

Propose a Polynomial Time Algorithm for Total Completion Time Objective

Yucel Ozturkoglu

Logistics Management,
Yasar University, Izmir, Turkey.

Corresponding author: yucel.ozturkoglu@yasar.edu.tr

Omer Ozturkoglu

Business Administration,
Yasar University, Izmir, Turkey.
E-mail: omer.ozturkoglu@yasar.edu.tr

(Received on September 11, 2020; Accepted on February 6, 2021)

Abstract

In this study, we integrate deteriorate jobs with repair&maintenance activity on a single machine scheduling subject to total completion time. This work has more than one motivation. First, jobs are assigned to machines in an automated production line. Later, to schedule the maintenance activities, if needed, to prevent machinery from breaking down later. There are some important mathematical models to solve this combination. However, due to the complexity of the problem which is Np-hard, a polynomial algorithm should be needed for solving large problems. Therefore, this article introduces several polynomial algorithms to determine the order of things best. With using these algorithms, it will be possible to determine where to assign to the schedule, taking into account the number of maintenance activities required and their optimum total completion time.

Keywords- Deteriorated job, Rate-modifying-activity, Scheduling, Total completion time.

1. Introduction

The competition among companies in the same sector and industry has become vital due to the recession in the national and the international economy. Today, besides production cost, the ability to respond quickly to customer needs is the basis for success in the market. Systems must first be more flexible in order to develop this skill. Flexible systems are linked to the completion times of the products. So, this leads them to redesign their processes and jobs. Hence, reducing the completion time of the product becomes one of the most important factors in job design. In this study, to be able to reduce the completion time of the products, we focus on the sequencing and scheduling of jobs, as well as some benefit activities.

In both manufacturing and service industries, jobs might have variable processing times. Delaying a job may result in additional time to complete it, such as cleaning dirty dishes the longer they wait, the harder they are to clean. These kinds of jobs are known as deteriorating jobs in the literature; if processed later they take more time than when processed earlier. In other words, deteriorating jobs need more effort and time to finish the process than when they are done earlier. The studies started for the first time at the end of 1980's to schedule the deteriorating jobs (Gupta and Gupta, 1988; Browne & Yechiali, 1990). Some examples of deteriorating jobs are searching for an enemy under growing darkness, firefighting, treating a patient under worsening health conditions, cleaning, resource allocation, refurbishing operations, like polishing steel rust or producing steel under decreasing oven temperature (Kunnathur and Gupta, 1990; Mosheiov, 1995; Ozturkoglu, 2013;

Cheng et al., 2020). In the scheduling literature, many researchers from different perspectives have studied deteriorating job. In almost all studies, their processing time ignores the time-dependent issue. Therefore, there is a big gap in the scheduling literature in changing the processing time of a job with different methods according to time.

If jobs deteriorate, there is an activity called a rate-modifying activity (RMA) which recovers the lost time of processing for a job because of the deterioration. RMA, which changes the production rate of a processor such as machines and workers (Lee and Lin, 2001; Zhu et al., 2017). In the literature, rate-modifying activity is called a repair or maintenance activity of a machines. So, when RMA is taken, there is some improvement on processing of that job, i.e., its processing time decreases.

Although there are many studies that work independently of these two important events in the scheduling literature, there are very few studies that deal with both at the same time. One of the first studies that merge deteriorated jobs and single maintenance activity at the same time is proved that only one maintenance activity should be assigned to center of the optimal sequence with the makespan objectives (Lodree and Geiger, 2010). Then, one study developed a mathematical model for multiple maintenance activity with minimizing makespan & total completion time objectives (Ozturkoglu and Bulfin, 2011). Their model can be solved up to fifteen jobs. Another studies extended the same problem with a general job dependent deterioration effect (Kim and Ozturkoglu, 2013; Yang and Yang, 2013). They also proposed polynomial time solutions to the makespan minimization problem. Due to the complexity environment of the problem Ozturkoglu (2015) proposed an algorithm for large number of job sets. For the same problem with identical parallel machines, Ozturkoglu (2017) developed an algorithm for makespan objectives with the optimal sequence which is consist of non-linearly deteriorating jobs. Their algorithm also determined the sequences of single and multiple RMAs. The completion time objectives of the problem is still opened (Tasgetiren et al., 2016).

The main aim of this paper is to develop different algorithms to fill the aforementioned gap. Therefore, in this study, we deal with similar problem of scheduling a set of deteriorating jobs with one/multiple repair and maintenance activities on a single machine with only focus on total completion time objective. Firstly, four propositions containing different features of the mentioned problem have been proved and different solution algorithms have been proposed. In addition, we propose three different polynomial time algorithms, which reflect real industrial applications which the processing time of a job decreases or increases according to maintenance time or breaking time for workers. All of the developed algorithms give optimal results.

The remainder of the paper consist of four section. In the next section, the problem definition which is the main subject of this study, and the variables used are given. In section 3, algorithms that give optimal results with four proven propositions are presented with total completion time objective. In Section 4, a numerical example is presented to show the accuracy and validity of the developed algorithms. Lastly, the study is concluded and the future work idea is given.

2. Problem Formulation and Parameters

This study is focused on one of the important scheduling problem. In the problem, we have n number of jobs which is waiting to assign on one machine. These scheduled jobs are affected by the deterioration rate (α) after a certain waiting period. Different jobs will have the same effect due to the same rate of distortion. One of the main assumption is the job changes the processing time

nonlinearly according to its location. The main problem in this study is to determine how many maintenance activities will be used. It is to find out where to add these in the program and set the schedules for all jobs.

Model Parameters:

The model has some important parameters which are shown in below;

N : number # of jobs,

i : # of position (from one to n)

k : # of position (from zero to n)

j : # of the job (from one to n)

α : rate of the jobs deterioration

q : duration of repair&maintenance activity

p_j : j 's processing time before deterioration

p_{ji} : j 's processing time if repair& maintenance is done in position i

Equation of p_{ij} is;

$$p_{ij} = (1 + \alpha_j)^{i-1} p_j \tag{1}$$

The problem can be showed with using Graham et al. (1979) notation;

$$1 | p_{ij} = (1 + \alpha_j)^{i-1} p_j, rm | \sum_{i=1}^n C_i \tag{2}$$

3. Total Completion Time Objective

Because of the NP-Hard complexity of this problem, the mathematical models can not solve large-size problems. Therefore, effective heuristic procedures need to be developed. In this section, we proposed several heuristic procedures under different cases for our problem.

3.1 Solution for Position-Based Deterioration of Jobs with/out RMA

In this part, we prove two fundamental properties of the mentioned scheduling problem with the objective of total completion time. The first theorem proves optimal scheduling with position based deterioration jobs by using the shortest processing time(SPT) rule. This kind of rules find the actual processing time of a job. But this jobs have special features which are depending on the position of the job sequenced without maintenance activity. The second theorem shows that sequencing minimizes the total completion time on a single machine problem with only use one activity of the maintenance & repair.

Theorem 1. *SPT sequencing minimizes the total completion time on single machine problem*

$1 | p_{ij} = (1 + \alpha_j)^{i-1} p_j | \sum_{i=1}^n C_i$ with non-decreasing order of p_{ij} without using any maintenance activity.

Proof. A pairwise job interchange argument is done for this theorem. Lets assume two jobs sequence, $S = (\pi, i, j, \pi')$ and $S' = (\pi, i, j, \pi')$ where, π and π' denote partial sequences.

Another assumption is; there are $k - 1$ jobs in π . Thus jobs i and job j are the k^{th} and $(k + 1)^{th}$ in S and jobs j and job i are the k^{th} and $(k + 1)^{th}$ in S' . Also $C = (B)$ represents the completion time of the last job in π . To proof S dominate S' , it is adequate to show that the $(k + 1)^{th}$ jobs in

S and S' satisfy the conditions that

- i) $C_j(S) < C_i(S')$
- ii) $C_i(S) + C_j(S) \leq C_j(S') + C_i(S')$

By the definition, the initial processing times of jobs j & job i in S are given by;

$$C_i(S) = C(B) + (1 + \alpha)^{k-1} p_i \tag{3}$$

and

$$C_j(S) = C(B) + (1 + \alpha)^{k-1} p_i + (1 + \alpha)^k p_j \tag{4}$$

The initial processing times of jobs j & job i in S' are given by;

$$C_j(S') = C(B) + (1 + \alpha)^{k-1} p_j \tag{5}$$

and

$$C_i(S') = C(B) + (1 + \alpha)^{k-1} p_j + (1 + \alpha)^k p_i \tag{6}$$

Since, $p_i < p_j$ and we have $C_j(S) < C_i(S')$ from Theorem 1. Thus, we only need to show that $C_j(S) < C_i(S) + C_j(S) \leq C_j(S') + C_i(S')$.

To calculate

$$[(C_j(S') + C_i(S')) - (C_i(S) + C_j(S))]$$

we use equations (3)-(4).

Then we obtain;

$$\begin{aligned} \Rightarrow (1 + \alpha)^{k-1} 2(p_i - p_j) &\leq (1 + \alpha)^k (p_i - p_j) \\ \Rightarrow 2(1 + \alpha)^{k-1} &\geq (1 + \alpha)^k \\ \Rightarrow 2(1 + \alpha)^{k-1} &\geq (1 + \alpha)^k (1 + \alpha)^k \\ \Rightarrow 2 &\geq (1 + \alpha)^k \end{aligned}$$

Since, $p_i < p_j$ and $0 \leq \alpha \leq 1$ the statement of $C_i(S) + C_j(S) \leq C_j(S') + C_i(S')$ is satisfy in this case. Therefore, S dominates S'. To repeat this change for each job which are not sorted based on the SPT.

Theorem 2. $1|p_{ij} = (1 + \alpha_j)^{i-1} p_j, rm| \sum_{i=1}^n C_i$ an optimal schedule is \wedge -shaped on single machine problem with nonlinear job deterioration with one maintenance activity.

Proof. \wedge -shaped schedule is applied when many jobs are arranged in a non-decreasing order of processing times. This schedule is consist of the remaining jobs arranged in a non-increasing order of processing times.

If S is a sequence with three different jobs (i, j, k) and their processing times are $p_j > p_i$ and $p_j > p_k$. Therefore, S is never optimal. Let S₁ be the sequence of three different jobs (i, j, k) T is the starting

time of job in the sequence and m is the repair & maintenance time (rma) of the jobs. The position of job i in the sequence S and S_1 by $r-1$.

The total completion time for S is:

$$\begin{aligned} C(S)_i &= T + p_i(1 + \alpha)^{k-1} \\ C(S)_j &= T + p_j(1 + \alpha)^{k-1} + m + p_j \\ C(S)_k &= T + p_i(1 + \alpha)^{k-1} + m + p_j + p_k(1 + \alpha)^{k+1} \\ \sum C(S)_K &= 3T + 3p_i(1 + \alpha)^{k-1} + 2m + 2p_j + p_k(1 + \alpha)^{k+1} \end{aligned} \quad (7)$$

and the total completion time for S_1 is:

$$\begin{aligned} C(S)_j &= T + p_j(1 + \alpha)^{k-1} \\ C(S)_i &= T + p_j(1 + \alpha)^{k-1} + m + p_i \\ C(S)_k &= T + p_j(1 + \alpha)^{k-1} + m + p_i + p_k(1 + \alpha)^{k+1} \\ \sum C(S)_1 &= 3T + 3p_j(1 + \alpha)^{k-1} + 2m + 2p_i + p_k(1 + \alpha)^{k+1} \end{aligned} \quad (8)$$

Subtracting equation (7) to (8) we get;

$$\begin{aligned} \Delta C(S) - \Delta C(S_1) &= 3(1 + \alpha)^{k-1}(p_i - p_j) + 2(p_j - p_i) \\ &= (p_j - p_i) [2 - 3(1 + \alpha)^{k-1}] \end{aligned}$$

If the sequence S is better than S_1 , the value of $\Delta C(S) - \Delta C(S_1)$ should be negative. One important note; $k = 1$ is impossible because a job is scheduled in position $k-1$.

For $k = 2$, recall that $(p_j - p_i)$ is positive by assumption. Thus, from second part of equation (8), $[2 - 3(1 + \alpha)^{k-1}]$ should be negative.

- $2 - 3(1 + \alpha)^{k-1} < 0$
- $\frac{2}{3} < 1 + \alpha$.

Thus, this inequality is always true for any value of α between $0 < \alpha < 1$. This means that the sequence with Λ -shaped is always optimal when we have job position deterioration with given only one maintenance activity. Some of the large jobs are always given after the maintenance activity and this activity is always given at the center of the position before the largest jobs.

3.2 Polynomial Time Algorithm for Total Flow Time

In this part of the study, due to the NP-Hard complexity of this problem, two polynomial time algorithms have been proposed for the solution of large jobs. Regardless of the size of the problem with the developed algorithm, it gives optimal results.

Algorithm 1. This algorithm will find an optimal solution for the problem $1|p_{ij} = (1 + \alpha_j)^{i-1} p, rm | \sum_{i=1}^n C_i$ with time complexity $O(n \log n)$.

Proof. As mentioned before n is # of jobs and k is # of determined maintenance activities and q is the duration of maintenance activities. Also, r is the remainder of $\frac{n}{k+1}$ and d is the division of $\frac{n}{k+1}$. B is a coefficient of the equation and every term has own coefficient. So, it will be changed depend on processing times and given # of maintenance activity. Therefore, we find B based on division

(d) of $\frac{n}{k+1}$.

$$B = \sum_{i=1}^k id \tag{9}$$

To find the optimal total completion we have two different ways based on remainder of the $\frac{n}{k+1}$.

Definition 1. A schedule with k_i is a balanced schedule;

$$k_i: \begin{cases} id & i: 1, \dots, m+1-r \\ id + [i - (m+1-r)] & i: r+1, \dots, m. \end{cases}$$

Therefore, any types of the sequence which has either r groups & $d+1$ jobs or $m+1-r$ groups & d jobs is called as balanced schedule.

Theorem 3. Based on the Definition1, balanced schedule is optimal under $r:0$ condition, if $k_i: id, i:1, \dots, m$.

$$\begin{aligned} \sum_{i:1}^n C_i = & p + \sum_{i:1}^{kj} p(1+\alpha)^{i-1} + C_{i-1} + p + \sum_{i:1}^{kl} p(1+\alpha)^{i-1} + C_{i-1} + \dots + p \\ & + \sum_{i:1}^{n-(k_1+k_k)} p(1+\alpha)^{i-1} + mq. \end{aligned}$$

Proof. If mq is constant, we should choose k_1, k_2, k_m to minimize $\sum_{i:1}^n C_i$. This occurs when $k_1, k_2, k_m, n-k_m$ are as close to equal as possible. Let $n=d(m+1)+r$ and d is the integer.

Assume, S is an optimal schedule with $k_j: d+1$ and $k_j: d-1$ and S' has the same schedule except $k_j: d$ and $k_i: d$. All terms of $\sum_{i:1}^n C_i$ for both S and S' are equal except those containing k_j and k_i .

$$\begin{aligned} \sum C(S) = & p + \sum_{i:1}^{kj} p(1+\alpha)^{i-1} + C_{i-1} + p + \sum_{i:1}^{kl} p(1+\alpha)^{i-1} + C_{i-1} + q \\ \sum C(S') = & p + \sum_{i:1}^{kj} p(1+\alpha)^{i-1} + C_{i-1} + p + \sum_{i:1}^{kl} p(1+\alpha)^{i-1} + C_{i-1} + q \\ C(S) - C(S') = & (C_d - C_{d-1}) + [p(1+\alpha)^d - p(1+\alpha)^{d-1}] \end{aligned}$$

$C_{\max}(S) - C_{\max}(S') \geq 0$. Therefore, S is better than S' and contradicting S is optimal.

If solution S has at least two different k_i than d or has a $k_i > d+1$ or $k_i < d-1$, it should use the same work to show it is not as good as solution S .

Theorem 4. If the remainder is zero, $r:0$;

Then the total completion time is calculated by given formula;

$$\sum_{i:1}^n C_i = Bq + \sum_{i:0}^{d-1} [(k+1)(d-i) + B] p(1+\alpha)^i \tag{10}$$

Theorem 5. If the remainder is not zero, then use the Algorithm 2.

Algorithm 2

Step 1. Calculate r and d based on $\frac{n}{k+1}$

Step 2. Write basic equation until term $p(1 + \alpha)^{d-1}$ which is given below:

$$xp + yp(1 + \alpha) + \dots + zp(1 + \alpha)^{d-1} \tag{11}$$

{ x, y and z are coefficients }

Step 3. Add basic equation to q (rma time) and obtain new equation.

$$xp + yp(1 + \alpha) + \dots + zp(1 + \alpha)^{d-1} + sq \tag{12}$$

Step 4. Rewrite equation (3.11) $(k - 1)^{th}$ times.

And obtain new equation.

$$x_1p + y_1p(1 + \alpha) + \dots + z_1p(1 + \alpha)^{d-1} + s_1q + x_2p + y_2p(1 + \alpha) + \dots + z_2p(1 + \alpha)^{d-1} + s_2q + \dots + x_kp + y_kp(1 + \alpha) + \dots + z_kp(1 + \alpha)^{d-1} \tag{13}$$

Step 5. Before term $p(1 + \alpha)^{d-1}$, add term $p(1 + \alpha)^d$ to equation (13) r^{th} times. Start to add at the end of the equation.

Step 6. Determine coefficients.

Start with n and every time count down one. Except for first p, coefficient of q has always the same coefficient of p.

$$np + (n - 1)p(1 + \alpha) + \dots + (n - mq) + (n - mp).$$

Step 7. Organized all terms and obtain main equation;

$$\sum_{i:1}^n C_i = \sum p + \sum p(1 + \alpha) + \dots + \sum p(1 + \alpha)^d + \sum q$$

Step 8. Equation check.

In the last equation we should have;

- $(k + 1)^{th}$ times $xp + yp(1 + \alpha) + \dots + p(1 + \alpha)^{d-1}$
- $(k)^{th}$ times q
- $(r)^{th}$ times $p(1 + \alpha)^d$
- $\sum_{i:1}^1 p(1 + \alpha)^{i-1} - \sum_{i:2}^2 p(1 + \alpha)^{i-1} = \sum_{i:m}^m p(1 + \alpha)^{i-1} - \sum_{i:m+1}^{m+1} p(1 + \alpha)^{i-1} = k + 1$
- $\sum_{i:1}^1 p(1 + \alpha)^{i-1} - \sum q = n$

In light of the above mentioned features, Algorithm 3 is the general equation for total completion time objective with consider both deteriorating job and maintenance & repair activities simultaneously.

The general algorithm for Total Completion Time is given below;

Algorithm 3

Step 1. Determine n , α and q .

Step 2. Find division (d) of $n/(k+1)$.

Step 3. Calculate remainder (r) of $n/(k+1)$.

If $r = 0$, then go to Step 4,

Or, go through Step 6.

Step 4. Calculate B ;

$$B = \sum_{i:1}^k id$$

Step 5. Calculate total completion time by given formula;

$$\sum_{i:1}^n C_i = Bq + \sum_{i:0}^{d-1} [(k+1)(d-i) + B] p(1+\alpha)^i$$

Step 6. If $r \neq 0$,

Write basic equation until term $p(1+\alpha)^{d-1}$.

Step 7. Add basic equation to term q .

Step 8. Rewrite equation on Step 7, $(k-1)^{th}$ times. And obtain new equation

Step 9. Before term $p(1+\alpha)^{d-1}$, add term $p(1+\alpha)^d$ to new equation r^{th} times. Start to add at the end of the equation.

Step 10. Determine coefficients. Start with n and every time count down one. Except for first p , coefficient of q has always the same coefficient of p .

Step 11. Obtain main equation and calculate total flow time

Step 12. For each possible repair&maintenance assignment, $k = 2, \dots, (n-1)$ repeat step 2 to 11.

Step 13. For each possible rma, take minimum $\sum_{i:1}^n C_i$. This value is the optimal solution.

Except this algorithm, if we want to determine a constant number of repair activities, propose equation is used to find the total completion time.

Equation: The general equation for m exact numbers of maintenance activity

$$\sum_{i:1}^n C = m \sum_{i:1}^d (d+1-i) p(1+\alpha)^{i-1} + mk \sum_{i:1}^d p(1+\alpha)^{i-1} + mk \sum_{i:1}^{n-kd} p(1+\alpha)^{i-1} + \sum_{i:1}^{n-kd} p(1+\alpha)^{i-1} + dq + q \sum_{i:2}^m (n-kd)i.$$

As mentioned above, we proposed different algorithms with different cases of our unique scheduling problem (see Table 1).

Table 1. Propose algorithms for different scheduling problems.

Problem	Method	# rma
$1 p_{ij} = (1 + \alpha_j)^{i-1} p_j \sum_{i=1}^n C_i$	SPT	-
$1 p_{ij} = (1 + \alpha_j)^{i-1} p_j, rm \sum_{i=1}^n C_i$	Poly. solvable $O(n \log n)$ time	multiple
$1 p_{ij} = (1 + \alpha_j)^{i-1} p_j \sum_{i=1}^n C_i$	λ -shaped	one
$1 p_{ij} = (1 + \alpha_j)^{i-1} p_j, rm \sum_{i=1}^n C_i$	Poly. solvable $O(n \log n)$ time	m exact number

4. Numerical Example

In this part, an example is demonstrated to apply proposed algorithms.

Example. Let us consider the following example which concerns 10 jobs with unit processing times; $n:10, k:3, p:3, \alpha:0.9$ and $q:10$.

Step 1. Calculate (d) of $n/(k+1)$, we get $r \neq 0$ and $d:2$
 { $d:2$ means my last term will be $p(1 + \alpha)^2$ }.

Step 2. Write basic equation until $p(1 + \alpha)^{d-1}$ { x and y are coefficients }
 $xp + yp(1 + \alpha)$
 Step 3. Add sq { w is coefficient }
 $xp + yp(1 + \alpha) + sq$.

Step 4. Rewrite equation (11) $(k-1)^{th}$ times. And obtain new equation;
 $x_1p + y_1p(1 + \alpha) + s_1q +$
 $x_2p + y_2p(1 + \alpha) + s_2q +$
 $x_3p + y_3p(1 + \alpha) + s_3q$.

Step 5. Before term $(1 + \alpha)^{d-1}$, add term $p(1 + \alpha)^d$ to equation (12) $(r)^{th}$ times. Start to add at the end of the equation. 2^{nd} times ($r:2$), $wp(1 + \alpha)^2$ { w is coefficient }

Step 6. Determine coefficients;
 $10p + 9p(1 + \alpha) + 8q + 8p + 7p(1 + \alpha) + 6q +$
 $6p + 5p(1 + \alpha) + 4p(1 + \alpha)^2 + 3q + 3p + 2p(1 + \alpha) + 1p(1 + \alpha)^2$.

Step 7. Organized all terms and obtain main equation;

$$\sum_{i=1}^n C_i = 27p + 23p(1 + \alpha) + 5p(1 + \alpha)^2 + 17q$$

Step 8. Equation check;
 ➤ 4 times $(k + 1)^{th}$, $xp + yp(1 + \alpha) + \dots + p(1 + \alpha)^{d-1}$
 ➤ $(3)^{rd}$ times q
 ➤ $(2)^{th}$ times $p(1 + \alpha)^d$.

$$\begin{aligned}
 & \triangleright \sum_{i:1}^1 p(1 + \alpha)^{i-1} - \sum_{i:2}^2 p(1 + \alpha)^{i-1} \\
 & = \sum_{i:m}^m p(1 + \alpha)^{i-1} - \sum_{i:m+1}^{m+1} p(1 + \alpha)^{i-1} \\
 & = k + 1 = 4 = 27 - 23. \\
 & \triangleright \sum_{i:1}^1 p(1 + \alpha)^{i-1} - \sum q = n.
 \end{aligned}$$

So, we get given equation for total flow time;

$$\sum_{i:1}^n C_i = 27p + 23p(1 + \alpha) + 5p(1 + \alpha)^2 + 17q.$$

When we put all parameters in the equation we get,

$$\sum_{i:1}^n C_i: 27 * 3 + 23 * 3(1,9) + 5 * 3(1,9)^2 + 17 * 10.$$

$$\sum_{i:1}^n C_i: 436,25 \text{ (optimalvalue)}.$$

As seen in the sample case, although a scheduling consisting of ten jobs takes a long time in the computer environment in today's conditions, optimum results can be obtained in a much shorter time with the operations performed in the proposed model. Due to the proposed model, it can be easily used both in time and in places where production planning is made, with easy access to optimum results.

5. Conclusions

The motivation of this study is based on one of the main problems faced by one or more machines in the automatic production line. The production process, which lasts for three shifts and twenty-four hours, is stopped in activities such as parts replacement, maintenance and repair. This waiting time increases the completion time for jobs that are in the queue and tend to break down. During the scheduling period, the maintenance and repair times of the machines should be planned in advance according to the optimum performance criteria.

This study focuses on scheduling of both repair&maintenance times and jobs. In real life applications, the sizes of the problems are very large. Therefore, previously developed mathematical models do not obtain solutions after a certain job number. Also, the complexity of the problem is NP-hard, we proposed several heuristic algorithms for completion time objective's to solve largescale problems in a very short time.

We have proved that the deterioration of position-based processing job is solved by SPT without the maintenance activity. We found that the λ -shaped algorithm gave the optimal result when we solved the deterioration of position-based processing job by simply adding only one maintenance activity with minimizing total completion time objectives. Also, we developed two algorithms which can be solved in time to solve multiple maintenance activity with deteriorated processing jobs. At the end, we propose a general equation to minimize total completion time objectives with combining deteriorated jobs and maintenance activities. With each algorithm proposed in this study, the jobs are first scheduled and then the location and amount of the maintenance activity are

determined.

The developed algorithms are especially useful for production engineers. With these algorithms, regardless of the size of the number of jobs, scheduling of jobs with long processing times will be done easily. In addition, the answer to when and after which work should be done to prevent the prolongation of work times can be found thanks to these algorithms. For future work, the same problem can be extended with different objective functions.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors would like to thank the editorial board and anonymous referees for the valuable, constructive comments and suggestions on an earlier version of this paper.

References

- Browne, S., & Yechiali, U. (1990). Scheduling deteriorating jobs on a single processor. *Operations Research Society*, 38(3), 495-498.
- Cheng, T.C.E., Kravchenko, S.A., & Lin, B.M. (2020). Scheduling step-deteriorating jobs to minimize the total completion time. *Computers & Industrial Engineering*, 144, 106329.
- Graham, R.L., Lawler, E.L., Lenstra, J.K., & Kan, A.R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5, 287-326.
- Gupta, J.N., & Gupta, S.K. (1988). Single facility scheduling with nonlinear processing times. *Computers & Industrial Engineering*, 14(4), 387-393.
- Kim, B.S., & Ozturkoglu, Y. (2013). Scheduling a single machine with multiple preventive maintenance activities and position-based deteriorations using genetic algorithms. *The International Journal of Advanced Manufacturing Technology*, 67(5-8), 1127-1137.
- Kunnathur, A.S., & Gupta, S.K. (1990). Minimizing the makespan with late start penalties added to processing times in a single facility scheduling problem. *European Journal of Operational Research*, 47(1), 56-64.
- Lee, C.Y., & Lin, C.S. (2001). Single-machine scheduling with maintenance and repair rate-modifying activities. *European Journal of Operations Research*, 135(3), 493-513.
- Lodree Jr, E.J., & Geiger, C.D. (2010). A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration. *European Journal of Operational Research*, 201(2), 644-648.
- Mosheiov, G. (1995). Scheduling jobs with step-deterioration; minimizing makespan on a single-and multi-machine. *Computers & Industrial Engineering*, 28(4), 869-879.
- Ozturkoglu, O. (2017). Identical parallel machine scheduling with nonlinear deterioration and multiple rate modifying activities. *An International Journal of Optimization and Control: Theories & Applications*, 7(2), 167-176.
- Ozturkoglu, Y. (2013). A bi-criteria single machine scheduling with rate-modifying-activity. *Gazi University Journal of Science*, 26(1), 97-106.
- Ozturkoglu, Y. (2015). An efficient time algorithm for makespan objectives. *An International Journal of Optimization and Control: Theories & Applications*, 5(2), 75-80.

- Ozturkoglu, Y., & Bulfin, R.L. (2011). A unique integer mathematical model for scheduling deteriorating jobs with rate modifying-activities on a single machine. *International Journal of Advanced Manufacturing Technology*, 57(5-8), 753-762.
- Tasgetiren, M.F., Pan, Q.K., Ozturkoglu, Y., & Chen, A.H. (2016, July). A memetic algorithm with a variable block insertion heuristic for single machine total weighted tardiness problem with sequence dependent setup times. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2911-2918). IEEE. Vancouver, Canada.
- Yang, S.J., & Yang, D.L. (2013). Note on :a unique integer mathematical model for scheduling deteriorating jobs with rate-modifying activities on a single machine. *The International Journal of Advanced Manufacturing Technology*, 64(9-12), 1759-1764.
- Zhu, Z., Zheng, F., & Chu, C. (2017). Multitasking scheduling problems with a rate-modifying activity. *International Journal of Production Research*, 55(1), 296-312.



Original content of this work is copyright © International Journal of Mathematical, Engineering and Management Sciences. Uses under the Creative Commons Attribution 4.0 International (CC BY 4.0) license at <https://creativecommons.org/licenses/by/4.0/>