

Some single-machine scheduling problems with the extended sum-of-processing-time-based aging effect

Radosław Rudek

Received: 21 March 2011 / Accepted: 14 June 2011 / Published online: 10 July 2011
© The Author(s) 2011. This article is published with open access at Springerlink.com

Abstract In this paper, we analyze some single machine scheduling problems with the aging effect. We extend the sum-of-processing-time-based aging model such that the fatigue caused by each job to the machine is equal to a non-increasing function dependent on the normal processing time of a job and the aging effect is job dependent. Although the proposed model is more general and describes more precisely the real-life settings, we show that the special cases of the maximum completion time and the maximum lateness minimization problems with this model are still polynomially solvable. However, we prove the maximum completion time minimization problem with the sum-of-processing-time-based aging model is strongly NP-hard if some jobs have deadlines and constant processing times. On this basis, we show that the maximum lateness minimization problem with this aging model is also strongly NP-hard.

Keywords Scheduling · Deteriorating · Aging effect · Computational complexity · Polynomial algorithm

1 Introduction

Deterioration of machines or tiredness of human workers is present in many manufacturing and industrial systems (see [4, 5, 16, 17, 20]). These phenomena usually increase times required to manufacture prod-

ucts, thereby have an undesirable influence on production rates, total output, and meeting deadlines. To prevent or to minimize such negative effects, some additional resources can be used (e.g., supplementary energy or workers), and also optimization of a production schedule (that takes into account increasing of times required to manufacture products) can improve production objectives. However, additional resources increase expenses; furthermore, it is not always possible to use them, whereas the optimization of a production schedule can at least decrease outlay. Therefore, in the further part, we will focus only on the optimization of production schedules that do not include rest periods of human workers nor maintenance activities of machines, thus are valid between maintenances or for individual shifts.

Nevertheless, to construct an efficient schedule that optimizes the given production objectives, it is required to use effective models that describe the fundamental aspects of the considered manufacturing and industrial systems, i.e., in this case the variability of times required to manufacture products due to deterioration/tiredness. In the scheduling theory, there are two groups of such models. In the first one, processing times of jobs (e.g., tasks, products, or semi-finished products) are non-decreasing functions of their starting times, and the phenomenon is known as “deteriorating effect” (e.g., [2]). However, scheduling models consistent with this approach are not relevant for environments, where deterioration is negligible during idle times of machines. Such inconveniences are absent in a second approach, called “aging effect,” in which job processing times are described by non-decreasing functions dependent on the actual condition (fatigue) of machines affected by already processed jobs. Therefore, this group of models

R. Rudek (✉)
Wrocław University of Economics, Komandorska 118/120,
53-345 Wrocław, Poland
e-mail: radoslaw.rudek@ue.wroc.pl

has attracted growing attention in research community (see [1, 3, 10, 12–14, 23, 24, 26]).

Furthermore, there are two main approaches to describe the fatigue. The first, called “position dependent/ based aging model”, assumes that job processing times are non-decreasing functions dependent on the number of processed jobs, i.e., a job position in a sequence (see [7, 11, 15, 18]). It models problems, where jobs have similar impact on the fatigue of a machine. However, in many cases, the fatigue can also depend on the time required to perform a job, i.e., more time-consuming jobs can have greater impact on the fatigue of the machine. Such settings are covered by “sum-of-processing-time-based aging model” that assumes the job processing time is a non-decreasing function dependent on the sum of the normal processing times of already processed jobs (see [8, 21, 22]). The normal processing time is defined (measured) as the time required to perform a job if a machine is not affected by aging.

Increase in job processing times is the result of a fatigue of a machine (e.g., tool wear), whereas the fatigue caused by each job to the machine depends on the time required to process it (see [20]). Different jobs can have various impact on the fatigue of a machine, but usually a job causes more fatigue if more time is required to perform it. In general, the relation between the time required to process a job and the fatigue this job causes to the machine can be described by a non-decreasing function (see [19, 20]). Nevertheless, it was not considered in the scheduling models. Therefore, in this paper, we extend the existing sum-of-processing-time-based aging models according to this approach.

Furthermore, in real-life problems, the fatigue of a machine can have different impact on processing times of jobs (see [25] and [26]), and some jobs can increase the fatigue of a machine, but the fatigue does not affect processing times of such jobs, e.g., tiredness of a human worker does not increase the time required to move him from one location to another; however, the dislocation process can make the worker tired. To fill this gap, we will further extend the considered aging model to be job dependent as it was done for position-based models (e.g., [24, 26]). Therefore, we present a new model of scheduling problems in the deteriorating/aging environments that describes real-life problems more precisely than sum-of-processing-time-based aging models. Following this approach, we will provide algorithms that allow the practitioners to construct an optimal schedule that minimizes the maximum completion time and some special cases of the maximum lateness criterion without having the knowledge about the exact shape of the function describing the relation between the time required to process a job and the fatigue it causes to

the machine nor the values of the aging characteristic of the machine. However, we will also prove that the maximum completion time minimization problem with the sum-of-processing-time-based aging model is strongly NP-hard if jobs can have deadlines. It means that the objective is to minimize the maximum completion time under constraints that jobs must not exceed their deadlines. On this basis, we will show that the maximum lateness minimization problem with this aging model is strongly NP-hard. Thus, fast algorithms that solve optimally these problems under aging of a machine do not exist. Therefore, to solve the considered NP-hard problems efficient heuristic or metaheuristic algorithms have to be applied, which require the additional analysis that determines the values of job parameters describing aging.

The remainder of this paper is organized as follows. A new model description and the problem formulation are presented in the next section, whereas polynomially solvable cases and the computational complexity status are given subsequently. The last section concludes the paper.

2 Problem formulation

There is given a single machine (e.g., a single worker or a group of human workers, CNC machine, tool, chemical cleaning bath) and a set $J = \{1, \dots, n\}$ of n jobs that have to be processed by the machine; there are no precedence constraints between jobs. The machine is continuously available and can process at most one job at a time. Once it begins processing a job, it will continue until this job is finished. Moreover, each job can have deadline \bar{d}_j or due date d_j , when it must or should be completed, respectively.

Due to the aging effect, the time required to complete a job increases with the fatigue of the machine. Wang et al. [22] proposed a model, where the processing time of job j that is scheduled in the v th position in a sequence is given by:

$$\tilde{p}_j(v) = p_j \cdot \left(1 + \sum_{l=1}^{v-1} p_{[l]} \right)^\alpha, \quad (1)$$

where p_j is the normal processing time that is the time required to perform job j if the machine is not influenced by aging (i.e., $p_j \triangleq \tilde{p}_j(1)$), $\sum_{l=1}^{v-1} p_{[l]}$ is the fatigue of the machine at the start of job j , where $p_{[l]}$ denotes the processing time of a job scheduled in the l th position in a sequence and $\alpha \geq 0$ is the aging index (i.e., an aging characteristic) that describes a relation between job processing times and the fatigue of the machine.

However, the fatigue caused by a job is seldom equal to the time required to process this job (see [19, 20]); thus, we extend model Eq. 1 such that the fatigue caused by job j is equal to $f(p_j)$, where f is a positive increasing function ($\frac{df}{dx}(x) > 0$ for $x > 0$ and $f(0) = 0$) that describes the relation between the time required to process a job and the fatigue this job causes to the machine (e.g., [20]); function f is specific for the machine. On this basis and taking into consideration that jobs can have distinct aging characteristics (see [24]), we propose a new more general model of job processing times:

$$\tilde{p}_j(v) = p_j \cdot \left(1 + \sum_{l=1}^{v-1} f(p_{[l]}) \right)^{\alpha_j}, \tag{2}$$

where $\alpha_j \geq 0$ is the aging index of job j and it models that the same level of the fatigue of the machine can have various impact on the processing times of different jobs.

An example of jobs whose processing times are described by Eq. 2 is as follows: Consider jobs that are processed on a CNC lathe machine. Their processing times are not affected by the fatigue of the worker ($\alpha_j = 0$); however, the monitoring process increases the worker’s fatigue as well as his relocation ($\alpha_j > 0$). Furthermore, the fatigue related with the relocation grows with the distance and the dependency distance-fatigue does not have to be linear, i.e., it can be described by the function f . Similarly, the more time-consuming jobs processed on a CNC lathe machine can cause more fatigue to the machine than the less time consuming jobs. Such dependency can be non-linear due to the growing temperature of a drill. Such dependency can also be modeled by f . Moreover, the fatigue of the machine can increase the cutting time of hard materials more significantly than in case of soft materials, i.e., α_j is greater for harder materials.

Let $\pi = \langle \pi(1), \dots, \pi(i), \dots, \pi(n) \rangle$ denote the sequence of jobs (permutation of the elements of the set J), where $\pi(i)$ is the job processed in position i in this sequence. By Π we will denote the set of all such permutations. For the given sequence (permutation) $\pi \in \Pi$, we can easily determine the completion time $C_{\pi(i)}$ of a job placed in the i th position in π from the following formulae:

$$C_{\pi(i)} = C_{\pi(i-1)} + \tilde{p}_{\pi(i)}(i) = \sum_{l=1}^i \tilde{p}_{\pi(l)}(l), \tag{3}$$

where $C_{\pi(0)} = 0$ and according to Eq. 2, we have $\tilde{p}_{\pi(i)}(i) = p_{\pi(i)} \cdot \left(1 + \sum_{l=1}^{i-1} f(p_{\pi(l)}) \right)^{\alpha_{\pi(i)}}$. The objective is to find such a schedule (sequence) π of jobs on the single machine, which minimizes the maximum completion

time (makespan) criterion $C_{\max}(\pi) = \max_{i=1, \dots, n} \{C_{\pi(i)}\}$. Since in some of the considered problems each job j can have deadline \bar{d}_j , then for such problems the feasible schedules have to hold $C_{\pi(i)} \leq \bar{d}_{\pi(i)}$ for $i = 1, \dots, n$.

Formally, the optimal schedule $\pi^* \in \Pi$ for the makespan minimization problem is defined as follows: $\pi^* \triangleq \operatorname{argmin}_{\pi \in \Pi} \{C_{\pi(n)}\}$, where Π is a set of all possible schedules (permutations), but if jobs have deadlines, then Π is the set of all feasible schedules such that $\Pi = \{\pi : C_{\pi(i)} \leq \bar{d}_{\pi(i)} \text{ for } i = 1, \dots, n\}$. Additionally, we also consider the maximum lateness minimization criterion that is defined as $L_{\max}(\pi) = \max_{i=1, \dots, n} \{L_{\pi(i)}\}$, where $L_{\pi(i)} = C_{\pi(i)} - d_{\pi(i)}$ and $\pi^* \triangleq \operatorname{argmin}_{\pi \in \Pi} \{ \max_{i=1, \dots, n} \{L_{\pi(i)}\} \}$.

For convenience and to keep an elegant description of the considered problem, we will use the standard three field notation scheme $X | Y | Z$ (see [9]), where X describes the machine environment, Y describes job characteristics and constraints, and Z represents the minimization objectives. According to this notation, the problems analyzed in the paper will be denoted as $1 | \tilde{p}_j(v) = p_j(1 + \sum_{l=1}^{v-1} f(p_{[l]})^{\alpha_j} | C_{\max}$, $1 | \tilde{p}_j(v) = p_j(1 + \sum_{l=1}^{v-1} f(p_{[l]})^{\alpha_j}, \alpha_j \in \{0, \alpha\} | C_{\max}$, $1 | \bar{d}_j, \tilde{p}_j(v) = p_j(1 + \sum_{l=1}^{v-1} \beta p_{[l]})^{\alpha_j}, \alpha_j \in \{0, 1\} | C_{\max}$, $1 | \tilde{p}_j(v) = p_j(1 + \sum_{l=1}^{v-1} \beta p_{[l]}) | L_{\max}$, $1 | \tilde{p}_j(v) = p_j(1 + \sum_{l=1}^{v-1} f(p_{[l]})^{\alpha_j}, \alpha_j \in \{0, \alpha\}, \operatorname{agr} | L_{\max}$ and $1 | \tilde{p}_j(v) = p_j(1 + \sum_{l=1}^{v-1} \beta p_{[l]})^{\alpha_j}, \alpha_j \in \{0, 1\} | L_{\max}$, where agr denotes agreeable job parameters such that $\alpha_j = \alpha > 0$ and $\alpha_k = 0$ or $p_j \leq p_k$ if $\alpha_j = \alpha_k > 1$ or $p_j \geq p_k$ if $\alpha_j = \alpha_k < 1$ implies $d_j \leq d_k$ for all $j, k \in J$.

Throughout the paper, we assume that $\tilde{p}_j(v)$ can be calculated in time bounded by a polynomial dependent on a problem size.

3 Computational complexity and algorithms

In this section, we will provide optimal polynomial time algorithms that minimize the maximum completion time with the special, however practical, cases of the considered aging model. Moreover, we will also prove that if jobs have deadlines (or due dates), then the problem to minimize the maximum completion time (or the maximum lateness) even with the sum-of-processing-time-based aging model is strongly NP-hard.

3.1 Polynomially solvable cases

The analysis of the special cases presented in this section not only determines the boundary between polynomially solvable and NP-hard instances but first and foremost provides solution algorithms that can be used by practitioners.

Due to the aging effect, the job processing times increase; however, the practitioners face the problem of determining values of job parameters such as the exact shape of f or the aging characteristics α_j . Therefore, the effective application of scheduling models to efficiently solve real-life problems requires additional analysis that is difficult. However, we will prove that to construct an optimal production schedule for the problems discussed in the further part of this section, it is required only to know: whether the time required to process a job increases with the fatigue of the machine (i.e., for which jobs $\alpha_j = 0$), an approximate shape of f (convex/concave) and relations (lower/equal/greater) between values of the normal processing times of jobs. Thus, the construction of an optimal schedule that minimizes the maximum completion time (makespan) or special cases of the maximum lateness does not require additional analysis to precisely determine the parameters of jobs.

Theorem 1 *The problem*

$$1|\tilde{p}_j(v) = p_j \left(1 + \sum_{l=1}^{v-1} f(p_{[l]}) \right)^\alpha |C_{\max}$$

is optimally solvable by scheduling jobs according to

- (a) an arbitrary order if $\alpha = 0$ or $\alpha = 1$ and $f(x) = \beta x$ (where $\beta > 0$),
- (b) the non-increasing order of p_j (LPT rule) if $0 < \alpha < 1$ and $\frac{d^2 f}{dx^2}(x) \leq 0$,
- (c) the non-decreasing order of p_j (SPT rule) if $\alpha > 1$ and $\frac{d^2 f}{dx^2}(x) \geq 0$.

Proof Assume there is given an optimal permutation π , which does not hold the rule from the thesis of this theorem. Therefore, for this permutation there exists a pair of jobs $\pi(i)$ and $\pi(i + 1)$ such that for each corresponding case

- (a) Relation between $p_{\pi(i)}$ and $p_{\pi(i+1)}$ is immaterial.
- (b) $p_{\pi(i)} \leq p_{\pi(i+1)}$.
- (c) $p_{\pi(i)} \geq p_{\pi(i+1)}$.

Obviously, the schedule of jobs is immaterial if $\alpha = 0$. For other cases, assume that there is given a permutation π' , which has been obtained from the permutation π by interchanging the jobs from the i th and the $(i + 1)$ th position. Based on Eq. 3, the difference Δ between the criterion values for π and π' is given as follows:

$$\begin{aligned} \Delta &= C_{\max}(\pi) - C_{\max}(\pi') \\ &= p_{\pi(i)} p_{\pi(i+1)} (g(p_{\pi(i)}) - g(p_{\pi(i+1)})). \end{aligned}$$

where $g(x) = \left[(A + f(x))^\alpha - A^\alpha \right] / x$ and $x > 0$ and $A = 1 + \sum_{l=1}^{i-1} f(p_{\pi(l)})$. To determine the value of subtraction, we have to analyze monotonicity of $g(x)$. Observe that the derivative of $g(x)$ is given as follows:

$$\begin{aligned} \frac{dg}{dx}(x) &= \left[\alpha x f'(x) (A + f(x))^{\alpha-1} - (A + f(x))^\alpha + A^\alpha \right] / x^2 \\ &= h(x) / x^2, \end{aligned}$$

where $h(x) = \alpha x f'(x) (A + f(x))^{\alpha-1} - (A + f(x))^\alpha + A^\alpha$ and $f'(x) = \frac{df}{dx}(x) > 0$ for $x > 0$. Thus, $g(x)$ is non-decreasing if $h(x) \geq 0$ and non-increasing if $h(x) \leq 0$. Since $h(0) = 0$, then to determine the sign of $h(x)$ for $x > 0$ it is enough to analyze its monotonicity. The derivative of $h(x)$ is equal to:

$$\begin{aligned} \frac{dh}{dx}(x) &= \alpha x (A + f(x))^{\alpha-2} \\ &\times \left[f''(x) (A + f(x)) + (\alpha - 1) (f'(x))^2 \right], \end{aligned}$$

where $f''(x) = \frac{d^2 f}{dx^2}(x)$. On this basis, for the considered cases we have

- (a) Since $f(x) = \beta x$ and $\alpha = 1$, then $h(x) = 0$ thereby $g(x)$ is constant; thus, $\Delta = 0$ independently on the schedule.
- (b) Since $\frac{d^2 f}{dx^2}(x) \leq 0$ and $0 < \alpha < 1$, then $\frac{dh}{dx}(x) < 0$ and $g(x)$ is decreasing; thus, $\Delta \geq 0$ if $p_{\pi(i)} \leq p_{\pi(i+1)}$.
- (c) Since $\frac{d^2 f}{dx^2}(x) \geq 0$ and $\alpha > 1$, then $\frac{dh}{dx}(x) > 0$ and $g(x)$ is increasing; thus, $\Delta \geq 0$ if $p_{\pi(i)} \geq p_{\pi(i+1)}$.

Note that $\Delta = C_{\max}(\pi) - C_{\max}(\pi')$ is non-negative for all cases. Thus, the permutation π cannot be optimal and the optimal solution can be constructed according to this theorem. \square

Obviously, the longer a job is processed, the more fatigue it causes to the machine and in many cases this relation can be described (see [19]) or approximated (see [20]) by a proportional relation, i.e., $f(x) = \beta x$; in real-life cases, β is usually a small number (i.e., $\beta \in (0, 1)$); otherwise, job processing times increase quickly to very high values that describe an instant machine wear. On the other hand, the impact of the fatigue on the job processing times does not have to be linear, which can be modeled by α . On this basis and following Theorem 1, we provide some useful hints for practitioners.

Corollary 1 *The optimal schedule of jobs for the problem*

$$1|\tilde{p}_j(v) = p_j \left(1 + \sum_{l=1}^{v-1} \beta p_{[l]} \right)^\alpha |C_{\max}$$

is obtained according to

- (a) an arbitrary order if there is no fatigue ($\alpha = 0$) or the aging characteristic is linear ($\alpha = 1$),
- (b) the non-increasing order of p_j if the aging characteristic is concave ($0 < \alpha < 1$),
- (c) the non-decreasing order of p_j if the aging characteristic is convex ($\alpha > 1$).

Furthermore, we will show that the problem is still polynomially solvable even if there are jobs that increase the fatigue of a machine, but the fatigue does not affect their processing times ($\alpha_j = 0$ for such jobs). An example is a job that is based mostly on moving a human worker from one location to another; thus, his tiredness does not increase the time required to move; however, the dislocation process can make the worker tired.

Theorem 2 *The problem*

$$1|\tilde{p}_j(v) = p_j \left(1 + \sum_{l=1}^{v-1} f(p_{[l]}) \right)^{\alpha_j}, \alpha_j \in \{0, \alpha\} |C_{\max}$$

is polynomially solvable according to the following rule: First, the jobs with $\alpha_j = \alpha$ are scheduled according to Theorem 1 and after them the jobs with $\alpha_j = 0$ are scheduled in an arbitrary order.

Proof Obviously, the maximum completion time is minimized if jobs with constant processing times (i.e., $\alpha_j = 0$) are scheduled after jobs with variable processing times (i.e., $\alpha_j = \alpha$); otherwise, the criterion value increases. The maximum completion time of jobs with $\alpha_j = \alpha$ is minimized according to Theorem 1. \square

Now we will analyze the maximum lateness minimization problem with the special cases of the considered aging model.

Theorem 3 *The problem*

$$1|\tilde{p}_j(v) = p_j \left(1 + \sum_{l=1}^{v-1} \beta p_{[l]} \right) |L_{\max}$$

is polynomially solvable by scheduling jobs according to the non-decreasing order of their due dates d_j .

Proof Assume there is given an optimal permutation π , which does not hold the rule from the thesis of this theorem. Therefore, for this permutation, there exists a pair of jobs $\pi(i)$ and $\pi(i + 1)$ such that $d_{\pi(i)} \geq d_{\pi(i+1)}$. Consider a permutation π' , which has been obtained from the permutation π by interchanging the jobs from the i th and the $(i + 1)$ th position, i.e., $\pi'(i) = \pi(i + 1)$ and $\pi'(i + 1) = \pi(i)$. Note that $\pi'(l) = \pi(l)$ for $l = 1, \dots, i - 1 \cup i + 2, \dots, n$; therefore, $L_{\pi'(l)} = L_{\pi(l)}$ for $l = 1, \dots, i - 1 \cup i + 2, \dots, n$. On the other hand, the lateness of jobs in positions i and $i + 1$ for both permutations π and π' are given as follows:

$$\begin{aligned} L_{\pi(i)} &= C_{\pi(i-1)} + p_{\pi(i)}(1 + A) - d_{\pi(i)}, \\ L_{\pi(i+1)} &= C_{\pi(i-1)} + p_{\pi(i)}(1 + A) \\ &\quad + p_{\pi(i+1)}(1 + A + \beta p_{\pi(i)}) - d_{\pi(i+1)}, \\ L_{\pi'(i)} &= C_{\pi(i-1)} + p_{\pi(i+1)}(1 + A) - d_{\pi(i+1)}, \\ L_{\pi'(i+1)} &= C_{\pi(i-1)} + p_{\pi(i+1)}(1 + A) \\ &\quad + p_{\pi(i)}(1 + A + \beta p_{\pi(i+1)}) - d_{\pi(i)}, \end{aligned}$$

where $A = \sum_{l=1}^{i-1} p_{\pi(l)}$. Since $d_{\pi(i)} \geq d_{\pi(i+1)}$, then $L_{\pi(i+1)} > L_{\pi(i)}$ and $L_{\pi(i+1)} > L_{\pi'(i)}$ and $L_{\pi(i+1)} > L_{\pi'(i+1)}$. Therefore, $\max_{i=1, \dots, n} \{L_{\pi(i)}\} \geq \max_{i=1, \dots, n} \{L_{\pi'(i)}\}$ thereby π cannot be optimal and the optimal solution is constructed according to the non-decreasing order of d_j . \square

Following Theorem 2 and using the job interchanging technique, it can be easily proved that the maximum lateness minimization problem under agreeable job parameters is still polynomially solvable; therefore, the proof is omitted.

Theorem 4 *The problem*

$$1|\tilde{p}_j(v) = p_j \left(1 + \sum_{l=1}^{v-1} f(p_{[l]}) \right)^{\alpha_j}, \alpha_j \in \{0, \alpha\}, agr |L_{\max}$$

with agreeable job parameters is polynomially solvable according to the following rule: First the jobs with $\alpha_j = \alpha$ are scheduled according to Theorem 1 and after them the jobs with $\alpha_j = 0$ are scheduled according to the non-decreasing order of their due dates d_j .

Note that it is easy to determine whether aging affects the processing times of jobs; therefore, the procedures described in the presented theorems can be easily used by practitioners. Further, we will show that the makespan minimization problem with the sum-of-processing-time-based aging model becomes strongly NP-hard if jobs can have distinct deadlines even if the

aging characteristics are linear, thereby the maximum lateness minimization is not less complex.

3.2 Strong NP-hardness

Although the problem $1|\tilde{p}_j(v)=p_j(1+\sum_{l=1}^{v-1}\beta p_{[l]})^{\alpha_j}$, $\alpha_j \in \{0, 1\}|C_{\max}$ is polynomially solvable in $O(n)$ (see Corollary 1), it becomes strongly NP-hard if some jobs have deadlines; note that the problem with deadlines but with constant job processing times ($\alpha_j = 0$ for $j = 1, \dots, n$) or with linear functions of job processing times ($\alpha_j = 1$ for $j = 1, \dots, n$) is polynomially solvable according to the non-decreasing order of deadlines that follows from Theorem 3.

Observe that if $\beta = 1$, then the processing times of jobs with $\alpha_j = 1$ model marginally the real-life aging effect, since they increase very quickly to high values; however, from the mathematical point of view, such model is simpler. On the other hand, the small values of β have strong practical justification. Therefore, we will prove that the considered problem is strongly NP-hard with arbitrary positive β (in particular $\beta = 1$) such that β and β^{-1} can be bounded by a polynomial dependent on the size of the considered problem and on its maximal value.

Theorem 5 The problem

$$1|\bar{d}_j, \tilde{p}_j(v)=p_j\left(1+\sum_{l=1}^{v-1}\beta p_{[l]}\right)^{\alpha_j}, \alpha_j \in \{0, 1\}|C_{\max}$$

is strongly NP-hard for arbitrary $\beta > 0$.

Proof We will show that the strongly NP-complete problem 3-PARTITION [6] can be transformed in a pseudopolynomial time to the decision versions of the considered scheduling problem.

3-PARTITION (3PP) [6] There are given positive integers m, B , and x_1, \dots, x_{3m} of $3m$ positive integers satisfying $\sum_{q=1}^{3m} x_q = mB$ and $\frac{B}{4} < x_q < \frac{B}{2}$ for $q = 1, \dots, 3m$. Does there exist a partition of the set $X = \{1, \dots, m\}$ into m disjoint subsets X_1, \dots, X_m such that $\sum_{q \in X_i} x_q = B$ for $i = 1, \dots, m$?

The decision version of the considered scheduling problem is given as follows (decision scheduling problem, DSP): Does there exist such a schedule π of jobs on the machine for which there are no late jobs and the criterion value C_{\max} is equal or lower than the given value y ?

The pseudopolynomial time transformation from 3PP to DSP is constructed as follows. There are given $n = 4m$ jobs and among them there are $3m$ partition jobs:

$$p_j = H + x_j, \quad \alpha_j = 1,$$

$$\bar{d}_j = D = p_e m + \sum_{l=1}^m V_l + \beta B^2 m,$$

for $j = 1, \dots, 3m$ and m enforcer jobs e_i ($i = 1 \dots, m$) with the following parameters:

$$p_{e_i} = p_e = m^2 B H, \quad \alpha_{e_i} = 0,$$

$$\bar{d}_{e_i} = p_e i + \sum_{l=1}^{i-1} V_l + \beta B^2 (i - 1),$$

and $y = D$, where

$$H = 2mB,$$

$$V_i = (3H + B)$$

$$+ \beta \left[(3H + B) \left(p_e i + H + (3H + B)(i - 1) \right) + HB \right]$$

for $i = 1, \dots, m$ and $\sum_{l=1}^0 V_l = 0$ and $\beta = 1/a$, where a is an arbitrary constant value that can be bounded by a polynomial dependent on the problem size, $O(m)$, and the maximal value, B , of 3PP. For this proof and practical cases, we assume that $a \geq 1$ (i.e., $0 < \beta \leq 1$), but the proof is also valid if $a < 1$ (i.e., $\beta > 1$); however, it models marginal real-life settings.

Note that computational complexity theory considers integer numbers, and the parameters of DSP are expressed by integer (also β that is expressed by a and its value does not have to be calculated). Moreover, all calculations can be done on integer numbers, i.e., to calculate y, \bar{d}_{e_i} , or \bar{d}_j , it is required to reduced the elements to the lowest common denominator, but it is not greater than a . Thus, the given transformation from 3PP to DSP is pseudopolynomial, since the maximum value of DSP does not increase exponentially in reference to 3PP (i.e., y is $O(am^6 B^3)$) and the problem size does not decrease exponentially in reference to 3PP (i.e., $n = 4m$); furthermore, all calculations can be done on integer numbers in pseudopolynomial time.

Now we will provide the property of an optimal schedule for DSP. For convenience let us renumber partition jobs $J' = \{1, \dots, j, \dots, 3m\}$ such that for the considered schedules, the jobs are in the natural order. At first observe that there exists an optimal schedule, where the enforcer jobs are scheduled in the

non-decreasing order of their deadlines \bar{d}_{e_i} . It results from the optimal feasible schedule for the classical problem $1|d_j|C_{\max}$ with constant job processing times, since for DSP we have $\bar{p}_{e_i}(v) = p_e$ for $i = 1, \dots, m$ and $v = 1, \dots, 4m$. Moreover, e_1 is the first processed job; otherwise, $C_{e_1} > \bar{d}_{e_1}$. Therefore, the optimal schedules have the following form $(e_1, J'_1, \dots, e_i, J'_i, \dots, e_m, J'_m)$, where J'_i are disjoint subsets of partition jobs $J' = J'_1 \cup \dots \cup J'_i \cup \dots \cup J'_m$. In the further part, we will show that $|J'_i| = 3$ for $i = 1, \dots, m$, where $|J'_i|$ denotes the cardinality of J'_i .

Let us define useful expressions for a schedule π^* , where each $J'_i = J_i$ and J_i consists of exactly three partition jobs corresponding to the elements from $X_i = \{3i - 2, 3i - 1, 3i\}$. Let $V(J_i)$ denote the sum of processing times of jobs from J_i for the schedule π^* :

$$\begin{aligned}
 V(J_i) &= \left(\sum_{j \in J_i} p_j \right) \left(1 + \sum_{l=1}^i \beta p_{e_l} + \sum_{l=1}^{i-1} \sum_{j \in J_l} \beta p_j \right) \\
 &\quad + \beta \sum_{j,k \in J_i \wedge j < k} p_j p_k \\
 &= \left(\sum_{j \in J_i} p_j \right) + \beta \left[\left(\sum_{j \in J_i} p_j \right) \left(\sum_{l=1}^i p_{e_l} + \sum_{l=1}^{i-1} \sum_{j \in J_l} p_j \right) \right. \\
 &\quad \left. + \sum_{j,k \in J_i \wedge j < k} p_j p_k \right]. \tag{4}
 \end{aligned}$$

Since jobs from J_i are constructed on the basis of the elements from X_i (for $i = 1, \dots, m$), then Eq. 4 can be expressed as follows:

$$\begin{aligned}
 V(X_i) &= \left(3H + \sum_{j \in X_i} x_j \right) \\
 &\quad + \beta \left[\left(3H + \sum_{j \in X_i} x_j \right) \left(p_e i + \sum_{l=1}^{i-1} \left(3H + \sum_{j \in X_l} x_j \right) \right) \right. \\
 &\quad \left. + \sum_{j,k \in X_i \wedge j < k} (H + x_j)(H + x_k) \right] \\
 &= \left(3H + \sum_{j \in X_i} x_j \right) + \beta \left[\left(3H + \sum_{j \in X_i} x_j \right) \right. \\
 &\quad \times \left(p_e i + H + \sum_{l=1}^{i-1} \left(3H + \sum_{j \in X_l} x_j \right) \right) \\
 &\quad \left. + H \sum_{j \in X_i} x_j + W(X_i) \right], \tag{5}
 \end{aligned}$$

where $W(X_i) = \sum_{j,k \in X_i \wedge j < k} x_j x_k$ for $i = 1, \dots, m$. Knowing that $\frac{B}{4} < x_j < \frac{B}{2}$ for $j = 1, \dots, m$, and

$$\frac{3}{16} B^2 < W(X_i) < \frac{3}{4} B^2 \tag{6}$$

for $i = 1, \dots, m$, we can estimate $V(X_i)$ as follows:

$$\begin{aligned}
 V(X_i) &> \left(3H + \frac{3}{4} B \right) \\
 &\quad + \beta \left[\left(3H + \frac{3}{4} B \right) \left(p_e i + H + \sum_{l=1}^{i-1} \left(3H + \frac{3}{4} B \right) \right) \right. \\
 &\quad \left. + \frac{3}{4} H B + \frac{3}{16} B^2 \right] > (3H + B) - \frac{1}{4} B \\
 &\quad + \beta \left[(3H + B)(p_e i + H + (3H + B)(i-1)) + H B \right. \\
 &\quad \left. - \frac{B}{4} \left(p_e i + 2H + \left(6H + \frac{7}{4} B \right) (i-1) \right) \right] \\
 &> V_i - \frac{B}{4} - \beta \left[\frac{B}{4} (1 + p_e i + 2H + 2(3H + B)(i-1)) \right] \\
 &> V_i - B - \beta B p_e i, \tag{7}
 \end{aligned}$$

since $2H + 2(3H + B)(i - 1) < 2(3H + B)i < p_e i$.

Now we will show that there exists an optimal solution π , where $|J'_i| = 3$ for $i = 1, \dots, m$. At first note that the maximum completion time C_{\max} is minimal if jobs with $\alpha_j = 0$ are scheduled after jobs with $\alpha_j = 1$ (Corollary 1). Therefore, in the optimal solution of DSP, the enforcer jobs ($\alpha_{e_i} = 0$) are scheduled after as many partition jobs ($\alpha_j = 1$) as it is possible. Let us consider job e_2 and assume that before this job at least four partition jobs are scheduled, i.e., $|J'_1| > 3$ and $J'_1 = \{J_1, Q_1\}$, where $Q_1 \subseteq J' \setminus \{J_1\}$. Thus, its completion time can be estimated as follows:

$$C_{e_2} > p_{e_1} + V(J_1) + p_q \left(1 + \beta p_{e_1} + \sum_{j \in J_1} \beta p_j \right) + p_{e_2},$$

where $q \in Q_1$. Knowing that $p_j > H$ (for $j \in J'$) and based on Eq. 7, we have

$$\begin{aligned}
 C_{e_2} &> 2p_e + V(X_1) + H(1 + \beta p_e + 3H\beta) \\
 &> 2p_e + V_1 - B - \beta B p_e + H + \beta [H p_e + 3H^2] \\
 &> \bar{d}_{e_2} - \beta B^2 - B - \beta B p_e + H + \beta [H p_e + 3H^2] \\
 &= \bar{d}_{e_2} + H - B + \beta [[H - B] p_e + 3H^2 - B^2] > \bar{d}_{e_2},
 \end{aligned}$$

since $H > B$. Therefore, before e_2 , no more than three partition jobs can be scheduled. On the other hand, the criterion value is minimized if enforcer jobs are scheduled after as many partition jobs as it is possible; thus, we conclude that in the optimal solution $J'_1 = J_1$, thereby $|J'_1| = 3$.

In order to proceed inductively, let us consider job e_i ($i = 2, \dots, m$) such that $J'_l = J_l$ and $|J'_l| = 3$ for $l = 1, \dots, i - 2$ (J'_0 denotes an artificial empty set) and $|J'_{i-1}| > 3$, where $J'_{i-1} = \{J_{i-1}, Q_{i-1}\}$ and $Q_{i-1} \subseteq J' \setminus \{J_1 \cup \dots \cup J_{i-1}\}$. The total completion time of e_i can be estimated as follows:

$$\begin{aligned}
 C_{e_i} &> \sum_{l=1}^i p_{e_l} + \sum_{l=1}^{i-1} V(J_l) + p_q \left(1 + \beta p_e(i-1) + \sum_{l=1}^{i-1} \sum_{j \in J_l} \beta p_j \right) \\
 &> p_e i + \sum_{l=1}^{i-1} (V_l - B - \beta B p_e l) \\
 &\quad + H(1 + \beta p_e(i-1) + 3H(i-1)\beta) \\
 &= \bar{d}_{e_i} - \beta B^2(i-1) - B(i-1) - \beta B p_e \sum_{l=1}^{i-1} l \\
 &\quad + H(1 + \beta p_e(i-1) + 3H(i-1)\beta) \\
 &> \bar{d}_{e_i} + H - B(i-1) \\
 &\quad + \beta [[H - B(i-1)] p_e + 3H^2 - B^2] (i-1) > \bar{d}_{e_i},
 \end{aligned}$$

where $q \in Q_{i-1}$ and $H > B(i-1)$ for $i = 1, \dots, m$. Thus, if J'_i consists of more than three partition jobs, then at least one enforcer job is late. On this basis, we conclude that in the optimal solution $|J'_i| = 3$ for $i = 1, \dots, m$.

Therefore, in the further part, we consider only schedules consistent with π^* , i.e., $(e_1, J_1, e_2, J_2, \dots, e_i, J_i, \dots, e_m, J_m)$, where the sequence of jobs within each set J_i is arbitrary, since it does not affect the objective function.

From the definition of 3PP follows that $\sum_{j \in X_i} x_j = B + \lambda_i$, where $\lambda_i \in (-\frac{B}{4}, \frac{B}{2})$ for $i = 1, \dots, m$; therefore, based on Eq. 5, we obtain

$$\begin{aligned}
 V(X_i) &= (3H + B + \lambda_i) \\
 &+ \beta \left[(3H + B + \lambda_i) \left(p_e i + H + \sum_{l=1}^{i-1} (3H + B + \lambda_l) \right) \right. \\
 &\quad \left. + H(B + \lambda_i) + W(X_i) \right] = V_i + \lambda_i + \beta \\
 &\times \left[\lambda_i \left(p_e i + 2H + (3H + B)(i-1) + \sum_{l=1}^{i-1} \lambda_l \right) \right. \\
 &\quad \left. + (3H + B) \sum_{l=1}^{i-1} \lambda_l + W(X_i) \right]. \tag{8}
 \end{aligned}$$

Now let us define the completion time of the enforcer job e_k and of the last job in X_k as follows ($k = 1, \dots, m$):

$$C_{e_k} = p_e k + \sum_{i=1}^{k-1} V(X_i), \tag{9}$$

$$C_{X_k} = p_e k + \sum_{i=1}^k V(X_i). \tag{10}$$

Following Eq. 8, the completion times of enforcer jobs are:

$$\begin{aligned}
 C_{e_k} &= \bar{d}_{e_k} + \sum_{i=1}^{k-1} \lambda_i \\
 &+ \beta \left[\sum_{i=1}^{k-1} \left(\lambda_i \left(p_e i + 2H + (3H + B)(i-1) + \sum_{l=1}^{i-1} \lambda_l \right) \right. \right. \\
 &\quad \left. \left. + (3H + B) \sum_{l=1}^{i-1} \lambda_l + W(X_i) \right) - B^2(k-1) \right] \tag{11}
 \end{aligned}$$

for $k = 1, \dots, m$ and based on Eq. 10, the criterion value C_{\max} (i.e., the completion time of the last job in X_m) is given as

$$\begin{aligned}
 C_{\max} &= y + \sum_{i=1}^m \lambda_i \\
 &+ \beta \left[\sum_{i=1}^m \left(\lambda_i \left(p_e i + 2H + (3H + B)(i-1) + \sum_{l=1}^{i-1} \lambda_l \right) \right. \right. \\
 &\quad \left. \left. + (3H + B) \sum_{l=1}^{i-1} \lambda_l + W(X_i) \right) - B^2 m \right]. \tag{12}
 \end{aligned}$$

Now we will show that the answer for DSP is *yes* (i.e., $C_{\max} \leq y = D$ and $C_{e_i} \leq \bar{d}_{e_i}$ for $i = 1, \dots, m$) if and only if it is *yes* for 3PP (i.e., $\lambda_i = 0$ for $i = 1, \dots, m$).

Only if Assume that the answer for 3PP is *yes*. Thus, for each subset X_i ($i = 1, \dots, m$) $\sum_{q \in X_i} x_q = B$ holds, thereby $\lambda_i = 0$ for $i = 1, \dots, m$. Based on Eqs. 6 and 11, the completion time of e_k ($k = 1, \dots, m$) can be estimated as follows:

$$C_{e_k} = \bar{d}_{e_k} + \beta \left[\sum_{i=1}^{k-1} W(X_i) - B^2(k-1) \right] < \bar{d}_{e_k},$$

thus, there are no late jobs. Further, from Eqs. 6 and 12 follows that

$$C_{\max} = y + \beta \left[\sum_{i=1}^m W(X_i) - B^2m \right] < y.$$

If Assume now that the answer for 3PP is *no*. Let $\sum_{q \in X_i} x_q = B + \lambda_i$ for $i = 1, \dots, m$; recall that from the assumption $\frac{B}{4} < x_q < \frac{B}{2}$ (for $q = 1, \dots, 3m$) follows that $\frac{3}{4}B < \sum_{q \in X_u} x_q < \frac{3}{2}B$, thereby $\lambda_i \in (-\frac{B}{4}, \frac{B}{2})$.

Thus, for any partition of the set $\{1, \dots, 3m\}$ into disjoint subsets X_1, \dots, X_m , there must exist at least two subsets X_u and X_w ($u \neq w$) such that $\sum_{q \in X_u} x_q \neq \sum_{q \in X_w} x_q$ for $u, w \in \{1, \dots, m\}$ and $u < w$. For this proof, it is enough to consider only two cases, since any distribution of λ_i (following the partition of jobs) can be represented by these cases. They are given as follows:

- (a) $\lambda_u > 0$ and $\lambda_w < 0$, such that $\sum_{i=1}^{u-1} \lambda_i = 0$ and w is the index of the first set X_w for which $\sum_{l=u}^w \lambda_l \leq 0$, i.e., $\sum_{l=u}^i \lambda_l > 0$ for $i = u, \dots, w - 1$.
- (b) $\lambda_u < 0$ and $\lambda_w > 0$, such that $\sum_{i=1}^{u-1} \lambda_i = 0$ and w is the index of the first set X_w for which $\sum_{l=u}^w \lambda_l \geq 0$, i.e., $\sum_{l=u}^i \lambda_l < 0$ for $i = u, \dots, w - 1$.

where $u, w \in \{1, \dots, m\}$ and $u < w$. Consider case (a) and assume that X_w is the first one such that $\sum_{i=u}^w \lambda_i \leq 0$ and if $\sum_{i=u}^w \lambda_i + \lambda_{w+1} < 0$ (i.e., $\lambda_{w+1} < 0$), then there must exist such $k > w + 1$, for which $\lambda_k > 0$ and $\sum_{i=w+1}^k \lambda_i \geq 0$, but this is represented by case (b). Thus, without loss of generality, we assume that $\lambda_i = 0$ for $i \in \{1, \dots, u - 1\} \cup \{w + 1, \dots, m\}$.

Consider case (a). Since $\lambda_i = 0$ for $i = 1, \dots, u - 1$ and $\lambda_u \geq 1$, then from Eq. 11 follows

$$\begin{aligned} C_{e_{u+1}} &= \bar{d}_{e_{u+1}} + \sum_{i=1}^u \lambda_i \\ &+ \beta \left[\sum_{i=1}^u \left(\lambda_i \left(p_e i + 2H + (3H + B)(i - 1) + \sum_{l=1}^{i-1} \lambda_l \right) \right. \right. \\ &\quad \left. \left. + (3H + B) \sum_{l=1}^{i-1} \lambda_l + W(X_i) \right) - B^2u \right] \\ &= \bar{d}_{e_{u+1}} + \lambda_u + \beta \left[\lambda_u (p_e u + 2H + (3H + B)(u - 1)) \right. \\ &\quad \left. + \sum_{i=1}^u W(X_i) - B^2u \right] \\ &> \bar{d}_{e_{u+1}} + \lambda_u + \beta [p_e u - B^2u] > \bar{d}_{e_{u+1}}. \end{aligned}$$

Consider case (b). Recall that $\lambda_i = 0$ for $i \in \{1, \dots, u - 1\} \cup \{w + 1, \dots, m\}$; thus, the criterion value is equal to

$$\begin{aligned} C_{\max} &= y + \sum_{i=u}^w \lambda_i \\ &+ \beta \left[\sum_{i=u}^w \left(\lambda_i \left(p_e i + 2H + (3H + B)(i - 1) + \sum_{l=1}^{i-1} \lambda_l \right) \right) \right. \\ &\quad \left. + (3H + B) \sum_{i=1}^m \sum_{l=1}^{i-1} \lambda_l + \sum_{i=1}^m W(X_i) - B^2m \right]. \end{aligned}$$

Note that $\sum_{l=1}^{i-1} \lambda_l > -(i - 1)\frac{B}{4}$ and $\sum_{i=1}^m \sum_{l=1}^{i-1} \lambda_l > -\frac{1}{4}m^2B$; thus, we have

$$\begin{aligned} C_{\max} &> y + \beta \left[p_e \sum_{i=u}^w i \lambda_i - \frac{1}{4}m^2B(3H + B) - B^2m \right] \\ &> y + \beta \left[p_e \sum_{i=u}^w i \lambda_i - \frac{1}{4}mB(3Hm + (m + 4)B) \right]. \end{aligned}$$

On this basis and taking into consideration that $\sum_{i=u}^w i \lambda_i = w \sum_{i=u}^w \lambda_i - \sum_{i=u}^{w-1} \sum_{l=u}^i \lambda_l \geq 1$ (note that $\sum_{l=u}^i \lambda_l < 0$ for $i = u, \dots, w - 1$ and $\sum_{i=u}^w \lambda_i \geq 0$) and $(m + 4)B < 2mB = H$, the criterion value can be further estimated

$$\begin{aligned} C_{\max} &> y + \beta \left[p_e \left(w \sum_{i=u}^w \lambda_i - \sum_{i=u}^{w-1} \sum_{l=u}^i \lambda_l \right) - \frac{1}{4}mB(3Hm + H) \right] \\ &> y + \beta [p_e - m^2BH] = y. \end{aligned}$$

Therefore, for all the cases at least one job is late or the criterion value C_{\max} is greater than y .

We hereby showed that DSP has an answer *yes* if and only if the answer for 3PP is also *yes*, which means DSP is strongly NP-complete, thereby the considered scheduling problem is strongly NP-hard. Note that the proof is correct for $\beta = 1$ and also for very small (practical) values, e.g., $\beta = 1/y$, and even for big numbers that have at last a marginal practical justification. \square

From the strong NP-hardness of $1|\bar{d}_j, \tilde{p}_j(v) = p_j(1 + \sum_{l=1}^{v-1} \beta p_{[l]})^{\alpha_j}, \alpha_j \in \{0, 1\}|C_{\max}$ follows that the maximum lateness minimization problem with the considered model of the aging effect is also strongly NP-hard.

Corollary 2 *The problem*

$$1|\tilde{p}_j(v) = p_j \left(1 + \sum_{l=1}^{v-1} \beta p_{[l]} \right)^{\alpha_j}, \alpha_j \in \{0, 1\} | L_{\max}$$

is strongly NP-hard even for arbitrary $\beta > 0$.

Therefore, to efficiently solve such real-life problems, more advanced algorithms have to be used (e.g., metaheuristics). However, such methods require additional analysis to determine the values of jobs parameters that describe aging.

4 Conclusions

In this paper, we introduced a new model of the aging effect that extended the sum-of-processing-time-based approach such that the fatigue caused by each job to the machine is equal to the non-decreasing function dependent on its normal processing time. Moreover, we constructed the polynomial time algorithm that optimally solves the single machine maximum completion time minimization problem with the considered model of the aging effect. The practitioners can apply the algorithm without the complete knowledge about the optimized system, but they only have to determine for which jobs $\alpha_j = 0$, approximate shape of f (convex/concave), and find relations (lower/equal/greater) between values of the normal processing times of jobs. Finally, we proved that the makespan minimization problem with the sum-of-processing-time-based aging model (where some job processing times are constant) is strongly NP-hard if jobs can have deadlines, thereby the maximum lateness minimization problem with this model is not less complex. Therefore, to efficiently solve such real-life problems, heuristic or metaheuristic algorithms have to be constructed that will be our future work.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Chang PC, Chen SH, Mani V (2009) A note on due-date assignment and single machine scheduling with a learning/aging effect. *Int J Prod Econ* 117:142–149
- Cheng TCE, Ding Q, Lin BMT (2004) A concise survey of scheduling with time-dependent processing times. *Eur J Oper Res* 152:1–13
- Cheng TCE, Lee WC, Wu CC (2010) Single-machine scheduling with deteriorating functions for job processing times. *Appl Math Model* 34:4171–4178
- Dababneh AJ, Swanson N, Shell RL (2001) Impact of added rest breaks on the productivity and well being of workers. *Ergonomics* 44:164–174
- Eilon S (1964) On a mechanistic approach to fatigue and rest periods. *Int J Prod Res* 3:327–332
- Garey MR, Johnson DS (1979) *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, San Francisco
- Gawiejnowicz S (1996) A note on scheduling on a single processor with speed dependent on a number of executed jobs. *Inf Process Lett* 57:297–300
- Gordon V, Potts C, Strusevich V, Whitehead J (2008) Single machine scheduling models with deterioration and learning: handling precedence constraints via priority generation. *J Sched* 11:357–370
- Graham RL, Lawler EL, Lenstra JK, Rinnooy Kan AHG (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. *Ann Discrete Math* 5:287–326
- Janiak A, Rudek R (2010) Scheduling jobs under an aging effect. *J Oper Res Soc* 61:1041–1048
- Kuo WH, Yang DL (2008) Minimizing the makespan in a single-machine scheduling problem with the cyclic process of an aging effect. *J Oper Res Soc* 59:416–420
- Lai PJ, Lee WC (2011) Single-machine scheduling with general sum-of-processing-time-based and position-based learning effects. *OMEGA Int J Manage S* 39:467–471
- Lai PJ, Lee WC, Chen HH (2011) Scheduling with deteriorating jobs and past-sequence-dependent setup times. *Int J Adv Manuf Technol* 54:737–741
- Lee WC, Lai PJ (2011) Scheduling problems with general effects of deterioration and learning. *Inf Sci* 181:1164–1170
- Liu P, Zhou X, Tang L (2010) Two-agent single-machine scheduling with position-dependent processing times. *Int J Adv Manuf Technol* 48:325–331
- Mandich NV (2003) Overview of surface preparation of metals prior to finishing: part 2. *Met Finish* 101:33–58
- McMullen PR, Clark M, Albritton D, Bell J (2003) A correlation and heuristic approach for obtaining production sequences requiring a minimum of tool replacements. *Comput Oper Res* 30:443–462
- Mosheiov G, Sarig A (2008) A due-window assignment problem with position-dependent processing times. *J Oper Res Soc* 59:997–1003
- Sheikh AK, Raouf A, Sekerdey UA, Younas M (1999) Optimal tool replacement and resetting strategies in automated manufacturing systems. *Int J Prod Res* 37:917–937
- Stanford M, Lister P (2004) Investigation into the relationship between tool-wear and cutting environments when turning EN32 steel. *Ind Lubr Tribol* 56:114–121
- Wang JB, Wang LY, Wang D, Huang X, Wang XR (2009) A note on single-machine total completion time problem with general deteriorating function. *Int J Adv Manuf Technol* 44:1213–1218
- Wang JB, Wang LY, Wang D, Wang XY (2009) Single-machine scheduling with a time-dependent deterioration. *Int J Adv Manuf Technol* 43:805–809

23. Yang SJ, Yang DL (2010) Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities. *OMEGA Int J Manage S* 38:528–533
24. Yang SJ, Yang DL (2010) Minimizing the total completion time in single-machine scheduling with aging/deteriorating effects and deteriorating maintenance activities. *Comput Math Appl* 60:2161–2169
25. Yang SJ, Yang DL, Cheng T (2010) Single-machine due-window assignment and scheduling with job-dependent aging effects and deteriorating maintenance. *Comput Oper Res* 37:1510–1514
26. Zhao CL, Tang HY (2010) Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan. *Appl Math Model* 34:837–841