

Research Article

Group Scheduling with Learning Effect and Random Processing Time

Dingyu Wang ^{1,2} and Chunming Ye ¹

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

²College of Finance and Mathematics, West Anhui University, Luan, Anhui, China

Correspondence should be addressed to Chunming Ye; yechm6464@163.com

Received 27 December 2020; Accepted 1 July 2021; Published 12 July 2021

Academic Editor: Antonio Di Crescenzo

Copyright © 2021 Dingyu Wang and Chunming Ye. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we establish a stochastic grouping scheduling model. In the model, there is no installation time between the jobs in the same group, but each group has an installation time before processing. There are learning effects between groups and within groups, and the completion time of jobs is a random variable. We take the long expected schedule and the expected total completion time as the objective function, and the noninterruptible static priority strategy is obtained. At the same time, heuristic algorithms and examples are given.

1. Introduction

Since Biskup [1] researched on a single-machine scheduling which is based on position learning effect, this kind of problem has become a hot research topic, and the research results have been very fruitful. Moshiov [2] studied a single-machine scheduling problem and proved that the SPT ordering rules is the optimal solution to minimize makespan and EDD ordering rules is optimal order to minimize the maximum delay. The authors in [3–5] have also carried out a lot of research studies on this problem. Mosheiov and Sidney [6] proved that model $1|p_{jr} = p_j r^a|C_{\max}$, $1|p_{jr} = p_j r^a|\sum C_j$, and $1|p_{jr} = p_j r^{a_i}, d_j = d|(\sum \omega_1 C_j + \omega_2 T_j + \omega_3 d)$ can be transformed into assignment problem, time complexity is $O(n^3)$. Mosheiov and Sidney [7] studied the problem $1|p_{jr} = p_j r^{a_i}, d_j = d|\sum U_j$. They transformed the problem into a classic assignment problem, and the time complexity of the problem is $O(n^3 \log n)$. Lin [8] proved the problem to minimize the tardy jobs when learning rates of the jobs. Bachman and Janiak [9] discussed the problem in special cases $1|p_{jr} = p_j r^a, d_j = d|\sum \omega_j C_j$, and $1|p_{jr} = p_j r^a|\sum \omega p_j C_j$. The former is based on the nonincreasing order, and the latter, according to the SPT sequence, can get the optimal solution. Zhao et al. [10] gave a solution to question

$1|p_{jr} = p_j r^a|\sum \omega_j C_j$. When $p_i \leq p_j \Rightarrow w_i \geq w_j$, the WSPT order rule is used. When $p_i \leq p_j \Rightarrow d_i \leq d_j$, by using EDD ordering rules for the problem of minimizing the maximum tardiness problem, we can get optimal solutions. Liu and Cao [11] solve a kind of difficult problem with the structure network of N prism. Cheng et al. [12] found that there is no gap between the jobs in the process of the jobs, that is to say, the jobs are continuous processing. In [13–15], the authors continue to study this kind of problem and get many results.

As people pay attention to the processing time, time of arrival, and uncertainty of machine breakdown, the problem of stochastic scheduling has been paid more and more attention. Pinedo and Hadavi [16], Pinedo and Rammouz [17], Frenk [18], and Liu et al. [19] have carried out a lot of pioneering work on stochastic scheduling problems, and their objective function are mostly the normal function. Then, the content of the research is also more and more diversified. In addition, Zhang et al. [20] continued to study this question. Li [21] considered a stochastic scheduling problem. However, in real life, the machining time is affected by many factors, the randomness of the universal existence. The learning effect of the machine has not changed. The authors in [22–28] continued to study this kind of problem and obtained good results in many aspects.

In the classical scheduling problems, most of the research studies on the problem that the installation time is needed to replace the jobs are assumed to be independent of the scheduling of machines and jobs, so the installation time is usually incorporated into the processing time of the jobs. Ham et al. [29] first proposed the “group technology,” but found that the production of products according to a certain same attribute or grouping can greatly improve the production efficiency, thus breaking through the classical assumption of sorting and scheduling. In group technology, the workpiece is divided into several groups according to the similar production technology. The workpiece in the group is processed continuously, and there is no installation time or cost between the parts in the group. Lee and Wu [30] proposed a model: $1|p_{ij}^{r,k} = p_{ij}r^{a_1}k^{a_2}, G, s_i^r = s_i r^{a_1}|C_{\max}$ and $1|p_{ij}^{r,k} = p_{ij}r^{a_1}k^{a_2}, G, s_i^r = s_i r^{a_1}|\sum C_j$, where $p_{ij}^{r,k}$ represents the processing time of J_{ij} in group G_i and r and k represent the position of groups and jobs, respectively, and $a_1 (< 0)$ and $a_2 (< 0)$ represent the learning factor. Yang and Chand [31] constructed three basic models: there is no forgetting, partial forgetting, and total forgetting in job processing, and the processing time of jobs is related to the location. The authors in [32–35] continue to study this problem and obtain many important results.

In this paper, we establish a stochastic grouping scheduling model, and we take the long expected schedule and the expected total completion time as the objective function.

2. Two Stochastic Grouping Scheduling Problems with Uniform Processing Time and Learning Effect

For the group scheduling and scheduling problem of this model, we assume that all jobs can be processed at zero; the processing time of jobs is a random variable, which obeys uniform distribution and has location-based learning effect; there is no installation time between jobs in the same group, but each group has an installation time before processing; the group installation time obeys the classical learning effect hypothesis.

The assumptions and symbols of the model are as follows:

- m : the number of groups in the sequence, $m \geq 2$
- G_i : group $i, i = 1, 2, \dots, m$
- 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_i : learning rates of jobs in group $G_i, a_i < 0$
- p_{ij} : the random processing time of the job J_{ij} , which 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 $G_i, p_{ij}^k = p_{ij}k^{a_i}$
- $E(C_{ij})$: expected completion time of job J_{ij}
- $f(C_j)$: function of processing time of job J_j , such as $C_{\max}, \sum C_j$

Now, we consider the stochastic problem of group scheduling, discuss the optimal group sequence and job sequence under the noninterruptible static priority strategy, and give their algorithms. This problem can be expressed by three parameters:

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

Theorem 1. For problem $1|p_{ij}^k = p_{ij}k^{a_i}, p_{ij} \sim U(0, \lambda_{ij}), G, s_i^r = s_i r^a | E(C_{\max})$, if the groups are arranged according to the nondecreasing order of the basic installation time $s_i (i = 1, 2, \dots, m)$ and the jobs in each group are arranged in the nondecreasing order of parameter $\lambda_{ij} (i, j \text{ are natural numbers, and } i \in [1, m], j \in [1, n_i])$, the optimal algorithm under the noninterruptible static priority strategy can be obtained.

Proof. Firstly, we prove that the optimal algorithm of noninterruptible static priority strategy by the non-subtraction order of parameters λ_{ij} , that is, the ESPT order, can be obtained. Considering a group, this problem can be attributed to

$$1|p_{ij}^k = p_{ij}k^{a_i}, p_{ij} \sim U(0, \lambda_{ij}) | E(C_{\max}). \tag{2}$$

Suppose S and S' are two sequences and J_{is} and $J_{it} (s < t)$ are two jobs; denote $S = (\pi_1, J_{is}, J_{it}, \pi_2)$ and $S' = (\pi_1, J_{it}, J_{is}, \pi_2)$. π_1 and π_2 are job sequences except job J_{is} and J_{it} . J_{is} and J_{it} are processing in S at r th and $(r + 1)$ th positions, respectively, but in S' , the case is reverse. $E(t_0)$ is the expected completion time in addition to the jobs J_{is} and J_{it} in the sequence. Here, we only need to compare $E[C_{it}(S)]$ and $E[C_{is}(S')]$.

By assuming conditions, we obtain

$$\begin{aligned} E[C_{it}(S)] &= E(t_0) + E[p_{is}^k(S)] + E[p_{it}^{k+1}(S)] \\ &= E(t_0) + \frac{\lambda_{is}k^a}{2} + \frac{\lambda_{it}(k+1)^a}{2}, \end{aligned} \tag{3}$$

$$\begin{aligned} E[C_{is}(S')] &= E(t_0) + E[p_{it}^k(S')] + E[p_{is}^{k+1}(S')] \\ &= E(t_0) + \frac{\lambda_{it}k^a}{2} + \frac{\lambda_{is}(k+1)^a}{2}. \end{aligned}$$

So,

$$E[C_{it}(S)] - E[C_{is}(S')] = \frac{1}{2} (\lambda_{is} - \lambda_{it}) \cdot [k^a - (k+1)^a] < 0. \tag{4}$$

Secondly, we prove the group sequence rule. Let $s_i < s_j, Q = \{\sigma_1, G_i, G_j, \sigma_2\}$ and $Q' = \{\sigma_1, G_j, G_i, \sigma_2\}$. Let the

expected completion time of the last job of group G_j in sequence Q be $E[C_{j n_j}(Q)]$ and the group G_i in sequence Q' be $E[C_{i n_i}(Q')]$. If $E[C_{j n_j}(Q)] \leq E[C_{i n_i}(Q')]$ is satisfied, it can be proved that sequence Q is superior to sequence Q' .

According to the hypothesis, we can obtain

$$E[C_{j n_j}(Q)] = E[C_{i n_i}(Q)] + s_j(r+1)^a + E\left(\sum_{k=1}^{n_j} p_{jk} k^{a_j}\right), \tag{5}$$

$$E[C_{i n_i}(Q')] = E[C_{j n_j}(Q')] + s_i(r+1)^a + E\left(\sum_{k=1}^{n_i} p_{ik} k^{a_i}\right), \tag{6}$$

where

$$E[C_{i n_i}(Q)] = E(t_0) + s_i r^a + E\left(\sum_{k=1}^{n_i} p_{ik} k^{a_i}\right), \tag{7}$$

$$E[C_{j n_j}(Q')] = E(t_0) + s_j r^a + E\left(\sum_{k=1}^{n_j} p_{jk} k^{a_j}\right),$$

where $E(t_0)$ is the expected completion time of the last job of group G_i and G_j . It is noted that $s_i < s_j$ and $a < 0$, by the difference between (1) and (2), we can obtain

$$E[C_{j n_j}(Q)] - E[C_{i n_i}(Q')] = (s_i - s_j) \cdot [r^a - (r+1)^a] < 0. \tag{8}$$

So, sequence Q is superior to sequence Q' . Therefore, by repeating such a swap operation, all groups can be arranged in nonsubtractive order of group installation time. Theorem 1 is proved.

By proving Theorem 1, we can design a simple algorithm for problem

$$1 | p_{ij}^k = p_{ij} k^{a_i}, \quad p_{ij} \sim U(0, \lambda_{ij}), G, s_i^r = s_i r^a | E(C_{\max}). \tag{9}$$

The time complexity of Algorithm 1 is $O(n \log n)$.

Next, we give an example of Theorem 1. □

Example 1. $m = 2, G_1 = \{J_{11}, J_{12}\}, G_2 = \{J_{21}, J_{22}, J_{23}\}, s_1 = 3, s_2 = 4, a_1 = -0.5, a_2 = -0.4, a = -0.3, \lambda_{11} = 2, \lambda_{12} = 1, \lambda_{21} = 2, \lambda_{22} = 4, \text{ and } \lambda_{23} = 3.$

Solution. The solution of Example 1 is as follows:

Step 1: in group G_1 , the optimal order of jobs is $J_{12} \rightarrow J_{11}$; in group G_2 , the optimal order of jobs is $J_{21} \rightarrow J_{23} \rightarrow J_{22}$

Step 2: because of $s_1 = 3 < s_2 = 4$, the groups are sorted as $G_1 \rightarrow G_2$

Therefore, the expected minimum maximum completion time of this example is $E(C_{\max}) = 11.1086$.

Next, we consider another stochastic problem of group scheduling. We have the following conclusion.

Theorem 2. *For the problem $1 | p_{ij}^k = p_{ij} k^{a_i}, p_{ij} \sim U(0, \lambda_{ij}) | E(\sum C_j)$, if the group installation time and the number of jobs in the group satisfy the consistent relationship $(s_i/n_i) \leq (s_j/n_j), i \in [1, m], j \in [1, m]$, then the jobs in the group are arranged according to the nondecreasing order of the parameters λ_{ij} , and the groups are in the nondecreasing order of $\sum_{l,k=1}^{n_i} \lambda_{il} k^{a_i}/n_i$. We can get the optimal algorithm under the noninterruptible static priority policy.*

Proof. First of all, we prove that the optimal algorithm under the noninterruptible static priority strategy can be obtained by arranging the jobs in the group according to the nondecreasing order of the parameters λ_{ij} . When only one group G_i is considered, the problem can be transformed into

$$1 | p_{ij}^k = p_{ij} k^{a_i}, \quad p_{ij} \sim U(0, \lambda_{ij}) | E\left(\sum_{j=1}^{n_j} C_{ij}\right). \tag{10}$$

Suppose S and S' are two sequences and J_{is} and J_{it} ($s < t$) are two jobs; denote $S = (\pi_1, J_{is}, J_{it}, \pi_2)$ and $S' = (\pi_1, J_{it}, J_{is}, \pi_2)$. π_1 and π_2 are job sequences, except job J_{is} and J_{it} . The meaning of J_{is}, J_{it} , and $E(t_0)$ are the same in Theorem 1. In order to prove that sequence S is superior to sequence S' , we just need proof $E[C_{is}(S)] + E[C_{it}(S)] \leq E[C_{is}(S')] + E[C_{it}(S')]$.

According to the definition, we can obtain

$$\begin{aligned} E[C_{is}(S)] &= E(t'_0) + E[p_{is} k^{a_i}] = E(t'_0) + \frac{1}{2} \lambda_{is} k^{a_i}, \\ E[C_{it}(S)] &= E(t'_0) + E[p_{is} k^{a_i}] + E[p_{it} (k+1)^{a_i}] = E(t'_0) + \frac{1}{2} \lambda_{is} k^{a_i} + \frac{1}{2} \lambda_{it} (k+1)^{a_i}, \\ E[C_{it}(S')] &= E(t'_0) + E[p_{it} k^{a_i}] = E(t'_0) + \frac{1}{2} \lambda_{it} k^{a_i}, \\ E[C_{is}(S')] &= E(t'_0) + E[p_{it} k^{a_i}] + E[p_{is} (k+1)^{a_i}] = E(t'_0) + \frac{1}{2} \lambda_{it} k^{a_i} + \frac{1}{2} \lambda_{is} (k+1)^{a_i}. \end{aligned} \tag{11}$$

By hypothesis $a_i < 0$ and when $s < t$, there is $\lambda_{is} < \lambda_{it}$; therefore,

Step 1: according to the nondecreasing order of parameter λ_{ij} , we arrange the processing of the job in the group
 Step 2: arrange the groups according to the nondecreasing order of the basic installation time s_i of the groups

ALGORITHM 1: Heuristic algorithm.

$$\begin{aligned} & E[C_{is}(S)] + E[C_{it}(S)] - E[C_{it}(S')] - E[C_{is}(S')] \\ &= (\lambda_{is} - \lambda_{it}) \cdot \left[k^{a_i} - \frac{1}{2}(k+1)^{a_i} \right] = (\lambda_{is} - \lambda_{it}) \cdot \left[\frac{1}{2}(k^{a_i} - (k+1)^{a_i}) + \frac{1}{2}(k+1)^{a_i} \right] \leq 0. \end{aligned} \quad (12)$$

That is, $E[C_{is}(S)] + E[C_{it}(S)] \leq E[C_{is}(S')] + E[C_{it}(S')]$. In the same way, the jobs in the group are arranged according to the nondecreasing order of parameter λ_{ij} through the operation of pairwise exchange, and the optimal algorithm under the noninterruptible static priority strategy can be obtained.

Secondly, we prove the group sequence rule. Suppose $s_i < s_j$, $Q = \{\sigma_1, G_i, G_j, \sigma_2\}$, and $Q' = \{\sigma_1, G_j, G_i, \sigma_2\}$. Let the expected total completion times of jobs in group G_i and G_j of sequence Q be $E[\sum_{l=1}^{n_i} C_{il}(Q)]$ and $E[\sum_{l=1}^{n_j} C_{jl}(Q)]$ and the

expected total completion times of jobs in group G_i and G_j of sequence Q' be $E[\sum_{l=1}^{n_i} C_{il}(Q')]$ and $E[\sum_{l=1}^{n_j} C_{jl}(Q')]$. Then, if the relation $E[\sum_{l=1}^{n_i} C_{il}(Q)] + E[\sum_{l=1}^{n_j} C_{jl}(Q)] \leq E[\sum_{l=1}^{n_i} C_{il}(Q')] + E[\sum_{l=1}^{n_j} C_{jl}(Q')]$ is satisfied, it can be proved that sequence Q is superior to sequence Q' .

According to the same assumption in the definition and Theorem 1, we can obtain

$$E\left[\sum_{l=1}^{n_i} C_{il}(Q)\right] = n_i E(t_0) + n_i s_i r^a + E\left[\sum_{k=1, l=1}^{n_i} (n_i - l + 1) p_{il} k^{a_i}\right], \quad (13)$$

$$E\left[\sum_{l=1}^{n_j} C_{jl}(Q)\right] = n_j E[C_{in_i}(Q)] + n_j s_j (r+1)^a + E\left[\sum_{k=1, l=1}^{n_j} (n_j - l + 1) p_{jl} k^{a_j}\right]. \quad (14)$$

And, we can also obtain

$$E\left[\sum_{l=1}^{n_j} C_{jl}(Q')\right] = n_j E(t_0) + n_j s_j r^a + E\left[\sum_{k=1, l=1}^{n_j} (n_j - l + 1) p_{jl} k^{a_j}\right], \quad (15)$$

$$E\left[\sum_{l=1}^{n_i} C_{il}(Q')\right] = n_i E[C_{jn_j}(Q')] + n_i s_i (r+1)^a + E\left[\sum_{k=1, l=1}^{n_i} (n_i - l + 1) p_{il} k^{a_i}\right]. \quad (16)$$

Notice $(s_i/n_i) \leq (s_j/n_j)$ and $(\sum_{l,k=1}^{n_i} \lambda_{il} k^{a_i}/n_i) \leq (\sum_{l,k=1}^{n_j} \lambda_{jl} k^{a_j}/n_j)$; from (5)-(16), we can obtain

$$\begin{aligned} & \left(E\left[\sum_{l=1}^{n_i} C_{il}(Q)\right] + E\left[\sum_{l=1}^{n_j} C_{jl}(Q)\right] \right) - \left(E\left[\sum_{l=1}^{n_i} C_{il}(Q')\right] + E\left[\sum_{l=1}^{n_j} C_{jl}(Q')\right] \right) \\ &= (n_i + n_j) \cdot (s_i - s_j) \cdot [r^a - (r+1)^a] + (n_j s_i - n_i s_j) \cdot (r+1)^a + \frac{1}{2} \left(n_j \sum_{l,k=1}^{n_i} \lambda_{il} k^{a_i} - n_i \sum_{l,k=1}^{n_j} \lambda_{jl} k^{a_j} \right) \leq 0. \end{aligned} \quad (17)$$

Step 1: in the group, if the parameter λ_{ij} of the job is large, it should be arranged first
 Step 2: when the group installation time and the number of jobs in the group meet the consistent relationship $(s_i/n_i) \leq (s_j/n_j)$, the group processing is arranged according to the nondecreasing order of $\sum_{l,k=1}^{n_i} \lambda_{il} k^{a_i}/n_i$

ALGORITHM 2: Heuristic algorithm.

By repeating the similar exchange procedure, we can make the group satisfy the nondecreasing order of $\sum_{l,k=1}^{n_i} \lambda_{il} k^{a_i}/n_i$, thus completing the proof of group sequence rule.

Therefore, we have completed the proof of Theorem 2.

We can design an algorithm for problem:

$$1 \mid p_{ij}^k = p_{ij} k^{a_i}, \quad p_{ij} \sim U(0, \lambda_{ij}) \mid E(\sum C_j). \quad (18)$$

The intragroup G_i time complexity in step 1 is $O(n_i \log n_i)$ and the total time complexity 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 2 is $O(n \log n)$.

Next, we give an example of Theorem 2. \square

Example 2. $m = 2$, $G_1 = \{J_{11}, J_{12}\}$, $G_2 = \{J_{21}, J_{22}, J_{23}\}$, $s_1 = 3$, $s_2 = 6$, $a_1 = -0.5$, $a_2 = -0.4$, $a = -0.3$, $\lambda_{11} = 2$, $\lambda_{12} = 1$, $\lambda_{21} = 2$, $\lambda_{22} = 4$, and $\lambda_{23} = 3$.

Solution. The solution of Example 2 is as follows:

Step 1: in group G_1 , the optimal order of jobs is $J_{12} \rightarrow J_{11}$; in group G_2 , the optimal order of jobs is $J_{21} \rightarrow J_{23} \rightarrow J_{22}$

Step 2: because of $(s_1/n_1) = (3/2) < (s_2/n_2) = (6/3)$ and $(\sum_{l,k=1}^{n_1} \lambda_{1l} k^{a_1}/n_1) = 1.3536 < (\sum_{l,k=1}^{n_2} \lambda_{2l} k^{a_2}/n_2) = 2.3215$, the groups are sorted as $G_1 \rightarrow G_2$

Therefore, the expected minimum total completion time of this example is $E(\sum C_j) = 39.5113$.

3. Conclusions

For the expected total completion time problem, the optimal solution can be obtained in polynomial time when there is a consistent relationship between the number of jobs in the group and the sum of the expected processing times of the jobs in the group. When the consistency relation is not satisfied, if the number of groups is small and the number of jobs in the group is small, the optimal algorithm can be found by comparison enumeration; however, when the number of groups is large, the polynomial time algorithm cannot be given, and the complexity of the algorithm is unknown.

Considering the randomness of machine learning effect and the randomness of job processing time, we can further discuss the possibility that machine processing time obeys different types of random functions and machine has the random learning index.

Data Availability

All data included in this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Shanghai Science and Technology Commission "Science and Technology Innovation Action Plan" Soft Science Key Project (20692104300), National Nature Science Foundation (no. 71840003), Technology Development Project of University of Shanghai for Science and Technology (2018KJFZ043), Key Projects of Humanities and Social Sciences in Colleges and Universities of Anhui Province (SK2018A0397), and Foundation for Key Project of Natural Science Research of West Anhui University (WXZR201702).

References

- [1] D. Biskup, "Single-machines scheduling with learning considerations," *European Journal of Operational Research*, vol. 115, pp. 73–178, 1999.
- [2] G. Moshiov, "Scheduling problems with a learning effect," *European Journal of Operational Research*, vol. 132, pp. 687–693, 2001.
- [3] G. Moshiov, "Parallel machine scheduling with a learning effect," *Journal of the Operational Research Society*, vol. 52, pp. 1165–1169, 2001.
- [4] W.-C. Lee, C.-C. Wu, and H.-J. Sung, "A bi-criterion single-machine scheduling problem with learning considerations," *Acta Informatica*, vol. 40, no. 4, pp. 303–315, 2004.
- [5] W. Lee and C. C. Wu, "Minimizing total completion time in a two-machine flowshop with a learning effect," *International Journal of Production Economics*, vol. 88, no. 1, pp. 85–93, 2004.
- [6] G. Mosheiov and J. B. Sidney, "Scheduling with general job-dependent learning curves," *European Journal of Operational Research*, vol. 147, no. 3, pp. 665–670, 2003.
- [7] G. Mosheiov and J. B. Sidney, "Note on scheduling with general learning curves to minimize the number of tardy jobs," *Journal of the Operational Research Society*, vol. 56, no. 1, pp. 110–112, 2005.
- [8] B. M. T. Lin, "Complexity results for single-machine scheduling with positional learning effects," *Journal of the Operational Research Society*, vol. 58, no. 8, pp. 1099–1102, 2007.
- [9] A. Bachman and A. Janiak, "Scheduling jobs with position-dependent processing times," *Journal of the Operational Research Society*, vol. 55, no. 3, pp. 257–264, 2004.
- [10] C. L. Zhao, Q. L. Zhang, and H. Y. Tang, "Machine scheduling problems with learning effects," *Dynamics of Continuous, Discrete and Impulsive Systems, Series A: Mathematical Analysis*, vol. 11, pp. 741–750, 2004.

- [11] J. B. Liu and J. Cao, "Applications of laplacian spectra for n-prism networks," *Neurocomputing*, vol. 198, Article ID 698C73, 2016.
- [12] M. B. Cheng, S. J. Sun, and Y. Yu, "A note on flow shop scheduling problems with a learning effect on no-idle dominant machines," *Applied Mathematics and Computation*, vol. 184, Article ID 9458C949, 2007.
- [13] T. Eren and E. Güner, "Minimizing total tardiness in a scheduling problem with a learning effect," *Applied Mathematical Modelling*, vol. 31, no. 7, pp. 1351–1361, 2007.
- [14] X. G. Zhang and G. L. Yan, "Single-machine scheduling problems with a sum-of-processing-time-based learning function," *International Journal of Combinatorics*, vol. 2009, Article ID 624108, 8 pages, 2009.
- [15] X. Zhang, G. Yan, W. Huang, and G. Tang, "A note on machine scheduling with sum-of-logarithm-processing-time-based and position-based learning effects," *Information Sciences*, vol. 187, no. 1, pp. 298–304, 2012.
- [16] M. Pinedo and K. Hadavi, *Scheduling: Theory, Algorithms and Systems Development*, C. N. Potts and L. N. Van Wassenhove, Eds., pp. 35–42, Springer, Berlin, Germany, 1991.
- [17] M. Pinedo and E. Rammouz, "A note on stochastic machine scheduling subject to breakdown and repair," *Probability in the Engineering & Information Sciences*, vol. 2, no. 1, pp. 41–49, 1987.
- [18] J. B. G. Frenk, "A general framework for stochastic one-machine scheduling problems with zero release times and no partial ordering," *Probability in the Engineering and Information Sciences*, vol. 5, no. 3, pp. 297–315, 1991.
- [19] J. B. Liu, S. Wang, C. Wang, and S. Hayat, "Further results on computation of topological indices of certain networks," *IET Control Theory & Applications*, vol. 11, no. 13, pp. 2065–2071, 2017.
- [20] Y. Zhang, X. Wu, and X. Zhou, "Stochastic scheduling problems with general position-based learning effects and stochastic breakdowns," *Journal of Scheduling*, vol. 16, no. 3, pp. 331–336, 2013.
- [21] H. Li, "Stochastic single-machine scheduling with learning effect," *IEEE Transactions on Engineering Management*, vol. 64, no. 1, pp. 94–102, 2017.
- [22] M. Ji, X. Tang, X. Zhang, and T. C. E. Cheng, "Machine scheduling with deteriorating jobs and DeJong's learning effect," *Computers & Industrial Engineering*, vol. 91, pp. 42–47, 2016.
- [23] S. Qin, S. Liu, and H. Kuang, "Piecewise linear model for multiskilled workforce scheduling problems considering learning effect and project quality," *Mathematical Problems in Engineering*, vol. 2016, no. 4, 11 pages, Article ID 3728934, 2016.
- [24] X. Zhang and G. Yan, "Single-machine group scheduling problems with deteriorated and learning effect," *Applied Mathematics & Computation*, vol. 47, no. 10, pp. 2402–2410, 2016.
- [25] J. Xu, C.-C. Wu, Y. Yin, C. Zhao, Y.-T. Chiou, and W.-C. Lin, "An order scheduling problem with position-based learning effect," *Computers & Operations Research*, vol. 74, pp. 175–186, 2016.
- [26] Y.-B. Wu and J.-J. Wang, "Single-machine scheduling with truncated sum-of-processing-times-based learning effect including proportional delivery times," *Neural Computing and Applications*, vol. 27, no. 4, pp. 937–943, 2016.
- [27] J. L. Vile, J. W. Gillard, P. R. Harper, and V. A. Knight, "Time-dependent stochastic methods for managing and scheduling Emergency Medical Services," *Operations Research for Health Care*, vol. 8, pp. 42–52, 2016.
- [28] O. Souissi, R. Benmansour, and A. Artiba, "An accelerated MIP model for the single machine scheduling with preventive maintenance," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1945–1949, 2016.
- [29] I. Ham, K. Hitomi, and T. Yoshida, *Group Technology: Applications to Production Management*, Kluwer-Nijhoff, Boston, MA, USA, 1985.
- [30] W.-C. Lee and C.-C. Wu, "A note on single-machine group scheduling problems with position-based learning effect," *Applied Mathematical Modelling*, vol. 33, no. 4, pp. 2159–2163, 2009.
- [31] W.-H. Yang and S. Chand, "Learning and forgetting effects on a group scheduling problem," *European Journal of Operational Research*, vol. 187, no. 3, pp. 1033–1044, 2008.
- [32] J. B. Liu, J. Zhao, and Z. Q. Cai, "On the generalized adjacency, laplacian and signless laplacian spectra of the weighted edge corona networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 540, Article ID 123073, 2019.
- [33] X. G. Zhang and G. L. Yan, "Single-machine group scheduling problems with deteriorated and learning effect," *Applied Mathematics and Computation*, vol. 216, Article ID 12598C1266, 2010.
- [34] W. M. Ma, L. Sun, L. Ning et al., "Group scheduling with deterioration and exponential learning effect processing times," *Systems Engineering Theory & Practice*, vol. 37, no. 1, Article ID 2058C211, 2017.
- [35] L. Sun, L. Ning, and J.-Z. Huo, "Group scheduling problems with time-dependent and position-dependent DeJong's learning effect," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5161872, 8 pages, 2020.