# Due-date assignment and parallel-machine scheduling with deteriorating jobs*

T.C.E. Cheng[1], L.Y. Kang[1,2], C.T. Ng[1]

[1] Department of Logistics, The Hong Kong Polytechnic University

[2]Department of Mathematics, Shanghai University, Shanghai 200444, China

**Abstract**

In this paper we study the problem of scheduling $n$ deteriorating jobs on $m$ identical parallel machines. Each job's processing time is a nondecreasing function of its start time. The problem is to determine an optimal combination of the due-date and schedule so as to minimize the sum of the due-date, earliness and tardiness penalties. We show that this problem is NP-hard, and we present a heuristic algorithm to find near-optimal solutions for the problem. When the due-date penalty is 0, we present a polynomial time algorithm to solve it.

**Key words**: deteriorating jobs; parallel-machine scheduling; due-date

## Introduction

Machine scheduling problems with time-dependent processing times have received increasing attention in recent years. Browne and Yechiali (Browne and Yechiali, 1990) introduced a scheduling problem with deteriorating jobs. In this problem the job processing time is a linear nondecreasing start-time-dependent function. Deterioration in processing time may occur when the machine gradually loses efficiency in the course of processing jobs. At the beginning the machine is at its highest level of efficiency. The efficiency loss is reflected by the fact that a job processed later has a longer processing time.

There are many applications of the model where the job processing time is an increasing function of the job start time. These include the control of queues and communication systems,

* *Correspondence: TCE Cheng, Department of Logistics, The Hong Kong Polytechnic University, 11 Yuk Choi Road, Hung Hom, Kowloon, Hong Kong. E-mail: lgtcheng@polyu.edu.hk*

shops with deteriorating machines, and/or delay of maintenance or cleaning, fire fighting, and hospital emergency wards, scheduling steel production in rolling mills (see Browne and Yechiali, 1990, Kubiak and van de Velde, 1998, Kunnathur and Gupta, 1990, Ng et al, 2002 ). Scheduling in the settings described above is known as scheduling of *deteriorating* jobs. The processing time of a deteriorating job $J_j$ is given by $f_j(t)$, where $f_j(t)$ is a nondecreasing function of the job start time $t$. Most of the related studies ( Bachman and Janiak, 2000, Bachman et al, 2002, Cheng et al, 2004, Cheng et al, 2004, Cheng and Ding, 2000, Hsieh and Bricker, 1997) are confined to linear deterioration.

In this paper we study the scheduling of deteriorating jobs in the context of the common due-date problem (CDDP), which deals with job scheduling on machines in a just-in-time (JIT) production environment (see Baker and Scudder, 1990, Gordon et al, 2002[a], Gordon et al, 2002[b], Cheng et al, 2002, Panwalker et al, 1982). In a JIT system, jobs are to be completed neither too early nor too late, which leads to the scheduling problem involving both earliness and tardiness costs and the cost of assigning due-dates. Completing a job early means having to bear the costs of holding unnecessary inventories, while finishing a job late results in a contractual penalty and a loss of customer goodwill.

The due-date assignment problem arises when a firm offers a due-date to its customers during sale negotiations. The firm has to offer a price reduction if the offered due-date far exceeds the one expected by the customers. In many instances, due-dates are negotiated rather than simply dictated by the customers. The later the due-dates are fixed, the higher the probability that the product will be completed or delivered on time. In order to maintain a good image with the customers, many companies tolerate reasonable holding costs in favour of keeping the established due-dates. Thus, the decision maker has to balance the losses resulting from the holding costs and the advantages of finishing the orders on time.

Applications of the common due-date problem in real-life situations abound. Baker and Scudder (Baker and Scudder, 1990) observed that treating due-date as decision variables reflects the practice in some shops of setting due-dates internally, as targets to guide the progress of shop floor activities. Prescribing a common due-date might represent a situation where several items constitute a single customer's order, or it might reflect an assembly environment in which the components should all be ready at the same time in order to avoid staging delays.

## Problem formulation

Let $\mathcal{J} = \{1, 2, \ldots, n\}$ be a set of $n$ jobs and $m$ identical parallel machines be given. The job processing times deteriorate linearly as a function of their start times. In the remainder of this paper, we denote the normal processing time of job $j$ by $a_j$, and its actual processing time if processed at time $t$ on a machine by $p_j(t) = a_j + b_j(t - t_0)$, where $t_0$ is the start time of the machine, $b_j > 0$ $(j = 1, 2, \ldots, n)$ is the job-dependent deterioration rate of job $J_j$, which determines the job's (actual) processing time at $t > t_0$. In this paper we study the case where the job-independent deterioration rates are identical for all the jobs, i.e., $b_i = b$. For any given schedule $\sigma$, define

$$\begin{aligned}
s_j(\sigma) &= \text{the processing time of the machine before job } j \text{ is processed,} \\
d &= \text{the common due date,} \\
p_j(\sigma) &= a_j + bs_j(\sigma), \text{ i.e., the actual processing time of job } j, \\
C_j(\sigma) &= \text{the completion time of job } j, \\
E_j(\sigma) &= \max\{0, d - C_j(\sigma)\}, \text{ i.e., the earliness of job } j, \\
T_j(\sigma) &= \max\{0, C_j(\sigma) - d\}, \text{ i.e., the tardiness of job } j, \\
f(d, \sigma) &= \sum(\alpha E_j(\sigma) + \beta T_j(\sigma) + \gamma d), \text{ i.e., the total penalty function, where } \alpha, \beta, \text{and } \gamma \text{ are} \\
&\quad\ \text{the unit earliness, tardiness and due-date penalty, respectively.}
\end{aligned}$$

Consider the problem of scheduling $n$ jobs on $m$ identical parallel machines. Each job can be processed by any of the $m$ machines, taking into account the following constraints: a machine performs at most one job at a time, and each job is processed by at most one machine at a time. The goal is to determine an optimal combination of the due date $d^*$ and schedule $\sigma^*$ so that $f(d, \sigma)$ is minimized. Using the three-field notation of Graham et al. (Graham et al, 1979), the problem is denoted as $Pm|p_j(s_j) = a_j + bs_j| \sum(\alpha E_j + \beta T_j + \gamma d)$.

Panwalker et al. (Panwalker et al, 1982) were the first researchers to consider $1|p_j| \sum(\alpha E_j + \beta T_j + \gamma d)$, and they gave an $O(n \log n)$ algorithm to solve the problem. For identical parallel machines, Cheng and Chen (Cheng and Chen, 1994), and De et al. (De et al, 1994) showed that $Pm|p_j| \sum(\alpha E_j + \beta T_j + \gamma d)$ is NP-hard even if $m = 2$. Cheng (Cheng, 1989) proposed a heuristic algorithm for this problem. For identical and uniform parallel machines, Emmons (Emmons, 1987) proposed an $O(n \log n)$ algorithms for $P|p_j| \sum(\alpha E_j + \beta T_j)$ and $Q|p_j| \sum(\alpha E_j + \beta T_j)$. Cheng et al. (Cheng et al, 2004) studied the problem $1|p_j(s_j) = a_j + bs_j| \sum(\alpha E_j + \beta T_j + \gamma d)$ and presented an $O(n \log n)$ algorithm to solve the problem.

In the following section, we give an NP-hardness proof for $Pm|p_j(s_j) = a_j + bs_j| \sum(\alpha E_j +$

$\beta T_j + \gamma d)$. In Section 4 we develop a polynomial-time algorithm that finds an optimal solution for the problem $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j)$, i.e., when the due-date penalty is 0. In the last section we present a heuristic algorithm to find an approximate solution for $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j + \gamma d)$.

## NP-hardness proof for the problem $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j + \gamma d)$

We prove the NP-hardness of the problem $Pm|p_j = a_j + bs_j|\sum(\alpha E_j + \beta T_j + \gamma d)$ by showing that its corresponding decision problem is NP-complete from a reduction of the PARTITION problem. The PARTITION problem can be stated as follows.

PARTITION: Given a finite set $A = \{1, 2, \ldots, n\}$ and a size $a_j \in Z^+$ for each $j \in A$, does there exist a subset $B \subseteq A$ such that the following holds?

$$\sum_{j \in B} a_j = \sum_{j \in A \setminus B} a_j = \frac{1}{2} \sum_{j \in A} a_j$$

**Theorem 1** *The problem $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j + \gamma d)$ is NP-hard even when $m = 2$.*

**Proof.** (i) It is easy to see that the problem is in NP. (ii) We show that PARTITION can be reduced to the problem $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j + \gamma d)$.

Let $I$ be an instance of PARTITION. Without loss of generality, we assume that $a_1 \leq a_2 \leq \ldots \leq a_n$ and $n \geq 4$. Let $M = \frac{1}{2}\sum_{j \in A} a_j$, $e = na_n + (n-1)a_{n-1} + \ldots + a_1$. We construct a corresponding instance $I'$ of the problem $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j + \gamma d)$: A set $\mathcal{J} = \{1, \ldots, n\}$ of jobs, a number $m \in Z^+$ of machines, a normal processing time $a_j \in Z^+$ for each job $j \in \mathcal{J}$, job-dependent deterioration rate $b_j = b = (1 + \frac{1}{a_n^2})^{\frac{1}{2n^4}} - 1$, three penalties $\alpha = 1, \beta = ((2M+1)e+1)/b, \gamma = 2e/n$ and $K = (2M+1)e+1$. We show that there exists a subset $B \subseteq A$ such that $\sum_{j \in B} a_j = \sum_{i \in A \setminus B} a_i$ for instance $I$, if and only if there exist a schedule $\sigma$ and a due-date $d$ such that $f(d, \sigma) \leq K$.

(a) If there exists a subset $B \subseteq A$ such that $\sum_{j \in B} = \sum_{j \in A \setminus B} a_j$ for instance $I$, we set $d = M + \frac{1}{2n^2 a_n}$, and construct a schedule $\sigma = (\sigma_1, \sigma_2)$ for instance $I'$ such that all the jobs in $B$ are processed on machine 1 from 0 to $d$, and all the remaining jobs are processed on machine

2. Then

$$\begin{aligned}
C_{max} &= \max\{a_{\sigma 1(|B|)} + (1+b)a_{\sigma 1(|B|-1)} + \ldots + (1+b)^{|B|-1}a_{\sigma 1(1)},\ a_{\sigma 2(|A\setminus B|)} \\
&\quad +(1+b)a_{\sigma 2(|A\setminus B|-1)} + \ldots + (1+b)^{|A\setminus B|-1}a_{\sigma 2(1)}\} \\
&\le \max\{a_{\sigma 1(|B|)} + (1+\frac{1}{2a_n^2 n^4})a_{\sigma 1(|B|-1)} + \ldots + (1+\frac{|B|-1}{2n^4 a_n^2})a_{\sigma 1(1)}),\ a_{\sigma 2(|A\setminus B|)} \\
&\quad +(1+\frac{1}{2a_n^2 n^4})a_{\sigma 2(|A\setminus B|-1)} + \ldots + (1+\frac{|A\setminus B|-1}{2n^4 a_n^2})a_{\sigma 2(1)})\} \\
&\le \sum_{j\in B} a_j + \frac{1}{2n^2 a_n} \\
&= M + \frac{1}{2n^2 a_n} = d.
\end{aligned}$$

Clearly, there is no tardy job in $\sigma$ and

$$\sum_{j\in\mathcal{J}} E_j \le \sum_{j\in\mathcal{J}} j a_j + \frac{1}{2na_n} = e + \frac{1}{2na_n}.$$

So we have

$$\begin{aligned}
f(d,\sigma) &= \sum_{j\in\mathcal{J}} E_j + 2e(M + \frac{1}{n^2 a_n}) \\
&\le e + 2Me + \frac{1}{2na_n} + \frac{2e}{2n^2 a_n} \\
&\le e + 2Me + 1 = K.
\end{aligned}$$

(b). If there is no subset $B \subseteq A$ such that $\sum_{j\in B} a_j = \sum_{j\in A\setminus B} a_j$, there is no schedule $\sigma$ and due-date $d \in Z^+ \cup \{0\}$ such that $f(d,\sigma) \le K$.

**Claim 1:** If $f(d,\sigma) \le K$, then $\max_{j\in\mathcal{J}}\{T_j\} \le b$.

**Proof.** Suppose that there exists a tardy job $j$ in $\sigma$ with $T_j > b$, then $f(d,\sigma) \ge \beta T_j > K$.

**Claim 2:** If $\max_{j\in\mathcal{J}}\{T_j\} \le b$ in $\sigma$, then $d \ge M + \frac{1}{2} - b$.

**Proof.** Since there is no subset $B \subseteq A$ such that $\sum_{j\in B} a_j = \sum_{j\in A\setminus B} a_j$ for instance $I$, all the sizes of $A$ are integers and $M = \frac{1}{2}\sum_{j\in A} a_j$, so the maximum completion time of $\sigma$ is no less than $M + \frac{1}{2}$. If $\max_{j\in\mathcal{J}}\{T_j\} \le b$, then $d \ge M + \frac{1}{2} - b$.

Suppose there exist a schedule $\sigma$ and a due-date $d$ such that $f(d,\sigma) \le K$. From Claims 1 and 2, $\max_{j\in\mathcal{J}}\{T_j\} \le b$ and $d \ge M + \frac{1}{2} - b$. It follows that

$$f(d,\sigma) = \sum_{j=1}^{n} (\alpha E_j + \beta T_j + \gamma d)$$

$$
\begin{aligned}
&> 2e(M + \frac{1}{2} - b) + (n-2)a_1 \\
&= 2eM + e - 2eb + (n-2)a_1 \\
&\geq 2eM + e - \frac{1}{n^2 a_n} + (n-2)a_1 \\
&\geq K.
\end{aligned}
$$

This leads to a contradiction. So the assumption cannot hold and the correctness of (b) is established. Combining (i) and (ii), we have shown that the problem $Pm|p_j = a_j + bs_j| \sum(\alpha E_j + \beta T_j + \gamma d)$ is NP-hard. ■

# A polynomial-time algorithm for the problem $Pm|p_j(s_j) = a_j + bs_j| \sum(\alpha E_j + \beta T_j)$

We first present some elementary results.

**Lemma 2** *There exists an optimal schedule in which the machines are not idle between the processing of the jobs.*

**Lemma 3** *( Alidaee and Ahmadian, 1993) Let $\sigma, \pi$ and $d$ be an optimal schedule, the optimal processing times and the optimal due-date for the single machine case, respectively. If the value of the common due-date $d$ is increased to $d_1$, then the value of the objective function remains the same where the schedule starts at time $\Delta = d_1 - d$ with $\sigma, \pi, d_1$ as the optimal schedule, the processing times and the common due-date, respectively.*

**Lemma 4** *( Cheng et al, 2004) Let $\sigma$ be any specified sequence for a single machine. There exists an optimal due-date equal to $C_{\sigma(K)}$, where $K$ is the smallest integer greater than or equal to $(n\beta)/(\alpha + \beta)$, and exactly $K$ jobs will be non-tardy.*

**Lemma 5** *For any given schedule $\sigma$, an optimum common due-date $d$ will coincide with the completion time of some job on each of the machines.*

**Proof.** For any specific machine this result follows from Lemma 4. Let $\sigma = (\sigma_1, \ldots, \sigma_m)$ be a schedule for the $m$ machines, and let $d = \max\{d_i : 1 \leq i \leq m\}$, where $d_i$ is the optimal due-date for the set of jobs scheduled on machine $i$ ($i = 1, \ldots, m$). Using Lemmas 3 and 4, the proof follows from the fact that we can increase the due-date and the start time of the schedule for any given machine without affecting the optimal value of the total cost. ■

For any given schedule $\sigma = (\sigma_1, \ldots, \sigma_m)$, let $n_i$ be the number of jobs scheduled on machine $i$ $(i = 1, \ldots, m)$, then $d_i = C_{\sigma_i(K_i)}$ is the optimal due-date for the set of jobs scheduled on machine $i$, where $K_i = \lceil \frac{n_i\beta}{\alpha+\beta} \rceil$. For notational convenience, we define the following:

$$m_{i,k_i} = \begin{cases} b\sum_{j=k_i}^{K_i}(\alpha(j-1))(1+b)^{j-k_i} & \\ +b\sum_{j=K_i+1}^{n_i}\beta(n_i+1-j)(1+b)^{j-k_i} & \text{for } 1 \le i \le m, 2 \le k_i \le K_i \\ b\sum_{j=k_i}^{n_i}\beta(n_i+1-j)(1+b)^{j-k_i} & \text{for } 1 \le i \le m, K_i+1 \le k_i \le n_i. \end{cases}$$

For $1 \le j \le n, 1 \le i \le m$, let

$$\begin{aligned} c_{j,(i,k_i)} &= (\alpha(k_i-1) + m_{i,k_i+1})a_j & \text{if } 1 \le k_i \le K_i \\ c_{j,(i,k_i)} &= (\beta(n_i+1-k_i) + m_{i,k_i+1})a_j & \text{if } K_i+1 \le k_i \le n_i. \end{aligned}$$

Introducing $d = \max\{C_{\sigma_i(K_i)} : i = 1, \ldots, m\}$, we get

$$\begin{aligned} f(d,\sigma) &= \sum_{i=1}^{m}(\sum_{k_i=1}^{K_i}\alpha(k_i-1)p_{\sigma_i(k_i)} + \sum_{k_i=K_i+1}^{n_i}\beta(n_i+1-k_i)p_{\sigma_i(k_i)}) \\ &= \sum_{i=1}^{m}(\sum_{k_i=1}^{K_i}\alpha(k_i-1)(a_{\sigma_i(k_i)} + ba_{\sigma_i(k_i-1)} + \ldots + b(1+b)^{k_i-2}a_{\sigma_i(1)}) \\ &\quad + \sum_{k_i=K_i+1}^{n_i}\beta(n_i+1-k_i)(a_{\sigma_i(k_i)} + ba_{\sigma_i(k_i-1)} + \ldots + b(1+b)^{k_i-2}a_{\sigma_i(1)})) \\ &= \sum_{i=1}^{m}(\sum_{k_i=1}^{K_i}(\alpha(k_i-1) + m_{i,k_i+1})a_{\sigma_i(k_i)} + \sum_{k_i=K_i+1}^{n_i}(\beta(n_i+1-k_i) + m_{i,k_i+1})a_{\sigma_i(k_i)}) \\ &= \sum_{i=1}^{m}\sum_{k_i=1}^{n_i}c_{\sigma_i(k_i),(i,k_i)}. \end{aligned}$$

Define $A = \{(n_1, n_2, \ldots, n_m) : n_i$ is an integer such that $1 \le n_i \le n-m$ and $\sum_{i=1}^{m} n_i = n\}$. For any $(n_1, \ldots, n_m) \in A$, introduce the variable $x_{j,(i,k_i)}$ $(1 \le i \le m, 1 \le j \le n, 1 \le k_i \le n_i)$ such that $x_{j,(i,k_i)} = 1$ if job $j$ is sequenced $k_i$th on machine $i$, and $x_{j,(i,k_i)} = 0$ otherwise. We can easily show that the restricted problem $P|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j)$ with $n_i$ jobs scheduled on machine $i$ is equivalent to the following weighted bipartite matching problem:

$$\begin{aligned} & \text{Minimize} && \sum_{j}\sum_{(i,k_i)} c_{j,(i,k_i)}x_{j,(i,k_i)} \\ (P) \qquad & s.t. && \sum_{(i,k_i)} x_{j,(i,k_i)} = 1 \qquad j = 1, \ldots, n \\ & && \sum_{j} x_{j,(i,k_i)} = 1 \qquad i = 1, \ldots, m; k_i = 1, \ldots, n_i \\ & && x_{j,(i,k_i)} \in \{0,1\} \quad j = 1, \ldots, n; i = 1, \ldots, m; k_i = 1, \ldots, n_i. \end{aligned}$$

Note that $f(d, \sigma)$ is a weighted sum of $a_j$ values. Number the jobs such that $a_1 \geq \ldots \geq a_n$. In the matching procedure, we consider the jobs in the order $1, 2, \ldots, n$ and match the current job with the smallest available weight. The weight is chosen from a priority queue of the smallest available weights. Since we need $O(n \log n)$ time to arrange the jobs in nonincreasing order of $a_j$, the matching procedure runs in $O(n \log n)$ time.

**Algorithm $\mathcal{A}$**

Step 1. Construct $A = \{(n_1, n_2, \ldots, n_m) : n_i$ is an integer such that$1 \leq n_i \leq n - m$ and $\sum_{i=1}^m n_i = n\}$.

Step 2. For any $(n_1, \ldots, n_m) \in A$, solve the corresponding weighted bipartite matching problem $(P)$. Let $g(n_1, \ldots, n_m) = \min\{\sum_j \sum_{(i,k_i)} c_{j,(i,k_i)} x_{j,(i,k_i)}\}$, $\sigma(n_1, \ldots, n_m)$ be the corresponding schedule, and $d(n_1, \ldots, n_m) = \max\{C_{\sigma_i(K_i)} : i = 1, \ldots, m\}$. If $C_{\sigma_i(K_i)} < d(n_1, \ldots, n_m)$, we change the start time of machine $i$ to $d(n_1, \ldots, n_m) - C_{\sigma_i(K_i)}$.

Step 3. Compute $\text{Min}\{g(n_1, \ldots, n_m) : (n_1, \ldots, n_m) \in A\}$. Let $g(n_1^*, \ldots, n_m^*) = \text{Min}\{g(n_1, \ldots, n_m)|(n_1, \ldots, n_m) \in A\}$, $\sigma^*$ be the corresponding optimal schedule for the restricted problem $P|p_i = a_i + bs_i|\sum(\alpha E_i + \beta T_i)$ with $n_i^*$ jobs scheduled on machine $i$.

Step 4. Let $d = d(n_1^*, \ldots, n_m^*)$, $\sigma = \sigma^*$, $f(d, \sigma) = g(n_1^*, \ldots, g_m^*)$. Output $d, \sigma, f(d, \sigma)$.

To evaluate the complexity of the algorithm, we first note that for any $(n_1, \ldots, n_m) \in A$, the corresponding weighted bipartite matching problem can be implemented to run in $O(n \log n)$ time. The total computational effort to solve the problem amounts to $O(n^{m+1} \log n)$.

**Theorem 6** *Algorithm $\mathcal{A}$ computes in $O(n^{m+1} \log n)$ time an optimal solution for the problem $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j)$.*

# A heuristic algorithm for $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j + \gamma d)$

In this section we give a heuristic algorithm for $Pm|p_j(s_j) = a_j + bs_j|\sum(\alpha E_j + \beta T_j + \gamma d)$.

**Heuristic Algorithm $\mathcal{B}$**

Step 1. Set $l = \lceil \frac{n}{m} \rceil$.

Step 2. Set $K = \lceil \frac{l(\beta - \gamma)}{\alpha + \beta} \rceil$.

Step 3. Set $m_k = \begin{cases} b\sum_{j=k}^{K}(\alpha(j-1)+l\gamma)(1+b)^{j-k} & \\ +b\sum_{j=K+1}^{l}\beta(l+1-j)(1+b)^{j-k} & \text{for } 2 \le k \le K \\ b\sum_{j=k}^{l}\beta(l+1-j)(1+b)^{j-k} & \text{for } K+1 \le k \le l. \end{cases}$

For $1 \le i \le m, 1 \le j \le n$, let

$$\begin{aligned} c_{j,(i,k)} &= (\alpha(k-1)+l\gamma+m_{k+1})a_j && \text{for } 1 \le k \le K \\ c_{j,(i,k)} &= (\beta(l+1-k)+m_{k+1})a_j && \text{for } K+1 \le k \le l. \end{aligned}$$

Step 4. Solve the weighted bipartite matching problem:

$$\begin{array}{lll} & \text{Minimize} & \sum_j \sum_{(i,k)} c_{j,(i,k)} x_{j,(i,k)} \\ (P_1) & s.t. & \sum_{(i,k)} x_{j,(i,k)} = 1 \qquad j = 1,\dots,n \\ & & \sum_j x_{j,(i,k)} \le 1 \qquad i = 1,\dots,m; k = 1,\dots,l \\ & & x_{j,(i,k)} \in \{0,1\} \qquad j = 1,\dots,n; i = 1,\dots,m; k = 1,\dots,l. \end{array}$$

Step 5. A job sequence $\sigma^*$ is generated by assigning job $j$ $k$th on machine $i$ if $(P_1)$ has an optimal solution $x^*$ with $x^*_{j,(i,k)} = 1$. Calculate the completion times of the jobs in $\sigma^*$ and re-index the jobs in non-decreasing order of $C_i$, i.e., $C_1 \le C_2 \le \ldots \le C_n$. Set $d^* = C_r$, where $r = \lceil \frac{n(\beta-\gamma)}{\alpha+\beta} \rceil$.

Step 6. For every machine $i$, compute $C_{\sigma_i^*(q)}$, where $q = \lceil \frac{l\beta}{\alpha+\beta} \rceil$. If $C_{\sigma_i^*(q)} < d^*$, we change the start time of machine $i$ to $d^* - C_{\sigma_i^*(q)}$.

Step 7. Evaluate $f(d^*, \sigma^*) = \sum(\alpha E_j(\sigma^*) + \beta T_j(\sigma^*) + \gamma d^*)$.

For the sake of clarity, we illustrate Heuristic Algorithm $\mathcal{B}$ with an example. We consider a problem of scheduling $n = 5$ jobs on $m = 2$ machines. The job normal processing times are given as $a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 4$, and $a_5 = 5$. The job-independent deterioration $b = 1$. The penalty costs are $\alpha = 2, \beta = 4$, and $\gamma = 1$. Applying Heuristic Algorithm $\mathcal{B}$ yields the following sequence $\sigma^* = (\sigma_1^*, \sigma_2^*)$, $\sigma_1^* = (1,2,4), \sigma_2^* = (3,5)$. In Step 5, $C_1 = 1$(job 1), $C_2 = 3$ (job 3), $C_3 = 4$ (job 2), $C_4 = 8$ (job 5), $C_5 = 12$ (job 4). Since $r = \lceil \frac{5(4-1)}{2+4} \rceil = 3$, set $d^* = C_3 = 4$. Evaluate $f(d^*, \sigma^*) = 88$. It is interesting to note that $\sigma$ is an optimal sequence for this problem. Table 1 shows the results of 9 test runs. It is noted that, for each of the test instances, the heuristic solution is close to the optimal solution.

Table 1. Computational results for Heuristic Algorithm $\mathcal{B}$ (each entry represents 9 randomly generated examples, and Error ratio=$\frac{\text{Heuristic solution}}{\text{Optimal solution}}$).

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Job number $n$ | 5 | 5 | 5 | 5 | 6 | 6 | 8 | 8 | 8 |
| Machine number $m$ | 2 | 2 | 3 | 3 | 2 | 2 | 3 | 2 | 2 |
| Error ratio | 1.071 | 1.064 | 1.151 | 1.216 | 1 | 1.003 | 1.052 | 1.024 | 1.124 |

## Conclusions

We considered the problem of assigning a common due-date and scheduling $n$ deteriorating jobs on $m$ identical parallel machines. The objective is to minimize the sum of the due-date, earliness and tardiness penalties. We showed that the problem is NP-hard even $m = 2$. We presented a heuristic algorithm to find near-optimal solutions for the problem. For the problem with the due-date penalty being 0, we gave a polynomial-time algorithm. Due to deterioration of the job processing times, it is difficult to apply the standard dynamic programming approach to solve the problem. Thus, whether the problem is pseudopolynomially solvable or strongly NP-hard remains open, which is an interesting topic for future research.

*Acknowledgments* This research was supported in part by The Hong Kong Polytechnic University under grant number G-YW81. L.Y. Kang was also supported by the National Natural Science Foundation of China(10571117).

## References

Alidaee B and Ahmadian A (1993). Two parallel machine sequencing problems involving controllable job processing times. European Journal of Operational Research **70**: 335-341.

Bachman A and Janiak A (2000). Minimizing maximum lateness under linear deterioration. European Journal of Operational Research **126**: 557-566.

Bachman A, Janiak A and Kovalyov MY (2002). Minimizing the total weighted completion time of deteriorating jobs. Information Processing Letters **81**: 81-84.

Baker KR and Scudder GD (1990). Sequencing with earliness and tardiness penalities: a review. Operations Research **38**: 22-36.

Browne S and Yechiali U (1990). Scheduling deteriorating jobs on a single processor. Operations Research **38**: 495-498.

Cheng TCE (1989). A heuristic for common due-date assignment and job scheduling on parallel machines. Journal of the Operational Research Society **40**: 1129-1135.

Cheng TCE and Chen ZL (1994). Parallel-machines scheduling problems with earliness and tardiness penalties. Journal of the Operational Research Society **45**: 685-695.

Cheng TCE, Chen ZL and Shakhlevich NV (2002). Common due date assignment and scheduling with ready times. Computers and Operations Research **29**: 1957-1967.

Cheng TCE, Kang LY and Ng CT (2004). Due-date assignment and single machine scheduling with deteriorating jobs. Journal of the Operational Research Society **55**: 198-203.

Cheng TCE , Ding Q and Lin BMT (2004). A concise survey of scheduling with time-dependent processing times. European Journal of Operational Research **152**: 1-13.

Cheng TCE and Ding Q (2000). Single machine scheduling with deadlines and increasing rates of processing times, Acta Informatica **36**, 673-692.

De P, Ghosh JB and Wells CE (1994). Due-date assignment and early/tardy scheduling on identical parallel machines. Naval Research Logistics **41**: 17-32.

Emmons H (1987). Scheduling to a common due date on parallel uniform processors. Naval Research Logistics Quarterly **34**: 803-810.

Gordon VS, Porth JM and Chu CB ($2002^a$). Asurvey of the state-of-art of common due date assignment and scheduling research, European Journal of Operational Research **139**: 1-25.

Gordon VS , Porth JM and Chu CB ($2002^b$). Due date assignment and scheduling: SLK, TWK and other due date assignment models. Production Planning and Control **13**: 117-132.

Graham RL, Lawler EL, Lenstra JK, and Rinnooy Kan AHG (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics **5**: 287-326.

Hsieh YC and Bricker DL (1997). Scheduling linearly deteriorating jobs on multiple machines. Computers and Industrial Engineering **32**: 727-734.

Kubiak W and van de Velde SL (1998). Scheduling deteriorating jobs to minimize makespan. Naval Research Logistics **45**: 511-523.

Kunnathur AS and Gupta SK (1990). Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem. European Journal of Operational Research **47**: 56-64.

Mosheiov G (1996). $\wedge$-shaped polices to schedule deteriorating jobs. Journal of the Operational Research Society **47**: 1184-1191.

Ng CT, Cheng TCE, Bachman A and Janiak A (2002). Three scheduling problems with deteriorating jobs to minimize the total completion time, Information Processing Letters **81**, 327-333.

Panwalker SS, Smith ML and Seidmann A (1982). Common due date assignment to minimize total penalty for one machine scheduling problem. Operations Research **30**: 391-399.