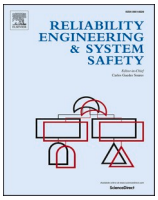
Reliability Engineering and System Safety 222 (2022) 108429

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress



Joint decision-making of parallel machine scheduling restricted in job-machine release time and preventive maintenance with remaining useful life constraints

Xinxin He [a], Zhijian Wang [a,b,*], Yanfeng Li [a], Svetlana Khazhina [c], Wenhua Du [a], Junyuan Wang [a], Wenzhao Wang [d]

[a] Department of Mechanical Engineering, North University of China, Taiyuan, Shanxi 030000, China
[b] Key Laboratory of Education Ministry for Modern Design and Rotor-Bearing System, Xi'an Jiaotong University, Xi'an, Shaanxi 710000, China
[c] Department of Medical Physics and Informatics, Bashkir State Medical University, Ufa Lenina 3, 450008, Russia
[d] School of Instrument and Electronics, North University of China, Taiyuan, Shanxi 030000, China

ARTICLE INFO

Keywords:
Gated recurrent unit
Job-machine release times
Makespan
Parallel machine scheduling
Preventive maintenance
Remaining useful life prediction
Teaching and learning based optimization

ABSTRACT

The machine remaining useful life (RUL), the job-machine release time and the correlation between the maintenance duration and the machine enlistment age are, in this paper, collectively emphasized at the parallel machine scheduling problem. Based on this, a corresponding mixed integer programming model is constructed to minimize the makespan and the processing loss beyond the machine RUL threshold, where a discrete teaching and learning based optimization algorithm is applied to solve this NP-hard problem, and a fault mode-assisted gated recurrent unit (FGRU) life prediction method is used to guide the predictive maintenance initiation time of all machines. In addition, this paper demonstrates that the FGRU method is more accurate than three common methods (Encoder-Decoder Recurrent Neural Network, Bidirectional Long Short-Term Memory and GRU) through two actual bearing degradation cases, and shows through three benchmark cases that the joint decision-making can effectively reduce the time cost of manufacturing enterprises.

1. Introduction

In the increasingly competed market, the job-shop scheduling problem (JSP) is derived from the pursuit of continuous profit in manufacturing enterprises. The constraints in the JSP generally include the job arrival time, the job delivery time, the job deterioration effect, the machine processing cost, the machine functional attributes, shop geographical distribution, etc. The scheduling objectives of JSP roughly include the makespan, the total delay time, the total production cost, etc. And its frameworks have been roughly divided into single machine, parallel machine, flow shop, hybrid flow shop and other targeted modes. Traditional parallel machine scheduling can be described as: m constant-speed machines with similar functional properties need to process n different jobs with only one process. As an extension of single machine scheduling and a very important research field in scheduling family, the parallel machine scheduling is extensively used in the actual semi-conductor manufacturing, wafer fabrication and call-in service center, and has attracted extensive attention of scholars since McNaughton [1] made the first study on it.

With the hysterical pursuit of better processing continuity, major industrial entities and research institutions have begun to implement attempts on various production models and scheduling theories. Mönch and Shen [2] studied a parallel machine scheduling problem with weighted delivery time as the objective function in distributed manufacturing environment. Basiri et al [3]. constructed a flexible parallel machine scheduling model for the reality of fuzzy processing time, sequence-dependent setup time and reentrant jobflow, where a Pareto-based multi-objective meta-heuristics was proposed. Li et al [4]. investigated a parallel machine scheduling problem with position-dependent deteriorating jobs and DeJong's learning effects in an uncertain system, which is fairly consistent with the intermediate products in chemical industry and the employees in human-computer interaction processing, respectively. Xu et al [5]. studied a parallel machine scheduling rule for the outside processing and the due window, motivated by the flexible delivery time and the sharing of trial-manufacture jobs among groups. Xiao et al [6]. analyzed the

* Corresponding author at: Department of Mechanical Engineering, North University of China, Taiyuan, Shanxi 030000, China.
E-mail address: wangzhijian1013@163.com (Z. Wang).

https://doi.org/10.1016/j.ress.2022.108429
Received 11 November 2021; Received in revised form 24 February 2022; Accepted 25 February 2022
Available online 27 February 2022

parallel machine scheduling in green manufacturing from the aspects of time cost and carbon emissions, and that conformed to the national policy of carbon neutralization. Beside the parallel machine framework, Zhang et al [7]. presented a two-stage hybrid flow shop scheduling problem (HFSP) with reentrant and limited waiting time constraints based on block painting operations. Gao et al [8]. addressed a flexible job shop scheduling problem (FJSP) with the constraints of fuzzy processing time and new job insertion based on artificial bee colony. Abdelmaguid et al [9]. studied the proportionate multiprocessor open shop scheduling problem (PMOSP) in which the processing time on a given center is not job-dependent. To sum up, they all have established the mixed integer programming models suitable for their concerned attributes, which has always been the materialization direction of traditional parallel machine problem.

The said literature have made a basic assumption that all jobs to be processed are released at zero time, which has undeniable application prospects in deterministic orders, such as schoolroom administration and port unloading. However, in many cases, e.g. call center supply, health code verification and wine-making, the job arrival time uncertainty has brought many troubles to the subsequent resource allocation and personnel arrangement, easily leading to the congestion of transportation channels and the idle of equipment for a long time. That greatly limits the applicability of the flawed hypothesis. Especially in the handling of intermediates (MI) with deterioration effect, the said assumption may bring the direct risk of MI waste and the secondary risk of equipment scrapping. The job arrive time has become a recognized major technical issue. To protect direct beneficiaries against such liability, a handful of scholars have made finite restrictions on the job arrival time. Long et al [10]. investigated the steelmaking-continuous casting production scheduling problem, and solved the dilemma of job release time fluctuation to some extent. Cevikcan and Durmusoglu [11] added a workload regulation module in the parallel machine system to determine when and where the jobs are released to the workshop under what conditions. Zhou et al [12]. addressed an energy-efficient scheduling problem with unequal job arrival times including productivity and energy cost measures, and provided decision makers with a good approximation of Pareto solutions. To summarize, manufacturing service systems with indefinite orders rely on the ability to deal with the random job arrival times.

In the same way, the importance of machine release time in scheduling underlying settings is also noted well. For instance, in component highly correlated and personnel intensive systems, e.g. engine assembly, machines may be frequently and unexpectedly occupied when new orders arrive. Nevertheless, in the literature published in the past few decades, the machine release time consideration is mainly limited to single machine scheduling [13], and they even crudely presume that all machines are available at time zero in other scheduling frameworks [14]. Instead, to provide a feasible machine allocation table for each job within the maximum allowable waiting time, the backstage supporters not only need to provide a limited buffer for deterministic orders, but also make the ultimate solution comply with local regulations and other cost constraints, which makes the scheduling model more complex and difficult to solve. The readable mathematical model and optimization algorithm with lower time complexity are worth discussing. Mario[13] and Defersha[15] are two pioneers in the machine release time filed, those interested can refer to them more.

The current machine availability is reflected not only in whether it is occupied by other jobs, but also in the machine performance. Today, defect and fatigue evolution will occur at any time as machines become more sophisticated, which causes emergent incidents or slow-changing breakdowns, and then makes it unavailable during shutdown maintenance. Adiri et al. [16]. first noticed the unavailability reality caused by machine failure and stated a single machine flow-time scheduling problem with a single breakdown. Wang et al [17]. addressed a proactive scheduling problem with stochastic machine breakdown arising from steel production. Abbas et al [18]. thought the emergent failures would disrupt pre-established planning. In total, they have made a forward-looking theoretical discussion on the predictive maintenance for different types of machine failures. During practical production, for the continuous benefits and low restart costs, e.g. die change and relocation [19], the mainstream maintenance has undergone the fundamental change from periodical or corrective maintenance to well-timed predictive maintenance around machine status, which not only can avoid redundant interference-induced unnecessary stoppages, but also prevent overdue renovation-induced cascading failures. Be careful here, although some literature have considered the machine unavailability constraint, they tend to focus on periodic or single maintenance and evade the changing reliability, like the Refs. [16–18]. Such an unrealistic intention is not valid in many real-world applications. Therefore, at present, how to balance production scheduling and preventive maintenance, how to make the machine continuously have good operation state and high reliability, and then how to realize the harmonious unity of scheduling, maintenance and reliability still are key issues to be solved in the realization of intelligent manufacturing.

As a foundation for studying complicated systems, the joint decision-making in single machine has been widely deliberated. Kacem [20] proposed a single machine scheduling case with a fixed maintenance interval, where all machines had only one definite unavailability period. Yu et al [21]. believed multi-cycle maintenance interval in single machine system with non-preemptive jobs were rigid and known in advance. Zou and Yuan [22] attached the constraints, a maximum allowable maintenance interval and a fixed maintenance duration, to the single machine scheduling with job rejection. For parallel machine systems with predictive maintenance, Hsieh and You [23] thought that current maintenance cycle was flexible, and it only depended on the job number in a continuous processing batch since the last maintenance. Marsili et al [24]. assumed the maintenance based on short and frequent interruptions had less impact on the system operability compared to those based on longer and sporadic interventions. Pang et al [25]. said that the flexible maintenance cycle was applicable to a part of machines, and it only related to the machine service age. In conclusion, the current quantitative criteria of unavailability constraints of machines basically do not match the current reliability indicators of machines. There is still no unified standard for the communication bridge between maintenance trigger and machine reliability. Fortunately, over the past a few decades, failure distribution model of manufacturing systems, a criterion for quantifying the machine ability to perform specific functions, has been intensively investigated. By considering the deterioration effect of jobs and machine, Cui et al [26]. studied the Weibull function for single machine system, whose strategy is to perform predetermined maintenance to recover the system whenever the machine performance from Weibull was lower than the reliability threshold. It is also called threshold based maintenance strategy. Tao et al [27]. introduced recursive decline factor and failure rate rise factor into Weibull distribution in incomplete maintenance scenario. Sousa et al [28]. used a radial basis function to reduce unnecessary maintenance insertion and resource delivery for corroded pipelines. Inspired by the fault perception classification, Ke et al [29]. evaluated each machine reliability based on the memoryless property of adjacent maintenances. It is worth noting that, for simplicity, they followed the condition based maintenance strategy, “maintenance at reliability threshold”. In fact, on the one hand, early maintenance can enhance product yield rate, reduce the overhaul cost and ensure the consistency of job physical properties, on the other hand, when machine reliability exceeds the given threshold, partial maintenance resources, e.g. spare parts inventory and outsourcing supply, may not meet the necessary conditions for immediate maintenance, which means that reliability threshold should be regarded as the latest maintenance cordon rather than the maintenance trigger. Furthermore, from their point of view, the maintenance duration associated with each breakdown was treated as expected or fixed values. On the contrary, in many engineering practices, the maintenance durations of machines in different degradation stages are actually discrepant. In

view of the gaps among current maintenance strategy, duration and realistic maintenance concept, this paper sets a linear relationship between maintenance duration and machine reliability, where the near-optimal maintenance initiation time wound be determined by the optimization algorithm described later within the reliability threshold, and no longer entirely controlled by the time when the reliability threshold is exceeded for the first time.

Subsequently, most literature including [21–29] utilize a ready-made statistical model for off-line monitoring of machines as the maintenance guidance, without online considering dynamic service environment-induced stochastic fault modes, e.g. unlubrication friction-created headstock overheating and transilient load-associated tool resonance. By the same token, although Failure Mode/Effect and Criticality Analysis (FMEA/FMECA) can predetermine the relative severity and probability of failures, what it is essentially dealing with is the failure possibility of a machine before the service or in a static service environment, not the failure probability of the monitored system on the go or in the dynamic service environment, nor can it give the approximate duration range of the fragile parts from early failure to complete failure. Rather than acquiring the time-variant failure probability of monitored components in the dynamic environment, we prefer to predict the degradation duration of monitored components to total failure, which can help business owners plan some time-bound preparations in advance.

In the interests of monitoring life-circle machine reliability on line, inspired by the ideas of literature [30,31], our motivation is to develop a data-driven remaining useful life (RUL) prediction model considering the influence of stochastic fault modes on RUL prediction accuracy throughout the lead time. To the best of our knowledge, this is an early attempt to combine RUL prediction with the joint decision-making of maintenance and scheduling. Nowadays, the intelligent transformation is making the monitoring data gradually change from the production by-product to the production key [32]. Through the analysis and mining of massive degradation data, the experience- or model-driven maintenance management of manufacturing unit is ushering in a significant revolution [33,34]. And indeed, that is what generalized predictive maintenance did, which integrates condition monitoring, fault diagnosis, RUL prediction, maintenance decision-making and maintenance activities into the dispatching automation system. Nonetheless, scholars in the RUL field prefer to pursue the improvement of RUL prediction accuracy, rather than inquire into the specific maintenance logic around resource supply [35,36], which is the stand aloof from intricate industrial society to a certain extent. In its way, our modest contribution is kind of the first application exploration of RUL prediction in parallel machine system.

Eventually, by pulling together three latent terms in the parallel machine system: job-machine release times, predicted RUL (online reliability) and maintenance duration associated with enlistment age, in this article, a joint decision-making for minimizing makespan and machining penalty beyond RUL threshold is formulated with the aim of obtaining the near-optimal job production/sequence sequences, job-machine combination and maintenance initiation time. Nevertheless, evaluating the MRP-II coherence under the uncertain maintenance duration and reliability is not straightforward and facing three challenges as following: (1) due to the uncertainty of job-machine release time, the idle time of waiting for jobs or machines is a stochastic quantity, which is challenging to propagate the uncertainty to the makespan; (2) under the stochastic maintenance duration and initiation time, the existing evaluation methods for maintenance insertion are no longer effective; (3) resolving the parallel machine scheduling and preventive maintenance model under the RUL guidance is computationally tedious, and it involves more constraints. Consequently, it necessitates the development of a short-cut discrete optimization algorithm to reduce the calculation burden.

In the combinatorial optimization, especially for the parallel machine system, the branch delimitation algorithm that can uniquely find the optimal solution has heavy computational burden, so here it is not considered. The heuristic or meta-heuristic algorithms are more effective because they do not require the objective function to satisfy specific conditions or have favorable mathematical properties [37]. These algorithms (particle swarm optimization (PSO) [38], whale optimization algorithm (WOA) [39], fruit fly optimization algorithm (FOA) [40], etc.) have attracted more and more attention and are widely used in various engineering problems. As one of the most representative heuristic algorithms, teaching and learning based optimization (TLBO) algorithm has been developed by Rao et al. [41] and does not require any specific parameters. Its entire optimization process includes the teaching stage and the learning stage. In the teaching stage, each learner learns from the teacher (the best learner). In the learning stage, each learner learns from the other learner in a random manner. Due to the relatively competitive global search and convergence performance, TLBO has long been regarded as a bright new star among many heuristic algorithms [42]. Despite all this universalism, all original meta heuristic algorithms are based on the resolution of continuous functions. In order to realize the successful execution of the optimizer on the discrete combinatorial optimization, this paper needs to encode the discrete variables that job-machine combination and job processing/release sequence into the continuous computable variables, and then establish a discrete TLBO algorithm.

To recognize the unique novelties of this work, comparing with the existing joint scheduling planning and predictive maintenance models, the main contributions of our work can be summarized as the following three aspects: (1) A new joint decision-making model for scheduling planning and predictive maintenance is put forth for the parallel machine system, where the proposed discrete teaching-learning based optimization algorithm is treated as the solver; (2) The job-machine release time, the job release sequence and the maintenance duration flexibility are considered into influence factors interactively, then a novel data-driven remaining useful life prediction method reconciled by stochastic fault modes is formulated to guide all maintenance activities. And it is the first time that data-driven remaining useful life prediction is considered in the dynamic preventive maintenance policy of parallel machines; (3) We made comparative experiments using two degeneration datasets and three job-machine benchmark data. And the results have proved the efficiency and superiority of the proposed RUL prediction method and joint decision-making model of parallel machines.

The remainder of this paper is organized as follows: Section 2 descripts the basis scheduling assumptions and RUL problem definition Section 3. formulates the new joint decision-making of parallel machine scheduling and predictive maintenance, and explains the discretization process of TLBO algorithm Section 4. introduces the structures of fault mode-assisted RUL prediction process in detail Section 5. starts the comparative test and the experimental analysis, and Section 6 concludes this study and discusses some future works.

2. Preliminaries

In this section, to clearly state the joint decision-making (SPM) and RUL prediction problems, we roughly standardize the framework of parallel machine system, the premise assumptions of joint decision-making and the RUL conception.

2.1. SPM framework and assumptions

SPM can be described as follows: n different jobs with fixed processing volume and only one process need to be processed on m identical-speed machines with inconsistent initial RULs, where (1) all machines are multifunctional, that is, any one of m machines can process any job, (2) any job is considered to be ready for delivery after it is completely processing by any machine, (3) all jobs can be delivered in a batch only after they are all processed completely, (4) in the start of a scheduling period, whether the jobs arrive and whether machines are

idle are equivocal, (5) only processing behaviors can and must cause gradual machine performance degradation, (6) during the processing, the vibration data monitored from the vulnerable parts of the machine is input into the proposed RUL prediction model, and the predicted RUL temporarily represents the on-line reliability value of the whole machine at the corresponding time, (7) a limited number of maintenance activities are launched before the above online reliability value is less than the RUL threshold for the first time, (8) each machine maintenance duration is positively related to the enlistment age of the corresponding machine at the initiation time of corresponding maintenance activity. Our purpose is to determine an appropriate job-machine combination and a job processing/release sequence through the discrete TLBO, so as to minimize the makespan and the processing penalty beyond the RUL threshold. Remarkably, the introduction of the penalty is only to decrease the MIP model complexity. At the later iteration stage, this penalty will be reduced to zero, that is, all jobs will be processed within the RUL threshold in the final solution. Additionally, eight basic assumptions of the joint decision-making are itemized as follows: (1) the processing time of each job and the RUL threshold of each machine are fixed and known in advance, (2) the jobs are released sequentially where each releasion cannot be interrupted and preempted, (3) the switching time between two consecutive items is ignored, (4) the maintenance and the processing both cannot be interrupted, and the former are only carried out in two consecutive jobs, (5) the fault is irrelevant among all machines and each predictive maintenance makes the machine "as good as new," (6) one machining position of a machine can only process one job, and one job can only be processed by one machining position of a machine, (7) the number of machine maintenance times is less than the number of its machining positions, and the maintenance resources are adequate, (8) in order to save experimental time and labor cost, the paper uses the accelerated degradation data of the machines, and thus, the units of corresponding job processing time and RUL of machines are replaced by minutes.

2.2. RUL conception

RUL prediction is an engineering discipline working on the prediction of the future state or response of a given system based on the synthesis observations, calibrated mathematical models and simulation. It generally refers to the study of predicting the specific time at which the component will no longer be able to have its expected functional performance.

For a specific component, let $\mathbf{H}(t) = \{H_0, H_1, \ldots, H_t\}$ be its cumulative degradation paths up to the t^{st} time period with $H_0 = 0$, and let $\mathbf{T}_{0\to i} = \{t_0, t_1, \ldots, t_i\}$ and $\mathbf{X}_{0\to i} = \{x_0, x_1, \ldots, x_i\}$ denote the history of the monitoring time and corresponding raw degradation signals, respectively, where the degradation path in the t^{st} period is $H_t = G(\mathbf{X}_{Ws*t\to Ws*t+Sl})$, $\mathbf{X}_{Ws*t\to Ws*t+Sl} = \{x_{Ws*t}, x_{Ws*t+1}, \ldots, x_{Ws*t+Sl}\} \in \mathbf{X}_{0\to i}$. $G(\cdot)$ is a health indicator extraction function, Ws and Sl are the step size and the length of the signal sliding window for $\mathbf{X}_{0\to i}$, respectively. In this setting, the first failure time of monitored component is usually defined as formula (1), and the current RUL is usually expressed as the formula (2):

$$T = \inf\{Ws * t : H_t \geq D | \mathbf{X}_{0\to i}\} \tag{1}$$

$$RUL = \inf\{Ws * r : H_{t+r} \geq D | \mathbf{X}_{0\to i}\} \tag{2}$$

where D represents the given threshold level of the environment in which the monitored object is located. We can see that both of T and RUL are pertinent to the historical degradation paths and current degradation path (Fig. 1a). To be clear, RUL prediction is not necessary at the beginning of operation due to the fact that the performance degradation is rather slow at this stage, while the embedding of RUL prediction at the rapid degradation stage is mandatory because of the abrupt increasing of degradation uncertainty. That is uneconomical to predict RUL too early or too late. However, the boundary between the two stages is vague, limiting the faster and more economical applications of RUL prediction. Thus, the first prediction time (FPT) is expected to appropriately determined (Fig. 1b).

3. Problem formulation

3.1. SPM formulation

In this study, there are m machines and n jobs in total, j is seen as the machine index, l_j jobs will be assigned to machine j, that is, l_j is the number of machining positions where k is the machining position index. i' and i are called the job number, and the orderly release position of each job is recorded as g' or g. The release preparation duration and the release time of job i are denoted as q_i and Q_i, respectively, the processing duration of job i is recorded as p_i. The processing duration and completion time in the machining position k of machine j are expressed as P_{jk} and C_{jk}, respectively. The parallelizable release time of machine j is denoted as b_j. In order to cope with the possible concurrence of machine maintenance and waiting for jobs, the candidate time W_{jk} waiting for job releasion at the machining position k is introduced. And whether machines are maintained is expressed in binary numbers, if the maintenance is performed before the machining position k of machine j, it is recorded as $Y_{jk} = 1$, otherwise recorded as $Y_{jk} = 0$. Similarly, if job i is released at the release position g, it is recorded as $V_{ig} = 1$, otherwise $V_{ig} = 0$. If machine j processes job i at its machining position k, it is recorded as $X_{ijk} = 1$, otherwise $X_{ijk} = 0$. In addition, note that the maintenance duration of machine j before machining position k is T_{jk} and the enlistment age of machine j after machining position k is μ_{jk}. For the RUL notation, we note that RUL_j^{PM} is the RUL threshold of machine j, $RUL_j^{\mathbf{t}}$ is the predicted RUL series of machine j in the time period $\mathbf{t}$, RUL_j^t is the predicted RUL value of machine j at the time point t, and RUL_j^{in} is the initial rated life of machine j. In the last signature, it is denoted that the time when the predicted RUL is less than the RUL threshold for the first time is t_{first} on the real timeline. An example is shown in Fig. 2.

The total objective function F of the joint decision-making problem can be formulated as:

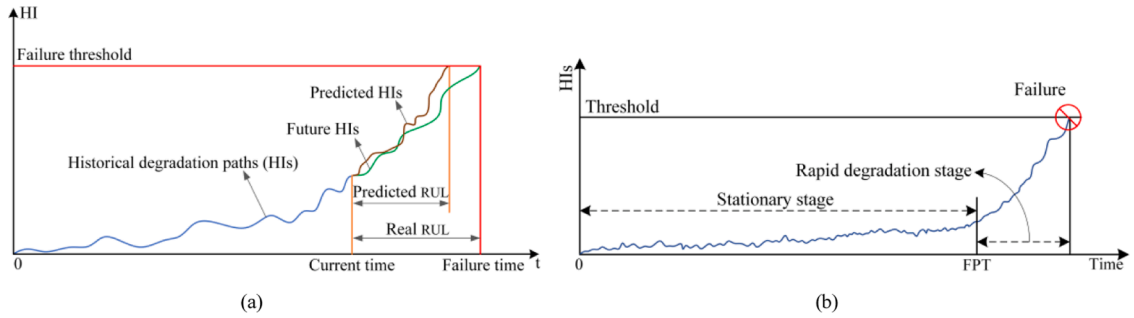


Fig. 1. The symbolization of RUL and FPT.

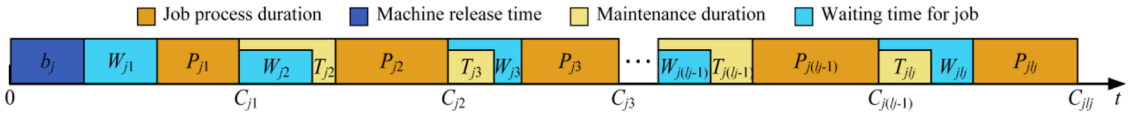


Fig. 2. The sequence of processing jobs and performing preventive maintenance on machine j.

$$\min F = \max_j C_{jl_j} + u\sum_{C_{jk}\in \mathbf{t}} P_{jk}, s.t. RUL_j^{PM} > RUL_j^{\mathbf{t}} \tag{3}$$

where the former is the makespan of all jobs and the latter is a weighted machining penalty beyond RUL threshold, u is the weight coefficient for the machining penalty beyond RUL threshold, which takes a larger positive value so that this item decreases to 0 during the later stage of iteration. It is worth noting that we only take the sum of the processing duration of all jobs whose completion time is beyond the machine RUL threshold as the machining penalty. The following is the formulaic process for calculating the makespan of all jobs.

First, the range of each index value is specified in formula (4) where n different jobs are allocated to m parallel machines, and l_j jobs are assigned to machine j. Then, through formulae 5–(7), the values of maintenance instruction Y_{jk}, processing instruction X_{ijk} and release instruction V_{jg} are limited to a certain range. And the lower limit of release time of each job is obtained by cumulative summation in chains structure in formula (8). Formula (9) gets the processing duration of each machining position by enumerating and searching. Next, considering that machine degradation only occurs in the machining stage, in order to distinguish the actual production process with the maintenance stage and the waiting stage for job arrival, the concept of machine enlistment age is introduced through formula (10), which is also the forerunner of calculating predictive maintenance time. After that, the linear relationship among each maintenance duration, the enlistment age at the corresponding time and the initial rated life of machine is established through the formula (11). And the formula (12) deduces the candidate time for the machines to wait for the arrival of the jobs. The reason why it is called "candidate time" is that waiting for the arrival of jobs and maintenance activities can be parallel, that is, when the maintenance duration is greater than the candidate waiting time, the real waiting time is subject to the current maintenance duration.

To sum up, the completion time of each machining position is determined by four aspects: (1) the duration for processing the corresponding job at the machining position and the processing durations of all previous jobs in the same machine, (2) the durations of all previous preventive maintenance activities before the machining position, (3) the machine release time, (4) the waiting time for the job in the machining position and the waiting time for all jobs before the machining position. We express the completion time of each machining position through the formula (13) and the temporal logic. Ultimately, the function transition of the RUL threshold from the trigger to the alert is determined by the formula (14).

$$\begin{cases} i, i', g, g' \in \{1, 2, \ldots, n\} \\ k \in \{1, 2, \ldots, l_j\} \\ j \in \{1, 2, \ldots, m\} \\ \sum_{j=1}^{m} l_j = n \end{cases} \tag{4}$$

$$\begin{cases} \sum_{k=2}^{l_j} Y_{jk} \le l_j - 1, \forall k \ge 2, j \\ Y_{j1} = 0 \\ Y_{jk} \in \{0, 1\} \end{cases} \tag{5}$$

where, in the normal production course, in order to improve the reliability of the machines, it is necessary to insert several maintenance activities between adjacent machining positions in time, that is, without interrupting the production process, preventive maintenance in the MIP model can only be executed at the end of each job. And $Y_{j1} = 0$ means all machines have no faults after being released and do not need maintenance, and because maintenance is only performed between adjacent machining positions, the maximum number of maintenance is $l_j - 1$.

$$\begin{cases} \sum_{j=1}^{m}\sum_{k=1}^{l_j} X_{ijk} = 1, \forall i \\ \sum_{i=1}^{n} X_{ijk} = 1, \forall j, k \\ X_{ijk} \in \{0, 1\} \end{cases} \tag{6}$$

where the first part formulates the assumption that a job can only be processed at one machining position of one machine, while the second part standardizes the assumption that a machining position of a machine can only process one job.

$$\begin{cases} V_{ig} \in \{0, 1\}, \sum_{g=1}^{n} V_{ig} = 1, \forall i \\ \sum_{i=1}^{n} V_{ig} = 1, \forall g \end{cases} \tag{7}$$

where the first part formulates the assumption that a job can only be released at one releasion position, while the second part standardizes the assumption that a releasion position can only release one job.

$$Q_i \ge \sum_{g'=1}^{g}\sum_{i'=1}^{n} V_{i'g'} q_{i'}, \forall g, i = \operatorname*{argmax}_{i'} V_{i'g} \tag{8}$$

where all jobs are released in a certain order, that is, after the job in the releasion position g is fully released, the job in the releasion position $g+1$ can be released, so, the release time of each job is highly related to the releasion durations of all jobs released before it. Considering the influence of the switching duration between adjacent job releasions, the lower bound form is used to describe the job releasion time, while the equality constraint is still used in the subsequent iterative refinement.

$$P_{jk} = \sum_{i=1}^{n} X_{ijk} p_i, \forall j, k \tag{9}$$

where the processing duration of the machine j at the machining position k is the processing duration of the job assigned to this position. By accumulating the product of the job processing indication X_{ijk} at the position k of machine j and the job processing duration p_i, the corresponding processing duration P_{jk} at this position can be retrieved.

$$\mu_{jk} = \begin{cases} \mu_{j(k-1)}(1 - Y_{jk}) + P_{jk}, k \ge 2 \\ P_{j1}, k = 1 \end{cases} \tag{10}$$

where the enlistment age is jointly determined by all previous processing durations of the machine and whether it has been maintained at each machining position. Before the initial machining position, the machine is considered healthy and will not be maintained, so the enlistment age after the initial machining position is P_{j1}. In each subsequent machining position, if the machine is maintained before a position, due to the assumption of "as good as new", the enlistment age is updated to 0, and then the machining duration of the position is regarded as the enlistment age after the position. If the machine is not maintained before a position, the enlistment age after the position is the sum of the machining

duration of the position and the enlistment age after the last position.

$$T_{jk} = \begin{cases} Y_{jk} \cdot \text{round}\left(\lambda \frac{\mu_{j(k-1)}}{RUL_j^{in}}\right), k \geq 2 \\ 0, k = 1 \end{cases} \tag{11}$$

where, in this work, we assume that the machine maintenance duration is directly proportional to the current enlistment age and inversely proportional to the initial rated life of the corresponding machine. Especially, for the first machining position, the machine is just put into use and does not need maintenance, so the maintenance durations of all machines before the first position are always 0. In addition, the maintenance instruction Y_{jk} introduced in the first part is only to simplify the representation. Separately, if the machine is maintained, the actual maintenance duration is calculated in detail, otherwise, the maintenance duration is always 0. Significantly, the round function is introduced to ensure that all times involved in the MIP model are in integer form, and λ is the comprehensive proportion coefficient.

$$W_{jk} = \begin{cases} \max\left(\sum_{i=1}^{n} X_{ijk} \cdot Q_i - b_j, 0\right), k = 1 \\ \max\left(\sum_{i=1}^{n} X_{ijk} \cdot Q_i - C_{j(k-1)}, 0\right), k \geq 2 \end{cases} \tag{12}$$

where all jobs have the release time constraint, so the possible waiting time for jobs is bound to occupy the production programme and increase the completion time of some jobs.

Specifically, there are two cases: (1) the candidate waiting time for job before the first machining position is jointly determined by the machine release time and the release time of the job to be machined at the first machining position. If the job release time is greater than the machine release time, the candidate waiting time for job is the difference between the said job release time and the corresponding machine release time, otherwise it is 0. It should be noted that there is no maintenance activity before this position, so this said candidate waiting time is also the real waiting time; (2) the candidate waiting times for jobs before the second and subsequent machining positions are jointly determined by the release time of the job to be machined at the corresponding machining position and the completion time of the last machining position. If the said job release time is greater than the said completion time, the candidate waiting time for job is the difference between the said job release time and the said completion time, otherwise, it is 0.

$$C_{jk} \geq \begin{cases} b_j + W_{jk} + P_{jk}, k = 1 \\ C_{j(k-1)} + \max\left(W_{jk}, T_{jk}\right) + P_{jk}, k \geq 2 \end{cases} \tag{13}$$

where the item defines the agreement that the processing start time of a job must be after the processing completion time of the job at the last machining position in the same machine.

In particular, (1) the completion time of the first machining position is greater than the sum of the machine release time, the candidate waiting time for job before the first machining position and the machining duration at the first machining position, (2) The completion time of the subsequent machining positions shall be greater than the sum of the completion time of corresponding last machining position, the processing duration and the real waiting time for job in corresponding machining position. The real waiting time for job takes the maximum value between the candidate waiting time and the maintenance duration in the same period.

$$Y_{j(k+1)} \cdot \left(C_{jk} + T_{j(k+1)}\right) \langle t_{first}, \forall RUL_j^{PM} > RUL_j^{t_{first}}, j, k \tag{14}$$

where, there are two cases: (1) when there is no maintenance in front of a machining position, that is, $Y_{j(k+1)} = 0$, the inequality is always true; (2) when there is maintenance in front of a machining position, that is, $Y_{j(k+1)} = 1$, the end time of the corresponding maintenance activity, which is equal to the sum of the completion time of the last position and the maintenance duration itself, should be before the machine RUL threshold.

In light of the convention that orders are usually sent to customers as a whole, in the proposed MIP model, the relevant cost is evaluated from the perspective of the completion time of the last job. Note that the previously completed jobs will also increase the inventory cost in vain. So the extended inventory cost objective function will be discussed in the follow-up work. Next, we will minimize the total objective formula (3) by iterating the discrete TLBO described later until the maximum number of iterations is met.

3.2. Discrete TLBO formulation

Obviously, the above joint decision-making is a NP-hard problem. In order to overcome the high time complexity, many population-based heuristic stochastic optimization (PHS) algorithms are used to deal with such NP-hard problems. In this study, the combination of discretization code and TLBO algorithm is tailored to seek for the global optimal solution of this specific problem. We call it "discrete TLBO". The original TLBO is inspired from the learning and teaching behaviors in a classroom, where learners are regarded as search points distributed in the decision variable space corresponding to the population of solutions in PHSs. Specifically, in this paper, the decision variable space refers to all scheduling and maintenance plans consisting of various job release sequences, job-machine combinations, job processing sequences and predictive maintenance nodes. The learner with the best fitness is defined as the teacher in the classroom. That is, the plan with the minimum objective function value F got by the formula (3) is regarded as the teacher of the classroom. Unlike traditional PHSs, the iterative evolution process of original TLBO includes the teaching phase and the learning phase. To enhance the average knowledge level of the class, learners improve their knowledge levels by learning from the teacher in the teaching phase, and they also improve their knowledge levels by learning interactively (group discussion) from another learner selected randomly in learning phase. The whole propagation of original TLBO is explained below.

For a minimization optimization problem with D-dimensional decision variables, let $\mathbf{X}_k = \{x_k^1, x_k^2, \ldots, x_k^D\}$ represents the kth learner (search point) and $f(\mathbf{X}_k)$ represents the fitness function (F) of this learner, NI is the number of learners in the population. The kth learner $\mathbf{X}_k = \{x_k^1, x_k^2, \ldots, x_k^D\}$ in the classroom can be randomly initialized generated by $X_k^j = X_j^{\min} + rand \cdot (X_j^{\max} - X_j^{\min})$, where $X_j^{\min}$ and $X_j^{\max}$ are the lower and upper bounds of the jth dimensional decision variable, respectively, rand is a random number within [0, 1]. During the teacher phase, learners improve their knowledge levels by learning from the difference between the teacher and the mean of the class, and the kth learner $\mathbf{X}_k$ in the classroom would update its knowledge according to the mechanism:

$$newX_k^j = X_k^j + rand_j \times \left(Teacher_j - T_F \times X_{mean}^j\right) \tag{15}$$

where $Teacher_j$is the jth dimensional decision variable of current best learner (search point) in the classroom, $X_{mean}^j = \frac{1}{NI}\sum_{k=1}^{NI} X_k^j$ is the current average knowledge level of the classroom in the jth dimension, $newX_k^j$ is the knowledge level of the kth learner in the jth dimension after teaching phase, $rand_j$ is the jth element in the random vector whose each element is a random number within the range [0, 1]. The value of T_F is either 1 or 2, which decides the magnitude of the mean to be changed. After learning from the teacher, the better learner between the learner and the new generated learner will be selected to enter the following learning phase, where the "better" meaning is that the smaller fitness function value.

In the learning phase, the learners will increase their knowledge through interaction among themselves. For the kth learner $\mathbf{X}_k$ in the

classroom, the updating mechanism is expressed as follows:

$$newX_k^j = \begin{cases} X_k^j + rand_j \cdot \left(X_k^j - X_r^j\right), & iff(\mathbf{X}_k) < f(\mathbf{X}_r) \\ X_k^j + rand_j \times \left(X_r^j - X_k^j\right), & otherwise \end{cases} \tag{16}$$

where $\mathbf{X}_r = \{X_r^1, X_r^2, \ldots X_r^j, \ldots, X_r^D\}$ is a randomly selected learner in the classroom, $r \neq k$. $newX_k^j$ is the knowledge level of the kth learner in the jth dimension after learning phase. $f(\mathbf{X}_k)$ and $f(\mathbf{X}_r)$ are the fitness values of the learner $\mathbf{X}_k$ and $\mathbf{X}_r$ respectively. Similarly, after learning from the other learner, the better learner between the learner and the new generated learner will be selected to enter the next teaching phase. And repeat the above until the maximum number of iterations is met. It should be noted that through the two stages, the decision variables may overflow their upper and lower limits. If overflowing, they only need to be re-selected randomly within their boundary range.

In the present problem, the job release sequence variable, job machining sequence variable, matching relationship variable between machines and jobs, and maintenance node variable cannot be directly applied to the mathematical operations. In this paper, we map these discrete variables to the continuous real coding based on the ranked order value and the random grouping. The specific process is as follows.

In the kth learner, for the job releasing sequences, a priority sequence vector $\mathbf{Se}_k = (Se_{1k}, Se_{2k}, \cdots, Se_{ik}, \cdots, Se_{nk})$ is created by $Se_{ik} = 1 + rand \times (n-1)$, n denotes the number of jobs in an order, k is the learner index. Then, we convert the priority sequence vector $\mathbf{Se}_k$ to the releasing permutation sequence $\mathbf{S}_k$ by decreasing order. The location code in $\mathbf{Se}_k$ represents the job index, while the location code in $\mathbf{S}_k$ represents the machining sequence index. The value at position i of the vector $\mathbf{Se}_k$ is displayed at position j of the vector $\mathbf{S}_k$ after the vector $\mathbf{Se}_k$ sorting, which means that the job i will be machined in the position j without considering the allocation relationship between the machines and the jobs. The priority machining sequence vector $\mathbf{Pr}_k = (pr_{1k}, pr_{2k}, \cdots, pr_{ik}, \cdots, pr_{nk})$ and the corresponding permutation sequence vector $\mathbf{P}_k$ are also set in the same way. For the matching relationship of machines and the jobs, we use the radian coding to digitize combinatorial numbers that cannot be mathematically operated, where, a radian vector $\mathbf{Ra}_k = (Ra_{1k}, Ra_{2k}, \cdots, Ra_{ik}, \cdots, Ra_{nk})$ is first created, $Ra_{ik} = 2\pi \times rand$. Then, we convert the radian vector $\mathbf{Ra}_k$ to the combination vector $\mathbf{Cv}_k = (Cv_{1k}, Cv_{2k}, \cdots, Cv_{ik}, \cdots, Cv_{nk})$ by the radian limit (17). For the predictive maintenance activity of machines, an initial maintenance vector $\mathbf{Md}_k = (md_{1k}, md_{2k}, \cdots, md_{ik}, \cdots, md_{hk})$, $md_{ik} = rand(\cdot)$ is randomly generated, then a maintenance node vector $\mathbf{Mn}_k = (mn_{1k}, mn_{2k}, \cdots, mn_{ik}, \cdots, mn_{hk})$ wound be obtained by the formula (18).

$$Cv_{ik} = \mathrm{fix}\left(\mathrm{mod}\left(Ra_{ik}, 2\pi\right) \Big/ \left(\frac{2\pi}{m}\right)\right) + 1 \tag{17}$$

$$mn_{ik} = \mathrm{mod}\left(\mathrm{round}(md_{ik}), 2\right) \tag{18}$$

where m is the number of all machines in one workshop, $\mathrm{fix}(\cdot)$, $\mathrm{mod}(\cdot)$ and $\mathrm{round}(\cdot)$ are the quotient function, remainder function and rounding function, respectively. $mn_{ik} = 1$ means that the machine for machining the job i needs to be repaired before processing the job i, $mn_{ik} = 0$ means the machine for preparing machining job i does not need to be repaired before processing the job i, h is the difference between the number of all the same values and the number of the same value categories in the vector $\mathbf{Cv}_k$. Such principle of valuing can ensure that there is no maintenance insertion before the first machining position of all machines.

In order to illustrate the discretization logic for $\mathbf{X}_k = \{\mathbf{Se}_k, \mathbf{Pr}_k, \mathbf{Ra}_k, \mathbf{Md}_k\}$ more clearly, we give the following numerical example: if the job number $n = 6$, the machine number $m = 3$, we can randomly generate the priority sequence vector $\mathbf{Se}_k = (2.7901, 4.5272, 2.6809, 1.3392, 3.8207, 5.0027)$ with $n = 6$ elements in the range of 1 to $n = 6$, the priority machining sequence vector $\mathbf{Pr}_k = (1.9004, 3.0210, 5.7248, 2.8977, 4.0091, 2.7728)$ with $n = 6$ elements in the range of 1 to $n = 6$, and the radian vector $\mathbf{Ra}_k = (3.6701, 0.0499, 6.2312, 5.3382, 2.1706, 4.2087)$ with $n = 6$ elements in the fixed range of 0 to 2π. Then, the priority sequence vector $\mathbf{Se}_k$ will become the releasing permutation sequence vector $\mathbf{Se}_k' = (5.0027, 4.5272, 3.8207, 2.7901, 2.6809, 1.3392)$ after the descending sort of $\mathbf{Se}_k$. In the same way, the sorted priority machining sequence vector $\mathbf{Pr}_k' = (5.7248, 4.0091, 3.0210, 2.8977, 2.7728, 1.9004)$ can be obtained by sorting $\mathbf{Pr}_k$ in the descending order. We can see the first element 5.0027 of $\mathbf{Se}_k'$ comes from the sixth element of $\mathbf{Se}_k$, then the first element of the releasing permutation sequence $\mathbf{S}_k$ is 6. And the second element 4.5272 of $\mathbf{Se}_k'$ comes from the second element of $\mathbf{Se}_k$, then the second element of the vector $\mathbf{S}_k$ is 2. We can deduce the whole releasing permutation sequence $\mathbf{S}_k = (6, 2, 5, 1, 3, 4)$ from this. In the same way, we can get the whole permutation sequence vector $\mathbf{P}_k = (3, 5, 2, 4, 6, 1)$ by the vectors $\mathbf{Pr}_k'$ and $\mathbf{Pr}_k$. $\mathbf{S}_k = (6, 2, 5, 1, 3, 4)$ means that the job release sequence is the job 6, job 2, job 5, job 1, job 3 and job 4 in turn, and $\mathbf{P}_k = (3, 5, 2, 4, 6, 1)$ means that the job machining sequence before assigning to machines is the job 3, job 5, job 2, job 4, job 6 and job 1 in turn. In addition, through the formula (17), the combination vector $\mathbf{Cv}_k = (2, 1, 3, 3, 2, 3)$ can be gained. The element 1 in the vector $\mathbf{Cv}_k$ is its second element, while the second element in the permutation sequence vector $\mathbf{P}_k$ is 5, which means that the machine 1 only processes the job 5. Similarly, the element 2 in the vector $\mathbf{Cv}_k$ is its first and fifth elements, while the first and fifth elements in the permutation sequence vector $\mathbf{P}_k$ are 3 and 6, which means that the machine 2 processes the job 3 and job 6 in turn. In the same way, the machine 3 will process the job 2, job 4 and job 1 in turn.

At this stage, for the vector $\mathbf{Cv}_k = (2, 1, 3, 3, 2, 3)$, the number of element 1 is 1 (the machine 1 has one job to process), the number of element 2 is 2 (the machine 2 has two jobs to process), and the number of element 3 is 3 (the machine 3 has three jobs to process). So, without considering the maintenance for the machines with no jobs and only one job, there are a total of 5 insertable maintenance positions in the assumption that maintenance activities can only be inserted between consecutive machining positions or before the first machining position, as shown in Fig. 3a. In addition, inspired by the assumption that any machine is intact before its first machining position, it is not worth inserting 2 maintenance activities before the first machining positions on the machine 2 and machine 3. Thus, there are actually $h = 5\text{-}2 = 3$ candidate maintenance positions in all for all machines in the kth learner (the kth solution). These three candidate maintenance positions are (1) between the job 3 and job 6 on the machine 2, (2) between the job 2 and job 4 on the machine 3, and (3) between the job 4 and job 1 on the machine 3.

Then, the initial maintenance vector $\mathbf{Md}_k = (0.3427, 0.7461, 0.2290)$ with h=3 elements in the fixed range of 0–1 can be randomly generated. Through the formula (18), the maintenance node vector $\mathbf{Mn}_k = (0, 1, 0)$ will be got, which can decide which candidate maintenance position is really selected. The element 1 in $\mathbf{Mn}_k$ indicates being selected, while the element 0 in $\mathbf{Mn}_k$ indicates no being selected. We can see that the element 1 is the second element of $\mathbf{Mn}_k$, that is, the second candidate maintenance position is selected. Therefore, as shown in Fig. 3a, a predictive maintenance activity needs to be inserted before machining the job 4 on the machine 3, while the maintenance activity is not required before other machining positions.

As mentioned above, considering the job-machine release time, Fig. 3b illustrates the said representative solution with a 6-jobs and 3-machines case. In conclusion, all independent variables that X_{ijk}, Y_{jk} and V_{ig} in the discrete form in the proposed MIP model can be obtained indirectly according to the four continuous vectors $\mathbf{X}_k = \{\mathbf{Se}_k, \mathbf{Pr}_k, \mathbf{Ra}_k, \mathbf{Md}_k\}$ and the said conversion relationship.

The discretization strategy of TLBO algorithm is to: (1) replace the job release sequence through the vector $\mathbf{Se}_k'$ in descending order of the first group of random consecutive values $\mathbf{Se}_k$; (2) replace the job machining sequence through the vector $\mathbf{Pr}_k'$ in descending order of the

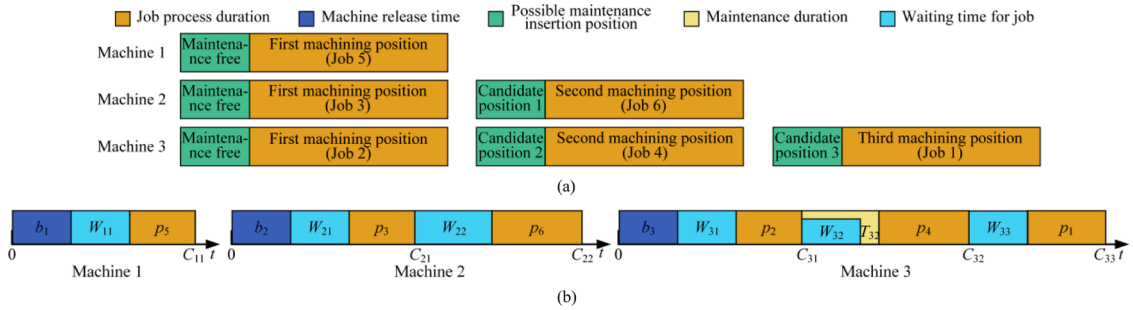


Fig. 3. Two illustrations of representative solutions. (a) Candidate maintenance position. (b) Makespan timeline.

second group of random consecutive values $\mathbf{Pr}_k$; (3) take the random consecutive values $\mathbf{Ra}_k$ as dices, and then throw these dices to the radian interval [0-2π] where the job-machine combination will be determined by all landing points; (4) judge whether the random consecutive values $\mathbf{Md}_k$ is closer to 1, where triggering maintenance if the value is closer to 1.

The basic steps of the discrete TLBO algorithm are as following:

Step 1: Randomly receive an unassigned order with n jobs and m machines;

Step 2: Count the machining duration of all jobs, the release preparation duration of all jobs and the release time of all machines. Set hyper-parameters: the maximum number of iterations $T_{\max}$, the number of learners *NI*, the comprehensive weight λ for the enlistment age and the initial rated life of machine, the weight coefficient u for the machining penalty beyond RUL threshold;

Step 3: Randomly generate NI groups of random vectors $\mathbf{Se}_k$, $\mathbf{Pr}_k$, and $\mathbf{Ra}_k$ in a certain range;

Step 4: Obtain NI discrete vectors $\mathbf{S}_k$, $\mathbf{P}_k$, $\mathbf{Cv}_k$ and corresponding scalar h indicating the number of elements of $\mathbf{Md}_k$;

Step 5: Through NI$\mathbf{Md}_k$ in the range of [0, 1] calculate $\mathbf{Mn}_k$;

Step 6: Convert $\mathbf{S}_k$, $\mathbf{P}_k$, $\mathbf{Cv}_k$ and $\mathbf{Mn}_k$ to the form of X_{ijk}, Y_{jk} and V_{ig};

Step 7: Receive the output of the RUL prediction model, and for each learner, calculate the release time of each job, the processing duration of each machining position, the enlistment age of machines after each machining position, the maintenance duration before each machining position, the candidate waiting time for jobs before each machining position and the completion time of each machining position in turn;

Step 8: Calculate the objective functions (fitness functions) of all learners: the sum of the makespan and the machining penalty beyond RUL threshold;

Step 9 (Teaching phase): Select the best learner (search point with the smallest fitness value) among all current learners as the teacher, update other learners' knowledge by learning from the teacher (formula (15)), and the better learner between the original learner and the updated learner will be selected to enter the following learning phase.

Step 10 (Learning phase): Learners learn interactively through group discussions to increase their knowledge levels by the formula (16), and then the better learner between the original learner and the increased-knowledge learner will be selected to enter the Step 4.

Step 11: After the maximum number of iterations is met, output the job release sequence, the job allocation in each machine, the job processing sequence and each maintenance starting point corresponding to the best learner in the $T_{\max}$ iteration.

4. Fault mode assited RUL prediction

In this study, the proposed RUL prediction method is divided into four parts: (1) the fault mode diagnosis, (2) the determination of the first prediction time (FPT), (3) the training of multiple candidate RUL prediction models based on degeneration data with multiple different fault modes and (4) predicting the RUL of monitored component after the FPT by using the RUL prediction model corresponding to the diagnosed fault mode.

4.1. Fault mode diagnosis

Here, we first extract the deep features from fault data through one-dimensional convolutional neural network (1D CNN), then classify these features with the cross entropy cost, and finally optimize all network parameters in each layer by using AdaBelif optimizer [43] until the maximum number of iterations is met.

1D CNN has been extensively employed as a feature extractor to cope with the degradation data, which has a unified process: (1) data input, (2) deep feature extraction, (3) feature classification Fig.4. clearly shows the basic structure of 1D CNN, including a feature extractor and a classifier. In this paper, the feature extractor is structured by a stack of one-dimensional convolutional building blocks, consisting of a convolutional layer and a pooling layer, to learn deep fault features of raw degradation signals, where the convolutional layer aims to extract multilayer high-dimensional features, while the pooling layer aims to reduce the feature dimension and the time complexity.

For each convolutional layer, it firstly uses a set of convolution kernels to convolve outputs of last pooling layer (the convolution objects are raw signals if the last layer is the input layer), and then applies an element-wise activation function on the outputs of convolution operations. Now, let $\mathbf{k}^l \in \mathbb{R}^{F\times 1\times C\times N}$ and $\mathbf{p}^{l-1} \in \mathbb{R}^{H\times 1\times C}$ denote the convolution kernels and the input volume, respectively, where $F\times 1$ is the size of convolution kernels, C is the number of input channels, N is the number of convolution kernels, and H is the length of the input volume. Mathematically, the nth feature of the lth convolutional layer $\mathbf{x}_n^l$ can be got by:

$$\mathbf{x}_n^l = \sigma_r\left(\mathbf{u}_n^l\right) \tag{19}$$

$$\begin{cases} \mathbf{u}_n^l = \mathbf{k}_n^l * \mathbf{p}^{l-1} + b_n^l = \sum_{c=1}^{C} \mathbf{k}_{n,c}^l * \mathbf{p}_c^{l-1} + b_n^l & l \geq 2 \\ \mathbf{u}_n^l = \mathbf{k}_n^l * \mathbf{x} + b_n^l = \sum_{c=1}^{C} \mathbf{k}_{n,c}^l * \mathbf{x}_c + b_n^l & l = 1 \end{cases} \tag{20}$$

where $\sigma_r(\cdot)$ is the rectified linear unit (ReLU) activation function, $\mathbf{u}_n^l$ is the output of convolution operations, $\mathbf{k}_n^l$ is the nth convolution kernel, $*$ is the convolution operator, and b_n^l is the bias term.

For each convolution block, its pooling layer is placed after its convolution layer, and then, in this paper the max pooling is selected as the pooling operations to reduce the dimension of each output of activation function. The operations divide the features into several non-overlapping segments and return their maximum values according to the following rule:

$$\mathbf{p}_{i,n}^l = \max\left\{\mathbf{x}_{j,n}^l \middle| p(i-1)+1 \leq j \leq pi\right\} \tag{21}$$

where i and j are the positional indexes, $\mathbf{p}_{i,n}^l$ and $\mathbf{x}_{j,n}^l$ are the elements of corresponding positions in the features $\mathbf{p}_n^l$ and $\mathbf{x}_n^l$, respectively, and p is the height of the segment.

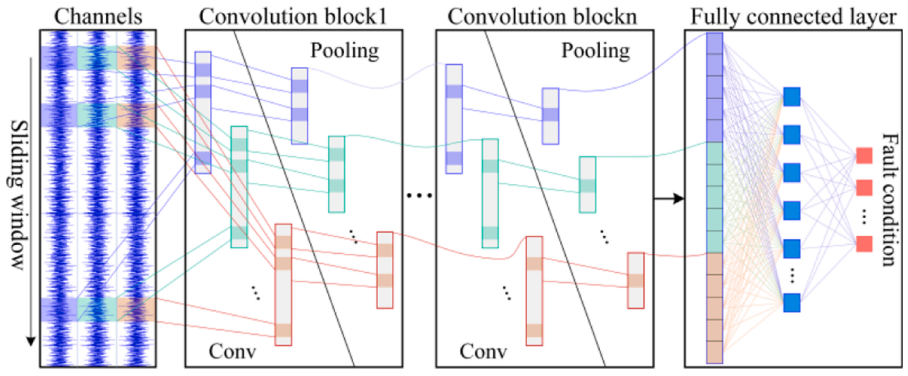


Fig. 4. The illustration of 1D CNN.

High-level features, containing more easily classified degradation information than shallow-level features, can be obtained by stacking convolutional blocks in order, and then the features in the last pooling layer are flatten into a one dimensional vector, which is the output of the first fully connected layer in the classier. Similar to the common neural networks, the features in the second fully connected layer are computed as follows

$$\mathbf{x}^{F2} = \sigma_r\left(\mathbf{w}^{F2}\mathbf{x}^{F1} + \mathbf{b}^{F2}\right) \tag{22}$$

where $\mathbf{w}^{F2}$ and $\mathbf{b}^{F2}$ denote the weights and bias in the layer F2.

The softmax function is applied in the last layer of the classier to predict the probability distribution of the fault modes for each sample by

$$P\left(y=j|\mathbf{x}^{F2}\right) = \frac{\exp\left(\left(\mathbf{w}_j^{F3}\right)^T\cdot\mathbf{x}^{F2} + \mathbf{b}^{F3}\right)}{\sum_{j=1}^{k}\exp\left(\left(\mathbf{w}_j^{F3}\right)^T\cdot\mathbf{x}^{F2} + \mathbf{b}^{F3}\right)} \tag{23}$$

where $P(y=j|\mathbf{x}^{F2})$ is the probability distribution of label j for a sample, $\mathbf{w}^{F3}$ and $\mathbf{b}^{F3}$ represent the weights and bias in the layer F3, respectively.

As for the backward propagation of 1D CNN, the gradient descent algorithm AdaBelief [43], having both the fast convergence of Adam and the excellent generalization of Stochastic Gradient Descent (SGD), is applied to optimize all network parameters by minimizing the cross entropy cost:

$$f = \underset{\theta=\{\mathbf{w},\mathbf{b}\}}{\operatorname{argmin}}\frac{1}{n}\sum_{i=1}^{n}J(fa_i, f\widehat{a}_i) \tag{24}$$

where n is the total number of samples, $J(\cdot,\cdot)$ is the cross entropy function, fa_i and $f\widehat{a}_i$ denote the real fault labels and predicted fault labels, respectively.

4.2. FPT determination

The international standard ISO 2372 [44] has gave an industry standard for mechanical vibration: when RMS value of the medium mechanical vibration signal reaches 2.0–2.2 g, the monitored component is on dangerous ground, while other time domain characteristics are still lack of unified definition standard. Moreover, different from the stable degradation stage when RMS value maintains a little fluctuation, in the late stage of degradation, RMS value will rapidly increase and shake because that the component damage has severely affected the normal operation of machines. Therefore, it is reasonable to regard RMS value as the health indicator (HI) of machines, which can meet the demand for the tracking metric and dimension reduction, as shown in the formulae (25) and (26).

$$\mathbf{HI} = \{H_0, H_1, H_2, \ldots, H_t\} \tag{25}$$

$$\begin{cases} H_t = \sqrt{\sum_{j=Ws*t}^{Ws*t+Sl}\left(x_j\right)^2 \Big/ Sl} \\ s.t.x_j \in \mathbf{X}_{Ws*t\to Ws*t+Sl} \end{cases} \tag{26}$$

where t is the index of sliding window of signals for RMS, H_t denotes the health indictor (RMS) in the tth period of degradation data, Ws and Sl are the step size and window length of signals for RMS, respectively.

The current HIs may have some random fluctuations caused by raw signals with environmental impacts and measurement errors, as shown in Fig. 5(b), which will result in the possible dispersion of RUL prediction results. Isotonic regression (IR) [45], which can remove all descending sections, is introduced to solve this problem.

Firstly, define a finite set of real numbers $\mathbf{y} = \{y_1, y_2, \ldots, y_c\}$ representing the observed response and an ordered unknown response set $\mathbf{h} = \{h_1, h_2, \ldots, h_c\}$. Then, train a model to minimize the following weighted sum of squared errors:

$$\sum_{i=1}^{c} w_i(y_i - h_i)^2, s.t.h_1 \le h_2 \ldots \le h_c \tag{27}$$

where w_i is the weight of the ith observed response, c is the number of responses, y denotes the original HIs, and h is the processed monotonous HIs, which is what we need.

A number of optimization approaches have been used to solve the IR optimization problem, in this study, a common method Pool-Adjacent-Violators (PAV) [45] is considered to acquire monotonous h. Firstly, we need to judge whether y is monotonous. If y is monotonically increasing, then h=y, otherwise, (resp. y_i>y_{i+1}), the two responses will be replaced with their weighted mean Wm_i, and the two weights w_i and w_{i+1} are also replaced by their sum Sm_i:

$$Wm_i = \frac{y_i w_i + y_{i+1}w_{i+1}}{w_i + w_{i+1}} \tag{28}$$

$$Sm_i = w_i + w_{i+1} \tag{29}$$

Next, if $y_1 \ll \ldots \ll y_{i-1} \ll Wm_i \ll \ldots \ll y_c$, then $h_i = h_{i+1} = Wm_i$, and $h_j = y_j\, j \neq i, i+1$. Then, if the current series is not isotonic, then the formulae (28) and (29) should be repeated until a sequence with global monotonicity is obtained, which is consistent with the physical performance degradation process of machine vulnerable systems. Here, we can use the monotonic HI paths for reliable stage division and FPT determination to economically predict the RUL of machines.

Finally, we use a comparison of the degradation increment per unit time with a given increment threshold to detect FPT. When the increment per unit time is lower than the given threshold ϑ, the monitored components are still at the health stage, while when the increment per unit time exceeds the threshold ϑ, they are regarded as entering the rapid degradation stage. In other words, the left side of the time period when the threshold is first exceeded is the FPT point. The specific pro-

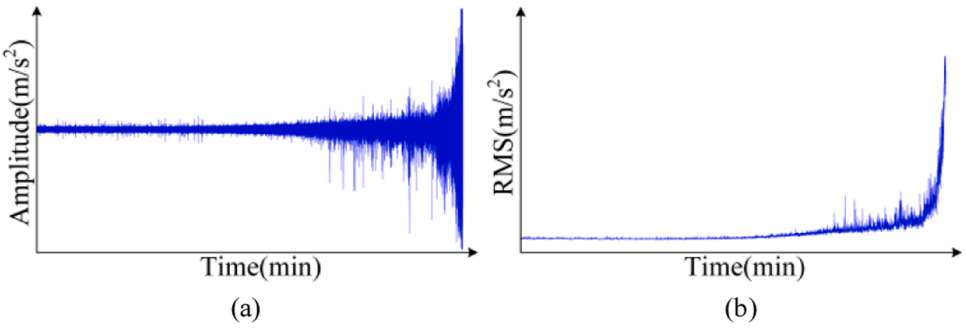


Fig. 5. The diagram of random fluctuation. (a) Raw signals. (b) RMS.

cess is as follows: a sliding window $\mathbf{R}_i = \{h_i, \ldots, h_{i+g-1}\}$ with g points is constructed from monotonous HIs, and the slope ϑ_i of HI segment in the sliding window $\mathbf{R}_i$ is fitted using least square method (LSM), as shown in the formula (30). For each window $\mathbf{R}_i$, the fitted slope ϑ_i is compared with the given threshold ϑ, if $\vartheta_i \leq \vartheta$, meaning that it is at the health stage, the sliding window will slide with a certain step p and accept new degradation features in h. This step will be repeated until the time period when $\vartheta_i > \vartheta$. At this time, the first time node of $\mathbf{R}_i$ is taken as the boundary point of degradation stage and then FPT is determined.

$$\vartheta_i = \frac{\sum_{j=i}^{i+g-1} t_j h_j - \frac{1}{n}\sum_{j=i}^{i+g-1} t_j \sum_{j=i}^{i+g-1} h_j}{\sum_{j=i}^{i+g-1} t_j^2 - \frac{1}{n}\left(\sum_{j=i}^{i+g-1} t_j\right)^2} \tag{30}$$

where t_j is the corresponding time point of h_j.

4.3. RUL prediction

Gate recurrent unit (GRU) is used to predict the RUL in this paper. The GRU network uses the hidden state k to memorize the historical fault knowledge regularly, where it only has two gates: a reset gate and an update gate. As shown in Fig. 6, the update gate function is to distinguish what degradation information should be kept in mind and what information should be discarded appropriately, while the reset gate function is to decide what to keep from the learned knowledge. The detailed derivation of GRU is as follows.

In Fig. 6, $\mathbf{k}_s$ and $\mathbf{o}_s$ denote the hidden state and the observation state in the sth time period. The input of prediction model $\mathbf{h}_s$ is a nonoverlapping segment divided from the health indicator $\mathbf{H}(t)$, $\mathbf{h}_s \in \mathbf{H}(t)$. The hidden state $\mathbf{k}_s$ can be obtained by exploring the relationship between the current model input $\mathbf{h}_s$ and the hidden state $\mathbf{k}_{s-1}$ in the last period. Obviously, $\mathbf{k}_s$ contains the current degradation information and the historical degradation information in the last time period. As an analogy, the current hidden state $\mathbf{k}_s$ covers all historical information. Thus, GRU network can be utilized to analyze and learn some time-related sequence data. Specially, its reset gate $\mathbf{r}_s$ is introduced to control the historical information content to be propagated from $\mathbf{k}_{s-1}$ to $\widetilde{\mathbf{k}}_s$, where $\widetilde{\mathbf{k}}_s$ is the hidden state candidate at the sth time period. If $\mathbf{r}_s = 0$, then the entire historical information is dismissed, while $\mathbf{r}_s = 1$, then the whole historical information is used.

$$\mathbf{r}_s = \sigma(\mathbf{W}_{rk}\mathbf{k}_{s-1} + \mathbf{W}_{rh}\mathbf{h}_s) \tag{31}$$

where $\sigma(\cdot)$ is the sigmoid function, $\mathbf{W}_{rk}$ and $\mathbf{W}_{rh}$ are the weights for $\mathbf{k}_{s-1}$ and $\mathbf{h}_s$, respectively.

Similarly, the update gate $\mathbf{z}_s$ helps the model decide how much information from the past should be passed on to the future. The greater the update gate, more historical degradation information is transferred. It can be obtained by (32).

$$\mathbf{z}_s = \sigma(\mathbf{W}_{zk}\mathbf{k}_{s-1} + \mathbf{W}_{zh}\mathbf{h}_s) \tag{32}$$

where $\mathbf{W}_{zk}$ and $\mathbf{W}_{zh}$ are the weights for $\mathbf{k}_{s-1}$ and $\mathbf{h}_s$, respectively. The added hidden state candidate $\widetilde{\mathbf{k}}_s$ is calculated by (33).

$$\widetilde{\mathbf{k}}_s = \tanh(\widetilde{\mathbf{W}}_{kh}\mathbf{h}_s + \widetilde{\mathbf{W}}_{kk}(\mathbf{r}_s \odot \mathbf{k}_{s-1})) \tag{33}$$

where $\widetilde{\mathbf{W}}_{kh}$ and $\widetilde{\mathbf{W}}_{kk}$ are the weights for $\mathbf{h}_s$ and $\mathbf{r}_s \odot \mathbf{k}_{s-1}$, respectively.$\odot$ represents the element-wise multiplication.

The hidden state in the sth time period $\mathbf{k}_s$ and the final predicted RUL can be gained by the formulae (34) and (35):

$$\mathbf{k}_s = (1 - \mathbf{z}_s) \odot \mathbf{k}_{s-1} + \mathbf{z}_s \odot \widetilde{\mathbf{k}}_s \tag{34}$$

$$RUL_s = \sigma(\mathbf{o}_s), \mathbf{o}_s = \sigma(\mathbf{W}_o\mathbf{k}_s) \tag{35}$$

where $\mathbf{W}_o$ are the weight for $\mathbf{k}_s$. It can be seen from the formulae in the process of forward propagation that the parameters to be learned are $\mathbf{W}_{rk}$, $\mathbf{W}_{rh}$, $\mathbf{W}_{zk}$, $\mathbf{W}_{zh}$, $\widetilde{\mathbf{W}}_{kh}$, $\widetilde{\mathbf{W}}_{kk}$ and $\mathbf{W}_o$. After getting theRUL_s, we can acquire the total loss E of a single sample at all time periods.

$$E = \sum_{s=1}^{S} \left(RUL_s^{real} - RUL_s\right)^2 \Big/ 2 \tag{36}$$

where S is the total number of time periods, and $RUL_s^{real}$is the real RUL value in the sth time period.

All the hyper-parameters involved in the above process, e.g. the window length g, the sliding step p and the increment threshold ϑ, etc., are related to the application environment and dynamic characteristics of the system in which they are located. Normally, we need to use a large amount of historical degradation information to analyze the relationship between the increment per unit time and the stage division, and then determine them appropriately. In conclusion, the fault mode assisted

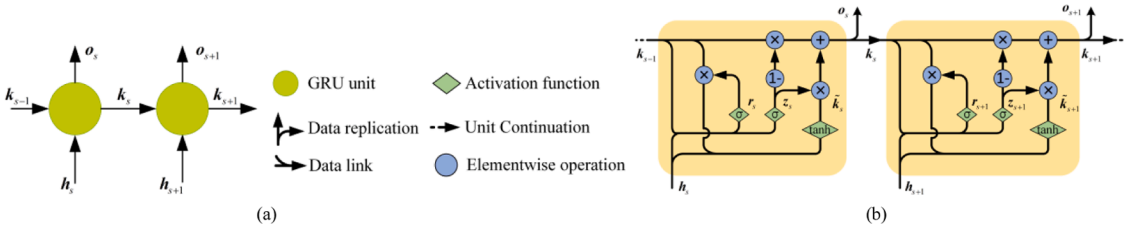


Fig. 6. The sketch maps for GRU. (a) Unexpanded diagram. (b) Expanded diagram.

GRU method (FGRU) is shown in Fig. 7.

The summary is as follows: multi GRU networks are trained to estimate RUL of monitored component offline, where reset gate discards some irrelevant history information, and update gate controls the historical degradation information content to be propagated from the last hidden state to the current hidden state. The major theme of RUL prediction module is that one GRU model corresponds to one kind of fault mode, that is to say, after FPTs of all samples are detected, each GRU is used to model one degradation path under one fault mode, and then multiple predicted RUL candidates are obtained. Finally, a unique predicted RUL value as final result output is selected from these RUL candidates based on the current diagnosed fault mode, as shown in Fig. 8.

The following algorithm # gives the detailed procedures of the proposed method

5. Case study

To verity the practicability and effectiveness of the proposed approach, we carry out two degeneration cases for RUL prediction and three simulation data for the joint decision-making. All of models and algorithms were coded in Matlab 2018 and ran on a personal computer with a 2.10 GHzn2 CPU and 4.0 GB RAM. In the first degeneration case, an accelerated degradation test platform for roller bearings regarded as a vulnerable part of machine was built, and the vibration signals were collected online to access their degradation status. In the second degeneration case, XJTU-SY bearing datasets [46] were used to further validate the stability of our method by comparing with other common RUL prediction methods. In the simulation data test of means of production, this paper compared the joint decision-making method with periodic maintenance strategy and other preventive maintenance methods. The detailed test processes are described below.

5.1. Test configuration and data discription

In the two degradation cases, it is considered that all bearings will fail completely when the amplitude of the collected signals is 10 times higher than that of stationary stage. The loads and speeds of two degradation cases are shown in Table 1. The test configurations of FGRU and joint decision-making are shown in Table 2, and the simulation data about jobs and machines are shown in Table 3, where RUL data includes XJTU-SY and own data not described above.

In the first case, a simplified test-bed was established to accelerate the bearing degradation (reinforced SKF6205). As shown in Fig. 9a, the accelerated degradation test bed consisted of a three-phase motor, a support bearing, a roller bearing for testing, a rotating shaft with coupling and a hydraulic load system. The adjustable load was loaded onto a test bearing through an ejector. The horizontal and vertical mounted accelerometers were utilized to collect vibration signals in the intersection direction. The 3-phase motor was the exclusive motive source, whose speed was set and regulated by its speed controller. As shown in Table 1, three intact bearings (R1, R2, R3) were used to perform the first test. The sampling frequency was 25.6 kHz, and the sampling interval and duration were designated as 1 min and 1.28 s. After testing, we got three fault modes: (1) combined failure of cage and ball on R1 with the lifetime of 1639 min, (2) outer ring fault on R2 with the lifetime of 61 min, and (3) inner ring fault on R3 with the lifetime of 340 min. Thus, we built three GRU networks corresponding to these three kinds of degradation data. Partial degradation data were shown in Fig. 5(a). In the second case, XJTU-SY datasets were applied to further access the generalization of our method by comparing with three common methods. Similarly, the layout of test bench was shown in Fig. 9b, where five bearings (LDK UER 204) (B1-1, B1-2, B1-3, B1-5, B3-5) were subjected to degradation tests. The test-bed consisted of a drive motor, a support bearing, a testing bearing, a rotating shaft and a manual

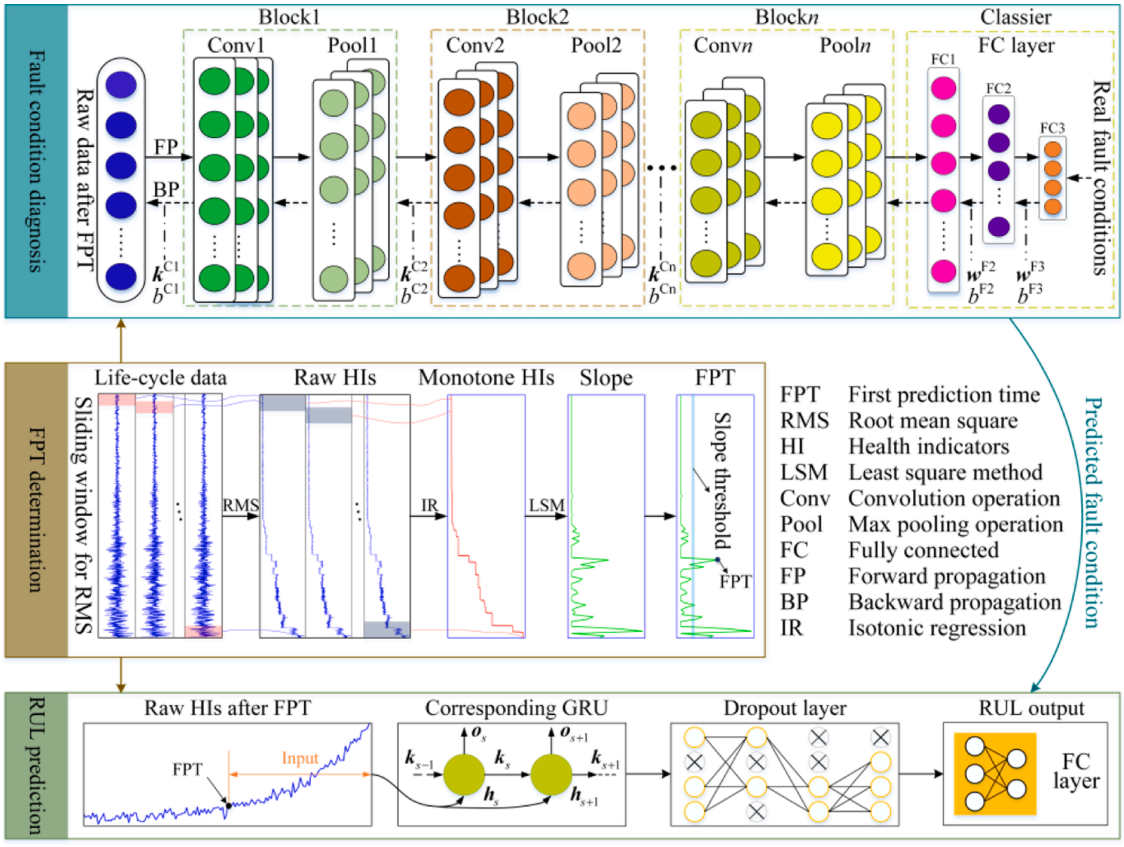


Fig. 7. The procedure of FGRU.

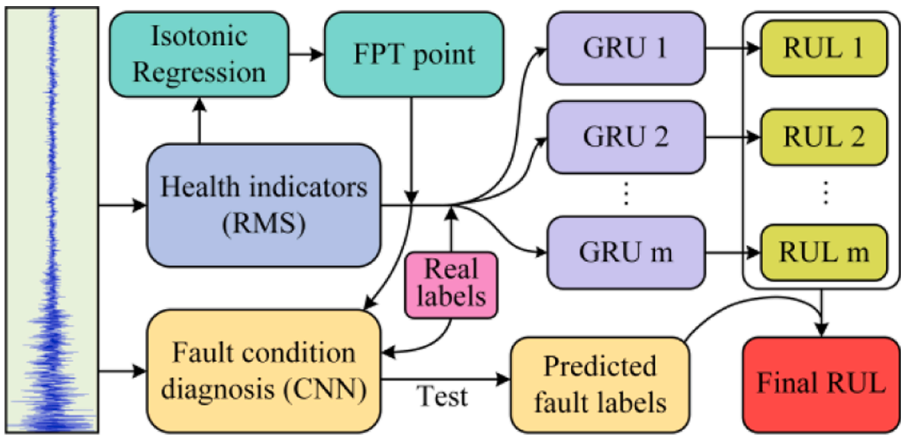


Fig. 8. The procedure of multiple GRU networks.

Table 1
Loads and speeds of two degradation cases.

Number	R1	R2	R3	B1-1/2/3/5	B3-5
Load (kg)	500	1000	1000	1200	4000
Speed (Hz)	20	35	30	35	10

Table 2
Test configurations of the proposed method.

Parameters	Values	Parameters	Values	Parameters	Values
Epoch1 of 1D CNN	180	Number of jobs	20	Channel number in Conv2	20
Epoch2 of GRUs	250	Initial θ of 1D CNN	0.01~0.03	Number of GRUs/fault modes	3
Epoch3 of TLBO	1500	Weight for processing penalty u	50	Number of learners	300
Sample length in 1D CNN	1200	Initial θ of GRUs	0.01~0.03	Number of machines	4
Iterative number of IR	300	Neuron number in F1 of 1D CNN	5940	Bias size in Conv1/2	20 × 1
Learning rate of 1D CNN	0.001	Neuron number in F3 of 1D CNN	3	Batch of 1D CNN	50
Number of convolution blocks	2	Pooling size	2	Number of kernels	20
Neuron number in F2 of 1D CNN	256	Channel number in Conv1	1	Size of kernels	5 × 1

hydraulic system. The sampling interval and duration were set as 1 min and 1.28 s, and the sampling frequency was 25.6 kHz. After testing, we got two fault modes: (1) outer ring fault on B1-1 with the lifetime 123 min, B1-2 with the lifetime of 161 min, B1-3 with the lifetime of 158 min and B3-5 with the lifetime of 114 min, (2) compound fault of inner and outer ring on B1-5 with the lifetime of 52 min Fig. 10. showed partial life-cycle vibration signals in the second case. In this study, Encoder-decoder RNN (ED-RNN) [47], Bidirectional LSTM (Bi-LSTM) [48] and single GRU are selected as the comparison models. Their configurations are similar to that of the proposed method.

5.2. Experiment results and analysis for FGRU

In this section, we focus on the feasibility and superiority of FGRU prediction model according to the method process and comparison methods. For FPT determination, Fig. 11 shows partial RMSs (original HIs) of bearings in two degeneration cases. It is seen clearly that there are a lot of random fluctuations in the original HIs. After IR processing, all fluctuations are eliminated completely and the degradation trend shows a monotonous growth Fig. 12. presents the fitted degradation gradients of processed HIs of three bearings in the first case for FPT determination.

According to Fig. 12, the FPTs of three bearings in the first case are 1072 min, 3 min and 257 min, respectively. Next, we will train a fault diagnosis model, 1D CNN, to obtain the prior knowledge of degradation modes online. The diagnosis performance can achieve 100 and 99.6% accuracy on the training set and the testing set, which is very promising for the subsequent selection of GRU models. Finally, according to the diagnosis results, we can choose a unique GRU prediction model trained by corresponding fault data after FPT to predict RUL of bearings online. The future degradation patterns of above three bearings are shown in Fig. 13. In Fig.13, considering that each predicted RUL curve after its FPT is not immediately lower than its RUL threshold, and all FPT points are within the first 90% of corresponding life cycles, the predicted RUL of the last 10% of life cycle is used here as a display of the fast degradation stage to highlight useful features. It can be clearly seen that the predicted RULs coincide well with the curve of real RUL, which can provide important dispatching guidance and sufficient time preparation for preventive maintenance planning. Moreover, we assume that RULs of bearings before their FPTs are replaced by the RULs at the FPTs, which does not affect the RUL prediction accuracy in the fast degradation stage whether in the testing sets or the training sets. Furthermore, ED-RNN, Bi-LSTM and Single GRU are also used to compare the prediction accuracy and the prediction divergence of the proposed method after FPT point. Kyunghyun et al [47]. proposed the ED-RNN in 2014 whose structure does not limit the sequence length of input and output, so it has a wide range of applications. Bi-LSTM combines the information of the input sequence in the forward and backward directions, and has achieved good results in the application scenarios of natural language processing and text translation.

In Fig.13(a, b), all methods follow the trend of observed values, and their predicted degradation trend is more intensive than the observed values. Especially, in terms of congruence, the RMS and RUL predicted by our method are more close to the center line of the actual dispersion, because that the proposed model has already known the fault type of the input degradation trajectory before prediction, and made the necessary selection of candidate prediction models. On the surface, the prediction result of single GRU is the most consistent with the observed value, but it is also caused by its large dispersion, in the actual operation of machine tool maintenance scheduling, which is very easy to trigger false alarm of machine failure. The RUL prediction of ED-RNN and Bi-LSTM is too

Table 3
Simulation data details (minutes).

Group	Job ID	Process duration	Job release duration	Job ID	Process duration	Job release duration	Job ID	Process duration	Job release duration	Job ID	Process duration	Job release duration	Machine ID	RUL	RUL threshold	Machine release duration
1	1	50	13	5	92	15	9	65	9	13	27	10	17	108	19	1	339	318	24
(λ =	2	62	20	6	34	12	10	75	18	14	58	8	18	48	17	2	114	108	17
9)	3	70	14	7	100	21	11	105	24	15	41	26	19	95	25	3	161	153	16
	4	90	16	8	28	0	12	80	11	16	84	24	20	37	4	4	240	224	25
2	1	313	21	5	353	30	9	151	10	13	390	16	17	162	12	1	1639	1520	100
(λ =	2	179	34	6	369	19	10	180	8	14	200	25	18	197	21	2	1515	1385	70
90)	3	250	7	7	260	28	11	400	37	15	360	37	19	276	19	3	821	790	35
	4	299	20	8	147	0	12	175	13	16	201	11	20	240	26	4	533	525	5
3	1	73	19	5	74	25	9	150	18	13	84	21	17	115	9	1	371	350	22
(λ =	2	169	31	6	140	30	10	160	20	14	185	30	18	65	27	2	491	452	36
40)	3	96	13	7	88	16	11	110	31	15	70	14	19	201	24	3	533	525	19
	4	129	27	8	145	0	12	60	15	16	175	16	20	130	35	4	339	318	27

conservative, where, at the life end of R1 bearing, it is still considered that there is a service life of about 350min, on the one hand, because that the sliding window for RMS we initially set is somewhat long and the dimension of available data is small enough, and the self-encoder aggravates the disappearance of feature details through dimension reduction, on the other hand, the existing fault state of R1 bearing is only related to the historical degradation trend, which is contrary to the original design intention of Bi-LSTM.

In Fig.13(c, d), it is clear that the performance degradation trend predicted by all prediction methods is no longer obvious, and seems that there is no rapid degradation stage here. This is not the problem of prediction models themselves, but the problem of the degradation data itself Fig.11.(b) can explain the phenomenon perfectly, where the slope of the degradation trend of R2 bearing does not increase significantly throughout the accelerated degradation test. That means that the historical degradation data of R2 bearing contains a smaller difference in the degradation trend at different times than that of R1 bearing, which makes it more difficult for the prediction models to extract the historical degradation features. Moreover, the vibration amplitude fluctuation of R2 bearing rises sharply when the life is about to end, when this piece of data is generally used as a prediction model test set, it further exacerbates the dispersion of the prediction results of the comparison methods. Our method happens to select the degradation mode that is closest to the real state of the bearing at each time period, and absorbs enough degradation trend characteristics, so the aggregation is better than the comparison methods, and its predicted degradation curve basically hoveres near the observed RUL curve.

In Fig.13(e, f), the output effect of the ED-RNN model is the worst among the four methods. Its point cloud center line is similar to a horizontal line, which hardly shows any degradation information of R3 bearing, due to its inherent characteristic of ignoring long-term degradation information and feature blinding of data encoding preprocessing. The Bi-LSTM model can independently and comprehensively weighs the effect of long-term and short-term historical degradation information on the future degradation information, which is consistent with the reality that the current actual degradation state of the bearing is affected by all historical degradation states. Therefore, its prediction effect is slightly improved, and the output point cloud is more biased towards the fitted center line of the observed value. For the single GRU model, without considering the dispersion, it is the most consistent with the real observation point cloud, but this aptitude has almost no practical significance, which is just to be closer to the normalized RMS dispersion. The original intention of our prediction is to bring the results closer to the degradation trend of normalized RMS. Obviously, the single GRU model has worked in the wrong direction here. On the contrary, the FGRU model recognizes this point. Although it also failed to follow the pace of real RUL in the later stage, its dispersion problem has been better solved, because it has learned a large amount of prior degradation knowledge that can be one-to-one with the candidate GRU models, is still the most excellent method among them. Ulteriorly, in addition to the qualitative analysis, we use RMSE and Mean Absolute Error (MAE) to quantitatively analyze the errors of all predicted RUL values over the life cycle.

$$RMSE = \sqrt{\frac{1}{S}\sum_{s=1}^{S}\left(RUL_s^{real} - RUL_s^{predict}\right)^2} \tag{37}$$

Similarly, FPTs of the five bearings in the second case are first detected by IR, whose results are shown in Table 4. Then, the diagnosis performance of fault mode estimator gets the accuracies of 100% and 98.4% on training sets and testing sets, respectively, which is sufficient for selecting the correct life prediction model. Here, taking B1-1, B1-2, B1-3, B3-5 and B1-5 as five examples, we use the above three comparison methods and FGRU to predict their RULs. The experiment results of B1-5 are shown in Fig. 14. We can see that the prediction errors of our method are better than those of other methods. The prediction results of

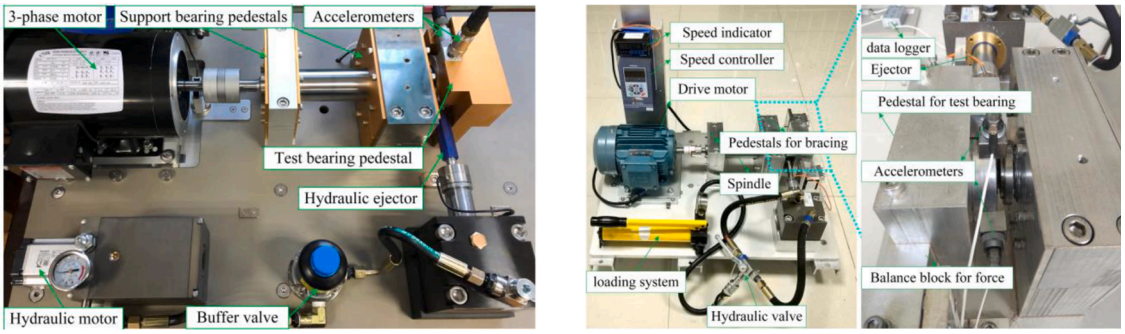


Fig. 9. The test bed layout in the two degeneration cases. (a) The first case. (b) The second case.

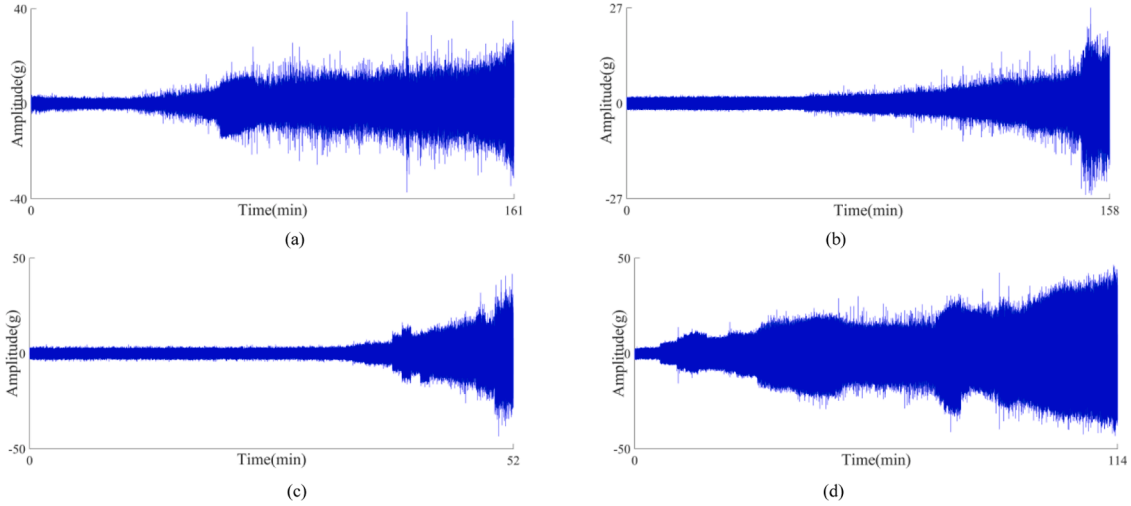


Fig. 10. The signals in the second case. (a) The horizontal direction of B1-2. (b) The horizontal direction of B1-3. (c) The horizontal direction of B1-5. (d) The vertical direction of B3-5.

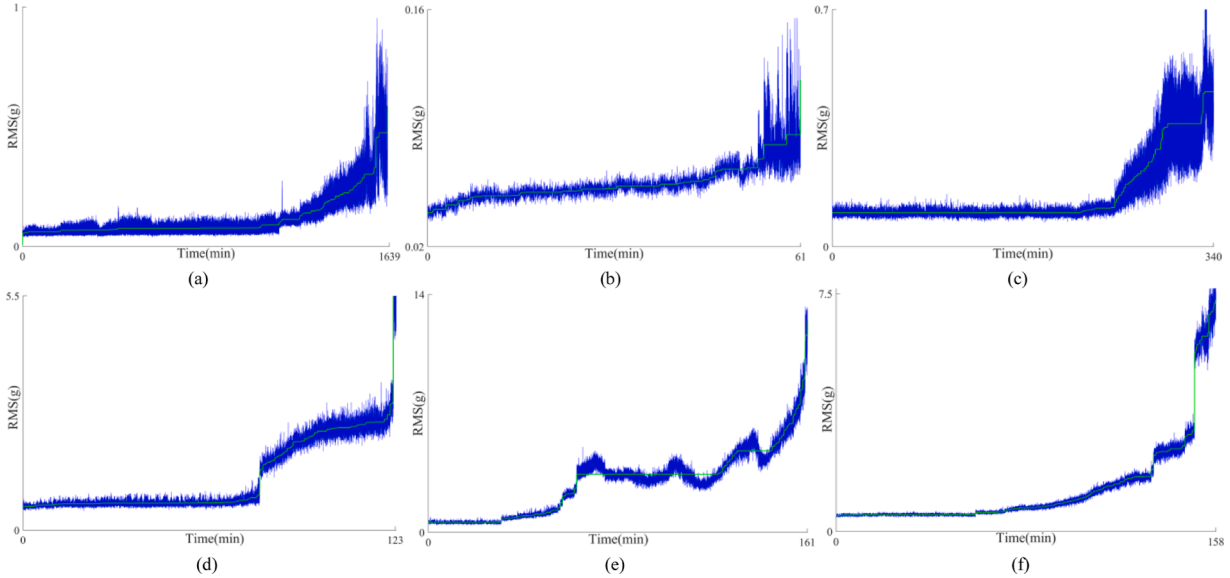


Fig. 11. Data preprocessing using IIR where the blue line is the original HIs while the green line is the processed HIs. (a) R1. (b) R2. (c) R3. (d) B1-1. (e) B1-2. (f) B1-3.

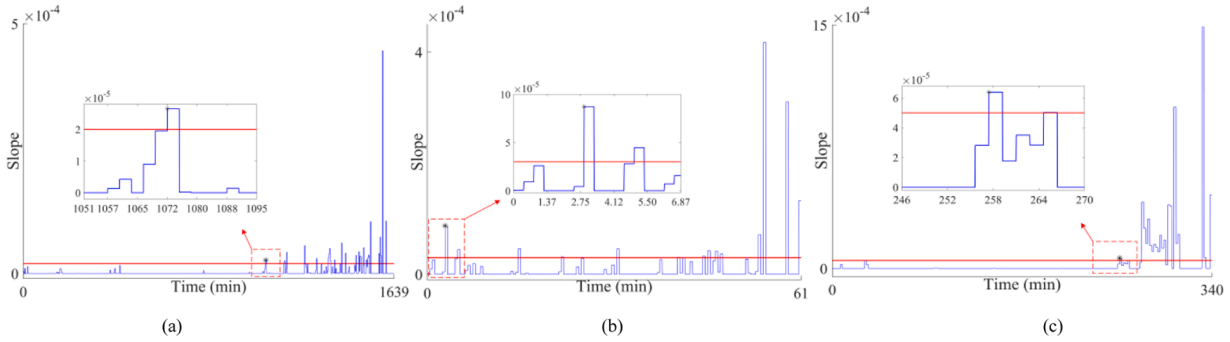


Fig. 12. The appropriate FPTs for three bearings in the first case where the red line is the given slope threshold and the blue line is the fitted slope. (a) R1. (b) R2. (c) R3.

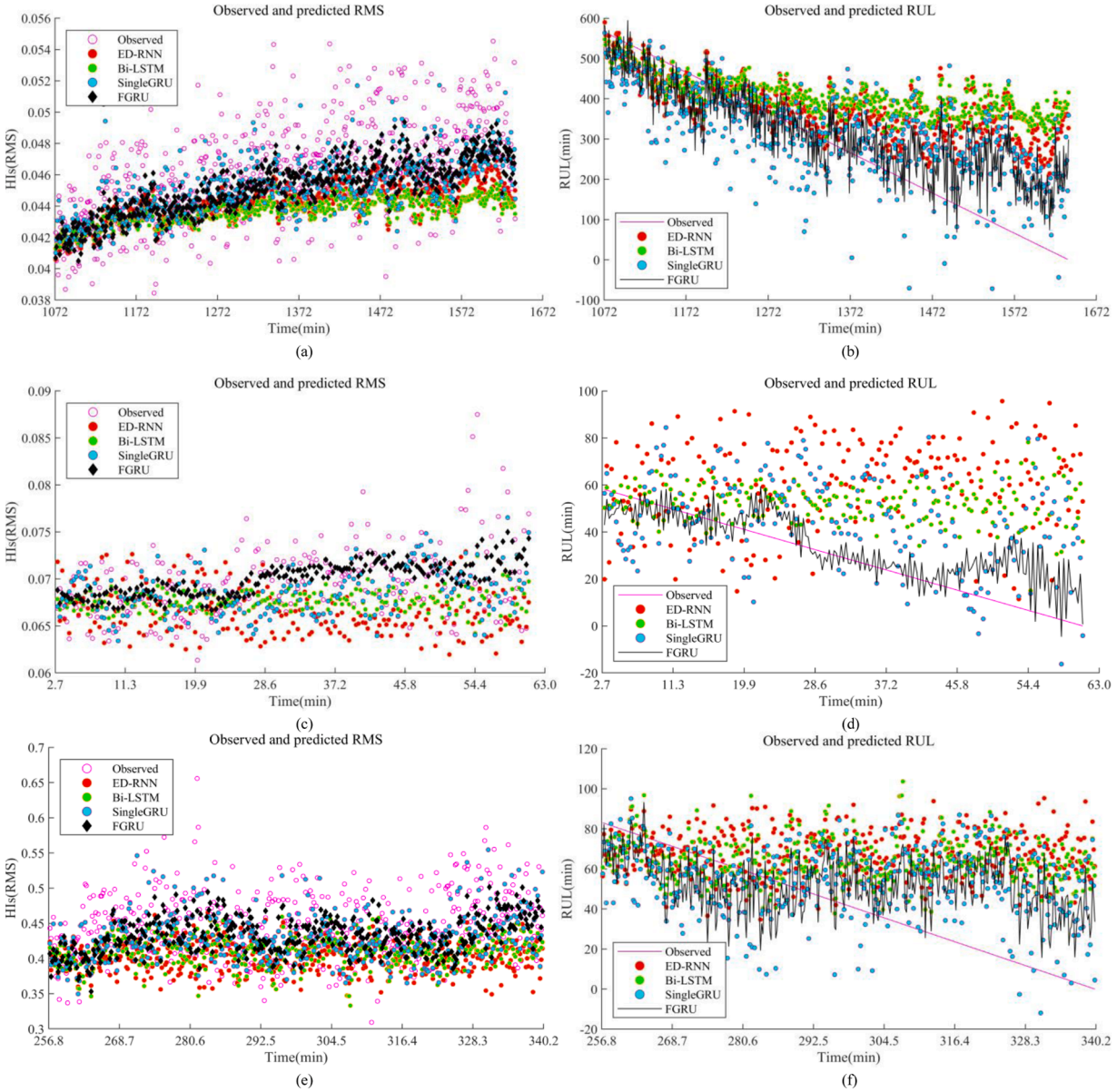


Fig. 13. RMS and RUL curves in the first case. (a,c,e) Normalized observed RMS value and predicted RMS value of R1, R2 and R3 bearings. (b, d, f) Real RUL value and predicted RUL value of R1, R2 and R3 bearings.

Table 4
FPTs and lifetimes of bearings in the second case.

Bearings	B1-1	B1-2	B1-3	B1-5	B3-5
Lifetime (min)	123	161	158	52	114
FPT (min)	75	32	58	35	6

ED-RNN and Bi-LSTM are also not ideal in the test, where there is a huge difference between the predicted and real value. That cannot provide effective guidance for subsequent production scheduling. The reasons are that: (1) ED-RNN has insufficient control over historical information from a long time ago, and only has the memory ability to the recent historical information. Especially in the later stage of bearing degradation, the defect of local concerning makes the prediction deviation no longer follow the actual degradation path. (2) Bi-LSTM insignificantly combines the input of backward LSTM on the backward LSTM in terms of the degradation prediction of rotating machinery without forward and backward correlation, unintentionally causing the true future degradation states of the bearing to incorrectly affect the predicted future degradation state.

Furthermore, the difference between the predicted and true RUL from the single GRU model is significantly improved compared with the previous two methods in the later life, but it is still not as good as FGRU. Nevertheless, it is worth noting that our method on this bearing lags behind the results of single GRU in terms of stability and dispersion, indicating that there is still room for improvement in the generalization of FGRU. The advantages of FGRU are mainly derived from the ability to judge failure modes in advance, and then to choose the appropriate life prediction model from multiple GRUs based on the predicted modes, instead of predicting RUL using a single model. Naturally, the multi-model strategy improves the generalization ability for the changing degradation paths, which is favorable especially in real industry prognosis applications. Furthermore, for the quantitative analysis of performance differences, the RMSE, MAE and testing times for the predicted RULs of five bearings in the proposed method and the compared methods are summarized as Table 5, where all the experiments are repeated for 10 trails to reduce randomness and only the predicted values after FPT point is calculated. It can be seen that the evaluated errors of the proposed method are the smallest and the prediction robustness is the most outstanding among the methods mentioned. And FGRU model generally achieves also a greater prediction performance on five bearings in the second case than the comparison methods. Specifically, for testing bearing B1-1, the prognosis performance with respect to the RMSE metric improved by approximately 45.54%, 31.51% and 6.18% compared to the ED-RNN-based, Bi-LSTM-based, and Single GRU-based models, respectively. For testing bearing B1-2, the forecast performance with respect to the MAE metric improved by about 21.5%, 30.90% and 9.17% compared to above comparison methods. For the rolling bearing B1-3, the prediction performance with respect to the RMSE metric improved by almost 45.70%, 51.02% and 9.04% compared to the three comparison methods. For the testing bearing B3-5, the performance also increases by 47.27%, 48.31%, and 11.77%, respectively. Unfortunately, the running time of the proposed method, e.g. offline testing time (resp. training time), almost is the longest one among that of the four methods, which is naturally affected by the sophisticated network structures of fault mode diagnosis model and multi GRUs. Note that the workshop scheduling is generally in the unit of day or month, therefore, when compared with the EDRNN-based model, Bi-LSTM-based model and Single GRU-based model, the additional running time of the proposed method is acceptable.

5.3. Experiment analysis for joint decision-making

In this section, for the joint optimization of parallel machine scheduling and predictive maintenance, we compare it with the following two mature meta-heuristic algorithms and two common production scheduling modes in actual production: (1) Cuckoo Search algorithm (CS) [49], (2) Adaptive weighted Particle Swarm Optimization (PSO) [50], (3) Joint decision-making with regular maintenance (RM), (4) Integrated decision-making no incorporating RUL of machines (NoRUL). Here, all methods are optimized for 900 iterations, and the solution dimension is 80 dimensions. The lower bounds of the generated

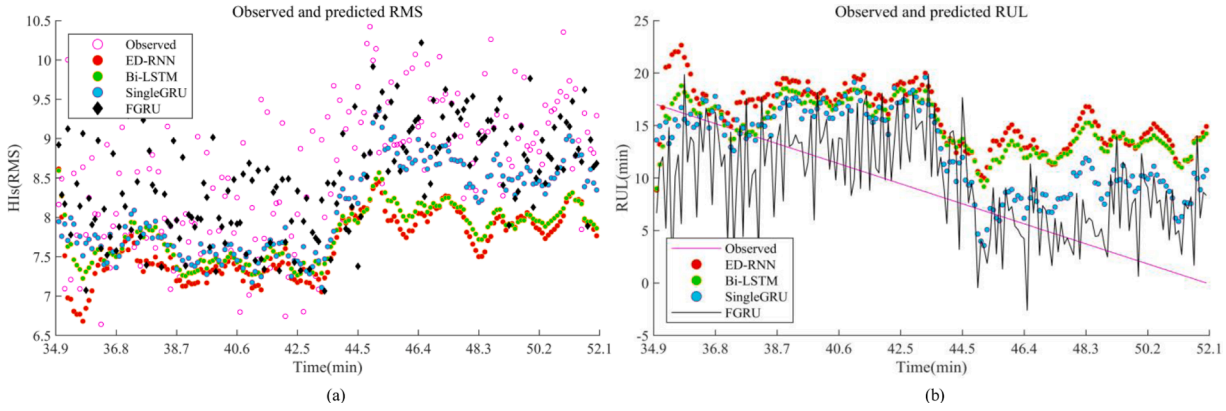


Fig. 14. RUL and RMS curves of B1-5. (a) RMS. (b) RUL.

Table 5
RMSE, MAE of predicted RULs and testing times from four life prediction models.

Bearings	B1-1			B1-2			B1-3			B1-5			B3-5		
Metrics	RMSE	MAE	Time (s)	RMSE	MAE	Time (s)	RMSE	MAE	Time (s)	RMSE	MAE	Time (s)	RMSE	MAE	Time (s)
ED-RNN	33.520	27.832	162.459	40.528	34.971	231.822	36.526	32.987	152.437	8.184	7.450	75.436	44.886	34.254	162.936
Bi-LSTM	26.653	22.858	198.221	45.371	39.730	230.069	40.518	31.735	246.184	7.471	6.533	80.628	44.112	34.946	212.704
Single GRU	19.457	17.077	157.197	33.798	30.217	226.065	21.817	19.921	163.760	5.302	4.491	66.983	24.941	20.471	152.952
FGRU	18.254	16.375	203.293	29.091	27.445	259.802	19.844	17.006	168.893	5.213	4.099	78.322	23.441	18.062	236.548

vectors of job release sequences, job processing sequences, job-machine combination and maintenance nodes are 1, 1, 0 and 0, and the corresponding upper bounds are 20, 20, 2π and 1, respectively. The shared penalty coefficient u for processing jobs beyond the RUL threshold is set as 50. The hyper-parameters of the comparison methods are set as follows: (1) the nest-finding probability and Levi random path coefficient of CS are set as 0.25 and 1.5, (2) the acceleration coefficient, speed upper limit, speed lower limit, inertia weight upper limit w_{max} and inertia weight lower limit w_{min} of the adaptive weight PSO method are set to 1.49445, 0.15 times the upper limit of the individual value, 0.15 times the lower limit of the individual value, 0.90 and 0.008, respectively, and the way of adaptive weighting is shown in the Eq. (38), (3) according to the current usage and maintenance habits of machines, the joint decision-making with regular maintenance preliminarily assumes that machines with less than 5 jobs are only maintained before processing the last job, while those with more than 4 jobs are maintained all before processing the last three jobs, and all maintenance durations of all machines are set to 40 min in the first/second cases and 25 min in the third case, which is not affected by the enlistment age of machines. Moreover, its optimization algorithm still uses the discrete TLBO, (4) unlike the RM, there are no restrictions on the maintenance nodes of machines in the NoRUL method, which are select independently by optimization algorithm. The same is that all maintenance durations of all machines are 40 min in the first/second cases and 25 min in the third case, and the optimization algorithm is the discrete TLBO.

$$\begin{cases} w(t) = (w_{\max} - w_{\min})\dfrac{1}{n}\displaystyle\sum_{i=1}^{n} S_i(t) + w_{\min} \\ S_i(t) = \begin{cases} 1, f^{(t)}(x_i) < f^{(t-1)}(x_i) \\ 0, f^{(t)}(x_i) \geq f^{(t-1)}(x_i) \end{cases} \end{cases} \tag{38}$$

where $w(t)$ is the adaptive weight in the t^{st} iteration, $w(1) = 0.80$, $S_i(t)$ is the success function of particle, when the t-round fitness function is better than the t-1-round fitness function, it is 1, otherwise it is 0.

After such optimization adjustment for the proposed method, we will get the best combination of processing/release sequences, matching relationships and the maintenance nodes. This combination can reliably ensure that all jobs are processed within the machine RUL threshold because of the large penalty ratio coefficient u, and then we can get a satisfactory makespan. Table 6 shows the processing details of simulation data in the first group formed by the combination of optimized outputs of the proposed method. It can be seen that, without considering the job waiting, the items on the 1st machine are the job 17, job 5, job 8, the first maintenance, job 9, job 14 and job 10 in turn. The items on the 2st machine are the job 2, the first maintenance and job 12 in turn. The items on the 3st machine are the job 13, the first maintenance, job 20, the second maintenance, job 3, job 18, job 6, the third maintenance, job 19 and job 15 in turn. Likewise, the items on the 4st machine are the job 7, the first maintenance, job 4, job 1, the second maintenance, job 11, the third maintenance and job 16 in turn. Therefore, the feasibility of the proposed method is proved. What needs to be clear is that, in the proposed method, the maintenance duration is jointly determined by the rated RUL of machines (original RUL) and the current enlistment age, where the rated RUL is determined independently by the manufacturers or the business owners.

In order to further illustrate the superiority of the proposed method, we conducted four comparison experiments on a computer with the same simulation data. Except for the differences in parameters and optimization structure described in the previous section, the model structures of the five methods are always consistent in the writing, whose fitness function changes are shown in Fig. 15. From Fig. 15a, we can obviously see that the PSO algorithm with adaptive weights can first reach the minimum fitness function value (the 20th iteration), but its result is not convergent, and even the fitness value of 4000 is obtained in about the 200th iteration. According to the characteristics of our

Table 6
Job assignment and sequence.

Machines	Jobs																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1					II☆●			III☆●	IV★●	VI☆●				V☆●			I☆●			
2		I☆●										II★●								
3			III★●			V☆●							I☆●		VII☆●			IV☆●	VI★●	II★●
4	III☆●			II★●			I☆●				IV★●					V★●				

Roman numerals: processing sequence of jobs. ★(☆): Machine (does not) needs maintenance before machining the job. ● The job is processed by the machine.

Algorithm #

FGRU model and joint decision-making model.

Input: N run-to-failure degradation datasets X^n of components failing in multiple fault modes, $n \in \{1, 2, \ldots, N\}$, for training. Run-to-failure datasets X^t for testing. Hyper-parameters of CNN (e.g. epoch1). Hyper-parameters of GRU (e.g. epoch2). Length Sl and step size Ws of sliding window for RMS. Length g and step size p of sliding window for LSM. Slope threshold ϑ. Hyper-parameters of IR, Initial population of discrete TLBO, Parameters for machines and jobs (in minutes).
Output: RUL of X^t, Job release sequence, Production sequence, Maintenance nodes, Job-machine combination.
1. Data processing for FPT: Shape X^n and X^t into cellular arrays with the same length.
2. HI extraction: Get all RMSs series of X^n and X^t.
3. Iterative isotonic regression: Monotonize RMSs of X^n and X^t.
4. Fitting increment: Get slope of each monotonous HI segment of RMSs by LSM algorithm.
5. Slope comparision: Compare the slope of each HI segment with the given constant threshold in chronological order.
6. FPT detection: Get the first prediction time when the fitted slope exceeds the given threshold for the first time.
7. Initialization: Construct 1D CNN and multi-GRU networks to be trained.
8. Data processing for 1D CNN: Label X^n and X^t after FPT and normalize them.
9. % 1D CNN Training
10. for $i = 1$:epoch1 do
11. Input random small-batch samples from processed X^n into 1D CNN;
12. Updating parameters of 1D CNN by minimizing (24) and AdaBelief.
13. end for
14. Input all samples from processed X^t into trained1D CNN.
15. Get diagnosed fault modes of X^t.
16. Data processing for GRU: Normalize HIs of training and testing samples after the FPT point.
17. %% Multi-GRUs Training
18. for $j = 1$:epoch2 do
19. Input normalized HIs of training samples after FPT into multiple GRUs along with the known fault modes;
20. Update all parameters of multi-GRUs by minimizing (39) and AdaBelief.
21. end for
22. % RUL Prediction
23. Input normalized HIs of testing samples after FPT into multi-GRUs along with the diagnosed fault modes.
24. Select the corresponding GRU network and output the predicted RUL value by (38).
25. %% Joint decision-making
26. Establish initial population with job production/release sequences, maintenance nodes and job-machine combination by a series of discrete codes.
27. While the maximum iteration is not met do
28. Generate X_{ijk}, Y_{jk}, and V_{ig} based on this round of population;
29. Each maintenance duration and processing duration beyond RUL threshold of each machine in all learners, release time of all jobs, the compl-etion time of each position are calculated;
30. Calculate the total objective function (3);
31. Enter the teaching phase and the learning phase in turn by the formulae (15), (16), then update all continuous variables, encrypt them to get new job production/release sequences, maintenance nodes and job-machine combination;
32. end.

objective function, there are two main reasons for divergence: (1) Although the proposed model converts the discrete permutation and combination numbers into the continuous values of being able to count and multiply in advance, the mapping relationship between the continuous inputs and fitness function is still not a continuous function, and local maxima and minima widely exist in the mapping relation. This is the opposite of PSO focusing on continuity optimization problems. (2) The adaptive weight of PSO in this experiment is very sensitive to the proportion change of better individuals in each round of iteration, which is also called the individual success ratio. It has more randomness in the special knapsack problem of permutation and combination with scheduling and maintenance constraints, and cannot adaptively weigh the speeds of the old and new individuals as presented in the original contribution. Next, in Fig. 15b, we can see that the convergence of CS algorithm has improved a lot, but it fell into a local minimum at the 473th iteration and stuck to it for a long time, which climbed out of the abyss at the 746th iteration when its fitness function decreased to 502, but it is still slightly inferior to discrete TLBO algorithm at that time. CS algorithm looks for the bird nest through Levi flight, which is a random walk composed of small step flight with short-distance and occasional large step flight with long-distance. Therefore, the nest seeking path of cuckoo is easy to jump between different search areas, resulting in the poor local fine search ability. The early optimization effect of integrated decision under NoRUL mode is analogous to that of CS algorithm. The difference is that it has a faster convergence rate where it has already converged at the 162nd iteration. However, for the more important fitness function for practical applications, the NoRUL mode is even more unsatisfactory, where it only obtained a fitness value of 570. The NoRUL model assumed that the maintenance duration is constant, which is not in line with ideal maintenance where the maintenance duration of machine with a long running time is generally longer, and the maintenance duration of machine with a shorter rated RUL is generally shorter, where the NoRUL model is indeed an assumption in the previous literature. Consequently, the result of this greater fitness value shows that it is a good idea to introduce the enlistment age and initial rated RUL to evaluate the maintenance duration. The optimization performance of RM mode is the worst among the five methods, and it converges at the 408th iteration when the maximum converged fitness value of 623 among the five methods is obtained. The regular maintenance mode neither considers the maintenance duration flexibility, but also believes that the maintenance operation is only carried out at a fixed time period, which is also the common operation and maintenance process in traditional industrial enterprises, being far behind the requirements of emerging smart factories and unmanned workshops.

For the proposed joint decision-making method, it obtains the best optimization performance both in terms of the convergence speed (the 190th iteration) and the fitness value 479. In addition, according to the measured data, the penalty items of processing jobs beyond RUL threshold after the fitness convergence are all zeros, that is, each machines will process jobs within its RUL threshold, so the fitness after convergence is only the makespan, which is also the smallest one among that of five methods, further verifying the effectiveness and superiority

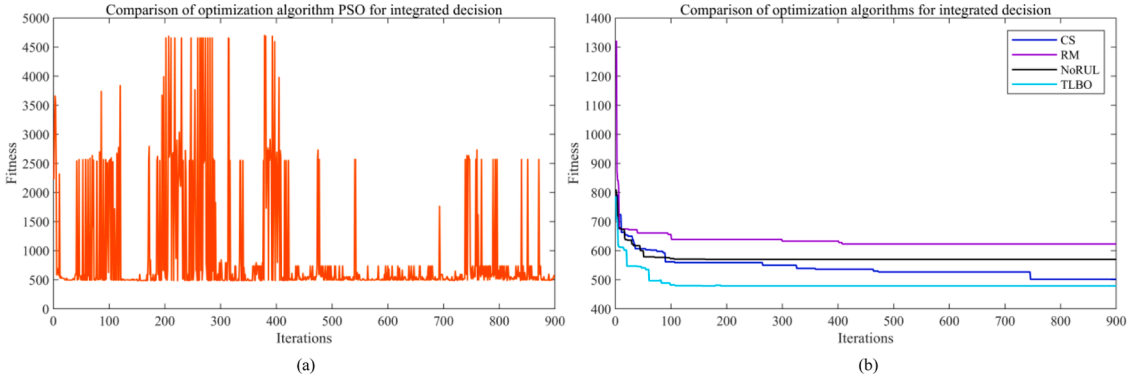


Fig. 15. Comparison of optimization algorithms for joint decision-making in the first simulation case. (a) PSO. (b) CS, RM, NoRUL and discrete TLBO.

of the discrete TLBO-optimized and RUL-integrated workshop scheduling planning considering the predictive maintenance. In conclusion, compared with the other four approaches, either separating the RUL prediction of machines and maintenance operations or neglecting the uncertainty of maintenance nodes will lead to the deviation on decision. Furthermore, the Gantt charts of the near-optimal joint production and preventive maintenance planning using four methods and the job release time using discrete TLBO algorithm are given in Fig. 16. It can be seen from the solutions that each job can be completed as soon as possible within the RUL threshold to ensure the finished product mass. For all that, all the solutions obtained by four methods are obviously distinguished from each other in the makespan, where, the CS algorithm gained the longest makespan 1593 min, followed by 1558 min gained by the RM, 1480 min gained by the NoRUL and the shortest makespan 1422 min jointly gained by the discrete TLBO algorithm and the said MIP model. The proposed method is about 10.7 percentage points better than the model with the worst makespan (1593 min). The cumulative maintenance durations of the integrated schemes obtained by all comparison methods are 238 min from our method, 316 min from the CS algorithm, 320 min from the NoRUL mode and 480 min from the RM mode in ascending order, respectively. Thus, the maintenance occupation rate of the integrated scheme from our method in the whole order cycle is 16.7%.

Another obvious point is that under the support of the job release sequences arranged by their respective methods, the cumulative waiting time for jobs in the proposed joint decision-making approach is the shortest (8 min) among all methods, followed by 225 min gained by the NoRUL, 258 min gained by the RM and 304 min gained by the CS algorithm, which are all caused by ignoring or generally assuming various constraints in the actual operation environment. At length, the CS algorithm has not fully learned and understood the matching relationship between job release sequence and job processing sequence, where some jobs with late release time are arranged in the front machining positions, resulting in unnecessary machine idle for a long time. In RM mode, by default, the machines will be maintained for a fixed or expected period before each rear machining position. This traditional practice is not only a seemingly perfect isolation between the maintenance duration and the accumulated enlistment age of the machines, but also a waste of maintenance resources and brings about frequent equipment occupation, which is easily prone to over maintenance. And the maintenance initiation time in the NoRUL maintenance mode is automatically selected before the RUL threshold by the optimization algorithm, which no longer follows the fixed or expected maintenance interval, or the state-based maintenance strategy. Naturally, its makespan has been slightly reduced, yet it still does not integrate the maintenance duration flexibility with the current state of the machines, as a result, which is not the best one. Again, the “minute” unit in the Fig 16 and the following Fig 17 is introduced only to cater to the premature failure of bearings in the accelerated degradation test, and the actual scale is still limited to weeks or months.

Meanwhile, to examine the impact of different setup parameters on the joint decision-making of parallel machine scheduling and preventive maintenance, additional comparative tests using four said methods are conducted on two parameters, i.e., the comprehensive proportion coefficient ($\lambda = 40$) for the flexible maintenance duration and the fixed maintenance duration (25 min). The near-optimal results of the joint decision-making problem are presented in Fig. 17, where, the RM mode gained the longest makespan 829 min, followed by the makespan 821 min gained by the CS algorithm, the makespan 743 min gained by the NoRUL and the shortest makespan 717 min jointly gained by the discrete TLBO algorithm and the said MIP model. The proposed method is about 13.5 percentage points better than the model with the worst makespan (829 min). And the cumulative waiting time for jobs in the proposed joint decision-making approach is the shortest (55 min) among that of all methods, followed by 141 min gained by the NoRUL, 256 min gained by the RM and 410 min gained by the CS algorithm, which means that the total machine vacancy rate of the resulting scheme in the proposed method is the lowest among that of four methods, reaching 7.7%. It is preliminarily concluded that the expected makespan improvement ratio of production and the vacancy rate of machines become greater, as the fixed maintenance duration for the RM and NoRUL modes and the comprehensive proportion coefficient λ for flexible maintenance duration decrease. For the maintenance occupancy, the cumulative maintenance durations of the integrated schemes obtained by all comparison methods are 126 min from our method, 137 min from the CS algorithm, 150 min from the NoRUL mode and 300 min from the RM mode in ascending order, respectively. Thus, the best maintenance occupation rate of the integrated scheme from our method is 17.6%, that is, the

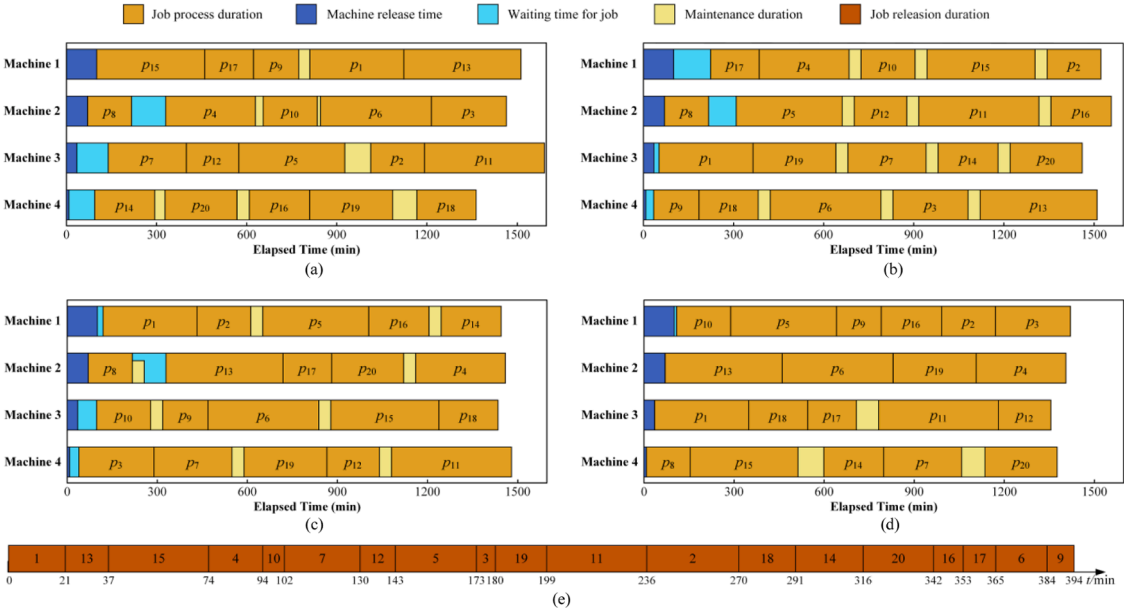


Fig. 16. The Gantt charts of the near-optimal scheduling and maintenance for the second case. (a) Gantt chart originating from the CS. (b) Gantt chart originating from the RM. (c) Gantt chart originating from the NoRUL. (d) Gantt chart originating from the discrete TLBO. (e) All job release times originating from the discrete TLBO.

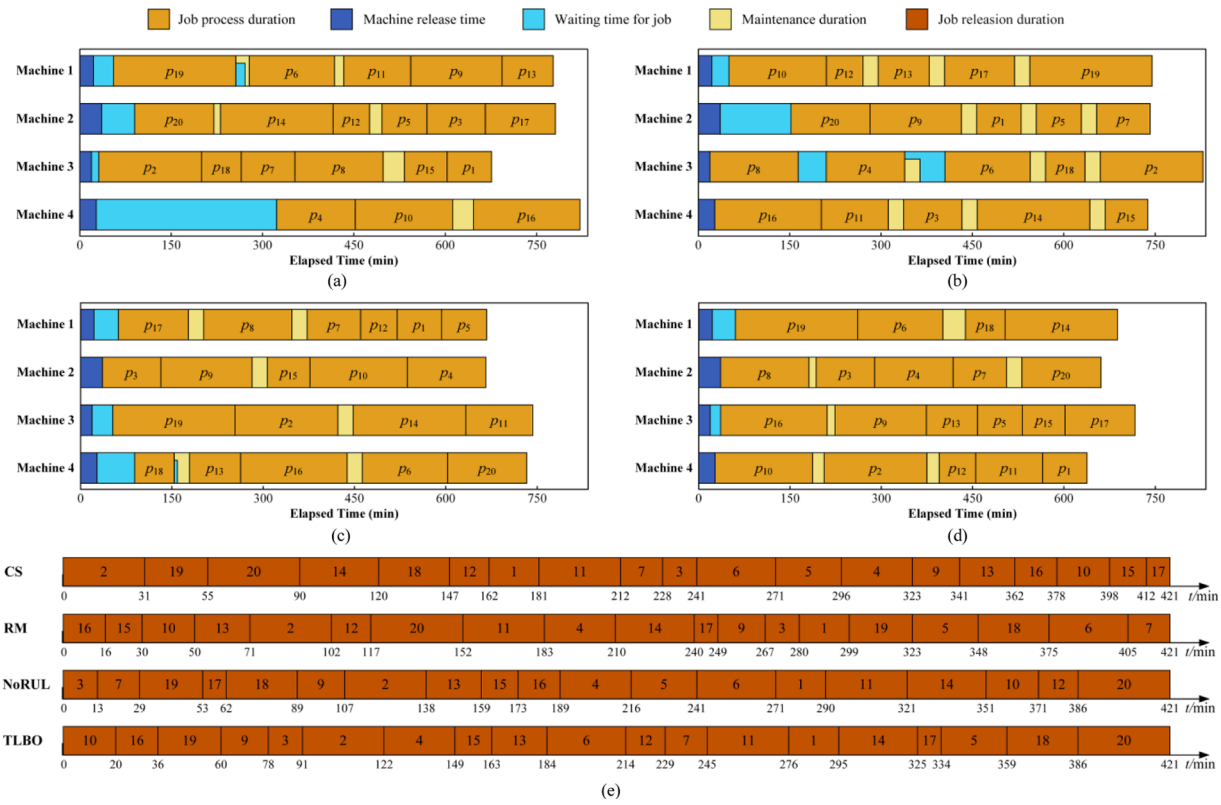


Fig. 17. The Gantt charts of the near-optimal scheduling and predictive maintenance schemes for the third simulation case. (a) The Gantt chart originating from the CS algorithm. (b) The Gantt chart originating from the RM mode. (c) The Gantt chart originating from the NoRUL mode. (d) The Gantt chart originating from the discrete TLBO algorithm. (e) All job release times originating from the CS, RM, NoRUL and discrete TLBO in turn.

maintenance occupancy ratio increases fractionally with the decreases of fixed maintenance duration in the RM/NoRUL models and the coefficient λ for flexible maintenance duration. From a broad angle, the proposed method greatly reduces the cumulative maintenance duration and maintenance times, by maintaining as early as possible and reducing the insertion of unnecessary maintenance activities, which can not only streamline the embedded costs of preventive maintenance and machine startup-shutdown, but also increase the machine in-service duration to a certain extent. Note that the resulting scheme from our method still is not optimal, for example, no maintenance is actually required before the fifth machining position of the second machine in Fig 17d, and inserting a maintenance activity there will increase the overall production cost in vain.

6. Conclusions and future works

In this paper, by taking account of (1) the online uncertainty of machine reliability caused by the dynamic service environment, (2) the random release times of jobs and machines, and (3) the relationship between the flexible maintenance duration and the enlistment age of machines, a joint decision-making MIP model of parallel machine scheduling and predictive maintenance is established and evaluated with the aim of minimizing the total makespan, where the processing penalty beyond RUL threshold of machines is added to the objective function of MIP model as an additional item to reduce the computational complexity. And the resulting optimization problem is resolved by a discrete TLBO algorithm. Especially, for the online reliability uncertainty, a data-driven fault mode predictor and a variety of RUL prediction models (GRUs) for different fault modes are proposed in this paper, where the RUL is regarded as a reliability index. In this case, when predicting RUL of machines, the proposed RUL prediction method can select the corresponding GRU module according to the fault mode diagnosed by the fault mode predictor. The improvement of prediction uncertainty and accuracy inspired by this auxiliary strategy has also been verified in subsequent experiments. Additionally, to reduce invalid prediction and computational complexity, the data preprocessing method of iterative isotonic regression based on gradient threshold is used to uniquely determine the FPT. As demonstrated in our simulation cases, the makespan jointly obtained by the FGRU method, MIP model and discrete TLBO algorithm is satisfactory, and the simplified method which ignores the maintenance duration flexibility, the online reliability uncertainty or job-machine release time can yield a large deviation in evaluating the order makespan. Several challenges are to be addressed in our future works. Firstly, in the present study the RUL of machines were assumed to be that of the spindle bearing, while the most intuitive representative for RUL of machines should be the installed tools with more serious abrasion. The RUL prediction for multi-vulnerable systems can be further considered in the specific problem. Secondly, we only evaluated the maintenance duration by taking account of the enlistment age and the rated RUL of machines. Some other statistics, e.g. product function attributes, spare parts inventory, third-party maintenance service scheduling, et al., will be derived in the future work. Thirdly, in reality, the processing duration of jobs are essentially random variables in various distributed forms. Future research can focus on this as an entry point to increase the commercialization of the proposed integrated model. Fourth, benefit from the development of wireless sensing and integrated simulation technology, the emerging digital twin technology has become a hot research field, which can build a communication bridge between field engineers and indirect resource providers in the planning of production scheduling and predictive maintenance. In the future, the project funds will be inclined to this intelligent agent that can perceive, analyze, understand the environment and respond to its current state. Lastly, other uncertainties, e.g. demand uncertainty and temporary order insertion, will be jointly studied in our future work, and

some alternative optimization methods will also be tailored to facilitate the joint optimization under uncertainty.

CRediT authorship contribution statement

Xinxin He: Methodology, Writing – original draft. **Zhijian Wang:** Conceptualization, Funding acquisition. **Yanfeng Li:** Writing – review & editing. **Svetlana Khazhina:** Methodology, Writing – review & editing. **Wenhua Du:** Project administration, Writing – review & editing. **Junyuan Wang:** Visualization, Data curation. **Wenzhao Wang:** Validation, Supervision, Writing – original draft.

Declaration of Competing Interest

There is no conflict of interest between the authors and all the authors mutually agree to submit the manuscript in the Journal.

Acknowledgements

The authors would like to acknowledge the Shanxi Provincial Natural Science Foundations of China (Nos. 201801D221339, 201801D221237, 201901D111239), Openresearch fund for Key Discipline Laboratory of Super-Pressure Hurting Technology (No. DXMBJJ2019-01) and National Natural Science Foundation of China (No. 51905496) for their support.

References

[1] McNaughton R. Scheduling with deadlines and loss functions. Manag Sci 1959;6:1–12.
[2] Mönch L, Shen L. Parallel machine scheduling with the total weighted delivery time performance measure in distributed manufacturing. Comput Oper Res 2021;127:105126.
[3] Basiri M, Alinezhad E, Tavakkoli-Moghaddam R, Shahsavari-Poure N. A hybrid intelligent algorithm for a fuzzy multi-objective job shop scheduling problem with reentrant workflows and parallel machines. J Intell Fuzzy Syst 2020;39:7769–85.
[4] Li K, Chen J, Fu H, Jia Z, Wu J. Parallel machine scheduling with position-based deterioration and learning effects in an uncertain manufacturing system. Comput Ind Eng 2020;149:106858.
[5] Xu Y, Zhi R, Zheng F, Liu M. Parallel machine scheduling with due date-to-deadline window, order sharing and time value of money. Asia Pac J Oper Res 2022. https://doi.org/10.1142/S021759592150024X.
[6] Xiao Y, Zheng Y, Yu Y, Zhang L, Lin X, Li B. A branch and bound algorithm for a parallel machine scheduling problem in green manufacturing industry considering time cost and power consumption. J Clean Prod 2021;320:128867.
[7] Zhang Z, Lin C, Yang L, Xue S, Liu J. Block-painting-operation-oriented hybrid flow shop scheduling. J Shanghai Jiaotong Univ 2014;48:382–7.
[8] Gao K, Suganthan PN, Pan Q, Tasgetiren MF, Sadollah A. Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion. Knowl Based Syst 2016;109:1–16.
[9] Abdelmaguid TF, Shalaby MA, Awwad MA. A tabu search approach for proportionate multiprocessor open shop scheduling. Comput Optim Appl 2014;58:187–203.
[10] Long J, Sun Z, Pardalos PM, Bai Y, Zhang S, Li C. A robust dynamic scheduling approach based on release time series forecasting for the steelmaking-continuous casting production. Appl Soft Comput J 2020;92:106271.
[11] Cevikcan E, Durmusoglu MB. An integrated job release and scheduling approach on parallel machines: an application in electric wire-harness industry. Comput Indus Eng 2014;76:318–32.
[12] Zhou S, Jin M, Du N. Energy-efficient scheduling of a single batch processing machine with dynamic job arrival times. Energy 2020;209:118420.
[13] Mario CV, Maya J, Jairo RM. A beam search heuristic for scheduling a single machine with release dates and sequence dependent setup times to minimize the makespan. Comput Oper Res 2016;73:132–40.
[14] Dong Y, Jin Y, Li Z, Ji H, Liu J. Scheduling optimization of a wheel hub production line based on flexible scheduling. Int J Ind Eng Theory Appl Prac 2020;27:694–711.
[15] Defersha FM, Chen M. Jobshop lot streaming with routing flexibility, sequence-dependent setups, machine release dates and lag time. Int J Prod Res 2012;50:2331–52.
[16] Adiri I, Bruno J, Frostig E, Kan A. Single machine flow-time scheduling with a single breakdown. Acta Inform 1989;26:679–96.
[17] Wang D, Liu F, Jin Y. A proactive scheduling approach to steel rolling process with stochastic machine breakdown. Nat Comput 2019;18:679–94.
[18] Abbas A, Heba A, Natalija L. Proposed procedure for optimal maintenance scheduling under emergent failures. J Civil Eng Manag 2020;26:396–409.
[19] In-Beom P, Jaeseok H, Joongkyun K, Jonghun P. A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities. IEEE Trans Autom Sci Eng 2020;17:1420–31.
[20] Kacem I. 2-approximation algorithm for the weighted flowtime minimization on a single machine with a fixed non-availability interval. In: Proceedings of the 37th international conference on computers & industrial engineering. 2; 2007. p. 1067–9.
[21] Yu X, Zhang Y, Steiner G. Single-machine scheduling with periodic maintenance to minimize makespan revisited. J Schedul 2022;17(2104):263–70.
[22] Zou J, Yuan J. Single-machine scheduling with maintenance activities and rejection. Discret Optim 2020;38:100609.
[23] Hsieh Y, You P. An immune-based algorithm for the multi-maintenance with sequential operations. J Intell Fuzzy Syst 2021;40:7701–10.
[24] Marsili F, Bödefeld J, Daduna H, Croce P. Scheduling of waterways maintenance interventions applying queueing theory. In: Proceedings of the life-cycle analysis and assessment in civil engineering; 2019. p. 999–1008.
[25] Pang J, Tsai Y, Chou F. Feature-extraction-based iterated algorithms to solve the unrelated parallel machine problem with periodic maintenance activities. IEEE Access 2021;9:139089–108.
[26] Cui W, Lu Z, Pan E. Production scheduling and preventive maintenance integration based on multi-objective optimization. Comput Int Manuf Syst 2014;20:1398–404.
[27] Tao X, Xia T, Xi L. Health-index-based joint optimization of preventive maintenance and multi-attribute production scheduling. J Shanghai Jiaotong Univ 2014;48:1170–4.
[28] Sousa PFS, Afonso SMB, Willmersdorf RB. Reliability-based preventive maintenance planning for corroded pipelines using a RBF surrogate model. J Braz Soc Mech Sci Eng 2022. https://doi.org/10.1007/s40430-021-03247-3.
[29] Ke S, Toon DP, Xu G, Martens L, Joseph W. Genetic optimization of energy-and failure-aware continuous production scheduling in pasta manufacturing. Sensors 2019;19:202–9.
[30] Li T, Si X, Pei H, Sun L. Data-model interactive prognosis for multi-sensor monitored stochastic degrading devices. Mech Syst Signal Process 2022;167:108526.
[31] Si X, Li T, Zhang J, Lei Y. Nonlinear degradation modeling and prognostics: a box-cox transformation perspective. Reliab Eng Syst Saf 2022;217:108120.
[32] Li X, Krivtsov V, Arora K. Attention-based deep survival model for time series data. Reliab Eng Syst Saf 2022;217:108033.
[33] Li T, Zhao Z, Sun C, Yan R, Chen X. Hierarchical attention graph convolutional network to fuse multi-sensor signals for remaining useful life prediction. Reliab Eng Syst Saf 2021;215:107878.
[34] Hu Q, Xie M, Ng S, Levitin G. Robust recurrent neural network modeling for software fault detection and correction prediction. Reliab Eng Syst Saf 2007;92:332–40.
[35] Liao G, Yin H, Chen M, Lin Z. Remaining useful life prediction for multi-phase deteriorating process based on Wiener process. Reliab Eng Syst Saf 2021;207:107361.
[36] Xu X, Tang S, Yu C, Xie J, Han X, Ouyang M. Remaining useful life prediction of lithium-ion batteries based on Wiener process under time-varying temperature condition. Reliab Eng Syst Saf 2021;214:107675.
[37] Ouyang H, Gao L, Kong X, Zhou D, Li S. Teaching-learning based optimization with global crossover for global optimization problems. Appl Math Comput 2015;265:533–56.
[38] Zhang E, Wu Y, Chen Q. A practical approach for solving multi-objective reliability redundancy allocation problems using extended bare-bones particle swarm optimization. Reliab Eng Syst Saf 2014;127:65–76.
[39] Zhang X, Liu Z, Miao Q, Wang L. Bearing fault diagnosis using a whale optimization algorithm-optimized orthogonal matching pursuit with a combined time-frequency atom dictionary. Mech Syst Signal Process 2018;107:29–42.
[40] Zhang Y, Cui G, Wu J, Pan W, He Q. A novel multi-scale cooperative mutation fruit fly optimization algorithm. Knowl Based Syst 2016;114:24–35.
[41] Rao RV, Savsani VJ, Vakharia DP. Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems. Inform Sci 2012;183:1–15.
[42] Črepinšek M, Liu S, Mernik L. A note on teaching-learning-based optimization algorithm. Inform Sci 2012;212:79–93.
[43] Zhuang J, Tang T, Ding Y, Tatikonda S, Dvornek N, Papademetris X, Duncan JS. AdaBelief optimizer: adapting stepsizes by the belief in observed gradients. NeurIPS; 2020.
[44] Blake MP, Mitchel WS. Vibration and acoustic measurement. New York: Spartan Books; 1972.
[45] Wang H, Liao H, Ma X, Bao R. Remaining useful life prediction and optimal maintenance time determination for a single unit using isotonic regression and gamma process model. Reliab Eng Syst Saf 2021;210:107504.
[46] Lei Y, Han T, Wang B, Li N, Yan T, Yang J. XJTU-SY rolling element bearing accelerated life test datasets: a tutorial. J Mech Eng 2019;55:1–6.
[47] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdana, Fethi Bougares, Holger Schwenk, Yoshua Bengi. Learning phrase representations using RNN encoderdecoder for statistical machine translation, The 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), doi: 10.3115/v1/D14-1179.

[48] Yu Y, Hu C, Si X, Zheng J, Zhang J. Averaged Bi-LSTM networks for RUL prognostics with non-life-cycle labeled dataset. Neurocomputing 2020;402:134–47.
[49] Gao S, Zhang S, Zhang Y, Gao Y. Operational reliability evaluation and prediction of rolling bearing based on isometric mapping and NoCuSa-LSSVM. Reliab Eng Syst Saf 2020;201:106968.
[50] Zand AD, Damghani KK. Design of SCADA water resource management control center by a bi-objective redundancy allocation problem and particle swarm optimization. Reliab Eng Syst Saf 2015;133:00311–21.