Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-020-01588-9

Preventive maintenance scheduling optimization based on opportunistic production-maintenance synchronization

Li Li[1,2] · Yong Wang[1] · Kuo-Yi Lin[1,2]

Received: 24 November 2019 / Accepted: 11 May 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
Equipment maintenance is momentous for improving production efficiency, how to integrate maintenance into production to address uncertain problems has attracted considerable attention. This paper addresses a novel approach for integrating preventive maintenance (PM) into production planning of a complex manufacturing system based on availability and cost. The proposed approach relies on two phases: firstly, this study predicts required capacity of each machine through extreme learning machine algorithm. Based on analyzing historical data, the opportunistic periods are calculated for implementing PM tasks to have less impact on production and obtain the PM interval and duration. Secondly, this study obtains the scheduling planning and the least number of maintenance personnel through an improved ant colony optimization algorithm. Finally, the feasibility and benefits of this approach are investigated based on empirical study by using historical data from real manufacturing execution system and equipment maintenance system. Experimental results demonstrate the effectiveness of proposed approach, reduce personnel number while guarantee the maintenance tasks. Therefore, the proposed approach is beneficial to improve the company's production efficiency.

Keywords Complex manufacturing system · Preventive maintenance · Production-maintenance synchronization · Extreme learning machine · Ant colony optimization

List of symbols

β	weight matrix of ELM output layer
B	bias matrix of ELM hidden layer
C	the adjusted parameter of ELM
$g(\cdot)$	activation function
I	identity matrix
L	number of ELM hidden layer nodes
L_{lop}	last overlap list
L_{op}	overlap list
m	number of ELM input features
N	number of ELM training samples
n	number of ELM output caregories
N_p	the number of PM tasks
N_{ant}	the number of ants in ACO algorithm
S_d	total distance
T	output matrix of ELM
T_c	sum of T_{prc_i}, $1 \leq i \leq 12$ in the past for each machine
T_{cc}	CM cycle time
T_{cds_i}	hourly CM duration of day shift
T_{cds}	CM duration of day shift
T_{cd}	CM duration of machine
T_{maxc}	maximum sum of T_{prc_i}, $1 \leq i \leq 12$ in the past for each machine
T_{maxpd}	maximum PM duration
T_{minc}	minimum sum of T_{prc_i}, $1 \leq i \leq 12$ in the past for each machine
T_{minpd}	minimum PM duration
T_{op_i}	hourly opportunity time
T_{p_i}	PM duration of one PM type
T_{pc}	PM cycle time
T_{prc_i}	hourly predicted required capacity
Tb_d	distance table

✉ Kuo-Yi Lin
19603@tongji.edu.cn

Li Li
lili@tongji.edu.cn

Yong Wang
1810847@tongji.edu.cn

1 College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

2 Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai 201804, China

Published online: 13 June 2020

Tb_p	path table
X	input feature matrix of ELM training samples
W	weight matrix of ELM hidden layer

Acronyms

ACO	ant colony optimization
CM	corrective maintenance
ELM	extreme learning machine
EMS	equipment maintenance system
MES	manufacturing execution system
MSE	mean square error
PCA	principal component analysis
PdM	predictive maintenance
PM	preventative maintenance
TSP	traveling salesman problem

Introduction

In recent years, under the wave of artificial intelligence, the traditional manufacturing industry has undergone tremendous changes. Countries have also formulated corresponding policies to promote the progress of manufacturing and rapidly develop intelligent manufacturing, such as Made in China 2025 (Wübbeke et al. 2016; Butollo and Lüthje 2017; Li 2018), German Industry 4.0 (Lasi et al. 2014; Lee et al. 2015; Lin 2018) and American Industrial Internet (van Lier 2014). As an important part of intelligent manufacturing, product lifecycle management has attracted the attention of many researchers (Saaksvuori and Immonen 2008; Stark 2015).

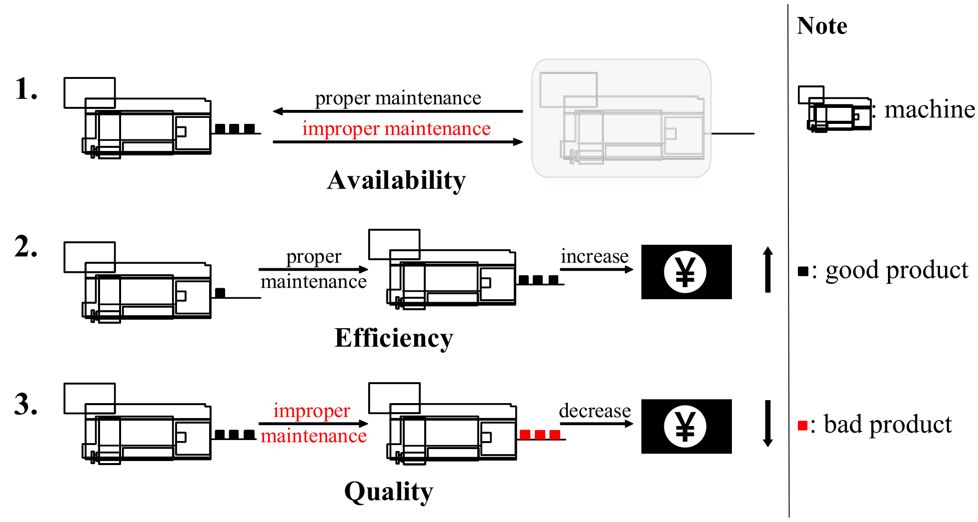
Equipment maintenance refers to all action taken to retain material in a serviceable condition or to restore it to serviceability (Venkataraman 2007). In view of product lifecycle management, maintenance is as important as production to ensure the quality. The former takes an important role on keeping and improving system availability and safety, as well as production quality. The latter is concerned about the use and synergy of equipment capacity and timely delivery of products. Figure 1 illustrates several effects of proper and improper maintenance. The improper arrangement of maintenance may affect production efficiency and cause work in process (WIP) accumulation, especially for the complicated production system with re-entrant flows, specially semiconductor wafer fabrication facility.

Maintenance is usually divided into preventative maintenance (PM), predictive maintenance (PdM) and corrective maintenance (CM) (Jezzini et al. 2013). PM refers to the regular maintenance based on maintenance experience. PdM is to predict the remaining useful life of the equipment and other factors based on equipment status, historical maintenance and other information. And maintenance before the equipment goes down. CM refers to maintenance after equipment failure. Due to the insufficient informationization of the factory and the lack of effective monitoring of the equipment status, the research in this paper focuses on PM.

In term of PM interval, PM can be categorized into periodic PM and non-periodic PM. Periodic PM usually means that the maintenance cycle is fixed. Periodic PM was first proposed by Barlow and Hunter (Barlow and Hunter 1960), they considered two policies: (1) perform PM after certain hours of continuing operation without failure, (2) perform PM on the system after it has been operating a total of certain hours regardless of the number of intervening failures. Based on this basic model, many researchers have proposed new research results. Periodic PM was used to achieve the reliability equipment of the fault-tolerant computer systems (Yak et al. 1985). Cost was considered in (Canfield 1986), proposed cost optimization of the PM intervention interval is obtained by determining the average cost-rate of system operation. Time interval of maintenance was considered in (Boland 1982; Berenguer et al. 1997; Ji et al. 2007), long time intervals lead to poor maintenance results, and short time intervals increase maintenance costs. Nakagawa (1986) considered periodic and sequential PM policies for the system with minimal repair at failure. Non-periodic PM can be regarded as dynamic PM, and PM period in different time periods is different, but the period of the same time period is constant. The reason for using Non-periodic PM is that the equipment status changes due to real environmental factors such as order, maintenance, and repair. Pereira et al. (2010) presented a Particle Swarm Optimization approach for non-periodic preventive maintenance scheduling optimization. Fitouhi and Nourelfath (2012) dealt with the problem of integrating noncyclical preventive maintenance and tactical production planning for a single machine. Lin et al. (2015) indicated that PM activities performed at a high reliability threshold can not only significantly improve the system availability but also efficiently extend the system lifetime. The optimization objectives of these articles focus on reliability and cost evaluation. A good way to achieve these optimization objectives is to integrate the maintenance with the production for developing opportunistic maintenance preserving conjointly the product-production-equipment performances.

Group maintenance and opportunistic maintenance are two popular PM methods (Zhou and Shi 2019). Group maintenance methods are intent to reduce the maintenance cost and improve the system performance by jointly performing several PM activities, they are more suitable for work-stations parallel systems. When the time interval or the number of damaged machines reach to a threshold, all work-stations perform maintenance tasks (Ab-Samat and Kamaruddin 2014; Chalabi et al. 2016). Opportunistic maintenance methods focus on performing the several components maintenance at opportunities, such as equipment downtime due to failure and

Fig. 1 Several effects of proper and improper maintenance



production plan. Xia et al. (2015) considered both machine degradation and characteristics of batch production in opportunistic maintenance. Opportunistic maintenance methods are more suitable for work-stations serial system (Aizpurua et al. 2017; Sheikhalishahi et al. 2017). The dependencies among components are usually classified into three kinds: economic, stochastic and structural dependence (Keizer et al. 2017), in which the economic dependence has been most studied in maintenance optimization process. Van Do et al. (2013) firstly proposed a dynamic group maintenance strategy for multi-component systems with positive economic dependence, they considered the maintenance opportunities to optimally update online the group maintenance planning. They further considered both "positive" and "negative" economic dependencies (Vu et al. 2014), the maintenance cost can be significantly reduced by considering system structure when the shutdown costs cannot be neglected.

For most PM models, they usually assume that the actual production process follows a mixed distribution of several common distributions or use simulation software to model the actual production process. The data from the actual production line are rarely used into the algorithm development process, which causes gap between the model and the actual application.

In this paper, a novel model is proposed for optimizing the PM tasks of a complex re-entrant production system and preserving conjointly the product-production-equipment performances. The first part uses an extreme learning machine algorithm to predict the required capacity distribution among the machines to pursue a smooth job flow, and find the opportunistic periods for implementing the PM tasks. The second part is responsible to calculate the PM interval and duration, optimize the schedule of the PM tasks, and obtain the least number of repairmen. The main contributions of this paper include: (1) uses historical data from actual MES and EMS to predict opportunities for maintenance tasks to have less impact on production based on availability, (2) applies improved ant colony algorithm to optimize staffing and obtains the least number of repairmen based on cost.

The rest of this paper is organized as follows. In "Problem description" section briefly introduces the manufacturing system and the maintenance problem. In "Proposed approach based on availability and cost" section details the proposed approach, including required capacity prediction, ant colony optimization (ACO) for PM scheduling and repairmen reduction. In "Experiments and results" section, the data from real MES and EMS are used to demonstrate the effectiveness of the proposed approach, and the results obtained from these experiments are described. In "Discussion" section gives some discussion of the paper. Finally, conclusion is drawn in "Conclusion" section.

Problem description

The problem considered in this paper is to integrate maintenance into production of a complex manufacturing system to have no or less impact on the production and to optimize the number of repairmen to save cost.

The manufacturing system can be outlined as Fig. 2.

- There are multiple work-centers in a manufacturing system. The jobs finished on one work-center will be transferred to the buffer of another work-center according to their process flow file.
- There are one or more than one similar machine in a work-center. They have the similar process ability, but their capacity may be different. All the machines in a work-center utilize the common buffer.

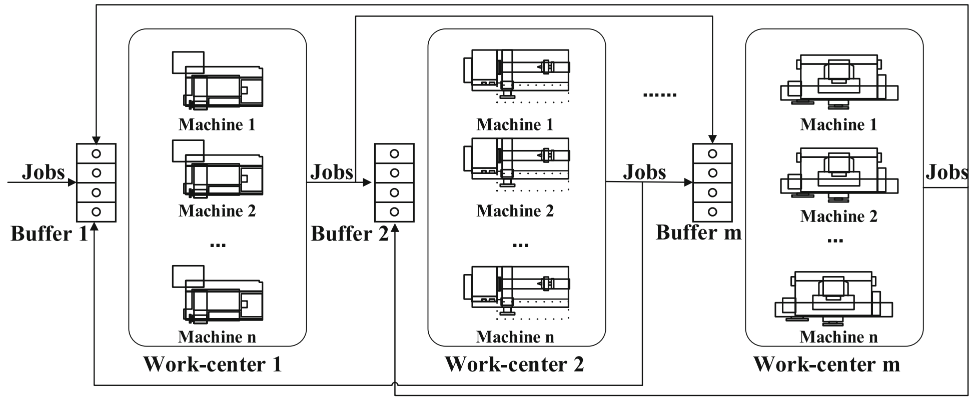


Fig. 2 Scheme diagram of the manufacturing system

– When a machine becomes idle, it will decide which job in the buffer is to be processed or which PM task is to be implemented.
– The whole manufacturing system is always working, except for the yearly maintenance. There is no specific stop for the machines' maintenance tasks. However, this is possible to find some periods for each machine when the work-center have less workload for the opportunities to implement its maintenance tasks without or with less impact on the production.

When calculate the opportunities for maintenance tasks, the assumptions are as follows:

– In the real manufacturing system, the maintenance tasks are implements at day shift (from 8:00 a.m. to 8:00 p.m.). As a result, the plan horizon for maintenance optimization is set to 12 hours (from 8:00 a.m. to 8:00 p.m.). The planning unit is one hour.
– It is assumed that the interval and duration of maintenance tasks are known by predicting with historical data. Information related to PM tasks include the PM type, PM cause, PM name, earliest start time, latest start time, best start time and duration, etc. That of CM tasks include the CM type, CM name, start time and duration, etc.
– Due to the processing flow of the jobs is re-entrant, it is difficult to calculate the required capacity of each machine with theoretical methods. In this paper, the historical data are utilized from MES and EMS to predict the required capacity of each machine (shown in "Prediction model of the required capacity") using regression algorithm. Then calculate the opportunities of each machine (shown in "Opportunities distribution and PM tasks acquisition" section).

The required capacity in this paper refers to the processing time period of the machine. The total time minus the required capacity and CM time is the opportunity of the machine. For the case where many chips are processed on the same machine, the start and end time of these are different. The gap of earliest start time and latest end time are chosen as the required capacity, and the repeated time period is only calculated once.

The optimization of maintenance tasks is to distribute the maintenance tasks to opportunities adequately. In this paper, an improved ant colony optimization algorithm is utilized to optimize maintenance tasks. The detailed information is shown in "ACO algorithm for distributing PM to oppurtunities" section.

Proposed approach based on availability and cost

Description of the proposed approach

The main steps of the proposed approach are outlined as follows, and the corresponding research framework is shown in Fig. 3.

1. The data from the real production line are processed first, such as data cleaning and filtering, feature extraction, dimension reduction, normalization.
2. The extreme learning machine method is applied to the processed data to predict the required capacity of the machine.
3. Extract and analyze historical maintenance data, obtain cycle and duration of maintenance time, use improved ant colony algorithm to assign maintenance tasks to opportunities, and optimize maintenance personnel.

Prediction model of the required capacity

The data from MES that record the production related data, such as facility code, start time, end time, and data from EMS that record the maintenance related data, such as facility code, actual start time, actual end time, activity cause.

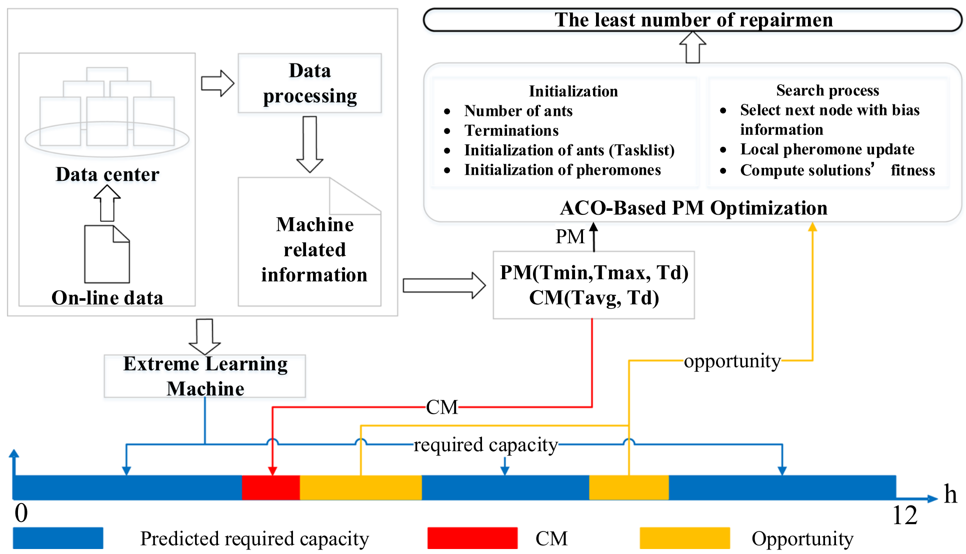


Fig. 3 The research framework of the proposed approach

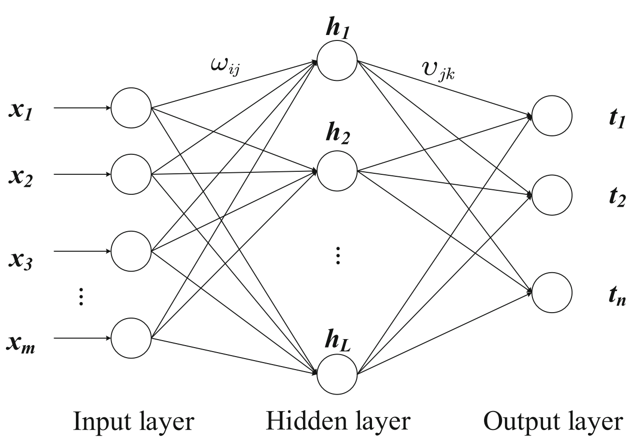


Fig. 4 The Structure of ELM

The prediction model of required capacity of each machine is constructed by the approach of extreme learning machine (ELM), which is an emerging feedforward neural network learning method that was proposed by Huang et al. (2006, 2011). ELM is a three-layer network including input layer, hidden layer and output layer. It randomly generates input weights and biases of the hidden layer. At the same time, the unique optimal solution can be obtained by setting the number of neurons in the hidden layer without any adjustment during the training process. Compared with the traditional feedforward neural network training methods, ELM has the advantages of fast training speed, global optimal solution and good generalization performance. Therefore, ELM is adopted as the training algorithm to construct required capacity prediction model.

The structure of ELM is presented in Fig. 4, where w_{ij} and v_{jk} are weight parameters. Therefore, the required capacity in the current state (i.e., online input data) is predicted.

Let us consider a regression problem of N training samples with m features $X = [x_1, x_2, \ldots, x_i, \ldots, x_N]^T, 1 \le i \le N, x_i = [x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{im}], 1 \le j \le m$ and n categories $T = [t_1, t_2, \ldots, t_i, \ldots, t_N]^T, 1 \le i \le N, t_i = [t_{i1}, t_{i2}, \ldots, t_{ij}, \ldots, t_{in}], 1 \le j \le n$.

Suppose hidden layer has L nodes and all weights and biases on these nodes are generated randomly, $W = [w_1, w_2, \ldots, w_i, \ldots, w_m]^T, 1 \le i \le m, w_i = [w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{iL}], 1 \le j \le L, B = [b_1, b_2, \ldots, b_i, \ldots, b_N]^T, 1 \le i \le N, b_i = [b_{i1}, b_{i2}, \ldots, b_{ij}, \ldots, b_{iL}], 1 \le j \le L$.

Therefore, the output of hidden layer is $H = g(X \cdot W + B)$, where g is an activation function that is sigmoid function $\frac{1}{1+e^{-x}}$ most popularly. Other activation functions have $tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, $Relu = \max(0, x)$, etc. The mathematical model of ELM can be described as:

$$H\beta = \mathrm{T} \tag{1}$$

where β is the weight matrix of the output layer, the least-square solution of Eq. (1) is indicated by Eq. (2):

$$\beta = H^+ T = \begin{cases} H^T (HH^T)^{-1} T, N \le L \\ (H^T H)^{-1} H^T T, N \ge L \end{cases} \tag{2}$$

where H^+ is the Moore–Penrose generalized inverse of the matrix H. When considering optimization of ELM by using parameter regularization, the result can be obtained as follows:

$$\beta = H^+ T = \begin{cases} H^T (HH^T + \frac{I}{C})^{-1} T, N \le L \\ (H^T H + \frac{I}{C})^{-1} H^T T, N \ge L \end{cases} \tag{3}$$

where C is a parameter that needs to be adjusted, I is a identity matrix.

Table 1 Some inputs of the ELM

Variable	Meaning
$x_1, \ldots, x_m$	The average of the difference between the processing start time and the sampling start time in the selected sampling time for m machines
$x_{m+1}, \ldots, x_{2m}$	The standard deviation of the difference between the processing start time and the sampling start time in the selected sampling time for m machines
$x_{2m+1}, \ldots, x_{3m}$	The num of processing in the selected sampling time for m machines
$x_{3m+1}, \ldots, x_{4m}$	The average of the difference between the processing finish time and the sampling start time in the selected sampling time for m machines
$x_{4m+1}, \ldots, x_{5m}$	The standard deviation of the difference between the processing finish time and the sampling start time in the selected sampling time for m machines
$x_{5m+1}, \ldots, x_{6m}$	The average of the difference between the maintenance start time and the sampling start time in the selected sampling time for m machines
$x_{6m+1}, \ldots, x_{7m}$	The standard deviation of the difference between the maintenance start time and the sampling start time in the selected sampling time for m machines
$x_{7m+1}, \ldots, x_{8m}$	The num of maintenance in the selected sampling time for m machines
$x_{8m+1}, \ldots, x_{9m}$	The average of the difference between the maintenance finish time and the sampling start time in the selected sampling time for m machines
$x_{9m+1}, \ldots, x_{10m}$	The standard deviation of the difference between the maintenance finish time and the sampling start time in the selected sampling time for m machines

Some inputs of the ELM are described in Table 1, there may be different depending on the actual problem. The corresponding output is the required capacity of each machine.

The ELM weights are determined by using grid search. The procedure is shown in Algorithm 1.

Algorithm 1 The weight determination of ELM

Require: input and output of train set T_i, T_o, input and output of validation set V_i, V_o, number of hidden nodes L, parameter of ELM C, max of parameters Max_c, Max_l, minimum mean squared error (MSE) mse_{min}, weight $\hat{W}_h$, $\hat{W}_h$, MSE of predicting mse_{pred}
Ensure: hidden layer weight $\mathbf{W}_h$, output layer weight $\mathbf{W}_o$

1: **initialize**: Set $L \leftarrow 10, C \leftarrow 1, mse_{min} \leftarrow 1e8, Max_c \leftarrow 1e6, Max_l \leftarrow 6000$
2: **while** $C < Max_c$ **do**
3:     **initialize**: Set $L \leftarrow 10$
4:     **while** $L < Max_l$ **do**
5:         Calculate weights: $\hat{W}_h, \hat{W}_h \leftarrow ELM\_train(T_i, T_o, C, L)$
6:         Calculate MSE of predicting: $mse_{pred} \leftarrow ELM\_predict(V_i, V_o, \hat{W}_h, \hat{W}_h)$
7:         **if** $mse_{pred} < mse_{min}$ **then**
8:             Update weights: $mse_{min} \leftarrow mse_{pred}, \mathbf{W}_h \leftarrow \hat{W}_h, \mathbf{W}_o \leftarrow \hat{W}_o$
9:         $L \leftarrow L + 10$
10:     $C \leftarrow C \cdot 2$

Opportunities distribution and PM tasks acquisition

Opportunities distribution

The main steps of finding opportunities are shown as follows:

1. Calculate CM duration of day shift T_{cds} for each machine using historical CM data. For CM, average CM cycle time T_{cc} and CM duration T_{cd} of each machine can be obtained. In order to simplify the calculation, the actual CM distribution is not simulated and directly assign them to the day shift ($12h$), the unit of these variables is $h(hour)$.

$$T_{cds} = \frac{12}{T_{cc}} \cdot T_{cd} \tag{4}$$

2. Predict the hourly required capacity of each machine T_{prc_i}, $1 \leq i \leq 12$. For example, the data of 8:00 p.m. yday-8:00 a.m. are utilized to predict the required capacity of each machine from 8:00 a.m. to 8:00 p.m., 9:00 a.m.–9:00 p.m, …, 7:00 p.m.–7:00 a.m. tomorrow. Details on calculating hourly required capacity of each machine can be found in "Prediction result".

3. Calculate the hourly opportunity T_{op_i}, $1 \leq i \leq 12$. First, in order to finish the CM task early, distribute the T_{cds} in the initial free time to obtain the opportunity time of each hour.

$$T_{op_i} = 1 - T_{cds_i} - T_{prc_i}, 1 \leq i \leq 12 \tag{5}$$

Next, determine which machine take which period as their opportunities for maintenance according to optimization objectives. If two or more consecutive hour have opportunities, the opportunities are preferred to be integrated into a bigger one.

PM tasks acquisition

The main steps of obtaining PM tasks are shown as follows:

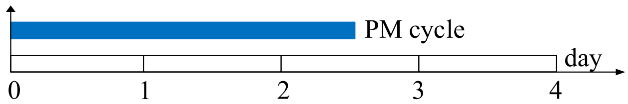


Fig. 5 PM day determination

1. Calculate the average PM cycle time T_{pc}, maximum and minimum PM duration T_{maxpd}, T_{minpd}. A machine may have different maintenance tasks that have different types and reasons, so subdivide the maintenance according to the maintenance types and maintenance reasons. The T_{pc} and PM duration distribution of maintenance after subdivision are obtained. According to PM duration distribution, set the confidence interval (default 0.95) to get the T_{maxpd} and T_{minpd} for avoiding false data interference.

2. PM day determination. According to T_{pc} of subsystem, if day satisfies Eq. (6), then will determine which PM need to be completed on which day. The unit of these variables is $h(hour)$.

$$\text{floor}((dy+1)\cdot 24/T_{pc}) - \text{floor}(dy \cdot 24/T_{pc}) \geq 1 \tag{6}$$

where floor is floor function, dy is the index of day, starting from 0.

Just as shown in Fig. 5, floor$((2+1)\times 24/60)-$floor$(2\times 24/60) \geq 1$, therefore there will be PM on the second day.

3. Determine PM duration T_{pd}. According to PM day determination, for a PM task that will be performed on a certain day after subdivision, corresponding PM duration is given by Eq. (7).

$$\begin{aligned} T_{pd} &= T_{minpd} + \tfrac{T_{maxc}-T_c}{T_{maxc}-T_{minc}} \cdot T_{pdd} \\ T_{pdd} &= T_{maxpd} - T_{minpd} \end{aligned} \tag{7}$$

where T_{maxc} is the maximum sum of T_{prc_i}, $1 \leq i \leq 12$ in the past for each machine. T_{minc} is the minimum sum of T_{prc_i}, $1 \leq i \leq 12$ in the past for each machine. T_c is the sum of T_{prc_i}, $1 \leq i \leq 12$ for each machine.

Therefore, if the sum of predicted required capacity is large, PM duration will be small. That is, less time is spent on maintenance, however it's not less than the T_{minpd}.

Now, the PM task and PM duration that needs to be done on a certain day has been determined, and the opportunity time has also been obtained. T_{p_i} is PM duration of i_{th} PM task after subdivision, $1 \leq i \leq N_p$, N_p is the number of PM tasks that need to be done according to Eq. (6). Next step is to distribute the PM to opportunity reasonably.

ACO algorithm for distributing PM to opportunities

Ant colony optimization (ACO) algorithm is an iterative optimization method (Colorni et al. 1992), which is mainly used to solve the traveling salesman problem (TSP) (Reinelt 1991; Valdez et al. 2020). Here, the ACO algorithm is improved for maintenance scheduling.

Optimization objectives

Two optimization objectives are considered:

1. The first is availability, to minimize the sum of quotients of T_{p_i} and T_{op_i} as Eq. (8). Note that T_p and T_{op} on the same machine should correspond. During optimization, T_p cannot be assigned to T_{op} of other machines.

$$\min \sum_{i=1}^{N_p} \frac{T_{p_i}}{T_{op_i}} \tag{8}$$

where i is i_{th} PM task.

The reason for choosing this objective is to minimize the impact of maintenance on production and to ensure that maintenance is completed as much as possible during opportunities.

$$\begin{aligned} &\tfrac{T_{p_1}}{T_{op_1}} + \tfrac{T_{p_2}}{T_{op_2}} + \Sigma < \tfrac{T_{p_2}}{T_{op_1}} + \tfrac{T_{p_1}}{T_{op_2}} + \Sigma \\ &T_{p_1}T_{op_2} + T_{p_2}T_{op_1} < T_{p_2}T_{op_2} + T_{p_1}T_{op_1} \\ &T_{p_1}(T_{op_2} - T_{op_1}) < T_{p_2}(T_{op_2} - T_{op_1}) \end{aligned} \tag{9}$$

where Σ is the sum of other quotients, suppose Σ is constant.

Just as Eq. (9), when $T_{op_2} > T_{op_1}$, $T_{p_2} > T_{p_1}$, $\frac{T_{p_1}}{T_{op_1}} + \frac{T_{p_2}}{T_{op_2}} < \frac{T_{p_2}}{T_{op_1}} + \frac{T_{p_1}}{T_{op_2}}$. That is to say, this objective enables small PM to be placed in small opportunity. While large PM is placed in small opportunity, and the probability of exceeding the opportunity is greater than that of large PM is placed in large opportunity. This will ensure that PM is completed as much as possible during opportunity. And leave the largest percentage of opportunity time for the next PM scheduling, because the actual prediction is always error.

Of course, there are also cases of $\frac{T_{p_1}}{T_{op_1}} > 1$ and $\frac{T_{p_2}}{T_{op_2}} > 1$. At this time, the impact of large PM on small opportunity time will have a greater impact on production than. Because a large T_{p_i} has a greater probability of affecting i+2 and later plans, while a small T_{p_i} may only affect i+1 plan. Therefore, the above optimization objective is eventually adopted.

2. The second is cost, to minimize the number of maintenance personnel.

To simplify the scheduling optimization model. This paper assumes that a maintenance personnel accepts all maintenance tasks for one or several devices, that is, maintenance personnel can perform different maintenance tasks on a device within the same time period. This paper does not allow maintenance personnel to perform maintenance on different devices within the same time period. This is because it takes time for maintenance personnel to switch between different devices. On the one hand, this part of the time is difficult to

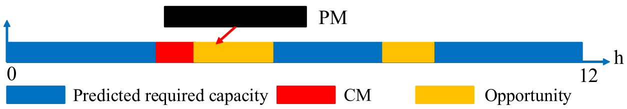


Fig. 6 Bottleneck machine (case1)

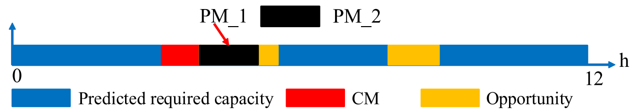


Fig. 7 Non-bottleneck machine (case2)

consider. On the other hand, this can improve maintenance efficiency.

Implementation steps

Some steps for realizing the improved ACO algorithm are shown as follows:

1. Determine the search space of the ACO algorithm. From above, the PM tasks and opportunity time of each machine are obtained, the unit is h(hour). The number of PM tasks N_p is the number of nodes that are need to be maintained at day shift in the search space, it should be noted that one machine may have multiple PM tasks.
2. Determine the number of the ants. The number of ants N_{ant} is set to be the range of 50–90% of N_p, this number is obtained through the analysis of the results of the iteration.
3. The termination conditions. Two kinds of termination conditions are set. One is the maximum number of iterations, this number is obtained through the analysis of the results of the iteration. The other is the minimum change of the minimum objective values in two consecutive iterations.
4. Initialization of each artificial ant (**first optimization objective**). First, a distance table Tb_d, path table Tb_p is built for each artificial ant (indexed with x), denoted by Tb_{d_x} and Tb_{p_x}, there are two cases when initialize the Tb_{d_x}. One is the bottleneck machine as Fig. 6.

For PM task i, opportunity time i, j, ant x:

$$Tb_{d_{x(i,j)}} = \frac{T_{p_i}}{T_{op_{i,j}} + T_{op_{i,j+1}} + \cdots + T_{op_{i,z}}},\ i <= N_p, z <= 12 \tag{10}$$

where j is the period of time, ranging from 1 to 12, z is the smallest index that T_{p_i} can be placed. If the sum of opportunity time $(i, j), (i, j+1), \ldots (i, 12)$ is less than the value of PM task i, the value of $Tb_{d_{x(i,j)}}$ is set to 1e6, in other words, the opportunity time i, j isn't suitable for PM task i. The purpose of this calculating is to merge several opportunities time together.

The other case is non-bottleneck machine as Fig. 7.

For PM task i, opportunity time i, j:

$$Tb_{d_{x(i,j)}} = \frac{T_{p_i}}{T_{op_{i,j}}} \tag{11}$$

Then, the start point is randomly distributed to each artificial ant. The node distributed to ant x is added to Tb_{p_x}, and the $Tb_{d_{x(i,j)}}$ is added to total distance S_{d_x}. If a machine contains multiple PM tasks a day, update the distance of next PM task $i+1$ for opportunity time i, j that have placed PM task i. The rule of updating is the same as above.

5. Optimize the number of repairmen (**second optimization objective**). For different machines to be maintained at the same period, punish them to optimize the number of maintenance personnel. Suppose L_{lop_x} is the times list of last hourly overlap for ant x, in other word, is the number list of different machine being maintained at the same period per hour and L_{op_x} is now. Therefore, the S_{d_x} is updated as:

$$\begin{aligned} &S_{d_x} = S_{d_x} + Tb_{d_{x(i,j)}} + L_{opd_{x(j)}} \\ &L_{opd_{x(j)}} = a \cdot (L_{op_{x(j)}} - L_{lop_{x(j)}}),\ L_{op_{x(j)}} = L_{lop_{x(j)}} \end{aligned} \tag{12}$$

where a is a parameter, set to 100 in this paper.
When an ant x finishes the scheduling task, the max of L_{op_x} is added to S_{d_x},

$$S_{d_x} = S_{d_x} + b \cdot \max(L_{op_x}) \tag{13}$$

where b is a parameter, set to 200 in this paper.
6. The pheromones initialization. The initial pheromones on the arcs are set as a small positive number, such as 1. Then choose next node according to the so-called pseudorandom proportional rule, given by

$$p_{i,z}(t) = \frac{\tau_{i,z}^{\alpha}(t)\eta_{i,z}^{\beta}}{\sum_{j=0}^{12} \tau_{i,j}^{\alpha}(t)\eta_{i,j}^{\beta}},\ \eta_{i,z} = \frac{1}{Tb_{d_{x(i,z)}}} \tag{14}$$

where $\tau_{i,z}^{\alpha}(t)$ is pheromone for PM task i and opportunity time i, z . α, β are parameters, usually set α to 1, β to 2.
7. Pheromone updating. When all ants complete the scheduling tasks, update the pheromone as:

$$\tau_{i,j}(t+1) = \rho\tau_{i,j}(t) + \sum_{x=1}^{N_{ant}} \Delta\tau_{i,j}^{x},\ \Delta\tau_{i,j}^{x} = Q/S_{d_x} \tag{15}$$

where Q is a parameter, set to 0.001 in this paper. The deposited pheromone is discounted by a factor ρ, which results in the new pheromone trail being a weighted average between the old pheromone value and the newly deposited pheromone.

8. When iterations of all ants are completed, then need to save the ants x_b with the shortest walking distance. And the max of $L_{op_{x_b}}$ is the number of repairmen that company want i.e. maximum number of maintenance personnel in the same time period.

$$repairmen = \max(L_{op_{x_b}}) \tag{16}$$

Experiments and results

The experimental data of the actual production line come from WeEn semiconductor multi-workstation system. WeEn Semiconductor is a large enterprise, has 7 work-centers (diffusion, dry etcher, evap, implanter, photo, test, wet etcher), 153 available machines, and 29 repairmen. Historical data contains production and maintenance data. The period of empirical study is 2017-01-01 to 2018-06-30. The amount of data in the work-centers and historical maintenance is shown in Fig. 8.
where "job_history" is history data of maintenance, "diffusion, dry etch, evap, implanter, photo, test, wet etch" are history data of production for different work-centers. The sum of maintenance data and production data can reach millions.

Maintenance data contain job information and time information. The main fields of job information contain job number, facility code, facility type, activity type, activity cause and job priority. The main fields of time information contain raised date, actual start date, actual end date. The main fields of Production data contain facility code, track in time, track out time, track in operator.

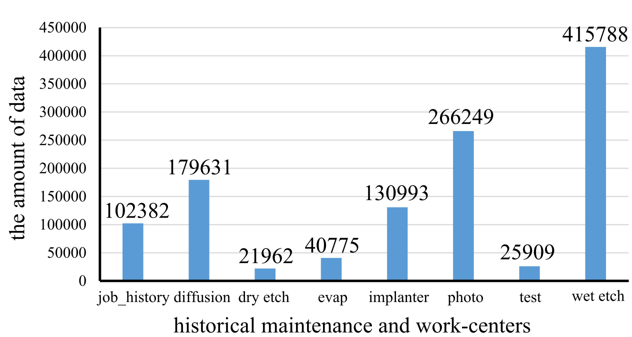


Fig. 8 The amount of data in the work-centers and historical maintenance

The next step is to process the data. Taking 20% of the data as the test set and 80% as the train set. The evaluation of the experimental results is divided into two parts: 1) the evaluation of the prediction result of required capacity, the metrics are Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), 2) the scheduling result of the maintenance staff, compared with the actual number of maintenance staff. The MSE and RMSE in this paper are defined in Eq. (17):

$$\begin{aligned} MSE &= \frac{1}{M \cdot N} \sum_{m=1}^{M} \sum_{n=1}^{N} (T_{m,n} - \hat{T}_{m,n})^2 \\ RMSE &= \sqrt{MSE} \end{aligned} \tag{17}$$

where T is actual capacity matrix, each row of the matrix represents the capacity for all machines a day in the different time periods, each for the matrix represents the capacity for a machine all days in the same time period, $\hat{T}$ is predicted capacity matrix, M, N are the number of rows and columns of the matrix.

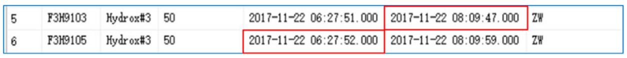
All the experiments were performed on a PC with Intel Core Intel i5-7300HQ CPU@2.50 GHz, 8 GB RAM and python3.6.

Data processing

The steps of processing the data include data cleaning, sampling time selection, feature extraction (selection of feature extraction sources, dimension reduction), normalization and regression models. Based on the basic settings, change the settings one step at a time to find the relatively optimal settings for processing data. The basic settings are: data cleaning method (discard error data), sampling time (8:00 p.m. yday-8:00 a.m.), feature extraction source (work-center feature: the features shown in Table 1 are machine-oriented, and the work-center feature is based on the work center. The extracted features are similar to those in Table 1), dimension reduction (principal component analysis (PCA)), normalization (global normalization), regression method (ELM with sigmoid activation function). The experiment uses 3 times 5-fold cross validation, and the ELM parameters are determined by using grid search, the range of ELM parameter C is: [2e-10, 2e-9,…, 2e9, 2e10], and the range of ELM parameter L is: [100, 200,…, 3900, 4000].

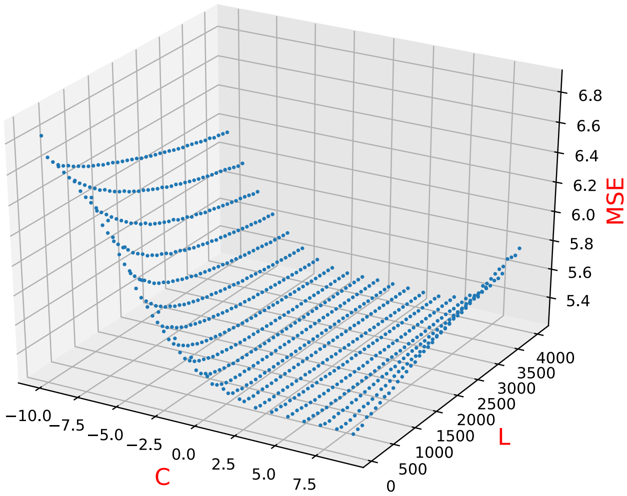
Data cleaning

In addition to the general data missing and data duplication, there is a special kind of data error: erroneous time stamp, as shown in Fig. 9:

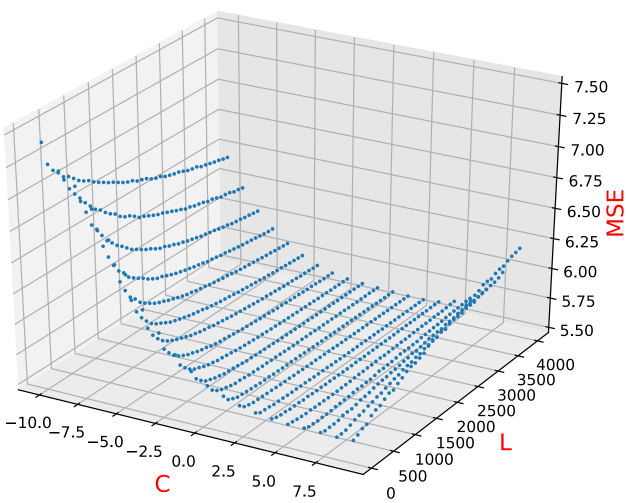


5	F3H9103	Hydrox#3	50	2017-11-22 06:27:51.000	2017-11-22 08:09:47.000	ZW
6	F3H9105	Hydrox#3	50	2017-11-22 06:27:52.000	2017-11-22 08:09:59.000	ZW

Fig. 9 Erroneous time stamp



(a) discard data



(b) merge data

Fig. 10 Parameter results of two data cleaning methods

For this type of error, there are two processing methods: the first discards part of the data (the second row of data in the Fig. 9), and the second merges the data, taking the minimum start time and the maximum finish time.

Figure 10 shows how the mean square error (MSE) of the prediction varies with the ELM parameters under different cleaning methods, the abscissa is log(C). The minimum MSE of discarding data is 5.310087, the minimum MSE of merging data is 5.588556, there is no significant difference between the two methods. The cleaning method with the best prediction effect is discarding data.

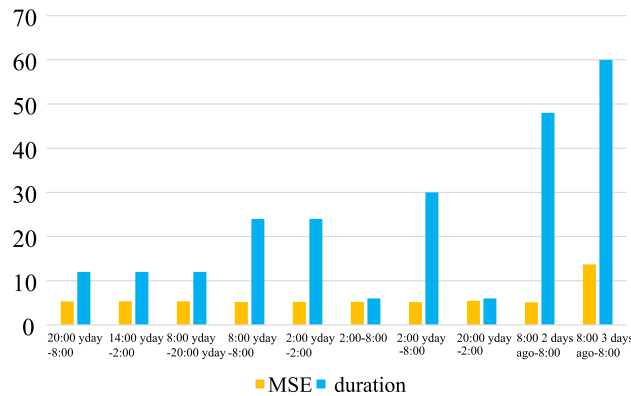


Fig. 11 Prediction results (MSE) of different sampling times

Sampling time

Sampling time refers to the sampling period, which determines the original input data of ELM algorithm. The starting point and duration of sampling time determine the sampling time. In this paper, different starting points and durations are tried. Figure 11 shows the prediction effect of different sampling times. Yellow part is the predicted MSE, and blue part is the duration for different sampling times.

From Fig. 11, several points can be found:

– With the same sampling duration, the prediction effect is worse when the starting point of sampling time is earlier.
– Before a specific sampling duration, the increase of duration can increase the prediction effect.

The final selected sampling time is 8:00 a.m. 2 days ago-8:00 a.m.

Feature extraction

Feature extraction source determines the original input features of the ELM algorithm. dimension reduction algorithm reduces the size of the original input features by extracting features that are relatively important to the algorithm, thereby reducing the runtime of the algorithm. In this paper, PCA dimension reduction algorithm are used. Figure 12 shows the prediction effect of different feature sources.

From Fig. 12, several points can be found:

– The more features are extracted after subdivision, the better the effect is.
– The increase of feature sources does not necessarily improve the prediction effect.

The final selected feature source is machine feature.

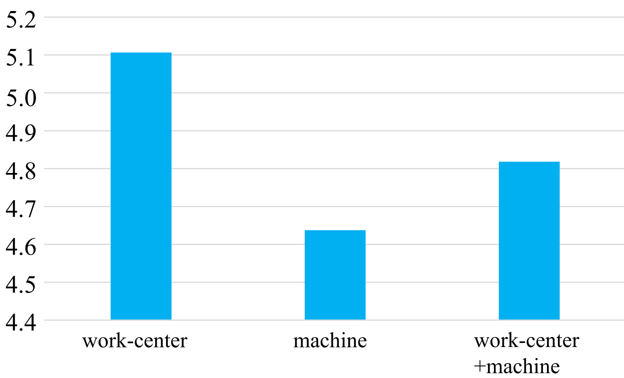


Fig. 12 Prediction results (MSE) of different feature sources

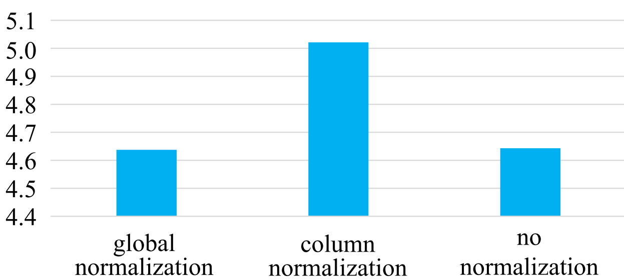


Fig. 13 Prediction results (MSE) of different normalization methods

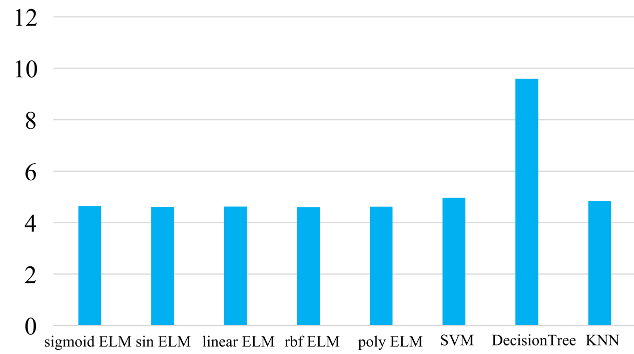


Fig. 14 Prediction results (MSE) of different regression methods

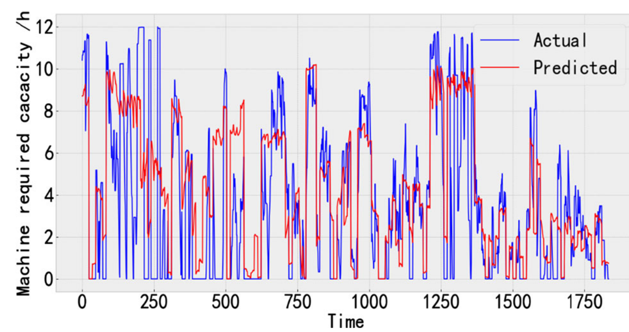


Fig. 15 The result of predicting required capacity for every machine at 12h

Normalization

The normalization operation transforms the data after dimension reduction to a decimal ranging from 0 to 1. Normalization can speed up the algorithm's convergence and avoid excessive values. Figure 13 shows the prediction effect of different normalization methods. Column normalization is normalization method that only normalize column vectors.

From Fig. 13, several points can be found:

– There is no significant difference between the prediction results of no normalization and global normalization.
– Column normalization reduces prediction effect.

The final selected normalization method is global normalization.

Regression methods

In order to predict the continuous required capacity, this paper has tried some regression methods. Figure 14 shows the prediction effect of different regression methods.

The abscissa lists some common regression methods, sigmoid ELM is ELM with sigmoid activation function, sin ELM is ELM with sin activation function, linear ELM is ELM with linear kernel function, rbf ELM is ELM with Radial Basis Function(rbf) kernel function, poly ELM is ELM with polynomial ELM, SVM is support vector machine, DecisionTree is decision tree, KNN is k-NearestNeighbor algorithm.

From Fig. 14 and experiment procedure, several points can be found:

– The running time of SVM, DecisionTree and KNN is very long, and the prediction effect is not outstanding, therefore they are excluded.
– The prediction effect of different ELM algorithms is similar, and the best one is rbf ELM.

The final selected regression method is rbf ELM.

Prediction result

Figure 15 shows the result of predicting required capacity for every machine at 12h.
where x-axis is 12h before and after each machine, such as 0 is machine1 [8:00 a.m.–8:00 p.m.], 1 is machine1 [9:00 a.m.-9:00 p.m.],…, 11 is machine1 [7:00 p.m.–7:00 a.m.], 12 is machine2 [8:00 a.m.–8:00 p.m.],…, machine2 [7:00 p.m.–7:00 a.m.]…Each machine has 12 data, and there are the data for 153 devices. The y-axis is predictive required capacity of each machine at 12 h. The best predictive RMSE is 2.35, predictive error is within 20% of actual.

According to machine required capacity at 12 h, the required capacity for every hour can be obtained. Suppose the required capacity of each hour [8:00–9:00, 9:00–10:00,…, 19:00–20:00,…, 6:00–7:00] as $x_1, x_2, \ldots, x_{11}, x_{12}, \ldots, x_{22}, x_{23}$, the result of predicting as $y_1, y_2, \ldots, y_{11}, y_{13}$.

$$\begin{cases} x_1 + x_2 + \cdots + x_{11} + x_{12} = y_1 & 1) \\ x_2 + x_3 + \cdots + x_{12} + x_{13} = y_2 & 2) \\ x_3 + x_4 + \cdots + x_{13} + x_{14} = y_3 & 3) \\ \vdots \\ x_{12} + x_{13} + \cdots + x_{22} + x_{23} = y_{12} & 12) \end{cases} \tag{18}$$

There are only 12 equations, but have 23 unknown variables, the equation has an infinite number of solutions, therefore set $x_{12}, x_{13}, \ldots, x_{22}, x_{23} = y_{12}/12$,

$$\begin{cases} x_{12} = y_{12}/12 \\ x_{11} = y_{11} - 11 \cdot x_{12} \\ x_{10} = y_{10} - x_{11} - 10 \cdot x_{12} \\ \vdots \\ x_1 = y_1 - x_2 - \cdots - x_{12} \end{cases} \tag{19}$$

The hourly data are calculated separately in order to make the data distribution relatively uniform and no sudden changes occur in the processing.

Ant colony optimization

Parameter determination

The number of ants and iterations are two parameters determined by trying multiple sets of parameters. The number of ants increases from 50% to 90% of *pm_num* with an interval of 5%, determined by analyzing the effect of optimization. The number of iterations increases from 50 with an interval of 10. When it increases to a certain number, the effect of iterations doesn't change, then the certain number is the number of iterations.

Optimization result

(1) Actual scheduling

Actual scheduling is schedule that required capacity, opportunity and PM total from actual data. The improved ACO algorithm be used to redistribute PM to opportunity.

Our scheduling time is 480 days from January 1. The number of ants is 50 and the number of iterations is 50. Figure 16 shows the Gantt chart of the actual scheduling results of devices Stripper#1 (the actual number of devices is 153, and all devices cannot be displayed). Black is production, blue is maintenance, and pink is opportunity. It can be seen that maintenance can be reasonably allocated to opportunities.

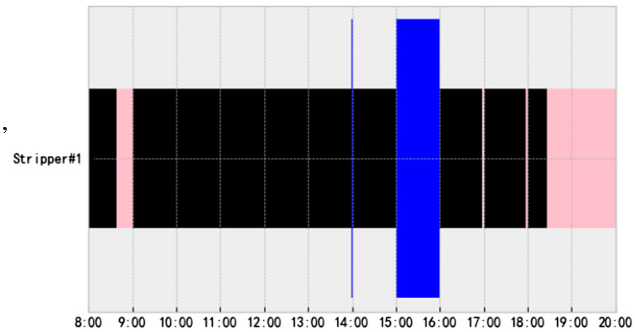


Fig. 16 Actual scheduling Gantt chart

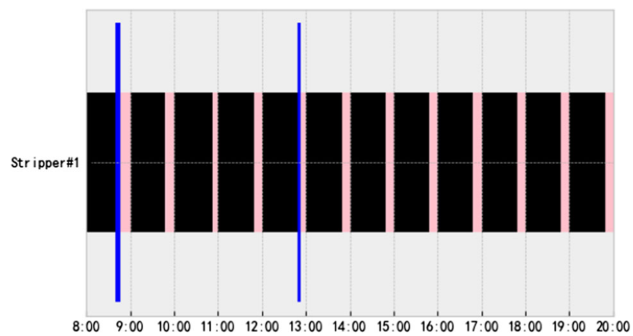


Fig. 17 Predictive scheduling Gantt chart

The result of the scheduling is the number of repairmen reduced from 29 to 23, and haven't "1e6" situations i.e. all PM can be placed reasonably. Compared with the previous routine scheduling, our algorithm optimizing the staffing and reducing the number of maintenance personnel.

(2) Predictive scheduling

Predictive scheduling is schedule that required capacity, opportunity and PM total obtained from historical data as mentioned earlier.

The parameter and scheduling time are consistent with the actual scheduling. It should be noted that the predictive scheduling is scheduled for the next day. The predictive scheduling only has day shift to improve the efficiency. Because of inconsistency with people's habits and lack of supervision, the maintenance efficiency of night shift is relatively low.

Figure 17 shows the Gantt chart of the predictive scheduling results of devices Stripper#1. Black is production, blue is maintenance, and pink is opportunity. It can be seen that maintenance can be reasonably allocated to opportunities.

The result of the proposed scheduling method is the number of repairmen reduced from 29 to 24, and have little "1e6" situations, it means that some opportunity can't place PM task. In other words, some production works need to be delayed.

Discussion

For most PM models, the data come from the simulation environment, and the actual production line data are rarely used into the algorithm development process, which causes gap between the model and the actual application. The reasons for this situation are: (1) the level of informatization of the factory is insufficient to obtain complete data, (2) the privacy of the factory, unable to provide data, (3) the actual data is not pure and missing, cannot be used, (4) the actual production line is too complicated to verify the model, etc.

This paper ignores various complex assumptions and develops a predictive model based on actual production line data. It schedules maintenance personnel, and optimizes the configuration of maintenance personnel reasonably. Compared with other models, this model can be more directly applied to the actual production environment. Experimental results verify the effectiveness of our method. However, the above problems also affect this paper. This paper only uses a real data set to verify the model, and the model's prediction accuracy still cannot reach an error of less than 10% when all parameters are optimal at present, which can be improved in future research.

Another issue in this paper is that the training time is relatively long, due to the huge amount of data and dimensions. It is difficult for other machine learning algorithms to improve training speed. In recent years, the rapid development of quantum computers and quantum algorithms has made it possible to develop quantum machine learning. Quantum computers can solve specific problems such as search and optimization more effectively than classical computers. In future research, we can try quantum algorithms to speed up the training of the model.

Conclusion

In this study, a novel model is proposed to optimize the maintenance tasks and preserve conjointly the production-equipment performances for a re-entrant production system. Superior to other algorithms which were mainly validated in simulation environment, this study could optimize staffing and reduce costs in the actual manufacturing system.

However, because the data comes from the real production line, which is greatly interfered by human beings and has great randomness, the prediction effect is not very ideal, this part can be improved in the future research. The second part to be improved is that although PM simulates the distribution of the real production line, it only simply selects the mean value for calculation, and later historical data can be selected for prediction. Finally, we will try quantum machine learning to speed up the training of the model.

Acknowledgements This research was supported by National Key R&D Program of China, No. 2018YFE0105000, the National Natural Science Foundation of China under Grant No. 51475334, the Shanghai Municipal Commission of science and technology No. 19511132100 and the Fundamental Research Funds for the Central Universities of China under Grant No. 22120170077.

References

Ab-Samat, H., & Kamaruddin, S. (2014). Opportunistic maintenance (OM) as a new advancement in maintenance approaches: A review. *Journal of Quality in Maintenance Engineering*, *20*(2), 98–121.

Aizpurua, J. I., Catterson, V. M., Papadopoulos, Y., Chiacchio, F., & D'Urso, D. (2017). Supporting group maintenance through prognostics-enhanced dynamic dependability prediction. *Reliability Engineering & System Safety*, *168*, 171–188.

Barlow, R., & Hunter, L. (1960). Optimum preventive maintenance policies. *Operations research*, *8*(1), 90–100.

Berenguer, C., Chu, C., & Grall, A. (1997). Inspection and maintenance planning: An application of semi-Markov decision processes. *Journal of intelligent manufacturing*, *8*(5), 467–476.

Boland, P. J. (1982). Periodic replacement when minimal repair costs vary with time. *Naval Research Logistics Quarterly*, *29*(4), 541–546.

Butollo, F. & Lüthje, B. (2017). 'Made in China 2025': Intelligent manufacturing and work. *The new digital workplace. How new technologies revolutionise work*, pp. 42–61.

Canfield, R. V. (1986). Cost optimization of periodic preventive maintenance. *IEEE Transactions on Reliability*, *35*(1), 78–81.

Chalabi, N., Dahane, M., Beldjilali, B., & Neki, A. (2016). Optimisation of preventive maintenance grouping strategy for multicomponent series systems: Particle swarm based approach. *Computers and Industrial Engineering*, *102*, 440–451.

Colorni, A., Dorigo, M., Maniezzo, V., et al. (1992). Distributed optimization by ant colonies. In: *Proceedings of the first European conference on artificial life* (Vol. 142, pp. 134–142). Cambridge, MA.

Van Do, P., Barros, A., Bérenguer, C., Bouvard, K., & Brissaud, F. (2013). Dynamic grouping maintenance with time limited opportunities. *Reliability Engineering and System Safety*, *120*, 51–59.

Fitouhi, M.-C., & Nourelfath, M. (2012). Integrating noncyclical preventive maintenance scheduling and production planning for a single machine. *International Journal of Production Economics*, *136*(2), 344–351.

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2011). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics Part B (Cybernetics)*, *42*(2), 513–529.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*(1–3), 489–501.

Jezzini, A., Ayache, M., Elkhansa, L., Makki, B., & Zein, M. (2013). Effects of predictive maintenance (PdM), Proactive maintenace (PoM) and Preventive maintenance (PM) on minimizing the faults in medical instruments. In: *2013 2nd International conference on advances in biomedical engineering* (pp. 53–56). IEEE.

Ji, M., He, Y., & Cheng, T. E. (2007). Single-machine scheduling with periodic maintenance to minimize makespan. *Computers and operations research*, *34*(6), 1764–1770.

Keizer, M. C. O., Flapper, S. D. P., & Teunter, R. H. (2017). Conditionbased maintenance policies for systems with multiple dependent components: A review. *European Journal of Operational Research*, *261*(2), 405–420.

Lasi, H., Fettke, P., Kemper, H.-G., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Business and Information Systems Engineering*, *6*(4), 239–242.

Lee, J., Bagheri, B., & Kao, H.-A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, *3*, 18–23.

Li, L. (2018). China's manufacturing locus in 2025: With a comparison of "Made-in-China 2025" and "Industry 4.0". *Technological Forecasting and Social Change*, *135*, 66–74.

Lin, K.-Y. (2018). User experience-based product design for smart production to empower industry 4.0 in the glass recycling circular economy. *Computers and Industrial Engineering*, *125*, 729–738.

Lin, Z.-L., Huang, Y.-S., & Fang, C.-C. (2015). Non-periodic preventive maintenance with reliability thresholds for complex repairable systems. *Reliability Engineering and System Safety*, *136*, 145–156.

Nakagawa, T. (1986). Periodic and sequential preventive maintenance policies. *Journal of Applied Probability*, *23*(2), 536–542.

Pereira, C. M., Lapa, C. M., Mol, A. C., & Da Luz, A. F. (2010). A particle swarm optimization (pso) approach for non-periodic preventive maintenance scheduling programming. *Progress in Nuclear Energy*, *52*(8), 710–714.

Reinelt, G. (1991). TSPLIB'A traveling salesman problem library. *ORSA Journal on Computing*, *3*(4), 376–384.

Saaksvuori, A., & Immonen, A. (2008). *Product lifecycle management*. Berlin: Springer.

Sheikhalishahi, M., Azadeh, A., & Pintelon, L. (2017). Dynamic maintenance planning approach by considering grouping strategy and human factors. *Process Safety and Environmental Protection*, *107*, 289–298.

Stark, J. (2015). Product lifecycle management. In: *Product lifecycle management* (Vol. 1, pp. 1–29). Springer, Cham.

Valdez, F., Moreno, F., & Melin, P. (2020). A Comparison of ACO, GA and SA for Solving the TSP Problem. In: *Hybrid intelligent systems in control, pattern recognition and medicine* (pp. 181–189). Springer, Cham.

van Lier, B. (2014). Developing the industrial internet of things with a network centric approach: A holistic scientific perspective on smart industries. In: *2014 18th International conference on system theory, control and computing (ICSTCC)* (pp. 324–329). IEEE.

Venkataraman, V. (2007). *Maintenance engineering and management*. New Delhi: PHI Learning Pvt. Ltd.

Vu, H. C., Do, P., Barros, A., & Berenguer, C. (2014). Maintenance grouping strategy for multi-component systems with dynamic contexts. *Reliability Engineering and System Safety*, *132*, 233–249.

Wübbeke, J., Meissner, M., Zenglein, M. J., Ives, J., & Conrad, B. (2016). Made in China 2025: The making of a high-tech superpower and consequences for industrial countries. *Mercator Institute for China Studies*, *17*, 2017–09.

Xia, T., Jin, X., Xi, L., & Ni, J. (2015). Production-driven opportunistic maintenance for batch production based on MAM-APB scheduling. *European Journal of Operational Research*, *240*(3), 781–790.

Yak, Y., Forward, K., & Dillon, T. (1985). Modelling the effect of transient faults in fault tolerant computer systems. *IFAC Proceedings Volumes*, *18*(12), 129–133.

Zhou, X., & Shi, K. (2019). Capacity failure rate based opportunistic maintenance modeling for series-parallel multi-station manufacturing systems. *Reliability Engineering and System Safety*, *181*, 46–53.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.