Applied Mathematics and Computation 216 (2010) 438–447

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Scheduling time-dependent jobs under mixed deterioration

Stanisław Gawiejnowicz [a,*], Bertrand M.T. Lin [b]

[a] Adam Mickiewicz University, Faculty of Mathematics and Computer Science, Umultowska 87, 61-614 Poznań, Poland
[b] Department of Information and Finance Management, Institute of Information Management, National Chiao Tung University, 1001 Ta Hsueh Road, Hsinchu 300, Taiwan

ARTICLE INFO

Keywords:
Scheduling
Single machine
Deteriorating jobs
Polynomial-time algorithms
$\mathcal{NP}$-hard problems

ABSTRACT

We consider a new model of time-dependent scheduling. A set of deteriorating jobs has to be processed on a single machine which is available starting from a non-zero time. The processing times of some jobs from this set are constant, while other ones are either proportional or linear functions of the job starting times. The applied criteria of schedule optimality include the maximum completion time, the total completion time, the total weighted completion time, the maximum lateness and the number of tardy jobs. We delineate a sharp boundary between computationally easy and difficult problems, showing polynomially solvable and $\mathcal{NP}$-hard cases.

1. Introduction

The notion of a job plays a basic role in scheduling theory. The processing time, in turn, is one of the basic parameters that describe a job in this theory. In the scheduling theory literature, the processing times of jobs are described in various ways. In classical scheduling (Conway et al. [1]), the processing times of jobs, once given, are *fixed* throughout the decision process. In time-dependent scheduling (Alidaee and Womer [2], Cheng et al. [3], Gawiejnowicz [4]), jobs may have *variable* processing times. The assumption that job processing times are variable allows to cover in the framework of time-dependent scheduling many problems that cannot be formulated as classical scheduling problems. For example, the problems of repayment of multiple loans (Gupta et al. [5]), recognizing aerial threats (Ho et al. [6]), scheduling maintenance assignments (Mosheiov [7]), scheduling resource-constrained jobs (Zhao and Tang [8]) or scheduling derusting operations (Gawiejnowicz et al. [9]) may be formulated as time-dependent scheduling problems.

In time-dependent scheduling, the processing time of a job is a function of the starting time of the job. In general, functions describing job processing times can be arbitrary non-negative functions of time. If job processing times are non-decreasing functions of time, we deal with the phenomenon of *job processing time deterioration*. Jobs with deteriorating processing times are called *deteriorating jobs*. In the simplest case of deteriorating jobs, we deal with proportional job processing times, i.e., the processing time of each job is described by a proportional function of the starting time of this job. A generalization of the proportional case is the linear case, where job processing times are linear functions of job starting times. Both the proportional and the linear job proportional times are most often encountered in time-dependent scheduling literature. We refer the reader to reviews by Alidaee and Womer [2] and Cheng et al. [3] for an introductory view on this subject. The book by Gawiejnowicz [4] includes detailed discussion of time-dependent scheduling and covers the results not presented in the two references. Recently published papers from the area include, e.g., Barketau et al. [10], Chung

* Corresponding author.
E-mail addresses: stgawiej@amu.edu.pl (S. Gawiejnowicz), bmtlin@iim.nctu.edu.tw (B.M.T. Lin).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2010.01.037

et al. [11], Gawiejnowicz et al. [12,13], Lee et al. [14], Qi et al. [15], Wang and Sun [16], Zhao and Tang [17], Gawiejnowicz and Kononov [18], Lodree and Geiger [19], Ng et al. [20], and Zhu et al. [21].

All the above-mentioned results concerning scheduling deteriorating jobs apply the same model of job processing time deterioration. There are, however, two points which may make this model not well justified in real applications: only a single deterioration form is assumed and the schedule starts at time zero. In our opinion, the assumptions are not well suited for some practical problems. Consider, for instance, the problem of modeling the work of a team of cleaning workers. In this team are engaged fully fledged workers, average workers and beginners. The team has to clean a set of rooms, devices etc. Each cleaning task requires a certain working duration and depends on the starting time, i.e., if the task starts later, it will take a longer time. The most experienced workers can do the cleaning tasks in a constant time; the average staff (the beginners) will need a proportional (a linear) time, with respect to the time when the task starts. Moreover, in view of organizational issues, we cannot assume that the team starts its work at the beginning of planning horizon but at some later time.

The leitmotiv of this paper is a new model of job processing time deterioration, called *mixed deterioration*, which allows us to extend the applicability of time-dependent scheduling to situations as the one described above. In subsequent sections of this paper, we formulate the new model and propose a new notation for denoting scheduling problems considered in this model. We also prove several properties of such problems. Finally, we indicate which of these problems are polynomially solvable and $\mathcal{NP}$-hard, thus delineating a sharp boundary between easy and hard cases.

The paper is organized as follows. In Section 2 we formulate the problem under consideration and introduce the notation used throughout this paper. Preliminary results are given in Section 3. We present the main results in Section 4. Conclusions and remarks for future research are given in Section 5.

2. Problem formulation and notation

We consider the following time-dependent scheduling problem. A set $\mathcal{J}$ of n deteriorating jobs $J_1, J_2, \ldots, J_n$ has to be processed on a single machine, available from the *initial starting time* $t_0 > 0$. All jobs are independent, non-preemptable and available for processing at time t_0. Each job $J_j \in \mathcal{J}$ has a due-date $d_j \geqslant 0$ and a weight $w_j > 0$ indicating the relative importance of the job in the set $\mathcal{J}$. The processing time p_j of job $J_j \in \mathcal{J}$ deteriorates in time, i.e., it is a non-decreasing function of the actual starting time s_j of the job. We admit job processing times that are constant, proportional or linear functions of s_j, i.e., $p_j = a_j, p_j = b_j s_j$ or $p_j = A_j + B_j s_j$, where $a_j > 0, b_j > 0, A_j > 0$ and $B_j > 0$ for $1 \leqslant j \leqslant n$. (If no ambiguity will arise, we write t instead of s_j, since the starting time s_j is the variable on which processing time p_j depends.)

Our aim is to find a schedule which minimizes the applied criterion of schedule optimality. In the paper, we consider the standard objectives of the maximum completion time ($C_{\max}$), the total completion time $\left(\sum C_j\right)$, the total weighted completion time $\left(\sum w_j C_j\right)$, the maximum lateness ($L_{\max}$), and the number of tardy jobs $\left(\sum U_j\right)$.

Describing scheduling problems, we use the $\alpha|\beta|\gamma$ notation (Graham et al. [22]) with the following extension. Symbols $a_j, b_j t$ and $A_j + B_j t$ denote the constant, the proportional and the linear processing times, respectively. Since we consider scheduling problems in which different jobs can have different forms of processing times, we add to the β field numbers indicating the number of jobs of a particular type. For example, by $1|p_j \in \{a_j, n_1; b_j t, n_2\}|C_{\max}$ we denote the problem of minimizing the $C_{\max}$ criterion for a set of $n = n_1 + n_2$ jobs, among which n_1 jobs have constant processing times and the other n_2 jobs have proportional processing times. If in the β field we use only one function describing job processing times, it means that all jobs have the same form of processing times; in such a case we omit the numbers of jobs of particular types. For example, by $1|p_j = A_j + B_j t|C_{\max}$ we denote the single machine problem with n jobs that have only linear processing times.

We write that a job is *fixed* (*proportional*, *linear*), if the processing time of the job is a constant (proportional, linear) function of its starting time. For a given schedule $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$, symbols $C_j(\sigma), \gamma(\sigma)$ and w_{σ_k} denote the completion time of job J_j in σ, the value of criterion γ for σ and the weight of the kth job in σ, respectively. Other parameters of jobs will be denoted in a similar way.

3. Preliminary results

In this section, we state some auxiliary results. We start with the following lemma which can be proved by contradiction.

Lemma 1. *Let $f_j(t)$ be a strictly increasing function of t, $f_j(t) > 0$ for $t > 0$ and γ be a regular criterion. Then an optimal schedule for problem $1|p_j = A_j + f_j(t)|\gamma$ is a non-delay schedule.*

Taking into account Lemma 1, hereafter we will identify a schedule and an appropriate sequence of job indices.

Lemma 2. *Let σ be a schedule of linear jobs starting from time $t_0 > 0$. Then the weighted completion time of the ith job in σ, $1 \leqslant i \leqslant n$, is equal to*

$$w_{\sigma_i} C_i(\sigma) = w_{\sigma_i}\left(\sum_{j=1}^{i} A_{\sigma_j} \prod_{k=j+1}^{i} (1 + B_{\sigma_k}) + t_0 \prod_{j=1}^{i} (1 + B_{\sigma_j})\right). \quad (1)$$

Proof. By mathematical induction with respect to the value of i. □

The next result is a generalization of the case where $t_0 = 0$ (Gawiejnowicz and Pankowska [23], Tanaev et al. [24]).

Property 1. *If $t_0 > 0$, then the optimal schedule for problem $1|p_j = A_j + B_j t|C_{\max}$ can be obtained by scheduling jobs in non-increasing order of $\frac{B_j}{A_j}$ ratios.*

Proof. Let σ be a schedule for problem $1|p_j = A_j + B_j t|C_{\max}$. By Lemma 2, assuming that $w_{\sigma_i} = 1$ for $1 \leqslant i \leqslant n$, the maximum completion time for schedule σ of linear jobs is equal to

$$C_{\max}(\sigma) = C_n(\sigma) = \sum_{j=1}^{n} A_{\sigma_j} \prod_{k=j+1}^{n} (1 + B_{\sigma_k}) + t_0 \prod_{j=1}^{n} (1 + B_{\sigma_j}). \tag{2}$$

Since the term $t_0 \prod_{j=1}^{n} (1 + B_{\sigma_j})$ has the same value for all possible job sequences, the value of $C_{\max}$ is minimized when the sum $\sum_{j=1}^{n} A_{\sigma_j} \prod_{k=j+1}^{n} (1 + B_{\sigma_k})$ is minimized. By Rau [25], the minimum is attained when the elements of the sum are in non-increasing order of $\frac{B_j}{A_j}$ ratios. □

By Lemma 2, summing (1) for $1 \leqslant i \leqslant l$, we obtain the formula for the total weighted completion time of the first l jobs in schedule σ of linear jobs.

Lemma 3. *Let σ be a schedule of linear jobs starting from time $t_0 > 0$. Then the total weighted completion time of the first l jobs in $\sigma, 1 \leqslant l \leqslant n$, is equal to*

$$\sum_{i=1}^{l} w_{\sigma_i} C_i(\sigma) = \sum_{i=1}^{l} w_{\sigma_i} \left(\sum_{j=1}^{i} A_{\sigma_j} \prod_{k=j+1}^{i} (1 + B_{\sigma_k}) + t_0 \prod_{j=1}^{i} (1 + B_{\sigma_j}) \right). \tag{3}$$

As a consequence of Lemma 3, the total weighted completion time for schedule σ of linear jobs is equal to

$$\begin{aligned} \sum w_j C(\sigma_j) &= \sum_{i=1}^{n} w_{\sigma_i} C_i(\sigma) = \sum_{i=1}^{n} w_{\sigma_i} \left(\sum_{j=1}^{i} A_{\sigma_j} \prod_{k=j+1}^{i} (1 + B_{\sigma_k}) + t_0 \prod_{j=1}^{i} (1 + B_{\sigma_j}) \right) \\ &= \sum_{i=1}^{n} w_{\sigma_i} \sum_{j=1}^{i} A_{\sigma_j} \prod_{k=j+1}^{i} (1 + B_{\sigma_k}) + t_0 \sum_{i=1}^{n} w_{\sigma_i} \prod_{j=1}^{i} (1 + B_{\sigma_j}). \end{aligned} \tag{4}$$

Assuming in (4) that $w_{\sigma_i} = 1$ for $1 \leqslant i \leqslant n$, we obtain a formula for the total completion time for schedule σ of linear jobs:

$$\sum C_j(\sigma) = \sum_{i=1}^{n} \sum_{j=1}^{i} A_{\sigma_j} \prod_{k=j+1}^{i} (1 + B_{\sigma_k}) + t_0 \sum_{i=1}^{n} \prod_{j=1}^{i} (1 + B_{\sigma_j}). \tag{5}$$

By Lemma 2, assuming in (1) that $A_{\sigma_j} = 0$ for all $1 \leqslant j \leqslant i$ for some $1 \leqslant i \leqslant n$, we obtain a formula for the weighted completion time for the ith job in schedule σ of proportional jobs.

Lemma 4. *Let σ be a schedule of proportional jobs starting from time $t_0 > 0$. Then the weighted completion time of the ith job in $\sigma, 1 \leqslant i \leqslant n$, is independent of the order of jobs in the schedule and it is equal to*

$$w_{\sigma_i} C_i(\sigma) = w_{\sigma_i} t_0 \prod_{j=1}^{i} (1 + b_{\sigma_j}). \tag{6}$$

Assuming in (6) that $w_{\sigma_j} = 1$ for all $1 \leqslant j \leqslant i$ for some $1 \leqslant i \leqslant n$, we obtain a formula for the completion time of the ith job in a schedule of proportional jobs (Mosheiov [26]).

Lemma 5. *Let σ be a schedule of proportional jobs starting from time $t_0 > 0$. Then, the completion time of the ith job in $\sigma, 1 \leqslant i \leqslant n$, is independent of the order of jobs in the schedule and it is equal to*

$$C_i(\sigma) = t_0 \prod_{j=1}^{i} (1 + b_{\sigma_j}). \tag{7}$$

By Lemma 5, assuming $i = n$, we obtain the next result (Mosheiov [26]).

Property 2. *If $t_0 > 0$, then the maximum completion time of schedule σ of proportional jobs is independent of the order of jobs in the schedule and it is equal to*

$$C_{\max}(\sigma) = C_n(\sigma) = t_0 \prod_{j=1}^{n} (1 + b_{\sigma_j}). \tag{8}$$

By Lemma 3, assuming in (3) that $A_{\sigma_j} = 0$ and $B_j \equiv b_j$ for $1 \leqslant j \leqslant n$, we obtain a formula for the total weighted completion time of the first l jobs in schedule σ of proportional jobs.

Lemma 6. *Let σ be a schedule of proportional jobs starting from time $t_0 > 0$. Then the total weighted completion time of the first l jobs in $\sigma, 1 \leqslant l \leqslant n$, is equal to*

$$\sum_{i=1}^{l} w_{\sigma_i} C_i(\sigma) = t_0 \sum_{i=1}^{l} w_{\sigma_i} \prod_{j=1}^{i} (1 + b_{\sigma_j}). \tag{9}$$

By Lemma 6, assuming $l = n$, and by applying Rau [25], we obtain the following result.

Property 3. *If $t_0 > 0$, then the total weighted completion time of schedule σ of proportional jobs is equal to*

$$\sum w_j C_j(\sigma) = \sum_{i=1}^{n} w_{\sigma_i} C_i(\sigma) = t_0 \sum_{i=1}^{n} w_{\sigma_i} \prod_{j=1}^{i} (1 + b_{\sigma_j}), \tag{10}$$

which is minimized by scheduling the jobs in non-decreasing order of $\frac{b_j}{w_j}$ ratios.

By Property 3, assuming $w_{\sigma_i} = 1$ for $1 \leqslant i \leqslant n$, and applying Rau [25], we obtain the next result (Mosheiov [26]).

Property 4. *If $t_0 > 0$, then the total completion time of schedule σ of proportional jobs is equal to*

$$\sum C_j(\sigma) = \sum_{i=1}^{n} C_i(\sigma) = t_0 \sum_{i=1}^{n} \prod_{j=1}^{i} (1 + b_{\sigma_j}), \tag{11}$$

which is minimized by scheduling the jobs in non-decreasing order of b_j values.

We close this section with the formulation of an $\mathcal{NP}$-complete problem which will be used in the subsequent discussion on $\mathcal{NP}$-hardness results in Section 4.

SUBSET PRODUCT (*SP*): given integer $B \in \mathbb{Z}^+$, a set $M = \{1, 2, \ldots, m\}$ and size $x_i \in \mathbb{Z}^+$ for each $i \in M$, does there exist a subset $M' \subseteq M$ such that $\prod_{i \in M'} x_i = B$?

Without loss of generality, we assume that $2 \leqslant x_i < B$ for all $i \in M$. We also assume that $B < X = \prod_{i \in M} x_i$, since otherwise the problem is trivial. The *SP* problem is $\mathcal{NP}$-complete in the ordinary sense (Johnson [27]).

4. Main results

In this section, we present the main results for the newly proposed model of mixed job deterioration. The results are organized into sections according to the applied criteria.

4.1. The maximum completion time criterion

In this section, we present our results for the $C_{\max}$ criterion. In particular, we propose a polynomial-time algorithm to deal with this objective. Let symbol $\oplus$ denote the operator of sequence concatenation.

Algorithm $\mathcal{A}_1$

Step 1: Arrange the proportional jobs in arbitrary order;
Denote the obtained sequence by σ^1;
Step 2: Arrange the linear jobs in non-increasing order of $\frac{B_j}{A_j}$;
Denote the obtained sequence by σ^2;
Step 3: Arrange the fixed jobs in arbitrary order;
Denote the obtained sequence by σ^3;
Step 4: Schedule the jobs in the order given by sequence $\sigma^1 \oplus \sigma^2 \oplus \sigma^3$.

Theorem 1. *Problem $1|p_j \in \{a_j, n_1; b_j t, n_2; A_j + B_j t, n_3\}|C_{\max}$ is solvable by algorithm $\mathcal{A}_1$ in $O(n \log n)$ time.*

Proof. Assume that we are given n_1 fixed jobs, n_2 proportional jobs and n_3 linear jobs. By Lemma 1, we know that in an optimal schedule all the jobs are scheduled without idle times.

Let σ be some schedule, and let J_i and J_j be two consecutive jobs not abiding by the order specified by algorithm $\mathcal{A}_1$. Assume that J_i precedes J_j. Let s_i denote the starting time of job J_i in σ. There are four cases to consider: job J_i and job J_j are linear and $\frac{B_i}{A_i} < \frac{B_j}{A_j}$ (Case 1), job J_i is fixed and job J_j is linear (Case 2), job J_i is fixed and job J_j is proportional (Case 3), job J_i is linear and job J_j is proportional (Case 4). We prove now by adjacent job interchange argument that in each of the four cases we can construct from schedule σ a better schedule, σ', by swapping jobs J_i and J_j.

In Case 1, by Property 1, we can reduce the length of schedule σ by swapping jobs J_i and J_j.

In Case 2, we have $C_i(\sigma) = s_i + a_i$ and $C_j(\sigma) = A_j + (B_j + 1)(s_i + a_i) = A_j + (B_j + 1)s_i + B_j a_i + a_i$. Let σ' be the schedule obtained by swapping jobs J_i and J_j. Then we have $C_j(\sigma') = A_j + (B_j + 1)s_i$ and $C_i(\sigma') = A_j + (B_j + 1)s_i + a_i$. Thus, since $C_i(\sigma') - C_j(\sigma) = -B_j a_i < 0$, schedule σ' is better than schedule σ.

In Case 3, assuming that $A_j = 0$ and $B_j \equiv b_j$ for $1 \leqslant j \leqslant n_2$, and by applying reasoning from Case 2, we also conclude that schedule σ' is better than schedule σ.

In Case 4, we have $C_i(\sigma) = A_i + (B_i + 1)s_i$ and $C_j(\sigma) = (b_j + 1)(A_i + (B_i + 1)s_i) = b_j A_i + A_i + (B_i + 1)(b_j + 1)s_i$. By swapping jobs J_i and J_j, we have $C_j(\sigma') = (b_j + 1)s_i$ and $C_i(\sigma') = A_i + (B_i + 1)(b_j + 1)s_i$. Thus, since $C_i(\sigma') - C_j(\sigma) = -b_j A_i < 0$, schedule σ' is better than schedule σ.

By swapping, if necessary, other pairs of jobs as above, we obtain a schedule composed of three blocks, each of which contains jobs of the same type. Moreover, the block of proportional jobs precedes the block of linear jobs, which is followed by the block of fixed jobs. The order of jobs in the block of linear jobs in the schedule is determined by Property 1, while the order of proportional jobs and the order of fixed jobs are immaterial. The time complexity of algorithm $\mathscr{A}_1$ is determined by Step 2, which takes $O(n \log n)$ time. □

4.2. The total completion time criterion

In this section, we focus on the $\sum C_j$ objective. We start with the case of fixed and proportional jobs. Consider the following example.

Example 1. We are given jobs J_1, J_2, J_3 and J_4 with job processing times $p_1 = 1, p_2 = 2, p_3 = 2t$ and $p_4 = 3t$, respectively. The initial starting time $t_0 = 1$. There are 24 possible sequences as shown in Table 1.

Example 1 shows that it is more difficult to deal with $\sum C_j$ than $C_{\max}$ because an optimal schedule for problem $1|p_j \in \{a_j, n_1; b_j t, n_2\}|\sum C_j$ does not necessarily consist of two disjoint blocks, each of which contains jobs of the same type. We can prove, however, some properties of an optimal schedule for the $\sum C_j$ criterion.

Property 5. *In an optimal schedule for problem $1|p_j \in \{a_j, n_1; b_j t, n_2\}|\sum C_j$, the fixed jobs are scheduled in non-decreasing order of a_j values.*

Proof. Assume that in an optimal schedule σ there exist fixed jobs J_i and J_j such that $a_i < a_j$ but J_j precedes J_i. Denote by $\mathscr{J}^1, \mathscr{J}^2$ and $\mathscr{J}^3$ the sets of jobs scheduled before job J_j, between jobs J_j and J_i, and after job J_i, respectively. Let σ^l denote the sequence of the jobs of the set $\mathscr{J}^l$, where $l = 1, 2, 3$. Thus, the considered optimal schedule is in the form of $\sigma = (\sigma^1, j, \sigma^2, i, \sigma^3)$. Let $\sigma' = (\sigma^1, i, \sigma^2, j, \sigma^3)$ be the schedule obtained by swapping jobs J_j and J_i. Then, since

Table 1
All possible schedules for data from Example 1.

$(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$	$(C_{\sigma_1}, C_{\sigma_2}, C_{\sigma_3}, C_{\sigma_4})$	$\sum C_j$
(1, 2, 3, 4)	(2, 4, 12, 48)	66
(1, 2, 4, 3)	(2, 4, 16, 48)	70
(1, 3, 2, 4)	(2, 6, 8, 32)	48
(1, 3, 4, 2)	(2, 6, 24, 26)	58
(1, 4, 2, 3)	(2, 8, 10, 30)	50
(1, 4, 3, 2)	(2, 8, 24, 26)	60
(2, 1, 3, 4)	(3, 4, 12, 48)	67
(2, 1, 4, 3)	(3, 4, 16, 48)	71
(2, 3, 1, 4)	(3, 9, 10, 40)	62
(2, 3, 4, 1)	(3, 9, 36, 37)	85
(2, 4, 1, 3)	(3, 12, 13, 39)	67
(2, 4, 3, 1)	(3, 12, 36, 37)	88
(**3, 1, 2, 4**)	(**3, 4, 6, 24**)	**37**
(3, 1, 4, 2)	(3, 4, 16, 18)	41
(3, 2, 1, 4)	(3, 5, 6, 24)	38
(3, 2, 4, 1)	(3, 5, 20, 21)	49
(3, 4, 1, 2)	(3, 12, 13, 15)	43
(3, 4, 2, 1)	(3, 12, 14, 15)	44
(**4, 1, 2, 3**)	(**4, 5, 7, 21**)	**37**
(4, 1, 3, 2)	(4, 5, 15, 17)	41
(4, 2, 1, 3)	(4, 6, 7, 21)	38
(4, 2, 3, 1)	(4, 6, 18, 19)	47
(4, 3, 1, 2)	(4, 12, 13, 15)	44
(4, 3, 2, 1)	(4, 12, 14, 15)	45

$p_i = a_i < a_j = p_j$, we have $C_i(\sigma') < C_j(\sigma)$. Because proportional functions are increasing with respect to job starting times, the actual processing times of the jobs in $\mathscr{J}^2$ either remain the same (fixed jobs) or decrease (proportional jobs). Therefore, the starting time of job J_j in σ' is not later than that of job J_i in σ, i.e., $C_j(\sigma') \leqslant C_i(\sigma)$. But this implies that $\sum C_j(\sigma') < \sum C_j(\sigma)$. A contradiction. □

Property 5 specifies the ordering of the fixed jobs in an optimal schedule for problem $1|p_j \in \{a_j, n_1; b_j t, n_2\}|\sum C_j$. A similar property holds also for proportional jobs.

Property 6. *In an optimal schedule for problem $1|p_j \in \{a_j, n_1; b_j t, n_2\}|\sum C_j$, the proportional jobs are scheduled in non-decreasing order of b_j values.*

Proof. The result follows from the same reasoning as in Property 5. □

Properties 5 and 6 imply that an optimal schedule for problem $1|p_j \in \{a_j, n_1; b_j t, n_2\}|\sum C_j$ is composed of separate blocks of fixed and proportional jobs, and that in each block the jobs of the same type are arranged accordingly. However, the number of jobs in each block and the mutual relations between these blocks are unknown. Example 1 shows that the proportional jobs scatter in the sequence among the blocks of fixed jobs. This suggests the following way of construction of an optimal schedule for problem $1|p_j \in \{a_j, n_1; b_j t, n_2\}|\sum C_j$.

Namely, the optimal schedule can be constructed by optimally interleaving the two sorted sequences of fixed jobs and proportional jobs. Given a sorted sequence of n_1 fixed jobs and a sorted sequence of n_2 proportional jobs, there are $\binom{n_1+n_2}{n_1}$ different interleaving sequences. Therefore, if one of n_1 and n_2 is constant, then the problem can be solved in polynomial time. To demonstrate the above result, first we consider problem $1|p_j \in \{a_j, n-1; b_j t, 1\}|\sum C_j$.

In this case, we have a set of n jobs and only one of them is proportional, i.e., $n_1 = n-1$ and $n_2 = 1$. For simplicity of further presentation, we will denote fixed jobs by $J_1, J_2, \ldots, J_{n-1}$ and the proportional job by J_n. The next result gives a formula which describes the total completion time of a given schedule for the problem.

Property 7. *Let $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_k, \sigma_{k+1}, \sigma_{k+2}, \ldots, \sigma_n)$ be a schedule for problem $1|p_j \in \{a_j, n-1; b_j t, 1\}|\sum C_j$ in which the proportional job J_n is scheduled in the $(k+1)$th position, i.e., $\sigma_{k+1} = n, 0 \leqslant k \leqslant n-1$. Then the total completion time of schedule σ is equal to*

$$\sum C_j(\sigma) = kt_0 + \sum_{j=1}^{n-1}(n-j)a_{\sigma_j} + (n-k)(b_n+1)\left(t_0 + \sum_{j=1}^{k} a_{\sigma_j}\right). \tag{12}$$

Proof. Let $\mathscr{J}^1$ (respectively, $\mathscr{J}^2$) denote the set of jobs scheduled before (respectively, after) the proportional job J_n in a given schedule σ, where $|\mathscr{J}^1| = k$ and $|\mathscr{J}^2| = n-k-1$. Then,

$$\sum C_j(\sigma) = \sum_{J_j \in \mathscr{J}^1} C_j(\sigma) + C_n(\sigma) + \sum_{J_j \in \mathscr{J}^2} C_j(\sigma).$$

Calculate the subsequent terms of the sum $\sum C_j(\sigma)$. First, note that

$$\sum_{J_j \in \mathscr{J}^1} C_j(\sigma) = \sum_{j=1}^{k} C_j(\sigma) = kt_0 + \sum_{j=1}^{k}(k-j+1)a_{\sigma_j}.$$

Job J_n is completed at the time

$$C_n(\sigma) = C_k(\sigma) + p_n = (1+b_n)C_k(\sigma), \quad \text{where } C_k(\sigma) = t_0 + \sum_{j=1}^{k} a_{\sigma_j}.$$

In order to calculate $\sum_{J_j \in \mathscr{J}^2} C_j(\sigma)$, it is sufficient to note that since job J_n is scheduled in the $(k+1)$th position in schedule σ, the completion times of jobs $J_{\sigma_{k+1}}, J_{\sigma_{k+2}}, \ldots, J_{\sigma_{n-1}}$ increase p_n units of time. Thus

$$\sum_{J_j \in \mathscr{J}^2} C_j(\sigma) = \sum_{j=k+2}^{n-1} C_j(\sigma) = \sum_{j=k+2}^{n-1}(p_n + (n-j+1)a_{\sigma_j}) = (n-k-1)(1+b_n)\left(t_0 + \sum_{j=1}^{k} a_{\sigma_j}\right) + \sum_{j=k+2}^{n-1}(n-j+1)a_{\sigma_j}.$$

Collecting all the terms together, we obtain formula (12). □

Note that Property 7 does not allow us to specify the optimal position of job J_n in schedule σ, since the total completion time $\sum C_j(\sigma)$ depends on mutual relations between the values of $t_0, a_1, a_2, \ldots, a_{n-1}, b_n$ and n.

However, since J_n is a single proportional job, we can choose the best schedule from among $n+1$ ones in which the proportional job is scheduled before (position 0), inside (positions $1, 2, \ldots, n-1$) or after (position n) the sequence of the fixed jobs. This can be done by the following algorithm.

Algorithm $\mathscr{A}_2$

Step 1: Arrange the fixed jobs in non-decreasing order of a_j;
Denote the obtained sequence by $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{n-1})$;
Step 2: For $0 \leqslant k \leqslant n$ construct $n+1$ schedules by scheduling job J_n in the kth position in σ;
Step 3: Choose the best schedule from among the schedules generated in Step 2.

Theorem 2. *Problem* $1|p_j \in \{a_j, n-1; b_j t, 1\}| \sum C_j$ *is solvable in* $O(n \log n)$ *time by algorithm* $\mathscr{A}_2$.

Proof. The optimality of algorithm $\mathscr{A}_2$ follows from Properties 5–6. Since Step 1 needs $O(n \log n)$ time, while both Step 2 and Step 3 require $O(n)$ time, the running time of the algorithm is $O(n \log n)$. □

Algorithm $\mathscr{A}_2$ can be generalized for any fixed number m of proportional jobs, $1 \leqslant m \leqslant n$. In this case, we have to construct $(n+1) \times n \times \cdots \times (n-m+1) = O(n^m)$ schedules and to choose the best from among them. Therefore, there holds the following result.

Theorem 3. *If* $1 \leqslant m \leqslant n$, *then problem* $1|p_j \in \{a_j, n-m; b_j t, m\}| \sum C_j$ *is solvable in* $O(\max\{n \log n, n^m\})$ *time.*

Before closing our discussion on the $\sum C_j$ objective, we note that the complexity status of problems $1|p_j \in \{a_j, n_1; A_j + B_j t, n_2\}| \sum C_j$, $1|p_j \in \{b_j t, n_1; A_j + B_j t, n_2\}| \sum C_j$ and $1|p_j \in \{a_j, n_1; b_j t, n_2; A_j + B_j t, n_3\}| \sum C_j$ is open, since the time complexity of problem $1|p_j = A_j + B_j t| \sum C_j$ problem is still unknown, even if $A_j = 1$ for $1 \leqslant j \leqslant n$.

4.3. The total weighted completion time criterion

In this section, we present the results for the $\sum w_j C_j$ criterion. Since this criterion is a generalization of the $\sum C_j$ criterion, the situation is similar to that we dealt with in Section 4.2, i.e., we can prove only some properties of optimal schedules for $\sum w_j C_j$. We start with the following example.

Example 2. We are given jobs J_1, J_2, J_3 with job processing times $p_1 = 1, p_2 = 2, p_3 = t$ and weights $w_1 = 8, w_2 = 1, w_3 = 3$, respectively. The initial starting time $t_0 = 1$. There are six possible schedules as shown in Table 2.

Similarly as in the case of criterion $\sum C_j$ (cf. Example 1), an optimal schedule for problem $1|p_j \in \{a_j, n_1; b_j t, n_2\}| \sum w_j C_j$ is not necessarily composed of two job blocks, with jobs of the same type in each of them. We can prove, however, some optimality properties concerning the jobs contained in the same block.

Property 8. *In an optimal schedule for problem* $1|p_j \in \{a_j, n_1; b_j t, n_2\}| \sum w_j C_j$, *the jobs in a block of fixed jobs are scheduled in non-decreasing order of* $\frac{a_j}{w_j}$ *values.*

Proof. Let $\sigma = (\sigma^1, j, i, \sigma^2)$ be an optimal schedule in which J_i and J_j are fixed jobs such that $\frac{a_i}{w_i} < \frac{a_j}{w_j}$ and J_j immediately precedes J_i. Denote by $\mathscr{J}^1$ (respectively, $\mathscr{J}^2$) the set of jobs scheduled before job J_i (respectively, after job J_j), and let sequence σ^1 (respectively, σ^2) contain the jobs of set $\mathscr{J}^1$ (respectively, $\mathscr{J}^2$).

Let s_j denote the starting time of job J_j in σ. Then $w_j C_j(\sigma) = w_j(s_j + a_j)$ and $w_i C_i(\sigma) = w_i(s_j + a_j + a_i)$. Consider now the schedule $\sigma' = (\sigma^1, i, j, \sigma^2)$, obtained by swapping jobs J_j and J_i. Then $w_i C_i(\sigma') = w_i(s_i + a_i)$ and $w_j C_j(\sigma') = w_j(s_i + a_i + a_j)$. Since the swap does not affect the completion of any job of $\mathscr{J}^1 \cup \mathscr{J}^2$, and since $C_j(\sigma) + C_i(\sigma) - C_i(\sigma') - C_j(\sigma') = w_i a_j - w_j a_i = w_i w_j \left(\frac{a_j}{w_j} - \frac{a_i}{w_i}\right) > 0$, the schedule σ' has a smaller objective value than σ. A contradiction. □

A similar property holds for the proportional jobs.

Property 9. *In an optimal schedule for* $1|p_j \in \{a_j, n_1; b_j t, n_2\}| \sum w_j C_j$, *the jobs in the block of proportional jobs are scheduled in non-decreasing order of* $\frac{b_j}{w_j}$ *values.*

Proof. The result follows from the same reasoning as in Property 8. □

Table 2
All possible schedules for data from Example 2.

$(\sigma_1, \sigma_2, \sigma_3)$	$(C_{\sigma_1}, C_{\sigma_2}, C_{\sigma_3})$	$\sum w_j C_j$
(1, 2, 3)	(2, 4, 8)	44
(1, 3, 2)	**(2, 4, 6)**	**34**
(2, 1, 3)	(3, 4, 8)	59
(2, 3, 1)	(3, 6, 7)	77
(3, 1, 2)	(2, 3, 5)	35
(3, 2, 1)	(2, 4, 5)	50

Note that we cannot apply Properties 8 and 9 in order to solve problem $1|p_j \in \{a_j, n_1; b_j t, 1\}|\sum w_j C_j$ using the approach from Section 4.2. The properties state that jobs within the same block should be ordered, but jobs in different blocks, even of the same type, are not necessarily confined by the mentioned sequencing rules. In consequence, an exponential number of possible blocks of jobs may exist. The complexity status of this problem remains open.

If we pass to the case of linear jobs, then the problem becomes computationally intractable.

Theorem 4. *Problems* $1|p_j \in \{a_j, n_1; A_j + B_j t, n_2\}|\sum w_j C_j$, $1|p_j \in \{b_j t, n_1; A_j + B_j t, n_2\}|\sum w_j C_j$ *and* $1|p_j \in \{a_j, n_1; b_j t, n_2; A_j + B_j t, n_3\}|\sum w_j C_j$ *are* $\mathcal{NP}$*-hard in the ordinary sense.*

Proof. The theorem follows from the fact that problem $1|p_j = A_j + B_j t|\sum w_j C_j$ is $\mathcal{NP}$-hard in the ordinary sense (Bachman et al. [28]). □

4.4. The maximum lateness

This section addresses the $L_{\max}$ criterion. We start with the case with fixed jobs and proportional jobs.

Theorem 5. *Problem* $1|p_j \in \{a_j, 1; b_j t, n-1\}|L_{\max}$ *is* $\mathcal{NP}$*-hard in the ordinary sense.*

Proof. First, we formulate the decision version *TDL* of the considered problem: given numbers $t_0 > 0$ and $y \geqslant 0$, the set $\{1, 2, \ldots, n\}$ for some natural n, a number $b_j > 0$ for each $1 \leqslant j \leqslant n-1$, and a number $a_n > 0$, does there exist a single-machine schedule σ for jobs with time-dependent processing times in the form of $p_j = b_j t$, $1 \leqslant j \leqslant n-1$, and $p_n = a_n$, such that these jobs are scheduled starting from time t_0 and $L_{\max}(\sigma) \leqslant y$?

Now, we will show that problem *SP* can be reduced to problem *TDL* in polynomial time.

Let I_{SP} denote an instance of the *SP* problem, composed of $B \in \mathbb{Z}^+$, set $M = \{1, 2, \ldots, m\}$ and size $x_j \in \mathbb{Z}^+ \setminus \{1\}$ for each $j \in M$. Construct instance I_{TDL} of the *TDL* problem as follows: $n = m+1, t_0 = 1, b_j = x_j - 1$ and $d_j = 2X$ for $1 \leqslant j \leqslant m$, $a_{m+1} = B, d_{m+1} = 2B$. Set the threshold value $y = 0$.

Given instance I_{TDL} and a particular schedule σ for it, we can check in polynomial time whether $L_{\max} \leqslant y$. Therefore, the *TDL* problem is in $\mathcal{NP}$. Now, we will show that the answer to instance I_{SP} is affirmative if and only if the answer to instance I_{TDL} is affirmative.

Assume that instance I_{SP} has a desired solution. Then there exists a set $M' \subseteq M$ such that $\prod_{j \in M'} x_j = B$. Denote by $\mathcal{J}_{M'}$ (respectively, $\mathcal{J}_{M \setminus M'}$) the set of jobs corresponding to the elements of set M' (respectively, $M \setminus M'$). Starting from $t_0 = 1$, we schedule the jobs without idle times as follows. First, in an arbitrary order, are scheduled the jobs of set $\mathcal{J}_{M'}$. Next, follows job J_{m+1}. Finally, in an arbitrary order, are scheduled the jobs of set $\mathcal{J}_{M \setminus M'}$. Denote the schedule σ. Let $C(S)$ denote the completion time of the last job from a particular job set S.

By Property 2, we have

$$C(\mathcal{J}_{M'}) = t_0 \prod_{j \in M'} (1 + b_j) = \prod_{j \in M'} x_j = B.$$

Then $C_{m+1} = C(\mathcal{J}_{M'}) + B = 2B$. Since $\prod_{j \in M \setminus M'} (1 + b_j) = \frac{X}{B}$, we have

$$C(\mathcal{J}_{M \setminus M'}) = C_{m+1} \prod_{j \in M \setminus M'} (1 + b_j) = 2B \times \frac{X}{B} = 2X.$$

Therefore, since $d_{m+1} = 2B$ and $d_j = 2X$ for $j \in M$, we have

$$L_{\max}(\sigma) = \max\{C_j - d_j : j \in M \cup \{m+1\}\} = \max\{2X - 2X, 2B - 2B\} = 0.$$

Hence, the instance I_{TDL} of problem *TDL* has a solution.

Assume now that for the instance I_{TDL} there exists schedule σ such that $L_{\max}(\sigma) \leqslant 0$. Let $\mathcal{J}^1 (\mathcal{J}^2)$ denote the set of jobs scheduled in σ before (after) job J_{m+1}. We will show that then the instance I_{SP} of problem *SP* has a solution.

First, we will show that job J_{m+1} can start only at time B. Indeed, job J_{m+1} cannot start at time t_0, since then

$$\mathcal{J}^1 = \emptyset, \quad \mathcal{J}^2 = \mathcal{J}, \quad C_{\max}(\sigma) = C_{m+1} \prod_{J_j \in \mathcal{J}} (1 + b_j) = BX > 2X,$$

and

$$L_{\max}(\sigma) = \max\{B - 2B, BX - 2X\} = \max\{-B, (B-2)X\} = (B-2)X > 0.$$

Moreover, job J_{m+1} cannot start later than B, since then $L_{\max}(\sigma) \geqslant C_{m+1} - d_{m+1} > 0$. Finally, this job cannot start earlier than B. Indeed, assume that the job starts at time $B - b$, where $0 < b < B$. It means that $\mathcal{J}^1 \neq \emptyset, C(\mathcal{J}^1) = B - b$ and $C_{m+1} = B - b + B = 2B - b$. This, in turn, implies that

$$C(\mathcal{J}^2) = (2B - b) \times \frac{X}{B - b} = 2X + \frac{bX}{B - b} > 2X,$$

and $L_{\max}(\sigma) > 0$. Therefore, job J_{m+1} must start its processing at time B and hence $C_{m+1} = B + B = 2B$. By Property 1, no idle interval is allowed in schedule σ. Therefore, the completion time of the last job in $\mathcal{J}^1$ must be coincident with the start time of job J_{m+1}, implying that $\prod_{j \in M'} x_j = B$. □

The case of linear jobs is not easier than the case of proportional jobs. Therefore, we have the following theorem.

Theorem 6. *Problems* $1|p_j \in \{a_j, n_1; A_j + B_j t, n_2\}|L_{\max}$, $1|p_j \in \{b_j t, n_1; A_j + B_j t, n_2\}|L_{\max}$ *and* $1|p_j \in \{a_j, n_1; b_j t, n_2; A_j + B_j t, n_3\}|L_{\max}$ *are* $\mathcal{NP}$*-hard in the ordinary sense.*

Proof. All the problems include problem $1|p_j = A_j + B_j t|L_{\max}$, which is known to be $\mathcal{NP}$-hard in the ordinary sense (Kononov [29], Bachman and Janiak [30]), as a special case. The result readily follows. □

4.5. The number of tardy jobs

Theorem 7. *Problem* $1|p_j \in \{a_j, 1; b_j t, n_2\}|\sum U_j$ *is* $\mathcal{NP}$*-hard in the ordinary sense.*

Proof. By the reduction used in the proof of Theorem 5, we can construct an instance of our problem for which we have that the *SP* problem has a solution if and only if $\sum U_j = 0$. □

The complexity status of problems $1|p_j \in \{a_j, n_1; A_j + B_j t, n_2\}|\sum U_j$, $1|p_j \in \{b_j t, n_1; A_j + B_j t, n_2\}|\sum U_j$ and $1|p_j \in \{a_j, n_1; b_j t, n_2; A_j + B_j t, n_3\}|\sum U_j$ is open, since the time complexity of problem $1|p_j = A_j + B_j t|\sum U_j$ remains unknown.

Closing the section, we note that the question about which of the problems mentioned in Theorems 4–7 are strongly $\mathcal{NP}$-hard remains open.

5. Conclusions

In the paper, we considered single machine time-dependent scheduling problems in a new mixed deterioration model. The objectives of the makespan, the total completion time, the total weighted completion time, the maximum lateness and the number of tardy jobs were investigated. We identified several problems to be polynomial by proposing polynomial-time algorithms. We also indicated several $\mathcal{NP}$-hard problems.

Further research may be focused on two topics. First, for these problems with mixed deterioration that are $\mathcal{NP}$-hard we may develop polynomial-time heuristics with good performance ratios. It would also be interesting to investigate which existing heuristics for pure deterioration problems can be adopted for our mixed deterioration model. Second, we can consider potential extensions of the set of job processing times by other functions than the fixed, proportional or linear ones.

Acknowledgments

Dr. Gawiejnowicz was supported in part by grant N519 1889 93 of the Ministry of Science and Higher Education of Poland. Dr. Lin was supported in part by the Taiwan-Russia Joint Project under grant NSC97-2923-H-009-001-MY3.

References

[1] R.W. Conway, W.L. Maxwell, L.W. Miller, Theory of Scheduling, Addison-Wesley, Reading, 1967.
[2] B. Alidaee, N.K. Womer, Scheduling with time dependent processing times: review and extensions, Journal of the Operational Research Society 50 (1999) 711–720.
[3] T.C.E. Cheng, Q. Ding, B.M.T. Lin, A concise survey of scheduling with time-dependent scheduling times, European Journal of Operational Research 152 (2004) 1–13.
[4] S. Gawiejnowicz, Time-Dependent Scheduling. Monographs in Theoretical Computer Science: An EATCS Series, Springer, Berlin-New York, 2008.
[5] S.K. Gupta, A.S. Kunnathur, K. Dandapani, Optimal repayment policies for multiple loans, Omega 15 (1987) 323–330.
[6] K.I.J. Ho, J.Y.T. Leung, W.D. Wei, Complexity of scheduling tasks with time-dependent execution times, Information Processing Letters 48 (1993) 315–320.
[7] G. Mosheiov, Scheduling jobs with step-deterioration: minimizing makespan on a single- and multi-machine, Computers and Industrial Engineering 28 (1995) 869–879.
[8] C.L. Zhao, H.Y. Tang, Single machine scheduling problems with deteriorating jobs, Applied Mathematics and Computation 161 (2005) 865–874.
[9] S. Gawiejnowicz, W. Kurc, L. Pankowska, Analysis of a time-dependent scheduling problem by signatures of deterioration rate sequences, Discrete Applied Mathematics 154 (2006) 2150–2166.
[10] M.S. Barketau, T.C.E. Cheng, C.T. Ng, V. Kotov, M.Y. Kovalyov, Batch scheduling of step deteriorating jobs, Journal of Scheduling 11 (2008) 17–28.
[11] Y.H. Chung, H.C. Liu, C.C. Wu, W.C. Lee, A deteriorating jobs problem with quadratic function of job lateness, Computers and Industrial Engineering 57 (2009) 1182–1186.
[12] S. Gawiejnowicz, W. Kurc, L. Pankowska, Equivalent time-dependent scheduling problems, European Journal of Operational Research 196 (2009) 919–929.
[13] S. Gawiejnowicz, W. Kurc, L. Pankowska, Conjugate problems in time-dependent scheduling, Journal of Scheduling 12 (2009) 543–553.
[14] W.C. Lee, Y.R. Shiau, S.K. Chen, C.C. Wu, A two-machine flow shop scheduling problem with deteriorating jobs and blocking, International Journal of Production Economics 124 (2010) 188–197.
[15] X.L. Qi, S.G. Zhou, J.J. Yuan, Single machine parallel-batch scheduling with deteriorating jobs, Theoretical Computer Science 410 (2009) 830–836.
[16] J.B. Wang, L.Y. Sun, Single-machine group scheduling with linearly decreasing time-dependent setup times and job processing times, International Journal of Advanced Manufacturing Technology (in press), doi: 10.1007/s00170-009-2444-6.
[17] C.L. Zhao, H.Y. Tang, Rescheduling problems with deteriorating jobs under disruptions, Applied Mathematical Modelling 34 (2010) 238–243.

[18] S. Gawiejnowicz, A. Kononov, Complexity and approximability of scheduling resumable proportionally deteriorating jobs, European Journal of Operational Research 200 (2010) 305–308.
[19] E.J. Lodree Jr., C.D. Geiger, A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration, European Journal of Operational Research 201 (2010) 644–648.
[20] C.T. Ng, J.B. Wang, T.C.E. Cheng, L.L. Liu, A branch-and-bound algorithm for solving a two-machine flow shop problem with deteriorating jobs, Computers and Operations Research 37 (2010) 83–90.
[21] V.C.Y. Zhu, L.Y. Sun, L.H. Sun, X.H. Li, Single-machine scheduling time-dependent jobs with resource-dependent ready times, Computers and Industrial Engineering 58 (2010) 84–87.
[22] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, Annals of Discrete Mathematics 5 (1979) 287–326.
[23] S. Gawiejnowicz, L. Pankowska, Scheduling jobs with varying processing times, Information Processing Letters 54 (1995) 173–175.
[24] V.S. Tanaev, V.S. Gordon, Y.M. Shafransky, Scheduling Theory: Single-Stage Systems, Kluwer, Dordrecht, 1994.
[25] G. Rau, Minimizing a function of permutations of n integers, Operations Research 19 (1971) 237–240.
[26] G. Mosheiov, Scheduling jobs under simple linear deterioration, Computers and Operations Research 21 (1994) 653–659.
[27] D.S. Johnson, The NP-completeness column: an ongoing guide, Journal of Algorithms 2 (1982) 393–405.
[28] A. Bachman, A. Janiak, M.Y. Kovalyov, Minimizing the total weighted completion time of deteriorating jobs, Information Processing Letters 81 (2002) 81–84.
[29] A. Kononov, Scheduling problems with linear increasing processing times, in: U. Zimmermann et al. (Eds.), Operations Research 1996, Springer, 1997, pp. 208–212.
[30] A. Bachman, A. Janiak, Minimizing maximum lateness under linear deterioration, European Journal of Operational Research 126 (2000) 557–566.