Scheduling Problem with Human Energy Recovery Function under Nonlinear Time-dependent Deterioration

Hua-Ping Wu, Member, IAENG, Min Huang, Vincent Cho, W. H. Ip, and Qun-Lin Fan

Abstract—In this paper, minimizing makespan for single-machine scheduling problem with human energy recovery function under nonlinear time-dependent deterioration is considered. Firstly, a human recovery function is proposed according to the length of a rate-modifying activity (RMA) (i.e., the length of the rest time). Meanwhile, it is the first paper to introduce the recovery function into a scheduling problem. And the problem is proved to be an NP-hard problem. Then, a special case of the problem is proved to be solved in polynomial time.

Index Terms—scheduling, rate-modifying activity, time-dependent deterioration, human recovery function, makespan, NP-hard problem.

I. INTRODUCTION

In this paper, minimizing makespan for single-machine scheduling problem with human energy recovery function under nonlinear time-dependent deterioration is considered. In classical scheduling problems, a machine always works efficiently and effectively. However, this does not fit in the life, even in the case with using the human operators. For example, in the process of loading and unloading cargos in ramp service, the processing time of a job is affected by human operators energy as this process is usually operated by human. It is easy to find that the total processing time of previous jobs and the deterioration rate of human operators have direct effects upon the processing time of later jobs. Hence, it should be more reasonable to describe the processing time of a job as a given constant that follows a nonlinear deterioration instead of linear one. More importance is that the length of break time plays a key role in the process of recovery. By affecting directly the extent of recovery it affects the efficiency of the whole services finally. Therefore, in this paper, the human energy recovery function is the first proposed. The scheduling problem with human energy recovery function under a nonlinear time-dependent deterioration is modeled as a single machine scheduling problem with recovery processing, i.e., a group of human operators is treated as a single machine, while airlines cargos are regarded as jobs which need to be processed. Moreover, for the general scheduling problems, the earlier the service is completed, the more efficient it is. So, the decision maker would more likes to seek minimal makespan as their goal.

Hence, in this paper, a single machine scheduling problem with human energy recovery function under nonlinear time-dependent deterioration is considered. A rational human recovery function is the first proposed to be integrated into deterioration jobs, RMA, and human operators fatigued to minimize makespan. Obviously, this research provides a reference for other practical scheduling problems particularly involved labors, such as maintenance scheduling, cleaning assignments and so on.

The remainder of this paper are organized as follows. Literature reviews are given in section 2. In section 3, the problem and the recovery function are presented and formulated. In section 4, the problem of minimizing makespan for single-machine scheduling with human energy recovery function under nonlinear time-dependent deterioration is proved to be an NP-hard. The special case is considered in section 5. And the conclusions are given in the last section.

II. LITERATURE REVIEW

Deterioration jobs scheduling problem is first introduced independently by Gupta and Gupta[1] and by Browne and Yechiali[2]. Since then, related models have been extensively studied from variety of perspectives. For instance, Wang et al.[3] present single-machine scheduling with deteriorating jobs in which the jobs are subject to a series-parallel graph constraint and prove that the problem of minimizing the makespan and the total weighted completion time can be solved in polynomial time. Toksari and Güner[4] focus on analyzing parallel machine earliness/tardiness scheduling problem with linear deterioration and simultaneous effects of learning. Cheng and Sun[5] consider the problem where the processing time of a job is a linear function of its starting time and jobs can be rejected by paying penalties. They show that the problems of minimizing the makespan, the total weighted completion time and the maximum lateness/tardiness plus the total penalty of the rejected jobs are NP-hard. Hence, they apply dynamic programming to solve the related problem. Li et al.[6] present polynomial-time algorithms to solve the problem that its objective is to determine the optimal due dates and the processing sequence.
simultaneously to minimize costs for earliness, due date assignment and weighted number of tardy jobs, in which the actual processing time of a job is a linear increasing function of its starting time. Ng et al.[7] also study the problem of scheduling deteriorating jobs with release dates on a single machine. These papers all focus on linear deterioration jobs. A few of papers refer to the nonlinear deterioration jobs. Cheng et al.[8] investigate the problem of common due-window assignment and scheduling of deteriorating jobs and a maintenance activity simultaneously on a single-machine, in which the job actual processing times are assumed to follow a linear time-dependent deteriorating model. And they provide polynomial time solutions for the objective of the problem to simultaneously minimize the earliness, tardiness, due-window starting time and due-window size costs. Kuo and Yang[9] introduce a time-dependent learning effect to a single-machine scheduling problem based on the notion that the more one practices, the better one learns. In this regard, the processing times of the subsequent jobs are smaller than their normal processing times because of the learning effect. They define a time-dependent learning effect as follows. Let $p_{it}$ be the actual processing time of job $J_i$ if it was scheduled in position $r$ in a sequence. $p_{ir}$ is the normal processing time of a job if scheduled in the $r^{th}$ position of a sequence. And $p_i$ is the normal (sequence-independent) processing time of job $J_i$. Then $p_{ir} = \left(1 + p_{[1]} + p_{[2]} + \cdots + p_{[r-1]}\right) \cdot \frac{a}{r}$, where $a \leq 0$ is a constant learning index. According to a time-dependent learning effect introduced by Kuo and Yang[9], Wang et al.[10] consider a single machine scheduling problem with a nonlinear time-dependent deterioration. They define the actual processing times as follows. $p_{ir} = \left(1 + p_{[1]} + p_{[2]} + \cdots + p_{[r-1]}\right) \cdot \frac{a}{r}$, where $a \geq 0$ is a constant deterioration index. $p_{ir}$ is the actual processing time of job $J_i$ if it is scheduled at position $r$ in a sequence. $p_{ir}$ is the normal processing time of a job if scheduled in the $r^{th}$ position in a sequence. And $p_i$ is the normal (sequence-independent) processing time of job $J_i$. They show that the single-machine makespan problem remains polynomially solvable under the proposed model. Wu and Cheng[11] address that the actual processing time of a job is a decreasing function of the sum of processing times based learning or increasing function of the sum of processing times based deteriorating effect to solve a two-agent single-machine scheduling problem. Li and Fan[12] address the nonresumable version of the scheduling problem with proportionally deteriorating jobs subject to availability constraints to minimize the total weighted completion time. Wu and Huang [13] investigate an NP-hard problem of single-machine scheduling problem with deteriorating jobs and different due dates to minimize total tardiness, and propose EDA to solve it. Wang et al.[14] consider single-machine scheduling problem with deterioration jobs in which the actual processing time of a job is a function of its position in a sequence, its starting time, and its resource allocation. Lee[15] provide the optimal schedules for the problem of single-machine scheduling with past-sequence-dependent setup times and general effects of deterioration and learning. And, besides that, another scholars study other scheduling problem with variety of processing time. For example, Wang and Choi[16] considered makespan minimization of a flexible flow shop scheduling problem with stochastic processing times. Lee[17] studied the scheduling problem with learning effect and setup time. Lai and Lee[18] discussed the scheduling problems with learning effect and forgetting effect.

Due to the characteristics of time-dependent deterioration in the loading and unloading cargos process, having a rest should be also considered. Moreover, having a rest implies that it modifies a performance rate of the machine. In this paper, it is also regarded as a rate-modifying activity (RMA). Lee and Leon[19] first consider single machine scheduling with a rate-modifying activity for a problem commonly found in electronic assembly lines. They provided polynomial and pseudo-polynomial algorithms for a number of objective functions. Motivated by a RMA, Wang and Wang[20] suggest single machine SLK due date assignment scheduling problem with a rate-modifying activity. A RMA is introduced into a scheduling problem with job-dependent learning effects by Ji and Cheng[21] and parallel machine scheduling problem by Wang and Wei[22]. Rebai et al.[23] consider maintenance tasks the same as RMA assigned to multiple single machines. Lodree and Geiger[24] integrate time-dependent processing time/RMA framework into machine environments characterized by deteriorating machine performance and rate-modifying maintenance activities. They assume that the normal processing time of each job is equal to 1, the task actual processing times are represented as a variation of simple linear deterioration and a single machine always fully restores after a break time. Zhao and Tang[25] consider a single machine scheduling and due-window assignment problem, where the normal processing time is a fixed value and the actual processing time of a job is a linear function of its starting time and the job-independent deterioration rates are identical for all jobs. The objective is to schedule the jobs, the due-window and the rate-modifying activity so as to minimize the sum of earliness, tardiness and due-window starting time and due-window size costs. Öztürkoglu et al.[26] determined an optimal sequence with the optimal number and the positions of RMAs.

However, research on the detailed recovery processing has not been studied until now. To the best of our knowledge, this paper is the first study to propose a rational recovery function integrated into deterioration jobs, RMA, and human operators fatigued.

### III. Problem Formulation

The problem in this paper can be formally described as follows: There are $n$ independent jobs $J = \{J_1, J_2, \ldots, J_n\}$ to be processed non-preemptively on a single machine. Assume that all jobs are available at the same time, and have the same processing time if they were processed at the beginning of a sequence. Otherwise, in order to improve the efficiency of the machine, a resting time is inserted into a sequence. The problem is where and how long the resting time should be. The detailed notations are as follows:

- $n$: The number of independent jobs to be processed on a single machine;
- $J = \{J_1, J_2, \ldots, J_n\}$: The set of jobs;
- $p_i$: The normal processing time for every job ($p_i > 0$);
of a rest time considered, the actual processing time of job
\(p_r\): The actual processing time of job \(J_i\) (\(i \in J\)) if it is
scheduled in position \(r\) (\(1 \leq r \leq n\)) in a given sequence;
\(p[r]:\) The normal processing time for job scheduled in position \(r\), when \(J_i\) is scheduled in position \(r\), \(p_i = p[r]\);
\(b:\) The deterioration rate (\(b > 0\));
k: A resting time is inserted in the front of the \(k^{th}\) job
processed in a sequence (such as Figure 1);
\(T:\) The duration of resting for a single machine to be fully
restored;
t: The duration of a resting time (\(0 \leq t \leq T\));
\(C_i(k, t):\) The completion time of job \(J_i\) (\(1 \leq i \leq n\))
associated with scheduling an RMA in the \(k^{th}\) position of a
sequence, where \(1 < k \leq n\), because it is never optimal to
schedule the RMA in the first sequence position \(k = 1[24];\)
\(C_{\text{max}}(k, t):\) The makespan associated a resting time in the
position \(k\) (\(1 < k \leq n\)) of a sequence.

According to Wang et al.[10], if the resting time was not
considered, the actual processing time of job \(J_{ir}\) could be
described as a nonlinear function as follows:

\[
p_{ir} = p_i[1 + p[i] + \ldots + p[r-1]]^b, \quad 0 < r \leq n
\]

However, if the resting time was scheduled, the processing
times of jobs after resting will be affected by the duration
t of resting. Suppose that the machine will fully recover if
and only if \(t = T\) where \(T\) depends on the nature of the
problem. Then the recovery function can be defined.

**Definition 3.1.** The recovery function is defined as:

\[
F(t) = \frac{1}{t} p[r] [(1 + p[k] + \ldots + p[r-1])^b - (1 + p[1] + \ldots + p[r-1])^b]
\]

Where \(k \leq r \leq n\).

Obviously, it is a linear function of recovery time \(t\), that is:

1. If \(t = 0\), then \(F(t) = 0\). It implies that the machine
   has no chance for recovering capacity. Namely, there is no
   rest time set;
2. If \(t = T\), then \(F(t) = p[r] [(1 + p[k] + \ldots + p[r-1])^b - (1 + p[1] + \ldots + p[r-1])^b]\). It indicates that the
capacity of a machine can recover entirely when the length of
a rest time \(T\) is scheduled in the sequence;
3. When \(0 < t < T\), then \(F(t) = \frac{1}{t} p[r] [(1 + p[k] + \ldots + p[r-1])^b - (1 + p[1] + \ldots + p[r-1])^b]\). It describes the
amount of recovery that the machine captured when the length of
a rest time \((0 < t < T)\) is scheduled in the sequence. It is
clear that \(F(T) = F(t) = 0\).

Hence, according to the recovery function, when \(0 \leq t \leq T\), the actual processing time of job \(p_{ir}\) can be described as follows:

\[
p_{ir} = \begin{cases} 
  p_i[1 + p[i] + \ldots + p[r-1]]^b & 0 < r < k \\
  p_i[1 + p[i] + \ldots + p[r-1]]^b + F(t) & k \leq r \leq n 
\end{cases}
\]

Based on Equation (3), we have the following scenarios:

1. When \(t = 0\), \(p_{ir} = p_i(1 + p[i] + p[2] + \ldots + p[r-1])^b\). It denotes that a resting time is not inserted in a sequence and
   the actual processing times of jobs are increasing according
to the job sequence;
2. When \(0 < t < T\), \(p_{ir} = p_i(1 + p[i] + p[2] + \ldots + p[r-1])^b + F(t)\). It implies all the jobs after the resting time will be processed more
   efficiently;
3. When \(t = T\), \(p_{ir} = p_i(1 + p[k] + p[k+1] + \ldots + p[r-1])^b\). When the resting
duration is equal to \(T\), then the machine is fully restored and
the processing time of the first job after rest is equal to the
normal processing time of it.

So the problem is how to assign jobs, how to schedule
time, how long \(t\) should be and where \(t\) should be located to
optimize the given objectives.

In the following sections, we consider the problem of
minimizing the makespan with recovery function on a single
machine under nonlinear time-dependent deterioration. We
denote it as \(1|p_{ir}, rm, rp|C_{\text{max}}\) and \(1|p_{ir}, rm|\sum_i C_i\) by
using the three-field notation scheme \(\alpha|\beta|\gamma\) introduced by
Graham et al.[27], where represents the resting time or time
of maintenance that would modify a processing rate of a
machine, i.e. an RMA, and \(rp\) the recovery processing.

IV. THE PROBLEM OF \(1|p_{ir}, rm, rp|C_{\text{max}}\)

In this section, the problem of \(1|p_{ir}, rm, rp|C_{\text{max}}\) is
proved to be an NP-hard problem.

**Theorem 4.1.** The problem \(1|p_{ir}, rm, rp|C_{\text{max}}\) is
an NP-hard problem.

**Proof.** We transform the partition problem to our problem.
Partition problem

Instance. A finite set of positive integers \(x_i \in X\) \((i = 1, 2, \ldots, n)\) and \(A\) with \(\sum_{i=1}^{2n} x_i = 2A\), where \(\frac{A}{2^n} < x_i < \frac{4A}{2^n}\)
and \(A > \sqrt{2^n}\).

Question. Can this set \(X\) be partitioned into two disjoint
subsets \(X_1\) and \(X_2\) such that the sum of each subset is equal
to \(\sum_{x_i \in X_1} x_i = \sum_{x_i \in X_2} x_i = A\)?

We construct the following instance of the scheduling
problem. These are \(2n + 2\) jobs. Among them there are \(2n\)
partition jobs \(a_i\) \((i = 1, 2, \ldots, 2n)\), and enforcer jobs \(a_{1n}\)
and \(a_{2n}\) with the following parameters:

\[a_i = H + x_i, i = 1, 2, \ldots, 2n, x_i \in X\]

where \(H = A\), the duration time of an RMA \(t = 1\), and the
deterioration rate \(b = 2\).

Calculating \(y = 2nH + 3A + 6A(1 + A)^2 + W\), where
\[W = \sum_{x_i, x_j \in X \land i < j} (H + x_i)(H + x_j) + \sum_{x_i, x_j \in X \land i < j} (H + x_j)^2(H + x_j)\]

We show that the partition problem has a solution if and
only if there exists a solution to construct the instance of
\(1|p_{ir}, rm|C_{\text{max}}\) with value \(C_{\text{max}} \leq y\).

Only if \(\Rightarrow\)

Assume that a partition exists. Let \(X_1\) and \(X_2\) denote the
disjoint subsets, i.e., \(\sum_{x_i \in X_1} x_i = \sum_{x_i \in X_2} x_i = A\). There are

(Advance online publication: 14 November 2015)
\(l\) jobs in subset \(X_1\). Assign \(a_{e1}\) and \(a_{e2}\) as the last jobs in subsets \(X_1\) and \(X_2\), respectively. In such a schedule, we have

\[
C_{\text{max}} = a_{[1]} + a_{[2]}(1 + a[1])^2 + \ldots + a_{[i]}(1 + a[1] + \ldots + a[2])^2 + a_{e1}(1 + \sum_{i=1} a_i)^2 + 1 + a_{[i+2]} + a_{e2}(1 + \sum_{i=1} a_i)^2 \tag{5}
\]

\[= 2nH + 2A + W + 1 + 6A(1 + A)^2 < 2nH + 3A + W + 1 + 6A(1 + A)^2 \tag{4}
\]

Hence, \(C_{\text{max}} < 2nH + 3A + W + 1 + 6A(1 + A)^2 = y\) for \(i = 1, 2, \ldots, 2n\).

If \(\left(\Leftrightarrow\right)\)

Assume that the partition problem has no solution. Then, we have

\[
C_{\text{max}} = a_{[1]} + a_{[2]}(1 + a[1])^2 + \ldots + a_{[i]}(1 + a[1] + \ldots + a[2])^2 + a_{e1}(1 + \sum_{i=1} a_i)^2 + 1 + a_{[i+2]} + a_{e2}(1 + \sum_{i=1} a_i)^2 \tag{5}
\]

\[= 2nH + 2A + W + 1 + 3A(1 + \sum_{i=1} x_i)^2 + 3A(1 + \sum_{i=1} x_i)^2 \]

Suppose that \(\sum_{i=1} x_i = D(D \neq A)\), then \(D + D' = A\) and Equation (5) can be expressed as follows:

\[
C_{\text{max}} = 2nH + 2A + W + 1 + 3A(1 + D')^2 = y + A[(D')^2 - 1] + \frac{A(D-D')^2}{2} \tag{6}
\]

Since \(A > \sqrt{2}\), \(D^2 + D'^2 > 1\).

Therefore, \(C_{\text{max}} = y + A[(D^2 - 2D^2) - 1] + \frac{A(D-D')^2}{2} > y\).

Since the makespan is greater than the required value \(y\), our scheduling problem has no solution, which ends the proof.

Therefore, the partition problem has a solution if and only if there exist a solution to the construct instance of our scheduling problem has no solution, which ends the proof.

A. Preliminary Analysis

In a sequence, while inserting an RNA at position \(k\) in a sequence, the makespan can be expressed as follows:

\[
C_{\text{max}}(k, t) = p \sum_{s=0}^{n-k} (1 + sp)^b + \frac{t}{p} \sum_{s=0}^{n-k} (1 + sp)^b + t \left(1 - \frac{1}{p}\right) \sum_{s=0}^{n-k} (1 + sp)^b \tag{7}
\]

Here, \(p \sum_{s=0}^{n-k} (1 + sp)^b\) denotes the total processing times of the front \((k-1)\) jobs; \(t\) is a resting time; \(t \left(1 - \frac{1}{p}\right) \sum_{s=0}^{n-k} (1 + sp)^b\) describes the total processing times of the rest \((n-k+1)\) jobs. Clearly, the makespan is related to a position \(k\) and a resting time \(t\). The problem can be accomplished by determining the value of \(k\) and \(t\) to minimize Equation (7). To determine the value of \(k\) and \(t\), we propose the following propositions and theorems.

**Proposition 5.1.** For any \(k\) \((1 < k \leq n)\) and \(t\) \((0 < t \leq T)\), if \(C_{\text{max}}(k, t) < C_{\text{max}}(k-1, t)\), then \(C_{\text{max}}(k, t) < C_{\text{max}}(k-1, t) < \cdots < C_{\text{max}}(n, t)\); if \(C_{\text{max}}(k, t) < C_{\text{max}}(k-1, t)\), then \(C_{\text{max}}(k, t) < C_{\text{max}}(k-1, t) < \cdots < C_{\text{max}}(2, t)\).

**Proof.** For decided \(k\) and \(t\), the makespan can be expressed as follows:

\[
C_{\text{max}}(k, t) = p \sum_{s=0}^{k-2} (1 + sp)^b + \frac{t}{p} \sum_{s=0}^{n-k} (1 + sp)^b + t \left(1 - \frac{1}{p}\right) \sum_{s=0}^{n-k} (1 + sp)^b + t \left(1 - \frac{1}{p}\right) \sum_{s=0}^{n-k} (1 + sp)^b \tag{8}
\]

\[C_{\text{max}}(k+1, t) = p \sum_{s=0}^{k-1} (1 + sp)^b + \frac{t}{p} \sum_{s=0}^{n-k} (1 + sp)^b + t \left(1 - \frac{1}{p}\right) \sum_{s=0}^{n-k} (1 + sp)^b + t \left(1 - \frac{1}{p}\right) \sum_{s=0}^{n-k} (1 + sp)^b \tag{9}
\]

\[C_{\text{max}}(k+2, t) = p \sum_{s=0}^{k} (1 + sp)^b + \frac{t}{p} \sum_{s=0}^{n-k} (1 + sp)^b + t \left(1 - \frac{1}{p}\right) \sum_{s=0}^{n-k} (1 + sp)^b + t \left(1 - \frac{1}{p}\right) \sum_{s=0}^{n-k} (1 + sp)^b \tag{10}
\]

Using Equations (8), (9), and (10), we obtain:

\[
C_{\text{max}}(k+2, t) - C_{\text{max}}(k+1, t) = \frac{t}{p}[(1 + (n-k)p)^b - (1 + (k-1)p)^b] \quad \text{and} \quad C_{\text{max}}(k+1, t) - C_{\text{max}}(k+2, t) = \frac{t}{p}[(1 + (n-k-1)p)^b - (1 + kp)^b].
\]

If \(C_{\text{max}}(k, t) < C_{\text{max}}(k+1, t)\), we have

\[
\frac{t}{p}[(1 + (n-k)p)^b < \frac{t}{p}(1 + (k) - 1)p^b].
\]

Since \(T > 0\), \(b > 0\), \(p > 0\) and \(t > 0\), then

\[
\frac{t}{p}(1 + (n-k-1)p^b) < \frac{t}{p}(1 + (n-k)p^b) < \frac{t}{p}(1 + (k-1)p^b) < \frac{t}{p}(1 + kp^b), \quad \text{i.e.,}
\]

\[
\frac{t}{p}(1 + (n-k-1)p^b) < \frac{t}{p}(1 + kp^b) \quad \text{and} \quad C_{\text{max}}(k+1, t) < C_{\text{max}}(k+2, t).
\]

Similarly, we can obtain \(C_{\text{max}}(k, t) < C_{\text{max}}(k+1, t) < \cdots < C_{\text{max}}(n, t)\).
An analogous proof holds if $C_{\text{max}}(k, t) < C_{\text{max}}(k-1, t)$, then $C_{\text{max}}(k, t) < C_{\text{max}}(k-1, t) < \cdots < C_{\text{max}}(2, t)$.

This concludes the proof.

**Proposition 5.2.** If an RMA is scheduled, then $T < \min \{ p \sum_{s=k-1}^{n-1} [(1+s)p] \}$.

**Proof.** Through the previous description, the RMA should be assigned to position $k$ in a sequence only if the resulting makespan less than that of associating with not scheduling the RMA, the makespan associated with not scheduling an RMA is:

$$C_{\text{max}}(k, t = 0) = p + p(1 + b) + p(1 + 2b) + \cdots + p(1 + (n-1)p)$$

(11)

Therefore, the RMA is scheduled only if $C_{\text{max}}(k, t) < C_{\text{max}}(k, t = 0)$. Using Equations (7) and (11), $C_{\text{max}}(k, t) < C_{\text{max}}(k, t = 0)$ becomes

$$T < \min \{ p \sum_{s=k-1}^{n-1} [(1+s)p] \}.$$ 

Hence, the proposition is proved.

**Theorem 5.1.** For any $t (0 < t < T)$, the optimal policy for scheduling an RMA of length $t$ for all jobs is as follows: If $n$ is an even integer and $T < p \sum_{s=\frac{n}{2}}^{n-1} [(1+s)p]$, assign the RMA to sequence position $k = \frac{n}{2} + 1$; If $n$ is an odd integer and $T < p \sum_{s=\frac{n-1}{2}}^{n-1} [(1+s)p]$, assign the RMA to sequence position $k = \frac{n+1}{2}$ or $k = \frac{n+1}{2} + 1$. Otherwise, do not schedule the RMA.

**Proof.** (For an even $n$) According to Equation (7),

$$C_{\text{max}}(\frac{n}{2} + 1, t) = \frac{n}{2} \sum_{s=0}^{\frac{n}{2}-1} (1+s)p + t + \frac{1}{2}p \sum_{s=0}^{\frac{n}{2}} (1+s)p + t$$

(12)

$$C_{\text{max}}(\frac{n}{2} + 2, t) = \frac{n}{2} \sum_{s=0}^{\frac{n}{2}-2} (1+s)p + t + \frac{1}{2}p \sum_{s=0}^{\frac{n}{2}-1} (1+s)p + t$$

(13)

$$C_{\text{max}}(\frac{n}{2}, t) = \frac{n}{2} \sum_{s=0}^{\frac{n}{2}-2} (1+s)p + t + \frac{1}{2}p \sum_{s=0}^{\frac{n}{2}-1} (1+s)p + t$$

(14)

Since $T > 0$, $b > 0$, $p > 0$ and $t > 0$, then $C_{\text{max}}(\frac{n}{2} + 1, t) - C_{\text{max}}(\frac{n}{2} + 2, t) = \frac{1}{2}p[1 + (\frac{n}{2} - 1)p - (\frac{n}{2} + 1)p] < 0$.

By Proposition 5.1, we have

$$C_{\text{max}}(\frac{n}{2} + 1, t) < C_{\text{max}}(\frac{n}{2} + 2, t) < C_{\text{max}}(\frac{n}{2} + 3, t) < \cdots < C_{\text{max}}(n, t)$$

(15)

$$C_{\text{max}}(\frac{n}{2} + 1, t) < C_{\text{max}}(\frac{n}{2} + 2, t) < C_{\text{max}}(\frac{n}{2} - 1, t) < \cdots < C_{\text{max}}(2, t)$$

(16)

Equations (15) and (16) imply that the minimum makespan obtains when $k = \frac{n}{2} + 1$.

The proof for odd is analogous. This proof ends.

**Theorem 5.2.** For any $t (0 < t < T)$, $p \sum_{s=\frac{n}{2}}^{n-1} [(1+s)p]$ and $C_{\text{max}}(k, t)$, the optimal policy for scheduling an RMA of length $t$ for all jobs is as follows: If $n$ is an even integer, assign the RMA to position $k = \frac{n}{2} + 1$ in a sequence and the length $t = T$; If $n$ is an odd integer, assign the RMA to position $k = \frac{n+1}{2}$ or $k = \frac{n+1}{2} + 1$ in a sequence and the length $t = T$. Otherwise, do not schedule the RMA.

**Proof.** (For an even $n$) According to Equation (7), the makespan of scheduling an RMA in a sequence is:

$$C_{\text{max}}(k, t) = p \sum_{s=0}^{k-1} (1+s)p + t + \frac{1}{2}p \sum_{s=0}^{n-1} (1+s)p + t$$

$$= \{ 1 + \frac{1}{2}p \} \sum_{s=0}^{k-1} (1+s)p - \frac{1}{2}p \sum_{s=0}^{n-1} (1+s)p$$

(17)

Based on Theorem 5.1, for $0 < t < T$ and $T < p \sum_{s=\frac{n}{2}}^{n-1} [(1+s)p]$, we have $C_{\text{max}}(\frac{n}{2} + 1, t) < C_{\text{max}}(\frac{n}{2} + 1, t, k = 2, \cdots, n, and \ k \neq \frac{n}{2} + 1)$. So $C_{\text{max}}(\frac{n}{2} + 1, t)$ is minimal. Thus, the makespan of
scheduling the RMA in a sequence is represented as follows:

\[ C_{\max}(\frac{n}{2} + 1, t) = \{1 - \frac{1}{p} \sum_{s=\frac{n}{2}}^{n-1} [(1 + sp)^b - (1 + (s - \frac{n}{2})p)^b)\} + p \sum_{s=\frac{n}{2}}^{n-1} [(1 + sp)^b + (1 + (s - \frac{n}{2})p)^b] \]

(18)

Let \( K = 1 - \frac{1}{p} \sum_{s=\frac{n}{2}}^{n-1} [(1 + sp)^b - (1 + (s - \frac{n}{2})p)^b] \) and 

\[ B = p \sum_{s=\frac{n}{2}}^{n-1} [(1 + sp)^b + (1 + (s - \frac{n}{2})p)^b], \]

then \( C_{\max}(\frac{n}{2} + 1, t) = Kt + B. \)

Since \( b > 0, p > 0 \) and \( 0 < t < T < \)

\[ p \sum_{s=\frac{n}{2}}^{n-1} [(1 + sp)^b - (1 + (s - \frac{n}{2})p)^b], \]

then \( K < 0. \)

Obviously, \( C_{\max}(\frac{n}{2} + 1, t) \) is decreasing function when \( 0 < t < T. \) Therefore, the minimum of \( C_{\max}(k, t) \) is \( C_{\max}(\frac{n}{2} + 1, T). \)

The proof for odd \( n \) is analogous. This proof ends.

### B. Algorithm

Based on above propositions and theorems, we give the following algorithm. Let 

\[ U = p \sum_{s=\frac{n}{2}}^{n-1} [(1 + sp)^b - (1 + (s - \frac{n}{2})p)^b] \] and 

\[ V = p \sum_{s=\frac{n}{2}}^{n-1} [(1 + sp)^b + (1 + (s - \frac{n}{2})p)^b]. \]

The procedure of the algorithm for solving the problem 1|\( p_{ir}, rm, rp, p_i = p | C_{\max} \) is as follows:

Step 1. For an even integer \( n \), computer whether \( T \) is less than \( U \) or not. For an odd integer \( n \), computer whether \( T \) is less than \( V \) or not. If it is not, then do not schedule an RMA. Otherwise, move to Step 2.

Step 2. Assign the RMA in position \( (\frac{n}{2} + 1) \) and Computer \( C_{\max}(\frac{n}{2} + 1, T) \) (For an odd integer \( n \), assign the RMA in position \( (\frac{n+1}{2}) \) or \( (\frac{n+1}{2} + 1) \) and compute \( C_{\max}(\frac{n+1}{2} + 1, T) \), which is the minimal value of the makespan.

Because the time complexity of \( p \sum_{s=\frac{n}{2}}^{n-1} [(1 + sp)^b - (1 + (s - \frac{n}{2})p)^b] \) is \( O(\frac{b}{2}) \), clearly, the time complexity of Algorithm 1 is \( O(n) \).

### C. Numerical Experiments

Example. Through investigating a certain large airport in China, an example is given here. There are 6 airplanes \( (n = 6) \) which arrives at an airport at time 0. The normal processing time for each airplane is 35min \( (p = 35) \), the deterioration rate \( b \) and the fully recovery time \( T \) are respectively designed as different values to solve the problem, i.e., \( b = 0.054, 0.056, 0.059, 0.100, \) and \( T = 10 \text{ min}, 20 \text{ min}, 30 \text{ min}. \)

All airplanes are available for ramp services at 0 and \( n \) is an even integer equal to 6. According to Proposition 5.1, Proposition 5.2 and Theorem 5.1, when \( T < U \), an RMA will be scheduled.

According to the algorithm in 5.2, we have the following results of comparisons in Table 1, in which RMAC denotes it is considered whether an RMA is scheduled or not, and RMANC denotes no considering the RMA. Besides, the value of \( k \) is obtained by the algorithm in 5.2 and it is the position of scheduling an RMA. denotes that there is not an RMA. Yes indicates that an RMA should be scheduled for minimizing the makespan or the total completion time of jobs. And No is just contrary to it.

<table>
<thead>
<tr>
<th>T</th>
<th>b</th>
<th>( T &lt; U? )</th>
<th>k</th>
<th>( C_{\max} )</th>
<th>( C_{\max} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.054</td>
<td>Yes</td>
<td>4</td>
<td>253.06</td>
<td>258.55</td>
</tr>
<tr>
<td>0.056</td>
<td>Yes</td>
<td>4</td>
<td>254.42</td>
<td>260.58</td>
<td></td>
</tr>
<tr>
<td>0.059</td>
<td>Yes</td>
<td>4</td>
<td>256.49</td>
<td>263.68</td>
<td></td>
</tr>
<tr>
<td>0.100</td>
<td>Yes</td>
<td>4</td>
<td>287.37</td>
<td>310.59</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.054</td>
<td>No</td>
<td></td>
<td>258.55</td>
<td>258.55</td>
</tr>
<tr>
<td>0.056</td>
<td>No</td>
<td></td>
<td>260.58</td>
<td>260.58</td>
<td></td>
</tr>
<tr>
<td>0.059</td>
<td>No</td>
<td></td>
<td>263.68</td>
<td>263.68</td>
<td></td>
</tr>
<tr>
<td>0.100</td>
<td>Yes</td>
<td>4</td>
<td>297.37</td>
<td>310.59</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.054</td>
<td>No</td>
<td></td>
<td>258.55</td>
<td>258.55</td>
</tr>
<tr>
<td>0.056</td>
<td>No</td>
<td></td>
<td>260.58</td>
<td>260.58</td>
<td></td>
</tr>
<tr>
<td>0.059</td>
<td>No</td>
<td></td>
<td>263.68</td>
<td>263.68</td>
<td></td>
</tr>
<tr>
<td>0.100</td>
<td>Yes</td>
<td>4</td>
<td>307.37</td>
<td>310.59</td>
<td></td>
</tr>
</tbody>
</table>

Analyses. From Table 1 we can find that:

1. Scheduling an RMA in a sequence can decrease the makespan. On the other hand, the makespan will increase with the deteriorating rate \( b \). This is because that the processing time of jobs become increasing if the labors are easily tired.

2. The value of \( T \) has an effect on whether an RMA is scheduled or not.
   a) When \( T < U \) for the minimal makespan problem, an RMA should be scheduled. Under these situations, for a certain deteriorating rate \( b \), he longer the fully recovery time \( T \) of a group labors needs, the larger the objective value is. However, compared with no RMA scheduled situation, it will decrease.
   b) When \( T > U \) for the minimal makespan problem, an RMA should not be scheduled because scheduling with a shorter RMA has weakly impact on the recovery of workers and scheduling with a longer recovery time will make the objective value exceeding the one under the situation without an RMA.
   c) While \( T = U \) for the minimal makespan problem, the schedule assigned with an RMA is the same as that without an RMA. In this case, scheduling an RMA is encouraged considering the workers healthy.

### VI. Conclusions

This paper firstly considers a single machine scheduling problem with human energy recovery function under nonlinear time-dependent deterioration. Firstly, a recovery function is firstly proposed. And the problem is proved to be an
NP-hard problem. Then, the special case of the problem is considered and solved in polynomial time. It provides a scheduling framework for other practical scheduling problems referring to labors including maintenance scheduling and cleaning assignments. In the future, the research will be extended to the problem with release dates and due dates, which is the more general situation. In this situation, the scheduling problems need to be solved by heuristic algorithm or intelligent optimization algorithm such as random search algorithm[28].

ACKNOWLEDGMENTS

The authors would like to thank the reviewer for their insightful comments. addressing them has improved the quality of the paper.

REFERENCES


