NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 2025; 14(2), 607-614

Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences

Araştırma makalesi / Research article

www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

Job rejection scheduling problems with deterioration and learning effects

Bozulma ve öğrenme etkisi altında iş reddetmeli çizelgeleme problemi

Berrin Atalay[1,*], M. Duran Toksarı[2]

[1,2]*Erciyes University, Industrial Engineering Department, 32280, Kayseri, Türkiye*

Abstract

In our study, deterioration and the learning effect in the job rejection scheduling environment was considered together. The processing time of the jobs will decrease with the effect of learning, but will also increase with the effect of deterioration. In our study, this situation was examined for four different objectives. These problems are total completion time minimization, makespan minimization and TADC minimization for single machine scheduling problems and parallel machine scheduling problems with makespan minimization. With three different learning rates (0.8, 0.7, 0.6) and two different deterioration rates (0.2, 0.1), 10 jobs, 20 jobs and 30 jobs problems are solved with mathematical models and algorithms and a comparison table is presented.

Keywords: Learning effect, Deterioration effect, Job rejection, Single machine, Parallel machines.

Öz

Çalışmamızda, iş reddetmeli çizelgeleme ortamında bozulma ve öğrenme etkisi birlikte ele alınmıştır. İşlerin işlem süresi öğrenme etkisiyle azalırken, bozulma etkisiyle de artacaktır. Bu durum, çalışmamızda dört farklı amaç doğrultusunda incelenmiştir. Ele alınan problemler, tek makineli çizelgeleme problemleri için toplam tamamlanma süresinin en aza indirilmesi, çevrim süresinin (makespan) en aza indirilmesi ve TADC minimizasyonu ile paralel makineli çizelgeleme problemleri için çevrim süresinin en aza indirilmesidir. Üç farklı öğrenme oranı (0.8, 0.7, 0.6) ve iki farklı bozulma oranı (0.2, 0.1) kullanılarak 10 iş, 20 iş ve 30 iş içeren problemler matematiksel modeller ve algoritmalar ile çözülmüş ve karşılaştırma tablosu sunulmuştur.

Anahtar kelimeler: Öğrenme etkisi, Bozulma etkisi, Iş reddetme, Tek makine, Paralel makineler.

1 Introduction

In our study, with job rejection scheduling problem under the learning and deterioration effects was considered. The learning effect refers to the phenomenon where the processing time of a job decreases as a function of its position in the sequence. This effect arises from the experience gained by workers or machines performing repetitive tasks, leading to increased efficiency and reduced completion times. In contrast, the deterioration effect describes a scenario where job processing times increase due to factors such as machine wear, worker fatigue, or material degradation. This effect is particularly relevant in manufacturing and maintenance-intensive environments, where prolonged operations lead to decreased efficiency over time. Four different objectives are discussed; these are rejection cost and makespan minimization for a single machine, rejection cost and total completion time minimization for a single machine, rejection cost and total absolute difference completion time minimization for a single machine, and rejection cost and makespan minimization on parallel machines.

Mosheiov is the first person to develop the terminology of learning effect [1]. In his study [2] he studied the total completion time minimization under the effect of learning. The equality that Biskup [3] thinks for the actual processing time is as follows in Equation 1.

$$p_{j[r]} = (p_j) r^{\alpha} \tag{1}$$

If job J is assigned in r^{th} position its actual processing time is $p_{j[r]}$. And basic processing time of job J is p_j and the learning – index is α $(\alpha < 0)$.

In the study of Lee [4], the effect of learning and deterioration was first considered together. The proposed equality for the processing time is as follows Equation 2 is given below.

$$p_{i,r} = (p_0 + \gamma_i t) r^{a} \tag{2}$$

If job i assigned in position r, actual processing time is $p_{i,r}$. p_0 is the basic processing time and γ $(\gamma > 0)$ is the deterioration rate. t is the started time of job i and α $(\alpha < 0)$ is the learning - index.

Also, in Wang [5]'s study, the deterioration effect and the learning effect are considered at the same time. In the study of Mosheiov [6], the problem of scheduling jobs under linear deterioration in a single machine was examined. The aim of the study is to minimize flow time. In the study of Kononov [7], the processing times of the jobs were defined as an increasing function of starting times. In the study of Hsieh and Bricker [8], the processing time of the works was examined as a linearly increasing function of starting time.

The rejection of jobs in machine scheduling was first considered by Bartal [9]. In Zhang et al. [10], scheduling

* Sorumlu yazar / Corresponding author, e-posta / e-mail: berrinucarkus@erciyes.edu.tr. (B. Atalay)
Geliş / Received: 25.10.2023 Kabul / Accepted: 07.03.2025 Yayımlanma / Published: 15.04.2025
doi: 10.28948/ngumuh.1381334

problem with a single machine under job rejection constraint was examined. The job is either rejected and the rejection cost is paid or the job is accepted and processed on one machine. Toksarı et al. [11] simultaneously considered the effect of nonlinear deterioration and time - based learning. In Gerstl and Mosheiov [12], position-based processing times and rejection of jobs are combined for the parallel machine scheduling problem. There are two objectives; the first objective function is to minimize the sum of flow time and the cost of rejected jobs, and the second objective function is to minimize the total load and the cost of rejected jobs.

In the study of Mor and Mosheiov [13], position-based processing times on parallel match machines and position-based processing times on separate machines and TADC (total absolute difference completion time) were studied. In the study of Li et al. [14], a single machine scheduling problem was investigated with deteriorated jobs. They defined the deterioration of the work as a proportional function of time. The aim is to minimize the TADC.

In second section, rejection cost and maximum completion time minimization under the effects of learning and deterioration in a single machine environment are discussed, theorem is defined, proof, mathematical models and algorithms are presented. In section 3, rejection cost and total completion time minimization under the effects of learning and deterioration in a single machine environment are discussed, theorem is defined, proof, mathematical models and algorithms are presented. In section 4, rejection cost and TADC minimization under the effects of learning and deterioration in a single machine environment are discussed, mathematical models and algorithms are presented. In fifth section, rejection cost and makespan minimization under the effects of learning and deterioration in parallel machines environment are discussed, theorem is defined, proof, mathematical models and algorithms are presented. In section 6, the results of the algorithms and mathematical models are given for 10 jobs, 20 jobs and 30 jobs. The developed algorithms were coded in Visual Studio C# 2017. Lingo 11.0 program was used to solve mathematical models. The processing times and penalty costs of all jobs were randomly generated.

2 Makespan and rejection cost minimization for single machine scheduling problem

In our study, one machine, n jobs were considered. P_{jr} is the duration of job j assigned to position r. $i, j, r = 1, \dots, n$. If j is not assigned to the machine, it receives a penalty cost of δ_j. α is the learning coefficient. In literature, it is generally taken as 0.8. γ: coefficient of deterioration $1 > \gamma > 0$. T_r: This is the starting time of the job that started in position r. X_{ijr} is our decision variable, if j is assigned to position r, then 1 is 0. When $i = 1$, it means that it is a job of the accepted set of jobs, when $i = 2$, it means that it is a job of the rejected set of jobs and receives a penalty cost of δ_j. The maximum completion time is shown as C_{max}.

Theorem 1: The optimum solution of the $1|p_{j[r]} = r^2(p_j + \gamma.T_r)r^\alpha|C_{max}$ problem is achieved by ordering the jobs accordingly the LPT rule.

The objective function is designed to minimize the makespan C_{max} while considering both job processing times and a specific penalty term $(\gamma.T_r)$. The function incorporates a job-dependent term, r^2, and an additional parameter, α, to adjust the impact of job characteristics. The optimal solution is achieved by arranging the jobs in the order specified by the Longest Processing Time (LPT) rule, which prioritizes jobs with the longest processing times.

Proof: Under the effect of learning, n work is considered a single machine problem. $p_k \leq p_j$ and $\alpha = log(a)/log(2) < 0$ are assumed. r is the position where the job is scheduled. A: This is the completion time of the last job scheduled before k and j jobs.

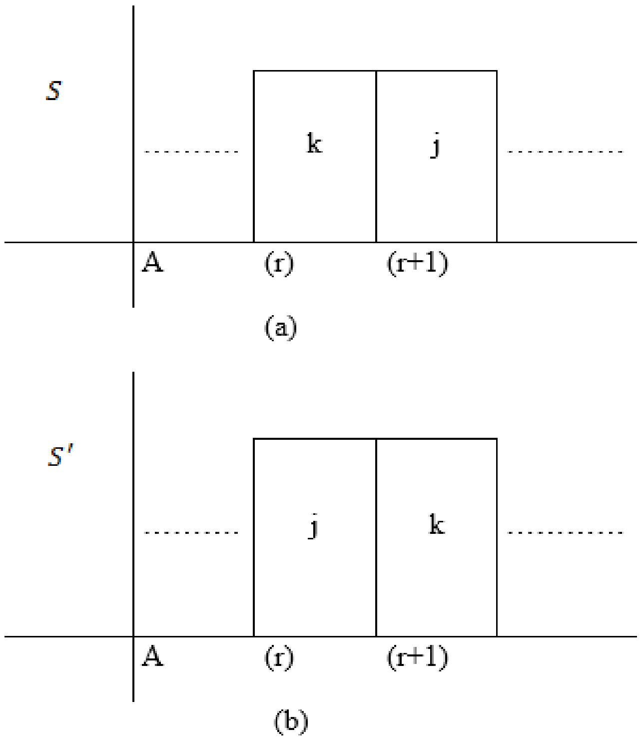

Figure 1. Representation of i and j jobs on chart S and S′

For the S chart seen in Figure 1 (a);

$$C_k(S) = A + r^2(p_k + \gamma.T_r)r^\alpha \tag{3}$$

$$C_j(S) = A + r^2(p_k + \gamma.T_r)r^\alpha + (r+1)^2\left(p_j + \gamma.T_{r+1}\right)(r+1)^\alpha \tag{4}$$

For the S′ chart shown in Figure 1 (b);

$$C_j(S') = A + r^2(p_j + \gamma.T_r)r^\alpha \tag{5}$$

$$C_k(S') = A + r^2\left(p_j + \gamma.T_r\right)r^\alpha + (r+1)^2(p_k + \gamma.T_{r+1})(r+1)^\alpha \tag{6}$$

The calculation of the completion time of job k in chart S is given in Equation (3). The calculation of the completion time of job j in chart S' is given in Equation (5). Equation (4) and (6)'s differences are obtained by taking the following situation. The chart S' is better than the chart S is shown Equation (7).

$$C_k(S') - C_j(S) = (p_j - p_k)(r^{2+\alpha} - (r+1)^{2+\alpha}(1-\gamma)) \text{ and } \alpha < 0 \text{ and } p_k \leq p_j \text{ if;} \quad (7)$$
$$C_k(S') - C_j(S) < 0$$

Theorem 1 shows that the chart S′ is better than the chart S and proves that the optimum solution of the problem $1|p_{j[r]} = r^2(p_j + \gamma.T_r)r^\alpha|C_{max}$ is obtained by LPT rule.

The mathematical model of our problem is as follows.

$$min \sum_{j=1}^{n}\sum_{r=1}^{n} r^2.((p_j + \gamma.T_r).r^\alpha.X_{1jr}) + \delta_j.X_{2jr} \quad (8)$$

$$\sum_{j=1}^{n} X_{ijr} \leq 1 \qquad (r = 1,2,3,\dots,n) \quad (i = 1,2) \quad (9)$$

$$\sum_{i=1}^{2}\sum_{r=1}^{n} X_{ijr} = 1 \qquad (j = 1,2,3,\dots,n) \quad (10)$$

$$T_1 = 0 \quad (11)$$

$$T_r = \sum_{j=1}^{n}\sum_{k}^{r-1} r^2.((p_j + \gamma.T_k).k^\alpha.X_{1jk}) \quad (r = 2,3,\dots,n) \ (k = 1,\dots n) \quad (12)$$

$$X_{ijr} \in (0,1) \quad (13)$$

The objective function (Equation (8) is the minimization of rejection cost and maximum completion time. Equation (9) guarantees that job j will be assigned to a single position, whether it is accepted or rejected. Equation (10) ensures that each job is assigned to a single position. Equation (11) indicates that the start time of the job in position 1 is 0. Equation (12) ensures the calculation of the start time of the job in position r. Equation (13) states that the decision variable must be either 0 or 1.

The steps for the proposed algorithm are given below Makespan algorithm under the effect of deterioration

Makespan algorithm under the effect of deterioration
Input: R, A, n, i, TC, γ, p_j, δ_j
R: Rejected jobs.
A: Accepted jobs.
n: Number of jobs.
TC: Total Cost= $\sum_{n=1}^{A} C_{max} + \sum_{j=1}^{R} \delta_j$.
p_j: Processing time of job j.
δ_j: Rejection cost of job j.
γ: Deterioration rate.

Step 1: $i = 0, R = 0, A = n$. Accept all jobs, reject no job. Schedule the accepted jobs according to the LPT rule. Compute the *TC* as much as the *C (n, i)* combination.

Step 2: $i = i + 1, R = i, A = n - 1$. Decrease 1 accepted jobs and increase as i rejected jobs. Assign the accepted jobs by LPT rule. Compute the *TC* of the plot as much as the *C (n, i)* combination.

Step 3: If $i < n$, Repeat Step 2, If $i = n$ stop and go to Step 4.

Step 4: $min\,\{TC_1, TC_2, \dots, TC_{2^n}\}$. Make comparison all TC, devise the smallest TC. The schedule in which the smallest TC value is optimum.

3 Rejection cost and total completion time minimization for single machine scheduling problem

In our study, one machine, n jobs were considered. p_{jr} is the duration of job *j* assigned to position *r*. *j, r = 1,…, n.* If *j* is not assigned to the machine, it receives a penalty cost of δ_j. α is the learning coefficient. In literature, it is generally taken as 0.8 [7, 11, 14]. γ: coefficient of deterioration $1 > \gamma > 0$. T_r: is the starting time of the job that started in position r. X_{ijr} is our decision variable. If *j* is assigned to position *r*, X_{ijr} becomes 1, otherwise it is 0. When $i = 1$, it means that it is a job of the accepted set of jobs, when $i = 2$, it means that it is a job of the rejected set of jobs and receives a penalty cost of δ_j.

Theorem 2: The optimum solution of the $1\left|p_{j[r]} = \left(\frac{r}{2}\right).\left(p_j + \gamma.T_r\right).r^\alpha\right|\sum C$ problem is achieved by ordering the jobs as required to the SPT rule.

The objective function is aims to minimize the total completion time by considering both the job processing times and a penalty term $(\gamma.T_r)$. The function also includes a factor, $\left(\frac{r}{2}\right)$, that adjusts the impact of the job position and a parameter α to account for job characteristics. The optimal solution is achieved by arranging the jobs according to the Shortest Processing Time (SPT) rule, which prioritizes jobs with the shortest processing times to minimize the total completion time.

Proof: Under the effect of learning, n work is considered a single machine problem. $\alpha = log(a)/log(2) < 0$ and $p_k \leq p_j$ are assumed. The position to which the job is scheduled is assigned is r. A: This is the completion time of the last job scheduled before i and j jobs.

For the S chart as seen in Figure 1 (a);

$$C_k(S) = A + (\frac{r}{2})(p_k + \gamma.T_r)r^\alpha \quad (14)$$

$$C_j(S) = A + \left(\frac{r}{2}\right)(p_k + \gamma.T_r)r^\alpha + \left(\frac{r+1}{2}\right)\left(p_k + \gamma\left(p_j + \gamma.T_r\right)\right)(r+1)^\alpha \quad (15)$$

$$\sum C(S) = 2A + r(p_k + \gamma.T_r)r^\alpha + \left(\frac{r+1}{2}\right)\left(p_k + \gamma\left(p_j + \gamma.T_r\right)\right)(r+1)^\alpha \quad (16)$$

For the S′ chart shown in Figure 1 (b);

$$C_j(S') = A + \left(\frac{r}{2}\right)(p_j + \gamma.T_r)r^\alpha \quad (17)$$

$$C_k(S') = A + \left(\frac{r}{2}\right)(p_j + \gamma.T_r)r^\alpha + \left(\frac{r+1}{2}\right)(p_j + \gamma(p_k + \gamma.T_r))(r+1)^\alpha \quad (18)$$

$$\sum C(S') = 2A + r(p_j + \gamma.T_r)r^\alpha + \left(\frac{r+1}{2}\right)\left(p_j + \gamma(p_k + \gamma.T_r)\right)(r+1)^\alpha \quad (19)$$

The calculation of the completion time of job k in chart S is given in Equation (14). The calculation of the completion time of job j in chart S is given in Equation (15). The calculation of the completion time of job j in chart S' is given in Equation (17). The calculation of the completion time of job k in chart S' is given in Equation (18). Equation (16) and (19)'s differences are obtained by taking the following situation in Equation (20). The chart S is better than the chart S' is shown Equation (21).

$$\sum C(S') - \sum C(S) = (p_j - p_k)\left(rr^\alpha - \left(\frac{r+1}{2}\right)(r+1)^\alpha\right) \quad (20)$$

$$= (p_j - p_k)\left(2r^{\alpha+1} - \left(\frac{r+1}{2}\right)(r+1)^\alpha(1-\gamma)\right) > 0 \quad (21)$$

According to Theorem 2, S is shown to be better than S' and $1\left|p_{j[r]} = \left(\frac{r}{2}\right).(p_j + \gamma.T_r).r^\alpha\right|\sum C$ prove that it is obtained by SPT rule.

The mathematical model of our problem is as follows.

$$min \sum_{j=1}^{n}\sum_{r=1}^{n}\left(\frac{r}{2}\right).(n-j+1).((p_j + \gamma.T_r).r^\alpha.X_{1jr}) + \delta_j.X_{2jr} \quad (22)$$

$$\sum_{j=1}^{n} X_{ijr} \le 1 \qquad (r = 1,2,3,\dots,n)\ (i = 1,2) \quad (23)$$

$$\sum_{i=1}^{2}\sum_{r=1}^{n} X_{ijr} = 1 \qquad (j = 1,2,3,\dots,n) \quad (24)$$

$$T_1 = 0 \quad (25)$$

$$T_r = \sum_{j=1}^{n}\sum_{k}^{r-1} r^2.((p_j + \gamma.T_k).k^\alpha.X_{1jk}) \qquad (r = 2,3,\dots,n) \quad (26)$$

$$X_{ijr} \in (0,1) \quad (27)$$

The objective function is the minimization of rejection cost and total completion time shown in Equation (22). Equation (23) guarantees that job j will be assigned to a single position, whether it is accepted or rejected. Equation (24) ensures that each job is assigned to a single position. Equation (25) indicates that the start time of the job in position 1 is 0. Equation (26) ensures the calculation of the start time of the job in position r. Equation (27) states that the decision variable must be either 0 or 1.

The steps for the proposed algorithm are given below:

Under the effect of deterioration total completion time algorithm

Input: R, A, n, i, TC, γ, p_j, δ_j.

R: Rejected jobs.

A: Accepted jobs.

n: Number of jobs.

TC: Total Cost= $\sum_{n=1}^{A} C_{max} + \sum_{j=1}^{R} \delta_j$.

p_j: Processing time of job j.

δ_j: Rejection cost of job j.

γ: Deterioration rate.

Step 1: $i = 0, R = 0,\ A = n$. Accept all jobs, reject no job. Schedule the accepted jobs according to the SPT rule. Compute the *TC* as much as the *C (n, i)* combination.

Step 2: $i = i + 1, R = i, A = n - 1$. Decrease 1 accepted jobs and increase as i rejected jobs. Assign the accepted jobs by SPT rule. Compute the *TC* of the plot as much as the *C (n, i)* combination.

Step 3: If $i < n$, Repeat Step 2, If $i = n$ stop and go to Step 4.

Step 4: $min\ \{TC_1, TC_2, \dots, TC_{2^n}\}$. Make comparison all TC, devise the smallest TC. The schedule in which the smallest TC value is optimum.

4 Rejection cost and total absolute difference completion time minimization for single machine scheduling problem

In our study, the intuitive method developed by Oron [15] was used and developed the jobs with under the effect of job rejection, deterioration and learning.

Kanet [16] expressed the position weight as $w\_r = (r-1)(n-r+1), r = 1,2,\dots,n$. Position weights were used in this way in our study.

The mathematical model of our problem is as follows.

$$min \sum_{j=1}^{n}\sum_{r=1}^{n}(r-1)(n-j+1).(p_j + \gamma.T_r).r^\alpha.X_{1jr} + \delta_j.X_{2jr} \quad (28)$$

$$\sum_{j=1}^{n} X_{ijr} \le 1 \qquad (r = 1,2,3,\dots,n) \quad (i = 1,2) \quad (29)$$

$$\sum_{i=1}^{2}\sum_{r=1}^{n} X_{ijr} = 1 \qquad (j = 1,2,3 \dots, n) \tag{30}$$

$$T_1 = 0 \tag{31}$$

$$T_r = \sum_{j=1}^{n}\sum_{k}^{r-1}((p_j + \gamma.T_k).k^{\alpha}.X_{1jk}) \qquad (r = 2, 3, \dots, n) \tag{32}$$

$$X_{ijr} \in (0,1) \tag{33}$$

The objective function is given in Equation (28) the minimization of total absolute difference completion time and rejection cost. Equation (29) guarantees that job j will be assigned to a single position, whether it is accepted or rejected. Equation (30) ensures that each job is assigned to a single position. Equation (31) indicates that the start time of the job in position 1 is 0. Equation (32) ensures the calculation of the start time of the job in position r. Equation (33) states that the decision variable must be either 0 or 1. The steps for the proposed algorithm are as follows:

Input: *R*, *A*, *n*, *i*, *TC*, γ, p_j, δ_j.
R: Rejected jobs.
A: Accepted jobs.
n: Number of jobs.
TC: Total Cost= $\sum_{n=1}^{A} TADC + \sum_{j=1}^{R} \delta_j$.
p_j: Processing time of job j.
δ_j: Rejection cost of job j.
γ: Deterioration rate.

If *t* is even go to Step 1 and if *t* is odd go to Step 3.

Step 1: Assign job $J_{t/2+2-i}$ to position *i*, $i = 1, 2, \dots, \frac{t}{2} + 1$.

Step 2: Assign job J_i to position *i*, $i = \frac{t}{2} + 2,\ \frac{t}{2} + 3, \dots, t$. Of the result form: $\{J_{\frac{t}{2}+1}, J_{\frac{t}{2}}, \dots, J_3, J_2, J_1, J_{\frac{t}{2}+2}, J_{\frac{t}{2}+3}, \dots, J_t\}$.

Step 3: Assign job $J_{\frac{t+5}{2}-i}$ to position *i*, $i = 1, 2, \dots, \frac{t+3}{2}$.

Step 4: Assign job J_i to position *i*, $i = \frac{t+5}{2}, \frac{t+7}{2}, \dots, t$. Go to Step 5.

Schedule of jobs: $\{J_{\frac{t+3}{2}}, J_{\frac{t+1}{2}}, \dots, J_3, J_2, J_1, J_{\frac{t+5}{2}}, J_{\frac{t+7}{2}}, \dots, J_t\}$.

Step 5: Calculate TC.

Step 6: $min\,\{TC_1, TC_2, \dots, TC_{2^n}\}$. Comparison all TC, devise the smallest TC. The schedule on which the smallest TC is obtained is nearest the optimum.

5 Rejection cost and total absolute difference completion time minimization for single machine scheduling problem

Theorem 1 proved that the optimum solution for the $2|p_{j[r]} = \mathrm{r}(p_j + \gamma.T_r)r^{\alpha}|C_{max}$ problem would be obtained by ordering the works according to the LPT rule according to the process time.

The mathematical model of our problem is as follows.

$$min \sum_{j=1}^{n} r.(p_j + \gamma.T_r).r^{\alpha}.X_{1jr} + \sum_{j=1}^{n} r.(p_j + \gamma.T1_r).r^{\alpha}.X_{2jr} + \delta_j.X_{3jr} \tag{34}$$

$$\sum_{j=1}^{n} X_{ijr} \le 1 \qquad (i = 1,2,3) \quad (r = 1,2, \dots, n) \tag{35}$$

$$\sum_{i=1}^{3}\sum_{r=1}^{n} X_{ijr} = 1 \qquad (j = 1,2, \dots, n) \tag{36}$$

$$T_r = \sum_{j=1}^{n}\sum_{k}^{r-1}((p_j + \gamma.T_k).k^{\alpha}.X_{1jk}) \qquad (r = 2, 3, \dots, n). \tag{37}$$

$$T1_1 = 0 \tag{38}$$

$$T1_r = \sum_{j=1}^{n}\sum_{k}^{r-1}((p_j + \gamma.T1_k).k^{\alpha}.X_{1jk}) \qquad (r = 2, 3, \dots, n). \tag{39}$$

$$X_{ijr} \in (0,1) \tag{40}$$

The objective function is the minimization of makespan and rejection cost is shown in Equation (34). Equation (35) guarantees that job j will be assigned to a single position, whether it is accepted or rejected. Equation (36) ensures that each job is assigned to a single position. Equation (37) ensures the calculation of the start time of the job in position r. Equation (38) indicates that the start time of the job in position 1 is 0. Equation (39) ensures the calculation of the start time of the job in position r. Equation (40) states that the decision variable must be either 0 or 1.

The algorithm used for single machine makespan minimization was also used for parallel machine.

6 Computational results

Mathematical model results and algorithm results obtained for 10 jobs, 20 jobs and 30 jobs are as follows.

The mathematical model and the algorithm results obtained for 10 jobs, 20 jobs, and 30 jobs are given in Table 1, Table 2, and Table 3, respectively. Three commonly used learning rates and two deterioration rates from the literature were used.

The learning ratio is a quantitative measure used to assess the impact of learning effects on processing times in scheduling and optimization problems. It represents the rate at which performance improves as a function of accumulated experience, typically observed in repetitive production or task execution scenarios. Mathematical Model Results have been obtained based on the objective functions in the mathematical models in Lingo 11.0 program. Algorithm Results are the outcomes obtained from the proposed algorithms. Algorithms were coded in Visual Studio C# 2017. Accuracy is calculated using the formula: (Mathematical Model Results) / (Algorithm Results) * 100. It compares the result obtained by the algorithm with the mathematical model result in percentage terms.

As shown in Table 1, the mathematical and algorithm model results for single machine makespan minimization and total completion time minimization were the same.

For TADC, the results were obtained with an average accuracy of about 87%.

For parallel machine makespan minimization, the proposed algorithm is more accurate.

As shown in Table 2, for the 20 jobs, the average of 100% accuracy was obtained in the problem of makespan minimization.

The accuracy obtained for total completion time minimization is 98%.

For TADC minimization, the results were obtained with an approximate accuracy of 94.5%.

As shown in Table 3, as the number of jobs increases, it is seen that the proposed algorithm works better for makespan minimization. In the total completion time minimization problem, except the 60% learning rate, the proposed algorithm is better for all other results.

For TADC minimization, the results were approximately 93.2% accurate.

For the parallel machine makespan minimization problem, all results except 80% learning rate and 10% deterioration rate are better than the proposed algorithm.

Table 1. Algorithm results and mathematical model results for 10 jobs.

Number of Jobs	Objective Functions	LR (Learning Ratio)	Deterioration Ratio	Mathematical Model Results	Algorithms Results	Accuracy
10 jobs	**Makespan for single machine**	80%	0.1	303.04	303.04	100.00
		80%	0.2	306.88	306.88	100.00
		70%	0.1	300.16	300.16	100.00
		70%	0.2	303.52	303.52	100.00
		60%	0.1	297.28	297.28	100.00
		60%	0.2	300.16	300.16	100.00
	Total Completion Time	80%	0.1	310.68	310.68	100.00
		80%	0.2	311.16	311.16	100.00
		70%	0.1	304.22	304.22	100.00
		70%	0.2	306.457	306.457	100.00
		60%	0.1	297.566	297.565	100.00
		60%	0.2	299.358	299.358	100.00
	TADC	80%	0.1	282.995	324.406	87.23
		80%	0.2	290.558	326	89.13
		70%	0.1	272.409	309.353	88.06
		70%	0.2	278.736	326	85.50
		60%	0.1	262.413	295.822	88.71
		60%	0.2	267.594	310.034	86.31
	Makespan for parallel machines	80%	0.1	247.489	213.72	115.80
		80%	0.2	258.244	218.75	118.05
		70%	0.1	235.213	200.108	117.54
		70%	0.2	248.0227	206.43	120.15
		60%	0.1	221.1609	189.545	116.68
		60%	0.2	229.111	193.532	118.38

Table 2. Algorithm results and mathematical model results for 20 jobs.

Number of Jobs	Objective Functions	LR (Learning Ratio)	Deterioration Ratio	Mathematical Model Results	Algorithms Results	Accuracy
20 jobs	Makespan for single machine	80%	0.1	669.41	669.41	100.00
		80%	0.2	705.25	686.989	102.66
		70%	0.1	660.0025	660	100.00
		70%	0.2	672.9083	672.909	100.00
		60%	0.1	642.02	642.016	100.00
		60%	0.2	659.5101	660.973	99.78
	Total Completion Time	80%	0.1	755.083	741.815	101.79
		80%	0.2	783.72	742.986	105.48
		70%	0.1	720.87	738.35	97.63
		70%	0.2	721.397	739.254	97.58
		60%	0.1	666.43	722.92	92.19
		60%	0.2	667.556	726.067	91.94
	TADC	80%	0.1	646.9	684.872	94.46
		80%	0.2	662.648	706.073	93.85
		70%	0.1	633.294	662.227	95.63
		70%	0.2	644.799	691.13	93.30
		60%	0.1	617.18	644.34	95.78
		60%	0.2	629.8	666.926	94.43
	Makespan for parallel machines	80%	0.1	515.082	505.923	101.81
		80%	0.2	547.573	499.316	109.66
		70%	0.1	457.664	465.08	98.41
		70%	0.2	506.676	494.553	102.45
		60%	0.1	362.634	374.616	96.80
		60%	0.2	410.821	405.767	101.25

Table 3. Mathematical model results and algorithm results for 30 jobs.

Number of Jobs	Objective Functions	LR (Learning Ratio)	Deterioration Ratio	Mathematical Model Results	Algorithms Results	Accuracy
30 jobs	Makespan for single machine	80%		1083.16	1073.858	100.87
		80%	0.2	1105.33	1087.684	101.62
		70%	0.1	1059.79	1059.789	100.00
		70%	0.2	1177.61	1072.05	109.85
		60%	0.1	1039.59	1031.77	100.76
		60%	0.2	1052.3	1045.187	100.68
	Total Completion Time	80%	0.1	1126.73	1109.673	101.54
		80%	0.2	1125.16	1086.94	103.52
		70%	0.1	1067.42	1055.78	101.10
		70%	0.2	1069.39	1060.86	100.80
		60%	0.1	971.308	1014.356	95.76
		60%	0.2	973.65	1022.706	95.20
	TADC	80%	0.1	1144.5	1164	98.32
		80%	0.2	1090.738	1164	93.71
		70%	0.1	1097.204	1157.96	94.75
		70%	0.2	1073.058	1164	92.19
		60%	0.1	1038.64	1147.96	90.48
		60%	0.2	1044.337	1159.22	90.09
	Makespan for parallel machines	80%	0.1	867.894	873.527	99.36
		80%	0.2	905.968	867.065	104.49
		70%	0.1	846.391	842.933	100.41
		70%	0.2	866.881	865.08	100.21
		60%	0.1	676.287	662.485	102.08
		60%	0.2	759.909	742.194	102.39

7 Conclusion

In our study, the effect of learning and deterioration in the job rejection scheduling environment was considered together. In our study, this situation was examined for four different objectives. These problems are makespan minimization, total completion time minimization and TADC minimization for single machine scheduling problems and makespan minimization for parallel machine scheduling problems. When comparing the results from the tables, it can be seen that the best results are obtained for makespan minimization in both the single machine and parallel machine cases. The results of TADC also perform better for 20 and 30 jobs, but overall, makespan minimization provides the best results.

As it can be seen from the results tables, the proposed algorithms provided solutions with very high accuracy. All the solutions found were obtained in polynomial time.

Conflict of interest

No potential conflict of interest was reported by the authors

Similarity rate (iThenticate): %19

References

[1] G. Mosheiov, Scheduling problems with a learning effect. European Journal of Operational Research, 132, 687– 693, 2001. https://doi.org/10.1016/S0377-2217(0 0)00175-2.

[2] G. Mosheiov, Parallel machine scheduling with a learning effect. Journal of the Operational Research Society, 52(10), 1165-1169, 2001. https://doi.org/10.1057/palgrave.jors.2601215.

[3] D. Biskup, Single-machine scheduling with learning considerations. European Journal of Operational Research, 115(1), 173-178, 1999. https://doi.org/10.1016/S0377-2217(98)00246-X.

[4] W. C. A. Lee, note on deteriorating jobs and learning in single-machine scheduling problems. International Journal of Business and Economics, 3(1), 83, 2004.

[5] J. B. Wang, Single-machine scheduling problems with the effects of learning and deterioration. Omega, 35(4), 397-402, 2007. https://doi.org/10.1016/j.omega.2005.07.008.

[6] G. Mosheiov, V-shaped policies for scheduling deteriorating jobs. Operations Research, 39(6), 979-991, 1991. https://doi.org/10.1287/opre.39.6.979.

[7] A. Kononov, NP-hard cases in scheduling deteriorating jobs on dedicated machines. Journal of the Operational Research Society, 52(6), 708-717, 2001. https://doi.org/10.1057/palgrave.jors.2601117.

[8] Y. C. Hsieh, D. L. Bricker, Scheduling linearly deteriorating jobs on multiple machines. Computers & Industrial Engineering, 32(4), 727-734, 1997. https://doi.org/10.1016/S0360-8352(97)00025-9.

[9] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, L. Stougie, Multiprocessor scheduling with rejection. SIAM Journal on Discrete Mathematics, 13(1), 64-78, 2000. https://doi.org/10.1137/S0895480196300522.

[10] L. Zhang, L. Lu, J. Yuan, Single-machine scheduling under the job rejection constraint. Theoretical Computer Science, 411(16), 1877-1882, 2010. https://doi.org/10.1016/j.tcs.2010.02.006.

[11] M. D. Toksarı, D. Oron, E. Güner, Single machine scheduling problems under the effects of nonlinear deterioration and time-dependent learning. Mathematical and Computer Modelling, 50(3-4), 401-406, 2009. https://doi.org/10.1016/j.mcm.2009.05.026.

[12] E. Gerstl, G. Mosheiov, Scheduling on parallel identical machines with job-rejection and position-dependent processing times. Information Processing Letters, 112(19), 743-747, 2012. https://doi.org/10.1016/j.ipl.2012.06.009.

[13] B. Mor, G. Mosheiov, Total absolute deviation of job completion times on uniform and unrelated machines. Computers & Operations Research, 38(3), 660-665, 2011. https://doi.org/10.1016/j.cor.2010.08.005.

[14] Y. Li, G. Li, L. Sun, Z. Xu, Single machine scheduling of deteriorating jobs to minimize total absolute differences in completion times. International Journal of Production Economics, 118(2), 424-429, 2009. https://doi.org/10.1016/j.ijpe.2008.11.011.

[15] D. Oron, Single machine scheduling with simple linear deterioration to minimize total absolute deviation of completion times. Computers & Operations Research, 35(6), 2071-2078, 2008. https://doi.org/10.1016/j.cor.2006.10.010.

[16] J. J. Kanet, Minimizing variation of flow time in single machine systems. Management Science, 27(12), 1453-1459, 1981. https://doi.org/10.1287/mnsc.27.12.1453.