Quality Engineering

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/lqen20

Deep reinforcement learning for maintenance optimization of multi-component production systems considering quality and production plan

Ming Chen, Yu Kang, Kun Li, Pengfei Li & Yun-Bo Zhao

QUALITY ENGINEERING
https://doi.org/10.1080/08982112.2024.2373362

Deep reinforcement learning for maintenance optimization of multi-component production systems considering quality and production plan

Ming Chen[a], Yu Kang[a,b,c], Kun Li[b], Pengfei Li[a,c], and Yun-Bo Zhao[a,b,c]

[a]Department of Automation, University of Science and Technology of China, Hefei, China; [b]Institute of Advanced Technology, University of Science and Technology of China, Hefei, China; [c]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China

ABSTRACT
In this article, the maintenance optimization of multi-component production systems is investigated by considering quality and production plan. On the one hand, the downtime determined by the production plan provides opportunities for reducing maintenance costs; on the other hand, the deterioration of product quality induced by poor health state leads to extra loss. The coupled relations between production plan, quality, and maintenance, as well as the dependence between multiple components, pose challenges for maintenance optimization. To overcome these challenges, a novel decision model and a deep reinforcement learning-based solving method are proposed. Specifically, in addition to the degradation states of all components, the remaining time of the current batch related to the production plan is also treated as the system state, and the quality loss related to the degradation states is added to the reward function. The deep Q-network algorithm is employed, solving the maintenance optimization problem that considers quality and production plan. The effectiveness of the proposed method is validated by a numerical experiment.

KEYWORDS
Deep reinforcement learning; maintenance optimization; multi-component; production plan; quality

1. Introduction

Multi-component production systems are common in modern industrial scenarios, particularly in assembly line production systems where each component represents production equipment responsible for distinct processes. Taking the example of a laptop motherboard production line, the printing machine is responsible for solder paste printing, the solder paste inspection equipment is responsible for quality inspection of the printing, and the placement machine is responsible for component mounting. In such systems, the unexpected failure of equipment often seriously affects production efficiency and quality. The main reason is that locating and maintaining a malfunction may lead to a long period of unplanned downtime and equipment on the brink of failure can easily produce defective products. For production systems, the maintenance cost caused by component failures often accounts for a large proportion of the production cost (Thomas and Thomas 2018), therefore effective maintenance methods are necessary.

Maintenance strategies can be roughly divided into two main types: time-based maintenance (TBM) and condition-based maintenance (CBM). In the case of TBM, maintenance activities occur according to predetermined schedules, whereas CBM relies on monitoring the real-time health status of the system. The development of sensing technology enables to obtain more comprehensive information related to the health state of equipment, which provides data support for CBM research. CBM selects the appropriate maintenance strategy based on the health state, achieving the optimization of production efficiency and the improvement of system reliability. CBM has achieved good results in practical applications such as aircraft engines (Liu et al. 2014), wind turbines (Ghamlouch, Fouladirad, and Grall 2019), and oil and gas pipelines (Parvizsedghy et al. 2015).

In actual production processes, formulating maintenance strategies solely based on equipment health state may not necessarily achieve optimal efficiency. A central factor lies in the intricate connection between the productivity of the manufacturing system and both quality control and production plan, and these two factors are also influenced by maintenance strategies (Farahani and Tohidi 2021). Details are as follows:

1. Quality control refers to keeping the product quality within the range that meets the requirements of

CONTACT Pengfei Li puffylee@ustc.edu.cn Department of Automation, University of Science and Technology of China, 230026 Hefei, China.

manufacturer. One of the main factors affecting product quality is the health state of equipment (Rahim and Ben-Daya 1998; Lee and Rosenblatt 1987). When the equipment is in poor health, the risk of product quality deterioration greatly increases, resulting in an increase in the quantity of defective products. There are two main methods for handling defective products: repair and disposal. However, both methods result in quality loss, namely repair cost and production waste. To reduce loss, an effective means is to keep the equipment in a good health state through maintenance. However, frequent maintenance increases maintenance cost, so maintenance decisions need to balance maintenance cost with quality loss.

2. Production plan determines future production tasks and scheduling. When equipment runs normally according to the production plan, any maintenance action that stops the equipment from working will result in downtime cost (Budai, Dekker, and Nicolai 2006). On the contrary, when maintain during the downtime planned by production department (such as cleaning, shift change, batch or tool change), downtime cost is not calculated. (Levrat, Thomas, and Iung 2008; Xia et al. 2015). In order to reduce cost, maintenance actions can be scheduled as much as possible during the planned downtime periods in the production plan.

In order to improve overall production efficiency, many scholars are considering production plan, quality control, and maintenance strategies simultaneously. For example, (Levrat, Thomas, and Iung 2008) proposed a method of integrating maintenance into production planning, combining product performance and component reliability to select the most suitable maintenance plan during planned downtime. In the work of Hu, Jiang, and Liao (2017), the focus was on examining maintenance issues in systems comprising multiple machines, particularly addressing the challenge of establishing an efficient dynamic preventive maintenance plan for machine clusters at critical batch transition stages. However, they assumed that maintenance decisions were made only at batch transfer points, making their methods not applicable to cases where batch duration is too long. (Nourelfath, Nahas, and Ben-Daya 2016) constructed a framework that unifies production, maintenance, and quality control aspects within a manufacturing setup, where system degradation occurs randomly across various operational states. (Beheshti Fakher, Nourelfath, and Gendreau 2017) proposed a joint approach that coordinates the decisions of production, maintenance, and process inspection. However, they did not consider downtime cost caused by maintenance or maintenance opportunities during planned downtime.

Additionally, formulating maintenance strategies confronts complexities due to the dependencies among multiple components. Fundamentally, these dependencies can be grouped into three principal categories: random dependence, structural dependence, and economic dependence (Nicolai and Dekker 2006; Keizer, Flapper, and Teunter 2017). Random dependence occurs when the condition of one component influences the deterioration progression of others (Nicolai and Dekker 2006; Bian and Gebraeel 2014). Structural dependence applies to components that form a connected set in terms of structure, and repairing or replacing one component requires disassembling other components (Dao and Zuo 2017). For production line systems, each component is usually physically independent of each other and there is generally no random dependence and structural dependence. This article mainly focuses on economic dependence. Economic dependence means that maintaining multiple components simultaneously can save costs compared to maintaining them separately. As economic dependence directly affects maintenance decisions through maintenance cost, researchers have extensively studied this issue. For instance, (Tian and Liao 2011) developed CBM of multi-component systems, where economic dependency exists among different components subject to condition monitoring. In the study of Zhou and Yin (2019), the economic dependence between wind turbines and components were taken into account. (Do and Bérenguer 2020) considered maintenance costs, economic dependence between components, and cost-effectiveness of maintenance operations. (Zhang and Si 2020) employed deep reinforcement learning, considering economic dependence and opportunistic maintenance. Considering economic dependence, maintenance decisions must be made from an overall perspective rather than optimizing individual components.

The current CBM methods for maintenance optimization in multi-component systems are mainly divided into two categories: the method based on dual thresholds and the method based on deep reinforcement learning (DRL). The method based on dual thresholds not only sets a health threshold for each component to determine whether there is a need for maintenance, but also sets an additional threshold to determine joint maintenance of all other components

(Dinh, Do, and Iung 2020; Zhao et al. 2019; Zheng and Fard 1992; Do, Scarf, and Iung 2015; Huynh, Barros, and Berenguer 2015). However, the maintenance effect depends entirely on the two thresholds, and how to determine these thresholds is very intractable. In addition, this method cannot consider the quality loss when setting the thresholds and it cannot dynamically adjust the thresholds. The second strategy is based on DRL. Compared to the previous strategy, the DRL method directly makes decisions based on the monitored information, without the need to set maintenance thresholds. Specifically, it establishes a direct link between the degradation data of components and maintenance action. Integrating reinforcement learning into the maintenance optimization process makes the maintenance optimization more flexible. Recently, many studies have used DRL to solve maintenance decision problems in multi-component systems (Zhang and Si 2020; Yousefi, Tsianikas, and Coit 2020, 2022). However, these approaches fail to incorporate the influence that quality and production plan exert on maintenance decision-making processes, making their methods unsuitable for our problem setting.

With the above motivations, we investigate the maintenance optimization problem for multi-component production systems by considering quality and production plan. The coupled relations between production plan, quality, and maintenance, as well as the dependence between multiple components, make the aforementioned methods fail to work. For this purpose, a novel DRL-based maintenance optimization method is proposed. In contrast to the existing DRL-based strategy, the decision model is reconstructed. Specifically, we introduce the concept of remaining time of current production batch and analyze its transitions, which contain the information of production plan, to reformulate the state space and state transition function, respectively. Moreover, the health state-related quality loss and production plan-related downtime cost are added to the reward function. With the new decision model and the employment of Deep Q-Network (DQN) algorithm, the corresponding optimal decision problem can then be solved. Furthermore, combining the DRL method with the method based on dual thresholds, Restricted-DQN is proposed to adapt to the actual production environment.

The subsequent structure of the article is as follows: Section 2 introduces the maintenance optimization problem considering quality and production plan; Section 3 introduces the maintenance method based on deep reinforcement learning; Section 4 presents numerical results; Section 5 concludes the article and suggests future research direction.

2. Problem formulation

Consider a production system composed of n components. During the production process, each component undergoes a degradation process. The degradation state can be represented by a physical health indicator or a composite health indicator constructed from different measurement values (Alaswad and Xiang 2017). Each component, labeled i (with i ranging from 1 to n), has its degradation state characterized through a singular random variable, $x_i(t)$. A situation wherein $x_i(t)$ surpasses the predefined failure benchmark, H_i, signifies the failure of component i. Notably, H_i denotes the specific point of failure for the respective component. The holistic system comprises a sequential arrangement of these n components, rendering the system's functionality vulnerable to the slightest malfunction in any one of them. Consequently, the aggregate degradation condition of the system can be encapsulated in a vector-valued random variable, $X(t)$, which encompasses the individual states as $(x_1(t), x_2(t), \dots, x_n(t))$. Assume that the system produces a final product after processing through all components, and the system produces in batches, with different product types produced in different batches. Consider all maintenance as perfect maintenance, reinstating the maintained components to an "as good as new" state. The problem is to formulate an appropriate maintenance strategy.

2.1. Degradation process model

In prior studies, some models have been put forth to emulate degradation process, with the Gamma process standing out as a prevalent choice (Grall, Bérenguer, and Dieulle 2002; van Noortwijk 2009; Thomas 1985). The Gamma process embodies a progressively ascending stochastic process, characterized by independent and non-negative changes, aligning well with modeling degradation phenomena induced by cumulative effects like wear, creep, and corrosion, etc. Hence, in this work, the Gamma process is employed to mimic the deterioration trajectory. Figure 1 visually illustrates this Gamma degradation progression. Formally, assuming the degradation state of component i at moment s as $x_i(s)$, the degradation state at a subsequent time t $(t > s)$ can be mathematically framed as follows

$$x_i(t) = x_i(s) + \Delta x_i(t - s) \tag{1}$$

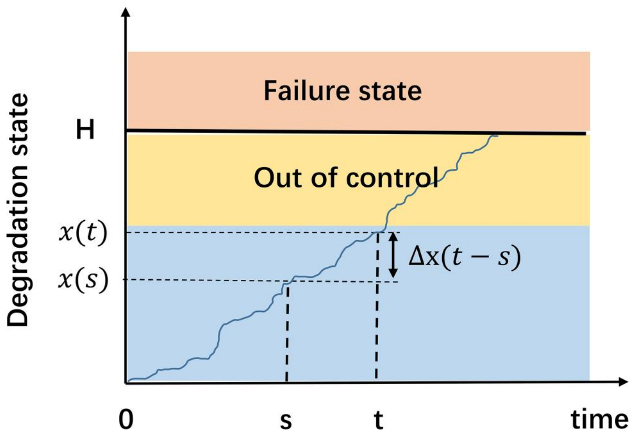

Figure 1. Degradation state of a component.

in which the increment $\Delta x_i(t-s)$ adheres to a Gamma distribution characterized by shape parameter α_i and scale parameter β_i, with its probability density function delineated as:

$$f(x;\alpha_i(t-s),\beta_i)=\frac{\beta_i^{(\alpha_i(t-s))}x^{(\alpha_i(t-s)-1)}\exp(-\beta_i x)}{\Gamma(\alpha_i(t-s))} \quad (2)$$

where $\Gamma(\alpha)=\int_0^\infty z^{\alpha-1}\, exp\,(-z)dz(\alpha>0)$ is the Gamma function.

The parameters α_i and β_i can be estimated through maximum likelihood estimation (MLE) or Bayesian filtering methods (Zhao and Smidts 2022). Compared to MLE, using Bayesian filtering methods can incorporate the prior probability of degradation model parameters into Bayesian inference. Those interested in a broad overview of Bayesian Filtering methodologies could benefit from consulting (Cappe, Godsill, and Moulines 2007; Schirru, Pampuri, and De Nicolao 2010).

2.2. Quality loss model

Before a machine stops working due to a failure, it can be in one of two states: in control and out of control (Duffuaa et al. 2020). Considering the degradation process model described in Section 2.1, the state depends on the degradation state. When the degradation state is far from the failure threshold, the component is in control and works normally; when the degradation state is close to but not exceeding the failure threshold, the component can still work but is out of control. During periods when the machine is out of control, it gives rise to products of inferior quality and can even result in defects. And the closer the degradation state is to the failure threshold, the more severe the component becomes out of control.

The following is a quantitative relationship between quality loss and the system state. Suppose that the quality characteristic, denoted as q, of the products follows a normal distribution characterized by a mean value μ and a standard deviation σ. When the system is out of control, the mean and standard deviation of the quality characteristic may change, so μ and σ are functions of $X(t)$. Given the qualified range as $[\underline{q},\bar{q}]$, that is, the quality is qualified when $q\in[\underline{q},\bar{q}]$, and the quality is unqualified when $q\notin[\underline{q},\bar{q}]$. Therefore, the probability of producing unqualified products $\mathbf{P}(X(t))$ is:

$$\begin{aligned}\mathbf{P}(X(t)) &= \mathbf{P}(q(X(t))\notin[\underline{q},\bar{q}])\\ &= 1-\int_{\underline{q}}^{\bar{q}}\frac{1}{\sigma(X(t))\sqrt{2\pi}}e^{-\frac{(x-\mu(X(t)))^2}{2\sigma(X(t))^2}}dx\end{aligned} \quad (3)$$

The quality loss per unit time is:

$$c_q=\rho\mathbf{P}(X(t)) \quad (4)$$

where ρ is the quality loss coefficient, representing multiplication of the unit time production quantity and the loss caused by producing one defective product.

2.3. Maintenance decision of production system

Developing suitable maintenance strategies becomes crucial to prevent system breakdowns and mitigate the degradation-induced quality loss. It is not possible to decide whether to maintain at every moment. Maintenance activities can only be performed at discrete intervals. That is, a checkpoint is set every interval T_0, and the system is checked to determine whether maintenance should be performed. It is worth noting that when selecting maintenance at a checkpoint, the inspection interval starts counting from the end of maintenance. Maintenance actions can be categorized based on their efficacy into two groups: those with incomplete restoration referred to as imperfect maintenance, and those that fully restore system functionality, known as perfect maintenance. Assuming that the degradation state before maintenance is x_{be} and the degradation state after maintenance is x_{af}. In the case of imperfect maintenance, $0<x_{af}<x_{be}$, and in the case of perfect maintenance, $x_{af}=0$. Whether it is imperfect maintenance or perfect maintenance, it does not affect the decision framework. Therefore, this article assumes that all maintenance actions are perfect maintenance.

The system works in batches, and the production plan can show the number of products in each production batch in the future time interval T. Assuming that the next batch can only be produced after the current batch is completed, that is, it is not allowed to interrupt the current batch and switch to other batches. We define the total cost within the time range T as $\mathbf{E}(C_T)$, including maintenance cost $\mathbf{E}(C_T^m)$ and quality loss $\mathbf{E}(C_T^q)$.

In fact, the maintenance cost can be divided into two parts: c_f and c_d. c_f is the fixed maintenance cost, including parts replacement, personnel cost, and transportation cost. c_d is the downtime cost, which depends on the maintenance duration. Let $a_t = (a_{t,1}, ..., a_{t,n})$ represents the maintenance action at time t, where $a_{t,i} \in \{0, 1\}$. $a_{t,i} = 0$ denotes that maintenance is not performed on component i, and $a_{t,i} = 1$ denotes that maintenance is conducted. Then we have:

$$c_f(a_t) = \sum_{i=1}^{n} c_i a_{t,i} \tag{5}$$

where c_i is the individual operation cost.

The maintenance duration is related to the number of components being maintained, but the time required to maintain multiple components together is less than the total time required to maintain each component separately. The downtime for maintenance also is affected by the production plan. On the one hand, performing maintenance activities takes time, so maintenance activities cause downtime cost for stopping working. On the other hand, the production system produces in batches, and adjustment is required to meet the production conditions of the next batch during batch transfer points. And the downtime cost caused by maintenance can be ignored during the batch transfer points. Then we have:

$$c_d(a_t) = \begin{cases} f(a_{t,1}, ..., a_{t,n}) & t \notin T_c \\ 0 & t \in T_c \end{cases} \tag{6}$$

where $f(a_{t,1}, ..., a_{t,n})$ is the function of downtime cost caused by maintenance of the components, and T_c is the union of all batch transfer intervals.

In summary, at each checkpoint, the optimal maintenance decision needs to be made based on the degradation state, maintenance cost, quality loss and production plan.

3. Maintenance decision based on deep reinforcement learning

3.1. Background of reinforcement learning

Among the three primary machine learning paradigms, Reinforcement Learning (RL) stands out with its objective to identify the most advantageous action scheme for maximizing rewards or minimizing losses within a setup involving agent interacting with the environment. The agent, guided by environmental cues, chooses actions that subsequently yield both a reward and a new environmental state. Consequently, RL naturally aligns with addressing challenges in Markov Decision Problems (MDPs), striving to derive an optimal policy, $\pi : S \rightarrow A$, aimed at accruing maximal cumulative reward over time *via* iterative agent-environment engagement. Applied to maintenance optimization scenarios, the decision-making agent bases its maintenance interventions on the diagnosed condition of the system at each inspection interval. Post-action, the production system evolves to a fresh state, and the agent absorbs this updated state alongside the consequent reward, thus informing future decisions.

3.2. Maintenance decision of multi-component system

The process of making maintenance decisions involves identifying which components require maintenance at each checkpoint, performing maintenance actions, and the system returns the production cost between two checkpoints. This process conforms to the framework of RL as shown in Figure 2. Additionally, the degradation model, the Gamma process, satisfies the Markov property. Therefore, RL is suitable for solving maintenance optimization. The remaining part of this section introduces the elements of RL.

The state space S encompasses the complete collection of all conceivable system states. Obviously, the system state s_t includes the degradation state of the system, which is the main basis for the agent to make decisions. Considering the fact that production plan affects maintenance decisions, the remaining time of the current production batch τ_t (as shown in Figure 3) is added to the system state. Since production duration is approximately proportional to the number of products, τ_t can be set to the remaining number of products in the current batch. Therefore, the system state is:

$$s_t = (x_{1,t}, x_{2,t} ... x_{n,t}, \tau_t) \tag{7}$$

where $x_{i,t}$ is the degradation state of component i.

The action space $A(S)$ comprises all actions available in any given state. At time t, the action $a_t = (a_{t,1}, ..., a_{t,n}) \in A(S)$, and each action affects the next state. In this case, the action is to select the components for maintenance.

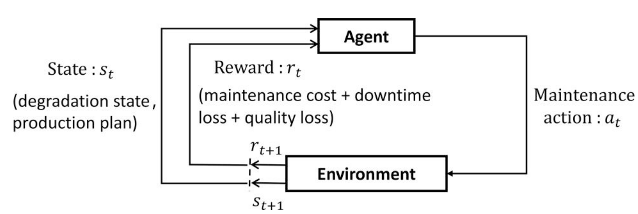

Figure 2. Process of maintenance decision.

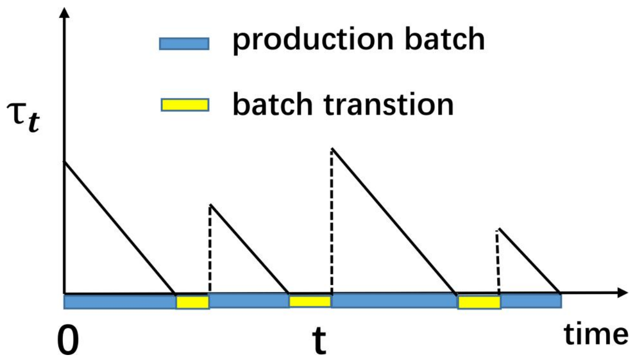

Figure 3. Remaining time of current production batch.

Each component can be maintained or not, so the number of available actions is 2^n.

After determining the state space and action space, it is necessary to determine the state transition process. When maintenance is not selected, the degradation state follows the Gamma process, and τ_t decreases based on the production plan, that is, the remaining number of products for the current batch; when selecting maintenance, the degradation state at time t is dictated by the specific components that are being maintained, and τ_t remains unchanged. It can be seen s_t is determined by s_{t-1} and a_{t-1}, so the Markov property hold.

$$x_{i,t} = \begin{cases} x_{i,t-1} + \Delta x_i & a_{t,i} = 0 \\ 0 & a_{t,i} = 1 \end{cases} \tag{8}$$

where Δx_i follows Gamma process.

$$\tau_t = \begin{cases} \tau_{t-1} & \sum_{i=1}^{n} a_{t,i} = 0 \\ \tau_{t-1} - 1 & \sum_{i=1}^{n} a_{t,i} \neq 0 \quad and \quad \tau_{t-1} > 0 \\ 0 & \sum_{i=1}^{n} a_{t,i} \neq 0, \quad \tau_{t-1} = 0 \quad and \quad o_t = 0 \\ B_t & \sum_{i=1}^{n} a_{t,i} \neq 0, \quad \tau_{t-1} = 0 \quad and \quad o_t = 1 \end{cases} \tag{9}$$

where B_t is batch duration at time t; o_t is working state, that is, $o_t = 0$ at batch transfer point and $o_t = 1$ in production.

The reward function, denoted as $r(s_t, a_t)$, quantifies the reward or loss by the system upon executing action a_t in the context of state s_t. Here, the reward function is a negative value, which represents a loss, including quality loss c_q, fixed maintenance cost c_f, and downtime cost c_d. Therefore, the total reward function is:

$$r(s_t, a_t) = c_q(s_t, a_t) + c_f(s_t, a_t) + c_d(s_t, a_t) \tag{10}$$

where c_q can be obtained from (4), c_f can be obtained from (5), c_d can be obtained from (6). In fact, they are all functions of s_t and a_t (or partially related).

The discount factor, γ, whose value resides within the interval (0, 1), is a common feature in RL frameworks, serving to guarantee the convergence of the potentially infinite series of accumulated rewards. Therefore, the long-term reward function is given by:

$$R_t = \sum_{l=0}^{\infty} \gamma^l r(s_{t+l}, a_{t+l}) \tag{11}$$

3.3. Deep reinforcement learning algorithm

Deep Reinforcement Learning emerges from the synergy of RL principles with deep learning architectures, offering potent approximation abilities and enhanced learning efficiency. This fusion significantly accelerates computation, rendering it particularly apt for handling complex CBM optimizations in high-dimensional spaces. In recent times, the advent of numerous DRL algorithms has invigorated research, with the Deep Q-Network (DQN) algorithm and its subsequent iterations gaining substantial traction due to their remarkable effectiveness in a plethora of practical implementations.

The action-value function $Q_\pi(s_t, a_t)$ and the optimal action-value function $Q^*(s_t, a_t)$ are defined:

$$Q_\pi(s_t, a_t) = \mathbf{E}[R_t | s_t, a_t, \pi] \tag{12}$$

$$Q^*(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t) \tag{13}$$

The optimal policy can be derived from the optimal action-value function:

$$\pi(s_t) = \operatorname*{argmax}_{a_t} Q^*(s_t, a_t) \tag{14}$$

The basic DQN encounters difficulties with overestimated action values, whereas the Double DQN (DDQN) presents a remedy to this predicament. Hence, the DDQN framework is employed.

The DDQN model comprises dual neural networks: an online network governed by parameters θ, and a target network parameterized with θ^-. The former is tasked with choosing actions, whereas the latter plays a pivotal role in assessing the efficacy of the adopted policy. The target value y_t, is formulated as follows:

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \operatorname*{argmax}_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_t); \theta_t^-) \tag{15}$$

The online network parameterized by θ is updated based on the squared loss $(y_t - Q(s_t, a_t; \theta))^2$:

$$\theta_{t+1} = \theta_t + lr(y_t - Q(s_t, a_t; \theta_t))\partial_{\theta_t} Q(s_t, a_t; \theta_t) \quad (16)$$

where lr represents the learning rate.

The training process is depicted in Algorithm 1. After training, the online network is ready to be deployed for directing maintenance-related decision-making processes. At each checkpoint, the degradation states and the remaining time of the current production batch can be obtained. Inputting these values into the online network yields the action values for all maintenance actions. The action corresponding to the maximum action value represents the optimal maintenance action.

Algorithm 1: DDQN algorithm.

Initialize θ arbitrarily and copy θ to θ^-
For each episode:
Initialize the starting state s
For each step within the episode:
Select action a based on a policy derived from $Q(s, a|\theta)$, such as an ϵ-greedy strategy
Execute action a, receive reward r, and observe new state s'
Calculate target value $y_t = r + \gamma Q(s', \text{argmax}_a\ Q(s', a; \theta); \theta^-)$
Update θ *via* a gradient descent step minimizing the loss $(y - Q(s, a; \theta))^2$
Every N steps, synchronize the target network by setting $\theta^- = \theta$

3.4. Restricted-DQN algorithm

In the practical production environment, safety considerations must take precedence in every operation. The primary goal is to ensure the smooth progress of the production process by avoiding any actions deemed risky. However, the algorithm implemented in Section 3.3 relies on deep learning methods, introducing elements of uncertainty. Although favorable results can be achieved through training, it cannot guarantee that the algorithm's actions are absolutely safe at all times. For instance, algorithms may make maintenance decisions while the device is still healthy, or failure to maintain the device that is about to fail. Therefore, the Algorithm 1 may not be feasible in real-world environments.

Algorithm 2: Restricted-DQN algorithm.

Train the DDQN algorithm using Algorithm 1
Obtain state s at the decision checkpoint
Input s into the DDQN algorithm to obtain action a
Determine the maintenance set based on action a
Repeat(for each component i, $i = 1, 2, \ldots, n$)
Judge whether the current component i is in the maintenance set
if *the current component i is in the maintenance set* **then**
  if *the degradation state of the current component* $x_i < H_{\min}$ **then**
    Remove component i from the maintenance set
else
  if *the degradation state of the current component* $x_i > H_{\max}$ **then**
    Add component i to the maintenance set
Return the maintenance set

To adapt to the actual production environment, it is necessary to make some restrictions to the results obtained by the Algorithm 1. Similar to the method based on dual thresholds, two thresholds H_{max} and H_{min} are set to ensure the security of the decision. H_{max} is judiciously established proximate to the failure threshold, thereby ensuring that maintenance actions are promptly initiated when components degrade to the brink of failure, guaranteeing timely intervention; H_{min} is determined based on the point at which system degradation starts to incur measurable quality losses, preventing premature interventions that would otherwise incur unnecessary costs.

Specifically, inputting the degradation states and the remaining time of the current batch into the DQN algorithm, after training, can output corresponding maintenance actions. After obtaining the results of the DQN algorithm, some restrictions can be added. If a component has a low degree of degradation $(d_i < H_{\min})$ and the DQN method suggests maintenance, the maintenance action is canceled. Conversely, if a component has a high degree of degradation $(d_i > H_{\max})$ and the DQN method suggests not maintaining, the maintenance action is increased. For convenience, the method adding restrictions is called Restricted-DQN, the specific process can be seen in Algorithm 2.

4. Numerical study

To validate the efficacy of the proposed approach, this section takes a three-component production system as an example, where each component undergoes a random degradation process. The production duration for each production batch follows a normal distribution $N(70, 5)$. Assuming the checkpoint interval is one unit of time (e.g., hour, day, etc.), the decision to maintain or not is made at each checkpoint. We compare the proposed DRL method with threshold-based opportunistic maintenance.

4.1. Experimental settings

Table 1 displays the parameters of the degradation process, and Figure 4 shows the degradation trajectory, with dashed lines indicating their failure thresholds. We set the same failure threshold for the three components, but their degradation rates are different. Component 3 degrades faster than components 1 and 2, making it more likely to fail than other components, while component 1 is the least likely to fail.

The system will enter an out-of-control state when the degradation process becomes severe, causing quality loss c_q. From (4), it can be seen that the quality loss c_q depends on the quality loss coefficient ρ, the mean μ and variance σ, as well as the upper bound $\bar{q}$ and lower bound $\underline{q}$. For simplicity, two thresholds H_1 and H_2 related to production efficiency are set, and all components are divided into three production states. The production state of each component is:

$$p_i = \begin{cases} 1 & x_i \leq H_1 \\ 2 & H_1 < x_i \leq H_2 \\ 3 & H_2 < x_i \end{cases} \tag{17}$$

The production state of the system is $p = \max(p_1, p_2, p_3)$. We assume that the product quality characteristic is determined by the component with the worst degradation state, that is, it is related to p, so the quality loss c_q is determined by p. In the experiment, we set H_1 to 28 and H_2 to 35. Assuming that when $p = 1$, the system is in good health and there is no production loss; when $p = 2$, the system is slightly out of control, causing a production loss of 0.1 per unit of time; and when $p = 3$, the system is severely out of control, causing a production loss of 0.2 per unit of time.

The maintenance-related costs include two parts: fixed maintenance cost c_f and downtime cost c_d. For the simulation, we assign a maintenance cost of 2.5 to each individual component. Furthermore, the downtime cost c_d is linked to the duration of system unavailability. Downtime includes maintenance time and production system adjustment time. The duration of maintenance is directly proportional to the quantity of components undergoing maintenance. Production system adjustment time represents the time required to interrupt and resume production for each maintenance action. We set the time cost of adjusting the production system for each maintenance action to be 2.0, and the maintenance time cost to be 1.5, that is, $f(a_{t,1}, \ldots, a_{t,n})$ is:

$$f(a_{t,1}, \ldots, a_{t,n}) = \begin{cases} 3.5 & \sum_i a_{t,i} = 1 \\ 5.0 & \sum_i a_{t,i} = 2 \\ 6.5 & \sum_i a_{t,i} = 3 \end{cases} \tag{18}$$

Within the implemented DQN algorithm, the architecture of the neural network incorporates two successive hidden layers, comprising 28 nodes in the initial layer and expanding to 36 nodes in the subsequent layer. The learning rate is set to 0.001, the greedy selection probability ϵ is 0.945, and the discount factor γ is 0.993.

Table 1. Parameters of degradation process for three-component.

Parameter	Component 1	Component 2	Component 3
α	0.4	0.2	0.2
β	1.2	1.4	1.0

4.2. A three-component system

Three scenarios are simulated to compare different maintenance approaches, each with six production batches. In the experimental results, OM represents the threshold-based opportunistic maintenance method. The optimal maintenance strategy is obtained by exhaustive search. When simulating maintenance decisions, the degradation process of three components within 400 time steps is determined randomly in advance. After fixing the degradation process of all components, it can be ensured that different maintenance strategies are all experimented under the same degradation scenario and that the actual optimal maintenance strategy can be obtained through exhaustive search.

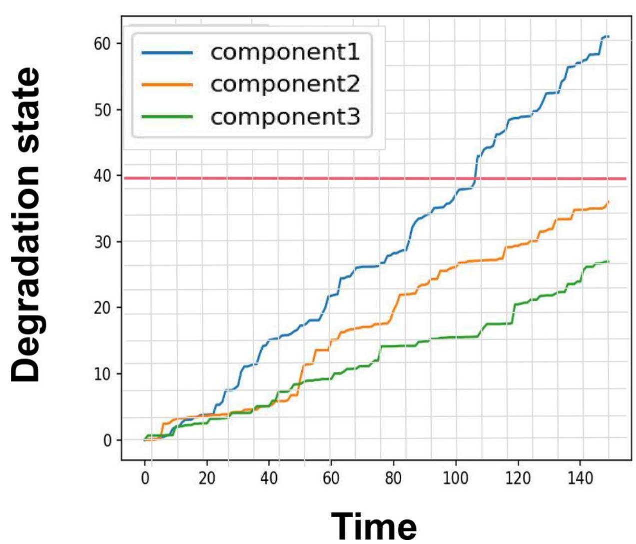

Figure 4. Degradation process of three components.

In scenario 1, the batch sizes are {54,75,58,74,63,72}, and the total production time is 396. The results are shown in Table 2 and Figure 5. The DQN method is basically the same as the optimal maintenance strategy, and the DQN method can learn a policy that maintains multiple components simultaneously without providing the algorithm with the concept of opportunistic maintenance. Comparing the method with the threshold-based opportunistic maintenance, the main difference lies in the choice of whether to maintain between the second and third batches of production. At the batch transfer point, the degradation states of the three components were 27, 31, and 30, respectively, and the system had already been slightly out of control state. Therefore, simultaneous maintenance of the three components can avoid quality loss, and maintenance during the batch transfer point can avoid downtime cost. The threshold-based opportunistic maintenance did not make good use of this opportunity.

In Scenario 2, the batch sizes are {72,63,87,64,58,79} and the total production time is 423. The results are shown in Table 2 and Figure 6. Overall, the DQN method is still closer to the optimal maintenance strategy. The threshold-based opportunistic maintenance not only has the drawback of not fully utilizing the planned downtime opportunity during batch transition, but also has the defect of being too conservative. For example, the optimal maintenance strategy is to maintain component 1 at the end of the first batch. Although earlier maintenance can reduce quality loss, maintenance during production process results in more severe downtime cost. Compared to opportunistic maintenance, the DQN method can consider the balance between quality loss and downtime cost and makes better choices. Moreover, from Figure 6, it can be calculated that the DQN method and the optimal maintenance strategy maintained 5 times, while the opportunistic maintenance maintained 8 times, which also reflects the superiority of the proposed method.

Table 2. The cost of different strategies.

	Optimal Maintenance	OM	DQN	Restricted-DQN
Scenario 1	35.0	62.5	36.6	—
Scenario 2	42.0	72.3	44.2	—
Scenario 3	44.1	—	63.1	44.1

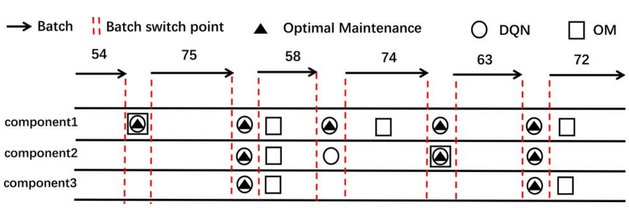

Figure 5. Maintenance schedule for scenario 1.

From scenario 1 and scenario 2, it can be seen that the DQN method is superior to the opportunistic maintenance method, but the DQN method is still inferior to the optimal maintenance strategy. There are two main reasons; one reason is that the DQN method gets trapped in local maxima. At some batch transfer points, the system is no need to be maintained, but the DQN method chooses maintenance. This is because the DQN method is trapped in the local optimal strategy of choosing maintenance as much as possible every time a batch is transferred. Some components are in good health, but the DQN method wrongly chooses to maintain them. The other reason is the limitation of the DQN algorithm itself. The DQN algorithm uses neural networks to fit the relationship between system state and maintenance actions, but deep learning is inherently uncertain, and it is not possible to guarantee that the results are accurate for any system state.

To address the above-mentioned gap, some restrictions need to be added to the DQN method. If a component has a low degree of degradation and the proposed method chooses to maintain it, the maintenance action is canceled. Conversely, if a component has a high degree of degradation and the proposed method chooses not to maintain it, the maintenance action is increased. A comparison experiment of the DQN method and Restricted-DQN was conducted in scenario 3, and the results are shown in Table 2 and Figure 7. The DQN method completely ignored component 3 throughout the entire time period, resulting in huge corrective maintenance cost. And it was trapped in a local maximum at the moment of batch transfer point in the fourth and fifth batches. Restricted-DQN, on the other hand, alleviated the above issues and made the optimal maintenance strategy in scenario 3.

Table 3 shows the mean and variance of the results of 100 experiments. The maximum cost occurs when no

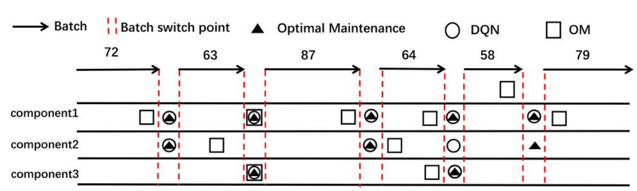

Figure 6. Maintenance schedule for scenario 2.

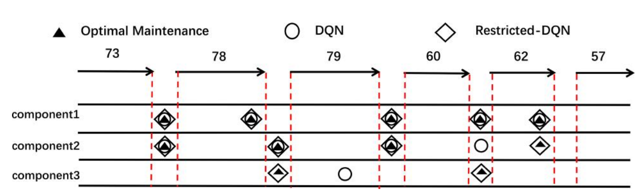

Figure 7. Maintenance schedule for scenario 3.

maintenance is performed. While the proposed method yields a lesser mean cost compared to the opportunistic maintenance strategy, it is characterized by a heightened degree of variance. That is, the stability of the DQN method is insufficient, and the Restricted-DQN not only has the lowest average cost, but also significantly reduces the variance compared to the original method.

4.3. Sensitivity analysis

We consider systems with five components, eight components and ten components. For the five-component system, there are $2^5 = 32$ actions, the number of neurons are extended to (60, 100) for more accurate estimations; For the eight-component system, there are $2^8 = 256$ actions, the number of neurons are extended to (100, 200); For the ten-component system, there are $2^{10} = 1024$ actions, the number of neurons are extended to (300, 300). The parameters of the Gamma process α_i follows $U(0.3, 0.1)$ and β_i follows $U(1.3, 0.1)$.

The costs of policies Restricted-DQN, DQN, the threshold-based opportunistic maintenance and corrective maintenance were computed for comparison purposes. The results are shown in Tables 4–6. For both five-component, eight-component and ten-component systems, the proposed Restricted-DQN policy obtain the lowest mean cost and reaches a low level of variance. The proposed method reduces costs by 16.07 for a five-component system, by 32.92 for a eight-component system, and by 40.73 for a ten-component system, all compared to threshold-based opportunistic maintenance. As the component count rises, the proposed methodology exhibits a more significant improvement compared to the threshold-based opportunistic maintenance, which indicates the effectiveness and scalability of the proposed method.

It is worth noting that in this work, the action space grows at an exponential rate as the component count rises; specifically, the action space for an n-component system is 2^n. Due to inherent limitations of the DQN algorithm, it is not suitable for managing large-scale state spaces. To address systems of larger scales, reinforcement learning algorithms tailored for substantial action spaces, such as BDQ and DDPG, can be employed. Fundamentally, these algorithms diverge primarily in their model architectures and loss functions from DQN, yet they remain compatible with the proposed decision framework.

To delve deeper into the efficacy of model in achieving minimal long-term average costs, the time span is increased from 400 to 4000. Table 7 shows the results. For systems with different numbers of components, the proposed method is still effective over a long time span.

Table 3. Mean and standard deviation of cost.

	Restricted-DQN	DQN	OM	CM
Mean	−52.41	−53.27	−66.28	−128.16
Standard deviation	78.75	91.32	56.88	140.86

Table 4. Results for five components.

	Restricted-DQN	DQN	OM	CM
Mean	**−64.81**	−66.43	−80.88	−367.37
Standard deviation	61.02	139.50	**57.78**	480.60

Bold values indicate the best number in the same row.

Table 5. Results for eight components.

	Restricted-DQN	DQN	OM	CM
Mean	**−93.76**	−101.42	−126.68	−504.0
Standard deviation	286.83	489.65	**142.89**	1121.5

Bold values indicate the best number in the same row.

Table 6. Results for ten components.

	Restricted-DQN	DQN	OM	CM
Mean	**−329.29**	−333.34	−370.02	−1122.71
Standard deviation	631.85	878.55	**531.11**	2525.35

Bold values indicate the best number in the same row.

Table 7. Cost for long time span.

	Restricted-DQN	DQN	OM	CM
five-component	**−647.00**	−667.08	−808.34	−3687.95
eight-component	**−922.65**	−980.05	−1972.6	−5037.5
ten-component	**−3692.79**	−3722.45	−4096.96	−12337.39

Bold values indicate the best number in the same row.

5. Conclusion

As modern production systems become increasingly complex, the design and optimization of maintenance for multi-component systems become challenging. With the escalation in the number of components, conventional threshold-based maintenance strategies get computationally burdensome and increasingly challenging to manage effectively. On the other hand, methods based on DRL rarely consider the relationship between maintenance, quality and production plan. To overcome these challenges, this article proposes a method that models the system degradation process using Gamma process. Then, the information of production batches is integrated into the system state, and product quality is considered. By leveraging the powerful ability of DRL, we significantly improve maintenance decision performance.

The proposed CBM methodology did not encompass ecological considerations, despite the intensifying urgency of energy and environmental challenges. As global understanding of ecological impacts deepens, future endeavors will delve into exploring maintenance strategies with a heightened focus on sustainability.

About the authors

Ming Chen received his BS degree from the University of Science and Technology of China (USTC), Hefei, China in 2021. He is currently pursuing the M.S. degree with the Department of Automation, USTC. His current research interests include maintenance optimization and reinforcement learning.

Yu Kang received the PhD degree in control theory and control engineering from the University of Science and Technology of China, Hefei, China, in 2005. From 2005 to 2007, he was a Postdoctoral Fellow with the Academy of Mathe- matics and Systems Science, Chinese Academy of Sciences. He is currently a Professor with the Department of Automation and the Institute of Advanced Technology, University of Science and Technology of China. His current research interests in- clude monitoring of vehicle emissions, fault prediction and diagnosis and maintenance.

Kun Li received the PhD degree in control science and engineer- ing from University of Science and Technology of China (USTC), Hefei, China, in 2019. He is currently a Postdoctoral Fellow with Institute of Advanced Technology, USTC. His research interests include nonlinear control theory and its application, robot control system.

Pengfei Li received the PhD degree in control science and engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2020. He is an Associate Research Fellow with the School of Information Science and Technology, USTC. His research interests include networked control systems, model predictive control, and learning based control.

Yun-Bo Zhao received his B.Sc. degree in mathematics from Shan- dong University, Jinan, China in 2003, M.Sc. degree in systems sciences from the Key Laboratory of Systems and Control, Chinese Academy of Sciences, Beijing, China in 2007, and Ph.D. degree in control engineering from the University of South Wales (formerly University of Glamorgan), Pontypridd, UK in 2008, respectively. He is cur- rently a Professor with University of Science and Technology of China, Hefei, China. He is mainly interested in AI-driven control and automation, specifically, AI-driven networked intelligent control, AI-driven humanmachine autonomies and AI-driven ma- chine gaming.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the National Key Research and Development Program of China (No. 2022YFB3206904), and the National Natural Science Foundation of China (62033012, 62103394 and 62173317).

References

Alaswad, S., and Y. Xiang. 2017. A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering & System Safety* 157:54–63. doi: 10.1016/j.ress.2016.08.009.

Beheshti Fakher, H., M. Nourelfath, and M. Gendreau. 2017. A cost minimisation model for joint production and maintenance planning under quality constraints. *International Journal of Production Research* 55 (8):2163–76. doi: 10.1080/00207543.2016.1201605.

Bian, L., and N. Gebraeel. 2014. Stochastic modeling and real-time prognostics for multi-component systems with degradation rate interactions. *IIE Transactions* 46 (5): 470–82. doi: 10.1080/0740817X.2013.812269.

Budai, G., R. Dekker, and R. Nicolai. 2006. *A review of planning models for maintenance & product.* The Nederlands.

Cappe, O., S. J. Godsill, and E. Moulines. 2007. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE* 95 (5):899–924. doi: 10.1109/JPROC.2007.893250.

Dao, C. D., and M. J. Zuo. 2017. Selective maintenance of multi-state systems with structural dependence. *Reliability Engineering & System Safety* 159:184–95. doi: 10.1016/j.ress.2016.11.013.

Dinh, D.-H., P. Do, and B. Iung. 2020. Maintenance optimisation for multi-component system with structural dependence: Application to machine tool sub-system. *CIRP Annals* 69 (1):417–20. doi: 10.1016/j.cirp.2020.04.004.

Do, P., and C. Bérenguer. 2020. Conditional reliability-based importance measures. *Reliability Engineering & System Safety* 193:106633. doi: 10.1016/j.ress.2019.106633.

Do, P., P. Scarf, and B. Iung. 2015. Condition-based maintenance for a two-component system with dependencies. *IFAC-PapersOnLine* 48 (21):946–51. doi: 10.1016/j.ifacol.2015.09.648.

Duffuaa, S., A. Kolus, U. Al-Turki, and A. El-Khalifa. 2020. An integrated model of production scheduling, maintenance and quality for a single machine. *Computers & Industrial Engineering* 142:106239. doi: 10.1016/j.cie.2019.106239.

Farahani, A., and H. Tohidi. 2021. Integrated optimization of quality and maintenance: A literature review. *Computers & Industrial Engineering* 151:106924. doi: 10.1016/j.cie.2020.106924.

Ghamlouch, H., M. Fouladirad, and A. Grall. 2019. The use of real option in condition-based maintenance scheduling for wind turbines with production and deterioration uncertainties. *Reliability Engineering & System Safety* 188: 614–23. doi: 10.1016/j.ress.2017.10.001.

Grall, A., C. Bérenguer, and L. Dieulle. 2002. A condition-based maintenance policy for stochastically deteriorating systems. *Reliability Engineering & System Safety* 76 (2): 167–80. doi: 10.1016/S0951-8320(01)00148-X.

Hu, J., Z. Jiang, and H. Liao. 2017. Preventive maintenance of a batch production system under time-varying operational condition. *International Journal of Production Research* 55 (19):5681–705. doi: 10.1080/00207543.2017.1330565.

Huynh, K. T., A. Barros, and C. Berenguer. 2015. Multi-level decision-making for the predictive maintenance of k-out-of-n: F deteriorating systems. *IEEE Transactions on Reliability* 64 (1):94–117. doi: 10.1109/TR.2014.2337791.

Keizer, M. C. A. O., S. D. P. Flapper, and R. H. Teunter. 2017. Condition-based maintenance policies for systems with multiple dependent components: A review. *European Journal of Operational Research* 261 (2):405–20. doi: 10.1016/j.ejor.2017.02.044.

Lee, H. L., and M. J. Rosenblatt. 1987. Simultaneous determination of production cycle and inspection schedules in a production system. *Management Science* 33 (9):1125–36. doi: 10.1287/mnsc.33.9.1125.

Levrat, E., E. Thomas, and B. Iung. 2008. Odds-based decision-making tool for opportunistic production-maintenance synchronization. *International Journal of Production Research* 46 (19):5263–87. doi: 10.1080/00207540802273793.

Liu, J., M. Zhang, H. Zuo, and J. Xie. 2014. Remaining useful life prognostics for aeroengine based on superstatistics and information fusion. *Chinese Journal of Aeronautics* 27 (5):1086–96. doi: 10.1016/j.cja.2014.08.013.

Nicolai, R., and R. Dekker. 2006. "*Optimal maintenance of multi-component systems: A review*." Report/Econometric Institute, Erasmus University Rotterdam (EI 2006–29).

Nourelfath, M., N. Nahas, and M. Ben-Daya. 2016. Integrated preventive maintenance and production decisions for imperfect processes. *Reliability Engineering & System Safety* 148:21–31. doi: 10.1016/j.ress.2015.11.015.

Parvizsedghy, L., A. Senouci, T. Zayed, S. F. Mirahadi, and M. S. El-Abbasy. 2015. Condition-based maintenance decision support system for oil and gas pipelines. *Structure and Infrastructure Engineering* 11 (10):1323–37. doi: 10.1080/15732479.2014.964266.

Rahim, M. A., and M. Ben-Daya. 1998. A generalized economic model for joint determination of production run, inspection schedule and control chart design. *International Journal of Production Research* 36 (1):277–89. doi: 10.1080/002075498194047.

Schirru, A., S. Pampuri, and G. De Nicolao. 2010. Particle filtering of hidden gamma processes for robust predictive maintenance in semiconductor manufacturing. In *2010 IEEE International Conference on Automation Science and Engineering*, 51–6. IEEE.

Thomas, D. SUS Department of Commerce, National Institute of Standards and Technology …., and D. S. Thomas. 2018. *The costs and benefits of advanced maintenance in manufacturing*.

Thomas, L. C. 1985. Replacement of systems and components in renewal decision problems. *Operations Research* 33 (2):404–11. doi: 10.1287/opre.33.2.404.

Tian, Z., and H. Liao. 2011. Condition based maintenance optimization for multi-component systems using proportional hazards model. *Reliability Engineering & System Safety* 96 (5):581–9. doi: 10.1016/j.ress.2010.12.023.

van Noortwijk, J. M. 2009. A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety* 94 (1):2–21. doi: 10.1016/j.ress.2007.03.019.

Xia, T., X. Jin, L. Xi, and J. Ni. 2015. Production-driven opportunistic maintenance for batch production based on MAM–APB scheduling. *European Journal of Operational Research* 240 (3):781–90. doi: 10.1016/j.ejor.2014.08.004.

Yousefi, N., S. Tsianikas, and D. W. Coit. 2020. Reinforcement learning for dynamic condition-based maintenance of a system with individually repairable components. *Quality Engineering* 32 (3):388–408. doi: 10.1080/08982112.2020.1766692.

Yousefi, N., S. Tsianikas, and D. W. Coit. 2022. Dynamic maintenance model for a repairable multi-component system using deep reinforcement learning. *Quality Engineering* 34 (1):16–35. doi: 10.1080/08982112.2021.1977950.

Zhang, N., and W. Si. 2020. Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. *Reliability Engineering & System Safety* 203:107094. doi: 10.1016/j.ress.2020.107094.

Zhao, H., F. Xu, B. Liang, J. Zhang, and P. Song. 2019. A condition-based opportunistic maintenance strategy for multi-component system. *Structural Health Monitoring* 18 (1):270–83. doi: 10.1177/1475921717751871.

Zhao, Y., and C. Smidts. 2022. Reinforcement learning for adaptive maintenance policy optimization under imperfect knowledge of the system degradation model and partial observability of system states. *Reliability Engineering & System Safety* 224:108541. doi: 10.1016/j.ress.2022.108541.

Zheng, X., and N. Fard. 1992. Hazard-rate tolerance method for an opportunistic-replacement policy. *IEEE Transactions on Reliability* 41 (1):13–20. doi: 10.1109/24.126664.

Zhou, P., and P. T. Yin. 2019. An opportunistic condition-based maintenance strategy for offshore wind farm based on predictive analytics. *Renewable and Sustainable Energy Reviews* 109:1–9. doi: 10.1016/j.rser.2019.03.049.