.

Let $E[\sum \omega U(1, 2)]$ denotes the expected weighted number of tardy jobs when we process J_1 first and then process J_2 , then

$$\begin{aligned} E\left[\sum \omega U(1, 2)\right] &= (\omega_1 + \omega_2) \cdot P(p_1^i > d - t) + \omega_2 \cdot P(p_2^i < d - t, p_2^i + p_1^{i+1} > d - t) \\ &= (\omega_1 + \omega_2) \cdot \int_{d-t}^{\lambda_1} i^a \cdot \frac{1}{\lambda_1} dx + \omega_2 \cdot \int_0^{d-t} dx \int_{d-t-x}^{\lambda_2} \frac{i^a \cdot (i+1)^a}{\lambda_1 \lambda_2} dy \\ &= (\omega_1 + \omega_2) \cdot i^a \left[1 - \frac{1}{\lambda_1} (d - t)\right] + \omega_2 \cdot \frac{i^a \cdot (i+1)^a}{\lambda_1 \lambda_2} \left[\lambda_2 (d - t) - \frac{1}{2} (d - t)^2\right]. \end{aligned} \quad (2.21)$$

The meaning of $E[\sum \omega U(2, 1)]$ is similar to $E[\sum \omega U(1, 2)]$, then

$$\begin{aligned} E\left[\sum \omega U(2, 1)\right] &= (\omega_1 + \omega_2) \cdot P(p_2^i > d - t) + \omega_1 \cdot P(p_2^i < d - t, p_2^i + p_1^{i+1} > d - t) \\ &= (\omega_1 + \omega_2) \cdot \int_{d-t}^{\lambda_2} i^a \cdot \frac{1}{\lambda_2} dx + \omega_1 \cdot \int_0^{d-t} dx \int_{d-t-x}^{\lambda_1} \frac{i^a \cdot (i+1)^a}{\lambda_1 \lambda_2} dy \\ &= (\omega_1 + \omega_2) \cdot i^a \left[1 - \frac{1}{\lambda_2} (d - t)\right] + \frac{i^a \cdot (i+1)^a}{\lambda_1 \lambda_2} \left[\lambda_1 (d - t) - \frac{1}{2} (d - t)^2\right]. \end{aligned} \quad (2.22)$$

According to (2.21) and (2.22), we obtain

$$\begin{aligned} &E\left[\sum \omega U(1, 2)\right] - E\left[\sum \omega U(2, 1)\right] \\ &= \frac{r^a \cdot (r+1)^a}{2\lambda_1 \lambda_2} (\omega_1 - \omega_2) (d - t) \left[(d - t) + \frac{2(\lambda_1 - \lambda_2)}{(r+1)^a} \cdot \frac{\omega_1 + \omega_2}{\omega_1 - \omega_2} + \frac{\omega_2 \lambda_2 - \omega_1 \lambda_1}{\omega_1 - \omega_2}\right]. \end{aligned}$$

When condition (2) of the theorem is satisfied we can get $E[\sum \omega U(1, 2)] < E[\sum \omega U(2, 1)]$. The proof of Theorem 2.5 is completed. \square

3. The case of GT with random learning rate of job

Now, we will discuss the case of group scheduling. All jobs in the model can be processed at zero time. In a group, the job can be worked continuously. The machine must have preparation time before

entering the next group, and it is subject to the classical learning effect hypothesis. p is a random with learning effect based on the position in the group, and the learning rate follows a uniform distribution.

Our model is expressed as

$$1 \left| p_{ij}^k = p_{ij} k^{-a_{ij}}, a_{ij} \sim U(0, \lambda_{ij}), G, s_i^r = s_i r^{-a} \right| E[f(C_j)].$$

The assumptions and symbols of the model (see Table 2) are as follows:

Table 2. Symbols of the model.

Symbol	Description
G_i	The i -th group
n_i	Number of jobs in group G_i
n	Number of total jobs, $n_1 + n_2 + \dots + n_i = n$
J_{ij}	The j -th job in group G_i , $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n_i$
a	Learning rate for group installation, $a < 0$
s_i	Normal installation time of group G_i , $i = 1, 2, \dots, m$
s_i^r	Actual installation time of group G_i in number r , $s_i^r = s_i r^a$
a_{ij}	Learning rate of job j in group G_i , $a_{ij} > 0$
p_{ij}	The random processing time of the job J_{ij} , obeys the uniform distribution in the interval $(0, \lambda_{ij})$
p_{ij}^k	Random processing time of job J_{ij} in position k of group G_{ij} , $p_{ij}^k = p_{ij} k^{a_{ij}}$
Q, Q'	Job group sequence
σ_1, σ_2	Partial workpiece sequence
$E[C_{jn_j}(Q)]$	Expected completion time of the last job of group G_j in sequence Q
$E[C_{in_i}(Q')]$	Expected completion time of the last job of group G_i in sequence Q'
$E(T_0)$	Expected completion time of the last job of σ_1

Theorem 3.1. For $1 \left| p_{ij}^k = p_{ij} k^{-a_{ij}}, a_{ij} \sim U(0, \lambda_{ij}), G, s_i^r = s_i r^{-a} \right| E(C_{\max})$, when the jobs in the group satisfy: For all J_{ik}, J_{il} , $p_{ik} \leq p_{il} \Rightarrow \lambda_{ik} \leq \lambda_{il}$, the internal parts of the group are in accordance with λ_{ik} large rule, the group is on the basis of the non-decreasing order of s_i , we can get the optimal solution.

Proof. Suppose $s_i \leq s_j$, and $a > 0$. We can get

$$\begin{aligned} & E[C_{jn_j}(Q)] - E[C_{in_i}(Q')] \\ &= E(T_0) + s_i r^{-a} + E\left(\sum_{k=1}^{n_i} p_{ik} k^{-a_{ik}}\right) + s_j (r+1)^{-a} + E\left(\sum_{k=1}^{n_j} p_{jk} k^{-a_{jk}}\right) \\ &\quad - E(T_0) - s_j r^{-a} - E\left(\sum_{k=1}^{n_j} p_{jk} k^{-a_{jk}}\right) - s_i (r+1)^{-a} - E\left(\sum_{k=1}^{n_i} p_{ik} k^{-a_{ik}}\right) \\ &= (s_i - s_j) \cdot [r^{-a} - (r+1)^{-a}] < 0. \end{aligned}$$

Then, we can get the sorting rules of the group by repeating the exchange operation.

The scheduling problem of jobs in a group can be summed up as

$$1 \left| p_{ij}^k = p_{ij} k^{-a_{ij}}, a_{ij} \sim U(0, \lambda_{ij}) \right| E(C_{\max}),$$

Theorem 2.1 has been proved.

We have completed the proof of Theorem 3.1. \square

Theorem 3.2. For $1 \left| p_{ij}^k = p_{ij} k^{-a_{ij}}, a_{ij} \sim U(0, \lambda_{ij}), G, s_i^r = s_i r^{-a} \right| E(\sum C_j)$, if $\frac{s_i}{n_i} \leq \frac{s_j}{n_j}$, the larger λ_{ij} of the job in the group, the better the arrangement. The group is arranged by the non-decreasing order of $\frac{\sum_{l,k=1}^{n_i} \frac{p_{ik}}{\ln(k^{\lambda_{ik}})} \cdot \left(1 - \frac{1}{k^{\lambda_{ik}}}\right)}{n_i}$, we can get the optimal sort.

Proof. Assumptions are the same as Theorem 3.1.

$$\text{And } \frac{s_i}{n_i} \leq \frac{s_j}{n_j}, \frac{\sum_{l,k=1}^{n_i} \frac{p_{ik}}{\ln(k^{\lambda_{ik}})} \cdot \left(1 - \frac{1}{k^{\lambda_{ik}}}\right)}{n_i} \leq \frac{\sum_{l,k=1}^{n_j} \frac{p_{jk}}{\ln(k^{\lambda_{jk}})} \cdot \left(1 - \frac{1}{k^{\lambda_{jk}}}\right)}{n_j}, \text{ then}$$

$$\begin{aligned} & \left\{ E \left[\sum_{l=1}^{n_i} C_{il}(Q) \right] + E \left[\sum_{l=1}^{n_j} C_{jl}(Q) \right] \right\} - \left\{ E \left[\sum_{l=1}^{n_i} C_{il}(Q') \right] + E \left[\sum_{l=1}^{n_j} C_{jl}(Q') \right] \right\} \\ &= n_i E(t_0) + n_i s_i r^{-a} + E \left[\sum_{k=1, l=1}^{n_i} (n_i - l + 1) p_{il} k^{-a_{il}} \right] \\ & \quad + n_j E[C_{in_i}(Q)] + n_j s_j (r+1)^{-a} + E \left[\sum_{k=1, l=1}^{n_j} (n_j - l + 1) p_{jl} k^{-a_{jl}} \right] \\ & \quad - n_i E[C_{jn_j}(Q')] - n_i s_i (r+1)^{-a} - E \left[\sum_{k=1, l=1}^{n_i} (n_i - l + 1) p_{il} k^{-a_{il}} \right] \\ & \quad - n_j E(t_0) - n_j s_j r^{-a} - E \left[\sum_{k=1, l=1}^{n_j} (n_j - l + 1) p_{jl} k^{-a_{jl}} \right] \\ &= (n_i + n_j) \cdot (s_i - s_j) \cdot [r^a - (r+1)^a] + (n_j s_i - n_i s_j) \cdot (r+1)^a \\ & \quad + \left[n_j \sum_{l,k=1}^{n_i} \frac{p_{ik}}{\ln(k^{\lambda_{ik}})} \cdot \left(1 - \frac{1}{k^{\lambda_{ik}}}\right) - n_i \sum_{l,k=1}^{n_j} \frac{p_{jk}}{\ln(k^{\lambda_{jk}})} \cdot \left(1 - \frac{1}{k^{\lambda_{jk}}}\right) \right] \leq 0. \end{aligned}$$

Then, we just need to repeat the previous method to complete the proof of the group order rule.

Secondly, for the scheduling problem of jobs in a group, we can reduce it to problem $1 \left| p_j^r = p_j r^{-a_{ij}}, a_{ij} \sim U(0, \lambda_{ij}) \right| E(\sum \omega_j C_j)$, and $\omega_j = 1$, Theorem 2.2 has been proved.

Therefore, we have completed the proof. \square

4. Algorithms and examples

The following are heuristic algorithms and examples for Theorems 3.1 and 3.2.

Algorithm 1

Step 1: When $p_{ik} \leq p_{il} \Rightarrow \lambda_{ik} \leq \lambda_{il}$, the priority principle shall be followed λ_{ij} in group;

Step 2: Priority arrangement for groups with small s_i .

The group G_i time complexity is $O(n_i \log n_i)$, the total time complexity of step 1 is $\sum_{i=1}^m O(n_i \log n_i)$; the optimal order time complexity of step 2 is $O(m \log m)$. The time complexity of Algorithm 1 is $O(n \log n)$.

An example of Algorithm 1:

Example 1. $m = 2$, $G_1 = \{J_{11}, J_{12}, J_{13}\}$, $G_2 = \{J_{21}, J_{22}\}$, $s_1 = 2$, $s_2 = 3$, $a = 2$, $a_{11} \sim U(0, 2)$, $a_{12} \sim U(0, 4)$, $a_{13} \sim U(0, 3)$, $a_{21} \sim U(0, 2)$, $a_{22} \sim U(0, 4)$, $p_{11} = 1$, $p_{12} = 3$, $p_{13} = 2$, $p_{21} = 4$, $p_{22} = 5$.

solution. Step 1: In G_1 , because $\lambda_{11} = 2 < \lambda_{13} = 3 < \lambda_{12} = 4$, $p_{11} = 1 < p_{13} = 2 < p_{12} = 3$; $\lambda_{21} = 2 < \lambda_{22} = 4$, $p_{21} = 4 < p_{22} = 5$, satisfy the consistency assumption, the order is $J_{12} \rightarrow J_{13} \rightarrow J_{11}$, $J_{22} \rightarrow J_{21}$;

Step 2: Because $s_1 = 2 < s_2 = 3$, so $G_1 \rightarrow G_2$.

The solution of this example is $E(C_{\max}) = 13.07$.

Algorithm 2

Step 1: Arrange the production of jobs in the group by the priority of λ_{ij} ;

Step 2: When the requirements are met $\frac{s_i}{n_i} \leq \frac{s_j}{n_j}$ in the group, the group processing shall be arranged

according to the non-decreasing order of $\frac{\sum_{l,k=1}^{n_i} \frac{p_{lk}}{\ln(k^{\lambda_{lk}})} \cdot \left(1 - \frac{1}{k^{\lambda_{lk}}}\right)}{n_i}$.

The complexity analysis of Algorithm 2 is the same as Algorithm 1.

We give an example of Algorithm 2.

Example 2. Same as Example 1.

solution. Step 1: Compare the values of λ_{ij} , we can get $J_{12} \rightarrow J_{13} \rightarrow J_{11}$, $J_{22} \rightarrow J_{21}$;

Step 2: Because $\frac{s_1}{n_1} = \frac{2}{3} < \frac{s_2}{n_2} = \frac{3}{2}$, $\frac{\sum \frac{p_{1j}}{\ln(k^{\lambda_{1j}})} \cdot \left(1 - \frac{1}{k^{\lambda_{1j}}}\right)}{n_1} = 0.866 < \frac{\sum \frac{p_{2j}}{\ln(k^{\lambda_{2j}})} \cdot \left(1 - \frac{1}{k^{\lambda_{2j}}}\right)}{n_2} = 2.844$. The group order is $G_1 \rightarrow G_2$.

After calculation, we have $E(C_{\max}) = 50.15$.

5. Conclusions

For the single machine stochastic scheduling and group stochastic scheduling problems established in this paper, we consider that the learning rate is a random variable, which is not available in previous literature. We find that such problems are very general, and we can also find examples in real life. For example, if workers participate in learning a new project training, their mastery time of new technology may be random, which can be expressed by random variables. After transforming the problem, we give corresponding theoretical assumptions for the proposed problem, and then give the solution of the optimal order. For the case of group technology, we give a numerical example to verify the theoretical results.

Next, we can combine it with intelligent algorithms to solve the problems of large number of jobs, machine maintenance, waiting time for job processing, and multi-stage processing of jobs.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by Shanghai Philosophy and Social Sciences General Project (2022BGL010); the Post-funded Project of the National Social Science Fund Project of China (22FGLB109); Laboratory of Philosophy and Social Sciences for Data Intelligence and Rural Revitalization of Dabie Mountains, Lu'an, Anhui 237012, China; Key projects of Humanities and Social Sciences in Colleges and Universities of Anhui Province (SK2018A0397).

Conflict of interest

The authors declare no conflicts of interest.

References

1. D. Biskup, Single-machines scheduling with learning considerations, *Eur. J. Oper. Res.*, **115** (1999), 173–178. [https://doi.org/10.1016/S0377-2217\(98\)00246-X](https://doi.org/10.1016/S0377-2217(98)00246-X)
2. G. Moshiov, Scheduling problems with a learning effect, *Eur. J. Oper. Res.*, **132** (2001), 687–693. [https://doi.org/10.1016/S0377-2217\(00\)00175-2](https://doi.org/10.1016/S0377-2217(00)00175-2)
3. G. Moshiov, Parallel machine scheduling with a learning effect, *J. Oper. Res. Soc.*, **52** (2001), 1165–1169. <https://doi.org/10.1057/palgrave.jors.2601215>
4. G. Mosheiov, J. B. Sidney, Scheduling with general job-dependent learning curves, *Eur. J. Oper. Res.*, **147** (2003), 665–670. [https://doi.org/10.1016/S0377-2217\(02\)00358-2](https://doi.org/10.1016/S0377-2217(02)00358-2)
5. G. Mosheiov, J. B. Sidney, Note on scheduling with general learning curves to minimize the number of tardy jobs, *J. Oper. Res. Soc.*, **56** (2005), 110–112. <https://doi.org/10.1057/palgrave.jors.2601809>
6. A. Bachman, A. Janiak, Scheduling jobs with position-dependent processing times, *J. Oper. Res. Soc.*, **55** (2004), 257–264. <https://doi.org/10.1057/palgrave.jors.2601689>
7. W. C. Lee, C. C. Wu, H. J. Sung, A bi-criterion single-machine scheduling problem with learning considerations, *Acta Inform.*, **40** (2004), 303–315. <https://doi.org/10.1007/s00236-003-0132-9>
8. W. C. Lee, C. C. Wu, Minimizing total completion time in a two-machine flowshop with a learning effect, *Int. J. Prod. Econ.*, **88** (2004), 85–93. [https://doi.org/10.1016/S0925-5273\(03\)00179-8](https://doi.org/10.1016/S0925-5273(03)00179-8)
9. A. Janiak, R. Rudek, Complexity results for single-machine scheduling with positional learning effects, *J. Oper. Res. Soc.*, **59** (2008), 1430. <https://doi.org/10.1057/palgrave.jors.2602622>
10. C. L. Zhao, Q. L. Zhang, H. Y. Tang, Machine scheduling problems with a learning effect, *Dyn. Contin. Discrete Impuls. Syst. Ser. A Math. Anal.*, **11** (2004), 741–750.
11. M. B. Cheng, S. J. Sun, Y. Yu, A note on flow shop scheduling problems with a learning effect on no-idle dominant machines, *Appl. Math. Comput.*, **184** (2007), 945–949. <https://doi.org/10.1016/j.amc.2006.05.206>

12. T. Eren, E. Güner, Minimizing total tardiness in a scheduling problem with a learning effect, *Appl. Math. Model.*, **31** (2007), 1351–1361. <https://doi.org/10.1016/j.apm.2006.03.030>
13. X. G. Zhang, G. L. Yan, Single-machine scheduling problems with a sum-of-processing-time-based learning function, *Int. J. Comb.*, **2009** (2009), 1–8. <https://doi.org/10.1155/2009/624108>
14. X. G. Zhang, G. L. Yan, W. Z. Huang, G. C. Tang, A note on machine scheduling with sum-of-logarithm-processing-time-based and position-based learning effects, *Inform. Sci.*, **187** (2012), 298–304. <https://doi.org/10.1016/j.ins.2011.11.001>
15. J. B. Liu, X. F. Pan, Minimizing Kirchhoff index among graphs with a given vertex bipartiteness, *Appl. Math. Comput.*, **291** (2016), 84–88. <https://doi.org/10.1016/j.amc.2016.06.017>
16. J. B. Liu, Y. Bao, W. T. Zheng, Analyses of some structural properties on a class of hierarchical scale-free networks, *Fractals*, **30** (2022), 2250136. <https://doi.org/10.1142/S0218348X22501365>
17. M. L. Pinedo, *Scheduling: theory, algorithms and systems*, Cham: Springer, 2016. <https://doi.org/10.1007/978-3-319-26580-3>
18. M. Pinedo, E. Rammouz, A note on stochastic machine scheduling subject to breakdown and repair, *Probab. Eng. Inform. Sci.*, **2** (1988), 41–49. <https://doi.org/10.1017/S0269964800000619>
19. J. B. G. Frenk, A general framework for stochastic one-machine scheduling problems with zero release times and no partial ordering, *Probab. Eng. Inform. Sci.*, **5** (1991), 297–315. <https://doi.org/10.1017/S0269964800002102>
20. Y. B. Zhang, X. Y. Wu, X. Zhou, Stochastic scheduling problems with general position-based learning effects and stochastic breakdowns, *J. Sched.*, **16** (2013), 331–336. <https://doi.org/10.1007/s10951-012-0306-9>
21. M. Ji, X. Y. Tang, X. Zhang, T. C. E. Cheng, Machine scheduling with deteriorating jobs and DeJong’s learning effect, *Comput. Indust. Eng.*, **91** (2016), 42–47. <https://doi.org/10.1016/j.cie.2015.10.015>
22. S. J. Qin, S. X. Liu, H. B. Kuang, Piecewise linear model for multiskilled workforce scheduling problems considering learning effect and project quality, *Math. Probl. Eng.*, **2016** (2016), 1–11. <https://doi.org/10.1155/2016/3728934>
23. X. Zhang, Y. Wang, S. K. Bai, Single-machine group scheduling problems with deteriorating and learning effect, *Int. J. Syst. Sci.*, **47** (2016), 2402–2410. <https://doi.org/10.1080/00207721.2014.998739>
24. J. Y. Xu, C. C. Wu, Y. Q. Yin, C. L. Zhao, Y. T. Chiou, W. C. Lin, An order scheduling problem with position-based learning effect, *Int. J. Syst. Sci.*, **74** (2016), 175–186. <https://doi.org/10.1016/j.cor.2016.04.021>
25. Y. B. Wu, J. J. Wang, Single-machine scheduling with truncated sum-of-processing-times-based learning effect including proportional delivery times, *Neural Comput. Appl.*, **27** (2016), 937–943. <https://doi.org/10.1007/s00521-015-1910-3>
26. J. L. Vile, J. W. Gillard, P. R. Harper, V. A. Knight, Time-dependent stochastic methods for managing and scheduling emergency medical services, *Oper. Res. Health Care*, **8** (2016), 42–52. <https://doi.org/10.1016/j.orhc.2015.07.002>

27. O. Souissi, R. Benmansour, A. Artiba, An accelerated MIP model for the single machine scheduling with preventive maintenance, *IFAC*, **49** (2016), 1945–1949. <https://doi.org/10.1016/j.ifacol.2016.07.915>
28. J. B. Liu, J. Zhao, J. Min, J. D. Cao, The Hosoya index of graphs formed by a fractal graph, *Fractals*, **27** (2019), 1950135. <https://doi.org/10.1142/S0218348X19501354>
29. J. B. Liu, C. X. Wang, S. H. Wang, B. Wei, Zagreb indices and multiplicative Zagreb indices of Eulerian graphs, *Bull. Malays. Math. Sci. Soc.*, **42** (2019), 67–78. <https://doi.org/10.1007/s40840-017-0463-2>
30. H. T. Li, Stochastic single-machine scheduling with learning effect, *IEEE Trans. Eng. Manag.*, **64** (2017), 94–102. <https://doi.org/10.1109/TEM.2016.2618764>
31. M. D. Toksari, B. Atalay, Some scheduling problems with job rejection and a learning effect, *Comput. J.*, **66** (2023), 866–872. <https://doi.org/10.1093/comjnl/bxab201>
32. J. C. Chen, Y. Y. Chen, T. L. Chen, Y. H. Lin, Multi-project scheduling with multi-skilled workforce assignment considering uncertainty and learning effect for large-scale equipment manufacturer, *Comput. Indust. Eng.*, **169** (2022), 108240. <https://doi.org/10.1016/j.cie.2022.108240>
33. Y. Shi, S. Mahdian, J. Blanchet, P. Glynn, A. Y. Shin, D. Scheinker, Surgical scheduling via optimization and machine learning with long-tailed data, *Health Care Manag. Sci.*, 2022, In press.
34. J. B. Wang, X. Jia, J. X. Yan, S. H. Wang, J. Qian, Single machine group scheduling problem with makespan objective and a proportional linear shortening, *RAIRO Oper. Res.*, **56** (2022), 1523–1532. <https://doi.org/10.1051/ro/2022078>
35. I. Ham, K. Hitomi, T. Yoshida, *Group technology: applications to production management*, Kluwer-Nijhoff Publishing, 1985. <https://doi.org/10.1007/978-94-009-4976-8>
36. W. C. Lee, C. C. Wu, A note on single-machine group scheduling problems with position-based learning effect, *Appl. Math. Model.*, **33** (2009), 2159–2163. <https://doi.org/10.1016/j.apm.2008.05.020>
37. W. H. Yang, S. Chand, Learning and forgetting effects on a group scheduling problem, *Eur. J. Oper. Res.*, **187** (2008), 1033–1044. <https://doi.org/10.1016/j.ejor.2006.03.065>
38. X. G. Zhang, G. L. Yan, Single-machine group scheduling problems with deteriorated and learning effect, *Appl. Math. Comput.*, **216** (2010), 1259–1266. <https://doi.org/10.1016/j.amc.2010.02.018>
39. W. M. Ma, L. Sun, L. Ning, N. N. Lin, Group scheduling with deterioration and exponential learning effect processing times, *Syst. Eng. Theory Pract.*, **37** (2017), 205–211.
40. L. Sun, L. Ning, J. Z. Huo, Group scheduling problems with time-dependent and position-dependent Dejong’s learning effect, *Math. Probl. Eng.*, **2020** (2020), 1–8. <https://doi.org/10.1155/2020/5161872>
41. J. B. Liu, Y. Bao, W. T. Zheng, S. Hayat, Network coherence analysis on a family of nested weighted n -polygon networks, *Fractals*, **29** (2021), 2150260. <https://doi.org/10.1142/S0218348X21502601>
42. W. X. Li, C. L. Zhao, Single machine scheduling problem with multiple due windows assignment in a group technology, *J. Appl. Math. Comput.*, **48** (2015), 477–494. <https://doi.org/10.1007/s12190-014-0814-1>

43. X. X. Liang, M. Q. Liu, Y. B. Feng, J. B. Wang, L. S. Wen, Solution algorithms for single-machine resource allocation scheduling with deteriorating jobs and group technology, *Eng. Optim.*, **52** (2020), 1184–1197. <https://doi.org/10.1080/0305215X.2019.1638920>
44. L. Y. Wang, M. Q. Liu, J. B. Wang, Y. Y. Lu, W. W. Liu, Optimization for due-date assignment single-machine scheduling under group technology, *Complexity*, **2021** (2021), 1–9. <https://doi.org/10.1155/2021/6656261>
45. D. Y. Wang, C. M. Ye, Group scheduling with learning effect and random processing time, *J. Math.*, **2021** (2021), 1–6. <https://doi.org/10.1155/2021/6685149>
46. A. Jain, A. Jain, An approach for optimisation of flexible flow shop group scheduling with sequence dependent set-up time and preventive maintenance, *Int. J. Comput. Aided Eng. Technol.*, **16** (2022), 40–66. <https://doi.org/10.1504/IJCAET.2022.119537>



AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)