COORDINATED OPTIMIZATION OF PRODUCTION SCHEDULING AND MAINTENANCE ACTIVITIES WITH MACHINE RELIABILITY DETERIORATION

CHAOMING HU, XIAOFEI QIAN*, SHAOJUN LU* AND XINBAO LIU

School of Management, Hefei University of Technology
193 Tunxi Road, Hefei City, Anhui Province, China

PANOS M PARDALOS

Center for Applied Optimization, Department of Industrial and Systems Engineering
University of Florida, Gainesville, USA

(Communicated by Oleg Prokopyev)

Abstract. In this paper, we investigate a coordinated optimization problem of production and maintenance where the machine reliability decreases with the use of the machine. Lower reliability means the machine is more likely to fail during the production stage. In the event of a machine failure, corrective maintenance (CM) of the machine is required, and the CM of the machine will cause a certain cost. Preventive maintenance (PM) can improve machine reliability and reduce machine failures during the production stage, but it will also cause a certain cost. To minimize the total maintenance cost, we must determine an appropriate PM plan to balance these two types of maintenance. In addition, the tardiness cost of jobs is also considered, which is affected not only by the processing sequence of jobs but also by the PM decision. The objective is to find the optimal job processing sequence and the optimal PM plan to minimize the total expected cost. To solve the proposed problem, an improved grey wolf optimizer (IGWO) algorithm is proposed. Experimental results show that the IGWO algorithm outperforms GA, VNS, TS, and standard GWO in optimization and computational stability.

1. Introduction. Machine reliability management is highly valued by manufacturing enterprises since the high reliability of the machine is becoming one of the key factors to make these enterprises competitive in fierce business competition. In many realistic manufacturing situations, machine reliability is subject to deterioration with the use of the machine, resulting in an increased probability of machine failure during the production stage. Once a machine failure happens, corrective maintenance (CM) of the machine is required. Too many machine failures will make enterprises bear a high maintenance cost. Preventive maintenance (PM) on the machine can effectively improve machine reliability and reduce machine failure frequency during the production stage. But the PM of the machine also takes a certain cost. Thus, enterprises have to develop a reasonable PM plan to balance...
these two types of maintenance to minimize the total maintenance cost. Under this background, researchers have done a lot of studies on preventive maintenance optimization, such as Jaturonnatee et al. (2006), Das et al. (2007), and Peng et al. (2010). However, in these studies on preventive maintenance optimization, researchers often ignore the influence of maintenance on the production process. Maintenance and production are not independent of each other since maintenance takes up the time that is originally used for production. Once the maintenance plan and the production schedule are treated separately, it will lead to conflicts between maintenance and production. Therefore, the problem of co-optimization of production and maintenance attracted the attention of researchers.

The studies on coordinated optimization of production and maintenance can be broadly divided into two categories. The first category can be called machine-unavailability-based production scheduling, where the PM plan is given in advance and the machine is not available for production during PMs. The second category can be called joint production and maintenance scheduling, in which the production scheduling and PM planning are determined simultaneously. The first type of problem was extensively investigated by researchers. As early as the end of the last century, Lee (1996) studied production scheduling problems with consideration of maintenance activities. In his study, the time period for PM was known in advance, and the machine was unavailable during the PM. Low et al. (2010) investigated a single machine scheduling problem with machine availability constraints, where machine unavailability was caused by periodic PM. They proved that the proposed problem was strong NP-hard and then presented an efficient heuristic algorithm. Additionally, Mati (2010) studied a job-shop scheduling problem with the objective to minimize the makespan where the machine was unavailable during the PM. To solve the problem, a taboo thresholding heuristic is developed. Some other related studies include Cheng et al. (2003), Liu et al. (2016), Li et al. (2017), Kong et al. (2020), and Perez-Gonzalez et al. (2020). However, such production scheduling based on deterministic PM lacks flexibility, and it is common to observe the phenomenon that some machines are awaiting maintenance while jobs are waiting to be processed by these machines under this type of scheduling.

Hence, the second type of problem attracted more attention from researchers in recent years. Berrichi et al. (2010) studied a joint production and preventive maintenance scheduling problem under the parallel machine environment and aimed at finding a compromise solution between the maximum completion time and the system availability. Mokhtari et al. (2012) did a study similar to Berrichi et al. (2010) and considered multiple-level of PM services. Their objective was to find an optimal schedule to minimize the system unavailability under the constraint of makespan. Lu et al. (2015) presented a joint production and maintenance scheduling problem on a single machine, in which the machine was prone to failure during the processing of jobs and the failure function was based on Weibull probability distribution. They aimed at finding a robust and stable integrated schedule of production and PM. Benmansour et al. (2011) investigated a stochastic scheduling problem on a single machine where the failure rate followed Erlang distribution and the objective was to minimize the cost related to production and maintenance. Liu et al. (2018) studied a single-machine-based joint optimization problem of maintenance and production with consideration of multiple-level PM services. An et al. (2020) proposed a flexible job-shop scheduling problem, where the actual processing time of the job was based on the condition of the cutting tool. In their study, the PM is considered
to be imperfect, meaning that the condition of the cutting tool could not turn to "as good as new" after PM. Feng et al.(2018) studied a multiple-stage flexible flowshop scheduling problem where the machines deteriorate with their use. Imperfect maintenance and stochastic failure were considered in their study, and the objective was to minimize the cost related to maintenance and production. Hu et al.(2017) proposed an integrated problem of PM and job scheduling with the consideration of imperfect PM. In their study, the operating condition of jobs could affect machine deterioration speed, and hence, the completion time of jobs. Lu et al.(2018) studied an unrelated parallel machine scheduling problem with the consideration of deteriorating maintenance activities and deteriorating jobs. Their objective is to find an appropriate integrated schedule of production and maintenance to minimize the makespan. Based on the plastics production process, Wong et al.(2013) proposed a joint production and maintenance scheduling problem with multiple-level PM services and multiple resources, and the objective was to minimize the makespan. Kumar et al.(2017) studied an integrated production and maintenance scheduling problem under the environment of the parallel machine and aimed at finding an optimal integrated scheduling solution to minimize overall operations cost.

Through a review of previous studies, we find that many studies belonging to the second category do not consider unexpected failures while one of the main purposes of preventive maintenance is to reduce unexpected failures during production. Besides, in most studies, PM is assumed to be perfect while imperfect PM is more practically applicable since it describes the deterioration nature of the machine in a more precise way. Therefore, in this paper, we propose a production and maintenance coordination optimization problem that considers both unexpected failures and imperfect PM. Table 1 shows comparisons between our research and previous research.

Table 1: Main features of previous studies and ours

| Authors | Production system | Unexpected failures | PM | Objectives | Solutions |
|---------|------------------|---------------------|----|------------|-----------|
| Berrichi et al. (2010) | Parallel machine | - | Perfect | Minimize $C_{max}$ as well as the system unavailability | Multi-Objective Ant Colony Optimization approach |
| Mokhtari et al. (2012) | Parallel machine | - | Multiple-level | Minimize the total system unavailability | Population-based variable neighborhood search |
| Liu et al.(2018) | Single machine | - | Multiple-level | Minimize total cost | Genetic algorithm |
| Hu et al.(2017) | Single machine | - | Imperfect | Minimize the total cost | Enumeration method |
Lu et al. (2018) | Parallel machine | - | Perfect | Minimize makespan | A hybrid ABC-TS algorithm
---|---|---|---|---|---
Wong et al. (2013) | Parallel machine | - | Multiple-level | Minimize the makespan | Genetic algorithm
Kumar et al. (2017) | Parallel machine | - | Perfect | Minimize overall operations cost | Simulation-based Genetic Algorithm
Feng et al. (2018) | Flexible flowshop | ✓ | Imperfect | Minimize the total cost | Simulated annealing embedded genetic algorithm
Lu et al. (2015) | Single machine | ✓ | Perfect | System robustness and stability | Genetic algorithm
Benmansour et al. (2011) | Single machine | ✓ | Perfect | Minimize the cost related to production and maintenance | Heuristic algorithm
An et al. (2020) | Flexible job-shop | - | Imperfect | Minimize the makespan and total cost | Evolutionary algorithm
This paper | Single machine | ✓ | Imperfect | Minimize the total expected cost | Improved grey wolf optimizer algorithm

The main contributions of this paper can be summarized as follows:

(1) We investigate a co-optimization problem of production scheduling and maintenance planning with the consideration of preventive maintenance, corrective maintenance, and job tardiness cost.

(2) The PM of the machine is assumed to be imperfect in this paper since the imperfect PM describes the deterioration nature of the machine in a more precise way.

(3) To solve the proposed problem, we develop an improved GWO algorithm that has a strong searching ability. The rest of the paper is organized as follows. In Section 2, we list the notations used in this paper and give a detailed description of the proposed problem. In Section 3, the mathematical model of the problem is presented. Section 4 describes the proposed improved GWO algorithm, and Section
5 reports the experimental results. The last section concludes this paper and puts forward the research direction in the future.

2. Notations and problem description. The notations used in this paper are given in Table 2.

| Notation | Description |
|----------|-------------|
| $N$      | Number of jobs |
| $p_j$    | The processing time of job $j$, $j = 1, \ldots, N$ |
| $d_j$    | The due date of job $j$, $j = 1, \ldots, N$ |
| $w_j$    | Tardiness cost per unit of time of job $j$, $j = 1, \ldots, N$ |
| $c_l$    | The unit basic cost of the factory. |
| $t_r$    | The time required for performing corrective maintenance (CM). |
| $c_r$    | The cost of a CM activity. |
| $t_p$    | The time required for performing preventive maintenance (PM). |
| $c_p$    | The cost of a PM activity. |
| $x_{i,j}$| Job sequencing decision variable, and it is a zero and one variable |
| $y_{i}$  | PM decision variable, and it is a zero and one variable |
| $p_{i}$  | The processing time of the $i$th job in the sequence |
| $f_{i}(t)$| The failure rate function during the processing of the $i$th job and $t$ is the machine age (i.e. the time since the last PM) |
| $m$      | The number of PMs performed on the machine before time 0. |
| $Z_{f_{0}}$| The machine age at time 0. |
| $Z_{s_{i}}$| The machine age when starting the processing of the $i$th job |
| $Z_{f_{i}}$| The machine age when finishing the processing of the $i$th job |
| $E(N_{i})$ | The expected number of failures during the processing of the $i$th job |
| $E(C_{i})$ | The expected completion time of the $i$th job |
| $E(T_{i})$ | The expected tardiness time of the $i$th job |
| $E(TC)$  | The expected total cost. |

In a factory, there are $N$ jobs to be scheduled on a single machine. The machine can process only one job at a time and no preemption is allowed. The processing time and the due date of the job $j$ are denoted by $p_j$ and $d_j$, respectively. Once a job is finished after its due date, there will be a tardiness cost and the tardiness cost per unit of time for job $j$ is $w_j$. In our model, stochastic failure is considered. Whenever a failure happens, a corrective maintenance (CM) activity must be conducted to make the machine re-operating. The maintenance for a failure is minimal and does not change the reliability of the machine. We use $t_r$ and $c_r$ to represent the time and cost required by a CM activity respectively. The reliability is not fixed but deteriorating with the use of the machine. Lower reliability means the machine is more likely to suffer failures during the production stage, which increases both the CM cost and the tardiness cost. Hence, PM is required to improve the reliability of the machine. The PM is not allowed when a machine is processing a job. The time and cost required by a PM activity are given as $t_p$ and $c_p$. It should be noted that even when the machine is under maintenance, the factory must pay the basic cost, which includes labor cost, site rental cost, machine rental cost, etc. The unit basic cost of the factory is denoted as $c_l$. 
Different from many other related studies, PM is assumed to be imperfect in this paper. To represent the characteristic of imperfect PM, a failure rate adjustment model (Lin et al, 2000) is introduced into our problem. The model assumes that the machine failure rate returns to 0 after PM and then, the failure rate increases more quickly than before. The model can be expressed by the following function:

\[ f(t, l) = \theta f(t, l - 1) = \theta^l f(t, 0) \] 

where \( \theta \) is a constant greater than 1, \( t \) is the machine age, and \( l \) means the number of PMs the machine undergoes. Figure 1 is the diagram of the failure rate adjustment model.

The machine age at time 0 is \( Z_{[0]} \), and the machine underwent \( m \) times of PM before time 0. Similar to the researches of Laohanan et al (2018), Ramos et al (2018), and Idoniboyeobu et al (2018), we assume that the initial reliability of the machine follows the Weibull distribution, and thus, the initial failure rate function is given as:

\[ f(t, 0) = \frac{\beta}{\eta} \left( \frac{t}{\eta} \right)^{\beta - 1} \] 

where \( t \) is the machine age, \( \beta \) is the shape parameter, and \( \eta \) is the scale parameter. The objective is to find a joint schedule of production and maintenance to minimize the total expected cost which includes the basic cost of the factory, PM cost, CM cost, and tardiness cost.

3. **Problem analysis and modeling.** To solve such a problem, two decisions should be made: the processing sequence of jobs and the time to perform PM. Let \( x_{[i]j} \) be the job sequencing decision variable, and \( y_{[i]} \) be the PM decision variable:

\[ x_{[i]j} = \begin{cases} 1, & \text{if the } i\text{th job in the sequence is job } j \\ 0, & \text{otherwise} \end{cases} \]  

\[ y_{[i]} = \begin{cases} 1, & \text{if PM is performed before the } i\text{th job} \\ 0, & \text{otherwise} \end{cases} \]
Thus the processing time of the $i$th job is:

$$p_i = \sum_{j=1}^{N} x_{i,j} P_j$$  \hspace{1cm} (3.3)$$

To calculate the expected completion time of each job, we need to know the expected number of failures during the processing of the $i$th job. According to formulas (2.1) and (2.2), the failure rate function during the processing of the $i$th job is:

$$f_i(t) = \theta m + \sum_{k=1}^{i} y_k \beta \eta^{\beta-1} \left( \frac{t}{\eta} \right)^{\beta-1} \hspace{1cm} (3.4)$$

where $m + \sum_{k=1}^{i} y_k$ is the total number of PMs the machine undergoes before the processing of the $i$th job. According to Fitouhi et al. (2012), for a given failure rate function $f(t)$, the expected number of failures during the interval $(a,b)$ can be calculated by $\int_{a}^{b} f(t) dt$. Hence, we need to know the machine age at the starting and ending of the processing of the $i$th job. Figure 2 clearly shows how the machine age changes.

![Figure 2. Schematic diagram of the machine age](image)

Since the CM for a failure does not change the machine age, the machine age at the start and end of the processing of the $i$th job can be obtained by the recursive formula:

$$\begin{cases}
Z^s_i = Z^s_{i-1}(1 - y_i) \\
Z^f_i = Z^s_i + p_i
\end{cases} \hspace{1cm} i = 1, ..., N \hspace{1cm} (3.5)$$

Then, the expected number of failures during the processing of the $i$th job is:

$$E(N_i) = \int_{Z^s_i}^{Z^f_i} f_i(t) dt = \int_{Z^s_i}^{Z^f_i} \theta m + \sum_{k=1}^{i} y_k \beta \left( \frac{t}{\eta} \right)^{\beta-1} dt$$

$$= \theta m + \sum_{k=1}^{i} y_k \beta \left( \frac{Z^s_i + p_i}{\eta} \right)^{\beta} - \left( \frac{Z^s_i}{\eta} \right)^{\beta} \hspace{1cm} (3.6)$$
Further, the expected completion time of the $i$th job can be calculated by the following recursive formula:

$$
\begin{cases}
E(C_{[i]}) = y_{[i]}t_p + E(N_{[i]})t_r + p_{[i]} \\
E(C_{[i]}) = y_{[i]}t_p + E(N_{[i]})t_r + p_{[i]} + E(C_{[i-1]}) & i = 2, ..., N
\end{cases}
$$

(3.7)

And the expected tardiness time of the $i$th job is:

$$E(T_{[i]}) = \max(0, E(C_{[i]}) - \sum_{j=1}^{N} x_{[i]j} \ast d_j)$$

(3.8)

Based on the above analysis, the model can be stated as:

$$\min E(TC) = c_1E(C_{[N]}) + \sum_{i=1}^{N} (E(T_{[i]})) \sum_{i=1}^{N} x_{[i]j} w_j) + c_p \sum_{i=1}^{N} y_{[i]} + c_r \sum_{i=1}^{N} E(N_{[i]})$$

(3.9)

$$\sum_{i=1}^{N} x_{[i]j} = 1, j = 1, ..., N$$

(3.10)

$$\sum_{j=1}^{N} x_{[i]j} = 1, i = 1, ..., N$$

(3.11)

$$x_{[i]j} = \begin{cases} 
1 & \\
0, i, j = 1, ..., N
\end{cases}$$

(3.12)

$$y_{[i]} = \begin{cases} 
1 & \\
0, i, j = 1, ..., N
\end{cases}$$

(3.13)

The objective (3.9) consists of four parts, the first part is the basic cost, the second part is the tardiness cost, the third part denotes the PM cost and the fourth part represents the CM cost. $E(T_{[i]})$ and $E(N_{[i]})$ are determined by the decision variables $x_{[i]j}$ and $y_{[i]}$, and can be calculated by formulas (3.4)-(3.8). Formula (3.10) indicates that job $j$ can only be processed once and formula (3.11) indicates that the machine can process only one job at a time.

4. Improved grey wolf optimizer. If the stochastic failures, corrective maintenance, and preventive maintenance are not considered, then the proposed problem can be reduced to a classic scheduling problem $1||\sum w_i \ast T_i$ which is proved to be NP-hard (Lenstra et al, 1977). Thus, the problem proposed in this paper is also NP-hard. To solve the problem in a reasonable time, we put forward an improved grey wolf optimizer (IGWO) algorithm. Also, some computational experiments are conducted to test the performance of the proposed algorithm.

4.1. Encoding method. In this paper, a solution is represented by a vector $X = (x_1, x_2, x_3, \ldots, x_N)$ where $x_j$ is a real number. All the jobs are sequenced in ascending order of the absolute value of $x_j$, and preventive maintenance will be carried out before the processing of the job $j$ if $x_j < 0$. To clearly illustrate the encoding method, an example of 6 jobs is shown in Figure.3.
4.2. **Original grey wolf optimizer.** The Grey Wolf Optimizer (GWO) algorithm, invented by Mirjalili et al. (2014), is inspired by the population hierarchy and predation process of the grey wolf. Since the GWO algorithm was proposed, it has been applied to solve optimization problems by many researchers due to its good performance, such as Komaki et al. (2015), Lu et al. (2017), and Sattar et al. (2020). The main concepts and steps of the GWO algorithm are as follows:

1) Social Hierarchy
Firstly, construct the gray wolf’s social hierarchy. Calculate the fitness of each wolf in the population, and set the best three as $\alpha$, $\beta$, and $\gamma$, while set the remaining as $\omega$. During the iteration process, the wolves update their positions according to the three best solutions ($\alpha$, $\beta$, and $\gamma$) in each generation of the population.

2) Encircling Prey
When the wolves search for prey, they will gradually approach the prey and then surround the prey. The mathematical model of the process is as follows:

\[
D = C \cdot X_{p}(t) - X(t) \quad (4.1)
\]

\[
X(t + 1) = X_{p}(t) - A \cdot D \quad (4.2)
\]

where $t$ is the current iteration, $X(t)$ is the current position vector of the gray wolf, and $A, C$ are coefficient vectors. The coefficients $A$ and $C$ can be calculated as follows

\[
A = 2a \cdot r_{1} - a \quad (4.3)
\]

\[
C = 2 \cdot r_{2} \quad (4.4)
\]

where $a$ is linearly decreased from 2 to 0 throughout iterations, and $r_{1}, r_{2}$ are random vectors in $[0,1]$.

3) Hunting
In GWO, $\alpha$, $\beta$, and $\gamma$ are thought to have a strong ability to recognize the location of potential prey and they can instruct other wolves to search for prey. Therefore, during the iteration, the best three gray wolves in the current population ($\alpha$, $\beta$, and $\gamma$) are saved, and other gray wolves will update their positions based on the position of the best three. The mathematical model of this behavior can be expressed as follows:

\[
D_{\alpha} = C_{1} \cdot X_{\alpha} - X, \quad D_{\beta} = C_{2} \cdot X_{\beta} - X, \quad D_{\gamma} = C_{3} \cdot X_{\gamma} - X \quad (4.5)
\]

\[
X_{1} = X_{\alpha} - A_{1} \cdot D_{\alpha}, \quad X_{2} = X_{\beta} - A_{2} \cdot D_{\beta}, \quad X_{3} = X_{\gamma} - A_{3} \cdot D_{\gamma} \quad (4.6)
\]
where \( D_\alpha, D_\beta, D_\gamma \) represent the distance between the candidate wolves and the best three.

4.3. Improvement strategies. Although the GWO algorithm has shown excellent performance in solving some optimization problems, researchers find that GWO is easy to fall into local optimal when solving large-scale optimization problems. To overcome this defect of the GWO algorithm, we present two improvement strategies. One strategy is an improvement on parameter \( a \), and the other is to add a local search strategy to some wolves in each generation.

1) Strategy 1: improvement of parameter \( a \)

According to formula (4.3), we know that \( A \) is a random value in the gap \([-a, a]\). During the iteration process, the parameter \( a \) decreases linearly from 2 to 0, so the fluctuation range of \( A \) also decreases as the iteration progresses. According to formula (4.1)-(4.2), we know that when \(|A| > 1\), the search agent is forced to stay away from the prey, which ensures the global search capability of the algorithm, and when \(|A| < 1\), the search agent is forced to approach the prey, which ensures the convergence of the algorithm. Therefore, in the original GWO algorithm, when the number of iterations exceeds \( t_{max} \) (\( t_{max} \) is the maximum number of iterations), all search agents stop searching in the global scope and start to approach the prey. To enhance the global searching capability of GWO, we proposed a new nonlinearly decreasing function for parameter \( a \):

\[
a = 2 \cos \left( \frac{\pi}{2} \frac{k}{t_{max}} \right)
\]  

**Figure 4.** A comparison of two functions

Figure 4 shows the comparison between the initial function and the new function of parameter \( a \). Compared with the linear function, parameter \( a \) decreases more slowly during the early iteration under the new function. In addition, parameter \( a \) under the new function begins to be less than 1 until the number of iterations
exceeds $\frac{2}{3}t_{max}$. That is, the time for searching in the global scope becomes longer under the new function.

2) Strategy 2: local search strategy
To ensure the diversity of the population, we add a local search strategy to the original GWO algorithm. We assume that some $\omega$ wolves would update their positions by local search strategy rather than following the directions of $\alpha$, $\beta$, and $\gamma$. The local search strategy for a solution $X = (x_1, x_2, \ldots, x_N)$ is shown in Table 3.

| Step | Description |
|------|-------------|
| 1. | Set the maximum number of searches $s_{max}$ and let $s = 0$ |
| 2. | Randomly generate three unequal integer $a, b, c$ in $[1, N]$ |
| 3. | Let $t = x_a, x_a = x_b, x_b = t, \text{ and } x_c = -x_c$ and then obtain a new solution $X'$ |
| 4. | Calculate the fitness of $X'$, and let $s = s + 1$ |
| 5. | Judge whether $f(X') < f(X)$ is true, if so, let $X \leftarrow X'$ and stop the search, otherwise, go to step 6 |
| 6. | Judge whether $s \leq \epsilon \leq s_{max}$, if so, turn to Step 2, otherwise keep $X$ and stop search |

Then, the procedure and flow chart of the IGWO algorithm are obtained and illustrated in Table 4 and Figure 5.

| Line | Description |
|------|-------------|
| 1. | Initializes the grey wolf population |
| 2. | Initializes the max number of iterations $t_{max}$ and the local search probability $\epsilon$, and set $a = 2, k = 1$ |
| 3. | Calculate the fitness of all individuals |
| 4. | Set the three individuals with the best fitness as $X_\alpha, X_\beta, \text{ and } X_\gamma$ |
| 5. | While ($k < t_{max}$) |
| 6. | For each search agent |
| 7. | Generate a random probability $p$ |
| 8. | If $p \geq \epsilon$ |
| 9. | Generate two random numbers $r_1, r_2$ in $(0, 1)$ |
| 10. | Calculate the coefficients $A, C$ by formula (4.3)-(4.4) |
| 11. | Update the position of the current search agent by formula (4.5)-(4.7) |
| 12. | Else if |
| 13. | Update the position of the current search agent by Local search strategy |
| 14. | End if |
| 15. | Update the population |
| 16. | Calculate the fitness of all individuals |
| 17. | Update $X_\alpha, X_\beta, \text{ and } X_\gamma$ |
| 18. | Update $a$ by formula (4.8) |
| 19. | $k = k + 1$ |
| 20. | End while |
| 21. | Return $X_\alpha$ |
5. **Computational experiments.** In order to validate the performance of the proposed IGWO algorithm, we employ it and other four compared algorithms such as the standard GWO (Mirjalili et al., 2014), Genetic Algorithm (Golmohammadi et al., 2017), Variable Neighborhood Search Algorithm (Lei, 2015), and Tabu Search.
Algorithm (Logendran et al., 2017) to solve the proposed problem. All the experiments are implemented in python and run on a computer with Intel (R) Core (TM) i7-7770 CPU @3.60 GHz, 8GB RAM, and Windows 10 operating system. The main experimental factors and their levels are shown in Table 5.

Table 5: Experimental factors and their levels

| Factor                                      | Levels                                      |
|---------------------------------------------|---------------------------------------------|
| The total number of jobs, \( N \)          | 10,20,30,…,180,190,200                      |
| The failure rate adjustment parameter, \( \theta \) | 1.05                                       |
| The failure rate parameter, \( \beta \)    | 2                                           |
| The failure rate parameter, \( \eta \)     | 50                                          |
| The processing time of job \( j \), \( p_j \) | Uniform \[1,10\]                            |
| The due date of job \( j \), \( d_j \)     | Uniform \[1,100\]                           |
| The tardiness cost per unit of time cost of job \( j \), \( w_j \) | Uniform \[0,1\]                            |
| The unit production cost, \( c_l \)        | 1                                           |
| The time required for performing corrective maintenance, \( t_r \) | 1                                           |
| The cost of a corrective maintenance activity, \( c_r \) | 1                                           |
| The time required for performing preventive maintenance, \( t_p \) | 2                                           |
| The cost of a preventive maintenance activity, \( c_p \) | 2                                           |

5.1. **Comparison of IGWO with GWO.** To test and verify the influence of the improvement strategies on the performance improvement of the standard algorithm, we conduct a series of compared experiments where both IGWO and GWO (Mirjalili et al., 2014) algorithms are employed to solve different instances of the proposed problem. The population size is set to 10, and the initial population is identical for both algorithms. The termination condition is that the maximum number of iterations exceeds 300. The local search probability of IGWO is set as \( \varepsilon = 0.3 \), and the maximum searching number is set as \( s_{max} = 100 \ast N \). For each problem instance, each algorithm runs 10 times. Table 6 is the results generated by these two algorithms. Column 1 is the list of problem instances, columns 2-5 give the maximum (MAX), minimum (MIN), average (AVE), and variance (VAR) of total expected cost generated by GWO (Mirjalili et al., 2014) for each problem instance. Columns 6-9 report the results generated by IGWO. The better results are in bold.

Since the first three metrics (MAX, MIN, AVE) represent the performance of the algorithm in optimization and the fourth metric (VAR) represents the stability of the algorithm, we can conclude that the proposed IGWO algorithm outperforms the standard GWO concerning solution optimality and stability according to Table 6.

Table 6: Comparison of GWO and IGWO

| Instance | GWO  | IGWO  |
|----------|------|-------|
|          | MIN | MAX | AVE | VAR | MIN | MAX | AVE | VAR |
| N=10     | 72  | 78  | 76  | 0.87| 65  | 67  | 66  | 0.28|
| N=20     | 315 | 355 | 338 | 124 | 244 | 262 | 253 | 28  |
| N=30     | 595 | 641 | 615 | 263 | 420 | 479 | 442 | 198|
5.2. Comparison of IGWO with other three algorithms. Similar to that in 5.1, GA (Golmohammadi et al., 2017), VNS (Lei, 2015), and TS (Logendran et al., 2017) algorithms are also employed to solve different problem instances. The population size is set to 10, and the initial population is identical for all algorithms. The termination condition is that the maximum number of iterations exceeds 300. For fair comparisons, the recommended parameters of GA (Golmohammadi et al., 2017) VNS (Lei, 2015), and TS (Logendran et al., 2017) are used.

| N | 1495 | 1680 | 1649 | 3668 | 1330 | 1489 | 1424 | 11075 |
|---|------|------|------|------|------|------|------|------|
| N | 2462 | 2683 | 2610 | 2375 | 2311 | 2518 | 2405 | 2231 |
| N | 4888 | 5268 | 5057 | 17758 | 4881 | 4722 | 4583 | 5833 |
| N | 6760 | 7107 | 6929 | 20810 | 5909 | 6339 | 6089 | 9859 |
| N | 7037 | 7328 | 7174 | 13697 | 6179 | 6544 | 6357 | 9622 |
| N | 10781 | 11429 | 11147 | 35652 | 9768 | 10393 | 10101 | 33610 |
| N | 13256 | 14317 | 13811 | 103514 | 12511 | 13184 | 12788 | 53694 |
| N | 17281 | 18388 | 17783 | 106493 | 15711 | 16953 | 16331 | 104363 |
| N | 20102 | 21580 | 20758 | 214099 | 18108 | 19249 | 18573 | 63512 |
| N | 20919 | 22114 | 21594 | 112116 | 18600 | 19460 | 19020 | 20809 |
| N | 23002 | 24554 | 23793 | 246926 | 19369 | 20752 | 19891 | 56352 |
| N | 29341 | 30523 | 29801 | 157775 | 25098 | 25768 | 25436 | 61450 |
| N | 34321 | 36133 | 35142 | 420426 | 28998 | 30418 | 29632 | 134003 |
| N | 38393 | 40478 | 39255 | 390366 | 32914 | 34377 | 33557 | 177787 |
| N | 49190 | 51420 | 50013 | 395090 | 42772 | 44139 | 43557 | 177696 |
| N | 50416 | 52568 | 51498 | 575500 | 43267 | 44589 | 44003 | 102377 |

(a) N=10  
(b) N=20
Figure 6. (a), (t) Convergence curves for different instances.
Figure. 6(a)-Figure. 6 (t) are the convergence curves for different problem instances. The points on a convergence curve represent the average of 10 runs of a given algorithm. From Figure 6 (a) - (t), we know that the optimization performance of IGWO is better than that of GA (Golmohammadi et al., 2017), VNS (Lei, 2015), TS (Logendran et al.,1997) and standard GWO (Mirjalili et al., 2014) in any problem instance. Furthermore, the convergence speed of IGWO is also faster than other algorithms. The standard GWO algorithm tends to fall into local optimal and performs worst among these comparison algorithms. To further explore the differences among these algorithms, the relative percentage deviations (RPD) analysis (Zandieh et al., 2017) is performed. The RPD is expressed as follows:

\[ RPD = \frac{Z_A - Z_B}{Z_B} \]

where \(Z_A\) is the result obtained by a given algorithm for an instance and \(Z_B\) is the best result obtained by all comparison algorithms for the same instance. Figure.7 (a) and Figure.7 (b) give the results generated by those comparison algorithms.

![Figure 7. (a) Relative percentage deviations of AVE](image1)

![Figure 7. (b) Relative percentage deviations of VAR](image2)
From Figure 7 (a), we can find that the IGWO algorithm has the best optimization performance while the standard GWO algorithm has the worst optimization performance. There is no significant difference in optimization among GA (Golumbamsadi et al., 2016), VNS (Lei, 2015), and TS (Logendran et al., 2017) when the problem size is small (N=100). However, with the increase of the problem scale, the optimization performance of the TS (Logendran et al., 2017) algorithm outperforms other algorithms except for the IGWO. From Figure 7 (b), we can find that there is no significant difference in RPD of AVR among GA (Golumbamsadi et al., 2017), VNS (Lei, 2015), and TS (Logendran et al., 2017), and GWO (Mirjalili et al., 2014). The RPD of VAR of IGWO is significantly lower than that of other algorithms except for the instances N=10, N=40, and N=170. In other words, the computational stability of the IGWO algorithm is better than those comparison algorithms in solving the proposed problem.

6. Conclusion. In this paper, we investigate a coordinated optimization problem of production and maintenance with consideration of machine reliability deterioration, and the objective is to minimize the total cost related to production and maintenance. We prove that the proposed problem is NP-hard, and then develop an improved grey wolf optimizer (IGWO) algorithm to solve it. To verify the efficiency of the proposed algorithm, we carry out comparative experiments on different instances, and the experimental results show that the proposed algorithm outperforms GA, VNS, TS, and the standard GWO algorithm. The main limitation of this paper is that what we study is based on a single machine production system, while in reality there are many more complex production systems, such as parallel machine systems and flow-shop systems. Besides, because the production system we studied is simple, we do not consider the constraints of maintenance resources. Therefore, in future studies, we will focus on more complex production systems and consider the limitation of maintenance resources.

Acknowledgments. This work is supported by the National Key Research and Development Program of China (2019YFB1705300), the Fundamental Research Funds for the Central Universities (Nos. JZ2021HGTA0134, JZ2021HGQA0200, JZ2020HGTB0035), the National Natural Science Foundation of China (Nos. 72071056, 71922009, 71871080, 71601065, 71690235, 71501058, 71601060), and Innovative Research Groups of the National Natural Science Foundation of China (71521001), Anhui Province Natural Science Foundation (No. 1908085MG223), Base of Introducing Talents of Discipline to Universities for Optimization and Decision-making in the Manufacturing Process of Complex Product (111 projects), the Project of Key Research Institute of Humanities and Social Science in University of Anhui Province, Open Research Fund Program of Key Laboratory of Process Optimization and Intelligent Decision-making(Hefei University of Technology), Ministry of Education.

REFERENCES
[1] J. Jaturonnatee, D. N. P. Murthy and R. Boondiskulchok, Optimal preventive maintenance of leased equipment with corrective minimal repairs, European Journal of Operational Research, 174 (2006), 201–215.
[2] K. Das, R. S. Lashkari and S. Sengupta, Machine reliability and preventive maintenance planning for cellular manufacturing systems, European Journal of Operational Research, 183 (2007), 162–180.
[3] H. Peng, Q. Feng and D. W. Coit, Reliability and maintenance modeling for systems subject to multiple dependent competing failure processes, *IIE transactions*, 43 (2010), 12–22.

[4] C.-Y. Lee, Machine scheduling with an availability constraint, *Optimization Applications in Scheduling Theory. J. Global Optim.*, 9 (1996), 395–416.

[5] C. Low, M. Ji, C.-J.g Hsu and C.-T. Su, Minimizing the makespan in a single machine scheduling problems with flexible and periodic maintenance, *Appl. Math. Model.*, 34 (2010), 334–342.

[6] Y. Mati, Minimizing the makespan in the non-preemptive job-shop scheduling with limited machine availability, *Computers & Industrial Engineering*, 59 (2010), 537–543.

[7] T. C. E. Cheng and Z. Liu, Approximability of two-machine no-wait flowshop scheduling with availability constraints, *Oper. Res. Lett.*, 31 (2003), 319–322.

[8] G. Li, M. Liu, S. P. Sethi and D. Xu, Parallel-machine scheduling with machine-dependent maintenance periodic recycles, *International Journal of Production Economics*, 186 (2017), 1–7.

[9] M. Liu, S. Wang, C. Chu and F. Chu, An improved exact algorithm for single-machine scheduling to minimise the number of tardy jobs with periodic maintenance, *International Journal of Production Research*, 54 (2016), 3591–3602.

[10] M. Kong, X. Liu, J. Pei, H. Cheng and P. M. Pardalos, A BRKGA-DE algorithm for parallel-batching scheduling with deterioration and learning effects on parallel machines under preventive maintenance consideration, *Ann. Math. Artif. Intell.*, 88 (2020), 237–267.

[11] P. Perez-Gonzalez, V. Fernandez-Vigas and J. M. Framinan, Permutation flowshop scheduling with periodic maintenance and makespan objective, *Computers & Industrial Engineering*, 143 (2020), 106369.

[12] A. Berrichi, F. Yalaoui, L. Amodeo and M. Mezghiche, Bi-objective ant colony optimization approach to optimize production and maintenance scheduling, *Comput. Oper. Res.*, 37 (2010), 1584–1596.

[13] H. Mokhtari, A. Mozdighir and I. N. Kamal Abadi, A reliability/availability approach to joint production and maintenance scheduling with multiple preventive maintenance services, *International Journal of Production Research*, 50 (2012), 5906–5925.

[14] Z. Lu, W. Cui and X. Han, Integrated production and preventive maintenance scheduling for a single machine with failure uncertainty, *Computers & Industrial Engineering*, 80 (2015), 236–244.

[15] R. Rennouso, H. Alloumi, A. Artiba, S. Iassinovski and R. Pellerin, Simulation-based approach to joint production and preventive maintenance scheduling on a failure-prone machine, *Journal of Quality in Maintenance Engineering*, 17 (2011).

[16] Q. Liu, M. Dong and F. F. Chen, Single-machine-based joint optimization of predictive maintenance planning and production scheduling, *Robotics and Computer-Integrated Manufacturing*, 51 (2018), 238–247.

[17] Y. An, X. Chen, J. Zhang and Y. Li, A hybrid multi-objective evolutionary algorithm to integrate optimization of the production scheduling and imperfect cutting tool maintenance considering total energy consumption, *Journal of Cleaner Production*, 268 (2020), 121540.

[18] H. Feng, L. Xi, L. Xiao, T. Xia and E. Pan, Imperfect preventive maintenance optimization for flexible flowshop manufacturing cells considering sequence-dependent group scheduling, *Reliability Engineering & System Safety*, 176 (2018), 218–229.

[19] J. Hu and Z. Jiang, Job scheduling integrated with imperfect preventive maintenance considering time-varying operating condition, *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, (2017), 578–582.

[20] C. S. Wong, F. T. S. Chan and S. H. Chung, A joint production scheduling approach considering multiple resources and preventive maintenance tasks, *International Journal of Production Research*, 51 (2013), 883–896.

[21] S. Kumar and B. K. Lad, Integrated production and maintenance planning for parallel machine system considering cost of rejection, *Journal of the Operational Research Society*, 68 (2017), 834–846.

[22] S. Lu, X. Liu, J. Pei, M. T. Thai and P. M. Pardalos, A hybrid ABC-TS algorithm for the unrelated parallel-batching machines scheduling problem with deteriorating jobs and maintenance activity, *Applied Soft Computing*, 66 (2018), 168–182.

[23] D. Lin, M. J. Zuo and R. C. M. Yam, General sequential imperfect preventive maintenance models, *International Journal of Reliability, Quality and Safety Engineering*, 7 (2000), 253–266.
[24] M.-C. Fitouhi and M. Nourelfath, Integrating noncyclical preventive maintenance scheduling and production planning for a single machine, *International Journal of Production Economics*, 136 (2012), 344–351.

[25] S. Laohanan and D. Banjerpongchai, Dynamic programming approach to optimal maintenance scheduling of substation equipment considering life cycle cost and reliability, *International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, (2018), 388–391.

[26] P. L. Ramos, D. C. Nascimento, C. Cocolo, et al., Reliability-centered maintenance: Analyzing failure in harvest sugarcane machine using some generalizations of the Weibull distribution, *Modelling and Simulation in Engineering*, (2018), 2018.

[27] D. C. Idonibo, B. A. Wokoma and V. C. Ibanibo, Preventive maintenance for substation with aging equipment using weibull distribution, *American Journal of Engineering Research*, 7 (2018), 95–101.

[28] J. K. Lenstra, A. H. G. R. Kan and P. Brucker, Complexity of machine scheduling problems, *Ann. of Discrete Math.*, North-Holland, Amsterdam, 1 (1977), 343–362.

[29] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, 69 (2014), 46–61.

[30] G. M. Komaki and V. Kayvanfar, Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time, *Journal of Computational Science*, 8 (2015), 109–120.

[31] C. Lu, L. Gao, X. Li and S. Xiao, A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry, *Engineering Applications of Artificial Intelligence*, 57 (2017), 61–79.

[32] M. K. Sattar, A. Ahmad, S. Fayyaz, et al., Ramp rate handling strategies in dynamic economic load dispatch (DELD) problem using grey wolf optimizer (GWO), *Journal of the Chinese Institute of Engineers*, 43 (2020), 200–213.

[33] A.-M. Golmohammadi, H. Bani-Asadi, H. J. Zanjani and H. Tikani, A genetic algorithm for preemptive scheduling of a single machine, *Journal of the Chinese Institute of Engineers*, 7 (2016), 607–614.

[34] R. Logendran and A. Sonthinen, A Tabu search-based approach for scheduling job-shop type flexible manufacturing systems, *Journal of the Operational Research Society*, 48 (1997), 264–277.

[35] D. Lei, Variable neighborhood search for two-agent flow shop scheduling problem, *Computers & Industrial Engineering*, 80 (2015), 125–131.

[36] M. Zandieh, A. R. Khataomi and S. H. A. Rahmati, Flexible job shop scheduling under condition-based maintenance: Improved version of imperialist competitive algorithm, *Applied Soft Computing*, 58 (2017), 449–464.

Received March 2021; revised May 2021; early access August 2021.

E-mail address: huchaoming1994@126.com
E-mail address: qianxiaofei888@126.com
E-mail address: lushaojun@hfut.edu.cn
E-mail address: lxb@hfut.edu.cn
E-mail address: p.m.pardalos@gmail.com