Int J Adv Manuf Technol (2010) 48:739–746
DOI 10.1007/s00170-009-2308-0

ORIGINAL ARTICLE

Scheduling with past-sequence-dependent setup times and learning effects on a single machine

Xue-Ru Wang · Ji-Bo Wang · Wen-Jun Gao · Xue Huang

Received: 26 May 2009 / Accepted: 9 September 2009 / Published online: 19 September 2009

Abstract In this paper, we study the single-machine scheduling problems with learning effect and setup time considerations. The setup times are proportional to the length of the already-processed jobs, i.e., the setup times are past-sequence-dependent (p-s-d). The objective functions are to minimize the sum of the quadratic job completion times, the total waiting time, the total weighted completion time, the maximum lateness, the total absolute differences in waiting times, and the sum of earliness penalties subject to no tardy jobs, respectively. We show that the sum of the quadratic job completion times minimization problem, the total waiting time minimization problem, the total absolute differences in waiting times minimization problem, and the sum of earliness penalties minimization problem subject to no tardy jobs can be solved in polynomial time, respectively. We also show that the total weighted completion time minimization problem and the maximum lateness minimization problem can be solved in polynomial time under some special cases.

Keywords Scheduling · Single machine · Setup times · Learning effect

X.-R. Wang (✉) · J.-B. Wang · W.-J. Gao · X. Huang
Operations Research and Cybernetics Institute,
School of Science,
Shenyang Institute of Aeronautical Engineering,
Shenyang 110136, China
e-mail: wxrlittle@163.com

1 Introduction

In classical scheduling problems, the processing time of a job is assumed to be a constant. However, in many realistic problems of operations management, both machines and workers can improve their performance by repeating the production operations. Therefore, the actual processing time of a job is shorter if it is scheduled later in a sequence. This phenomenon is known as the "learning effect" in the literature. Biskup [1] and Cheng and Wang [2] were among the pioneers that brought the concept of learning into the field of scheduling. Biskup [1] assumed that the processing time of a job is a log-linear learning curve, i.e., if job J_j is scheduled in position r in a sequence, its actual processing time is $p_{jr} = p_j r^a$, where p_j is the normal processing time of job J_j, $a \leq 0$ is a constant learning effect. He proved that single-machine scheduling problems to minimize the sum of job flow times and the total deviations of job completion times from a common due date are polynomial solvable. Cheng and Wang [2] considered a single-machine scheduling problem in which the job processing times decrease as a result of learning. A volume-dependent piecewise linear processing time function was used to model the learning effect. The objective is to minimize the maximum lateness. They showed that the problem is NP-hard in the strong sense and then identified two special cases that are polynomially solvable. They also proposed two heuristics and analysed their worst-case performance. Later, Mosheiov [3, 4] investigated several other single-machine problems and the minimum total flow time problem on identical parallel machines. Mosheiov and Sidney [5] considered a job-dependent learning curve, where the learning rate of some jobs is faster than that of the others. They showed that the makespan minimization problem, the total flow time minimization problem, a due date assignment problem, and total flow time minimization on unrelated parallel machines remain polynomially solvable. Wang and Xia [6] considered flow shop scheduling problems with a learning effect. The objective was to minimize one of two regular performance measures, namely the makespan and the total flow time. They gave a heuristic algorithm with a worst-case error bound of m for each criterion, where m is the number of machines. They also found polynomial time solutions for two special cases of the problem, i.e., identical processing times on each machine and an increasing series of dominating machines. Wang [7] considered the same problem of Wang and Xia [6]. He suggested the use of Johnson's rule as a heuristic algorithm for two-machine flow shop scheduling to minimize the makespan. Kuo and Yang [8] considered a single-machine scheduling problem with a time-dependent learning effect. The time-dependent learning effect of a job is assumed to be a function of the total normal processing time of the jobs scheduled in front of it. They showed that the shortest processing time (SPT) sequence is the optimal sequence for the objective of minimizing the total completion time. Eren and Guner [9] considered the single-machine total tardiness problem with a learning effect. They developed an integer programming model for the problem. Wang, Ng, Cheng, and Liu [10] considered the same model of Kuo and Yang [8]. They proved that the weighted shortest processing time (WSPT) rule, the earliest due date (EDD) rule, and the modified Moore–Hodgson algorithm can, under certain conditions, construct the optimal schedule for the problem to minimize the following three objectives: the total weighted completion time, the maximum lateness, and the number of tardy jobs, respectively. They also gave an error estimation for each of these rules for the general cases. Cheng, Wu, and Lee [11] considered some scheduling problems where the actual job processing time is a function of jobs already processed. Eren and Guner [12] considered a bicriteria parallel machine scheduling with a learning effect. They developed a mathematical programming model for solving the problem. Wang, Wang, Wang, Yin, Huang, and Feng [13] proposed a new sum-of-processing-time-based learning effect and deteriorating jobs. This model has two opposing parameters. One makes the processing time longer and the second one shorter. They showed that the makespan minimization problem can be solved by the largest processing times first (LPT) rule. Wang, Wang, Gao, Huang, and Feng [14] considered two single-machine scheduling problems with the effects of deterioration and learning. For the weighted sum of completion times minimization problem and the maximum lateness minimization problem, they gave two heuristics according to the corresponding problems without learning effect. They also gave the worst-case error bound for the heuristics. An extensive survey of research related to scheduling with learning effects was provided by Biskup [15].

On the other hand, it is reasonable and necessary to consider scheduling problems with setup times (Allahverdi, Gupta, and Aldowaisan [16]). There are two types of setup time or setup cost: sequence-independent and sequence-dependent. In the first case, the setup time/cost depends solely on the task to be processed, regardless of its preceding task. While in the sequence-dependent type, setup time/cost depends on both the task and its preceding task. For recent results and trends in scheduling problems with setup times or costs, the reader may refer to the recent review paper of Allahverdi, Ng, Cheng, and Kovalyov [17]. Koulamas and Kyparisis [18] first introduced a scheduling problem with past-sequence-dependent (p-s-d) setup times, i.e., the setup time is dependent on all already scheduled jobs. The objectives are the makespan, the total completion time, and the total absolute differences in completion times. They proved that the standard single-machine scheduling with p-s-d setup times and any of the above objectives can be solvable in polynomial time. They also extended their results to nonlinear p-s-d setup times.

However, to the best of our knowledge, apart from the recent papers of Kuo and Yang [19], Wang [20], Wang, Wang, Wang, Lin, Yin, and Wang [21], and Wang, Jiang, and Wang [22], the scheduling models considering the setup times and learning effect at the same time have not been investigated. Kuo and Yang [19] considered single-machine scheduling with p-s-d setup times and job-independent (job-dependent) learning effect. They considered the following objective functions: the makespan, the total completion time, the total absolute differences in completion times and the sum of earliness, tardiness, and common due-date penalty. They proposed the polynomial time algorithms to optimally solve the above objective functions. Wang [20] studied single-machine scheduling problems with p-s-d setup times and time-dependent learning effect. They proved that the makespan minimization problem, the total completion time minimization problem, and the sum of the quadratic job completion times minimization problem can be solved by the SPT rule, respectively. Wang, Wang, Wang, Lin, Yin, and Wang [21] considered single-machine scheduling problems with p-s-d setup times and exponential time-dependent learning effect. They proved that the makespan minimization problem, the total completion time minimization problem, and the sum of the quadratic job completion times minimization problem can be solved by the SPT rule, respectively. Wang, Jiang, and Wang [22] studied single-machine scheduling problems with p-s-d setup times and the effects of deterioration and learning. They proved that the makespan minimization problem, the total completion time minimization problem, and the sum of the δth power of the job completion times minimization problem can be solved in polynomial time, respectively.

The phenomena of p-s-d setup times occurring can be found in many real-life situations. For example, in high-tech manufacturing environments, in which a batch of jobs consists of a group of electronic components mounted together on an integrated circuit (IC) board. These jobs must be processed one-by-one by a machine while they are mounted together on the board. The machine's operation on any of these components has an adverse effect on the "readiness" of all the other components that have not yet been processed due to the flow of electrical current through the IC board while the machine is operating. Once a component is fully processed, its condition is not affected by the subsequent operation of the machine even if it remains mounted on the IC board. The degree of "un-readiness" of any component is proportional to the amount of time it has been exposed to the machine's operation on other components. Consequently, prior to a component's processing, a setup operation, proportional to the degree of "un-readiness" of the respective component, is needed to restore it to "full-readiness" status; this setup operation has no effect on the "readiness" of the remaining unprocessed components. The overall manufacturing process is completed when all components on the IC board have been processed by the machine (Koulamas and Kyparisis [18]). The learning effect in scheduling may arise in a company that produces similar jobs on a machine for a number of customers. In many cases, jobs will have different normal processing times due to varying (order) quantities or slightly different components that make up the products. Nevertheless, by processing one job after the other, the skills of the workers continuously improve, e.g., the ability to perform setups faster or to deal with the operations of the machines and software or handle raw materials or components or similar operations of the jobs at a greater pace (Biskup [1]).

In this paper, we consider the same model as that of Kuo and Yang [19], but with different objective functions. The remaining part of this paper is organized as follows: In Section 2, we formulate the model. In Section 3, we consider several single-machine scheduling problems. The last section presents the conclusions.

2 Problems description

Assume that there are n independent jobs to be processed on a single machine. Each of them is available at time zero. The machine can handle only one job at a time and is permanently available at time zero. For each job J_j $(j = 1, 2, \ldots, n)$, the value of its normal processing time p_j is known. Let $p_{[k]}$ be the normal processing time of a job if it is scheduled in the kth position in a sequence. As in Biskup [1] and Kuo and Yang [19], we assume that the actual processing time of job J_j if it is scheduled in position r is given by:

$$p^A_{jr} = p_j r^a, r, j = 1, 2, \ldots, n, \tag{1}$$

where $a \leq 0$ is a constant learning effect. Also, as in Koulamas and Kyparisis [18], we assume that the p-s-d setup time of job $J_{[r]}$ if it is scheduled in position r is given by:

$$s_{[1]} = 0 \text{ and } s_{[r]} = b \sum_{i=1}^{r-1} p^A_{[i]}, \tag{2}$$

where $b \geq 0$ is a normalizing constant, $\sum_{i=1}^{0} p_{[i]} := 0$. For convenience, we denote by LE the learning effect given by Eq. 1 (see Kuo and Yang [19]) and s_{psd} the p-s-d setup given by Eq. 2 (see Koulamas and Kyparisis [18]). Using the standard three-field notation scheme $\alpha|\beta|\gamma$ introduced by Graham et al. [23], our scheduling problem can be denoted as $1|\text{LE}, s_{\text{psd}}|\gamma$. Let C_j be the completion time of job J_j. In this paper, we will study the minimization of the following function: the sum of the quadratic job completion times $\sum_{j=1}^{n} C_j^2$, the total waiting time $\text{TW} = \sum_{j=1}^{n} W_j$ (where W_j represent the waiting time of job J_j, i.e., $W_j = C_j - p_j$, $j = 1, 2, \ldots, n$), the total weighted completion time $\sum_{j=1}^{n} w_j C_j$, the maximum lateness $L_{\max} = \max\{C_j - d_j | j = 1, 2, \ldots, n\}$, the total absolute differences in waiting times $\text{TADW} = \sum_{i=1}^{n} \sum_{j=i}^{n} |W_i - W_j|$, and the sum of earliness penalties subject to no tardy jobs $\sum_{j=1}^{n} g(E_j)$, where $d_j = d$ is a common due date for all the jobs, $E_j = d - C_j$ is the earliness of job J_j, and $g(x)$ is a strictly increasing function.

3 Main results

First, we give some lemmas; they are useful for the following theorems.

Lemma 1 *For a given schedule* $\pi = [J_1, J_2, \ldots, J_n]$ *of* $1|s_{psd}|\gamma$*, the completion* C_j *of job* J_j *is*

$$C_j(\pi) = \sum_{i=1}^{j}(b(j-i)+1)p_i^A.$$

Proof From Eqs. 1 and 2, we have

$$C_j(\pi) = \sum_{i=1}^{j}(s_i + p_i^A) = \sum_{i=1}^{j}\left(b\sum_{l=1}^{j-1} p_l^A + p_i^A\right)$$
$$= \sum_{i=1}^{j}(b(j-i)+1)p_i^A,$$

where $\sum_{i=1}^{0} p_l^A := 0$. □

Lemma 2 *(Kuo and Yang [19]) For the problem* $1|LE, s_{psd}|C_{max}$*, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* p_j *(the SPT rule).*

Lemma 3 *(Hardy et al. [24]) The sum of products* $\sum_{j=1}^{n} x_j y_j$ *is minimized if sequence* $x_1, x_2, \ldots, x_n$ *is ordered non-decreasingly and sequence* $y_1, y_2, \ldots, y_n$ *is ordered non-increasingly or vice versa, and it is maximized if the sequences are ordered in the same way.*

3.1 The $1|s_{psd}|\sum_{j=1}^{n} C_j^2$ scheduling problem

Townsend [25] considered a single-machine scheduling problem with a quadratic cost function of completion times, i.e., the sum of the quadratic job completion times. He showed that the problem $1||\sum C_j^2$ can be solved optimally by the SPT rule. By using the job interchanging technique, we can show that the solution of Townsend still holds for the problem $1|LE, s_{psd}|\sum C_j^2$.

Theorem 1 *For the problem* $1|LE, s_{psd}|\sum_{j=1}^{n} C_j^2$*, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* p_j *(the SPT rule).*

Proof Let π and π' be two job schedules, where the difference between π and π' is a pairwise interchange of two adjacent jobs J_j and J_k, that is, $\pi = [S_1, J_j, J_k, S_2]$, $\pi' = [S_1, J_k, J_j, S_2]$, where S_1 and S_2 are partial sequences and S_1 or S_2 may be empty. Furthermore, we assume that there are $r-1$ jobs in S_1. Thus, J_j and J_k are the rth and the $(r+1)$th jobs, respectively, in π and with $p_j \le p_k$. In order to show π dominates π', it suffices to show that (1) $C_k(\pi) \le C_j(\pi')$ and (2) $C_j^2(\pi) + C_k^2(\pi) \le C_k^2(\pi') + C_j^2(\pi')$. Under π, from Lemma 1, the completion times of jobs J_j and J_k are

$$C_j(\pi) = \sum_{i=1}^{r-1}(b(r-i)+1)p_{[i]}^A + p_j r^a.$$

$$C_k(\pi) = \sum_{i=1}^{r-1}(b(r+1-i)+1)p_{[i]}^A + (b+1)p_j r^a$$
$$+ p_k(r+1)^a.$$

Whereas, under π', they are

$$C_k(\pi') = \sum_{i=1}^{r-1}(b(r-i)+1)p_{[i]}^A + p_k r^a$$

and

$$C_j(\pi') = \sum_{i=1}^{r-1}(b(r+1-i)+1)p_{[i]}^A + (b+1)p_k r^a$$
$$+ p_j(r+1)^a.$$

The proof of part 1 is given in Lemma 2. In addition, from $p_j \le p_k$, we have $C_j(\pi) \le C_k(\pi')$; hence,

$$C_j^2(\pi) + C_k^2(\pi) \le C_k^2(\pi') + C_j^2(\pi').$$

This completes the proof of part 2 and, thus, of the theorem. □

3.2 The $1|LE, s_{psd}|TW$ scheduling problem

In this subsection, we consider the problem $1|s_{psd}|TW$. Clearly,

$$TW = \sum_{j=1}^{n} W_j = \sum_{r=1}^{n}(n-r)(s_{[r]} + p_{[r]}r^a)$$
$$= \sum_{r=1}^{n}(n-r)\left(1 + b\,\frac{n-r-1}{2}\right) r^a p_{[r]}. \quad (3)$$

Equation 3 can be viewed as the scalar product of two vectors, the $\lambda_r = (n-r)\left(1 + b\frac{n-r-1}{2}\right) r^a$ and $p_{[r]}$ vectors, respectively ($r = 1, 2, \ldots, n$). From Lemma 3, we know that Eq. 3 is minimized by sorting the elements of the λ_r and $p_{[r]}$ vectors in opposite orders. Since the elements of the λ_r vector are already sorted in non-increasing order, hence, we have the following theorem.

Theorem 2 *For the problem* $1|LE, s_{psd}|TW$*, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* p_j *(the SPT rule).*

3.3 The $1|LE, s_{psd}|\sum_{j=1}^{n} w_jC_j$ scheduling problem

In this subsection, we show that the total weighted completion time minimization problem can be solved in polynomial time under some special conditions.

Theorem 3 *For the problem* $1|LE, s_{psd}|\sum_{j=1}^{n} w_jC_j$*, if the jobs have agreeable weights, i.e.,* $p_j < p_k$ *implies* $w_j \geq w_k$ *for all the jobs* J_j *and* J_k*, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* p_j/w_j *(the WSPT rule).*

Proof Here, we still use the same notations as in the proof of Theorem 1. In order to show π dominates π', it suffices to show that (1) $C_k(\pi) \leq C_j(\pi')$ and (2) $w_jC_j(\pi) + w_kC_k(\pi) \leq w_kC_k(\pi') + w_jC_j(\pi')$.

The proof of part 1 is given in Lemma 2. We provide the proof of part 2 as follows.

From Theorem 1, the completion times of jobs J_j and J_k in sequence π and π' are

$$C_j(\pi) = \sum_{i=1}^{r-1}(b(r-i)+1)p_{[i]}^A + p_jr^a.$$

$$C_k(\pi) = \sum_{i=1}^{r-1}(b(r+1-i)+1)p_{[i]}^A + (b+1)p_jr^a + p_k(r+1)^a,$$

$$C_k(\pi') = \sum_{i=1}^{r-1}(b(r-i)+1)p_{[i]}^A + p_kr^a$$

and

$$C_j(\pi') = \sum_{i=1}^{r-1}(b(r+1-i)+1)p_{[i]}^A + (b+1)p_kr^a + p_j(r+1)^a.$$

So we have

$$\begin{aligned} &w_kC_k(\pi') + w_jC_j(\pi') - w_jC_j(\pi) - w_kC_k(\pi) \\ &\quad = b(w_j - w_k)\sum_{i=1}^{r-1} p_{[i]}^A \\ &\qquad + (w_j + w_k)(p_k - p_j)(r^a - (r+1)^a) \\ &\qquad + (br^a + (r+1)^a)(w_jp_k - w_kp_j). \end{aligned}$$

From $p_j/w_j \leq p_k/w_k$, we have $(br^a + (r+1)^a)(w_jp_k - w_kp_j) \geq 0$. In addition, from $p_j < p_k$, which implies $w_j \geq w_k$ and $r^a \geq (r+1)^a$, we have $(w_j + w_k)(p_k - p_j)(r^a - (r+1)^a) \geq 0$. Hence, $w_jC_j(\pi) + w_kC_k(\pi) \leq w_kC_k(\pi') + w_jC_j(\pi')$. This completes the proof of part 2 and, thus, the theorem. □

Using the similar method of Theorem 3, the following corollaries can be easily obtained.

Corollary 1 *For the problem* $1|LE, s_{psd}, p = p|\sum_{j=1}^{n} w_jC_j$*, there exists an optimal schedule in which jobs are sequenced in non-increasing order of* w_j*.*

Corollary 2 *For the problem* $1|LE, s_{psd}, w_j = kp_j|\sum_{j=1}^{n} w_jC_j$*, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* p_j *(the SPT rule).*

3.4 The $1|LE, s_{psd}|\sum_{j=1}^{n} w_jW_j$ scheduling problem

In this subsection, we show that the total weighted waiting time minimization problem can be solved in polynomial time under some special conditions.

Theorem 4 *For the problem* $1|LE, s_{psd}|\sum_{j=1}^{n} w_jW_j$*, if the jobs have agreeable weights, i.e.,* $p_j < p_k$ *implies* $w_j \geq w_k$ *for all the jobs* J_j *and* J_k*, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* p_j/w_j *(the WSPT rule).*

Proof Similar to the proof of Theorem 3. □

Using the similar method of Theorem 4, the following corollaries can be easily obtained.

Corollary 3 *For the problem* $1|LE, s_{psd}, p_j = p|\sum_{j=1}^{n} w_jW_j$*, there exists an optimal schedule in which jobs are sequenced in non-increasing order of* w_j*.*

Corollary 4 *For the problem* $1|LE, s_{psd}, w_j = kp_j|\sum_{j=1}^{n} w_jW_j$*, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* p_j *(the SPT rule).*

3.5 The $1|LE, s_{psd}|L_{max}$ scheduling problem

In this subsection, we show that the maximum lateness minimization problem can be solved in polynomial time under some special conditions.

Theorem 5 *For the problem* $1|LE, s_{psd}|L_{max}$*, if the jobs have agreeable conditions, i.e.,* $p_j < p_k$ *implies* $d_j \leq d_k$ *for all the jobs* J_j *and* J_k*, there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* d_j *(the EDD rule).*

Proof We still use the same notations mentioned above. Now we use the job interchanging technique to prove the theorem. From the proof of Theorem 1, under π, the lateness of jobs J_j and J_k are

$$L_j(\pi) = \sum_{i=1}^{r-1}(b(r-i)+1)p_{[i]}^A + p_j(r+1)^a - d_j,$$

$$L_k(\pi) = \sum_{i=1}^{r-1}(b(r+1-i)+1)p_{[i]}^A + (b+1)p_j r^a + p_k(r+1)^a - d_k,$$

whereas, under π', they are

$$L_k(\pi') = \sum_{i=1}^{r-1}(b(r-i)+1)p_{[i]}^A + p_k r^a - d_k,$$

$$L_j(\pi') = \sum_{i=1}^{r-1}(b(r+1-i)+1)p_{[i]}^A + (b+1)p_k r^a + p_j(r+1)^a - d_j.$$

If $d_j \le d_k$, we have $L_k(\pi') \le L_j(\pi')$. In addition, if $d_j \le d_k$ and $p_j \le p_k$, from Theorem 1, we have $L_k(\pi) \le L_j(\pi')$ and $L_j(\pi) \le L_j(\pi')$. Therefore, we have $\max\{L_j(\pi'), L_k(\pi')\} \ge \max\{L_j(\pi), L_k(\pi)\}$. This completes the proof of the theorem. □

Using the similar method of Theorem 3, the following corollaries can be easily obtained.

Corollary 5 *For the problem* $1|LE, s_{psd}, p_j = p|L_{max}$, *there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* d_j *(the EDD rule).*

Corollary 6 *For the problem* $1|LE, s_{psd}, d_j = d|L_{max}$, *there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* p_j *(the SPT rule).*

Corollary 7 *For the problem* $1|LE, s_{psd}, d_j = kp_j|L_{max}$, *there exists an optimal schedule in which jobs are sequenced in non-decreasing order of* d_j *(the EDD rule).*

3.6 The $1|LE, s_{psd}|TADW$ scheduling problem

In this subsection, we consider the single-machine scheduling problems with the objective of minimizing the total absolute differences in waiting times (TADW). This scheduling measure was first considered by Bagchi [26]. From Bagchi [26], we have

$$\text{TADW} = \sum_{j=1}^{n} W_j = \sum_{r=1}^{n} r(n-r)(s_{[r]} + p_{[r]}r^a)$$
$$= \sum_{r=1}^{n}\left[r(n-r) + b\sum_{j=r+1}^{n} j(n-j)\right] r^a p_{[r]}. \quad (4)$$

Equation 4 can be viewed as the scalar product of two vectors, the $\lambda_r = (r(n-r) + b\sum_{j=r+1}^{n} j(n-j))r^a$ and $p_{[r]}$ vectors, respectively ($r = 1, 2, \ldots, n$). From Lemma 3, we know that the problem $1|LE, s_{psd}|TADW$ can be solved in $O(n \log n)$ time by sorting the elements of the λ_r and $p_{[r]}$ vectors in opposite orders.

The application of the weight-matching solution approach is demonstrated in the following example (Bagchi [26]).

Example 1 Data: $n = 7, p_1 = 2, p_2 = 3, p_3 = 6, p_4 = 9, p_5 = 21, p_6 = 65, p_7 = 82, b = 0.1$. Assume now an 80% learning curve, i.e., $a = -0.322$ (Biskup [1]). The positional weights are: $\lambda_1 = 11, \lambda_2 = 11.199, \lambda_3 = 10.390, \lambda_4 = 8.703, \lambda_5 = 6.313, \lambda_6 = 3.370, \lambda_7 = 0$. According to Lemma 3, the optimal sequence is $[J_2, J_1, J_3, J_4, J_5, J_6, J_7]$, with TADW = 547.688.

3.7 Minimize the sum of earliness penalties

In this subsection we consider the problem $1|LE, s_{psd}|\sum_{j=1}^{n} g(E_j)$ under the condition $C_{[n]} \le d$. For the classical problem, there are some results in Chang and Schneeberger [27], and Qi and Tu [28]. A schedule is feasible if and only if there is no tardy job in the schedule. For the optimal schedule, it is obvious that (1) the last job completion time is d and (2) there is no idle time between the jobs, the idle time can only exist before the first job.

Lemma 4 *For a given schedule* $\pi = [J_1, J_2, \ldots, J_n]$ *of* $1|LE, s_{psd}|C_{max}$, *if the makespan is C, then the starting time of the first job is*

$$t_0 = C - \sum_{i=1}^{n}(b(n-i)+1)p_i i^a.$$

Based on the above lemma, we have the following theorem.

Theorem 6 *For problem* $1|LE, s_{psd}|\sum_{j=1}^{n} g(E_j)$, *an optimal schedule can be obtained by sequencing the jobs in non-increasing order of* p_j *(the LPT rule), where the first job starting time is* $t_0 = d - \sum_{i=1}^{n}(b(n-i)+1)p_i i^a$.

Proof Consider an optimal schedule π; suppose there are two adjacent jobs J_j and J_k, job J_j followed by job J_k, such that $p_j < p_k$, in which job J_k is scheduled in the $(r+1)$th position. Let the completion time of job J_k be C_0. Perform an adjacent pair-wise interchange on jobs J_j and J_k and get a new schedule π'; then, we have:

$$C_k = C_0, C_j = C_0 - (b(n-r-1)+1)p_k(r+1)^a,$$

$$E_k = d - C_0, E_j = d - C_0 + (b(n-r-1)+1)p_k(r+1)^a;$$

$$C'_j = C_0, C'_k = C_0 - (b(n-r-1)+1)p_j(r+1)^a,$$

$$E'_j = d - C_0, E'_k = d - C_0 + (b(n-r-1)+1)p_j(r+1)^a.$$

Since $p_j < p_k$, then $E'_k < E_j$, $E'_j = E_k$; thus, we have $g(E'_j) + g(E'_k) < g(E_k) + g(E_j)$.

The completion times of the jobs processed after jobs J_j and J_k are not affected by the interchange; the completion times of the jobs processed before jobs J_j and J_k become larger and earliness become smaller. Hence, the value of objective under π' is strictly less than that under π. This contradicts the optimality of π and proves the theorem. □

Remark When $g(x) = x$, $\sum_{j=1}^{n} g(E_j) = nd - \sum_{j=1}^{n} C_j$. Since nd is a constant, then the sum of completion times maximization problem can be solved by the LPT rule, as though the sum of completion times minimization problem can be solved by the SPT rule (Kuo and Yang [8]).

4 Conclusions

In this paper, we studied the problems of single-machine scheduling jobs with p-s-d setup times and learning effect. We proved that the sum of the quadratic job completion times minimization problem, the total waiting time minimization problem, the total absolute differences in waiting times minimization problem, and the sum of earliness penalties minimization problem subject to no tardy jobs can be solved in polynomial time, respectively. For some special cases, we also proved that the total weighted completion time minimization problem and the maximum lateness minimization problem are polynomially solvable. We note that the computational complexity of the total weighted completion time minimization problem and the maximum lateness minimization problem remains open. Future research may focus on considering these open problems or investigating the model in the context of other scheduling problems.

Acknowledgements We are grateful to the editor and two anonymous referees for their helpful comments on an earlier versions of this paper. This research was supported in part by the Science Research Foundation of the Educational Department of Liaoning Province, China, under grant number 20060662.

References

1. Biskup D (1999) Single-machine scheduling with learning considerations. Eur J Oper Res 115:173–178
2. Cheng TCE, Wang G (2000) Single machine scheduling with learning effect considerations. Ann Oper Res 98:273–290
3. Mosheiov G (2001) Scheduling problems with a learning effect. Eur J Oper Res 132:687–693
4. Mosheiov G (2001) Parallel machine scheduling with a learning effect. J Oper Res Soc 52:1165–1169
5. Mosheiov G, Sidney JB (2003) Scheduling with general job-dependent learning curves. Eur J Oper Res 147:665–670
6. Wang J-B, Xia Z-Q (2005) Flow-shop scheduling with a learning effect. J Oper Res Soc 56:1325–1330
7. Wang J-B (2005) Flow shop scheduling jobs with position-dependent processing times. J Appl Math Comput 18:383–391
8. Kuo W-H, Yang D-L (2006) Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. Eur J Oper Res 174:1184–1190
9. Eren T, Guner E (2007) Minimizing total tardiness in a scheduling problem with a learning effect. Appl Math Model 31:1351–1361
10. Wang J-B, Ng CT, Cheng TCE, Liu LL (2008) Single-machine scheduling with a time-dependent learning effect. Int J Prod Econ 111:802–811
11. Cheng TCE, Wu CC, Lee W-C (2008) Some scheduling problems with sum-of-processing-times-based and job-position-based learning effects. Inf Sci 178(11):2476–2487
12. Eren T, Guner E (2009) A bicriteria parallel machine scheduling with a learning effect. Int J Adv Manuf Technol 40:1202–1205
13. Wang L-Y, Wang J-B, Wang D, Yin N, Huang X, Feng E-M (2009) Single-machine scheduling with a sum-of-processing-time based learning effect and deteriorating jobs. Int J Adv Manuf Technol. doi:10.1007/s00170-009-1950-x
14. Wang L-Y, Wang J-B, Gao W-J, Huang X, Feng E-M (2009) Two single-machine scheduling problems with the effects of deterioration and learning. Int J Adv Manuf Technol. doi:10.1007/s00170-009-2145-1
15. Biskup D (2008) A state-of-the-art review on scheduling with learning effects. Eur J Oper Res 188:315–329
16. Allahverdi A, Gupta JND, Aldowaisan T (1999) A review of scheduling research involving setup considerations. Omega 27:219–239
17. Allahverdi A, Ng CT, Cheng TCE, Kovalyov MY (2008) A survey of scheduling problems with setup times or costs. Eur J Oper Res 187:985–1032
18. Koulamas C, Kyparisis GJ (2008). Single-machine scheduling problems with past-sequence-dependent setup times. Eur J Oper Res 187:1045–1049

19. Kuo W-H, Yang D-L (2007) Single-machine scheduling with past-sequence-dependent setup and learning effects. Inf Process Lett 102:22–26
20. Wang J-B (2008) Single-machine scheduling with past-sequence-dependent setup times and time-dependent learning effect. Comput Ind Eng 55:584–591
21. Wang J-B, Wang D, Wang L-Y, Lin L, Yin N, Wang W-W (2009) Single machine scheduling with exponential time-dependent learning effect and past-sequence-dependent setup times. Comput Math Appl 57:9–16
22. Wang J-B, Jiang Y, Wang G (2009) Single machine scheduling with past-sequence-dependent setup times and effects of deterioration and learning. Int J Adv Manuf Technol 41:1221–1226
23. Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Annal Discrete Math 5:287–326
24. Hardy GH, Littlewood JE, Polya G (1967) Inequalities. Cambridge University Press, London
25. Townsend W (1978) The single machine problem with quadratic penalty function of completion times: a branch-and-bound solution. Manage Sci 24:530–534
26. Bagchi UB (1989) Simultaneous minimization of mean and variation of flow-time and waiting time in single machine systems. Oper Res 37:118–125
27. Chang S, Schneeberger H (1988) Single machine scheduling to minimize weighted earliness subject to no tardy jobs. Eur J Oper Res 34:221–230
28. Qi X, Tu F-S (1998) Scheduling a single machine to minimize earliness penalties subject to the SLK due-date determination method. Eur J Oper Res 105:502–508