International Journal of Innovative Computing, Information and Control
Volume 12, Number 3, June 2016
ICIC International ©2016 ISSN 1349-4198
pp. 875–884

SCHEDULING WITH SIMULTANEOUS CONSIDERATION OF LEARNING EFFECT, DETERIORATING JOBS AND CONVEX RESOURCE DEPENDENT PROCESSING TIME

Lin Li[1], Shengwu Yang[1] and Ping Yan[2]

[1]School of Science
[2]School of Economics and Management
Shenyang Aerospace University
No. 37, Daoyi South Street, Shenyang 110136, P. R. China
{ lilinsau; ysw_0515 }@163.com; 0316yanping@126.com

Received October 2015; revised March 2016

Abstract. *This paper discusses the extension problems of scheduling with learning effect, deteriorating jobs and resource allocation simultaneously. For a convex resource allocation function, our first extension version is to find the optimal resource allocations and the optimal job sequence that minimize a cost function containing makespan, total completion (waiting) time and total absolute differences in completion (waiting) time subject to the constraint that the total resource cost is less than or equal to a fixed number. It also considers another extension version problem, i.e., minimizing the total compression cost subject to the constraint that a cost function containing makespan, total completion (waiting) time and total absolute differences in completion (waiting) time is less than or equal to a fixed number. We prove that these two extension versions can be solved in polynomial time respectively.*
Keywords: Scheduling, Learning effect, Deteriorating job, Resource allocation

1. **Introduction.** Scheduling problems and models with learning effects and/or deteriorating jobs and/or resource allocation constraint (controllable processing time) have received increasing attention in recent years. Extensive surveys of different scheduling models and problems with learning effects, and/or deteriorating jobs and/or resource allocation constraint (controllable processing time) could be found in Biskup [1], Janiak et al. [2], Shabtay and Steiner [3] and Janiak et al. [4] respectively. Many recent papers considered scheduling problems with learning effects. Wang et al. [5] discussed exponential learning effect, that is, jobs processing time was defined by an exponent function of the sum of the normal processing time of the already processed jobs. They considered different objective functions, such as, the makespan, the total completion time, and the sum of the quadratic job completion times. Wang and Wang [6] studied single machine multiple common due dates scheduling with learning effects. Job's penalty was assumed to be a linear function of the due date and the earliness/tardiness for the job. Their objective function was to minimize the total penalty for all jobs. Hsu et al. [7] considered an unrelated parallel machine scheduling problem with setup time and learning effects simultaneously. Job's setup time was proportional to the length of the already processed jobs. Their objective was to minimize the total completion time. Wang et al. [8] discussed single machine scheduling problem with truncated job-dependent learning effect. Job's actual processing time was a function, which depended on the job-dependent learning effect and a control parameter. Their objectives were to minimize the makespan, the earliness, tardiness and common (slack) due-date penalty, etc. Wang and Zhang [9] discussed the problem of minimizing the weighted sum of total completion time and makespan in a

permutation flowshop. Job's processing time was a function of the sum of the logarithms of the processing time of the jobs already processed and job's position in the sequence. Niu et al. [10] considered machine scheduling problems with a more general learning effect. Job's actual processing time was a function, which was the sum of the function of the processing time of the jobs already processed and job's position. They showed that some single machine scheduling problems were polynomially solvable under the proposed model. Many authors discussed scheduling problems with deteriorating jobs. Wang et al. [11] discussed single machine scheduling problem with deteriorating jobs. Job's processing time was a linear function of its execution starting time. For the jobs with chain precedence constraints, they proved that the weighted sum of squared completion times minimization problem with strong chains and weak chains could be solved in polynomial time respectively. Cheng et al. [12] proposed a scheduling model with an accelerating deterioration effect. This case always exists in the food manufacturing industry. They proved that the single-machine problems under the model to minimize the makespan, total completion time, total weighted completion time, maximum lateness, maximum tardiness, and total tardiness remain polynomially solvable. Yin et al. [13] introduced a deterioration model. Job's actual processing time depended not only on the starting time of the job but also on its scheduled position. The objective was to find the optimal schedule such that the makespan or total completion time was minimized. They proved that both problems could be solved in $O(n \log n)$ time. Some authors studied scheduling problems with learning effects and deteriorating jobs. Bai et al. [14] considered the groups' setup time as general linear function of groups' starting time. The jobs in the same group had general position-dependent and time-dependent learning effects. Their objective functions were to minimize the makespan and the sum of completion time, respectively. They also showed that these problems could be solved in polynomial time. Wang et al. [15] and Pakzad-Moghaddam et al. [16] discussed this kind of problem. Some researchers studied scheduling problems with resource allocation (controllable job processing time). Wang and Wang [17] considered single machine scheduling problem. Job's processing time was assumed to be a convex decreasing resource consumption function. Their objective function was to minimize the total amount of resource consumed subject to a constraint on total weighted flow time. Yang et al. [18] considered single machine scheduling problem with multiple due windows assignment and resource allocation. Yin et al. [19] discussed single-machine due window assignment and scheduling with a common flow allowance and controllable job processing time, subjected to unlimited or limited resource availability. Many papers considered single machine scheduling problems with learning effects and resource allocation (controllable job processing time). Yin and Wang [20] discussed the case that job's processing time was controllable variables with linear costs and also was defined as functions of positions in a schedule. They modeled the problem as assignment problem and solved them. Zhu et al. [21] considered learning effect and resource allocation in a group technology environment. Their objective functions were to minimize the weighted sum of makespan and total resource cost, and the weighted sum of total completion time and total resource cost. They proved that these problems were polynomially solvable under certain conditions. Lu et al. [22] discussed earliness-tardiness scheduling problem with due-date assignment. Job's processing time was a function of its position in a sequence and its resource allocation. Their aim was to minimize an integrated objective function, which included earliness, tardiness, due date assignment, and total resource consumption costs. They proposed a polynomial time algorithm. Li et al. [23] addressed a single machine due-window assignment scheduling problem based on a common flow allowance. Job's processing time was a function of its position in a sequence (learning effect) and its continuously divisible and non-renewable resource allocation. Their objective function was to minimize costs for earliness, tardiness, the window location, window size, makespan and resource consumption. For a convex or a linear function of the amount of a resource allocated to the job, they provided a polynomial time algorithm respectively. They also considered some extensions of the problem.

In recent papers, Wang et al. [24] and Li et al. [25] considered single machine scheduling problems with learning effects, deteriorating jobs and resource allocation (controllable job processing time). Li et al. [25] considered a scheduling problem with convex resource allocation, learning effect and deterioration effect simultaneously, i.e., the actual processing time of job J_j is $p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha$. k is a positive constant, $\alpha \leq 0$ is a learning index (Biskup [26]), $a_j \geq 0$ is a positive parameter representing the workload of job J_j, and $u_j > 0$ is the amount of a non-renewable resource allocated to job J_j. r is the position of job J_j, which is scheduled in a sequence. $t \geq 0$ is the start time of job J_j and $b \geq 0$ is the common deterioration rate. The objective function was to minimize a linear combination of makespan, total completion (waiting) time (TC (TW)), total absolute differences in completion (waiting) time (TADC (TADW)) and total convex resource allocation cost. By using the extended three-field notation scheme (Graham et al. [27]), they showed that the problem $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha|Z$ ($Z \in \{\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} + \delta_4 \sum_{j=1}^n v_j u_j, \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} + \delta_4 \sum_{j=1}^n v_j u_j\}$) could be solved in polynomial time. They also presented a polynomial time algorithm for the proposed model.

In the real life, we always meet the problems as minimizing time costs with limited resource constraint or minimizing resource costs with limited time costs constraint. It is useful and important to consider such problems. It is also helpful for people to make decision. This paper extends the results of Li et al. [25], by considering two extension versions of scheduling with simultaneous considerations of learning effect, deteriorating jobs and convex resource dependent processing time.

The rest of this paper is organized as follows. Notations and assumptions are given in Section 2. In Section 3, we address problems subject to limited resource cost. In Section 4, we address problems subject to limited schedule criterion. The last section presents the conclusions and some future research directions.

2. **Problem Statement.** We consider the problem of scheduling n jobs $J_1, J_2, \ldots, J_n$ on a single machine, where all the jobs are available for processing at time 0 and may not be preemption. As in Li et al. [25], we assume that the actual processing time of job J_j is

$$p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha. \tag{1}$$

For a given sequence $\pi = (J_1, J_2, \ldots, J_n)$, let $C_j = C_j(\pi)$ be the completion time of job J_j, $C_{\max} = \max\{C_j | j = 1, 2, \ldots, n\}$ be the makespan of all jobs, $\mathrm{TC} = \sum_{j=1}^n C_j$ ($\mathrm{TW} = \sum_{j=1}^n W_j$) be the total completion (waiting) time, $\mathrm{TADC} = \sum_{i=1}^n \sum_{j=i}^n |C_i - C_j|$ ($\mathrm{TADW} = \sum_{i=1}^n \sum_{j=i}^n |W_i - W_j|$) be the total absolute differences in completion (waiting) time, where $W_j = C_j - p_j$ is the waiting time of job J_j. The problems under discussion will be denoted by the three-field notation scheme (Graham et al. [27]).

3. **The Problems Subject to Limited Resource Cost.** In this section, we seek an optimal resource allocation combined with an optimal sequence of jobs so as to minimize $Z = \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ and $Z = \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}$ subject to $\sum_{j=1}^n v_j u_j \leq$

$\bar{U}$ respectively, where non-negative parameters δ_1, δ_2 and δ_3 are the weights, and v_j is the per unit cost associated with the resource allocation.

Let $[r]$ denote the rth position of a processing sequence. Then from Li et al. [25], we have

$$Z = \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} = \sum_{j=1}^{n} \omega_j p_{[j]} = \sum_{j=1}^{n} \Omega_j \left(\frac{a_{[j]}}{u_{[j]}} \right)^k, \tag{2}$$

where $\omega_j = \delta_1 + \delta_2(n+1-j) + \delta_3(j-1)(n-j+1)$ and

$$\begin{aligned}
&\Omega_1 = \omega_1 1^\alpha + b\omega_2 1^\alpha 2^\alpha + b\omega_3 1^\alpha (1+b2^\alpha) 3^\alpha + \ldots + b\omega_n 1^\alpha n^\alpha \prod_{l=2}^{n-1}(1+bl^\alpha)\\
&\Omega_2 = \omega_2 2^\alpha + b\omega_3 2^\alpha 3^\alpha + b\omega_4 2^\alpha (1+b3^\alpha) 4^\alpha + \ldots + b\omega_n 2^\alpha n^\alpha \prod_{l=3}^{n-1}(1+bl^\alpha)\\
&\Omega_3 = \omega_3 3^\alpha + b\omega_4 3^\alpha 4^\alpha + b\omega_5 3^\alpha (1+b4^\alpha) 5^\alpha + \ldots + b\omega_n 3^\alpha n^\alpha \prod_{l=4}^{n-1}(1+bl^\alpha)\\
&\ldots\\
&\Omega_{n-1} = \omega_{n-1}(n-1)^\alpha + b\omega_n (n-1)^\alpha n^\alpha\\
&\Omega_n = \omega_n n^\alpha.
\end{aligned} \tag{3}$$

Similarly, $Z = \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} = \sum_{j=1}^{n} \omega_j p_{[j]} = \sum_{j=1}^{n} \Omega_j \left(\frac{a_{[j]}}{u_{[j]}} \right)^k$, where $\omega_j = \delta_1 + \delta_2(n-j) + \delta_3 j(n-j)$ and Ω_j is calculated by Equation (3).

Obviously, for the problems $1 \left| p_j = \left(\left(\frac{a_j}{u_j} \right)^k + bt \right) r^\alpha, \sum_{j=1}^{n} v_j u_j \le \bar{U} \right| \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ and $1 \left| p_j = \left(\left(\frac{a_j}{u_j} \right)^k + bt \right) r^\alpha, \sum_{j=1}^{n} v_j u_j \le \bar{U} \right| \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}$, the optimal resource constraints are satisfied as equality respectively.

Lemma 3.1. *For a given sequence of the problem* $1 \left| p_j = \left(\left(\frac{a_j}{u_j} \right)^k + bt \right) r^\alpha, \sum_{j=1}^{n} v_j u_j \le \bar{U} \right| \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ $\left(1 \left| p_j = \left(\left(\frac{a_j}{u_j} \right)^k + bt \right) r^\alpha, \sum_{j=1}^{n} v_j u_j \le \bar{U} \right| \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} \right)$*, the optimal resource allocation can be determined by*

$$u_{[j]}^* = \frac{(\Omega_j)^{\frac{1}{k+1}} v_{[j]}^{-\frac{1}{k+1}} \left(a_{[j]}\right)^{\frac{k}{k+1}}}{\sum_{j=1}^{n} (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]} v_{[j]}\right)^{\frac{k}{k+1}}} \times \bar{U}, \tag{4}$$

where $\omega_j = \delta_1 + \delta_2(n+1-j) + \delta_3(j-1)(n-j+1)$ $(\omega_j = \delta_1 + \delta_2(n-j) + \delta_3 j(n-j))$ *for* $\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ $(\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW})$ *and* Ω_j *is calculated by Equation (3).*

Proof: For a given sequence $\pi = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n]} \right)$, the Lagrange function is

$$\begin{aligned}
&L(\mathbf{u},\lambda) \\
&= \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}(\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}) + \lambda\left(\sum_{j=1}^{n} v_{[j]}u_{[j]} - \bar{U}\right) \\
&= \sum_{j=1}^{n} \Omega_j \left(\frac{a_{[j]}}{u_{[j]}}\right)^k + \lambda\left(\sum_{j=1}^{n} v_{[j]}u_{[j]} - \bar{U}\right) \qquad (5)
\end{aligned}$$

where λ is the Lagrangian multiplier. Deriving (5) with respect to $u_{[j]}$ and λ respectively, we have

$$\frac{\partial L(\mathbf{u},\lambda)}{\partial u_{[j]}} = \lambda v_{[j]} - k\Omega_j \times \frac{\left(a_{[j]}\right)^k}{(u_{[j]})^{k+1}} = 0,\ \forall j = 1, 2, \ldots, n. \qquad (6)$$

$$\frac{\partial L(\mathbf{u},\lambda)}{\partial \lambda} = \sum_{j=1}^{n} v_{[j]}u_{[j]} - \bar{U} = 0 \qquad (7)$$

Using (6) and (7) we obtain

$$u_{[j]} = \frac{\left(k\Omega_j \left(a_{[j]}\right)^k\right)^{\frac{1}{k+1}}}{\left(\lambda v_{[j]}\right)^{\frac{1}{k+1}}} \qquad (8)$$

and

$$\lambda^{\frac{1}{k+1}} = \frac{\sum_{j=1}^{n} (k\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]}v_{[j]}\right)^{\frac{k}{k+1}}}{\bar{U}}. \qquad (9)$$

From (8) and (9), we have

$$u^*_{[j]} = \frac{(\Omega_j)^{\frac{1}{k+1}} v_{[j]}^{-\frac{1}{k+1}} \left(a_{[j]}\right)^{\frac{k}{k+1}}}{\sum_{j=1}^{n} (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]}v_{[j]}\right)^{\frac{k}{k+1}}} \times \bar{U}.$$

Lemma 3.2. *For a given sequence of the problem* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha,\ \sum_{j=1}^n v_j u_j \le \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}\ \left(1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha,\ \sum_{j=1}^n v_j u_j \le \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}\right)$, *we have*

$$Z(\mathbf{u}^*,\pi) = \bar{U}^{-k}\left(\sum_{j=1}^{n} (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]}v_{[j]}\right)^{\frac{k}{k+1}}\right)^{k+1}, \qquad (10)$$

where $\omega_j = \delta_1 + \delta_2(n+1-j) + \delta_3(j-1)(n-j+1)$ $(\omega_j = \delta_1 + \delta_2(n-j) + \delta_3 j(n-j))$ *for* $\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ $(\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW})$ *and* Ω_j *is calculated by Equation (3).*

Proof: By substituting Equation (4) into Equation (2), the result can be obtained easily.

Obviously, minimizing $\bar{U}^{-k}\left(\sum_{j=1}^{n} (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]}v_{[j]}\right)^{\frac{k}{k+1}}\right)^{k+1}$ can be obtained by the HLP rule (Hardy, Littlewood and Polya [28]), i.e., matching the smallest $(\Omega_j)^{\frac{1}{k+1}}$ value to the job with the largest $(a_j v_j)^{\frac{k}{k+1}}$ value, the second smallest $(\Omega_j)^{\frac{1}{k+1}}$ value to the job with the second largest $(a_j v_j)^{\frac{k}{k+1}}$ value, and so on.

Based on the above analysis, the problems $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \leq \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ and $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \leq \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}$ can be solved by the following algorithm respectively.

Algorithm 3.1
Step 1. Calculate $\omega_j = \delta_1 + \delta_2(n+1-j) + \delta_3(j-1)(n-j+1)$ ($\omega_j = \delta_1 + \delta_2(n-j) + \delta_3 j(n-j)$) for $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \leq \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ $\left(1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \leq \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}\right)$, $(\Omega_j)^{\frac{1}{k+1}}$ (by using Equation (3)) and $(a_j v_j)^{\frac{k}{k+1}}$ for $j = 1, 2, \ldots, n$.
Step 2. Apply HLP rule to determining the optimal schedule, and denote the resulting optimal sequence by $\pi^* = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n]}\right)$.
Step 3. Calculate the optimal resource allocation $u^*_{[j]}(\pi^*)$ by using Equation (4).

Theorem 3.1. *Algorithm 3.1 solves the problems* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \leq \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ *and* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \leq \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}$ *in* $O(n \log n)$ *time respectively.*

Proof: The correction of Algorithm 3.1 follows from Lemmas 3.1, 3.2 and Equation (10). Steps 1 and 3 of the algorithm can be performed in $O(n)$ time respectively. Step 2 can be performed in $O(n \log n)$ time. Therefore, the overall computation time of Algorithm 3.1 is $O(n \log n)$.

The following example illustrates applying Algorithm 3.1 to finding the optimal solution for the problem $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \leq \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$.

Example 3.1. *Data* $n = 10$, $b = 0.05$, $\alpha = -0.3$, $\delta_1 = \delta_2 = \delta_3 = 2$, $k = 2$, $\bar{U} = 50$, $a_1 = 10$, $a_2 = 13$, $a_3 = 12$, $a_4 = 7$, $a_5 = 3$, $a_6 = 8$, $a_7 = 14$, $a_8 = 2$, $a_9 = 15$, $a_{10} = 9$, $v_1 = 1$, $v_2 = 3$, $v_3 = 2$, $v_4 = 8$, $v_5 = 7$, $v_6 = 6$, $v_7 = 3$, $v_8 = 6$, $v_9 = 4$, $v_{10} = 5$. $\omega_1 = 22$, $\omega_2 = 38$, $\omega_3 = 50$, $\omega_4 = 58$, $\omega_5 = 63$, $\omega_6 = 63$, $\omega_7 = 58$, $\omega_8 = 50$, $\omega_9 = 38$, $\omega_{10} = 22$, $\Omega_1 = 37.2228$, $\Omega_2 = 41.5432$, $\Omega_3 = 43.8393$, $\Omega_4 = 44.0397$, $\Omega_5 = 42.9482$, $\Omega_6 = 39.5082$, $\Omega_7 = 33.9859$, $\Omega_8 = 27.6241$, $\Omega_9 = 19.9419$, $\Omega_{10} = 11.0261$.

From Algorithm 3.1, we have $(a_1v_1)^{\frac{2}{3}} = (10)^{\frac{2}{3}} = 4.6416$, $(a_2v_2)^{\frac{2}{3}} = (39)^{\frac{2}{3}} = 11.5003$, $(a_3v_3)^{\frac{2}{3}} = (24)^{\frac{2}{3}} = 8.3203$, $(a_4v_4)^{\frac{2}{3}} = (56)^{\frac{2}{3}} = 14.6372$, $(a_5v_5)^{\frac{2}{3}} = (21)^{\frac{2}{3}} = 7.6117$, $(a_6v_6)^{\frac{2}{3}} = (48)^{\frac{2}{3}} = 13.2077$, $(a_7v_7)^{\frac{2}{3}} = (42)^{\frac{2}{3}} = 12.0828$, $(a_8v_8)^{\frac{2}{3}} = (12)^{\frac{2}{3}} = 5.2415$, $(a_9v_9)^{\frac{2}{3}} = (62)^{\frac{2}{3}} = 15.3262$, $(a_{10}v_{10})^{\frac{2}{3}} = (45)^{\frac{2}{3}} = 12.6515$, $(\Omega_1)^{\frac{1}{3}} = (37.2228)^{\frac{1}{3}} = 3.3389$, $(\Omega_2)^{\frac{1}{3}} = (41.5432)^{\frac{1}{3}} = 3.4634$, $(\Omega_3)^{\frac{1}{3}} = (43.8393)^{\frac{1}{3}} = 3.5260$, $(\Omega_4)^{\frac{1}{3}} = (44.0397)^{\frac{1}{3}} = 3.5314$, $(\Omega_5)^{\frac{1}{3}} = (42.9482)^{\frac{1}{3}} = 3.5020$, $(\Omega_6)^{\frac{1}{3}} = (39.5092)^{\frac{1}{3}} = 3.4059$, $(\Omega_7)^{\frac{1}{3}} = (33.9859)^{\frac{1}{3}} = 3.2392$, $(\Omega_8)^{\frac{1}{3}} = (27.6241)^{\frac{1}{3}} = 3.0229$, $(\Omega_9)^{\frac{1}{3}} = (19.9419)^{\frac{1}{3}} = 2.7118$, $(\Omega_{10})^{\frac{1}{3}} = (11.0261)^{\frac{1}{3}} = 2.2257$.

From Step 2 of Algorithm 3.1, the optimal schedule is $\pi^* = (J_7, J_3, J_8, J_1, J_5, J_2, J_{10}, J_6, J_4, J_9)$, the optimal resource allocations are

$u_7^* = \frac{(\Omega_j)^{\frac{1}{k+1}} v_{[j]}^{-\frac{1}{k+1}} \left(a_{[j]}\right)^{\frac{k}{k+1}}}{\sum_{j=1}^n (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]} v_{[j]}\right)^{\frac{k}{k+1}}} \times \bar{U} = 2.0716,$
$u_3^* = 2.2196$, $u_8^* = 0.4745$, $u_1^* = 2.5251$, $u_5^* = 0.5866$,
$u_2^* = 2.0113$, $u_{10}^* = 1.2626$, $u_6^* = 1.0251$, $u_4^* = 0.7643$, $u_9^* = 1.3137$,

and the total cost is $\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} = \bar{U}^{-k} \left(\sum_{j=1}^n (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]} v_{[j]}\right)^{\frac{k}{k+1}} \right)^{k+1} =$ 13676.6569.

4. **The Problems Subject to Limited Schedule Criterion.** In this section, we consider the 'inverse version' of the $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \le \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ and $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \le \bar{U}|\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}$, that is, the problems of minimizing the total amount of resource consumed subject to limited schedule criterions, i.e., $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} \le \bar{D}| \sum_{j=1}^n v_j u_j$ and $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} \le \bar{D}| \sum_{j=1}^n v_j u_j$, where $\bar{D}$ is a given real number.

Similar to Section 3, we have the following lemmas.

Lemma 4.1. *For a given sequence of scheduling problems* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha$, $\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} \le \bar{D}| \sum_{j=1}^n v_j u_j$ *and* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha$, $\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} \le \bar{D}| \sum_{j=1}^n v_j u_j$, *the optimal resource allocation can be determined by*

$$u_{[j]}^* = \bar{D}^{-\frac{1}{k}} (\Omega_j)^{\frac{1}{k+1}} \left(v_{[j]}\right)^{-\frac{1}{k+1}} \left(a_{[j]}\right)^{\frac{k}{k+1}} \left(\sum_{j=1}^n (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]} v_{[j]}\right)^{\frac{k}{k+1}} \right)^{\frac{1}{k}}, \quad j = 1, 2, \ldots, n, \tag{11}$$

where $\omega_j = \delta_1 + \delta_2(n+1-j) + \delta_3(j-1)(n-j+1)$ $(\omega_j = \delta_1 + \delta_2(n-j) + \delta_3 j(n-j))$ *for* $\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ $(\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW})$ *and* Ω_j *is calculated by Equation (3).*

Lemma 4.2. *For a given sequence of scheduling problems* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha$, $\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} \le \bar{D}| \sum_{j=1}^n v_j u_j$ *and* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha$, $\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} \le \bar{D}| \sum_{j=1}^n v_j u_j$, *we have*

$$Z(\mathbf{u}^*, \pi) = \bar{D}^{-\frac{1}{k}} \left(\sum_{j=1}^n (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]} v_{[j]}\right)^{\frac{k}{k+1}} \right)^{\frac{1}{k}+1}, \tag{12}$$

where $\omega_j = \delta_1 + \delta_2(n+1-j) + \delta_3(j-1)(n-j+1)$ $(\omega_j = \delta_1 + \delta_2(n-j) + \delta_3 j(n-j))$ *for* $\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$ $(\delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW})$ *and* Ω_j *is calculated by Equation (3).*

Similar to Section 3, the problems $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} \le \bar{D}| \sum_{j=1}^n v_j u_j$ and $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} \le \bar{D}| \sum_{j=1}^n v_j u_j$ can be solved by the following algorithm respectively.

Algorithm 4.1
Step 1. Calculate $\omega_j = \delta_1 + \delta_2(n+1-j) + \delta_3(j-1)(n-j+1)$ ($\omega_j = \delta_1 + \delta_2(n-j) + \delta_3 j(n-j)$) for $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} \le \bar{D}| \sum_{j=1}^n v_j u_j$ $\left(1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} \le \bar{D}| \sum_{j=1}^n v_j u_j\right)$, $(\Omega_j)^{\frac{1}{k+1}}$ (by using Equation (3)) and $(a_j v_j)^{\frac{k}{k+1}}$ for $j = 1, 2, \ldots, n$.
Step 2. Apply HLP rule (Hardy, Littlewood and Polya [28]) to determining the optimal schedule, and denote the resulting optimal sequence by $\pi^* = \left(J_{[1]}, J_{[2]}, \ldots, J_{[n]}\right)$.
Step 3. Calculate the optimal resource allocation $u^*_{[j]}(\pi^*)$ by using Equation (11).

Theorem 4.1. *Algorithm 4.1 solves the problems* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} \le \bar{D}| \sum_{j=1}^n v_j u_j$ *and* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} \le \bar{D}| \sum_{j=1}^n v_j u_j$ *in* $O(n \log n)$ *time respectively.*

The following example illustrates applying Algorithm 4.1 to finding the optimal solution for the problem $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} \le \bar{D}| \sum_{j=1}^n v_j u_j$.

Example 4.1. $\bar{D} = 400$ *and the other input date are the same as Example 3.1. From Algorithm 4.1, the optimal schedule is* $\pi^* = (J_7, J_3, J_8, J_1, J_5, J_2, J_{10}, J_6, J_4, J_9)$, *the optimal resource allocations are* $u^*_7 = \bar{D}^{-\frac{1}{k}} (\Omega_j)^{\frac{1}{k+1}} \left(v_{[j]}\right)^{-\frac{1}{k+1}} \left(a_{[j]}\right)^{\frac{k}{k+1}} \left(\sum_{j=1}^n (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]} v_{[j]}\right)^{\frac{k}{k+1}}\right)^{\frac{1}{k}} = 12.1135$, $u^*_3 = 12.9788$, $u^*_8 = 2.7747$, $u^*_1 = 14.7652$, $u^*_5 = 3.4302$, $u^*_2 = 11.7609$, $u^*_{10} = 7.3829$, $u^*_6 = 5.9942$, $u^*_4 = 4.4694$, $u^*_9 = 7.6820$, *and the total cost is* $\sum_{j=1}^n v_j u_j = \bar{D}^{-\frac{1}{k}} \left(\sum_{j=1}^n (\Omega_j)^{\frac{1}{k+1}} \left(a_{[j]} v_{[j]}\right)^{\frac{k}{k+1}}\right)^{\frac{1}{k}+1} = 292.3681$.

Remark 4.1. *From Li et al. [25], Theorem 3.1 and Theorem 4.1, we have that the optimal schedule of the problems* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha|\theta + \delta_4 \sum_{j=1}^n v_j u_j$, $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha$, $\sum_{j=1}^n v_j u_j \le \bar{U}|\theta$ *and* $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha$, $\theta \le \bar{D}| \sum_{j=1}^n v_j u_j$ *are identical, where* $\theta \in \{\delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}, \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}\}$.

If there are no learning effect and deteriorating jobs $\left(\text{i.e., } p_j = \left(\frac{a_j}{u_j}\right)^k\right)$ or there is no resource allocation (i.e., $p_j = (a_j + bt)\, r^\alpha$), then the problem $1|p_j = \left(\frac{a_j}{u_j}\right)^k$, $\sum_{j=1}^n v_j u_j \le \bar{U}|\theta$ $\left(1|p_j = \left(\frac{a_j}{u_j}\right)^k, \theta \le \bar{D}| \sum_{j=1}^n v_j u_j\right)$, $1|p_j = (a_j + bt)\, r^\alpha|\theta$ and the problem $1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha$, $\sum_{j=1}^n v_j u_j \le \bar{U}|\theta$ $\left(1|p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \theta \le \bar{D}| \sum_{j=1}^n v_j u_j\right)$ have different optimal schedules.

5. **Conclusions.** This article discusses single machine scheduling problems with learning effect, deteriorating jobs and convex resource dependent processing time. For two extension versions, i.e., minimizing a cost function containing makespan, total completion (waiting) time and total absolute differences in completion (waiting) time subject to the constraint that the total compression resource amount is less than or equal to a fixed number, and minimizing the total compression cost subject to the constraint that a cost function containing makespan, total completion (waiting) time and total absolute differences in completion (waiting) time is less than or equal to a fixed number, it shows that these problems can be solved in polynomial time respectively (see Table 1).

Table 1. Summary of main results

Problem	Complexity	Ref.
$1\vert p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha \vert \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} + \delta_4 \sum_{j=1}^n v_j u_j$	$O(n \log n)$	Li et al. [25]
$1\vert p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha \vert \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} + \delta_4 \sum_{j=1}^n v_j u_j$	$O(n \log n)$	Li et al. [25]
$1\vert p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \le \bar{U} \vert \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC}$	$O(n \log n)$	Theorem 3.1
$1\vert p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \sum_{j=1}^n v_j u_j \le \bar{U} \vert \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW}$	$O(n \log n)$	Theorem 3.1
$1\vert p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TC} + \delta_3 \mathrm{TADC} \le \bar{D} \vert \sum_{j=1}^n v_j u_j$	$O(n \log n)$	Theorem 4.1
$1\vert p_j = \left(\left(\frac{a_j}{u_j}\right)^k + bt\right) r^\alpha, \delta_1 C_{\max} + \delta_2 \mathrm{TW} + \delta_3 \mathrm{TADW} \le \bar{D} \vert \sum_{j=1}^n v_j u_j$	$O(n \log n)$	Theorem 4.1

In the future, we can consider jobs' processing time with different factors, such as jobs' position, their starting time, and group technology. It is also interesting to extend single machine scheduling problem with resource allocation constraint to different machine environment, such as parallel machines, flow shop, and job shop. All these problems need further research.

Acknowledgment. This research was supported by the National Natural Science Foundation of China (Grant Nos. 61403260, 71471120 and U1433124). This work was also supported by the Humanistic Social Science Fund of the Education Department of Liaoning Province under Grant W2015316.

REFERENCES

[1] D. Biskup, A state-of-the-art review on scheduling with learning effects, *European Journal of Operational Research*, vol.188, no.2, pp.315-329, 2008.

[2] A. Janiak, T. Krysiak and R. Trela, Scheduling problems with learning and ageing effects: A survey, *Decision Making in Manufacturing and Services*, vol.5, no.1, pp.19-36, 2011.

[3] D. Shabtay and G. Steiner, A survey of scheduling with controllable processing times, *Discrete Applied Mathematics*, vol.155, no.13, pp.1643-1666, 2007.

[4] A. Janiak, W. Janiak and M. Lichtenstein, Resource management in machine scheduling problems: A survey, *Decision Making in Manufacturing and Services*, vol.1, nos.1-2, pp.59-89, 2007.

[5] J.-J. Wang, J.-B. Wang and X.-P. Wang, Scheduling problems with exponential learning functions, *International Journal of Innovative Computing, Information and Control*, vol.6, no.7, pp.3265-3274, 2010.

[6] J.-B. Wang and M.-Z. Wang, Single machine multiple common due dates scheduling with learning effects, *Computers and Mathematics with Applications*, vol.60, no.11, pp.2998-3002, 2010.

[7] C. J. Hsu, W. H. Kuo and D. L. Yang, Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects, *Applied Mathematical Modelling*, vol.35, no.3, pp.1492-1496, 2011.

[8] X. R. Wang, J. B. Wang, J. Jin and P. Ji, Single machine scheduling with truncated job-dependent learning effect, *Optimization Letters*, vol.8, no.2, pp.669-677, 2014.

[9] J. J. Wang and B. H. Zhang, Permutation flowshop problems with bi-criterion makespan and total completion time objective and position-weighted learning effects, *Computers & Operations Research*, vol.58, pp.24-31, 2015.

[10] Y. P. Niu, L. Wan and J. B. Wang, A note on scheduling jobs with extended sum-of-processing-times-based and position-based learning effect, *Asia-Pacific Journal of Operational Research*, vol.32, no.2, pp.1-12, 2015.

[11] J. B. Wang, J. J. Wang and P. Ji, Scheduling jobs with chain precedence constraints and deteriorating jobs, *Journal of the Operational Research Society*, vol.62, no.9, pp.1765-1770, 2011.

[12] T. C. E. Cheng, S. C. Tseng, P. J. Lai and W. C. Lee, Single-machine scheduling with accelerating deterioration effects, *Optimization Letters*, vol.8, no.2, pp.543-554, 2014.

[13] Y. Yin, W. H. Wu, T. C. E. Cheng and C. C. Wu, Single-machine scheduling with time-dependent and position-dependent deteriorating jobs, *International Journal of Computer Integrated Manufacturing*, vol.28, no.7, pp.781-790, 2015.

[14] J. Bai, Z. R. Li and X. Huang, Single-machine group scheduling with general deterioration and learning effects, *Applied Mathematical Modelling*, vol.36, no.3, pp.1267-1274, 2012.

[15] J. B. Wang, L. Liu and C. Wang, Single machine SLK/DIF due window assignment problem with learning effect and deteriorating jobs, *Applied Mathematical Modelling*, vol.37, nos.18-19, pp.8394-8400, 2013.

[16] S. H. Pakzad-Moghaddam, H. Mina and R. Tavakkoli-Moghaddam, An approach for modeling a new single machine scheduling problem with deteriorating and learning effects, *Computers & Industrial Engineering*, vol.78, pp.33-43, 2014.

[17] J. B. Wang and M. Z. Wang, Single-machine scheduling to minimize total convex resource consumption with a constraint on total weighted flow time, *Computers & Operations Research*, vol.39, no.3, pp.492-497, 2012.

[18] D. L. Yang, C. J. Lai and S. J. Yang, Scheduling problems with multiple due windows assignment and controllable processing times on a single machine, *International Journal of Production Economics*, vol.150, pp.96-103, 2014.

[19] Y. Yin, T. C. E. Cheng, C. C. Wu and S. R. Cheng, Single-machine due window assignment and scheduling with a common flow allowance and controllable job processing time, *Journal of the Operational Research Society*, vol.65, no.1, pp.1-13, 2014.

[20] N. Yin and X. Y. Wang, Single-machine scheduling with controllable processing times and learning effect, *International Journal of Advanced Manufacturing Technology*, vol.54, nos.5-8, pp.743-748, 2011.

[21] Z. G. Zhu, L. Y. Sun, F. Chu and M. Liu, Single-machine group scheduling with resource allocation and learning effect, *Computers & Industrial Engineering*, vol.60, no.1, pp.148-157, 2011.

[22] Y. Y. Lu, G. Li, Y. B. Wu and P. Ji, Optimal due-date assignment problem with learning effect and resource-dependent processing times, *Optimization Letters*, vol.8, no.1, pp.113-127, 2014.

[23] G. Li, M. L. Luo, W. J. Zhang and X. Y. Wang, Single-machine due-window assignment scheduling based on common flow allowance, learning effect and resource allocation, *International Journal of Production Research*, vol.53, no.4, pp.1228-1241, 2015.

[24] J. B. Wang, M. Z. Wang and P. Ji, Scheduling jobs with processing times dependent on position, starting time and allotted resource, *Asia-Pacific Journal of Operational Research*, vol.29, no.5, pp.1-15, 2012.

[25] X. J. Li, J. J. Wang and X. R. Wang, Single-machine scheduling with learning effect, deteriorating jobs and convex resource dependent processing times, *Asia-Pacific Journal of Operational Research*, vol.32, no.5, pp.1-12, 2015.

[26] D. Biskup, Single-machine scheduling with learning considerations, *European Journal of Operational Research*, vol.115, no.1, pp.173-178, 1999.

[27] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. R. Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey, *Annals of Discrete Mathematics*, vol.5, pp.287-326, 1979.

[28] G. H. Hardy, J. E. Littlewood and G. Polya, *Inequalities*, Cambridge University Press, Cambridge, 1952.