J Biochem Tech (2020) Special Issue (S1): 189-201
ISSN: 0974-2328

Unrelated Parallel Machine Scheduling Taking into Account the Rate of Activities Correction and their Deterrence in the Problem

Mohaddeseh Maghrebi*, Iraj Mahdavi, Javad Rezaeian

Received: 04 October 2019 / Received in revised form: 12 February2020, Accepted: 19 February 2020, Published online: 29 February2020

Abstract

In this study, a model for scheduling projects with the aim of minimizing the total weight of earliness and tardiness costs of works, a situation where prerequisite work and rate of correction of activities and their depreciation is allowed in unrelated parallel machine. Completion of activities earlier and later sometimes imposes heavy costs on the organization, which, if completed earlier, incurs maintenance costs and penalties for tardiness. To receive feedback and appropriate decisions on any project, project management must use a system that has the ability to integrate between the scope of work, cost, and time. In the real world, zeroing the total cost of tardiness and earliness is somewhat impossible, and pre-work is one of the most important issues in scheduling theory. The project's scheduling issues seem to have received less attention with criteria such as earliness and tardiness, in which pre-work permitting is allowed and depreciation is involved, as well as activity correction rates. This scheduling is one of the NP-hard issues that will be discussed below.

Key words: Unrelated Parallel Machine Scheduling, Activity Correction Rate, Depreciation, Minimize Earliness and Tardiness Costs

Introduction

Today, due to the growing growth of industries and the existence of new technologies in industries, there is fierce competition between industries and organizations. If they want to protect their position, they must not only be flexible in this situation, but they must create privileges for themselves that are superior to others in order to attract customers. Scheduling and sequencing operations is a decision-making issue that plays a key role in production and service systems. Among industries with a competitive environment, a scheduling process and sequence of labor operations are needed to survive in the business environment. Scheduling is also limited to activities that require resources. It deals with allocating resources to activities over a given period of time, and the goal is to optimize one or more goals. Resources and activities in an organization have many forms. Production scheduling issues are very diverse.

In the field of production planning, which usually includes medium-term horizons, scheduling operations in short-term horizon areas is considered as a branch of it. There are several classes in scheduling, the most important of which are parallel machines. Unrelated parallel machine scheduling is known as an important category of scheduling issues that are of great theoretical and experimental importance. In scheduling production systems, it should be noted that any delay in the start of work processing can lead to increased work time and system costs. This phenomenon is known as “depreciation” in the literature.

Also, each activity has special conditions such as duration, specific weight, start time and delivery date. On the other hand, the environment in which the activities are carried out can have limitations such as machine failure, preparation time, start time, cutting work, prerequisite limitation, activity correction rate. A number of goals that are considered and evaluated for the objective function include minimizing the total time to complete the jobs, minimizing the earliness, minimizing the maximum delay time, minimizing the number of delayed jobs, minimizing the total earliness times, and so on. One of the organizational policies that cause more flexibility to respond to the customer and to create higher productivity and efficiency for the organization and through it can reduce the high costs imposed on the organization. The aim of this study is to investigate the minimization of total earliness and tardiness of work on parallel machines, because the study of a group of machines that are placed in parallel is of particular importance from both theoretical and practical perspectives. From a theoretical point of view, parallel machines are a generalization of the state of a single machine and a particular state of a flexible flow workshop. From a practical point of view, this is even more important because placing resources in parallel in the real world is more common. In addition, parallel machine methods are mostly used in multi-stage systems decomposition processes. In scheduling, parallel machines are considered for different purposes. The three main goals are minimizing the longest completion time, total completion time, and minimizing the total tardiness and earliness.

Scheduling problems with time-dependent depreciation rates and the solution method for all problems such as mathematical formulation

Mohaddeseh Maghrebi, Iraj Mahdavi, Javad Rezaeian

Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran.

and polysyllabic algorithms have been expressed by Zhou et al. (2015).

In scheduling production systems, it should be noted that any delay in the start of processing work, which is called depreciation, increases the processing time and as a result, the time to get things done increases, and this delay in completion leads to costs such as the cost of losing a customer and ordering. In recent years, scheduling issues have been widely explored.

Optimal activity correction rates or maintenance times are considered. However, research has been limited by not allowing depreciation. Finally, for the third category, most research deals with the effects of depreciation and RMA simultaneously (Chang and Kim, 2016; Wu and Kim, 2016).

Zheng et al. (2017) considered a single-machine scheduling problem with multiple RMAs and proposed polynomial-time algorithm with reference to the properties of Gupta and Gupta (1988) articles. They were able to find an optimal solution for scheduling problems, but with the explanation that the number of RMAs is constant, unfortunately, past research has dealt with single machine scheduling problems.

Yang (2013) has considered the RMA depreciation times on the unrelated parallel machine scheduling with depreciation and several correction rates, and has obtained an optimal solution by a polynomial-time algorithm for the problem. However, these studies have limited solutions to scheduling problems due to the presence of two major decision-making factors: the number of RMAs and the number of machines. Then, the general optimal solution obtained by counting all the local optimal points that originate from the mathematical models corresponding to all the combinations of RMAs, which is less than the upper limit of the number of RMAs. Thus, the polynomial algorithms shown by Yang (2013) found that it is difficult to find a general optimal solution or a logical solution time, even if the number of jobs and RMAs increases.

Joe and Kim (2015) have considered a single-machine scheduling problem with time-dependent depreciation and multiple RMAs, and have shown different genetic algorithms to overcome the limitations of polynomial algorithms. For example, a certain number and position of RMAs and a certain number of machines can be named. Although there is a lot of research on scheduling issues with depreciation and RMA, no research has developed a good metaheuristic algorithm for the unrelated parallel machine scheduling problem, taking into account depreciation, and multiple RMAs, and considering the prerequisite constraint between jobs. So that there are no restrictions on the number of machines and RMAs.

One of the most widely used objective functions in parallel machine optimization problems is to minimize costs. If this cost is related to the tardiness or earliness activities, in both cases it will cost the production unit because achieving this goal will cause the works to be distributed as uniformly as possible between the machines and the working capacity of all the machines should be used to the desired extent, and as a result, the accumulation of works on one or a number of machines should be prevented. Therefore, the cost minimization criterion has been used as the optimization criterion in the proposed model.

There has been a lot of research on scheduling issues over the years. A review of the literature suggests that the problem of parallel machine scheduling is one of the most important and widely used topics. This is a generalized problem of single-machine problems. From a practical point of view, it can be acknowledged that in most production units there is more than one machine. Today, in most manufacturing plants, the needs of production can be attributed to the optimal use of resources and their timely allocation, so that a proper sequence of work is created on the machines so that the maximum capacity of the machines can be used. In the meantime, timely allocation is more important because if the sequence is done, it leads to earliness of activities, costs such as maintenance and warehousing costs are imposed on the manufacturing sector, and if they suffer from tardiness, they will be fined, which will result in the loss of the customer and the order. For this reason, this study seeks to minimize costs so that the organization can invest elsewhere. It is noteworthy that in the real world, we are faced with issues that increase processing time due to delays in starting time. As a result, when scheduling and planning to do things, it has to consider the limitations associated with depreciating things. In addition, in the research, the problem of unrelated parallel machine to the limitation of activity correction rate along with depreciation and prerequisite of activities has been investigated. Therefore, according to the real world, a model of minimizing the total weight of tardiness and earliness has been presented with the limitation of depreciation of works and the rate of improvement of activity and prerequisites of works.

In this study, a unrelated parallel machine scheduling problem with time-dependent depreciation and several activity correction rates is stated, which is a linear function of the interval between the start and end time of the last rate-modifying activity (RMA). The actual start time processing rate is the work that can be done for the main processing time *via* RMA. At the same time, the scheduling of RMA jobs, numbers, and locations is specified to minimize completion time. To solve this problem, a mixed integer programming model is developed to find an optimal solution. The issue of unrelated parallel machine is expressed by time-dependent depreciation and several improvement rates.

Material and Methods

Problem Hypothesis

1. Prerequisites are allowed.
2. Things are independent of each other.
3. A number of jobs are placed in a cluster (time frame).
4. Each job is assigned to only one cluster (time frame).
5. Each cluster of jobs can be assigned to a single machine.
6. All jobs are available in zero time.
7. The main processing time and depreciation rate are pre-determined amounts.
8. Machines are parallel and unrelated.
9. Delivery time is also a pre-determined factor.
10. It is not possible to fail at things.
11. The effect of depreciation is considered on processing time.
12. The impact of rate-modifying activity (RMA) is also considered.

Proposed Math Model

In this section, the proposed model is shown with the approach of a nonlinear problem of mixed integer programming. The following are the indices, parameters, decision variables, objective function, and its limitations.

Model input indices and parameters

i & j: Index of jobs i & j = 1, 2, 3,., n
k: Index of machines k = 1, 2, 3,., m
l: Cluster index (time frame) l = 1, 2, 3,., n
M: a large number
e_i: cost earliness work i
t_i: Labor delay cost i
du_i: delivery date i
p_{ik}: work processing time i on k machine
d_{ik}: work depreciation time i on k machine
r_k: time rate of activity modification in k machine
G (i,j): The set G is a set of 0 and 1 that includes the prerequisites of the work so that the work i is a prerequisite for the work j.

Decision variables

MC_k: This variable calculates the time it takes to complete jobs on the k machine.
BC_{kl}: This variable indicates the time of completion of the l time cluster in the k machine.
W_{ik}: The distance between the start time of i and the end time of the last rate-modifying activity in the k machine.
E_i: It's time for earliness of i job.
T_i: This variable specifies the tardiness duration of work i according to its delivery date.
C_i: Indicates the time of completion of each work i.
C_{max}: Specifies the longest time to complete all jobs despite the rate-modifying activities in the machine.
X_{ijl}: Expresses the assignment of each job i and j in the time cluster l.
Y_{kl}: This variable indicates the time cluster of l to k machine.
Z_{ik}: Indicates that each work i must be assigned to k machine.
A_{ikl}: This is a positive variable that links the variables of assigning work i to cluster l, cluster l to machine k, and work i to machine k.

Objective function

$$\text{Min } Z=\sum_{i=1}^{n} e_i E_i + \sum_{i=1}^{n} t_i T_i \quad (1)$$

The objective function indicates minimizing the total tardiness and earliness of the work weight.

Sensitivity analysis (SA)

It is the study of the variability of output variables from the input variables of a statistical model. In other words, it is a method for changing the inputs of a statistical model in an organized (systematic) way that can predict the effects of these changes on the output of the model. In this section, we analyze the sensitivity of different parameters of the problem. First, we model the problem with the following data in Gomez.

Problem 1

In these data, the depreciation rate is a significant number. The limitations of the prerequisites in the problem are as follows:

G= {(1,2)}

According to the above data, the following values are obtained and the order of placing the works on the machines is as follows.

Cmax=229.79
Objective value=4499.17

Machine2	Job1	Job2	RMA	Job3

Machine2	Job4

Arrange the allocation of jobs to machines in the first problem

Problem 2

Now we lower the depreciation rate, increase the processing time of the work on one of the machines, and evaluate the results and the duration of the rate-modifying activity of each machine and the delivery time of each work and the cost of tardiness and earliness are considered unchanged. The limitations of the prerequisites in the problem are as follows:

G={(1,2)}
Cmax=135.53
Objective value=2850.83

Machine1	Job4	RMA	Job3

Machine2	Job1	Job2

Arrange the allocation of jobs to machines in the second problem

According to the above data, the C_{max} value has not changed, but the objective function of the problem is improve, which is minimized here, so the lower the depreciation rate, the better the machine.

Problem 3

Now we reduce the time of rate-modifying activity of each machine, consider the depreciation rate as the second problem, and present the results. The limitations of the prerequisites in the problem are as follows:

G= {(1,2)}
Cmax=141.5
Objective value:2691.5

Machine1	Job3	RMA	Job4

Machine2	Job1	Job2

Arrange the allocation of jobs to machines in the third problem

Problem 4

With regard to the third problem, here we will increase the cost of tardiness and earliness of works. The limitations of the prerequisites in the problem are as follows:

G={(1,2)}
Cmax=150.5

Objective value=3046

Machine1	Job3	RMA	Job4

Machine2	Job1	Job2

Arrange the allocation of jobs to machines in the fourth problem

Problem 5

This will change the delivery time. First, we reduce the delivery time according to the previous problem and then increase it.

Increase delivery time

The limitations of the prerequisites in the problem are as follows:

G= {(1,2)}
Cmax=173.3
Objective value=3436.6

Machine1	Job4	RMA	Job3

Machine2	Job1	Job2

Arrange the allocation of jobs to machines in the fifth problem

Reduce delivery time

In this case, the reduction in the delivery time of the works is evaluated.

The limitations of the prerequisites in the problem are as follows:

G= {(1,2)}
Cmax=132.6
Objective value=3463.2

Machine1	Job3	RMA	Job4

Machine2	Job1	Job2

Arrange the allocation of jobs to machines in the fifth problem

Problem 6

A change in prerequisites is considered.

The limitations of the prerequisites in the problem are as follows:

G= {(1,2,3,4)}
Cmax=391.41
Objective value=12862.79

Machine1	Job1	Job2	Job3	Job4

Machine 2	Nothing has been allocated.

Arrange the allocation of jobs to machines in the sixth problem

Implementation of genetic algorithms

In order to solve the problem of unrelated parallel machine scheduling, genetic algorithm has been proposed by considering the depreciation rate of works and modifying activities by considering the prerequisite between tasks with the aim of minimizing the total weight of tardiness and earliness costs. The quasi-code of the proposed algorithm is shown in Figure (1).

```
Procedure Genetic
Generate Population
LOOP
Calculate Fitness Value of Each Chromosome
Select some Chromosomes
Crossover the Selected Chromosomes
Mutate the Chromosomes
Replace Current Population with Offspring
UNTIL stop criterion is not satisfied
END
```

Figure 1. Quasi-code of genetic algorithm

The details of the implementation steps of the problem genetic algorithm are as follows.

The most important parts of the genetic algorithm are determining the chromosome structure (displaying the answer) and the neighborhood structure, which includes intersections and mutations, and is a function of evaluation. The structure of the chromosome must be determined in such a way that it includes all the variables of the model as much as possible, and on the one hand, many of the model's limitations are included in this structure. On the other hand, this structure must be such that we can easily apply intersection and mutation operators to it. The structure of the intersection and the mutation must also be determined in such a way that we can examine the space of the answer almost completely and have the ability to produce quality answers using these two operators. The main parts of the genetic algorithm are described in full below.

Chromosome structure

The chromosome designed for this problem has three parts. The first part has three rows and I column, the second part of the answer display has 1 row and I column, and the third part of the answer display has one row and the L column. The first part is to determine the assignment of each job to a time cluster and the time interval between the start and end time of the last rate-modifying activity and to determine the sequence of processing jobs on each machine. The second part of the answer display is to determine the allocation of each time cluster to one of the machines. Figure (2) shows an example of a chromosome designed for 5 jobs, 2 machines, 5 time clusters. According to the first part of Chromosome Figure (2), it can be seen that task 1 is assigned to time cluster 5 and task 2 to time cluster 4 and so on. According to the first part of chromosome in Figure (2), it can be seen that job 1 is assigned to time cluster 5 and job 2 to time cluster 4, and so on. Based on the third part of the chromosome, it is also observed that time cluster 1 is assigned to machine 2, time cluster 2 to machine 2, and so on. Therefore, based on these two parts of the chromosome, it can be determined that job 1 is assigned to machine 2 (considering that job 1 is assigned to time 5 cluster and time 5 cluster is assigned to machine 2) and job 2 is assigned to machine 1. In the same way, it is possible to determine which machines are used for the rest of the jobs. After determining which machine each is assigned to, the answer can be determined based on the second row of the first part of the display, which shows the time interval between the start and end of the last rate-modifying activity on the machine that processes that activity.

The first part of the chromosome	I_1	I_2	I_3	I_4	I_5
	5	4	1	3	2
	22.68	24.66	24.16	24.83	23.63

The second part of the chromosome	4	1	5	2	3

The third part of the chromosome	L_1	L_2	L_3	L_4	L_5
	2	1	2	1	2

Figure 2. Example of chromosome structure

The second part of the chromosome also shows the sequence of job processing on machines. For example, based on the first and third sections of the chromosome, it is observed that activities 1, 3, and 5 are processed on machine 2. The sequence of processing these three activities on machine 2 is based on the second part of chromosomes 1, 3 and 5.

Selection of primary population

Evaluation function

Using the chromosome described, the model variables are easily computed. In other words, the variables of allocating time clusters to machines, assigning jobs to time clusters, assigning jobs to machines, and sequencing the performance of jobs on machines can be calculated. Variables related to job completion and tardiness and earliness times can be calculated using relationships in constraints. The evaluation function in the proposed algorithm is the same as the objective function of the mathematical model.

Parent selection strategy

Genetic operators

Intersection

The intersection operator increases the scattering of the responses and examines the response space extensively. In this algorithm, a point intersection is used to produce children. In this method, after selecting two parents to perform the intersection, first randomly select a point from the chromosome structure as the cutting point, and then change the cut-off point on the right side and have two children. In this way, the children produced use the information of both their parents. Figure 3 shows an example of this type of intersection (randomly selected point 4 as the cutting point).

Intersection point

First parent	4 24.66	1 24.16	3 24.83	2 23.63	5 22.56	8 22.75	7 20.19	6 23.67	9 21.12
Second parent	8 23.08	3 22.48	1 20.45	4 23.07	5 21.38	9 22.57	8 24.22	4 20.98	7 23.23
First child	4 24.66	1 24.16	3 24.83	2 23.63	5 22.56	8 22.75	7 20.19	6 23.67	9 21.12
Second child	8 23.08	3 22.48	1 20.45	4 23.07	5 21.38	9 22.57	8 24.22	4 20.98	7 23.23

Figure 3. An example of an intersection operator

On other parts of the answer display, a one-point intersection is applied as above.

Mutation

As stated, this operator makes the probability of searching anywhere in space never zero. In other words, regardless of the other members in the population, it makes small changes to the chromosome so that, if possible, it can improve the good responses found in the algorithm during the optimization process. In this algorithm, two methods of swap and reversion mutation have been used.

Swap mutation

In this type of mutation, first two columns of the primary chromosome structure are randomly selected and the values in those two columns are changed. Figure (4) shows an example of this type of mutation.

Primary chromosome	9 21.12	4 24.66	1 24.16	3 24.83	2 23.63	5 22.56	8 22.75	7 20.19	6 23.67	9 21.12
Secondary chromosome	9 21.12	4 24.66	1 24.16	7 20.19	2 23.63	5 22.56	8 22.75	3 24.83	6 23.67	9 21.12

Figure 4. An example of a swap mutation

In Figure 4, columns 4 and 8 are randomly selected and replaced. For example, in the primary chromosome, Team 4 first offers Service 1 to Customer 1 and then goes to Customer 2 and provides Service 3, but this change has caused Team 4 to first go to Customer 2 and provide Service 3 and then go to Customer 1 to provide Service 1. The same changes have been made for Team 2.

Reversion mutation

In this type of mutation, first two columns of the primary chromosome structure are randomly selected and the columns between the two selected columns are reversed from right to left. Figure (5) shows an example of this type of mutation.

Primary chromosome	9 21 .12	4 24 .66	1 24 .16	3 24 .83	2 23 .63	5 22 .56	8 22 .75	7 20 .19	6 23 .67	9 21 .12
Secondary chromosome	9 21 .12	4 24 .66	1 24 .16	7 20 .19	8 22 .75	5 22 .56	2 23 .63	3 24 .83	6 23 .67	9 21 .12

Figure 5. An example of a reversion mutation

According to Figure 5, columns 4 and 8 are selected as mutation points, then columns 4 to 8 are reversed from right to left and produce a secondary chromosome. On other parts of the answer display, one of the mutation methods expressed is applied randomly.

Stop Criterion

There are several criteria for stopping the algorithm, but in this algorithm, we consider the passage of a certain number of repetitions of generations as the criterion for stopping the algorithm.

Computational results

In order to evaluate the performance of the proposed algorithm in this study, 20 experimental problems including 8 problems in small dimensions, 6 problems in medium dimensions and 6 problems in large dimensions were generated randomly and the results of solving problems are presented in the following tables. This algorithm performs all of these problems 5 times, and then the best answer is obtained, the average of the answers and the computational time related to the best answer are reported. The proposed algorithm is coded in the MATLAB R2014a programming environment and implemented on a personal computer with specified specifications (CPU core i7- 2.4 GHz & RAM 16 GB). In addition, in order to better evaluate the proposed algorithm, all small-scale problems were solved with the aim of obtaining optimal global responses by Gamz software.

In small-scale problems, all samples produced by the genetic algorithm are solved and the results are compared with the optimal answers obtained from solving the problems by Gamz software. The results of these samples are shown in Table (1) and as a comparison in Figure (6). As is clear from the results, the genetic algorithm is in most cases able to find the optimal answers at a given computational time. In addition, the error percentage index (PRE) has been used to more accurately analyze the performance of the genetic algorithm for small-scale problems. The reason for using the PRE index for small-scale problems is that in these problems the optimal answer is calculated by a precise solution approach (Gamz). The PRE index for each sample is calculated as follows:

$$PRE=\frac{sol_{avg}-sol_{opt}}{sol_{opt}} \times 100\% \quad (2)$$

In this equation, the values of sol_{avg} and sol_{opt} indicate the mean of the results obtained for a sample by the genetic algorithm and the optimal answer obtained by Gamz, respectively. The results of the PRE values for the genetic algorithm are shown in Table (2). As it is clear from the results, the average distance of the answers from the optimal answer is not a significant distance, which shows the job efficiency and effectiveness of the proposed algorithms in solving small problems.

Table 1. Computational results from solving small-scale problems

				gams		GA		
TEST PROBLEM	N	M	L	gams	CPU(S)	GA	AVERAGE	CPU(S)
1	6	2	3	2309	1.546	2309	2309	17.49
2	6	3	3	2309	3.344	2309	2309	17.732
3	7	3	4	2537	5.451	2537	2537.2	19.25
4	8	4	4	3521	14.745	3521	3521.2	21.819
5	9	3	5	3803	11.933	3803	3803.4	21.96
6	10	4	6	4129	46.27	4129	4129.2	24.119
7	10	5	7	4129	02:38.3	4129	4129.2	23.397
8	11	6	8	4308	02:46.4	4308	4308.2	25.105

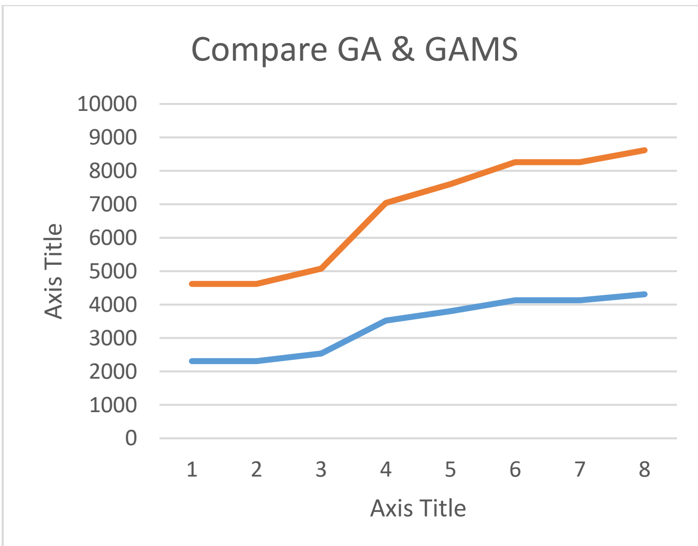

Figure 6. Comparison of algorithm performance with accurate solution for small problems

Table 2. The PRE value of the genetic algorithm for small-scale problems

TEST PROBLEM	N	M	L	PRE
1	6	2	3	0
2	6	3	3	0
3	7	3	4	0.007883
4	8	4	4	0.00568
5	9	3	5	0.010518
6	10	4	6	0.004844
7	10	5	7	0.004844
8	11	6	8	0.004643

On the other hand, in order to evaluate the performance of the genetic algorithm in solving larger problems with real dimensions, experiments have been implemented using medium and large dimensions and their results have been presented in Tables (3) and (4). In addition, a relative deviation index (RPD) index has been used to accurately analyze the performance of the genetic algorithm for medium- and large-scale problems.

$$RPD = \frac{sol_{avg} - sol_{min}}{sol_{min}} \times 100\% \quad (3)$$

The reason for using the RPD criterion for large and medium problems is that there is no definite optimal answer in these problems. Tables (5) and (6) show the RPD values of the genetic algorithm for medium- and large-scale problems.

Table 3. Computational results from solving medium-scaled problems

				gams		GA		
TEST PROBLEM	N	M	L	BEST	CPU(S)	BEST	AVERAGE	CPU(S)
1	13	6	6	5047	15:58.8	5526	5526.2	27.921
2	15	6	8	8830	07:03.1	8830	8830.2	30.649
3	15	7	9	9106	17:05.0	8830	8830.2	31.623
4	17	7	9	11220	16:52.0	11220	11220.2.2	33.744
5	17	8	10	11220	16:34.019	10536	10536.4	33.426
6	20	8	9	13421	16:37.49	14936	14936.2	37.952

Table 4. Computational results from solving large-scaled problems

				GA		
TEST PROBLEM	N	M	L	BEST	AVERAGE	CPU(S)
1	20	10	15	13344	13344.2	38.024
2	25	10	15	22733	22733.2	45.57
3	30	13	20	32941	32941.2	53.004
4	40	14	30	56159	56159.2	67.231
5	50	15	35	80023	80023.2	86.771
6	60	20	45	135380	135380.2	86.271

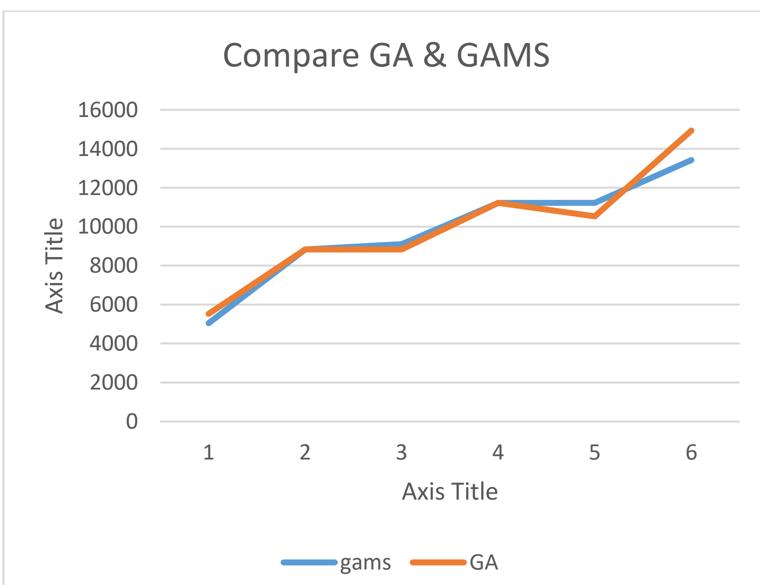

Figure 7. Comparison of algorithm performance with accurate solution for medium problems

Table 5. The PRE value of the genetic algorithm for medium-scale problems

TEST PROBLEM	N	M	L	PRE
1	13	6	6	9.494749356
2	15	6	8	0.002265006
3	15	7	9	-3.02877224
4	17	7	9	0.001782531
5	17	8	10	-6.09447415
6	20	8	9	11.29125997

Table 6. The value of the RPD of the genetic algorithm for medium-sized problems

TEST PROBLEM	N	M	L	RPD
1	13	6	6	0.003619254
2	15	6	8	0.002265006
3	15	7	9	0.002265006
4	17	7	9	0.001782531
5	17	8	10	0.001898254
6	20	8	9	0.002678093

Table 7. The value of the RPD of the genetic algorithm for large-sized problems

TEST PROBLEM	N	M	L	RPD
1	20	10	15	0.001498801
2	25	10	15	0.000879778
3	30	13	20	0.000607146
4	40	14	30	0.000356132
5	50	15	35	0.000499856
6	60	20	45	0.000147732

Discussion and Conclusion

The problem of unrelated parallel machine scheduling, considering the depreciation of the job and the rate-modifying activity, is explained as follows: a set of n jobs that are to be processed on a machine m that are parallel and unrelated, each job can only be processed on one machine and it is not possible to break the jobs. The problem also has the ability to cluster so that each cluster can range from one job to several jobs, and the end time of the job cluster is when all the jobs in the cluster are finished. Each machine is assigned a time interval for rate-modifying activities. Rate-modifying activity (RMA) is never set to zero, and is only placed on the machine when the cluster is complete, and the rate-modifying activity is applied until the next cluster is placed on the machine, the number of rate-modifying activities on a device is always one unit less than the number of clusters assigned to the device. With this explanation, the duration of rate-modifying activity in a machine can be divided among the allocated clusters, which varies according to the time of the start of the next cluster, but the total time-rate-modifying activity of a device is known as an input parameter. The model is based on the research of Young Bee Woo, San Wong, and Beung (2017). In this study, according to the above theory, tardiness and earliness fines have been investigated, taking into account the prerequisite limitation, because the tardiness and earliness in production systems plays an important role in increasing costs, if the job is delivered later than the delivery date, it suffers from tardiness and costs such as the cost of losing the order and the customer, and if the delivery time expires earlier, the organization will incur costs such as maintenance costs, warehousing, and so on. There is also the issue of prerequisites in most production units, and industrial, service, and commercial environments in the real world, so the issue raised by Young Bee Woo has been studied with the aim and limitations mentioned, and the results are also expressed using a genetic algorithm.

In this study, the problem of unrelated parallel machine scheduling with job depreciation limitations and rate-modifying activity (RMA) and the need for pre-jobs with the aim of minimizing the total weight of earliness and tardiness job costs was examined and a nonlinear programming model of the integer number was presented to model it. Based on studies conducted in this field, it has been proven that the issue under study is Np-hard issues. Therefore, it is not possible to solve this problem with real dimensions by reasonable computational time, and it is necessary to use the basic job and meta-basic algorithms to solve the problem in a suitable computational time. Therefore, in this research, genetic innovative algorithm has been used to solve the problem. In order to evaluate the performance of the proposed algorithm, a set of experimental problems in small, medium and large dimensions was designed and the results were examined in the form of different analyzes. The results of small and medium-sized problems in which the proposed algorithm was compared with the answers obtained from the precision solution method (Gams) show that the proposed algorithm is in most cases able to find the optimal answers at a suitable computational time. In addition, the analysis of computational results from large-scale problem solving suggests that the genetic algorithm can provide near-optimal answers with appropriate computational time. In this study, it was observed that the answers produced by the genetic algorithm with a small error rate compared to the exact solution to the production of the answer are close to optimal and less time than the exact solution to solve the problem.

In order to conduct future research, other issues can be considered in this area, some of which are presented below. Since in most cases the effect of deterrence is associated with the effect of learning, the learning effect can be considered as a new issue along with the assumptions of this issue. In addition to the algorithm taken from the job, it used other non-innovative algorithms or hybrid algorithms to solve the problem in order to achieve better answers and less computational time for the problem under study.

Since the proposed model in this study is a single-objective model aimed at minimizing the total weight of tardiness and earliness job costs, we can examine this model in future research using multiple optimization criteria**.**

References

Do Chung, B., & Kim, B. S. (2016). A hybrid genetic algorithm with two-stage dispatching heuristic for a machine scheduling problem with step-deteriorating jobs and rate-modifying activities. *Computers & Industrial Engineering*, *98*, 113-124.

Gupta, J. N., & Gupta, S. K. (1988). Single facility scheduling with nonlinear processing times. *Computers & Industrial Engineering*, *14*(4), 387-393.

Joo, C. M., & Kim, B. S. (2015). Machine scheduling of time-dependent deteriorating jobs with determining the optimal number of rate modifying activities and the position of the activities. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *9*(1), JAMDSM0007-JAMDSM0007.

WOO, Y. B., & KIM, B. S. (2016). A Study on Identical Parallel Machine Scheduling with Deterioration and Rate-Modifying Activities. *ICIC express letters. Part B, Applications: an international journal of research and surveys*, *7*(10), 2117-2122.

Woo, Y. B., Jung, S., & Kim, B. S. (2017). A rule-based genetic algorithm with an improvement heuristic for unrelated parallel machine scheduling problem with time-dependent deterioration and multiple rate-modifying activities. *Computers & Industrial Engineering*, *109*, 179-190.

Yang, S. J. (2013). Unrelated parallel-machine scheduling with deterioration effects and deteriorating multi-maintenance activities for minimizing the total completion time. *Applied Mathematical Modelling*, *37*(5), 2995-3005.

Zhang, X., Yin, Y., & Wu, C. C. (2017). Scheduling with non-decreasing deterioration jobs and variable maintenance activities on a single machine. *Engineering Optimization*, *49*(1), 84-97.

Zhu, H., Li, M., & Zhou, Z. (2015). Machine scheduling with deteriorating and resource-dependent maintenance activity. *Computers & Industrial Engineering*, *88*, 479-486.